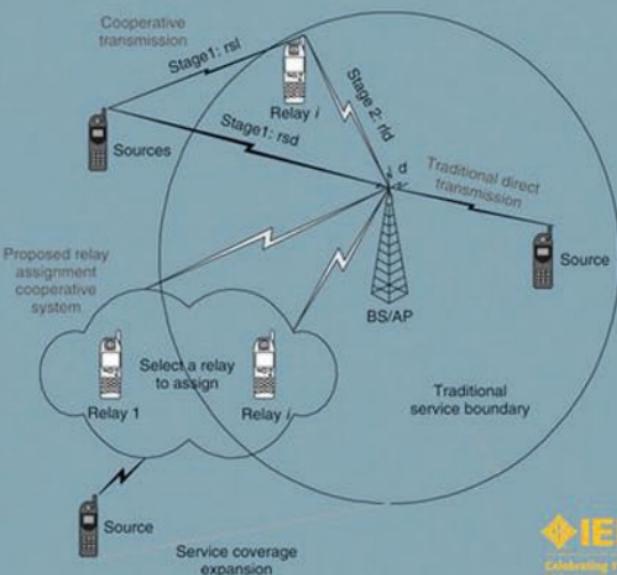


# HANDBOOK ON ARRAY PROCESSING AND SENSOR NETWORKS

SIMON HAYKIN • K. J. RAY LIU





# **HANDBOOK ON ARRAY PROCESSING AND SENSOR NETWORKS**



---

# HANDBOOK ON ARRAY PROCESSING AND SENSOR NETWORKS

---

Simon Haykin  
K. J. Ray Liu



Celebrating 125 Years  
*of Engineering the Future*



A JOHN WILEY & SONS, INC., PUBLICATION

Copyright © 2009 by John Wiley & Sons, Inc. All rights reserved

Published by John Wiley & Sons, Inc., Hoboken, New Jersey  
Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at [www.copyright.com](http://www.copyright.com). Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Limit of Liability/Disclaimer of Warranty:** While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at [www.wiley.com](http://www.wiley.com).

***Library of Congress Cataloging-in-Publication Data:***

Haykin, Simon

Handbook on array processing and sensor networks / Simon Haykin, K. J. Ray Liu.

p. cm.

Includes bibliographical references and index.

ISBN 978-0-470-37176-3 (cloth)

1. Sensor networks. 2. Antenna arrays. 3. Array processors. I. Liu, K. J. Ray, 1961- II. Title.  
TK7872.D48H39 2009

621.382'4—dc22

2008055880

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

## CONTENTS

---

<b>Preface</b>	xiii
<i>Simon Haykin and K. J. Ray Liu</i>	
<b>Contributors</b>	xv
<b>Introduction</b>	1
<i>Simon Haykin</i>	
<b>PART I: FUNDAMENTAL ISSUES IN ARRAY SIGNAL PROCESSING</b>	9
<b>1 Wavefields</b>	11
<i>Alfred Hanssen</i>	
1.1 Introduction	11
1.2 Harmonizable Stochastic Processes	12
1.3 Stochastic Wavefields	15
1.4 Wave Dispersion	19
1.5 Conclusions	26
Acknowledgments	27
References	27
<b>2 Spatial Spectrum Estimation</b>	29
<i>Petar M. Djurić</i>	
2.1 Introduction	29
2.2 Fundamentals	33
2.3 Temporal Spectrum Estimation	34
2.4 Spatial Spectrum Estimation	41
2.5 Final Remarks	56
References	56
<b>3 MIMO Radio Propagation</b>	59
<i>Tricia J. Willink</i>	
3.1 Introduction	59
3.2 Space–Time Propagation Environment	60
3.3 Propagation Models	64
3.4 Measured Channel Characteristics	75
3.5 Stationarity	81

3.6 Summary	86
References	87
<b>4 Robustness Issues in Sensor Array Processing</b>	<b>91</b>
<i>Alex B. Gershman</i>	
4.1 Introduction	91
4.2 Direction-of-Arrival Estimation	92
4.3 Adaptive Beamforming	102
4.4 Conclusions	107
Acknowledgments	108
References	108
<b>5 Wireless Communication and Sensing in Multipath Environments Using Multiantenna Transceivers</b>	<b>115</b>
<i>Akbar M. Sayeed and Thiagarajan Sivanadyan</i>	
5.1 Introduction and Overview	115
5.2 Multipath Wireless Channel Modeling in Time, Frequency, and Space	118
5.3 Point-to-Point MIMO Wireless Communication Systems	133
5.4 Active Wireless Sensing with Wideband MIMO Transceivers	156
5.5 Concluding Remarks	165
References	166
<b>PART II: NOVEL TECHNIQUES FOR AND APPLICATIONS OF ARRAY SIGNAL PROCESSING</b>	<b>171</b>
<b>6 Implicit Training and Array Processing for Digital Communication Systems</b>	<b>173</b>
<i>Aldo G. Orozco-Lugo, Mauricio Lara and Desmond C. McLernon</i>	
6.1 Introduction	173
6.2 Classification of Implicit Training Methods	180
6.3 IT-Based Estimation for a Single User	186
6.4 IT-Based Estimation for Multiple Users Exploiting Array Processing: Continuous Transmission	191
6.5 IT-Based Estimation for Multiple Users Exploiting Array Processing: Packet Transmission	199
6.6 Open Research Problems	201
Acknowledgments	203
References	203
<b>7 Unitary Design of Radar Waveform Diversity Sets</b>	<b>211</b>
<i>Michael D. Zoltowski, Tariq R. Qureshi, Robert Calderbank and Bill Moran</i>	
7.1 Introduction	211

7.2	2 × 2 Space–Time Diversity Waveform Design	213
7.3	4 × 4 Space–Time Diversity Waveform Design	217
7.4	Waveform Families Based on Kronecker Products	220
7.5	Introduction to Data-Dependent Waveform Design	226
7.6	3 × 3 and 6 × 6 Waveform Scheduling	228
7.7	Summary	229
	References	229
<b>8</b>	<b>Acoustic Array Processing for Speech Enhancement</b>	<b>231</b>
	<i>Markus Buck, Eberhard Hänsler, Mohamed Krini, Gerhard Schmidt and Tobias Wolff</i>	
8.1	Introduction	231
8.2	Signal Processing in Subband Domain	233
8.3	Multichannel Echo Cancellation	236
8.4	Speaker Localization	240
8.5	Beamforming	242
8.6	Sensor Calibration	249
8.7	Postprocessing	252
8.8	Conclusions	264
	References	264
<b>9</b>	<b>Acoustic Beamforming for Hearing Aid Applications</b>	<b>269</b>
	<i>Simon Doclo, Sharon Gannot, Marc Moonen and Ann Spriet</i>	
9.1	Introduction	269
9.2	Overview of noise reduction techniques	270
9.3	Monaural beamforming	272
9.4	Binaural beamforming	286
9.5	Conclusion	296
	References	296
<b>10</b>	<b>Underdetermined Blind Source Separation Using Acoustic Arrays</b>	<b>303</b>
	<i>Shoji Makino, Shoko Araki, Stefan Winter and Hiroshi Sawada</i>	
10.1	Introduction	303
10.2	Underdetermined Blind Source Separation of Speeches in Reverberant Environments	305
10.3	Sparseness of Speech Sources	307
10.4	Binary Mask Approach to Underdetermined BSS	312
10.5	MAP-Based Two-Stage Approach to Underdetermined BSS	321
10.6	Experimental Comparison with Binary Mask Approach and MAP-Based Two-Stage Approach	328
10.7	Concluding Remarks	335
	References	337

<b>11</b>	<b>Array Processing in Astronomy</b>	<b>343</b>
<i>Douglas C.-J. Bock</i>		
11.1	Introduction	343
11.2	Correlation Arrays	343
11.3	Aperture Plane Phased Arrays	361
11.4	Future Directions	362
11.5	Conclusion	364
References		365
<b>12</b>	<b>Digital 3D/4D Ultrasound Imaging Array</b>	<b>367</b>
<i>Sergios Stergiopoulos</i>		
12.1	Background	367
12.2	Next-Generation 3D/4D Ultrasound Imaging Technology	372
12.3	Computing Architecture and Implementation Issues	392
12.4	Experimental Planar Array Ultrasound Imaging System	394
12.5	Conclusion	403
References		404
<b>PART III: FUNDAMENTAL ISSUES IN DISTRIBUTED SENSOR NETWORKS</b>		<b>407</b>
<b>13</b>	<b>Self-Localization of Sensor Networks</b>	<b>409</b>
<i>Joshua N. Ash and Randolph L. Moses</i>		
13.1	Introduction	409
13.2	Measurement Types and Performance Bounds	411
13.3	Localization Algorithms	420
13.4	Relative and Transformation Error Decomposition	427
13.5	Conclusions	434
References		435
<b>14</b>	<b>Multitarget Tracking and Classification in Collaborative Sensor Networks via Sequential Monte Carlo Methods</b>	<b>439</b>
<i>Tom Vercauteren and Xiaodong Wang</i>		
14.1	Introduction	439
14.2	System Description and Problem Formulation	440
14.3	Sequential Monte Carlo Methods	446
14.4	Joint Single-Target Tracking and Classification	448
14.5	Multiple-Target Tracking and Classification	452
14.6	Sensor Selection	456
14.7	Simulation Results	459
14.8	Conclusion	464

Appendix: Derivations of (14.38) and (14.40)	465
References	466
<b>15 Energy-Efficient Decentralized Estimation</b>	<b>469</b>
<i>Jin-Jun Xiao, Shuguang Cui and Zhi-Quan Luo</i>	
15.1 Introduction	469
15.2 System Model	471
15.3 Digital Approaches	472
15.4 Analog Approaches	476
15.5 Analog versus Digital	485
15.6 Extension to Vector Model	487
15.7 Concluding Remarks	492
Acknowledgments	494
References	494
<b>16 Sensor Data Fusion with Application to Multitarget Tracking</b>	<b>499</b>
<i>R. Tharmarasa, K. Punithakumar, T. Kirubarajan and Y. Bar-Shalom</i>	
16.1 Introduction	499
16.2 Tracking Filters	500
16.3 Data Association	511
16.4 Out-of-Sequence Measurements	521
16.5 Results with Real Data	524
16.6 Summary	527
References	527
<b>17 Distributed Algorithms in Sensor Networks</b>	<b>533</b>
<i>Usman A. Khan, Soummya Kar and José M. F. Moura</i>	
17.1 Introduction	533
17.2 Preliminaries	535
17.3 Distributed Detection	538
17.4 Consensus Algorithms	539
17.5 Zero-Dimension (Average) Consensus	542
17.6 Consensus in Higher Dimensions	544
17.7 Leader–Follower (Type) Algorithms	545
17.8 Localization in Sensor Networks	548
17.9 Linear System of Equations: Distributed Algorithm	551
17.10 Conclusions	553
References	553
<b>18 Cooperative Sensor Communications</b>	<b>559</b>
<i>Ahmed K. Sadek, Weifeng Su and K. J. Ray Liu</i>	
18.1 Introduction	559

18.2	Cooperative Relay Protocols	561
18.3	SER Analysis and Optimal Power Allocation	568
18.4	Energy Efficiency in Cooperative Sensor Networks	589
18.5	Experimental Results	599
18.6	Conclusions	606
	References	606
<b>19</b>	<b>Distributed Source Coding</b>	<b>609</b>
	<i>Zixiang Xiong, Angelos D. Liveris and Yang Yang</i>	
19.1	Introduction	609
19.2	Theoretical Background	610
19.3	Code Designs	619
19.4	Applications	631
19.5	Conclusions	638
	References	639
<b>20</b>	<b>Network Coding for Sensor Networks</b>	<b>645</b>
	<i>Christina Fragouli</i>	
20.1	Introduction	645
20.2	How Can We Implement Network Coding in a Practical Sensor Network?	649
20.3	Data Collection and Coupon Collector Problem	653
20.4	Distributed Storage and Sensor Network Data Persistence	657
20.5	Decentralized Operation and Untuned Radios	660
20.6	Broadcasting and Multipath Diversity	662
20.7	Network, Channel, and Source Coding	663
20.8	Identity-Aware Sensor Networks	664
20.9	Discussion	666
	Acknowledgments	666
	References	666
<b>21</b>	<b>Information-Theoretic Studies of Wireless Sensor Networks</b>	<b>669</b>
	<i>Liang-Liang Xie and P. R. Kumar</i>	
21.1	Introduction	669
21.2	Information-Theoretic Studies	670
21.3	Relay Schemes	674
21.4	Wireless Network Coding	684
21.5	Concluding Remarks	688
	Acknowledgments	689
	References	689

<b>PART IV: NOVEL TECHNIQUES FOR AND APPLICATIONS OF DISTRIBUTED SENSOR NETWORKS</b>	<b>693</b>
<b>22 Distributed Adaptive Learning Mechanisms</b>	<b>695</b>
<i>Ali H. Sayed and Federico S. Cattivelli</i>	
22.1 Introduction	695
22.2 Motivation	697
22.3 Incremental Adaptive Solutions	698
22.4 Diffusion Adaptive Solutions	707
22.5 Concluding Remarks	720
Acknowledgments	721
References	721
<b>23 Routing for Statistical Inference in Sensor Networks</b>	<b>723</b>
<i>A. Anandkumar, A. Ephremides, A. Swami and L. Tong</i>	
23.1 Introduction	723
23.2 Spatial Data Correlation	724
23.3 Statistical Inference of Markov Random Fields	730
23.4 Optimal Routing for Inference with Local Processing	731
23.5 Conclusion and Future Work	744
23.6 Bibliographic Notes	745
References	745
<b>24 Spectral Estimation in Cognitive Radios</b>	<b>749</b>
<i>Behrouz Farhang-Boroujeny</i>	
24.1 Filter Bank Formulation of Spectral Estimators	750
24.2 Polyphase Realization of Uniform Filter Banks	751
24.3 Periodogram Spectral Estimator	752
24.4 Multitaper Spectral Estimator	757
24.5 Filter Bank Spectral Estimator	766
24.6 Distributed Spectrum Sensing	773
24.7 Discussion	776
Appendix A: Effective Degree of Freedom	777
Appendix B: Explanation to the Results of Table 24.1	779
References	779
<b>25 Nonparametric Techniques for Pedestrian Tracking in Wireless Local Area Networks</b>	<b>783</b>
<i>Azadeh Kushki and Konstantinos N. Plataniotis</i>	
25.1 Introduction	783

25.2 WLAN Positioning Architectures	785
25.3 Signal Models	786
25.4 Zero-Memory Positioning	788
25.5 Dynamic Positioning Systems	790
25.6 Cognition and Feedback	796
25.7 Tracking Example	799
25.8 Conclusions	801
References	801
<b>26 Reconfigurable Self-Activating Ion-Channel-Based Biosensors</b>	<b>805</b>
<i>Vikram Krishnamurthy and Bruce Cornell</i>	
26.1 Introduction	805
26.2 Biosensors Built of Ion Channels	807
26.3 Joint Input Excitation Design and Concentration Classification for Biosensor	812
26.4 Decentralized Deployment of Dense Network of Biosensors	816
26.5 Discussion and Extensions	826
References	827
<b>27 Biochemical Transport Modeling, Estimation, and Detection in Realistic Environments</b>	<b>831</b>
<i>Mathias Ortner and Arye Nehorai</i>	
27.1 Introduction	831
27.2 Physical and Statistical Models	832
27.3 Transport Modeling Using Monte Carlo Approximation	835
27.4 Localizing the Source(s)	843
27.5 Sequential Detection	846
27.6 Conclusion	849
References	851
<b>28 Security and Privacy for Sensor Networks</b>	<b>855</b>
<i>Wade Trappe, Peng Ning and Adrian Perrig</i>	
28.1 Introduction	855
28.2 Security and Privacy Challenges	856
28.3 Ensuring Integrity of Measurement Process	860
28.4 Availability Attacks against the Wireless Link	868
28.5 Ensuring Privacy of Routing Contexts	876
28.6 Conclusion	882
References	883
<b>Index</b>	<b>889</b>

## Preface

---

More than a decade ago, a book edited by Simon Haykin on array processing was a huge success with significant impact. Ever since, the field of array processing has grown to the extent that one can see its applications everywhere. Indeed, traditional array techniques form the foundation of the more general sensor processing and networking that continue to advance the state-of-the-art research and find ubiquitous applications. Sensor networks and array processing form the two pillars of the proposed handbook.

Sensors and array processing, in their own individual ways, have been active areas of research for several decades: Wireless communications, radar, radio astronomy, and biomedical engineering, just to name a few important ones. This new *Handbook on Array Processing and Sensor Networks* addresses these topics in an organized manner under a single umbrella.

The major goal of this *Handbook* is to collect tutorial discussions on recent advancements and state-of-the-art results by providing a comprehensive overview of array processing and sensor networks. It covers fundamental principles as well as applications. This *handbook* features some of the most prominent researchers from all over the world, addressing the important topics that we consider to be essential for making the *handbook* highly valuable to the readers; this point is well borne out by the list of contents.

This *Handbook* consists of an introductory chapter, followed by 28 chapters that are written by leading authorities in sensor networks and array signal processing. Putting all this material together under a single umbrella, we have a *Handbook* that is one of a kind.

This *Handbook* should appeal to researchers as well as graduate students and newcomers to the field of sensors and array processing, and thereby learn not only about the many facets of these two subjects but also exploit the possibility of cross fertilization between them. Moreover, this *Handbook* may also appeal to professors in teaching graduate courses on sensor networks and/or array signal processing.

Simon Haykin  
McMaster University

K. J. Ray Liu  
University of Maryland, College Park



## Contributors

---

**A. Anandkumar**, Adaptive Communications & Signal Processing, Electrical & Computer Engineering, Cornell University, Ithaca, NY, USA

**Shoko Araki**, NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan

**Joshua N. Ash**, The Ohio State University, Columbus, OH, USA

**Y. Bar-Shalom**, Electrical & Computer Engineering Department, University of Connecticut, Storrs, CT, USA

**Douglas C.-J. Bock**, Project Manager and Assistant Director for Operations, CARMA, Big Pine, CA, USA

**Markus Buck**, Harman/Becker Automotive Systems, Ulm, Germany

**Robert Calderbank**, Department of Electrical Engineering, Princeton University, Princeton, NJ, USA

**Federico S. Cattivelli**, Electrical Engineering Department, University of California, Los Angeles, CA, USA

**Bruce Cornell**, Surgical Diagnostics Ltd., St. Leonards, Australia

**Shuguang Cui**, Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX, USA

**Petar M. Djurić**, Stony Brook University, Stony Brook, NY, USA

**Simon Doclo**, University of Oldenburg, Signal Processing Group, Oldenburg, Germany

**A. Ephremides**, Department of Electrical & Computer Engineering, University of Maryland, College Park, MD, USA

**Behrouz Farhang-Boroujeny**, Department of Electrical & Computer Engineering, University of Utah, Salt Lake City, UT, USA

**Christina Fragouli**, School of Computer & Communication Sciences, EPFL, Switzerland

**Sharon Gannot**, Bar-Ilan University, School of Engineering, Ramat-Gan, Israel

**Alex B. Gershman**, Communications Research Laboratory, McMaster University, Hamilton, Ontario, Canada

**Eberhard Hänsler**, Technische Universität Darmstadt, Darmstadt, Germany

**Alfred Hanssen**, Department of Physics, University of Tromsø, Tromsø, Norway

**Simon Haykin**, Department of Electrical Engineering, McMaster University, Hamilton, Ontario, Canada

**Soummya Kar**, Department of Electrical & Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA

**Usman A. Khan**, Department of Electrical & Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA

**T. Kirubarajan**, Electrical & Computer Engineering Department, Communications Research Laboratory, McMaster University, Hamilton, Ontario, Canada

**Mohamed Krini**, Harman/Becker Automotive Systems, Ulm, Germany

**Vikram Krishnamurthy**, Department of Electrical & Computer Engineering, The University of British Columbia, Vancouver, B.C. Canada

**P. R. Kumar**, Department of Electrical & Computer Engineering, & Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, USA

**Azadeh Kushki**, Department of Electrical & Computer Engineering, University of Toronto, Toronto, Ontario, Canada

**Mauricio Lara**, Ingeniería Eléctrica, Cinvestav, México

**K. J. Ray Liu**, Department of Electrical & Computer Engineering, University of Maryland, College Park, MD, USA

**Angelos D. Liveris**, Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX, USA

**Zhi-Quan Luo**, Department of Electrical & Computer Engineering, University of Minnesota, Minneapolis, MN, USA

**Shoji Makino**, NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan

**Desmond C. McLernon**, University of Leeds, United Kingdom

**Marc Moonen**, Katholieke Universiteit Leuven, Dept. of Electrical Engineering, Leuven, Belgium

**Bill Moran**, University of Melbourne, Australia

**Randolph L. Moses**, The Ohio State University, Columbus, OH, USA

**José M. F. Moura**, Department of Electrical & Computer Engineering, Carnegie Mellon University, Pittsburgh PA, USA

**Arye Nehorai**, Department of Electrical & Systems Engineering, Washington University, St. Louis, MO, USA

**Peng Ning**, North Carolina State University, Raleigh, NC, USA

**Aldo G. Orozco-Lugo**, Cinvestav-IPN, México

**Mathias Ortner**, INRIA, Sophia Antipolis, France

**Adrian Perrig**, Carnegie Mellon University, Pittsburgh, PA, USA

**Konstantinos N. Plataniotis**, Department of Electrical & Computer Engineering, University of Toronto, Toronto, Ontario, Canada

**K. Punithakumar**, Electrical & Computer Engineering Department, McMaster University, Hamilton Ontario, Canada

**Tariq R. Qureshi**, Purdue University, West Lafayette, IN, USA

**Ahmed Sadek**, Qualcomm, San Diego, CA, USA

**Hiroshi Sawada**, NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan

**Ali H. Sayed**, Electrical Engineering Department, University of California, Los Angeles, CA, USA

**Akbar M. Sayeed**, University of Wisconsin-Madison, Madison, WI, USA

**Gerhard Schmidt**, Harman/Becker Automotive Systems, Acoustic Signal Processing Research, Ulm, Germany

**Thiagarajan Sivanadyan**, Wireless Communications Research Laboratory, Department of Electrical & Computer Engineering, University of Wisconsin-Madison, Madison, WI, USA

**Ann Spriet**, Katholieke Universiteit Leuven, Dept. of Electrical Engineering, Leuven, Belgium

**Stergios Stergiopoulos**, Diagnosis & Prevention/IRS, DRDC Toronto, Ontario, Canada

**Weifeng Su**, Department of Electrical Engineering, SUNY, Buffalo, NY, USA

**A. Swami**, U.S. Army Research Laboratory, Adelphi, MD, USA

**R. Tharmarasa**, Electrical & Computer Engineering Department, McMaster University, Hamilton, Ontario, Canada

**L. Tong**, Cornell University, Ithaca, NY, USA

**Wade Trappe**, WINLAB, Rutgers University, North Brunswick, NJ, USA

**Tom Vercauteren**, Asclepios Research Project, INRIA Sophia Antipolis, France

**Xiaodong Wang**, Electrical Engineering Department, Columbia University, New York, NY, USA

**Tricia J. Willink**, Communications Research Centre, Ottawa, Ontario, Canada

**Stefan Winter**, NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan

**Tobias Wolff**, Harman/Becker Automotive Systems, Ulm, Germany

**Jin-Jun Xiao**, Department of Electrical & Systems Engineering, Washington University, St. Louis, MO, USA

**Liang-Liang Xie**, Department of Electrical & Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada

**Zixiang Xiong**, Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX, USA

**Yang Yang**, Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX, USA

**Michael D. Zoltowski**, School of Electrical & Computer Engineering, Purdue University, West Lafayette, IN, USA

## Introduction

---

Simon Haykin

Department of Electrical Engineering, McMaster University, Hamilton, Ontario, Canada

The purpose of this introductory chapter is to provide an overview of the material covered in the subsequent 28 chapters of the *Handbook on Array Processing and Sensor Networks*. These 28 chapters are organized in four parts, as described next. Parts I and II constitute the first pillar of this *Handbook*: array signal processing; Parts III and IV constitute the second pillar, sensor networks.

### PART I: FUNDAMENTAL ISSUES IN ARRAY SIGNAL PROCESSING

This first part of this *Handbook*, embodying Chapters 1 through 5, addresses the following issues that are considered to be basic to the subject matter of the *Handbook*:

1. The theory of stochastic processes is of fundamental importance in the modeling of practically all the physical systems encountered in practice. Chapter 1, Wavefields by Alfred Hanssen, generalizes the theory of stochastic processes to *stochastic wavefields*. At first, such a generalization may seem to be straightforward; in reality, however, it is not. The starting point of the chapter is harmonizable stochastic processes, the roots of which are traced to the pioneering works of Loève in the 1940s.
2. Chapter 2, Spatial Spectrum Estimation, authored by Petar Djuric follows on quite nicely from Chapter 1 by viewing array signal processing as a spatial spectrum-estimation problem. The many facets of spectrum estimation, nonparametric as well as parametric, are discussed in the chapter.
3. The next chapter, MIMO Radio Propagation, is authored by Tricia J. Willink. After presenting a tutorial treatment of the MIMO channel capacity for wireless communications, building on Shannon's information theory, Willink describes various propagation models that have been developed for the statistical characterization of MIMO channels. Most importantly, the treatment includes analytical model parameterization, supported with practical measurements.
4. The chapter on Robustness Issues in Sensor Array Processing, written by Alex B. Gershman, addresses yet another fundamental issue in array signal processing, namely, *robustness*. In particular, Gershman presents an overview of the state-of-the-art algorithms for adaptive beamforming in a narrowband environment.

5. Chapter 5, entitled Wireless Communications and Sensing in Multipath Environments Using Multiantenna Transceivers, co-authored by Akbar M. Sayeed and Thiagarajan Sivanadyan, addresses the fundamental issue of how multipath channels, basic to wireless communications, can be modeled in a generic sense. The aim here is to account for the three dimensions of sensing: time, frequency, and space. Point-to-point MIMO wireless communications and active wireless sensing with wideband transmitter–receiver (transceiver) configurations are discussed in the chapter.

## PART II: NOVEL TECHNIQUES FOR AND APPLICATIONS OF ARRAY SIGNAL PROCESSING

The second part of this *Handbook* embodies seven chapters devoted to practical applications of array processing in a multitude of fields, as described next:

6. Chapter 6 on Implicit Training and Array Processing for Digital Communications Systems is co-authored by Aldo G. Orozco, Mauricio Lara, and Desmond C. McLernon. The concept of “implicit training” refers to a strategy in digital communications, where a special sequence is embodied in the transmitted signal to aid the receiver to perform specific parameter-estimation tasks in such a way that no additional bandwidth is required. After discussing this issue in detail, the chapter describes its application in array-processing systems, using continuous and packet forms of transmission.
7. Next, chapter 7 on Unitary Design of Radar Waveform Diversity Sets, co-authored by Michael D. Zoltowski, Tariq R. Qureshi, Robert Calderbank, and Bill Moran, discusses the radar counterpart of MIMO wireless communications, namely, MIMO radar. The challenge in MIMO radar is that of separating the transmitted waveforms at the receiver. This problem is complicated by the relative delay and Doppler shift of the transmitted waveforms, which may destroy the desired property of orthogonality. To mitigate these practical difficulties, examples of diversity waveforms for 2-by-2 and 4-by-4 MIMO radar configurations are studied.
8. Chapter 8 on Acoustic Array Processing for Speech Enhancement is co-authored by Markus Buck, Eberhard Hänsler, Mohamed Krini, Gerhard Schmidt, and Tobias Wolff. To deal with the issue of speech enhancement in a speech communication system, the discussion focuses on frequency-domain procedures, which are preferred over time-domain procedures for practical reasons. The techniques considered in the chapter include multichannel echo cancellation, beamforming structures, combined echo cancellation and beamforming, and sensor calibration. The final topic in the chapter is devoted to postprocessing techniques, exemplified by the suppression of residual interferences and background noise, dereverberation, spatial postfiltering, and other issues of practical importance.
9. Chapter 9 on Acoustic Beamforming for Hearing Aid Applications is co-authored by Simon Doclo, Sharon Gannot, Marc Moonen, and Ann Sprriet. With this chapter devoted to speech processing, it follows quite nicely on the previous chapter on speech enhancement. As the title of the chapter would indicate, the discussion focuses on the design of multimicrophone algorithms in order to improve speech intelligibility in background noise for hearing-aid applications. The techniques

studied in the chapter include minimum variance distortionless response (MVDR) beamformer, frequency-domain generalized side-lobe canceller (GSLC), and multichannel Wiener filter for monaural processing. Then, with binaural processing as the issue of interest, the multichannel Wiener filter and its variants are discussed in the latter part of the chapter. Experimental results for both monaural and binaural processing are presented.

10. Chapter 10, entitled Underdetermined Blind Source Separation Using Acoustic Arrays and co-authored by Shoji Makino, Shoko Araki, Stefan Winter and Hiroshi Sawada, completes the study of acoustic arrays for speech-related applications. More specifically, this chapter addresses a difficult blind source separation problem that is underdetermined, that is, the number of sensors is smaller than the number of sources (i.e., microphones) responsible for generating the observables. Design of the acoustic array of microphones exploits *sparseness*, which is an inherent characteristic of speech signals. Two methods are discussed in the chapter. The first method is rooted in the time–frequency domain. The second method uses a maximum a posteriori (MAP) probability approach, which combines blind system identification and blind source recovery. Experimental results are presented in support of both methods.
11. Chapter 11, entitled Array Processing in Astronomy, is authored by Douglas C.-J Bock. The chapter focuses primarily on the theory, design, and signal processing of a special kind of arrays called “correlation arrays,” which consist of arrays of antennas (typically, parabolic reflectors) that are analyzed by cross-correlating the outputs of each pair of antennas as in interferometry. An important feature of correlation arrays is that they permit imaging of the entire field of view of an individual antenna at a resolution determined by the overall extent of the array. Other techniques, namely, aperture-plane phased arrays, focal-plane phased arrays, and array processing at optical and infrared wavelengths are also mentioned in the chapter.
12. The next chapter, entitled Digital 3D/4D Ultrasound Imaging Technology and authored by Stergios Stergiopoulos, completes Part II of the *Handbook*. Starting with practical limitations of two-dimensional (2D) medical ultrasonic imaging technology in terms of poor resolution, the stage is set for the study of the next 3D/4D ultrasonic image technology for medical diagnostic applications. The study includes synthetic aperture processing for digitizing large-size planar arrays, multifocus transmit beamformer for linear and planar phased arrays, and multifocus receive beamformer for linear and planar phased arrays. The chapter also includes the description of an experimental planar array ultrasound imaging system, including some performance results of the system.

### PART III: FUNDAMENTAL ISSUES IN DISTRIBUTED SENSOR NETWORKS

In Part III of this *Handbook*, we move onto the study of sensor networks, constituting its second pillar. This third part consists of nine chapters that cover the following issues:

13. Chapter 13 on Self-Localization of Sensor Networks, authored by Joshua N. Ash and Randolph L. Moses, studies the issue of inference from spatially distributed sensors so as to acquire knowledge of where the individual sensors are located,

- hence the use of “selflocalization” in the title. With this objective in mind, the chapter describes Cramér–Rao bounds for self-localization parameters, localization algorithms, and measurement errors.
14. Chapter 14 on Multitarget Tracking and Classification in Collaborative Sensor Networks via Sequential Monte Carlo methods, written by Tom Vercauteren and Xiaozhang Wang, discusses the problem of jointly tracking and classifying targets that evolve within an environment of scattered sensor nodes. The enablers of such a capability involve microelectromechanical systems (MEMS) and microprocessors, coupled with ad hoc networking protocols, all of which make it possible for low-cost sensors to collaborate and attain prescribed tasks at relatively low-power levels. With target dynamics as the topic of interest, use is made of sequential Monte Carlo (SMC) methods and related importance sampling and resampling procedures.
  15. The next chapter on Energy-Efficient Decentralized Estimation, co-authored by Jin-Jun Xiao, Shuguang Cui, and Zhi-Quan Luo, addresses a major challenge in wireless sensor networks, namely, how to design these networks subject to a hard-energy constraint. This practical problem is a difficult one because operation of the sensors is dependent on small-size batteries, the replacement costs of which can be expensive if not impossible. The threefold theme of the chapter is energy-efficient distributed estimation in wireless sensor networks that embody:
    - Local data compression
    - Wireless communications
    - Data fusion
  16. With multisensor data fusion as an issue of interest in the preceding chapter, its discussion is continued in the chapter on Sensor Data Fusion with Application to Multitarget Tracking, which is written by R. Tharmarasa, K. Punithakumar, T. Kirubarajan, and Y. Bar-Shalom. Naturally, tracking plays a vital role in sensor data fusion, hence the need for algorithms required to perform state estimation, given data received from one or more sensors. Moreover, data association is an essential component in sensor fusion, particularly when there is uncertainty in the original source of data used in the fusion process. Yet another practical issue of concern is that the data may not arrive at the fusion center in the right sequence due to the unavoidable presence of network delays. Confronted with all these practical issues, a reliable solution to the data fusion problem is discussed in the chapter.
  17. Chapter 17, entitled Distributed Algorithms in Sensor Networks, by Usman A. Khan, Soummya Kar, and José M.F. Moura, describes an architecture for resource-constrained networks that have a *weblike topology*. This topology embodies decentralized and distributed inference algorithms, where each sensor in the network updates its own local detector using state information gathered by neighboring sensors. The updating is performed iteratively in such a way that the state of the particular sensor in question converges to the state of the optimal centralized or parallel detector. The iterative nature of the distributed algorithm is attributed to the fact that information flow is limited because of sparse connectivity of the network. The chapter focuses on distributed algorithms that are *linear*, for which a systematic study is provided.
  18. In chapter 18, the study of sensor networks moves onto another important topic: Cooperative Sensor Communications written by Ahmed Sadek, Weifeng Su, and

Ray Liu. With cooperative communications among the sensors as the goal, the chapter focuses on how to attain this goal with two issues in mind:

- The limited energy available to the component nodes
- The possibility of the nodes attempting to cooperate while the operations of sensing and communication are being carried out

With wireless communication as the method of choice, the cooperative-sensor communication problem becomes complicated by channel fading due to the multipath phenomenon, for which the use of diversity is the standard solution. More specifically, the chapter generalizes multiple-input multiple-output (MIMO) communication and related protocols to tackle cooperative communications among wireless sensor nodes. (The use of MIMO was discussed previously in Chapter 3.)

19. In a distributed sensor network, we typically find that the transmitters in the network are not permitted to communicate with each other due to increased complexity and/or power constraints. To address this issue, we look to *distributed source coding*, the foundation of which was laid out in a 1973 study by Slepian and Wolf. During the past 35 years, many important contributions have been made to this topic. Chapter 19, *Distributed Source Coding*, written by Zixiang Xiong, Angelos D. Liveris, and Yang Yang, reviews the theory, design, and applications of distributed source coding. In particular, detailed descriptions of the following topics are presented in the chapter:
  - Multiterminal source-coding methods of the direct and indirect kinds
  - The designs of Slepian–Wolf, Wyner–Ziv codes and their variants
  - The applications of distributed source codes in secure biometrics, lossless compression in multiterminal networks, and, most importantly, distributed video coding
20. The next chapter, *Network Coding for Sensor Networks*, by Christina Fragouli, explores the relatively new idea of *network coding*, which has the potential to revolutionize the way information is treated in a sensor network. As such, network coding may impact various network functionalities such as routing, network storage, and network design. Basically, network coding deals with information flow across a sensor network. However, it may well be that it is in ad hoc wireless sensor networks, where network coding may have an immediate impact.
21. The final chapter of Part III is by Liang-Liang Xie and P. R. Kumar on *Information-Theoretic Studies of Wireless Sensor Networks*. A basic characteristic of wireless networks is their *broadcast nature*, which necessarily causes *interference* across a network. In the current approach to the formulation of protocols for wireless networks, interference is usually viewed to be undesirable. In reality, however, interference is not “noise;” rather, it should be viewed as a signal that carries information, but it is unintentionally received. This reality motivates the challenge of finding ways to exploit interference rather than succumb to it. Using information-theoretic ideas, Xie and Kumar develop wireless communication schemes that exploit unintentionally received signals in sensor networks.

## PART IV: NOVEL TECHNIQUES FOR AND APPLICATIONS OF DISTRIBUTED SENSOR NETWORKS

In the fourth and final part of this *Handbook* dealing with distributed sensor networks, we have assembled seven chapters on novel techniques and applications, which start with adaptivity and finish with security and privacy.

22. Chapter 22 by Ali H. Sayed and Federico S. Cattivelli on ‘Distributed Adaptive Learning Mechanisms’ describes recent developments in distributed processing over networks. The presentation covers adaptive learning algorithms that make it possible for neighboring nodes in a distributed network to communicate with each other at each iteration of the algorithm. Specifically, each node in the network exchanges estimates with its neighboring nodes, with the estimates being fused and quickly incorporated into local adaptation rules. In this way, the network as a whole becomes adaptive, whereby it is enabled to respond to space–time variations in the underlying statistical profile of the data. The chapter describes different learning rules at the nodes along with different cooperation protocols, thereby yielding adaptive networks with varying complexity and potential application.
23. The classical approach in distributed sensor networks addresses two problems: distributed statistical inference and minimum-cost routing of the measurements to the fusion center. A shortcoming of this approach is the failure to exploit the “inherent” saving in routing costs. In Chapter 23, entitled Routing for Statistical Inference in Sensor Networks, its co-authors A. Anandkumar, A. Ephremides, A. Swami, and L. Tong, take a different approach in the following sense: In-network processing of the likelihood function, representing the minimal sufficient statistic, is performed and delivered to the fusion center for inference. To this end, the *Markov random field (MRF) model* of spatial correlation of sensor data is employed. Accordingly, the underlying structure of the likelihood function is known for the MRF model by invoking the well-known *Hammersley–Clifford theorem*. By exploiting this structure, it is shown that the minimum-cost routing for computing and delivering the likelihood function is a *Steiner tree on a transformed graph*. With the approximation ratio preserved, it follows that any Steiner tree approximation can be employed for minimum-cost fusion at the same approximation ratio. An overview of the minimum-cost fusion procedure is presented in the chapter.
24. In this chapter, Behrouz Farhang-Boroujeny introduces the idea of cognitive radio. Lately, interest in cognitive radios has been growing exponentially as a way of solving the current underutilization of the electromagnetic radio spectrum. Simply put, there are *spectrum holes* (i.e., underutilized subbands of the radio spectrum) at certain points in time and geographic locations. These spectrum holes can be made available to *secondary users*, which are to be distinguished from the primary users who are legally entitled to occupy those subbands. A basic problem on which the very essence of cognitive radio rests is that of identifying the location of spectrum holes in the radio spectrum. One way of accomplishing this task is to use spectral estimation, hence the title of Chapter 24: Spectral Estimation in Cognitive Radios. In particular, Farhang-Boroujeny describes the multitaper spectral estimator (MTSE), which has several attributes as a spectral estimator. Instead of following the original approach used by David Thomson in 1982 to derive the MTSE, its formulation in this chapter is centered on the idea of filter

- banks, the underlying theory of which is well known in the signal processing literature.
25. Chapter 25, by Azadeh Kushki, and Konstantinos N. Plantaniotis, discusses Nonparametric Techniques for Pedestrian Tracking in Wireless Local Area Networks. The choice of nonparametric techniques for tracking is influenced by the fact that an explicit form for the position-received signal strength (RSS) is typically unknown. The discussion starts with nonparametric kernel-density estimation, which has the advantage of providing a covariance matrix that is used to gauge the reliability of the position estimate. It is this feature that prompted the development of state-space filters, which augment memoryless estimates with knowledge of pedestrian motion dynamics. Global feedback is employed to guide the selection of anchor points and wireless access points used during the estimation procedure. This is done to mitigate difficulties that arise due to practical discrepancies between training and testing conditions, which do arise from the nonstationary character of the indoor wireless environment.
  26. Chapter 26, by Vikram Krishnamurthy and Bruce Cornell, addresses a challenging problem: Reconfigurable Self-Activating Ion-Channel-Based Biosensors: Signal Processing and Networking via the Theory of Global Games. Biological ion channels are water-filled subnano-sized ports that are formed by protein molecules in the membrane of all living cells. These ion channels play a crucial role in living organisms in that their flow into and out of a cell regulates the biochemical activities of the cell. The chapter builds on classical to state-of-the-art tools in signal processing and control theory to model the underlying dynamics of ion channel biosensors in a novel way. Moreover, the powerful concept of *global games* is used to derive biosensor activation algorithms that appear to have a simple threshold Nash equilibrium.
  27. Chapter 26 on biological ion channels is followed nicely by Chapter 27 written by Mathias Ortner and Arye Nehorai, dealing with a biochemical problem, namely, Biochemical Transport Modeling, Estimation, and Detection in Realistic Environments. This chapter introduces a new approach for computing and using a numerical forward physical dispersion model, the purpose of which is to relate the source given by an array of biochemical sensors in realistic environments. The approach described therein provides a modeling framework, which accounts not only for complex geometries but also permits the full use of software-simulated random wind turbulence. The key feature of the model is that the “fluid simulation” part of the model is decoupled from the “transport computation” part. The chapter also includes an illustrative example on monitoring biochemical events. Other related topics covered in the chapter include the following: a sequential detector for dealing with unknown parameters, namely, start time of the spread, original concentration, and location of the initial delivery.
  28. The study of distributed sensor networks would be incomplete without a discussion of two critical issues: security and privacy, which is precisely what we have in the very final chapter of this *Handbook*. More specifically, the chapter entitled Security and Privacy for Sensor Networks written by Wade Trappe, Peng Ning, and Adrian Perrig, explores the following pair of issues:
    - Security and privacy challenges that face designers of sensor networks

- Defense strategies that may be employed to protect a sensor network against external attacks

Naturally, the need for security and privacy is prompted by the fact that sensor networks are deployed in environments that are typically unattended. The attacks confronting sensor networks may range from threats that seek to corrupt the basic processes of measurements (hence, compromising the reliability of the network), to attacks aimed at wireless connectivity, and to attacks where knowledge of routing functionality is used by an adversary for its own advantage, discussions of which are all covered in the chapter.

---

**PART I**

---

## **FUNDAMENTAL ISSUES IN ARRAY SIGNAL PROCESSING**



---

## CHAPTER 1

---

# Wavefields

Alfred Hanssen

Department of Physics and Technology, University of Tromsø, Norway

### 1.1 INTRODUCTION

The theory of univariate stochastic processes [1, 2] forms the backbone of the analysis and processing of single sensor data. Stochastic processes  $X(t)$  are parameterized by a single free variable  $t$ , which is a timelike variable. By performing some appropriate processing on samples from realizations of the process  $X(t)$ , one hopes to be able to infer physical properties about the source of the signal, about the transmission medium, or both. To aid in the development of processing techniques, it is customary to seek simplifying assumptions. A standard assumption is that of stationarity [3], which allows for drastic simplifications of moment functions and related quantities. Needless to say, the standard assumptions that we apply are often questionable for real-world situations.

However, our problems of interest in the physical world are often parameterized in four-dimensional space–time coordinates rather than in time alone. Examples of such are abundant; just think about any wave or fluctuation phenomena in physical space [4–7]. For engineering applications, the propagation and space–time distribution of radio waves is an important example [8–11], as is the propagation and space–time distribution of acoustic waves in audio applications [12]. In the geosciences and in oil-related prospecting and research, space–time statistics enter at all levels [13–15]. Such phenomena necessitate a generalization of the theory of univariate stochastic processes to a theory of multivariate stochastic wavefields [11, 16–18]. An appropriate notation for a space–time wavefield could then be  $X(\mathbf{r}, t)$ , where  $\mathbf{r}$  is a three-dimensional position vector in some preferred coordinate system.

One would think that the generalization of stochastic processes to stochastic wavefields should be straightforward. On the surface, it may seem so, but in reality, it is not that simple. The reason is that for wave or fluctuation phenomena in physical media, the supporting medium imposes severe constraints on the waves and fluctuations that can be supported. This is the ubiquitous phenomenon of wave dispersion, which implies that the medium only allows certain combinations of frequencies and wavenumbers (spatial vector frequencies) to exist [19, 20]. While dispersion is well understood for classical deterministic space–time systems, it is poorly understood for stochastic systems.

## 1.2 HARMONIZABLE STOCHASTIC PROCESSES

A recent and very welcome trend in signal processing is to abandon some of the standard simplifying and limiting assumptions from the early days of the field. This strategy would allow us to deal with situations that are of great practical importance. When dealing with stochastic processes, one often assumes that the process under study obeys some kind of stationarity or time invariance. However, in practice, crucial statistical parameters may be time variant, thus violating the stationarity assumption.

To be able to deal with time-variant statistics, we need to arm ourselves with a stringent theory, rather than qualitative and ad hoc descriptions. Luckily, a very systematic and useful theoretical framework for so-called harmonizable stochastic processes, developed mainly by Michel Loève and Harald Cramér in the 1940s, is at our disposal. We are of the opinion that engineering has now matured to such a degree that the theory of harmonizable stochastic processes should be part of the toolbox of any advanced engineer. In this chapter, we will briefly review this beautiful and by now classical theory, and we will emphasize an intuitive understanding rather than going to the deepest theoretical level.

In this chapter, we will invariably assume that the real valued stochastic process  $X(t)$  belongs to the *harmonizable class* [21–24]. Any processes belonging to the harmonizable class has the following stochastic integral representation:

$$X(t) = \int \exp(j\omega t) dZ(\omega), \quad (1.1)$$

which should be understood in the mean-square sense, and where the integration is over the real axis,  $\mathbb{R}$ . Note that since  $X(t)$  is a stochastic process, its conventional Fourier transform does not exist. Instead, in Eq. (1.1) the complex exponential  $\exp(j\omega t)$  is integrated against a complex valued infinitesimal random process  $dZ(\omega)$  parameterized by the radian frequency  $\omega$ . This crucial quantity is the so-called *increment process* or the *generalized Fourier transform* for the stochastic process  $X(t)$ . Equation (1.1) is often called the *spectral representation* of  $X(t)$ , and this particular type of integral is often called a Fourier–Stieltjes integral.

From an engineering point of view, it is useful to realize that the Fourier–Stieltjes integral is nothing more exotic than an infinite sum of infinitesimal complex phasors, one phasor for each frequency  $\omega$ . Depending on the distribution of the infinitesimal complex amplitudes, and the correlation among the phasors for different frequencies, a great variety of stochastic processes  $X(t)$  can be constructed. We now understand that the statistical characteristics of the generalized Fourier transform  $dZ(\omega)$  completely determines the behavior and properties of the observable stochastic process  $X(t)$ .

### 1.2.1 Moment Functions

The most important aspect of the harmonizable class is that nonstationarities are readily included in this formulation. To appreciate this fact, we start by defining the dual-time second-order autocorrelation function (ACF) by

$$M_{XX}(t, \tau) = E[X(t)X(t + \tau)], \quad (1.2)$$

where we understand that  $t$  is a global time variable, and  $\tau$  is a local (or time shift) variable. For nonstationary processes, it is evident that we need to keep track of both time arguments, as the statistics may change with the global time  $t$ . Wide-sense stationary processes would be those processes for which the ACF is independent of global time  $t$  and where the mean process  $E[X(t)]$  is a constant independent of  $t$ .

Inserting the spectral representation Eq. (1.1) into the dual-time ACF Eq. (1.2), we readily find that the ACF can be formulated as [25]

$$M_{XX}(t, \tau) = \int \int \exp[j(vt + \omega\tau)] S_{XX}(\omega, v) \frac{d\omega dv}{(2\pi)^2}, \quad (1.3)$$

where the dual-frequency complex-valued density  $S_{XX}(\omega, v)$  is defined by

$$S_{XX}(v, \omega) \frac{d\omega dv}{(2\pi)^2} = E[dZ(\omega) dZ^*(\omega - v)]. \quad (1.4)$$

Here,  $\omega \in \mathbb{R}$  is a global frequency variable, and  $v \in \mathbb{R}$  is a local (or frequency shift) variable, and the asterisk denotes complex conjugation. The dual-frequency spectrum  $S_{XX}(v, \omega)$  is often called the Loève spectrum. It is important to observe that the Loève spectrum is nothing but the correlation between the frequency components of the increment process that generates our stochastic process of interest. We also immediately notice that the dual-time ACF and the dual-frequency Loève spectrum are a two-dimensional Fourier transform pair, with the global time  $t$  transforming into the local frequency  $v$ , and the local time  $\tau$  transforming into the global frequency  $\omega$ .

Although it is quite obvious that the dual-frequency spectrum must contain important information about nonstationary stochastic processes, it has so far found surprisingly few uses in practice. It is, however, interesting to see that very advanced applications of dual-frequency spectra were discussed already in the early 1960s by Hagfors [9, 26, 27]. His analysis of radar returns from the lunar surface was instrumental for the Apollo missions and lunar landings in the 1960s and 1970s.

Since we have two time variables ( $t$  and  $\tau$ ) and two frequency variables ( $\omega$  and  $v$ ) available, we may construct two more relevant second-order moment functions just by invoking partial Fourier transforms. First, by transforming the ACF with respect to  $\tau$ , or by inverse transforming the Loève spectrum with respect to  $v$ , we obtain the following important time–frequency spectrum [25]:

$$R_{XX}(t, \omega) \frac{d\omega}{2\pi} = E[X(t) dZ^*(\omega)] e^{-j\omega t}. \quad (1.5)$$

This is the so-called Kirkwood–Rihaczek (KR) time–frequency spectrum, which has a special status among all bilinear time–frequency density functions. This distribution first appeared in quantum mechanics in a work by John Kirkwood in 1933 [28], and it was later rediscovered by August Rihaczek in the context of signal theory in 1968 [29]. The KR spectrum enjoys many fundamental and important properties, and we see that it is basically the correlation between the process itself at time  $t$  and its generator, the generalized Fourier transform at frequency  $\omega$ . As such, it is a measure of the similarity between the stochastic process at time  $t$  with an infinitesimal stochastic phasor at frequency  $\omega$ .

$$\begin{array}{ccc}
 M_{XX}(t, \tau) & \xrightarrow{\tau \rightarrow \omega} & R_{XX}(t, \omega) \\
 \downarrow t \rightarrow \nu & & \downarrow t \rightarrow \nu \\
 A_{XX}(\nu, \tau) & \xrightarrow{\tau \rightarrow \omega} & S_{XX}(\nu, \omega)
 \end{array}$$

**Figure 1.1** Four-corners diagram: Fourier relations between the basic densities of harmonizable stochastic processes.

The fourth possibility results either from a partial Fourier transform of the dual-time ACF  $M_{XX}(t, \tau)$  with respect to the global time  $t$  or a partial inverse Fourier transform of the dual-frequency Loève spectrum  $S_{XX}(\nu, \omega)$  with respect to the global frequency  $\omega$ . The resulting function  $A_{XX}(\nu, \tau)$  is the so-called ambiguity function, which is a time–frequency spectrum in local coordinates.

It turns out that the four possible second-order moment functions can be arranged into a useful pattern often called a *four-corners diagram* [25]. In the four-corners diagram shown in Figure 1.1, functions in adjacent corners are one Fourier transform apart, while functions in opposite corners are two Fourier transforms apart.

### 1.2.2 Wide-Sense Stationarity

Note that the four-corners diagram is a direct generalization of the familiar Einstein–Wiener–Kinchine relation for wide-sense stationary (WSS) stochastic processes. Hence, for WSS processes, the four-corners diagram will collapse into the familiar single Fourier transform pair

$$\tilde{M}_{XX}(\tau) \longleftrightarrow \tilde{S}_{XX}(\omega), \quad (1.6)$$

where  $\tilde{M}_{XX}(\tau) = E\{X(t)X(t + \tau)\}$  is the conventional autocorrelation function that depends only on local time  $\tau$ , and  $\tilde{S}_{XX}(\omega) = \int \tilde{M}_{XX}(\tau) \exp(-j\omega\tau) d\tau$  is the conventional power spectral density.

In greater detail, we find that the conventional stationary power spectrum is always completely included in the dual-frequency spectrum Eq. (1.4), and that it can be extracted by inserting  $\nu = 0$ . For WSS processes, the dual-frequency spectrum has the particularly simple form

$$S_{XX}(\nu, \omega) = \tilde{S}_{XX}(\omega)\delta(\nu), \quad (1.7)$$

that is, the conventional power spectral density  $\tilde{S}_{XX}(\omega)$  rides on a delta ridge concentrated on the single line  $\nu = 0$ . For this reason, the axis  $\nu = 0$  is called the *stationary manifold*. In general, however, the Loève spectrum has contributions also outside the stationary manifold for an arbitrary nonstationary process.

Since the dual-frequency spectrum is a frequency correlation, we now understand that for a harmonizable process to be WSS, its frequency components are orthogonal, since for WSS,

$$E\{dZ(\omega_1) dZ^*(\omega_2)\} = 0 \quad \forall \omega_1 \neq \omega_2. \quad (1.8)$$

We now understand that if at least one frequency pair of the generalized Fourier transform is nonorthogonal, then evidently the process will be nonstationary.

### 1.3 STOCHASTIC WAVEFIELDS

Stochastic wavefields are multidimensional generalizations of stochastic processes. In this chapter, we will specifically consider space–time wavefields; thus the time variable  $t$  must be augmented by a three-dimensional spatial vector variable  $\mathbf{r} = [x, y, z]^T$ , where  $x$ ,  $y$ , and  $z$  are Cartesian coordinates. Assume now that a scalar spatiotemporal stochastic wavefield  $\xi(\mathbf{r}, t)$  is harmonizable in space and time, that is, that we can generalize Eq. (1.1) so that we can express  $\xi(\mathbf{r}, t)$  by the following four-dimensional integral:

$$\xi(\mathbf{r}, t) = \int \exp[j(\mathbf{k}^T \mathbf{r} - \omega t)] dZ(\mathbf{k}, \omega), \quad (1.9)$$

where  $dZ(\mathbf{k}, \omega)$  is the increment processes or generalized Fourier transform for the random field,  $\mathbf{k} = [k_x, k_y, k_z]^T$  is a three-dimensional wavenumber vector variable with components  $k_x$ ,  $k_y$ , and  $k_z$ , and  $\omega$  is a scalar radian frequency variable.

Depending on the application, the stochastic wavefield may be a scalar, a vector, or a tensor function of space–time. For example, a scalar field may be used to represent acoustic fluctuations, a vector field may be used to represent electric and magnetic field vectors in electromagnetics, whereas tensor fields may be used to model mechanical stress and strain, or electric conductance in anisotropic media. In the present chapter, we will only consider scalar wavefields.

By reading some of the statistics literature (e.g., [30, 31]), one may be left with the impression that it is trivial to extend the harmonizable class of random processes to random fields. According to these sources, one would simply have to replace the “time” variable with a “time” vector, and hence all defining integrals would inherit the dimensionality of the time vector, and  $\omega$  would become a corresponding frequency vector. If these statements were true, it would be straightforward to generalize the concept of nonstationary processes to inhomogeneous random fields.

In reality, however, physics dictates the dynamics of random fields. In short, random fields evolving in space–time have to respond to the actual physical medium in which they operate. It should be obvious that any physical medium supporting a fluctuation or disturbance of some kind, puts severe constraints on the fluctuations. The most important physical constraint is that of *dispersion*, which among other things implies that components with different frequencies (or wavenumbers) in general have different propagation speeds. In more technical terms, this means that a given frequency can only correspond to a single (or a certain set of) wavenumbers (spatial frequencies).

It has proven difficult to combine inhomogeneity and dispersion in a single description for random fields. In this chapter, we will outline how such a stringent and systematic theory may be built within the class of harmonizable random fields. We expect that much of the theory and practice of array processing would benefit from being generalized and rewritten in terms of this theory.

#### 1.3.1 Second-Order Moment Functions

Paralleling the discussion for stochastic processes, we seek a systematic theory for the second-order moment functions of stochastic wavefields. Depending on which variables one chooses to emphasize, one may formulate the second-order moments in space–time, frequency–wavenumber, and combinations thereof. By a proper

generalization of the theory for harmonizable random processes to random wavefields, we will be able to deal with nonstationary and inhomogeneous random fields. Among the attractive properties of the resulting formulation is that we will be able to retain our understanding of frequencies and wavenumber vectors, even in the presence of nonstationary and inhomogeneous fields.

**1.3.1.1 Spatiotemporal Correlation Functions** By autocorrelating the wavefield at two different position vectors ( $\mathbf{r}$  and  $\mathbf{r} + \boldsymbol{\rho}$ ) and two different time instants ( $t$  and  $t + \tau$ ), we define the following space–time correlation function:

$$M_{\xi\xi}(\mathbf{r}, \boldsymbol{\rho}; t, \tau) = E \{ \xi(\mathbf{r}, t) \xi(\mathbf{r} + \boldsymbol{\rho}, t + \tau) \}. \quad (1.10)$$

Here,  $\mathbf{r}$  and  $t$  are global spatial and temporal coordinates, and  $\boldsymbol{\rho}$  and  $\tau$  are the corresponding local coordinates.

**1.3.1.2 Frequency–Wavenumber Spectrum** By inserting the four-dimensional generalized Fourier description from Eq. (1.9) into the definition of the second-order space–time correlation function, Eq. (1.10), we find that the space–time correlation function can be formulated through the following eight-dimensional integral:

$$M_{\xi\xi}(\mathbf{r}, \boldsymbol{\rho}; t, \tau) = \int \exp [j(\mathbf{k}^T \boldsymbol{\rho} + \boldsymbol{\kappa}^T \mathbf{r} - \omega \tau - \nu t)] S_{\xi\xi}(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu) \frac{d\mathbf{k} d\boldsymbol{\kappa} d\omega d\nu}{(2\pi)^8}, \quad (1.11)$$

where the complex-valued dual-wavenumber, dual-frequency density is defined by

$$S_{\xi\xi}(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu) \frac{d\mathbf{k} d\boldsymbol{\kappa} d\omega d\nu}{(2\pi)^8} = E [dZ(\mathbf{k}, \omega) dZ^*(\mathbf{k} - \boldsymbol{\kappa}, \omega - \nu)]. \quad (1.12)$$

Basically, this important density is the nonnormalized correlation or inner product between the stochastic generator of the wavefield, at a pair of wavenumber vectors, and at a pair of frequencies.

**1.3.1.3 Spatiotemporal Frequency–Wavenumber Spectrum: Global Variables** When dealing with wave propagation and stochastic fluctuations, it is often desirable to formulate the wavefields simultaneously in space, wavenumber, frequency, and time. Such a formulation can be derived from either of the two preceding second-order moment functions. For example, if we Fourier transform the dual-space, dual-time correlation function with respect to local space and local time, we obtain a function that is defined in terms of global space, global wavenumber, global time, and global frequency. The same function can be derived by an inverse Fourier transform of the dual-wavenumber, dual-frequency correlation function, with respect to the local wavenumber and local time variables. In both cases, the following interesting complex-valued correlation function appears:

$$R_{\xi\xi}(\mathbf{k}, \mathbf{r}; \omega, t) = \int S_{\xi\xi}(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu) \exp[j(\nu t - \boldsymbol{\kappa}^T \mathbf{r})] \frac{d\nu d\boldsymbol{\kappa}}{(2\pi)^4}. \quad (1.13)$$

The resulting spatiotemporal frequency–wavenumber spectrum is a direct generalization of the Kirkwood–Rihaczek time–frequency spectrum to wavefields.

$$\begin{array}{ccc}
 M_{\xi\xi}(\mathbf{r}, \boldsymbol{\rho}; t, \tau) & \xrightarrow{\boldsymbol{\rho} \rightarrow \mathbf{k}, \tau \rightarrow \omega} & R_{\xi\xi}(\mathbf{k}, \mathbf{r}; \omega, t) \\
 \downarrow r \rightarrow \boldsymbol{\kappa}, t \rightarrow \nu & & \downarrow r \rightarrow \boldsymbol{\kappa}, t \rightarrow \nu \\
 A_{\xi\xi}(\boldsymbol{\kappa}, \boldsymbol{\rho}; \nu, \tau) & \xrightarrow{\boldsymbol{\rho} \rightarrow \mathbf{k}, \tau \rightarrow \omega} & S_{\xi\xi}(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu)
 \end{array}$$

**Figure 1.2** Fourier relations between the basic spectral densities of nonstationary and inhomogeneous harmonizable wavefields.

**1.3.1.4 Spatiotemporal Frequency–Wavenumber Spectrum: Local Variables** To derive the spatiotemporal frequency–wavenumber spectrum in terms of global variables, we performed an inverse Fourier transform of the dual-frequency dual-wavenumber spectrum  $S_{\xi\xi}(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu)$  with respect to the local variables ( $\boldsymbol{\kappa}$  and  $\nu$ ). However, one more possibility remains. If we perform an inverse Fourier transform of  $S_{\xi\xi}(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu)$  with respect to the global variables  $\mathbf{k}$  and  $t$ , we obtain a direct generalization of the well-known ambiguity function. The resulting spatiotemporal frequency–wavenumber spectrum is expressed exclusively in terms of local variables, and takes the form

$$A_{\xi\xi}(\boldsymbol{\kappa}, \boldsymbol{\rho}; \nu, \tau) = \int S_{\xi\xi}(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu) \exp[j(\omega\tau - \mathbf{k}^T \boldsymbol{\rho})] \frac{d\mathbf{k} d\omega}{(2\pi)^4}. \quad (1.14)$$

This quantity is a direct generalization of the ambiguity function that often appears in sonar and radar problems. However, the present generalization includes both temporal and spatial ambiguity.

**1.3.1.5 Stationary and Homogeneous Manifold** When dealing with stochastic processes and time series, a standard simplifying assumption is that of wide-sense stationarity. Wide-sense stationarity implies that the second-order temporal moment function is invariant with respect to shifts in global time, which implies that the dual-frequency spectrum is invariant with respect to shifts in the local frequency. Thus, we may formulate the stationary manifold of stochastic processes as being those processes for which the dual frequency is confined to the manifold:

$$\{(f, \nu) \mid f \in \mathbb{R}, \nu = 0\}. \quad (1.15)$$

Recall also from Section 1.2 that assuming wide-sense stationarity is equivalent to the assumption that the frequency components of the process are orthogonal.

For spatiotemporal wavefields, one must also generalize the spatial statistics so that the dual-wavenumber part becomes invariant with respect to shifts in the local wavenumber vector. We thus define the stationary and homogeneous manifold to be the subspace:

$$\{(\mathbf{k}, \boldsymbol{\kappa}; f, \nu) \mid \mathbf{k} \in \mathbb{R}^3, \boldsymbol{\kappa} = \mathbf{0}, f \in \mathbb{R}, \nu = 0\}. \quad (1.16)$$

The orthogonality argument carries directly over to the wavefields, implying that for stationary and homogeneous wavefields we must have

$$S_{\xi\xi}(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu) = \tilde{S}_{\xi\xi}(\mathbf{k}, \omega) \delta(\nu) \delta(\boldsymbol{\kappa}) \quad (1.17)$$

for some nonnegative frequency–wavenumber spectral density function  $\tilde{S}_{\xi\xi}(\mathbf{k}, \omega)$ . It is important to realize that  $\tilde{S}_{\xi\xi}(\mathbf{k}, \omega)$  is the conventional frequency–wavenumber

spectrum. It is now evident that the conventional stationary and homogeneous frequency–wavenumber spectrum resides on a four-dimensional subspace (defined by  $\kappa = \mathbf{0}$ ,  $v = 0$ ) in  $\mathbb{R}^8$ .

### 1.3.2 Hilbert Space and Generalized Coherences

To gain some insight into the meaning of the dual-frequency dual-wavenumber spectrum for nonstationary and inhomogeneous stochastic wavefields, we will now take a closer look at the geometry of this quantity. First, we note that the spectrum in Eq. (1.12) can be written as

$$S_{\xi\xi}(\mathbf{k}, \boldsymbol{\kappa}; \omega, v) \frac{d\mathbf{k} d\boldsymbol{\kappa} d\omega dv}{(2\pi)^8} = \langle dZ(\mathbf{k}, \omega), dZ(\mathbf{k} - \boldsymbol{\kappa}, \omega - v) \rangle, \quad (1.18)$$

where the Hilbert space inner product between two complex-valued stochastic variables  $U$  and  $V$  is defined by  $\langle U, V \rangle \equiv E\{UV^*\}$ .

Any inner product can be associated with the cosine of an angle between subspaces. In our case, we can define a magnitude-squared dual-frequency dual-wavenumber generalized coherence function as [17, 18]

$$\gamma_{\xi\xi}^2(\mathbf{k}, \boldsymbol{\kappa}; \omega, v) = \cos^2 \psi_{\xi\xi}(\mathbf{k}, \boldsymbol{\kappa}, \omega, v) = \frac{|E\{dZ(\mathbf{k}, \omega) dZ^*(\mathbf{k} - \boldsymbol{\kappa}, \omega - v)\}|^2}{E\{|dZ(\mathbf{k}, \omega)|^2\} E\{|dZ(\mathbf{k} - \boldsymbol{\kappa}, \omega - v)|^2\}}. \quad (1.19)$$

By employing the Cauchy–Schwartz inequality, one can show that

$$0 \leq \gamma_{\xi\xi}^2(\mathbf{k}, \boldsymbol{\kappa}; \omega, v) \leq 1. \quad (1.20)$$

Full coherence [ $\gamma_{\xi\xi}^2(\mathbf{k}, \boldsymbol{\kappa}; \omega, v) \equiv 1$ ] is achieved whenever [17, 18]

$$dZ(\mathbf{k} - \boldsymbol{\kappa}, \omega - v) = \alpha dZ(\mathbf{k}, \omega), \quad (1.21)$$

for some  $\alpha \in \mathbb{R}$ . If we express the stochastic infinitesimal wavefield generator by its magnitude and phase,

$$dZ(\mathbf{k}, \omega) = |dZ(\mathbf{k}, \omega)| \exp[j\phi_{\xi}(\mathbf{k}, \omega)], \quad (1.22)$$

we reach the conclusion that full coherence is obtained for those  $\mathbf{k}, \boldsymbol{\kappa}, \omega, v$  for which

$$\phi_{\xi}(\mathbf{k} - \boldsymbol{\kappa}, \omega - v) = \phi_{\xi}(\boldsymbol{\kappa}, \omega) + n\pi, \quad (1.23)$$

where  $n \in \mathbb{Z}$ , where  $\mathbb{Z}$  denotes the signed integers. This fundamental result shows that full coherence corresponds to the case where the phases of the components at  $(\mathbf{k} - \boldsymbol{\kappa}, \omega - v)$  and  $(\boldsymbol{\kappa}, \omega)$  are identical modulo  $\pi$ .

Generalizations to complex stochastic processes and to time–frequency coherences are discussed in [32, 33].

### 1.3.3 Linear Spatiotemporal Systems

Wavefields can be temporally, spatially, or spatiotemporally filtered [16]. If the wavefield  $\eta(\mathbf{r}, t)$  is filtered by a spatiotemporal impulse response  $h(\mathbf{r}, t)$ , then the linear spatiotemporal output wavefield  $\xi(\mathbf{r}, t)$  is defined by the following four-dimensional convolution integral:

$$\xi(\mathbf{r}, t) = \int h(\mathbf{r}', t')\eta(\mathbf{r} - \mathbf{r}', t - t') d\mathbf{r}' dt'. \quad (1.24)$$

If the input process  $\eta(\mathbf{r}, t)$  is stationary and spatially homogeneous, then the output filtered wavefield  $\xi(\mathbf{r}, t)$  is also stationary and homogeneous. In frequency–wavenumber space, we now readily find the conventional  $(\mathbf{k}, \omega)$  spectrum of the filtered wavefield to be given by

$$\tilde{S}_{\xi\xi}(\mathbf{k}, \omega) = |H(\mathbf{k}, \omega)|^2 \tilde{S}_{\eta\eta}(\mathbf{k}, \omega). \quad (1.25)$$

Here,  $H(\mathbf{k}, \omega) = \int h(\mathbf{r}, t) \exp[-j(\mathbf{k}^T \mathbf{r} - \omega t)] d\mathbf{r} dt / (2\pi)^4$  is the frequency–wave-number function of the linear spatiotemporal system.

### 1.3.4 Estimation

The state-of-the-art nonparametric estimator for conventional power spectral densities is David J. Thomson's multitaper estimator [34]. Thomson's estimator utilizes a set of orthonormal data tapers called discrete prolate spheroidal sequences. The beauty of the estimator is that through the tuning of a single-frequency bandwidth parameter, one attains control over spectral leakage, and one stabilizes the estimate through an inherent variance reduction offered by the estimator. The superiority of Thomson's multitaper technique has been demonstrated in numerous publications, and it has become a widespread estimator among practitioners with demanding data.

Thomson's technique can readily be extended to higher dimensional estimation problems, as has been demonstrated among others by [17, 18, 35]. We have long-term experience with higher dimensional extensions of the multitaper estimators, and our recommendation to other users is to employ the multitaper class.

## 1.4 WAVE DISPERSION

While a substantial amount of literature has been devoted to the correlation theory of stationary and homogeneous random waves and fluctuations (e.g., [6, 8, 10, 11, 16, 36]), one would have a hard time identifying contributions dealing with systems that are simultaneously nonstationary and inhomogeneous. Similarly, while dispersive effects are well treated for nonrandom space–time systems [7, 37], the combination of random fluctuations and dispersion seems to be totally lacking in the literature. In this chapter, we will take the first steps toward a full theory for random, nonstationary, inhomogeneous, and dispersive fluctuations.

### 1.4.1 Linear Systems Approach

Consider the following governing dynamical model for a random scalar spatiotemporal quantity (i.e., a scalar random field)  $\xi(\mathbf{r}, t)$ :

$$L\{\xi(\mathbf{r}, t)\} = \eta(\mathbf{r}, t). \quad (1.26)$$

Here,  $L\{\cdot\}$  is a linear spatiotemporal differential operator;  $\mathbf{r} = [x, y, z]^T$  is a three-dimensional spatial vector variable with Cartesian components  $x$ ,  $y$ , and  $z$ ;  $t$  is a temporal variable, and  $\eta(\mathbf{r}, t)$  is a random scalar space–time driving force field. (Note that the theory presented in this chapter could readily be generalized to vector and tensor fields.)

Assume that the two random fields in Eq. (1.26) are harmonizable in space and time, that is, that we can generalize Eq. (1.1) so that we can express  $\xi(\mathbf{r}, t)$  and  $\eta(\mathbf{r}, t)$  by the following four-dimensional integrals:

$$\xi(\mathbf{r}, t) = \int \exp[j(\mathbf{k}^T \mathbf{r} - \omega t)] dZ(\mathbf{k}, \omega) \quad (1.27)$$

and

$$\eta(\mathbf{r}, t) = \int \exp[j(\mathbf{k}^T \mathbf{r} - \omega t)] dW(\mathbf{k}, \omega), \quad (1.28)$$

respectively, where  $dZ(\mathbf{k}, \omega)$  and  $dW(\mathbf{k}, \omega)$  are the increment processes or generalized Fourier transforms for the two random fields,  $\mathbf{k} = [k_x, k_y, k_z]^T$  is a three-dimensional wavenumber vector variable with components  $k_x$ ,  $k_y$ , and  $k_z$ , and  $\omega$  is a scalar radian frequency variable.

A physically meaningful and sufficiently general form of the linear partial differential operator  $L\{\cdot\}$  is

$$L\{\cdot\} = \sum_{m=0}^M \sum_{n=0}^N \sum_{p=0}^P \sum_{q=0}^Q A_{m,n,p,q} \frac{\partial^m}{\partial t^m} \frac{\partial^n}{\partial x^n} \frac{\partial^p}{\partial y^p} \frac{\partial^q}{\partial z^q}, \quad (1.29)$$

where  $A_{m,n,p,q}$  are coefficients, and  $M$ ,  $N$ ,  $P$ , and  $Q$  are the respective orders of the four differential operators. By inserting Eqs. (1.27)–(1.29) into Eq. (1.26), we find that the partial differential equation can be recast into a wavenumber–frequency integral equation

$$\int D(\mathbf{k}, \omega) \exp[j(\mathbf{k}^T \mathbf{r} - \omega t)] dZ(\mathbf{k}, \omega) = \int \exp[j(\mathbf{k}^T \mathbf{r} - \omega t)] dW(\mathbf{k}, \omega) \quad (1.30)$$

where

$$D(\mathbf{k}, \omega) = \sum_{m=0}^M \sum_{n=0}^N \sum_{p=0}^P \sum_{q=0}^Q A_{m,n,p,q} (-j\omega)^m (jk_x)^n (jk_y)^p (jk_z)^q \quad (1.31)$$

is the wavenumber–frequency version of the differential operator. The function  $D(\mathbf{k}, \omega)$  is directly connected to wave dispersion, normal modes, damping, and physical resonances.

### 1.4.2 Dispersion Relation, Phase, and Group Velocities

From the previous section, we see directly that the increment processes themselves are related through the relation

$$D(\mathbf{k}, \omega) dZ(\mathbf{k}, \omega) = dW(\mathbf{k}, \omega). \quad (1.32)$$

For the undriven case,  $\eta(\mathbf{x}, t) = 0$ , so  $dW(\mathbf{k}, \omega) = 0$ , and we obtain the dispersion relation, or the equation for the normal modes as [6, 7]

$$D(\mathbf{k}, \omega) = 0. \quad (1.33)$$

The solutions to Eq. (1.33) can be written either as  $\omega = \Omega(\mathbf{k})$  or  $\mathbf{k} = \mathbf{k}(\omega)$ , for some scalar function  $\Omega(\cdot)$  and some vector function  $\mathbf{k}(\cdot)$ . Note that many fluctuating media or systems are such that several roots (or modes) result from the dispersion relation [6, 7].

Two important concepts from classical wave propagation theory are phase velocity and group velocity [4, 5, 7, 37, 38]. The phase velocity vector is defined by

$$\mathbf{v}_\phi(\mathbf{k}) = \frac{\omega(\mathbf{k})}{||\mathbf{k}||} \mathbf{u}_k, \quad (1.34)$$

where the unit wave vector is defined by  $\mathbf{u}_k = \mathbf{k}/||\mathbf{k}||$ . The group velocity vector is defined as the gradient of  $\omega(\mathbf{k})$  in wavenumber space:

$$\mathbf{v}_g(\mathbf{k}) = \frac{\partial \omega(\mathbf{k})}{\partial \mathbf{k}}, \quad (1.35)$$

where the wavenumber space gradient operator is defined by

$$\frac{\partial}{\partial \mathbf{k}} \equiv \mathbf{u}_x \frac{\partial}{\partial k_x} + \mathbf{u}_y \frac{\partial}{\partial k_y} + \mathbf{u}_z \frac{\partial}{\partial k_z}, \quad (1.36)$$

and where  $\mathbf{u}_x$ ,  $\mathbf{u}_y$ , and  $\mathbf{u}_z$  are the unit vectors along the  $x$ ,  $y$ , and  $z$  axes, respectively.

When the phase velocity vector  $\mathbf{v}_\phi(\mathbf{k})$  and the group velocity vector  $\mathbf{v}_g(\mathbf{k})$  are equal and independent of the wavenumber  $\mathbf{k}$ , the disturbance propagates dispersionless, that is, a pulse will not change its shape during propagation. In general, however, one must expect that wave dispersion occurs.

While the group velocity defines the propagation velocity of wave groups in non-absorptive media, and as such corresponds to a physical propagation, it must be borne in mind that the phase velocity does not correspond to any actual physical propagation.

### 1.4.3 Moment Functions

For random fields, the relation for the increment processes, Eq. (1.32), is by itself of little value. Instead, we must consider relevant moment functions. The most important moments are of second order, and we now construct a second-order moment function by multiplying (1.32) by its complex conjugate at a wavenumber–frequency pair  $(\mathbf{k}', \omega')$

and subsequently taking the ensemble average. This yields the following important relation:

$$E \{dZ(\mathbf{k}, \omega) dZ^*(\mathbf{k}', \omega')\} = \frac{E \{dW(\mathbf{k}, \omega) dW^*(\mathbf{k}', \omega')\}}{D(\mathbf{k}, \omega) D^*(\mathbf{k}', \omega')}. \quad (1.37)$$

Note that the moment functions  $E \{dZ(\mathbf{k}, \omega) dZ^*(\mathbf{k}', \omega')\}$  and  $E \{dW(\mathbf{k}, \omega) dW^*(\mathbf{k}', \omega')\}$  are direct generalizations of the standard Loève spectrum in Eq. (1.4) to dual frequency/dual wavenumber. From Eq. (1.37) we understand that this is in fact a generalized resonance theory for systems that may be simultaneously nonstationary in time, inhomogeneous in space, and include linear dispersive effects.

It is often more convenient to work in terms of global and local space and time coordinates, and global and local frequency and wavenumber coordinates. Let  $\mathbf{x}$  and  $\boldsymbol{\rho}$  be global and local spatial positions, respectively; let  $t$  and  $\tau$  be global and local time, respectively; let  $\mathbf{k}$  and  $\boldsymbol{\kappa}$  be global and local wavenumber vectors, respectively; and let  $\omega$  and  $\nu$  be global and local frequencies, respectively. Then we can rewrite the dual-frequency/dual-wavenumber spectrum in Eq. (1.37) as

$$S_\xi(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu) = \frac{S_\eta(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu)}{Q(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu)}, \quad (1.38)$$

where

$$\begin{aligned} S_\xi(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu) \frac{dk d\boldsymbol{\kappa} d\omega d\nu}{(2\pi)^8} &= E [dZ(\mathbf{k}, \omega) dZ^*(\mathbf{k} - \boldsymbol{\kappa}, \omega - \nu)], \\ S_\eta(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu) \frac{dk d\boldsymbol{\kappa} d\omega d\nu}{(2\pi)^8} &= E [dW(\mathbf{k}, \omega) dW^*(\mathbf{k} - \boldsymbol{\kappa}, \omega - \nu)], \end{aligned}$$

and

$$Q(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu) = D(\mathbf{k}, \omega) D^*(\mathbf{k} - \boldsymbol{\kappa}, \omega - \nu).$$

It is now possible to formulate a spatiotemporal frequency–wavenumber spectrum that includes linear dispersion and that directly generalizes the Kirkwood–Rihaczek spectrum. By a Fourier transformation over local frequency and local wavenumber space, we can write

$$R_\xi(\mathbf{k}, \mathbf{r}; \omega, t) = \int \int \frac{S_\eta(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu)}{Q(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu)} \exp[j(\nu t - \boldsymbol{\kappa}^T \mathbf{r})] \frac{d\nu d\boldsymbol{\kappa}}{(2\pi)^4}. \quad (1.39)$$

We see that the resonances of the system, that is, where  $Q(\mathbf{k}, \boldsymbol{\kappa}; \omega, \nu) \simeq 0$ , are of special importance for the time–frequency/space–wavenumber spectrum. Often, the integration may be carried out by means of classical (but multidimensional) complex residue calculation.

**1.4.3.1 Stationary and Homogeneous Driving Force Field** The stationary manifold discussed in [25] can now be generalized to a *stationary and homogeneous manifold* described by  $\nu = 0$  and  $\boldsymbol{\kappa} = \mathbf{0}$ . In terms of the second-order moment function for the driving term, this translates to the relation

$$E \{|dZ(\mathbf{k}, \omega)|^2\} = \frac{E \{|dW(\mathbf{k}, \omega)|^2\}}{|D(\mathbf{k}, \omega)|^2}. \quad (1.40)$$

Here,  $E\{|dZ(\mathbf{k}, \omega)|^2\}$  and  $E\{|dW(\mathbf{k}, \omega)|^2\}$  are the conventional wavenumber–frequency spectra for stationary and homogeneous fields.

#### 1.4.4 Applications

By choosing the appropriate linear differential operator  $L\{\cdot\}$  in the random dynamical model, one can derive the crucial function  $D(\mathbf{k}, \omega)$ , which lies at the heart in the description of the second-order dual-frequency/dual-wavenumber and the time–frequency/space–wavenumber moment functions. In the following, we derive  $D(\mathbf{k}, \omega)$  for a few relevant and physically important one-dimensional models. Armed with this function, one can in principle calculate the generalized Loéve spectrum, and the generalized Kirkwood–Rihaczek spectrum for dispersive, nonstationary, and inhomogeneous random fields.

**1.4.4.1 Nondispersive Wave Equation** The ubiquitous (one-dimensional) linear wave equation (e.g., [4, 6, 7]) has the second-order differential operator

$$L = \frac{\partial^2}{\partial t^2} - c^2 \frac{\partial^2}{\partial x^2}, \quad (1.41)$$

where  $c$  is a constant wave propagation speed. This model appears in a number of linearized acoustic, electromagnetic, and mechanical systems. This operator leads to dispersionless propagation of disturbances. However, it is still of great practical interest to solve for the resulting fluctuation spectra using the theory of this chapter since inhomogeneities driving fields would be included in our description.

The orders and coefficients of the basic linear operator in Eq. (1.29) are  $p = q = 2$ , and  $A_{0,0} = A_{1,0} = A_{0,1} = A_{1,1} = A_{2,1} = A_{1,2} = A_{2,2} = 0$ , and  $A_{2,0} = 1$ ,  $A_{0,2} = c^2$ . The corresponding wavenumber–frequency operator becomes

$$D(k, \omega) = c^2 k^2 - \omega^2. \quad (1.42)$$

By inserting this operator into Eqs. (1.38) and (1.39), we can explicitly derive the second-order properties of the inhomogeneous random field for nondispersive media.

**1.4.4.2 Euler–Bernoulli Beam Equation** A classical linear model for a deflecting beam under axial loading is the Euler–Bernoulli beam equation. Using Newton’s second law, one can show that one useful form of the beam equation leads to the fourth-order differential operator [39]

$$L = \frac{\partial^2}{\partial t^2} + \gamma^2 \frac{\partial^4}{\partial x^4} - \beta^2 \frac{\partial^4}{\partial t^2 \partial x^2}, \quad (1.43)$$

where  $\gamma^2$  and  $\beta^2$  are bulk mechanical parameters for the beam. This is evidently a dispersive model, with associated wavenumber–frequency operator

$$D(k, \omega) = -\omega^2 - \gamma^2 k^4 - \beta^2 \omega^2 k^2. \quad (1.44)$$

It will now be possible to study the nonstationary and inhomogeneous characteristics for a dispersive beam under nonstationary and inhomogeneous loading using the framework described in this chapter.

**1.4.4.3 Long Water Waves** Two interesting linear models that appear in approximate theories of long water waves are the linear Korteweg–de Vries (KdV) equation and the linear Boussinesq equation [7]. The partial differential operator for the linear KdV equation is

$$L = \frac{\partial}{\partial t} + c_0 \frac{\partial}{\partial x} + \beta \frac{\partial^3}{\partial x^3}, \quad (1.45)$$

with associated wavenumber–frequency operator

$$D(k, \omega) = j(c_0 k - \beta k^3 - \omega), \quad (1.46)$$

where  $c_0$  and  $\beta$  are constant parameters.

The partial differential operator for the linear Boussinesq equation reads

$$L = \frac{\partial^2}{\partial t^2} - \alpha^2 \frac{\partial^2}{\partial x^2} - \beta^2 \frac{\partial^4}{\partial x^2 \partial t^2}, \quad (1.47)$$

with an associated wavenumber–frequency operator

$$D(k, \omega) = \alpha k^2 - \beta^2 \omega^2 k^2 - \omega^2, \quad (1.48)$$

where  $\alpha$  and  $\beta$  are constant parameters.

**1.4.4.4 Plasma Waves** A plasma is a partially ionized gas with very rich dynamics. Since a plasma will be governed mainly by electromagnetic effects, it can support a host of wave modes that do not even exist in neutral gases or fluids. As the plasma models soon become very complicated, we will not list any models in terms of their partial differential equations. We will, however, list the wavenumber–frequency operators for a few linear but dispersive modes, to indicate what kind of models one can expect. To derive these operators, one would have to treat the plasma as a dielectric medium, and take the full set of Maxwell's equations as the starting dynamical model. Depending on the assumptions made, one may derive a variety of different wave equations.

If the direction of the fluctuating electric field is the same as the wavenumber, one calls the fluctuations “electrostatic” (note that there is nothing “static” about these waves—they are highly dynamic). The high-frequency electrostatic plasma waves can be shown to have the following wavenumber–frequency operator (e.g., [40]):

$$D(k, \omega) = \omega^2 - \omega_p^2 - 3c_{\text{th}}^2 k^2, \quad (1.49)$$

where  $\omega_p$  is the so-called plasma frequency, and  $c_{\text{th}}$  is electron thermal speed.

Another often discussed operator is the one that describes the so-called ion-acoustic waves in the plasma (e.g., [40])

$$D(k, \omega) = \omega^2(1 + k^2 \lambda_D^2) - c_s^2 k^2, \quad (1.50)$$

where  $c_s$  is the ion-acoustic speed, and  $\lambda_D$  is the Debye length or screening length of the plasma.

### 1.4.5 Example

As an illustrative analytical demonstration of how a time–frequency/space–wavenumber spectrum may be derived from this theory, we make two assumptions. First, we assume that the linear partial differential operator has the form

$$L = \frac{\partial^2}{\partial t^2} + \gamma \frac{\partial}{\partial x}, \quad (1.51)$$

where  $\gamma > 0$  is a constant. Then, the corresponding operator in the transform domain reads

$$D(k, \omega) = j\gamma k - \omega^2. \quad (1.52)$$

Second, we assume that the spectrum of the source random field  $\eta(x, t)$  is stationary in time. Then we may formulate the source Loève spectrum as

$$S_\eta(k, \kappa; \omega, \nu) = 2\pi \tilde{S}_\eta(k, \kappa; \omega) \delta(\nu), \quad (1.53)$$

for a function  $\tilde{S}_\eta(k, \kappa; \omega)$ , which is independent of the local frequency  $\nu$ .

With these assumptions, it is easy to carry out the integral over  $\nu$  in Eq. (1.39), so that the two-dimensional integral for the response field reduces to the following one-dimensional integral over  $\kappa$ :

$$R_\xi(k, x; \omega, t) = \int \frac{\tilde{S}_\eta(k, \kappa; \omega) \exp(-j\kappa x)}{(j\gamma k - \omega^2)[j\gamma(\kappa - k) - \omega^2]} \frac{d\kappa}{2\pi}. \quad (1.54)$$

To carry out the integral in Eq. (1.54), we use ordinary residue integration in the complex plane. We see that the integrand has a simple pole at  $\kappa = k - j\omega^2/\gamma$ . Then, through the residue theorem (e.g., [38]), we can readily complete the  $\kappa$  integral to obtain the closed-form solution:

$$R_\xi(k, x; \omega, t) = \frac{\tilde{S}_\eta(k, k - j(\omega^2/\gamma); \omega)}{j\gamma k - \omega^2} e^{jkx + j\pi/2 - \omega^2 x/\gamma}. \quad (1.55)$$

We observe that the spatiotemporal frequency–wavenumber spectrum for the response random field in this case becomes independent of global time  $t$ , due to the stationarity assumption for the source field  $\eta(x, t)$ . Still, the spectrum is a nontrivial function of space, wavenumber, and frequency. Note also that the complex residue integration caused the wavenumber–frequency dispersion operator to appear explicitly as an argument in the source random field.

The generalized Kirkwood–Rihaczek spectrum we found in Eq. (1.55) is complex valued, as expected. In practice, however, one often displays the modulus of  $R_\xi(k, x; \omega, t)$ , which in our case simplifies to

$$|R_\xi(k, x; \omega, t)| = \frac{|\tilde{S}_\eta(k, k - j\frac{\omega^2}{\gamma}; \omega)|}{\sqrt{\gamma^2 k^2 + \omega^4}} e^{-\omega^2 x/\gamma}. \quad (1.56)$$

Now, assume that the spatial inhomogeneity of the source is such that all the source wavenumbers are equally correlated (implying a very inhomogeneous source), then we may formulate the source spectrum as

$$\tilde{S}_\eta \left( k, k - j \frac{\omega^2}{\gamma}; \omega \right) = \alpha F(\omega), \quad (1.57)$$

for some constant  $\alpha$  independent of  $k$ , and some source frequency spectrum function  $F(\omega)$ . For this particular example, the modulus of the spatiotemporal frequency–wavenumber spectrum collapses to the following relatively simple form:

$$|R_{\xi}(k, x; \omega, t)| = \frac{|\alpha| F(\omega)}{\sqrt{\gamma^2 k^2 + \omega^4}} e^{-\omega^2 x / \gamma}. \quad (1.58)$$

We see that for this particular example, the spectral decay is much faster in the frequency variable  $\omega$  than in the wavenumber variable  $k$ . This is a property inherited from properties of the wavenumber–frequency operator  $D(k, \omega)$ , and it shows us how a perturbation of the dynamical system is influenced by dispersion and damping.

## 1.5 CONCLUSIONS

We have presented a physical-statistical review of the theory for harmonizable stochastic wavefields. Our approach emphasized the wavelike aspects of the wavefields. In particular, we derived a systematic way of dealing with simultaneously nonstationary and inhomogeneous stochastic wavefields.

As demonstrated, we could generalize the conventional Einstein–Wiener–Khintchine relation to a relation between four different second-order moment functions. We found it beneficial and intuitively appealing to express the moment functions in terms of global and local spatial, temporal, wavenumber, and frequency variables. The relevant moment functions apparent from our reasoning were (1) a dual-space dual-time correlation, (2) a spatiotemporal frequency–wavenumber correlation, generalizing the conventional Kirkwood–Rihaczek time–frequency spectrum, (3) a dual-wavenumber dual-frequency correlation, generalizing the conventional Loève dual-frequency spectrum, and (4) another spatiotemporal frequency–wavenumber spectrum generalizing the conventional ambiguity function.

We presented a general and formalistic framework for how to handle linear wave dispersion in conjunction with stochastic system theory. As shown, the linear dispersion could readily be included in the theory for stochastic wavefields, and we were able to formulate all the relevant second-order moment functions in a rather elegant fashion. We believe that in array processing applications, our framework could be used to improve beamforming, delay estimation, signal estimation, and imaging in nonstationary, inhomogeneous, and dispersive environments. Spatiotemporal frequency–wavenumber adaptive algorithms generalizing the results from [41] seems desirable for numerous array processing tasks and applications in nonstationary, inhomogeneous, and dispersive environments.

## ACKNOWLEDGMENTS

The author acknowledges The Research Council of Norway for generous support under project 162831/V30. Discussions with Simon Haykin, Yngve Birkelund, and Heidi Hindberg are greatly appreciated.

## REFERENCES

1. A. M. Yaglom, *An Introduction to the Theory of Stationary Random Functions*, Englewood Cliffs, NJ: Prentice-Hall, 1962.
2. W. Paul and J. Baschnagel, *Stochastic Processes: From Physics to Finance*, Berlin: Springer-Verlag, 1999.
3. M. B. Priestley, *Non-Linear and Non-Stationary Time Series Analysis*, London: Academic, 1988.
4. J. D. Jackson, *Classical Electrodynamics*, New York: Wiley, 1992.
5. L. D. Landau and E. M. Lifshitz, *Fluid Mechanics*, 2nd ed., Oxford: Pergamon, 1987.
6. H. Pécseli, *Fluctuations in Physical Systems*, Cambridge: Cambridge University Press, 2000.
7. G. B. Whitham, *Linear and Nonlinear Waves*, New York: Wiley, 1974.
8. Yu. A. Buevich and V. V. Butkov, “Fluctuations and transport in an electric field,” *J. Eng. Phys. Thermophys.*, vol. 46, pp. 282–287, Mar. 1984.
9. T. Hagfors, “Some properties of radio waves reflected from the moon and their relation to the lunar surface,” *J. Geophys. Research*, vol. 66, pp. 777–785, 1961.
10. S. M. Rytov, Yu. A. Kravtsov, and V. I. Tatarskii, *Principles of Statistical Radiophysics 2: Correlation Theory of Random Processes*, Berlin: Springer-Verlag, 1988.
11. S. M. Rytov, Yu. A. Kravtsov, and V. I. Tatarskii, *Principles of Statistical Radiophysics 3: Elements of Random Fields*, Berlin: Springer-Verlag, 1989.
12. M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*, Springer, 2001.
13. J. Kiyono, “Simulation of stochastic waves in a non-homogeneous random field,” *Soil Dynam. Earthquake Eng.*, vol. 14, pp. 387–396, 1995.
14. M. G. Maginness, “The reconstruction of elastic wave fields from measurements over a transducer array,” *J. Sound Vib.*, vol. 20, pp. 219–240, 1972.
15. H. Wackernagel, *Multivariate Geostatistics: An Introduction with Applications*, 3rd ed., Springer, 2003.
16. J. F. Böhme, “Array processing,” in *Advances in Spectrum Analysis and Array Processing*, Vol. II, S. Haykin (Ed.), Englewood Cliffs, NJ: Prentice Hall, 1991.
17. Y. Larsen and A. Hanssen, “Spectral properties of nonstationary and inhomogeneous random fields,” in *Proc. 37th Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 9–12, 2003.
18. Y. Larsen, “Spectral properties of harmonizable random processes and fields,” PhD dissertation, University of Tromsø, Tromsø, Norway, 2003.
19. L. Cohen, *Time-Frequency Analysis*, Englewood Cliffs, NJ: Prentice-Hall, 1995.
20. R. J. Mellors, F. L. Vernon, and D. J. Thomson, “Detection of dispersive signals using multitaper dual-frequency coherence,” *Geophys. J. Int.*, vol. 135, pp. 146–154, 1998.

21. H. Cramér, “On the theory of stationary random processes,” *Ann. Math.*, vol. 41, pp. 215–230, 1940.
22. M. Loèvè, *Probability Theory*, 3rd ed., Princeton, NJ: Van Nostrand, 1963.
23. S. Cambanis and B. Liu, “On harmonizable stochastic processes,” *Inform. Control*, vol. 17, pp. 183–203, 1970.
24. A. M. Yaglom, *Correlation Theory of Stationary and Related Random Functions*, New York: Springer-Verlag, 1987.
25. A. Hanssen and L. L. Scharf, “A theory of polyspectra for nonstationary stochastic processes,” *IEEE Trans. Signal Process.*, vol. 51, pp. 1243–1252, May 2003.
26. T. Hagfors, “The description of a random propagation circuit by the coherence between adjacent frequencies,” in *Proceedings of the Symposium on Electromagnetic Theory and Antennas*, Copenhagen, June 25–30, 1962.
27. T. Hagfors, “Time-varying propagation circuits, description and applications,” *J. Atmospher. Solar-Terrestrial Phys.*, vol. 63, pp. 215–220, 2001.
28. J. G. Kirkwood, “Quantum statistics of almost classical assemblies,” *Phys. Rev.*, vol. 44, pp. 31–37, 1933.
29. A. W. Rihaczek, “Signal energy distribution in time and frequency,” *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 369–374, 1968.
30. M. Rosenblatt, *Stationary Sequences and Random Fields*, Boston, MA: Birkhäuser, 1985.
31. E. Vanmarcke, *Random Fields: Analysis and Synthesis*, Cambridge, MA: MIT Press, 1983.
32. H. Hindberg and A. Hanssen, “Generalized spectral coherences for complex-valued harmonizable processes,” *IEEE Trans. Signal Process.*, vol. 55, pp. 2407–2344, Nov. 2007.
33. L. L. Scharf, P. J. Schreier, and A. Hanssen, “The Hilbert space geometry of the Rihaczek distribution for stochastic analytic signals,” *IEEE Signal Process. Lett.*, vol. 12, pp. 297–300, 2005.
34. D. J. Thomson, “Spectrum estimation and harmonic analysis,” *Proc. IEEE*, vol. 70, pp. 1055–1096, Sept. 1982.
35. A. Hanssen, “Multidimensional multitaper spectral estimation,” *Signal Process.*, vol. 58, pp. 327–332, Feb. 1997.
36. H. B. Callen and T. A. Welton, “Irreversibility and generalized noise,” *Phys. Rev.*, vol. 83, pp. 34–40, 1951.
37. L. Cohen, “Pulse propagation in dispersive media,” in *Proc. Tenth IEEE Workshop on Statistical and Array Processing*, Pocono Manor, PA, Aug. 14–16, 2000, pp. 485–489.
38. L. Brillouin, *Wave Propagation and Group Velocity*, New York: Academic, 1960.
39. L. Brekhovskikh and V. Goncharev, *Mechanics of Continua and Wave Dynamics*, Berlin: Springer-Verlag, 1985.
40. G. K. Parks, *Physics of Space Plasmas*, Redwood City, CA: Addison-Wesley, 1991.
41. S. Haykin and D. J. Thomson, “Signal detection in a nonstationary environment reformulated as an adaptive pattern classification problem,” *Proc. IEEE*, vol. 86, pp. 2325–2344, Nov. 1998.
42. H. Hindberg, Y. Birkelund, T. A. Øigård, and A. Hanssen, “Kernel-based estimators for the Kirkwood-Rihaczek time-frequency spectrum,” in *Proc. European Signal Processing Conference (EUSIPCO)*, Florence, Italy, Sept. 4–8, 2006.

---

## CHAPTER 2

---

# Spatial Spectrum Estimation

Petar M. Djurić

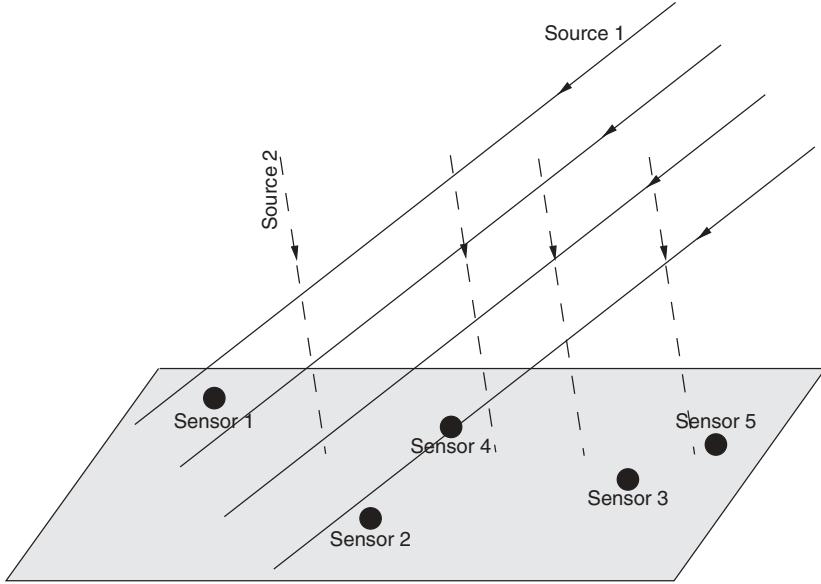
Stony Brook University, Stony Brook, New York

### 2.1 INTRODUCTION

In practice, spectrum estimation is usually a preliminary step of many signal processing problems. When we analyze temporal signals and when they satisfy certain statistical conditions, we apply *temporal* spectrum estimation that yields the distribution of the signal power over frequency. This is important for subsequent signal analysis, which may include developing parametric models that provide improved description of the data or building filters for suppressing noise and passing useful portions of the signal. For example, spectrum estimation of a temporal sequence with unknown spectral contents can reveal if the signal samples contain harmonic signals or not. If they do, one can propose a model for the data that can extract the number of signals in the data as well as their parameters.

Spectrum estimation is also very important in the analysis of signals obtained by an array of passive sensors and is referred to as *spatial* spectrum estimation. In general,  $L$  identical sensors may be deployed in a sensor field in an arbitrary or predefined way with their exact locations being known. The sensors are basically antennas, hydrophones, or seismometers, and they take measurements that can be electromagnetic, acoustic, or vibrational signals. The received signals are possibly emitted by more than one source. The problem of spatial spectrum estimation is to find the distribution of the received signal power in space and thereby determine the number and the location of the sources in the field. Large values of power spectral density (PSD) in a specific part of the space would most likely indicate the presence of one or more radiating sources and vice versa; low values of PSD would mean absence of such sources. Due to the importance of the problem, spatial spectrum estimation methods have been of great interest in various fields including wireless communications, radar, sonar, speech, astronomy, biomedicine, and seismology.

Thus, as with temporal spectrum estimation, spatial spectrum estimation as a first step of signal processing provides information about the “nature” of the data, how many signals the data contain, how much noise, and so on. In this chapter, the emphasis is on spatial spectrum estimation. However, before we address the methods for spatial spectrum estimation, we review the basics of temporal spectrum estimation because they represent the foundations of the spatial methods.



**Figure 2.1** Sensor array in sensor field with two impinging signals.

In the context of spatial spectrum estimation, we distinguish two general classes of problems. In the first class, the sources of the signals are relatively near to the sensors, and the other is when they are in the far field of the array. In this chapter, we focus on the latter problem. If the sources are far away, one can model the received wavefronts as plane waves. In Figure 2.1, we see a field with five sensors and two impinging signals. If in this scenario the task is to locate the sources, the best one can do is to estimate the directions of arrival (DOAs) of the signals. To that end one exploits the measurements of the sensors made simultaneously, which amounts to spatial sampling of the signals.

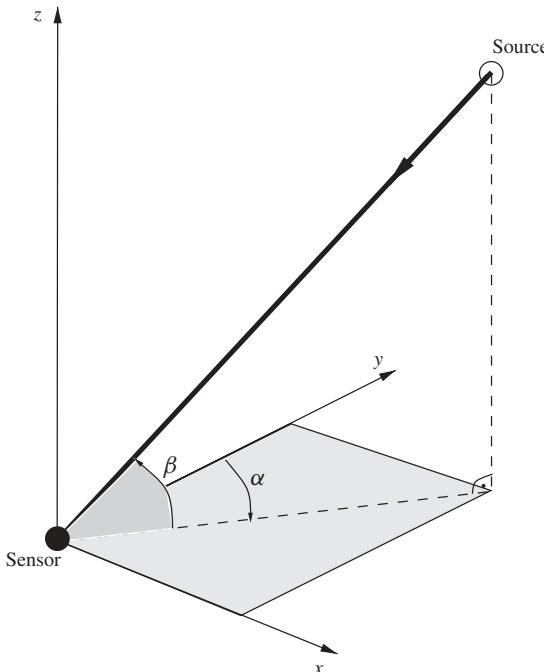
In Figure 2.2, we see how we describe the direction of arrival of the signal. The sensors are all considered to be positioned in a plane, and the DOA of the incoming signal is defined by two angles, the azimuth ( $\alpha$ ) and the elevation ( $\beta$ ). In this chapter, for most of it, we consider even a simpler geometry, where  $\alpha = 0$ .

Figure 2.3 shows a uniform array of sensors. The array in Figure 2.3 is called uniform because the distance between the sensors is kept constant. In the figure, the array and the sources are confined to a plane. There, we define the DOA as  $\theta$ , which in terms of the elevation angle  $\beta$  is expressed by  $\theta = \pi/2 - \beta$ .

We refer to the vector of measurements taken at time instant  $t$  as a snapshot, and we denote it by  $\mathbf{y}[t]$ , where

$$\mathbf{y}[t] = [y_1[t] y_2[t] \dots y_L[t]]^T$$

with  $y_1[t], y_2[t], \dots, y_L[t]$  being the samples of the signal obtained by the first, second, and the  $L$ th sensor, respectively. In general, we may have more than one snapshot available for analysis, for example,  $T$  snapshots, which we denote by  $\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{T-1}$ , where  $T > 1$ . Therefore, besides spatially sampling the signals, we also sample them in the time domain. The analysis is then done on spatially and temporally sampled



**Figure 2.2** Geometry of impinging signal with respect to sensor plane.

data. An important concept here is the array's *aperture*, which represents the space occupied by the array measured in units of the received signal wavelength.

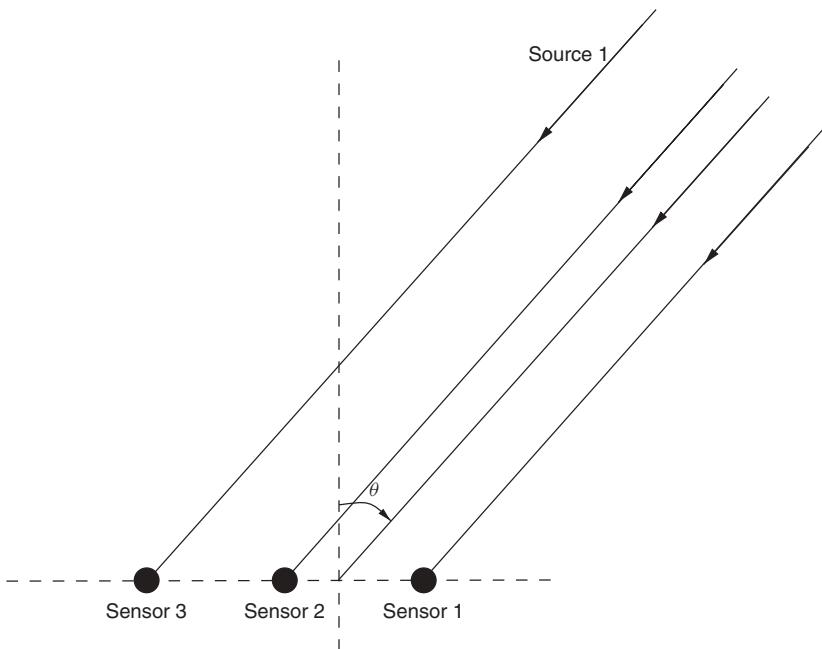
We can construct an  $L \times T$  matrix with the measurements arranged according to

$$\mathbf{Y} = \begin{bmatrix} y_1[0] & y_1[1] & y_1[2] & \cdots & y_1[T-1] \\ y_2[0] & y_2[1] & y_2[2] & \cdots & y_2[T-1] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_L[0] & y_L[1] & y_L[2] & \cdots & y_L[T-1] \end{bmatrix}, \quad (2.1)$$

where the snapshots are the columns of  $\mathbf{Y}$ . We can interpret the columns of  $\mathbf{Y}$  as vectors obtained from realizations of a sampled continuous-space random process, as they represent variation of a quantity over a region of space, for example, the variation of acoustic signal over space. The rows of  $\mathbf{Y}$  on the other hand are vectors obtained from realizations of a sampled continuous-time random process, for example, the variation of a radio signal over time as seen by a particular sensor. Clearly, the elements of  $\mathbf{Y}$  show how a quantity of interest varies over both time and space.

The theory of spatial spectrum estimation builds on its counterpart of temporal spectrum estimation. In other words, many of the methods that are used for estimating the PSD of temporal signals are also used for finding the PSD of spatial signals. Due to this similarity, we briefly review the basics of classical nonparametric and parametric spectrum estimation of temporal signals before addressing spatial spectrum estimation.

The history of spectrum estimation can be traced back to ancient times [1]. The discoveries that provided the foundation of todays' spectrum estimation theory were made in the eighteenth century, perhaps most prominently with the advance of Fourier



**Figure 2.3** Geometry of uniform linear array.

theory. It was followed by Sturm–Liouville spectral theory of differential equations and Von Neuman's and Wiener's spectral representation in quantum and classical physics. The most well-known method for spectrum estimation, the periodogram, was introduced by Schuster in 1898 when he was searching for hidden periodicities in sunspot data [2]. The use of spectrum estimation in statistical theory dates back to 1949 when Tukey proposed numerical methods for computing spectra from empirical data. In 1965, Cooley and Tukey reinvented the fast Fourier transform (FFT) [3], which is an efficient algorithm for computing the discrete Fourier transform. Ever since, this work has been influencing the research on spectrum estimation. Soon after, Burg proposed an alternative way of computing spectra, fundamentally different and based on the principle of maximum entropy [4, 5]. This principle has led to an explosion of research of so-called parametric spectrum estimation methods.

A related problem to spectrum estimation is the estimation of the number of signals, their parameters, and waveforms. This problem is more than 200 years old and dates back to 1795 when Gaspard Riche, the Baron de Prony, published his work related to the study of fitting data with superimposed exponentials. Capon's work from more than three decades ago on designing spatial filters has had its own impact on the further development of spectrum estimation theory [6]. He proposed a method that is based on a bank of filters where each filter passes the signal on which it is tuned without distortions while minimizing the powers of all the other signals present in the data. Another class of important methods, known as subspace-based methods, was born when researchers started looking at the structure of the covariance matrix of the data collected by the sensors. Early work on this approach is due to Hotelling [7], Koopmans [8], and later due to Pisarenko [9]. More recently, the multiple signal classification (also known as MUSIC) algorithm [10, 11], which was originally described as a DOA estimator, drew

also a lot of attention. Currently, the field of spectrum estimation, although considered mature, undergoes steady advances as new contributions continue to appear in various journal publications. For more elaborate accounts of the history of spectrum estimation, the reader is referred to [1, 12].

The organization of the rest of the chapter is as follows. In Section 2.2 we explain the fundamental problem of spectrum estimation. In Section 2.3, we provide a brief review of temporal spectrum estimation methods. The main material of the chapter is in Section 2.4, where first we outline the mathematical model of the data and then describe the nonparametric and parametric methods. We note that here we only present the basics of spatial spectrum estimation. More recent books that cover the subject in more detail are [13] and [14] and two excellent tutorials are [15] and [16].

## 2.2 FUNDAMENTALS

The basic problem of spectrum estimation is extracting spectral information from a temporal and/or spatial set of data that represents a stationary random process. More specifically, the objective is to obtain the distribution of the power of the data as a function of frequency.<sup>1</sup> This is considered to be one of the most fundamental problems of statistical signal processing.

A random process by definition is an infinite collection of random variables. In this chapter we are interested in discrete-time and/or discrete-space wide-sense stationary random processes. Wide-sense stationarity implies that the mean of the random process does not depend on time and/or space. In other words, if, for example, we have a discrete-time random process denoted by  $\mathcal{Y}[t]$ , where  $t$  is an integer, then

$$E(\mathcal{Y}[t]) = \mu_y, \quad (2.2)$$

where the operator  $E(\cdot)$  stands for expectation. In this chapter, we assume that the random processes have zero mean. For wide-sense stationarity, we need that the autocovariance sequence of  $\mathcal{Y}[t]$ , which is defined by

$$r[t, t - k] = E(\mathcal{Y}[t]\mathcal{Y}^*[t - k]), \quad (2.3)$$

where  $k$  represents the autocovariance lag, satisfies the following identity<sup>2</sup>:

$$r[t, t - k] = r[k]. \quad (2.4)$$

The above implies that the autocovariance sequence is only a function of the lag  $k$ .

By Wold's theorem, the PSD of the random process  $\mathcal{Y}[t]$  is obtained from the discrete-time Fourier transform (DTFT) of the autocovariance sequence of  $\mathcal{Y}[t]$ ,  $r[k]$  [17]. If we denote the PSD by  $p(\omega)$ , where  $\omega$  is the angular frequency measured in radians/sampling interval,<sup>3</sup> we have [17]

$$p(\omega) = \sum_{k=-\infty}^{\infty} r[k]e^{-j\omega k}, \quad -\pi \leq \omega < \pi. \quad (2.5)$$

It is well known that  $p(\omega)$  is a periodic function in  $\omega$ .

<sup>1</sup>When we deal with spatial data, the frequency is known as spatial frequency.

<sup>2</sup> $\mathcal{Y}^*[t - k]$  signifies the complex conjugate of  $\mathcal{Y}[t - k]$ .

<sup>3</sup>To obtain the frequency in cycles per sampling interval, we use  $f = \omega/2\pi$ .

We classify the random processes according to their PSDs in three groups, that is, processes with:

1. Purely continuous spectra;  $p(\omega)$  is an absolutely continuous function of  $\omega$ .
2. Line spectra;  $p(\omega)$  is identically equal to zero for all  $\omega$  except at some frequencies  $\omega_1, \omega_2, \dots$ , where the PSD takes infinite values.
3. Mixed spectra; a combination of continuous and line spectra.

In practice, we often observe a sequence  $y[t], t = 0, 1, \dots, T - 1$  from the random process  $\mathcal{Y}[t]$  or  $y_l$  from  $\mathcal{Y}_l$ , where the latter is a spatial process. Given these sequences, the objective is to estimate the PSD of  $\mathcal{Y}[t]$  or  $\mathcal{Y}_l$ . Next, we briefly explain some methods for temporal spectrum estimation.

## 2.3 TEMPORAL SPECTRUM ESTIMATION

We can classify the methods for PSD estimation as nonparametric and parametric methods. In general, if the method is not based on any specific assumption about the observed sequence  $y[t]$  except that it comes from a random process that is wide-sense stationary, we call the method a nonparametric one. Otherwise, if it is based on some assumed parametric model of  $y[t]$ , it is a parametric method. We proceed first by outlining some nonparametric methods and then by reviewing parametric ones.

### 2.3.1 Nonparametric Methods

**2.3.1.1 Correlogram** Theoretically, the PSD of a random process is obtained by the DTFT of the autocovariance function of the process as per (2.5). Then, one obvious way of obtaining the PSD estimate of  $\mathcal{Y}[t]$  would be to estimate the autocovariance sequence of the process,  $\hat{r}[k]$ , from the finite set of data samples  $y[t], t = 0, 1, 2, \dots, T - 1$  and then take the DTFT of  $\hat{r}[k]$ . There are two standard ways of computing the estimate of  $r[k]$ , one that is based on

$$\hat{r}[k] = \frac{1}{T-k} \sum_{t=k}^{T-1} y[t]y^*[t-k], \quad k = 0, 1, \dots, T-1, \quad (2.6)$$

and another on

$$\hat{r}[k] = \frac{1}{T} \sum_{t=k}^{T-1} y[t]y^*[t-k], \quad k = 0, 1, \dots, T-1. \quad (2.7)$$

In both cases for negative  $k$  we use

$$\hat{r}[-k] = \hat{r}^*[k]. \quad (2.8)$$

Then the PSD of  $\mathcal{Y}[t]$  is found from

$$\hat{p}_c(\omega) = \sum_{k=-T+1}^{T-1} \hat{r}[k]e^{-j\omega k}. \quad (2.9)$$

The estimate in (2.9) is known as a correlogram [13], which is why we added the subscript  $c$  to the estimate,  $\hat{p}_c(\omega)$ .

Of the two estimates of  $r[k]$ , the one given by (2.6) represents an unbiased estimate and the one by (2.7) a biased one. The more commonly used is the biased estimate because on average it provides more accurate estimates of  $r[k]$  for medium and large values of  $k$ , and it is guaranteed to be a positive semidefinite sequence (a property that autocovariance sequences should satisfy). The latter implies that the estimated PSD is guaranteed to be nonnegative for any  $\omega$ .

**2.3.1.2 Periodogram** Another common estimator of  $p(\omega)$  is the periodogram of Schuster [2], which is defined by

$$\hat{p}_{\text{per}}(\omega) = \frac{1}{T} \left| \sum_{t=0}^{T-1} y[t] e^{-j\omega t} \right|^2. \quad (2.10)$$

We note that if we use the biased estimate of  $r[k]$  in (2.9), we obtain the periodogram defined by (2.10). Also, if we define the discrete set of frequencies  $\omega_k = 2\pi k/T$ , for  $k = 0, 1, \dots, T - 1$ , we can write for the discrete Fourier transform (DFT) of  $y[t]$ ,

$$Y[k] = \sum_{t=0}^{T-1} y[t] e^{-j2\pi k/T} \quad (2.11)$$

and from it directly obtain the periodogram computed at the set of frequencies  $\omega_k$ ,  $k = 0, 1, \dots, T - 1$ ,

$$\hat{p}_{\text{per}}(\omega_k) = \frac{1}{T} |Y[k]|^2. \quad (2.12)$$

We point out that we can compute (2.11) by the FFT.

The periodogram is an estimator of the PSD of the random process  $\mathcal{Y}[t]$ , and so it is important to know its statistical properties. It can be shown that the periodogram is asymptotically unbiased, which means that

$$\lim_{T \rightarrow \infty} E(\hat{p}_{\text{per}}(\omega)) = p(\omega). \quad (2.13)$$

On the other hand, the variance of the periodogram does not tend to zero as  $T \rightarrow \infty$ , which entails that the periodogram is not a consistent estimator. In general, we have [13, 18]

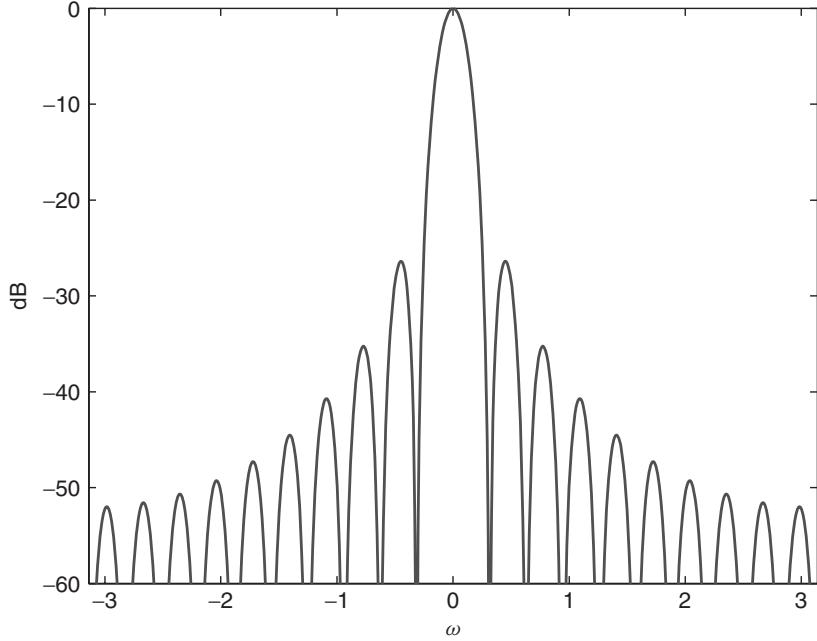
$$\lim_{T \rightarrow \infty} \text{var}(\hat{p}_{\text{per}}(\omega)) = p^2(\omega). \quad (2.14)$$

A final remark about the periodogram refers to an important interpretation of it, where its expectation is expressed by

$$E(\hat{p}_{\text{per}}(\omega)) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |W_B(\omega - \xi)|^2 p(\xi) d\xi, \quad (2.15)$$

where  $W_B(\omega)$  is the DTFT of a *triangular* window, known also as the Fejer kernel, and given by

$$W_B(\omega) = \frac{1}{T} \left[ \frac{\sin(\omega T/2)}{\sin(\omega/2)} \right]^2. \quad (2.16)$$



**Figure 2.4** Plot of  $20 \log_{10} W_R(\omega)/W_R(0)$  for  $T = 20$ .

A plot of  $W_B(\omega)$  is shown in Figure 2.4, where the length of the data record is assumed to be  $T = 20$  samples. Thus, (2.15) implies that the expected periodogram of a finite data record is a smeared version of the true PSD. It is clear that ideally  $W_B(\omega)$  should be a Dirac impulse. However, due to the finite length of the window, the main lobe of the Fejer kernel has a width that is approximately equal to  $2\pi/T$ . This indicates that if the data contain two sinusoids with a difference in their frequencies less than  $2\pi/T$ , they would not be *resolved*. Here, resolution is qualitatively defined as the ability of the spectrum estimator to discriminate two signals that in the frequency domain are close to each other. It is important to note that the resolution is inversely proportional to the length of observed data  $T$ .

**2.3.1.3 Blackman – Tukey Spectrum Estimator** It can be shown that the periodogram is identical to the correlogram from (2.9) where for the estimate of  $r[k]$  one uses (2.7). From (2.9), it is clear that the estimates of all the autocovariances are treated equally even though the autocovariances with smaller lags are estimated more accurately than the ones with larger lags. Blackman and Tukey proposed to account for this by weighting the autocovariance sequence with a window, which is a real sequence with the following properties [19]:

- Property 1 :  $0 \leq w[k] \leq w[0] = 1,$
- Property 2 :  $w[-k] = w[k],$  (2.17)
- Property 3 :  $w[k] = 0, \quad M < |k|, \quad M \leq T - 1.$

They defined the new spectrum by

$$\hat{p}_{\text{bt}}(\omega) = \sum_{k=-T+1}^{T-1} w[k] \hat{r}[k] e^{-j\omega k}. \quad (2.18)$$

Note that the symmetry of  $w[k]$  guarantees that the spectrum remains real.

The periodogram is thus a special case of the Blackman–Tukey spectrum estimator, when the window is rectangular. There are various choices of other windows including the Hanning, Hamming, Bartlett, and Blackman windows. The difference among them is in the width of the main lobe and the strength of the side lobes. Ideally, one would like to have a window with narrow main lobe and low side lobes. The rectangular window has the narrowest main lobe but the highest side lobes. It can be shown that the approximate width of its main lobe is  $2\pi/T$  and the level of its side lobe is  $-13$  dB. In comparison, for example, the Bartlett window has a main lobe width of about  $4\pi/T$ , thus twice as large, but a side lobe level of  $-25$  dB.

One problem of the periodogram is the high variance of the estimate. There are two classical approaches for reducing the variance of the periodogram. They are based on splitting the data sets into nonoverlapping or overlapping segments, finding the periodograms of each of the segments, and finally computing the average periodogram. The approach with nonoverlapping segments is known as the Bartlett method and the one with overlapping segments the Welch method [18]. The reduction of variance of these methods is paid by reduction in resolution since each periodogram is found from shorter segments.

**2.3.1.4 Capon's Method** We note here that the PSD estimation of  $\mathcal{Y}[t]$  is an ill-conditioned problem because we have to estimate the PSD at an infinite number of frequencies with a finite number of samples. To make the problem well posed, we could either assume that the PSD can be represented with finite set of parameters, which would lead to parametric PSD estimation (next subsection), or we could assume that the PSD  $p(\omega)$  is (almost) constant over a narrow bandwidth defined by  $[\omega - \eta\pi, \omega + \beta\pi]$ , where  $\eta << 1$ . The objective then is to estimate the PSD of the random process within these bandwidths. Methods that are based on estimating the PSD using this idea are known as *filter bank* methods. If we want to have a consistent PSD estimator, the number of different PSD estimates (bandwidths), has to be less than  $T$ , the length of the observed sequence. Since the number of bandwidths is  $1/\eta$ , we deduce that we must have  $\eta T > 1$ . It is clear that the smaller the value of  $\eta$ , the higher the resolution, but with the price of larger statistical variability and vice versa.

The periodogram can be interpreted as a filter bank method. Namely, if we use the notation

$$\mathbf{h}(\omega) = \frac{1}{T} \mathbf{e}(\omega), \quad (2.19)$$

where

$$\mathbf{e}(\omega) = [1 \quad e^{j\omega} \quad e^{j2\omega} \quad \dots \quad e^{j(T-1)\omega}]^T, \quad (2.20)$$

we can rewrite the periodogram as

$$\hat{p}_{\text{per}}(\omega) = T |\mathbf{h}^H(\omega) \mathbf{y}|^2, \quad (2.21)$$

where the superscript H denotes the Hermitian transpose of a matrix. The vector  $\mathbf{h}(\omega)$  can be considered a finite impulse response (FIR) of a filter centered at  $\omega$  and with a bandwidth approximately equal to  $2\pi/T$  radians per sampling interval.

A method proposed by Capon [6] is based on a bank of filters whose bandwidths are data dependent. If the impulse response of the filter centered at  $\omega_0$  is  $\mathbf{h}(\omega_0)$ , then one would minimize

$$\rho = \int_{-\pi}^{\pi} |H(\omega)|^2 p(\omega) d\omega \quad (2.22)$$

with the constraint

$$H(\omega_0) = 1,$$

where  $H(\omega)$  is the DTFT of  $\mathbf{h}(\omega)$ . The constraint implies that we want the filter to pass the desired signal at frequency  $\omega_0$  without distortion, while it minimizes the total output power, as specified by (2.22). This is a constrained minimization problem, and it can be shown that the solution for the FIR of the filter is [13]

$$\mathbf{h}(\omega_0) = \frac{\mathbf{R}^{-1} \mathbf{e}(\omega_0)}{\mathbf{e}^H(\omega_0) \mathbf{R}^{-1} \mathbf{e}(\omega_0)}. \quad (2.23)$$

The estimate of the PSD, which is also known as the minimum variance PSD estimate, is obtained from

$$\hat{p}_{mv}(\omega) = \frac{T}{\mathbf{e}^H(\omega) \hat{\mathbf{R}}^{-1} \mathbf{e}(\omega)}, \quad (2.24)$$

where

$$\hat{\mathbf{R}} = \begin{bmatrix} \hat{r}[0] & \hat{r}[1] & \cdots & \hat{r}[T-1] \\ \hat{r}^*[1] & \hat{r}[0] & \cdots & \hat{r}[T-2] \\ \vdots & \vdots & \vdots & \vdots \\ \hat{r}^*[T-1] & \hat{r}^*[T-2] & \cdots & \hat{r}[0] \end{bmatrix}. \quad (2.25)$$

**2.3.1.5 Multitaper Method** There is another filter bank method also known as multitaper (multiwindow) PSD estimator due to Thomson [20]. The idea behind the method is similar as before in that we want to design filter(s) that are most selective subject to suppressing signals outside the bandwidth of interest as much as possible. The resulting impulse responses are obtained by the use of discrete prolate spheroidal (Slepian) sequences, which for an observed vector of length  $T$  are defined as eigenvectors of a  $T \times T$  matrix  $\mathbf{C}$  whose elements are given by

$$c_{mn} = \frac{\sin(2\pi\eta(m-n))}{\pi(m-n)}, \quad m, n = 1, 2, \dots, T, \quad (2.26)$$

where  $\eta$ , as before, is the size of the filter bandwidth. The multitaper PSD estimate is obtained from

$$\hat{p}_{mw}(\omega) = \frac{1}{K} \sum_{k=1}^K \hat{p}_k(\omega), \quad (2.27)$$

where

$$\widehat{p}_k(\omega) = \frac{1}{\lambda_k} \left| \sum_{t=0}^{T-1} y[t] w_k[t] e^{-j\omega t} \right|^2 \quad (2.28)$$

with  $\lambda_k$  being the  $k$ th largest eigenvector (Slepian sequence) of  $\mathbf{C}$  and  $w_k[t]$  the eigenvector corresponding to  $\lambda_k$ . The number of windows  $K$  depends on the chosen filter bandwidth  $\eta$  and is given by  $\lfloor 2\eta T \rfloor$ , where  $\lfloor x \rfloor$  denotes the largest integer less than or equal to  $x$ . Note that the PSD estimate is obtained by averaging of  $K$  spectra, which allows for reduced variance of the estimate. For additional reading on the method, see [21–24].

### 2.3.2 Parametric Methods

The parametric methods for PSD estimation are based on the idea of describing the random process by mathematical models that depend on several parameters and that the PSD of the process can be expressed in terms of these parameters. Therefore, the basic procedure is first to estimate the parameters of the process and then to compute the PSD by using the estimated parameters. In practice, there are two big classes of parametric models:

1. Processes representing sinusoids in noise.
2. Processes with rational spectra.

Models of sinusoids in noise are written as

$$y[t] = \sum_{k=1}^K A_k e^{j\omega_k t} + \varepsilon[t], \quad (2.29)$$

where  $A_k$  and  $\omega_k$  are the complex amplitude and frequency of the  $k$ th sinusoid,  $K$  is the number of sinusoids, and  $\varepsilon[t]$  is the additive zero-mean noise that is independent from the sinusoids. Depending on the nature of the noise process, one can define problems with various levels of difficulty. Methods for estimating the PSD of the signal then simply focus on estimating the signal parameters,  $K$ ,  $A_k$ , and  $\omega_k$ , for  $k = 1, 2, \dots, K$ . If the estimates of the parameters are  $\widehat{A}_k$  and  $\widehat{\omega}_k$ , one possible estimate of the PSD of the signal is the line spectrum given by

$$\widehat{p}(\omega) = \sum_{k=1}^K |\widehat{A}_k|^2 \delta(\omega - \widehat{\omega}_k) + \widehat{p}_\varepsilon(\omega), \quad (2.30)$$

where  $\widehat{p}_\varepsilon(\omega)$  is the PSD of the noise process. The estimation of the number of sinusoids in the data is also known as a model order selection problem. For a recent review of the theory, see [25].

For estimating line spectra, Pisarenko proposed a pseudospectrum defined by [9]

$$\widehat{p}_{\text{pis}}(\omega) = \frac{1}{|\mathbf{e}^H(\omega) \boldsymbol{\xi}_{K+1}|^2}, \quad (2.31)$$

where

$$\mathbf{e}(\omega) = [1 e^{j\omega} \dots e^{j\omega K}]^T$$

and  $\xi_{K+1}$  is the eigenvector corresponding to the smallest eigenvalue of the  $(K + 1) \times (K + 1)$  autocorrelation matrix of the data. One can obtain the complete spectrum (with information about the actual powers of the sinusoids and the noise) with some additional straightforward computation.

One brief and insightful interpretation of the method is as follows. The eigenvector  $\xi_{K+1}$  spans the noise subspace. If the vector  $e(\omega)$  points in the direction of the signal, then it should be orthogonal to  $\xi_{K+1}$  because  $e(\omega)$  lies in the signal subspace. Since  $|e^H(\omega)\xi_{K+1}|^2$  is in the denominator of (2.31), it should produce a large value of the PSD whenever  $\omega$  is equal to a frequency of one of the sinusoids [ideally, the value of  $\hat{p}_{\text{pis}}(\omega)$  is infinity because then  $|e^H(\omega)\xi_{K+1}| = 0$ ].

Processes with rational spectra are in general modeled by autoregressive moving-average (ARMA) models defined by

$$y[t] = - \sum_{k=1}^n a_k y[t-k] + \sum_{k=0}^m b_k u[t-k], \quad (2.32)$$

where  $a_k$  and  $b_k$  are the autoregressive and moving-average parameters, respectively (with  $b_0 = 1$ ),  $m$  and  $n$  define the order of the ARMA process [denoted by ARMA  $(n, m)$ ], and  $u[t]$  is the driving noise of the process, which is assumed to be zero mean and wide-sense stationary with variance  $\sigma^2$ . The theoretical PSD of the process is given by

$$p(\omega) = \sigma^2 \left| \frac{1 + b_1 e^{-j\omega} + b_2 e^{-j2\omega} + \cdots + b_m e^{-jm\omega}}{1 + a_1 e^{-j\omega} + a_2 e^{-j2\omega} + \cdots + a_n e^{-jn\omega}} \right|^2. \quad (2.33)$$

The parametric procedure for estimating the PSD amounts to estimating the unknown parameters of the model, that is, the coefficients  $a_k$ , where  $k = 1, 2, \dots, n$ , the coefficients  $b_k$ , where  $k = 1, 2, \dots, m$ , and the variance  $\sigma^2$ . Once the estimates are found, the power spectral density is computed by (2.33), where for the true parameter values we substitute the estimated ones. There is a wide variety of methods for estimating the unknown parameters (see [13, 18]).

If all the  $b_k$ 's in (2.32) are set to zero, we have the random process

$$y[t] = - \sum_{k=1}^n a_k y[t-k] + u[t], \quad (2.34)$$

which is known as the autoregressive (AR) random process. Its PSD is given by

$$p(\omega) = \sigma^2 \left| \frac{1}{1 + a_1 e^{-j\omega} + a_2 e^{-j2\omega} + \cdots + a_n e^{-jn\omega}} \right|^2. \quad (2.35)$$

Clearly, the AR process is a special case of the ARMA process.

From the ARMA process one can also define a process where all the AR coefficients are set to zero, that is,

$$y[t] = u[t] + b_1 u[t-1] + \cdots + b_m u[t-m], \quad (2.36)$$

which is known as the moving-average (MA) process. The PSD of the MA process is

$$p(\omega) = \sigma^2 |1 + b_1 e^{-j\omega} + b_2 e^{-j2\omega} + \cdots + b_m e^{-jm\omega}|^2. \quad (2.37)$$

Besides estimating the AR and MA parameters of these models, a typical problem is the determination of the model orders  $m$  and/or  $n$ . As in the case of sinusoids, this is also a model order selection problem. Well-known procedures for estimating  $m$  and  $n$  are the AIC criterion due to Akaike [26] and the minimum description length (MDL) proposed by Rissanen [27].

## 2.4 SPATIAL SPECTRUM ESTIMATION

We proceed by first defining the problem of spatial spectrum estimation and, in particular, the model of the data and the made assumptions. The methods for processing spatial data can also be classified as nonparametric and parametric methods. As in Section 2.3, we first describe the nonparametric methods (Section 2.4.2) and then the parametric ones (Section 2.4.3). We also briefly address the problem of determining the number of signals impinging on the array (Section 2.4.4).

### 2.4.1 Model

The objective of spatial spectrum estimation is determining the distribution of the PSD of signals as a function of space, where the signals are received by an array of  $L$  sensors. The received signals are sampled in time, and since they are also received by several sensors, we consider that they are also sampled in space (cf. Fig. 2.1). From the spatial PSD one may be able to infer the number and the location of radiating sources of the received signals.

We now establish the mathematical model of the received measurements. First, we assume that the sources are point emitters and that they are far away from the sensors. In addition, we assume that the sources and the sensors are in the same plane and that the propagation medium is homogeneous.<sup>4</sup> The signal on the surface of a sphere centered at the emitter's location has a common phase and is called a *wavefront*. If the radius of the sphere is very large compared to the size of the array, the received signals can be approximated as planar signals. If the transmitted signal is narrowband with a carrier frequency  $\omega_c$ ,<sup>5</sup> the  $l$ th sensor receives a signal that can be expressed as

$$y_l[t] = s[t]e^{j\omega_c(t-\tau_l)}, \quad (2.38)$$

where  $s[t]$  is a slowly varying complex signal that modulates the carrier, and  $\tau_l$  is the delay of the received signal with respect to a reference time instant. For simplified presentation, we remove the carrier term in (2.38) and write

$$y_l[t] = s[t]e^{-j\omega_c\tau_l}. \quad (2.39)$$

A critical parameter in this representation is the delay  $\tau_l$ , which reveals the “location” of the source, that is, the DOA of the signal. We point out that if the sources are near field or we need to specify the exact location of the sources, we need three parameters for each source, its azimuth, elevation, and range.

<sup>4</sup>For a good introduction to elementary plane waves and their parameters of interest, homogeneous wavefields, and related signal processing, see [14, 28].

<sup>5</sup>A narrowband assumption means that the bandwidth of the signal is only a small fraction of the carrier frequency, which is typical for radar systems. In sonar systems this usually is not the case.

The array is composed of  $L$  sensors. Therefore, we arrange the received signals at time instant  $t$  as a snapshot defined by

$$\mathbf{y}[t] = [y_1[t] \ y_2[t] \ \cdots \ y_L[t]]^T, \quad (2.40)$$

where

$$\mathbf{y}[t] = \mathbf{a}(\tau)s[t], \quad (2.41)$$

and where

$$\mathbf{a}(\tau) = [e^{-jw_c\tau_1} e^{-jw_c\tau_2} \cdots e^{-jw_c\tau_L}]^T. \quad (2.42)$$

The vector  $\tau$  is defined by

$$\tau = [\tau_1 \tau_2 \cdots \tau_L]^T, \quad (2.43)$$

where  $\tau_l$  is the delay of the received signal at the  $l$ th sensor. Without loss of generality, as a reference point for measuring the time delay, we use the time of signal arrival at the first sensor. In that case, we can write

$$\mathbf{a}(\tau) = [1 \ e^{-jw_c\tau_2} \cdots \ e^{-jw_c\tau_L}]^T. \quad (2.44)$$

In the sequel, we work with uniform linear arrays, like the one from Figure 2.3. There,  $L$  identical sensors are deployed on a line and are separated by  $d$  meters. The signal DOA  $\theta$  is defined as the angle between the direction of propagation and the line that is normal to the line of sensor deployment, and  $\theta \in [-90^\circ, 90^\circ]$  (see Fig. 2.3). With the assumption that the waves are planar, we can easily deduce that

$$\tau_l = \frac{d(l-1)}{c} \sin \theta, \quad 1 < l \leq L, \quad (2.45)$$

where  $c$  represents the velocity of wave propagation. With this array geometry, we perform uniform spatial sampling of the signal. It is important to note that we avoid aliasing in the spatial direction of the signal spectrum if

$$d < \frac{\tilde{\lambda}}{2}, \quad (2.46)$$

where  $\tilde{\lambda}$  is the wavelength of the signal. This condition can readily be derived by using the analogy with sampled continuous-time signals [13].

We rewrite (2.44) as

$$\mathbf{a}(\theta) = [1 \ e^{-jw_c d \sin \theta / c} \cdots \ e^{-j(L-1)w_c d \sin \theta / c}]^T, \quad (2.47)$$

where  $\theta \in [-90^\circ, 90^\circ]$ , and we refer to it as steering vector. To simplify the notation furthermore, we define the spatial frequency

$$w_s = \frac{w_c d \sin \theta}{c} \quad (2.48)$$

and (2.47) becomes

$$\mathbf{a}(\theta) = [1 \ e^{-jw_s} e^{-j2w_s} \cdots \ e^{-j(L-1)w_s}]^T, \quad (2.49)$$

where in the notation of the steering vector we kept the argument  $\theta$  (which defines the spatial frequency  $\omega_s$ ).

We now rewrite (2.41)

$$\mathbf{y}[t] = \mathbf{a}(\theta)s[t] + \boldsymbol{\epsilon}[t], \quad (2.50)$$

where on the right-hand side we have included an additive noise vector  $\boldsymbol{\epsilon}[t]$  to reflect actual noise and errors due to inaccuracies of the model. It is striking here that this model is identical to the one of representing a complex sinusoid in noise.

The generalization of (2.50) to the case when the array receives  $K$  signals is straightforward. Then we have

$$\mathbf{y}[t] = \mathbf{A}(\theta)s[t] + \boldsymbol{\epsilon}[t], \quad (2.51)$$

where  $\mathbf{y}[t]$  and  $\boldsymbol{\epsilon}[t]$  are  $L \times 1$  vectors with the same meaning as before,  $\mathbf{A}(\theta)$  is an  $L \times K$  matrix whose columns are the steering vectors of the array, that is,

$$\mathbf{A}(\theta) = [\mathbf{a}(\theta_1) \ \mathbf{a}(\theta_2) \cdots \mathbf{a}(\theta_K)], \quad (2.52)$$

where  $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \cdots \ \theta_K]^T$  is the vector of DOAs, and  $s[t] = [s_1[t] \ s_2[t] \ \cdots \ s_K[t]]^T$  is the vector of signals, where  $s_1[t]$  is the signal from the first source,  $s_2[t]$  the signal from the second source, and so on. We make the assumption that the number of sensors in the array is larger than the number of impinging signals, that is,  $K < L$ .

Finally, as already pointed out, we may receive more than one snapshot of data. If that is the case, the snapshots can be organized in a matrix  $\mathbf{Y}$ , which we rewrite here for convenience, as

$$\mathbf{Y} = \begin{bmatrix} y_1[0] & y_1[1] & y_1[2] & \cdots & y_1[T-1] \\ y_2[0] & y_2[1] & y_2[2] & \cdots & y_2[T-1] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_L[0] & y_L[1] & y_L[2] & \cdots & y_L[T-1] \end{bmatrix}. \quad (2.53)$$

We recall that the columns of  $\mathbf{Y}$ ,  $\mathbf{y}[t] = [y_1[t] \ y_2[t] \ \cdots \ y_L[t]]^T$  are the samples of the  $L$  sensors taken at time instants  $t = 0, 1, \dots, T-1$ .

We make some additional assumptions. One of them is related to the noise vector  $\boldsymbol{\epsilon}[t]$ . We assume that the noise is spatially white and with covariance matrix given by

$$\begin{aligned} \mathbf{C}_\epsilon &= E(\boldsymbol{\epsilon}[t]\boldsymbol{\epsilon}^H[t]) \\ &= \sigma_\epsilon^2 \mathbf{I}. \end{aligned} \quad (2.54)$$

For the signal, we assume that it is zero mean and has a covariance matrix

$$\mathbf{C}_s = E(s[t]s^H[t]), \quad (2.55)$$

which is nonsingular and in general nondiagonal. If  $\mathbf{C}_s$  is singular, then (some) signals are fully correlated, and are therefore referred to as being coherent. The latter scenario arises when, for example, one signal is a delayed and scaled replica of another signal.

We reiterate that the general goal of spatial spectrum estimation is to obtain the distribution of the PSD of  $\mathbf{Y}$  as a function of space. More specifically, the objective may be formulated as one of estimating the DOAs of the signals and possibly of determining the number of impinging signals  $K$  (if that number is unknown).

## 2.4.2 Nonparametric Methods

Here we address two nonparametric methods, the classical and Capon's beamforming. Their advantage over parametric methods is that they are derived under minimal assumptions about the statistics of the data.

**2.4.2.1 Classical Beamforming** The idea of estimating DOA by beamforming is based on forcing the array to “look” in one direction at a time and to compute the power that it sees. In other words, a beamformer combines the received signals of the sensors in a way that passes a signal from a given direction undistorted and suppresses signals from other directions. This is accomplished by amplifying the signals of the various receivers by different weights. One can interpret the operation of beamforming as spatial filtering [16].

One approach to beamforming is based on making the output of the beamformer a linear combination of the data of the  $L$  sensors, that is,

$$\begin{aligned} y_h[t] &= \sum_{l=1}^L h_l^* y_l[t] \\ &= \mathbf{h}^H \mathbf{y}[t], \end{aligned} \quad (2.56)$$

where  $y_h[t]$  is the output of the beamformer,  $\mathbf{h} = [h_1 \ h_2 \ \dots \ h_L]^T$  is a vector of weight coefficients that need to be determined, and  $\mathbf{y}[t]$ , as before, is an  $L \times 1$  vector of samples taken by the  $L$  sensors at time  $t$ , that is,  $\mathbf{y}[t] = [y_1[t] \ y_2[t] \ \dots \ y_L[t]]^T$ .

Let the array data be modeled as

$$\mathbf{y}[t] = \mathbf{a}(\theta)s[t] + \boldsymbol{\varepsilon}[t], \quad (2.57)$$

where  $\mathbf{a}(\theta)$  is given by (2.49),<sup>6</sup> and where  $\boldsymbol{\varepsilon}[t]$  is spatially white noise, with covariance matrix  $\sigma_\varepsilon^2 \mathbf{I}$ .

We would like to obtain  $\mathbf{h}$  so that it satisfies two conditions:

1. It passes the signal from DOA  $\theta$  without distortion.
2. It minimizes the total output power, thereby minimizing all the signals coming from different DOAs from  $\theta$ .

We express the first condition as

$$\mathbf{h}^H \mathbf{a}(\theta) = 1 \quad (2.58)$$

and it represents the optimization constraint.

Now we find the mathematical expression of the second condition. The output power of the beamformer is given by

$$\rho_h = E(|y_h[t]|^2). \quad (2.59)$$

More specifically, we have

$$\begin{aligned} \rho_h &= E(\mathbf{h}^H \mathbf{y}[t] \mathbf{y}^H[t] \mathbf{h}) \\ &= |s[t]|^2 + \mathbf{h}^H E(\boldsymbol{\varepsilon}[t] \boldsymbol{\varepsilon}^H[t]) \mathbf{h}[t] \\ &= |s[t]|^2 + \sigma_\varepsilon^2 \mathbf{h}^H \mathbf{h}, \end{aligned} \quad (2.60)$$

<sup>6</sup>The derivation is analogous for a more general array geometry.

where in deriving (2.60) we used the constraint (2.58). We want to minimize  $\rho_h$  given the constraint, and therefore, the optimal  $\mathbf{h}$  is obtained from

$$\mathbf{h}_o = \arg \min_{\mathbf{h}} \{\mathbf{h}^H \mathbf{h}\} \quad \text{s.t.} \quad \mathbf{h}^H \mathbf{a}(\theta) = 1, \quad (2.61)$$

where s.t. means such that. It is straightforward to show that

$$\begin{aligned} \mathbf{h}_o &= \frac{\mathbf{a}(\theta)}{\mathbf{a}^H(\theta) \mathbf{a}(\theta)} \\ &= \frac{\mathbf{a}(\theta)}{L}. \end{aligned} \quad (2.62)$$

In the literature, this beamformer is known as the Bartlett beamformer. Its expected power output is

$$\begin{aligned} \rho_h &= E(\mathbf{h}^H \mathbf{y}[t] \mathbf{y}^H[t] \mathbf{h}) \\ &= \frac{1}{L^2} \mathbf{a}^H(\theta) E(\mathbf{y}[t] \mathbf{y}^H[t]) \mathbf{a}(\theta) \end{aligned} \quad (2.63)$$

$$= \frac{1}{L^2} \mathbf{a}^H(\theta) \mathbf{C}_y \mathbf{a}(\theta), \quad (2.64)$$

where  $\mathbf{C}_y$  is the covariance matrix of the data.

For the PSD of the data we write

$$p_b(\theta) = \frac{1}{L} \mathbf{a}^H(\theta) \mathbf{C}_y \mathbf{a}(\theta). \quad (2.65)$$

The matrix  $\mathbf{C}_y$  is a priori unknown, and we estimate it from the data. If the array has collected  $T$  snapshots, the covariance matrix is estimated according to

$$\widehat{\mathbf{C}}_y = \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{y}[t] \mathbf{y}^H[t]. \quad (2.66)$$

Finally, we write for the estimate of the PSD

$$\widehat{p}_b(\theta) = \frac{1}{L} \mathbf{a}^H(\theta) \widehat{\mathbf{C}}_y \mathbf{a}(\theta) \quad (2.67)$$

or

$$\widehat{p}_b(\theta) = \frac{1}{T} \sum_{t=0}^{T-1} \frac{|\mathbf{a}(\theta)^H \mathbf{y}[t]|^2}{L}. \quad (2.68)$$

Note that we choose as an estimate of  $\theta$  the value of  $\theta$  that maximizes  $\widehat{p}_b(\theta)$ . Also, we can interpret (2.68) as the average (over time) spatial PSD of the data. The spectra from the snapshots are basically periodograms.

If there are two or more signals, one would look for two or more peaks in the spatial spectrum. Clearly, the method has the same limitations as the periodogram in terms of resolution. In other words, if two sources are spatially separated by angles that correspond to spatial frequency separation of  $2\pi/L$  or less, this approach will not resolve them as two sources.

**2.4.2.2 Capon's Method** Capon's approach to designing is similar to the classical one with one important difference. Namely, in the Bartlett beamformer no attempt is made to be “optimally” selective while passing a set of specific input data. The optimization was carried out if there was only one impinging signal in the data. However, one may use properties of the input signals and define a bank of filters that are data dependent and are optimized so that they minimize the filters' responses outside the bandwidth of interest. In other words, we pass the desired signal with a given  $\theta$  undistorted (as before) but attenuate the ones with different angles as much as possible.

In mathematical terms, we again have the constraint (2.58). Now we do not make the assumption for one signal present in the data, but instead simply write for the output power of the beamformer

$$\begin{aligned}\rho_h &= E(\mathbf{h}^H \mathbf{y}[t] \mathbf{y}^H[t] \mathbf{h}) \\ &= \mathbf{h}^H \mathbf{C}_y \mathbf{h},\end{aligned}\quad (2.69)$$

where  $\mathbf{C}_y$  is the covariance matrix of the data. We want to minimize the output power  $\rho_h$  in (2.69) given the constraint, that is, we have

$$\mathbf{h}_o = \arg \min_{\mathbf{h}} \{\mathbf{h}^H \mathbf{C}_y \mathbf{h}\} \quad \text{s.t.} \quad \mathbf{h}^H \mathbf{a}(\theta) = 1. \quad (2.70)$$

The solution, again, is straightforward to obtain. It is given by

$$\mathbf{h}_o = \frac{\mathbf{C}_y^{-1} \mathbf{a}(\theta)}{\mathbf{a}^H(\theta) \mathbf{C}_y^{-1} \mathbf{a}(\theta)}. \quad (2.71)$$

When plugged-in in (2.69), we obtain that the total power in a given direction  $\theta$  is found from

$$\rho_h(\theta) = \frac{1}{\mathbf{a}^H(\theta) \mathbf{C}_y^{-1} \mathbf{a}(\theta)}. \quad (2.72)$$

Since  $\mathbf{C}_y$  is not known, it has to be estimated from the data. Thus, the final expression for the PSD of the data is

$$\hat{p}_{mv}(\theta) = \frac{L}{\mathbf{a}^H(\theta) \hat{\mathbf{C}}_y^{-1} \mathbf{a}(\theta)}. \quad (2.73)$$

This expression should be compared with (2.24), which is Capon' PSD estimator for temporal data.

Recall that Capon's method is also known as the minimum variance spectrum estimation method [6]. It has been generalized in [29].

**2.4.2.3 Multitaper Method** The method described in Section 2.3.1.5 can also be applied for spatial spectrum estimation [30]. The idea is the same, except that here we have to modify the steering vectors  $\mathbf{a}(\theta)$  by using the various windows (Slepian sequences). More specifically, we construct the PSD estimate according to

$$\hat{p}_{MW}(\theta) = \frac{1}{K} \sum_{k=1}^K \hat{p}_k(\theta), \quad (2.74)$$

where the terms of the summation are defined by

$$\hat{p}_k(\theta) = \frac{1}{T \lambda_k} \sum_{t=0}^{T-1} \frac{|\tilde{\mathbf{a}}_k(\theta)^H \mathbf{y}[t]|^2}{L}. \quad (2.75)$$

The symbol  $\tilde{\mathbf{a}}_k(\theta)$  is given by

$$\tilde{\mathbf{a}}_k(\theta) = \boldsymbol{\omega}_k \odot \mathbf{a}(\theta) \quad (2.76)$$

where  $\odot$  denotes element-wise multiplication (a Hadamard product of two vectors), and  $\boldsymbol{\omega}_k$  is the  $k$ th window as defined by the eigenvector corresponding to the  $k$ th largest eigenvalue  $\lambda_k$  of the  $L \times L$  matrix with elements given by (2.26). As before, the number of windows  $K$  depends on the chosen filter bandwidth  $\eta$ . It is interesting that this method can be combined with the minimum variance method of Capon [31].

**2.4.2.4 Discussion** The above beamforming methods are conventional. They are also called fixed because once the beamformer is determined, it does not change with time. In that case, one only uses the information about the location of the sensors (which is fixed) and the signal DOAs of interest. In some applications, we can improve on beamforming by allowing for dynamic changes of the beamformer with time. This is usually referred to as adaptive beamforming. For optimization of the reception of the signal of interest or the rejection of interference, in adaptive beamforming we use additional information (that varies with time). For the adaptation, one uses a specific criterion such as the minimization of the total noise output.

In adaptive beamforming, it is of essence to produce the weights of the beamformer in real time. Radar-phased arrays have high data rates, which creates a challenge for implementation of such beamformers. However, this challenge has been met by the use of field-programmable gate arrays.

### 2.4.3 Parametric Methods

The parametric methods exploit the mathematical model

$$\mathbf{y}[t] = \mathbf{A}(\theta)\mathbf{s}[t] + \boldsymbol{\epsilon}[t], \quad t = 0, 1, \dots, T - 1, \quad (2.77)$$

where  $\mathbf{A}(\theta)$  is defined by (2.52) and  $\boldsymbol{\epsilon}[t]$  is assumed to be spatially white, that is,  $E(\boldsymbol{\epsilon}[t]\boldsymbol{\epsilon}^H[t]) = \sigma_\epsilon^2 \mathbf{I}$ . The unknowns of interest are the DOAs  $\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \dots \ \theta_K]^T$ , and the remaining unknowns,  $\mathbf{s}[t]$ ,  $t = 0, 1, \dots, T - 1$ , and  $\sigma_\epsilon^2$ , are assumed to be nuisance parameters. We already mentioned that some of the signals in (2.77) may be coherent. Parametric methods can deal with this kind of adversity relatively efficiently.

First, we describe the least-squares method and proceed with subspace-based methods, including MUSIC and ESPRIT. For all of these methods, we assume that we know the number of signals  $K$ . When this information is not available, we can apply some of the methods described in Section 2.4.4.

**2.4.3.1 Least-Squares Method for Deterministic Model of Signals** One approach to finding the unknown DOAs is by using the least-squares criterion and obtaining them from

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \left\{ \sum_{t=0}^{T-1} |\mathbf{y}[t] - \mathbf{A}\mathbf{s}[t]|^2 \right\}, \quad (2.78)$$

where in the notation, for convenience, we dropped the dependence of  $\mathbf{A}$  on  $\boldsymbol{\theta}$ . We emphasize that the signals  $\mathbf{s}[t]$  are assumed deterministic. Here we have two sets of

unknowns, linear ( $s[t]$ ) and nonlinear ( $\theta$ ). It is well known that given  $\theta$ , the minimization with respect to  $s[t]$  yields the solution

$$\hat{s}[t] = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{y}[t]. \quad (2.79)$$

Upon substituting it in (2.78), we obtain

$$\begin{aligned} \hat{\theta} &= \arg \min_{\theta} \left\{ \sum_{t=0}^{T-1} |\mathbf{y}[t] - \mathbf{A}(\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{y}[t]|^2 \right\} \\ &= \arg \min_{\theta} \left\{ \sum_{t=0}^{T-1} \mathbf{y}^H[t] (\mathbf{I} - \mathbf{A}(\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H) \mathbf{y}[t] \right\} \\ &= \arg \min_{\theta} \left\{ \sum_{t=0}^{T-1} \mathbf{y}^H[t] \mathbf{P}^\perp \mathbf{y}[t] \right\}, \end{aligned} \quad (2.80)$$

where  $\mathbf{P}^\perp$  is a projection matrix defined by

$$\begin{aligned} \mathbf{P}^\perp &= \mathbf{I} - \mathbf{A}(\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \\ &= \mathbf{I} - \mathbf{P} \end{aligned} \quad (2.81)$$

and which projects  $\mathbf{y}[t]$  onto the “noise subspace,” that is, the subspace that is orthogonal to the signal subspace spanned by the columns of  $\mathbf{A}$ . In (2.81) the matrix  $\mathbf{P}$ , too, is a projection matrix and is defined by

$$\mathbf{P} = \mathbf{A}(\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H. \quad (2.82)$$

It projects  $\mathbf{y}[t]$  onto the signal subspace, which is spanned by the columns of  $\mathbf{A}$ . We can rewrite (2.80) as follows:

$$\begin{aligned} \hat{\theta} &= \arg \min_{\theta} \left\{ \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{y}^H[t] \mathbf{P}^\perp \mathbf{y}[t] \right\} \\ &= \arg \min_{\theta} \left\{ \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{y}^H[t] \mathbf{y}[t] - \mathbf{y}^H[t] \mathbf{A}(\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{y}[t] \right\} \\ &= \arg \max_{\theta} \{ \text{tr}(\mathbf{P} \widehat{\mathbf{C}}_y) \}, \end{aligned} \quad (2.83)$$

where  $\text{tr}(\cdot)$  stands for trace of the matrix in the argument of the operator.

The maximization of (2.83) is a nonlinear problem, and therefore, in finding  $\hat{\theta}$ , we usually apply iterative procedures of Gauss–Newton type [32]. It should be noted that the function to be maximized in (2.83) is multimodal with a sharp global maximum. This entails that we need a very good initialization of the method; otherwise, the global maximum may easily be missed (see, e.g., [33]). It is not difficult to see that in the case of one signal in the data, the method is equivalent to Bartlett’s beamformer (2.68).

If the noise vectors  $\mathbf{e}[t]$  are assumed independent and identically distributed (i.i.d.) according to a complex Gaussian distribution (spatially white and circularly symmetric), the criterion (2.83) yields the maximum likelihood (ML) solution. We

also point out that the least-squares criterion provides a good estimate of the DOAs even if the assumptions about the noise being white are not correct.

In [34], a method for ML DOA estimation is presented that is based on alternating projections. The multidimensional maximization of the likelihood is implemented as a sequence of one-dimensional maximizations.

**2.4.3.2 Stochastic Maximum-Likelihood Method** In the previous section, we assumed that the signals are unknown and deterministic. If the signals  $s[t]$  are stochastic, zero mean, and Gaussian as well as independent from the Gaussian noise  $\epsilon[t]$ , then the data  $y[t]$  are also zero mean and Gaussian with a covariance matrix given by

$$\begin{aligned} \mathbf{C}_y &= E(y[t]y^H[t]) \\ &= E((As[t] + \epsilon[t])(As[t] + \epsilon[t])^H) \\ &= \mathbf{A}\mathbf{C}_s\mathbf{A}^H + \sigma_\epsilon^2 \mathbf{I}, \end{aligned} \quad (2.84)$$

where  $\mathbf{C}_s$  is the covariance matrix of the signal  $s[t]$ , and  $\sigma_\epsilon^2 \mathbf{I}$  is the noise covariance matrix. If the signals are not coherent, the matrix  $\mathbf{C}_s$  is nonsingular; otherwise, it is a rank-deficient matrix. Here we also assume

$$E(s[t_1]s^H[t_2]) = \mathbf{C}_s \delta[t_1 - t_2], \quad (2.85)$$

$$E(s[t_1]s^T[t_2]) = \mathbf{0}, \quad (2.86)$$

where  $\delta(\cdot)$  is the Kronecker delta function.

The likelihood function is given by

$$f(\mathbf{Y}; \boldsymbol{\theta}, \mathbf{C}_s, \sigma_\epsilon^2) = \prod_{t=0}^{T-1} \frac{1}{\pi^L |\mathbf{C}_y|} e^{-\mathbf{y}^H[t] \mathbf{C}_y^{-1} \mathbf{y}[t]}, \quad (2.87)$$

where  $\mathbf{C}_y$  via (2.84) contains information about the unknowns  $\boldsymbol{\theta}$ ,  $\mathbf{C}_s$ , and  $\sigma_\epsilon^2$ . When there are  $K$  signals in the data and  $\mathbf{C}_s$  is of full rank, of the total number of unknown parameters only  $K$  are of primary interest and the rest are nuisance parameters.

Under the above assumptions, it can be shown that the ML estimates of the DOAs are obtained from

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \left\{ \log |\mathbf{A}\hat{\mathbf{S}}\mathbf{A}^H + \sigma_\epsilon^2(\boldsymbol{\theta})\mathbf{I}| \right\}, \quad (2.88)$$

where

$$\hat{\mathbf{S}} = \mathbf{A}^\dagger (\hat{\mathbf{C}}_y - \sigma_\epsilon^2(\boldsymbol{\theta})\mathbf{I}) \mathbf{A}^H \quad (2.89)$$

and

$$\sigma_\epsilon^2(\boldsymbol{\theta}) = \frac{1}{L-K} \text{tr}(\mathbf{P}^\perp). \quad (2.90)$$

In (2.88),  $\mathbf{A}^\dagger$  denotes pseudoinverse, which is defined by

$$\mathbf{A}^\dagger = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H, \quad (2.91)$$

where  $\mathbf{P}^\perp$  is given by (2.81). This, again, is clearly a nonlinear optimization problem that can be addressed similarly like the nonlinear least-squares problem from the previous section [35].

**2.4.3.3 Subspace Fitting Method** Another method for finding the DOAs is based on the decomposition of the covariance matrix  $\mathbf{C}_y$  according to

$$\mathbf{C}_y = \sum_{i=1}^L \lambda_i \boldsymbol{\xi}_i \boldsymbol{\xi}_i^H, \quad (2.92)$$

where  $\lambda_i$  and  $\boldsymbol{\xi}_i$ ,  $i = 1, 2, \dots, L$  are the eigenvalues and eigenvectors of  $\mathbf{C}_y$ , respectively. If we rank all the eigenvalues of  $\mathbf{C}_y$ , we can write theoretically  $\lambda_1 \geq \lambda_2 \geq \lambda_K \geq \lambda_{K+1} = \lambda_{K+2} = \dots = \lambda_L = \sigma_\varepsilon^2$ . Thus, in theory the  $L - K$  smallest eigenvalues are identical and equal to the noise variance  $\sigma_\varepsilon^2$ . Furthermore, it is convenient to express  $\mathbf{C}_y$  as

$$\mathbf{C}_y = \boldsymbol{\Xi}_s \Lambda_s \boldsymbol{\Xi}_s^H + \boldsymbol{\Xi}_\varepsilon \Lambda_\varepsilon \boldsymbol{\Xi}_\varepsilon^H, \quad (2.93)$$

where  $\boldsymbol{\Xi}_s$  is an  $L \times K$  matrix whose columns are the eigenvectors corresponding to the  $K$  largest eigenvalues and  $\Lambda_s$  is a  $K \times K$  diagonal matrix whose diagonal elements are the  $K$  largest eigenvalues. Similarly,  $\boldsymbol{\Xi}_\varepsilon$  is an  $L \times (L - K)$  matrix whose columns are the eigenvectors corresponding to the  $L - K$  smallest eigenvalues and  $\Lambda_\varepsilon$  is an  $(L - K) \times (L - K)$  diagonal matrix whose diagonal elements are the  $L - K$  smallest eigenvalues. We also say that the columns of  $\boldsymbol{\Xi}_s$  span the signal subspace and those of  $\boldsymbol{\Xi}_\varepsilon$  the noise subspace.

Recall that the covariance matrix  $\mathbf{C}_y$  is given by

$$\mathbf{C}_y = \mathbf{A} \mathbf{C}_s \mathbf{A}^H + \sigma_\varepsilon^2 \mathbf{I}, \quad (2.94)$$

where the rank of  $\mathbf{C}_s$  is assumed to be  $K$ . We also have the identity

$$\mathbf{I} = \boldsymbol{\Xi}_s \boldsymbol{\Xi}_s^H + \boldsymbol{\Xi}_\varepsilon \boldsymbol{\Xi}_\varepsilon^H. \quad (2.95)$$

Then from (2.93)–(2.95), we can write

$$\mathbf{A} \mathbf{C}_s \mathbf{A}^H + \sigma_\varepsilon^2 \boldsymbol{\Xi}_s \boldsymbol{\Xi}_s^H = \boldsymbol{\Xi}_s \Lambda_s \boldsymbol{\Xi}_s^H. \quad (2.96)$$

Next we postmultiply both sides with  $\boldsymbol{\Xi}_s$  and obtain

$$\boldsymbol{\Xi}_s = \mathbf{A} \mathbf{T}, \quad (2.97)$$

where

$$\mathbf{T} = \mathbf{C}_s \mathbf{A}^H \boldsymbol{\Xi}_s (\Lambda_s - \sigma_\varepsilon^2 \mathbf{I})^{-1}. \quad (2.98)$$

Both  $\mathbf{A}$  and  $\mathbf{T}$  depend on  $\boldsymbol{\theta}$  and their product yields  $\boldsymbol{\Xi}_s$  [as per (2.97)]. The idea now is to estimate  $\boldsymbol{\theta}$  by minimizing the Frobenius norm

$$[\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{T}}] = \arg \min_{\boldsymbol{\theta}, \mathbf{T}} \|\boldsymbol{\Xi}_s - \mathbf{A} \mathbf{T}\|_F^2. \quad (2.99)$$

Note that we do not know the matrix  $\boldsymbol{\Xi}_s$  but, instead, we estimate it from the data snapshots. Since, the quality of the estimated  $\boldsymbol{\Xi}_s$  depends on the signal-to-noise ratios

(SNRs), one introduces a weighting matrix  $\mathbf{W}$  to maximize for the accuracy of the estimated parameters. Then the criterion becomes

$$[\hat{\boldsymbol{\theta}}, \hat{\mathbf{T}}] = \arg \min_{\boldsymbol{\theta}, \mathbf{T}} \|\hat{\boldsymbol{\Xi}}_s \mathbf{W}^{1/2} - \mathbf{A}\mathbf{T}\|_{\text{F}}^2. \quad (2.100)$$

One can show that the DOAs are optimally estimated from

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \text{tr} \left( \mathbf{P}^\perp \hat{\boldsymbol{\Xi}}_s \mathbf{W} \hat{\boldsymbol{\Xi}}_s^H \right), \quad (2.101)$$

where  $\mathbf{P}^\perp$  is given by (2.81), and

$$\mathbf{W} = (\hat{\boldsymbol{\Xi}}_s - \hat{\sigma}_\varepsilon^2 \mathbf{I}) \hat{\boldsymbol{\Xi}}_s^{-1} \quad (2.102)$$

with  $\hat{\sigma}_\varepsilon^2$  being any consistent estimate of the noise variance [36].

The above method is more accurately called the signal subspace fitting method. One can formulate the above idea into a method based on noise subspace fitting [35].

**2.4.3.4 MUSIC** Another subspace-based method is known under the name MUSIC, which stands for multiple signal classification [11, 37]. The method is derived by enforcing the solution for the signal to be orthogonal to the noise subspace. Namely, we express the covariance matrix  $\mathbf{C}_y$  again in terms of its spectral decomposition,

$$\mathbf{C}_y = \boldsymbol{\Xi}_s \boldsymbol{\Lambda}_s \boldsymbol{\Xi}_s^H + \boldsymbol{\Xi}_\varepsilon \boldsymbol{\Lambda}_\varepsilon \boldsymbol{\Xi}_\varepsilon^H, \quad (2.103)$$

where the  $L - K$  columns of  $\boldsymbol{\Xi}_\varepsilon$  are eigenvectors of  $\mathbf{C}_y$  spanning the noise subspace. It should be noted that the range space of  $\boldsymbol{\Xi}_s$  is the same as that of  $\mathbf{A}$ . An important result in deriving MUSIC, which is not hard to prove, is

$$\mathbf{A}^H \boldsymbol{\Xi}_\varepsilon = \mathbf{0}. \quad (2.104)$$

The expression states that the columns of the matrix  $\mathbf{A}$ , as defined by (2.52) and that span the signal subspace, are orthogonal to the noise subspace. One can then show that the signal DOAs are the only solution to the equation

$$\mathbf{a}^H(\theta) \boldsymbol{\Xi}_\varepsilon \boldsymbol{\Xi}_s^H \mathbf{a}(\theta) = 0 \quad (2.105)$$

when  $L > K$ .

The MUSIC algorithm basically exploits (2.105). The idea is to find the set of signals that are orthogonal to the noise subspace defined by the columns of  $\boldsymbol{\Xi}_\varepsilon$ . Note that in order to determine the noise space, we need to know its dimension, which is equivalent to specifying the number of signals in the data. MUSIC is implemented as follows:

1. Compute the covariance matrix of  $\mathbf{y}[t]$  by

$$\hat{\mathbf{C}}_y = \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{y}[t] \mathbf{y}^H[t]. \quad (2.106)$$

2. Find the eigenvectors and eigenvalues of  $\widehat{\mathbf{C}}_y$  and construct  $\widehat{\mathbf{\Xi}}_\varepsilon$ .
3. Compute the DOAs from the peak locations of the pseudospectrum

$$\widehat{p}_{\text{mu}}(\theta) = \frac{1}{\mathbf{a}^H(\theta) \widehat{\mathbf{\Xi}}_\varepsilon \widehat{\mathbf{\Xi}}_\varepsilon^H \mathbf{a}(\theta)}. \quad (2.107)$$

The second step of MUSIC could be replaced by finding the roots of the equation

$$\mathbf{a}^T(z^{-1}) \widehat{\mathbf{\Xi}}_\varepsilon \widehat{\mathbf{\Xi}}_\varepsilon^H \mathbf{a}(z) = 0, \quad (2.108)$$

where

$$\mathbf{a}(z) = [1 \ z^{-1} \cdots z^{-(L-1)}]^T \quad (2.109)$$

and obtaining the DOAs from the angles of the  $K$  closest roots to the unit circle. This method is known as Root MUSIC [38].

It is interesting to point out that Pisarenko's method is a special case of MUSIC. Namely, we obtain it if  $L = K + 1$ , and we can view it as the simplest version of MUSIC.

**2.4.3.5 ESPRIT** ESPRIT is another subspace-based method in that it relies on the spectral decomposition of  $\mathbf{C}_y$  from which we obtain the signal and noise subspaces. ESPRIT stands for estimation of signal parameters by rotational invariance techniques.

For the derivation of ESPRIT, it is important to rewrite the matrix  $\mathbf{A}$  [of size  $(L \times K)$ ]

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{-j\omega_c \tau_1} & e^{-j\omega_c \tau_2} & \cdots & e^{-j\omega_c \tau_K} \\ e^{-j2\omega_c \tau_1} & e^{-j2\omega_c \tau_2} & \cdots & e^{-j2\omega_c \tau_K} \\ \vdots & \vdots & \vdots & \vdots \\ e^{-j(L-1)\omega_c \tau_1} & e^{-j(L-1)\omega_c \tau_2} & \cdots & e^{-j(L-1)\omega_c \tau_K} \end{bmatrix}. \quad (2.110)$$

From  $\mathbf{A}$ , we define the  $(L - 1) \times K$  matrices  $\mathbf{A}_1$  and  $\mathbf{A}_2$ , which are obtained by removing the last and the first row of  $\mathbf{A}$ , respectively. It is not difficult to see that they are related by

$$\mathbf{A}_2 = \mathbf{A}_1 \Psi, \quad (2.111)$$

where

$$\Psi = \text{diag}\{e^{-j\omega_c \tau_1} e^{-j\omega_c \tau_2} \cdots e^{-j\omega_c \tau_K}\}. \quad (2.112)$$

Our objective is to find  $\Psi$ .

Now we rewrite (2.97)

$$\mathbf{\Xi}_s = \mathbf{A} \mathbf{T}. \quad (2.113)$$

From  $\mathbf{\Xi}_s$  we obtain  $\mathbf{\Xi}_{s1}$  and  $\mathbf{\Xi}_{s2}$  analogously as we obtained  $\mathbf{A}_1$  and  $\mathbf{A}_2$  from  $\mathbf{A}$ . Then we can write

$$\mathbf{\Xi}_{s1} = \mathbf{A}_1 \mathbf{T}, \quad (2.114)$$

$$\mathbf{\Xi}_{s2} = \mathbf{A}_2 \mathbf{T}. \quad (2.115)$$

Then

$$\begin{aligned}\Xi_{s2} &= A_2 T \\ &= A_1 \Psi T \\ &= \Xi_{s1} T^{-1} \Psi T \\ &= \Xi_{s1} \Phi,\end{aligned}\tag{2.116}$$

where we define  $\Phi$  by

$$\Phi = T^{-1} \Psi T.\tag{2.117}$$

Clearly,  $\Psi$  and  $\Phi$  have the same eigenvalues. Therefore, if we solve for  $\Phi$  in (2.116) and find its eigenvalues, we have the solution, that is, the diagonal elements of  $\Psi$ . For the solution of  $\Phi$  we can write

$$\widehat{\Phi} = \left( \widehat{\Xi}_{s1}^H \widehat{\Xi}_{s1} \right)^{-1} \widehat{\Xi}_{s1}^H \widehat{\Xi}_{s2},\tag{2.118}$$

where  $\widehat{\Xi}_{s1}$  and  $\widehat{\Xi}_{s2}$  are obtained from  $\widehat{\Xi}_s$ , which is constructed from the eigenvectors spanning the signal subspace. Once  $\widehat{\Phi}$  is obtained, we compute its eigenvalues, which, as already pointed out, are also estimates of the eigenvalues of  $\Psi$ , and from them we determine the DOAs.

In summary, ESPRIT is implemented as follows:

1. Compute the covariance matrix of  $y[t]$  by

$$\widehat{\mathbf{C}}_y = \frac{1}{T} \sum_{t=0}^{T-1} y[t] y^H[t].\tag{2.119}$$

2. Find the eigenvectors and eigenvalues of  $\widehat{\mathbf{C}}_y$  and construct  $\widehat{\Xi}_s$ ,  $\widehat{\Xi}_{s1}$ , and  $\widehat{\Xi}_{s2}$ .
3. Solve the linear system of equations

$$\widehat{\Xi}_{s1} \Phi = \widehat{\Xi}_{s2}.\tag{2.120}$$

4. Compute the eigenvalues of  $\widehat{\Phi}$ , and from them the DOAs of the signals.

The estimation of  $\Phi$  in step 3 can be accomplished by either standard least-squares or total least-squares methods [15].

**2.4.3.6 Bayesian Method Based on MCMC** A Bayesian approach to the DOA estimation would amount to estimating the a posteriori density of the unknowns. We note that this is a more ambitious objective than simply obtaining point estimates of the unknowns. Here, we briefly lay out an approach to finding the posterior that exploits Markov chain Monte Carlo (MCMC) sampling, a methodology that applies Markov chains in order to draw samples from a target distribution (which in this case is the posterior distribution of the unknowns) [39].

First, we write the likelihood of the data

$$f(\mathbf{Y}|\boldsymbol{\theta}, \mathbf{S}, \sigma_e^2) = \prod_{t=0}^{T-1} \frac{1}{\pi^L \sigma_e^2} \exp \left[ -\frac{1}{\sigma_e^2} (\mathbf{y}[t] - \mathbf{A}\mathbf{s}[t])^H (\mathbf{y}[t] - \mathbf{A}\mathbf{s}[t]) \right],\tag{2.121}$$

where the unknown parameters are the DOAs  $\boldsymbol{\theta}$ , the signal amplitudes  $s[t]$ , and the noise variance  $\sigma_\varepsilon^2$ . With a prior for these parameters given by  $f(\boldsymbol{\theta}, \mathbf{S}, \sigma_\varepsilon^2)$ , we can formally obtain their posterior, that is,

$$f(\boldsymbol{\theta}, \mathbf{S}, \sigma_\varepsilon^2 | \mathbf{Y}) \propto f(\mathbf{Y} | \boldsymbol{\theta}, \mathbf{S}, \sigma_\varepsilon^2) f(\boldsymbol{\theta}, \mathbf{S}, \sigma_\varepsilon^2) \quad (2.122)$$

where  $\propto$  signifies “proportional to.” Given the model and the data, all the information about the unknowns is contained in (2.122).

As before, the signal amplitudes and the noise variance are considered unimportant and therefore we integrate them out. This can be accomplished analytically with a proper choice of the priors. We then have

$$f(\boldsymbol{\theta} | \mathbf{Y}) \propto \int_{\mathbf{S}, \sigma_\varepsilon^2} f(\mathbf{Y} | \boldsymbol{\theta}, \mathbf{S}, \sigma_\varepsilon^2) f(\boldsymbol{\theta}, \mathbf{S}, \sigma_\varepsilon^2) d\mathbf{S} d\sigma_\varepsilon^2. \quad (2.123)$$

Note that in the above expression, we do not need to know the proportionality constant of the posterior density.

Once we obtain the solution in (2.123), we want to draw samples from it. Since it is difficult to generate these samples directly, we construct a Markov chain that basically amounts to drawing samples from a proposal density  $q(\boldsymbol{\theta})$  and accepting the proposed sample with probability that is computed for the proposed  $\boldsymbol{\theta}$ . The MCMC method needs time to converge and, therefore, all the samples that were drawn before the chain reached convergence are thrown away.

With the generated samples we can construct all kinds of point estimates and obtain various confidence intervals. MCMC-based estimation of DOA estimation was reported in [40].

#### 2.4.4 Estimation of Number of Impinging Signals

For the parametric methods, it is critical that we know the number of signals that impinge on the array. In applying the nonparametric methods, knowledge of  $K$  is not necessary, and when  $K$  is unknown, we determine it from the number of peaks in the spectrum of the data. This task, however, is often not easy either because of low SNRs or due to lack of resolution of the applied method. In this section we briefly present two general approaches to determining the number of impinging signals. One is based on information-theoretic criteria and the other on the reversible-jump MCMC (RJMCMC) sampling methodology. They are both based on parametric models of the data.

**2.4.4.1 Information-Theoretic Approaches** The determination of number of signals in the data  $\mathbf{Y}$  can be viewed as a model selection problem. The data can represent noise only or one signal in noise, or two signals in noise, and so on. Given  $\mathbf{Y}$  and the mathematical description of the models, the objective is to determine the correct model of the data. This is a special type of model selection, where the models are nested, meaning that the simpler models are obtained from the more complex ones by setting some of the parameters of the more complex models to zero.

The literature on information theory, statistics, and signal processing is abundant with articles on model selection. Here we briefly describe two approaches that are based on information-theoretic criteria. One of them is due to Akaike [26] and the

other due to Rissanen [27] and Schwartz [41]. The former criterion is known as AIC and the latter as MDL (where the acronym stands for minimum description length).

These criteria are composed of two terms, a data term and a penalty term, that is,

$$\zeta_k = -\log f(\mathbf{Y}; \hat{\varphi}_k) + \rho_k, \quad (2.124)$$

where  $\zeta_k$  is the value of the criterion for a model with  $k$  signals,  $\hat{\varphi}_k$  is the vector of ML estimates of the model parameters,  $f(\mathbf{Y}; \hat{\varphi}_k)$  is the probability density function of the data computed for  $\hat{\varphi}_k$ , and  $\rho_k$  is the penalty term for the model that penalizes for adding unnecessary parameters to the model. So, one would first compute the ML estimates of the parameters of all the models, compute the criteria  $\zeta_k$ , and choose the model that has the smallest  $\zeta_k$ .

In Akaike's method, under i.i.d. assumptions, the penalty is given by

$$\rho_k = m_k, \quad (2.125)$$

where  $m_k$  is the number of free adjusted parameters of the model, and in Rissanen/Schwartz's method by

$$\rho_k = \frac{1}{2}m_k \log T, \quad (2.126)$$

where  $m_k$  has the same meaning as in (2.125) and  $T$  is the number of observations.

The problem of choosing the number of impinging signals on an array of sensors with model assumptions that allow for writing the likelihood function as in (2.87) was studied in [42]. There, it was shown that the log-likelihood term in (2.124) can be expressed as

$$\log f(\mathbf{Y}; \hat{\varphi}_k) = \log \left( \frac{\prod_{l=k+1}^L \hat{\lambda}_l^{1/(L-k)}}{[1/(L-k)] \sum_{l=k+1}^L \hat{\lambda}_l} \right)^{(L-k)T}, \quad (2.127)$$

where  $k$  is the assumed number of signals and  $\hat{\lambda}_l$ ,  $l = k+1, k+2, \dots, L$  are the smallest  $L-k$  eigenvalues of  $\hat{\mathcal{C}}_y$ . The penalties of the AIC and MDL criteria are

$$\begin{aligned} \text{AIC : } & \rho_k = k(2L - k), \\ \text{MDL : } & \rho_k = \frac{1}{2}k(2L - k) \log T. \end{aligned} \quad (2.128)$$

It can be shown that the AIC criterion is not consistent, which means that as the number of data snapshots tends to infinity, the probability of correct selection of the model does not tend to one. By contrast, the MDL criterion is consistent. In general, the AIC criterion tends to overestimate the number of impinging signals. The theoretical performance of these criteria is provided in [43]. Modified information-theoretic rules appeared in [44], and more recent results on these criteria can be found in [45].

**2.4.4.2 RJMCMC for Estimating Number of Signals and Their DOAs** The MCMC method described in Section 2.4.3.6 can be extended to drawing samples from spaces that are of different dimensions [46]. This method is known as reversible jump

MCMC (RJMCMC) and would be of interest in scenarios when the number of signals is unknown. Then we formally write

$$f(\mathbf{Y}|k, \boldsymbol{\theta}, \mathbf{S}, \sigma_{\varepsilon}^2) = \prod_{t=0}^{T-1} \frac{1}{\pi^L \sigma_{\varepsilon}^2} \exp \left[ -\frac{1}{\sigma_{\varepsilon}^2} (\mathbf{y}[t] - \mathbf{A}\mathbf{s}[t])^H (\mathbf{y}[t] - \mathbf{A}\mathbf{s}[t]) \right], \quad (2.129)$$

where  $k$  denotes the number of assumed signals and the rest of the parameters have the same meaning as before. The posterior is given by

$$f(\boldsymbol{\theta}, \mathbf{S}, \sigma_{\varepsilon}^2, k | \mathbf{Y}) \propto f(\mathbf{Y} | \boldsymbol{\theta}, \mathbf{S}, \sigma_{\varepsilon}^2, k) f(\boldsymbol{\theta}, \mathbf{S}, \sigma_{\varepsilon}^2, k). \quad (2.130)$$

The objective now is to obtain the joint posterior  $f(\boldsymbol{\theta}, k | \mathbf{Y})$ , so we integrate the nuisance parameters, that is,

$$f(\boldsymbol{\theta}, k | \mathbf{Y}) \propto \int_{\mathbf{S}, \sigma_{\varepsilon}^2, k} f(\mathbf{Y} | \boldsymbol{\theta}, \mathbf{S}, \sigma_{\varepsilon}^2) f(\boldsymbol{\theta}, \mathbf{S}, \sigma_{\varepsilon}^2, k) d\mathbf{S} d\sigma_{\varepsilon}^2. \quad (2.131)$$

The integration can be carried out as in (2.123).

The implementation of the method is similar to the one of regular MCMC. The proposal density has three types of moves, staying in the current parameter space, increasing the number of signals by one (i.e., giving birth to a signal) or removing one of the existing signals in the current space (known as a death move). The posterior probability of a model is proportional to the time a chain spends in a particular space. From the samples generated from specific space, we can construct posterior marginal distributions. For details of applying the method, see [40].

## 2.5 FINAL REMARKS

In this chapter we reviewed the methods for spatial and temporal spectrum estimation. We discussed the classical nonparametric approaches and the more recent parametric methods. This field, although mature, still attracts attention of the research community simply because the areas of its applications steadily increase.

## REFERENCES

1. E. A. Robinson, “A historical perspective of spectrum estimation,” *Proc. IEEE*, vol. 70, pp. 886–907, 1982.
2. A. Schuster, “On the investigation of hidden periodicities with application to a supposed 26-day period of meteorological phenomena,” *Terrestr. Magnet.*, vol. 3, pp. 13–41, 1898.
3. J. W. Cooley and J. W. Tukey, “An algorithm for the machine calculation of complex Fourier series,” *Math. Comput.*, vol. 19, pp. 297–301, 1965.
4. J. P. Burg, “A new analysis technique for time series data,” in *NATO Advanced Study Institute on Signal Processing with Emphasis on Underwater Acoustics*, Enschede, Netherlands, 1968.
5. J. P. Burg, “Maximum entropy spectral analysis”, PhD dissertation, Stanford University, 1975.
6. J. Capon, “High-resolution frequency-wavenumber spectrum analysis,” *Proc. IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

7. H. Hotelling, "Analysis of a complex of statistical variables with principal components," *J. Ed. Psychol.*, vol. 24, pp. 417–441, 498–520, 1933.
8. T. Koopmans, *Linear Regression Analysis of Economic Time Series*, De Erven F. Bohn N. V.: Harlem, 1937.
9. V. F. Pisarenko, "The retrieval of harmonics from a covariance function," *Geophys. J. R. Astron. Soc.*, vol. 33, pp. 347–366, 1973.
10. G. Bienvenu and L. Kopp, "Adaptivity to background noise spatial coherence for high resolution passive methods," in the *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1980, pp. 307–310.
11. R. O. Schmidt, "A signal subspace approach to multiple emitter location and spectral estimation," PhD dissertation, Stanford University, 1981.
12. L. Marple, *Digital Spectral Analysis with Applications*, Englewood Cliffs, NJ: 1997.
13. P. Stoica and R. Moses, *Spectral Analysis of Signals*, Upper Saddle River, NJ: Pearson Prentice Hall, 2005.
14. H. L. Van Trees, *Optimum Array Processing*, Wiley, 2002.
15. H. Krim and M. Viberg, "Two decades of array signal processing," *IEEE Signal Process. Mag.*, vol. 4, pp. 67–94, 1996.
16. B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE Signal Process. Mag.*, vol. 2, pp. 4–24, 1988.
17. M. B. Priestley, *Spectral Analysis and Time Series*, New York: Academic 1981.
18. S. M. Kay, *Modern Spectral Estimation*, Englewood Cliffs, NJ: Prentice Hall, 1988.
19. R. B. Blackman and J. W. Tukey, *The Measurement of Power Spectra from the Point of View of Communications Engineering*, New York: Dover, 1958.
20. D. J. Thomson, "Spectrum estimation and harmonic analysis," *Proc. IEEE*, vol. 72, no. 9, pp. 1055–1096, 1982.
21. C. T. Mullis and L. L. Scharf, "Quadratic estimators of the power spectrum," in *Advances in Spectrum Analysis and Array Processing*, S. Haykin (Ed.), Englewood Cliffs, NJ: Prentice Hall, 1991, pp. 395–402.
22. D. B. Percival and A. T. Walden, *Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques*, Cambridge: Cambridge University Press, 1993.
23. K. Reidel and A. Sidorenko, "Minimum bias multiple taper spectral estimation," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 188–195, 1989.
24. D. J. Thomson, "Jackknifing multitaper spectrum estimates," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 20–30, 2007.
25. P. Stoica and Y. Selén, "Model order selection," *IEEE Signal Process. Mag.*, vol. 21, no. 4, pp. 36–47, 2004.
26. H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Control*, vol. AC-19, no. 4, pp. 716–723, 1974.
27. J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.
28. J. F. Böehme, "Array processing," in *Advances in Spectrum Analysis and Array Processing*, S. Haykin (Ed.), Prentice Hall, 1991, pp. 97–63.
29. J. Benesty, J. Chen, and Y. Huang, "A generalized MVDR spectrum," *IEEE Signal Process. Lett.*, vol. 12, no. 12, pp. 827–830, 2005.
30. A. Drosopoulos and S. Haykin, "Adaptive radar parameter estimation with Thomson's multiple window method," in *Adaptive Radar Detection and Estimation*, S. Haykin and A. Steinhardt, (Eds.), New York: Wiley, 1991.

31. T.-C. Liu and D. Van Veen, "Multiple window based minimum variance spectrum estimation for multidimensional random fields," *IEEE Trans. Signal Process.*, vol. 40, no. 3, pp. 578–589, 1992.
32. M. Viberg and A. L. Swindlehurst, "A Bayesian approach to auto-calibration for parametric array signal processing," *IEEE Trans. Signal Process.*, vol. 42, no. 12, pp. 3073–3083, 1989.
33. P. Stoica, R. Moses, B. Friedlander, and T. Söderström, "Maximum likelihood estimation of the parameters of multiple sinusoids from noisy measurements," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 3, pp. 378–392, 1989.
34. I. Ziskind and M. Wax, "Maximum likelihood localization of multiple sources by alternating projection," *IEEE Trans. Signal Process.*, vol. 36, no. 10, pp. 1553–1560, 1988.
35. B. Ottersten, M. Viberg, P. Stoica, and A. Nehorai, "Exact and large sample ML techniques for parameter estimation from sensor array data," in *Radar Array Processing*, S. Haykin, J. Litva, and T. J. Shepherd (Eds.), Springer-Verlag, 1993, pp. 99–151.
36. B. Ottersten, M. Viberg, and T. Kailath, "Analysis of subspace fitting and ML techniques for parameter estimation from sensor array data," *IEEE Trans. Signal Process.*, vol. 40, no. 3, pp. 590–600, 2002.
37. G. Biennvenu, "Influence of the spatial coherence of the background in high resolution passive methods," in the *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1979, pp. 306–309.
38. A. J. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in the *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1983, pp. 336–339.
39. W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, New York: Chapman & Hall, 1996.
40. J.-R. Larocque and J. P. Reilly, "Reversible jump MCMC for joint detection and estimation of sources in colored noise," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 231–240, 2002.
41. G. Schwartz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, pp. 461–464, 1978.
42. M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 33, no. 2, pp. 387–392, 1985.
43. M. Kaveh, H. Wang, and H. Hung, "On the theoretical performance of a class of estimators of the number of narrowband sources," *IEEE Trans. Acoust. Speech, Signal Process.*, vol. 35, no. 19, pp. 1350–1352, 1987.
44. K. M. Wong, Q.-T. Zhang, J. Reilly, and P. Yip, "On information theoretic criteria for determining the number of signals in high resolution array processing," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 38, no. 11, pp. 1959–1971, 1990.
45. A. P. Liavas and P. A. Regalia, "On the behavior of information theoretic criteria for model order selection," *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1689–1695, 2001.
46. P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 4, pp. 711–732, 1995.

## CHAPTER 3

---

# MIMO Radio Propagation

Tricia J. Willink

Communications Research Centre, Ottawa, Ontario, Canada

### 3.1 INTRODUCTION

With the proliferation of wireless devices and the continuing introduction of new, higher bandwidth services, there is a need for high-data-rate technologies that can achieve high spectral efficiency. Multiinput multioutput (MIMO) communications systems are a strong contender to satisfy this need in many propagating environments. MIMO systems use diversity in space and time (and are hence also called space–time systems) to achieve higher capacities, which translates into improved error rates, larger range, and/or higher throughputs than conventional systems. The diversity results from the scattering of signal energy from objects in the surrounding environment and is exploited by using multiple antenna elements at both the transmitter and receiver.

Channel capacity analysis with early, idealized MIMO channel models suggested that huge gains could be achieved—up to a linear increase with the number of antenna elements in the smaller of the two arrays. These idealized models assumed that the channel responses between each pair of transmitter/receiver antenna elements were independent: This follows from the assumptions that there is an infinite number of scatterers distributed around the antenna arrays and that the antenna elements are separated enough to make the fading correlation between them negligible.

As the development of MIMO technologies advances, the significance of these assumptions becomes more apparent. In real operating environments, the scatterers are not as numerous, nor as ideally distributed. This results in correlated fading, reducing the channel capacity and the performance of the MIMO techniques. The need for improved MIMO channel models was clear, and many resources have been invested in achieving a better understanding of the space–time propagating environment. Mobile communications systems are of increasing interest—these introduce the need to consider time-varying channel characteristics. Particular attention is paid to mobility aspects of space–time channels in this chapter.

An introduction to the space–time propagation environment is given in Section 3.2, and a survey of MIMO propagation models is presented in Section 3.3. The range of modeling philosophies and parameterizations indicates that there is no “right model”

and that the model used should be selected according to the situation. Some models oversimplify the channel characteristics but are analytically tractable. Others represent the physical propagation mechanisms more accurately but are better suited to simulation. Users must be aware of the limitations of the model they select and be careful about making conclusions and generalizations based on the model that are unsupportable in the real world.

In Section 3.4, measured data obtained using a fixed base station and mobile terminal in urban Ottawa, Canada, are used to parameterize some of the models discussed in the previous section. The model parameters are observed for large-scale variations in the propagating environment, illustrating the need to test MIMO systems using a wide range of model realizations.

An important assumption in the modeling and simulation of radio channels is that they are wide-sense stationary. In Section 3.5, a methodology is presented to test whether short series of channel measurements are wide-sense stationary. This is applied to the measured data from Section 3.4, demonstrating that this stationarity assumption does not hold for approximately 60–90% of one-half-second intervals.

### 3.2 SPACE-TIME PROPAGATION ENVIRONMENT

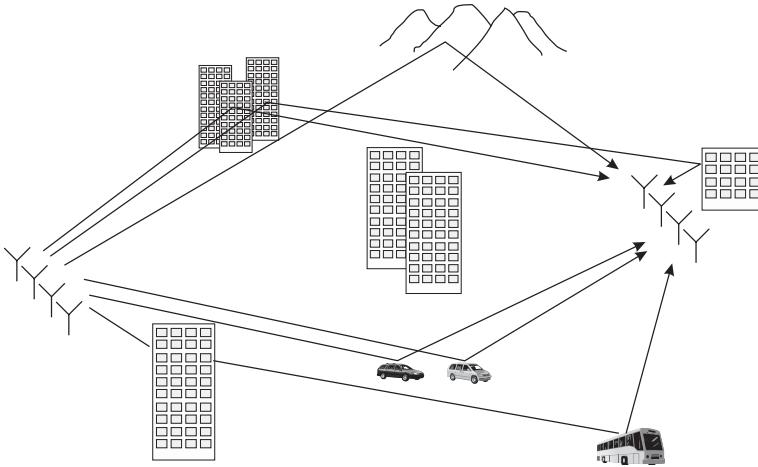
When signal wavefronts propagate through the physical environment, they interact with objects of different sizes, textures, and materials. The wavefronts may be diffracted over or around large objects, reflected or scattered from others (reflection occurs from objects that are smooth on a scale relative to the wavelength of the propagating signal, and scattering occurs when the object is rough on the same scale).

Consider single-element antennas at fixed transmitter and receiver locations and static interacting objects. Interacting objects are described in [1] as objects that interact with the electromagnetic field and influence the signal observed at the receiver. The signal therefore arrives at the receiver over many different paths, as multipath components (MPCs) that have different amplitudes, phases, delays, and angles of departure and arrival (Fig. 3.1). The resulting impulse response is described by the double-directional channel model, which was presented in [2] and earlier work by the same authors and their colleagues. For  $L$  MPCs, the double directional channel model is

$$h(\tau, \theta_r, \theta_t, \phi_r, \phi_t) = \sum_{\ell=1}^L h_\ell(\tau, \theta_r, \theta_t, \phi_r, \phi_t). \quad (3.1)$$

In a static environment, the parameters of each MPC are constant and dependent on the precise location of the transmitter and receiver antenna elements. The  $\ell$ th MPC departs the transmitter with angles of elevation,  $\theta_{t,\ell}$ , and azimuth,  $\phi_{t,\ell}$ , and arrives with complex amplitude  $a_\ell$ , delay  $\tau_\ell$ , and elevation and azimuthal angles  $\theta_{r,\ell}$  and  $\phi_{r,\ell}$ , respectively. The contribution of the  $\ell$ th MPC is then given by

$$h_\ell(\tau, \theta_r, \theta_t, \phi_r, \phi_t) = a_\ell \delta(\tau - \tau_\ell) \delta(\theta_r - \theta_{r,\ell}) \delta(\theta_t - \theta_{t,\ell}) \delta(\phi_r - \phi_{r,\ell}) \delta(\phi_t - \phi_{t,\ell}). \quad (3.2)$$



**Figure 3.1** Space–time multipath environment.

To include the effects of polarization, which provide significant diversity in many propagating environments, the double-directional expression for the  $\ell$ th MPC can be written as a polarimetric  $2 \times 2$  matrix [1] in which the complex amplitude  $a_\ell$  is replaced by

$$\mathbf{a}_\ell = \begin{bmatrix} a_\ell^{vv} & a_\ell^{vh} \\ a_\ell^{hv} & a_\ell^{hh} \end{bmatrix}, \quad (3.3)$$

where the superscripts  $v$  and  $h$  denote the vertical and horizontal polarizations, respectively. Each MPC in (3.1) is then replaced by its corresponding  $2 \times 2$  polarimetric matrix to give

$$\mathbf{h}(\tau, \theta_r, \theta_t, \phi_r, \phi_t) = \sum_{\ell=1}^L \begin{bmatrix} h_\ell^{vv}(\tau, \theta_r, \theta_t, \phi_r, \phi_t) & h_\ell^{vh}(\tau, \theta_r, \theta_t, \phi_r, \phi_t) \\ h_\ell^{hv}(\tau, \theta_r, \theta_t, \phi_r, \phi_t) & h_\ell^{hh}(\tau, \theta_r, \theta_t, \phi_r, \phi_t) \end{bmatrix}. \quad (3.4)$$

The remainder of this chapter will assume a single polarization.

For small changes in the positions of the antenna elements, the interacting objects are assumed to be in the far field, which means that the wavefronts arriving from the interacting objects at a second receiver element, located a short distance from the first, will have the same amplitudes, delays, and angles of arrival but different phases due to the different path lengths. At the second element, the MPCs combine in a different way than at the first. Similarly, signals emitted from a second transmitter antenna element, located near the first, will have the same angles of departure, interacting with the same objects, but the MPCs will arrive at a receiver element with different phases.

Consider  $N_t$  transmitter elements and  $N_r$  receiver elements, arranged in arrays small enough that the interacting objects can be considered to be in the far field. The transmitter array pattern is the length  $N_t$  vector  $\psi_t(\theta_t, \phi_t)$ , which is the vector of complex amplitudes at each antenna element for a signal transmitted in the direction  $(\theta_t, \phi_t)$ . Similarly, the  $N_r$  elements of the receiver array pattern  $\psi_r(\theta_r, \phi_r)$  give the

complex responses of the antenna array elements to a signal arriving from the direction  $(\theta_r, \phi_r)$ . For example, assuming that the MPCs propagate in the horizontal plane ( $\theta_t = \theta_r = 0$ ), the array patterns for uniform linear arrays with antenna elements spaced by  $d$  are

$$\psi_t(\phi_t) = [1 \quad \exp(-j\frac{2\pi d}{\lambda} \sin \phi_t) \quad \dots \quad \exp[-j\frac{2\pi d}{\lambda}(N_t - 1) \sin \phi_t]]^T, \quad (3.5)$$

$$\psi_r(\phi_r) = [1 \quad \exp(-j\frac{2\pi d}{\lambda} \sin \phi_r) \quad \dots \quad \exp[-j\frac{2\pi d}{\lambda}(N_r - 1) \sin \phi_r]]^T, \quad (3.6)$$

where  $\lambda$  is the carrier wavelength, and the departure and arrival angles are measured such that  $\phi = 0$  is perpendicular to the array. A linear array cannot distinguish signals in  $[-\pi/2, \pi/2]$  from those in  $[-3\pi/2, -\pi/2]$ , that is, whether the signals arrive from in front of or behind the array baseline, because as seen from (3.5) and (3.6), the angles  $\phi$  and  $\pi - \phi$  result in the same array patterns.

Using the double-directional channel model (3.1), the impulse response between the  $m$ th element of the transmit array and the  $k$ th element of the receive array can be found using

$$h_{k,m}(\tau) = \int_{\phi_t} \int_{\phi_r} h(\tau, \phi_r, \phi_t) \psi_t^{(m)}(\phi_t) \psi_r^{(k)}(\phi_r) d\phi_r d\phi_t, \quad (3.7)$$

where  $\psi_t^{(m)}$  and  $\psi_r^{(k)}$  are the  $m$ th and  $k$ th elements of  $\psi_t$  and  $\psi_r$ , respectively. Note that the elevation angles will be omitted henceforth, for simplicity. For the uniform linear arrays considered above,

$$h_{k,m}(\tau) = \sum_{\ell=1}^L a_\ell \delta(\tau - \tau_\ell) \exp\left[-j\frac{2\pi d}{\lambda}(m-1) \sin \phi_{t,\ell}\right] \exp\left[-j\frac{2\pi d}{\lambda}(k-1) \sin \phi_{r,\ell}\right]. \quad (3.8)$$

If the separation between antenna elements is large, the signals arriving from different transmitter elements, or at different receiver elements, may interact with distinct objects in the environment. In this case the MPCs do not have the same trajectories and may differ not just in phase but also in number, amplitude, delay, and angle. Similarly, if the interacting objects are in the near field of either the transmitter or receiver arrays, the multipath angles of departure or arrival are different for each element in the array. In both cases, the impulse response  $h(\tau, \phi_r, \phi_t)$  must be computed separately for the location of each element in the array using (3.2).

When the environment is not static, that is, if some of the interacting objects, the transmitter, or the receiver are moving, the combination of the MPCs at the receiver changes with time, resulting in multipath fading. Then the link between the  $m$ th element of the transmitter array and the  $k$ th element of the receiver array has the time-dependent impulse response

$$h_{k,m}(t, \tau) = \int_{\phi_t} \int_{\phi_r} h(t, \tau, \phi_r, \phi_t) \psi_t^{(m)}(\phi_t) \psi_r^{(k)}(\phi_r) d\phi_r d\phi_t. \quad (3.9)$$

The MIMO system model is

$$\mathbf{r}(t) = \int_{\tau} \mathbf{H}(t, \tau) \mathbf{s}(t - \tau) d\tau + \mathbf{n}(t), \quad (3.10)$$

where the  $(k, m)$ th element of the  $N_r \times N_t$  channel response matrix  $\mathbf{H}(t, \tau)$  is  $h_{k,m}(t, \tau)$ . The length  $N_r$  noise vector  $\mathbf{n}(t)$  is assumed to be independent of both the length  $N_t$  signal vector and the channel response. For a narrowband (frequency-flat) system model, the delays of the MPCs cannot be resolved and the system model becomes

$$\mathbf{r}(t) = \mathbf{H}(t)\mathbf{s}(t) + \mathbf{n}(t). \quad (3.11)$$

In a multielement antenna system, the fading observed at each antenna element is different: This is the key to MIMO communications. In an ideal scenario, the fading experienced by signals transmitted from each of the  $N_t$  transmitter elements and received by each of the  $N_r$  receiver elements would be uncorrelated. This is the “rich-scattering,” uncorrelated fading environment often used in the literature, in which there is a large number of interacting objects distributed in all directions around the transmitter and receiver. In practice, particularly in mobile, outdoor environments, the interacting objects are limited in number and are located with a nonuniform distribution around the transmitter and receiver. This causes correlation among the  $N_t \cdot N_r$  communication links, and reduces the theoretical capacity of the MIMO radio channel.

### 3.2.1 MIMO Channel Capacity

The extensive interest in MIMO communications systems in the last decade is rooted in their potential increases in spectral efficiency or achievable throughput per unit bandwidth. Early pioneering work [3–5] indicated the capacity of MIMO channels, relative to using single antenna elements at the transmitter and receiver, could increase linearly with the minimum number of antenna elements in the two arrays. The capacity of MIMO channels with various assumptions about the type of channel state information (CSI) was evaluated by Goldsmith et al. in [6]; only the simple cases of CSI at the receiver, and CSI at the transmitter and receiver, are discussed here.

In the following discussion, the singular value decomposition (SVD) of the channel matrix will be used. This is

$$\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^H, \quad (3.12)$$

where the columns of the  $N_r \times N_r$  unitary matrix  $\mathbf{U} = [\mathbf{u}_1 \ \cdots \ \mathbf{u}_{N_r}]$  are the left singular vectors, the columns of the  $N_t \times N_t$  unitary matrix  $\mathbf{V} = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_{N_t}]$  are the right singular vectors. The  $\min(N_t, N_r)$  leading diagonal elements of the  $N_r \times N_t$  matrix  $\Sigma$  are the real, nonnegative singular values,  $\sigma_i$ .

Consider a narrowband  $N_r \times N_t$  MIMO channel with channel response matrix  $\mathbf{H}(t)$ . For a time-varying channel with no CSI at the transmitter, the optimal signaling strategy is to transmit each signal stream from a different antenna element and to apply equal transmit power to each. The capacity is given here per unit bandwidth, in bits/second/Hertz. In this case, the mean capacity is

$$C = \mathcal{E} \left\{ \log_2 \det \left( \mathbf{I} + \frac{\gamma}{N_t} \mathbf{H}(t) \mathbf{H}^H(t) \right) \right\} = \sum_{i=1}^{\min(N_t, N_r)} \mathcal{E} \left\{ \log_2 \left( 1 + \frac{\gamma}{N_t} \sigma_i^2(t) \right) \right\}, \quad (3.13)$$

where  $\gamma$  is the mean signal-to-noise ratio (SNR) and  $\mathcal{E} \{\cdot\}$  denotes the expectation over time.

For a static channel with response  $\mathbf{H}$ , when perfect CSI is available at the transmitter, the capacity is maximized when the channel is separated into  $\min(N_t, N_r)$  orthogonal subchannels and the power is allocated according to the well-known “water-filling” strategy [7, Chapter 8]. Then

$$C = \max_{\mathbf{Q}: \text{tr}\{\mathbf{Q}\}=P} \log_2 \det (\mathbf{I} + \gamma \mathbf{HQH}^H), \quad (3.14)$$

where  $\mathbf{Q}$  is the covariance of the transmitted signal vector, and  $\text{tr}\{\cdot\}$  denotes the matrix trace. The optimum transmitted signal vector is  $\mathbf{Vs}$ , which is the data vector weighted by the right singular vectors of the channel matrix, and the power allocated to the  $i$ th data substream is

$$P_i = \left( \mu - \frac{1}{\gamma \sigma_i^2} \right)^+, \quad (3.15)$$

where  $(a)^+ = \max(0, a)$  and  $\mu$  is the water-filling constant. The resulting capacity (per unit bandwidth) is

$$C = \sum_{i=1}^{\min(N_t, N_r)} (\log_2(\mu \gamma \sigma_i^2))^+. \quad (3.16)$$

Although this orthogonalization approach is appealing, it is not realistic for mobile communications. In a time-varying channel, the CSI at the transmitter will be both noisy and out of date, resulting in self-interference between the subchannels [8]. It was shown in [9] that the capacity in this case is limited, regardless of SNR, by the estimation error.

Both (3.13) and (3.16) show that the capacity is dependent on the singular values of the channel response matrix  $\mathbf{H}$ . The capacity would be maximized in each case if the columns of  $\mathbf{H}$  were orthonormal, in which case the singular values would be equal. Correlated fading among the  $N_t \cdot N_r$  links between pairs of transmitter and receiver antenna elements changes the distribution of the singular values, making the large values larger, and the small ones even smaller. This results in a decrease in the channel capacities in (3.13) and (3.16).

### 3.3 PROPAGATION MODELS

The marketing of MIMO as the source of huge capacity increases uses the assumption of uncorrelated fading. These assumptions were also the foundation of many developments in space–time signal processing strategies. The reality of MIMO channels is often quite distant from the ideal, hence modeling the characteristics of the space–time propagating environment is essential in the design of efficient strategies to exploit the potential gains provided by real MIMO channels.

There are two broad approaches to modeling MIMO radio channels. The first is to represent the physical environment literally: The contribution of each multipath component to the overall impulse response is quantified, that is, its amplitude, direction, and delay are characterized. This provides a parameterization of the double-directional channel model (3.1), which can subsequently be used to compute the MIMO channel response for given antenna arrays, as described in (3.9). Physical channel models describe the features of the propagation environment; when the features are site specific, the models can be used for network planning, alternatively the features may be

generated using statistical distributions, then the models provide realizations typical of selected environments.

The second approach is to model important effects of the physical environment, such as correlation, without linking them to specific features in the propagation path. This incorporates the effects of the antenna array into the model and leads directly to a model for the channel matrix  $\mathbf{H}$ . Analytical channel models describe the effects of the propagation environment without specifying the physical environment—they are site independent and are often used for evaluating signal processing schemes.

In this section, the main characteristics of these two types of propagation models are introduced. The measurements needed to obtain parameterizations of these models require sophisticated equipment and careful planning. Channel models used for communal purposes such as standardization activities must be based on large numbers of measurements, taken in many representative environments. Successful examples of this process have been seen within the European Union COST program; detailed descriptions of physical and analytical models developed within the collaborative COST 273 Action can be found in [10].

### **3.3.1 Physical Channel Models**

A brute-force approach to computing the double-directional channel model for a given location is to use ray tracing. This relies on an accurate modeling of the local environment and the corresponding solution of physical equations to generate the multipath components,  $h_\ell(\tau, \phi_r, \phi_t)$ . This approach poses a number of challenges, in particular the detail necessary to adequately describe the propagation environment, and the computation time and memory required to generate the spatial channel impulse response. Simplifications are required, for example, in dealing with the electromagnetic characteristics of unknown construction materials, and diffractive effects require that objects be approximated by wedges or other basic shapes to simplify the computation. To reduce computational resources, models are often limited to including just single and double “bounces,” or reflections. Typically, only specular reflection and diffraction are included in ray-tracing methods. Diffuse scattering, produced by rough surfaces, is often ignored. This, plus the inability to model the numerous smaller clutter in the propagation environment, may result in estimated correlation between antenna elements being higher than would be observed in practice as the missing phenomenon, scattering, tends to decorrelate the signals. Overall, ray tracing provides a model incorporating the dominant propagation effects and has been seen to have reasonable agreement with measurements. A discussion of recent advances in ray-tracing methods can be found in [10].

**3.3.1.1 Ring Models** Geometric models place scatterers in prescribed locations and use a simple ray-tracing approach to determine certain characteristics of the channel response, such as the Doppler spread, or the auto- and cross-correlation functions. Early geometric models use rings to specify the location of scatterers, and each scatterer is assumed to be a perfect reflector, reradiating the incoming signal omnidirectionally. Unlike true ray tracing, the effect of the material and shape of the interacting object is generally not considered explicitly but is incorporated into the path loss model used. The impact of the scatterer on the phase of each MPC is not considered, and the phases of the MPCs are assumed to be independent and uniformly distributed on  $[0, 2\pi]$ .

In an early model for a mobile transmitter and a fixed base station, Lee assumed that the scatterers were uniformly distributed around the edge of a circular disk centered on the mobile terminal [11]. As the mobile moved, the distribution of the scatterers

relative to the terminal remained constant. This is a single-bounce model in which the signal wavefronts are assumed to have a single interaction with a scattering object. This model is also the basis for the well-known Clarke two-dimensional isotropic scattering model [12, 13, Chapter 1]. Petrus et al. [14] assumed that the scatterers were uniformly distributed within the scattering ring to model the angular and Doppler characteristics of a macrocell. An alternate single-bounce model for microcells and picocells is the elliptical model introduced in [15] in which the transmitter and receiver form the foci of a series of ellipses. The path delays for MPCs interacting with scatterers on the same ellipse are the same, thus multiple concentric ellipses can be used to generate a tap-delay wideband MIMO model, as in [16].

One of the drawbacks of a single-bounce model is that it is unable to account separately for the scattering environments at the transmitter and receiver. In locations where there is scattering near both terminals, such as microcellular and picocellular environments, a double-bounce model may be more applicable in which the wavefronts are assumed to be reflected first from a scattering object near the transmitter and then from another one in the neighborhood of the receiver before arriving at the receiver terminal. This means that the angles of arrival are decoupled from the angles of departure and the delay, that is, the model is separable.

The two-ring MIMO scattering model was used by Byers and Takawira in [17] to account for both scattering at the transmitter and receiver as well as terminal mobility. As in the one-ring model, the radius of each scattering ring was assumed to be much smaller than the distance between the transmitter and receiver. Unlike Lee's model, in which the scatterers were assumed to move with the mobile, in the two-ring model used in [17] the scatterers are static and the mobile moves relative to them. As the mobile terminal moves over short distances, perhaps a few wavelengths, the time-varying channel response can be assumed to be wide-sense stationary. With this assumption, the space-time cross-correlation function

$$\rho_{ip,jq}(\tau) = \frac{\mathcal{E}\{h_{i,p}(t)h_{j,q}^*(t + \tau)\}}{\sqrt{\mathcal{E}\{|h_{i,p}(t)|^2\}}\sqrt{\mathcal{E}\{|h_{j,q}(t)|^2\}}} \quad (3.17)$$

was computed; this is the cross-correlation of the delayed complex channel responses on the link between transmitter element  $i$  and receiver element  $p$  and the link between transmitter and receiver elements  $j$  and  $q$ , respectively.

The double-bounce assumption leads to a separability of the transmitter and receiver characteristics, which means that the space-time cross-correlation function can be written as the product of the transmit and receive correlation functions, that is,

$$\rho_{ip,jq}(\tau) = \rho_{i,j}^t \rho_{p,q}^r(\tau), \quad (3.18)$$

where

$$\begin{aligned} \rho_{i,j}^t &= \frac{\mathcal{E}\{h_{i,m}(t)h_{j,m}^*(t)\}}{\sqrt{\mathcal{E}\{|h_{i,m}(t)|^2\}}\sqrt{\mathcal{E}\{|h_{j,m}(t)|^2\}}} \quad \text{and} \\ \rho_{p,q}^r(\tau) &= \frac{\mathcal{E}\{h_{n,p}(t)h_{n,q}^*(t + \tau)\}}{\sqrt{\mathcal{E}\{|h_{n,p}(t)|^2\}}\sqrt{\mathcal{E}\{|h_{n,q}(t)|^2\}}}. \end{aligned} \quad (3.19)$$

These transmit and receive autocorrelation functions are assumed to be independent of receive element index  $m$  and transmit element index  $n$ , respectively, as the same scattering objects are assumed to be visible to each element within the respective arrays. The transmit correlation is independent of  $\tau$  because the signals emitted have experienced no propagation delays. A similar observation was made by Pätzold et al. [18], who extended the two-ring model to the case of mobile-to-mobile MIMO.

The two-ring model was also used in [19] to model mobile-to-mobile MIMO narrowband channels. That model incorporates a line-of-sight component as well as both single bounces and double bounces due to scattering in the neighborhood of both terminals.

The distribution of scatterers in real scenarios is not uniform. Local concentrations of scatterers can be modeled in the one-ring, elliptical, and two-ring models by replacing the uniform angular distribution with one that more accurately reflects real scenarios. It has been observed that in many cases, the distribution is well modeled by the von Mises distribution function [20]

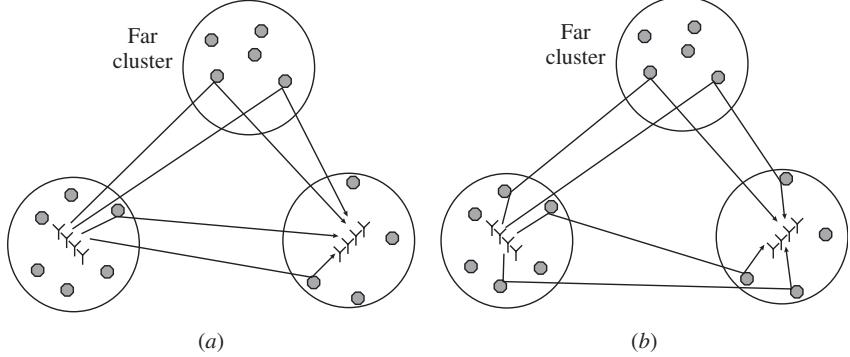
$$p_\phi(\phi) = \frac{1}{2\pi I_0(\kappa)} e^{\kappa \cos(\phi - \bar{\phi})}, \quad (3.20)$$

where  $I_0(\cdot)$  is the zeroth-order modified Bessel function and  $\bar{\phi}$  is the mean angle,  $\bar{\phi} \in [0, 2\pi)$ . The parameter  $\kappa \geq 0$  controls the angular spread:  $\kappa = 0$  leads to the uniform distribution of the original one-ring model; as  $\kappa$  increases, the angular spread decreases.

**3.3.1.2 Geometrically Based Stochastic Models** The ring models are analytically tractable for the purposes of obtaining statistical distributions such as the Doppler and space–time cross-correlation characteristics that can be used in simulating space–time channels. These models oversimplify the scattering environment; more recent geometrically based stochastic channel models (GSCMs), such as that introduced by Molisch et al. in [21], provide a more realistic representation of real scenarios while maintaining the generality not achieved by true ray tracing.

In GSCMs, scatterers are located over a geographic area using a specified statistical distribution, and the space–time channel response is computed using a simple ray-tracing routine. Implementations of the GSCM can use different distributions of scatterers, as discussed in [22], and relative path loss can be modeled by weighting the contributions of scatterers based on their proximity to the mobile terminal or by selecting a distribution in which the number of scatterers decreases with distance. The stochastic characteristics of many aspects of the channel model have been determined by measurements for different scenarios.

A significant collaborative effort has been directed at developing generalized physical models, such as within the EU COST Actions (e.g., former Actions 259 [23] and 273 [10]), IEEE Standards, and EU Partnerships NEWCOM [24] and WINNER [25]. Measurements obtained from a variety of locations, primarily in Europe and North America, have been analyzed and combined to generate databases of characteristics applicable to different types of radio environments, such as macrocellular, microcellular, and picocellular. The parameters used in the physical model then do not replicate the exact conditions of the operating environment but rather describe a typical setting of that broad type. Note that micro- and macrocells are distinguished by the



**Figure 3.2** Illustration of cluster phenomenon: (a) single bounce; (b) double bounce.

height of the base-station antenna: If this antenna is below the dominant roof level, the environment is generally considered to be microcellular. Higher base-station antennas indicate macrocellular conditions. Typically, as in the COST 259 models described in [1], picocell base stations are indoors while those of micro- and macrocells are outdoors.

An important feature observed in both indoor and outdoor propagation is clustering (Fig. 3.2). A cluster consists of multipath groups that have similar delays and angles of azimuth and elevation at the receiver terminal and are shadowed simultaneously, but that cannot be resolved. Clusters may arise, for example, when MPCs are created by reflection from surfaces within a group of physical objects such as tall buildings. As discussed in [1], a group of objects close to the mobile terminal can also cause a cluster scatterer effect when illuminated by a specular reflection or line-of-sight (LOS) wavefront. This clustering behavior can be incorporated into the double-directional channel model, whereby the  $\ell$ th cluster has  $K_\ell$  MPCs with mean angle of departure (AOD)  $\Phi_{t,\ell}$  and mean angle of arrival (AOA)  $\Phi_{r,\ell}$  as

$$h(\tau, \phi_r, \phi_t) = \sum_{\ell=1}^L \sum_{k=1}^{K_\ell} a_{\ell,k} \delta(\tau - T_\ell - \tau_{\ell,k}) \delta(\phi_r - \Phi_{r,\ell} - \phi_{r,\ell,k}) \delta(\phi_t - \Phi_{t,\ell} - \phi_{t,\ell,k}). \quad (3.21)$$

The  $k$ th MPC within the  $\ell$ th cluster has azimuthal angles  $\phi_{t,\ell,k}$  and  $\phi_{r,\ell,k}$  with respect to the means  $\Phi_{t,\ell}$  and  $\Phi_{r,\ell}$ , respectively. The time of arrival of the first multipath component within cluster  $\ell$  is  $T_\ell$ , and the excess delays of the other multipath components within the same cluster are  $\tau_{\ell,k}$ .

For example, in the COST 273 model [10], there are different types of clusters, including local clusters around the mobile terminal and possibly also the base station that provide single-bounce scattering and multiple-bounce clusters. For the microcell implementation, the model uses seven MPCs per cluster, whereas for the macrocell and picocell implementations, there are 20 MPCs per cluster.

As noted in the previous section, multipath fading results when there is relative movement in the physical environment of the terminals and/or the interacting objects. For larger changes, for example, as the mobile drives over a distance of several city blocks, there are much larger effects observed in the channel response. These effects

include shadowing, which is caused by blockage of the signal power, for example, by a building, changes in path loss due to increasing or decreasing separation of the transmitter and receiver, and changes in the multipath structure. This last effect includes the appearance and disappearance of clusters, and changes in delay and angular properties of MPCs and clusters. For a MIMO system, shadowing and path loss can be modeled quite simply as they affect all antenna elements equally. The changing multipath structure is the most important effect for MIMO systems, as it alters the correlation properties of the MIMO channel response matrix,  $\mathbf{H}(t, \tau)$ , and can significantly impact system performance.

One of the drawbacks of the GSCM is the single-bounce assumption. As noted above, in many environments this assumption is not valid, and a single-bounce model is unable to reproduce the correct delay and directions at both terminals. This, in turn, impacts the properties of the channel response matrix  $\mathbf{H}(t, \tau)$ . Double scattering was incorporated into the COST 273 MIMO channel model [10, Chapter 6] and into a generic MIMO model by Molisch [26]. In the latter, an illumination function is proposed to account for the fact that not all scatterers in a cluster will reradiate toward all scatterers in another cluster.

Another propagation observed that is not modeled by the single-bounce assumption is waveguiding, for example, along urban canyons. Waves experience multiple reflections from each side of the street, resulting in delay dispersion and increased attenuation; this was modeled in [27]; similar conditions are seen in multiroom indoor picocells. In extreme cases, waveguiding could cause the keyhole effect in which the channel response matrix  $\mathbf{H}$  is rank deficient even though there is scattering near the transmitter and receiver. This is similar to the case where the distance between the terminals is much larger than the distance between each terminal and the scatterers clustered around them. The appearance of keyholes was hypothesized by Chizhik et al. in [28], and their impact on MIMO capacity was evaluated in [29]. Although it was suggested in [29] that diffraction over rooftops might cause keyholes, it was noted there and in [10, Chapter 6] that it is highly unlikely that this phenomenon will be observed in real environments. However, waveguiding does affect the structure of the MIMO channel response by changing the distribution of the eigenvalues, and it was incorporated into the generic MIMO model in [26] using a combination of geometric and stochastic modeling.

Movement of the terminal can be modeled using the GSCM by fixing the location of the interacting objects and computing the angular power delay profile for each position of the terminal. The COST 259 model allows movement only of the terminal and not the interacting objects; this is simpler to implement in simulation but is somewhat limiting in application.

Modeling the effect of clusters for a mobile terminal in the COST 259 model was described in [30]. Each cluster is defined to have a visibility region and a transition region. While the mobile is within the visibility region, for example, a city block, the cluster is considered active, that is, the MPCs resulting from the interacting objects contained within the cluster are included in the received signal. A smoothing function is introduced, defined over the transition region, which might be on the scale of the street width, to provide a gradual appearance of the MPCs as the mobile enters the cluster visibility region, and a gradual disappearance as the mobile leaves. No experimental validation of this proposed technique has been reported.

In analyzing and simulating channel characteristics or system performance, it is generally assumed that the time-varying propagation effects are wide-sense stationary. Large-scale variations including changes in the number, strength, or direction of MPCs, such as those incorporated in the COST 273 model, result in nonstationarity, therefore the channel response  $\mathbf{H}(t)$  can be assumed to be wide-sense stationary only over small areas of a few wavelengths. This will be addressed in more detail in Section 3.5.

**3.3.1.3 Stochastic Models** While the geometry-based stochastic channel models apply a statistical distribution to the location of the scatterers and cluster scatterers, other parametric models consider the statistics of the signal direction. These models are generated using measurements to determine the appropriate statistics, without explicitly considering the source of scattering. The Saleh–Valenzuela statistical model of the multipath power delay profiles for an indoor environment used a Poisson process to model the delays associated with each cluster, then the power of the first MPC within that cluster was computed using an exponential function of the cluster delay [31]. Powers were assigned to the remaining MPCs within the cluster using another, steeper exponential function. Modifications to this model to incorporate spatial statistics are generally referred to as extended Saleh–Valenzuela models (ESVMs). One such model was developed for macrocells in [32], where the angular spread of the clusters at the mobile terminal was characterized. The delay and angular distributions were modeled for indoor measurements by Spencer et al. in [33], where it was found that for the environments measured the mean cluster angle of arrival was uniformly distributed, while the angles within each cluster were approximately Laplacian distributed. A further extension in [34] incorporated both AOD and AOA statistics into the Saleh–Valenzuela model.

The parameterization of the ESVM reported in [33] and [34] assumes that the delay, AOA, and AOD are independent. Measurements reported in [35] suggest this assumption is reasonable for non-line-of-sight scenarios, but for line-of-sight environments, a strong relationship between AOA and delay was observed in an indoor environment at 5 GHz. The assumption of independence of the angles of departure and arrival indicates multiple bounces, in contrast to the single-bounce GSCM.

Mobility is modeled through the incorporation of fading into these ESVMs, but this is applicable only over small areas where the angular and delay properties of the MPCs change minimally. This was illustrated in [36], where the angular spectrum was extracted from measured data and applied by simulating the motion of the terminal. Large-scale variations, such as those discussed above, cannot readily be simulated using ESVMs.

### 3.3.2 Analytical Models

The computation of channel response matrices from physical channel models, taking into account array geometry, is described by (3.7). For many applications, such as evaluating the performance of space–time signaling and detection schemes, the cause of the channel’s spatial structure is of less interest than its impact, and an analytical model that can describe particular spatial correlation properties is more desirable. These provide the narrowband channel response matrix  $\mathbf{H}$  in (3.11) directly. Generally, these models incorporate not just the physical effects of the propagation environment but also the impact of the communications system itself, such as the antennas, amplifiers, filters and frequency converters.

Analytical models can be used to simulate the effects of Rayleigh fading by randomly generating different realizations of the channel matrix,  $\mathbf{H}$ , which have a common description such as autocorrelation. As with realizations generated using the ESVMs, these represent channel responses typical of those obtained over a small area in which the MPC characteristics change very little. Large-scale changes lead to changes in the base description, for example, a new autocorrelation matrix. The time-varying effects of the mobility are not incorporated into most of the models, even over small distances, except for the multivariate complex normal model.

When there is a Ricean component, the channel response matrix can be separated to give

$$\mathbf{H} = \frac{\sqrt{K}}{\sqrt{K+1}} \mathbf{H}_{\text{spec}} + \frac{1}{\sqrt{K+1}} \mathbf{H}_{\text{scat}}, \quad (3.22)$$

where  $K \geq 0$  is the Rice factor,  $\mathbf{H}_{\text{spec}}$  is the deterministic response due to the specular, possibly line-of-sight, component, and  $\mathbf{H}_{\text{scat}}$  is the Rayleigh-fading stochastic response due to the scattering.

The physical models discussed in Section 3.3.1 incorporate the relative delays of the multipath components and thereby provide wideband, or frequency-selective, characterizations. The analytical models described here, however, are narrowband or flat-fading characterizations. They can be readily extended to obtain the MIMO impulse responses using the tap-delay line structure

$$\mathbf{H}(\tau) = \sum_{\ell=1}^L \mathbf{H}_\ell \delta(\tau - \tau_\ell), \quad (3.23)$$

where  $\mathbf{H}_\ell$  is the matrix of complex channel responses at delay  $\tau_\ell$ . The mean square values of the elements of  $\mathbf{H}_\ell$  depend on the power delay profile being modeled. It is assumed, based on physical considerations, that  $\mathbf{H}_\ell$  and  $\mathbf{H}_k$  are uncorrelated for  $\ell \neq k$  [37], and they can be constructed using any of the narrowband analytical models described here.

The first analytical models were very simple, based on the assumption that the signal fading observed at two antenna elements is independent if those elements are at least half a wavelength apart. This assumption follows from the isotropic scattering model [38, Chapter 1]. The resulting channel response matrix has independent zero-mean complex Gaussian distributed entries,  $h_{k,m}$ . With this simple model, the potential of MIMO systems to achieve huge capacity gains was observed [4, 5].

In practice, the assumptions that there are an infinite number of scatterers and that they are uniformly distributed on  $[0, 2\pi)$  do not hold, resulting in correlation of the path gains and a reduction in the system capacity [39]. For an  $N_r \times N_t$  channel response matrix  $\mathbf{H} = [\mathbf{h}_1 \ \mathbf{h}_2 \ \dots \ \mathbf{h}_{N_t}]$ , the full correlation matrix, which contains the correlation values for all the transmitter/receiver element pairs, is

$$\mathbf{R}_{\mathbf{H}} = \mathcal{E} \left\{ \text{vec}(\mathbf{H}) \text{vec}(\mathbf{H})^H \right\}, \quad (3.24)$$

where  $\text{vec}(\mathbf{H}) = [\mathbf{h}_1^T \ \dots \ \mathbf{h}_{N_t}^T]^T$ . A channel response matrix with this correlation matrix can be obtained from

$$\mathbf{H} = \text{unvec} \left( \mathbf{R}_{\mathbf{H}}^{1/2} \mathbf{g} \right), \quad (3.25)$$

where the elements of the  $N_r \cdot N_t \times 1$  vector  $\mathbf{g}$  are independent and identically distributed (i.i.d.) zero-mean complex Gaussian with unit variance, and  $\text{unvec}\left(\begin{bmatrix} \mathbf{h}_1^T & \cdots & \mathbf{h}_{N_t}^T \end{bmatrix}^T\right) = \mathbf{H}$ . The matrix square root  $\mathbf{R}_H^{1/2}$  is any  $N_t \cdot N_r \times N_t \cdot N_r$  matrix satisfying  $\mathbf{R}_H^{1/2}(\mathbf{R}_H^{1/2})^H = \mathbf{R}_H$ .

**3.3.2.1 Kronecker Model** A total of  $(N_t \cdot N_r)^2$  terms are required to specify  $\mathbf{R}_H$ ; therefore, a large number of independent measurement samples is necessary to obtain an accurate estimate. A lower complexity model was developed in [39], based on the spatial correlation matrices at the transmitter and receiver, which are defined as, respectively,

$$\mathbf{R}_t = \mathcal{E}\{\mathbf{H}^H \mathbf{H}\} \quad \text{and} \quad \mathbf{R}_r = \mathcal{E}\{\mathbf{H} \mathbf{H}^H\}. \quad (3.26)$$

When the spatial correlation at the transmitter is assumed to be independent from that at the receiver, these correlation matrices are related by

$$\mathbf{R}_H = \frac{1}{\text{tr}\{\mathbf{R}_r\}} \mathbf{R}_t \otimes \mathbf{R}_r, \quad (3.27)$$

where  $\otimes$  denotes the Kronecker product.<sup>1</sup> This condition gives its name to the Kronecker model. As shown in [40], realizations of  $\mathbf{H}$  with the second-order statistics given by  $\mathbf{R}_H$  are obtained using

$$\mathbf{H} = \frac{1}{\sqrt{\text{tr}\{\mathbf{R}_r\}}} \mathbf{R}_r^{1/2} \mathbf{G} \mathbf{R}_t^{1/2}, \quad (3.28)$$

where the elements of the  $N_r \times N_t$  matrix  $\mathbf{G}$  are i.i.d. zero-mean complex Gaussian with unit variance. The Kronecker model has similar limitations to the stochastic models in Section 3.3.1, as it is also based on the assumption that the correlation properties at the receiver are separable from those at the transmitter. The model therefore cannot represent the case where an MPC or cluster at the transmitter is coupled into the receiver, that is, like the ESVM, it does not model single-bounce geometry.

**3.3.2.2 Finite Scatterer Model** A physically based analytical model can be obtained using the double-directional channel model, from which the narrowband channel response matrix  $\mathbf{H}$  can be computed as in (3.7). This leads to the finite scatterer model (FSM) introduced by Burr in [41] in which each scatterer, or scatterer cluster (3.21), is identified by its location, that is,

$$\mathbf{H} = \sum_{\ell=1}^L a_\ell \psi_r(\phi_{r,\ell}) \psi_t^T(\phi_{t,\ell}) = \boldsymbol{\Psi}_r \mathbf{A} \boldsymbol{\Psi}_t^T, \quad (3.29)$$

<sup>1</sup>The Kronecker product of the  $N \times N$  matrix  $\mathbf{A}$  and the  $M \times M$  matrix  $\mathbf{B}$  is given by the  $NM \times NM$  matrix

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} A_{1,1}\mathbf{B} & A_{1,2}\mathbf{B} & \cdots \\ A_{2,1}\mathbf{B} & A_{2,2}\mathbf{B} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

where  $A_{i,j}$  is the  $(i, j)$ th element of  $\mathbf{A}$ .

where the columns of  $\Psi_t = [\psi_t(\phi_{t,1}) \dots \psi_t(\phi_{t,L})]$  and  $\Psi_r = [\psi_r(\phi_{r,1}) \dots \psi_r(\phi_{r,L})]$  are given by (3.5) and (3.6), respectively. The complex amplitudes of the angular components are given by the diagonal elements of  $\mathbf{A} = \text{diag}(a_1, \dots, a_L)$ . This model can be used to represent single- and multiple-bounce scattering, specifying the distribution of scatterers as for the physical models described above.

**3.3.2.3 Virtual Channel Model** Practical array sizes, in terms of aperture and number of elements, limit the resolvability of the multipath components; therefore, the angles  $\phi_{t,\ell}$  and  $\phi_{r,\ell}$  in the FSM are difficult to measure accurately. The virtual channel representation (VCR) [42] model applies beam-steering vectors at the transmitter and receiver arrays, thereby combining the effects of all scatterers within a given beam. The application of beam-steering limits its application to cases where the arrays at the transmitter and receiver consist of single polarized elements, linearly arranged with uniform spacing.

At the transmitter array, the steering vector along the direction  $\phi_{t,m}$  is given by (3.5), that is,

$$\psi_t(\phi_{t,m}) = \frac{1}{\sqrt{N_t}} [1 \quad \exp(-j2\pi\frac{d}{\lambda} \sin \phi_{t,m}) \quad \dots \quad \exp[-j2\pi(N_t - 1)\frac{d}{\lambda} \sin \phi_{t,m}]]^T, \quad (3.30)$$

where  $\lambda$  is the signal wavelength and  $d$  is the array element spacing. If the values of  $(d/\lambda) \sin \phi_{t,m}$  are uniformly spaced, then the matrix  $\Psi_t = [\psi_t(\phi_{t,1}) \dots \psi_t(\phi_{t,N_t})]$  is the  $N_t \times N_t$  unitary discrete Fourier transform matrix,  $\mathbf{F}_{N_t}$ . In a similar way, the  $N_r \times N_r$  matrix  $\Psi_r = \mathbf{F}_{N_r}$  is defined for the receiver array using (3.6). The signal power coupled from the transmitter beam  $\psi_t(\phi_{t,m})$  into the receiver beam  $\psi_r(\phi_{r,n})$  through various scattering mechanisms is given by  $|\omega_{n,m}|^2$ . The VCR model is then written

$$\mathbf{H} = \mathbf{F}_{N_r} (\boldsymbol{\Omega}_v \odot \mathbf{G}) \mathbf{F}_{N_t}^H, \quad (3.31)$$

where  $\boldsymbol{\Omega}_v$  is the real  $N_r \times N_t$  coupling matrix with  $(n, m)$ th element  $\omega_{n,m}$ . As before, the elements of the  $N_r \times N_t$  matrix  $\mathbf{G}$  are i.i.d. zero-mean complex Gaussian with unit variance, and  $\odot$  is the Schur–Hadamard product.<sup>2</sup>

**3.3.2.4 Weichselberger Model** The Weichselberger model [43] removes the array structure limitations of the VCR model by replacing the beam-steering matrices with eigenbases of the spatial correlation matrices,  $\mathbf{R}_t$  and  $\mathbf{R}_r$ . The eigendecompositions of these matrices are

$$\mathbf{R}_t = \mathbf{U}_t \boldsymbol{\Lambda}_t \mathbf{U}_t^H \quad \text{and} \quad \mathbf{R}_r = \mathbf{U}_r \boldsymbol{\Lambda}_r \mathbf{U}_r^H, \quad (3.32)$$

<sup>2</sup>The Schur–Hadamard product of  $N \times M$  matrices  $\mathbf{A}$  and  $\mathbf{B}$  is given by the  $N \times M$  matrix

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} A_{1,1}B_{1,1} & A_{1,2}B_{1,2} & \cdots \\ A_{2,1}B_{2,1} & A_{2,2}B_{2,2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

where  $A_{i,j}$  is the  $(i, j)$ th element of  $\mathbf{A}$ .

where  $\mathbf{U}_t$  and  $\mathbf{U}_r$  are unitary matrices of eigenvectors at the transmitter and receiver, respectively, and  $\Lambda_t$  and  $\Lambda_r$  are diagonal matrices of the corresponding eigenvalues.

The Weichselberger model is then given by

$$\mathbf{H} = \mathbf{U}_r (\boldsymbol{\Omega}_w \odot \mathbf{G}) \mathbf{U}_t^H, \quad (3.33)$$

where, as in the Kronecker and VCR models,  $\mathbf{G}$  is an  $N_r \times N_t$  matrix with independent elements that have zero-mean, unit-variance complex Gaussian distributions. The real  $N_r \times N_t$  coupling matrix  $\boldsymbol{\Omega}_w$  describes the coupling between the eigenbases at the transmitter and receiver.

The pattern of significant nonzero elements within  $\boldsymbol{\Omega}_w$  gives insight into the spatial structure of the scattering objects. For example, a single nonzero element in  $\boldsymbol{\Omega}_w$  indicates there is only one resolvable multipath component. If all the nonzero elements in  $\boldsymbol{\Omega}_w$  form a column, this suggests that all the scatterers are located near the receiver and their angular spread at the transmitter is small. Similarly, a single nonzero row in  $\boldsymbol{\Omega}_w$  would suggest the scatterers are located close to the transmitter. A rank-one  $\boldsymbol{\Omega}_w$  is equivalent to the Kronecker model; while if  $\boldsymbol{\Omega}_w$  is the all-ones matrix, the spatial structure can be considered as rich isotropic scattering, leading to the i.i.d. complex Gaussian model.

This narrowband model was extended in [44] to the wideband channel by extending the correlation matrices to incorporate the delay domain, and thereby model correlation in the three dimensions of space, time, and delay.

**3.3.2.5 MVCN Model** A time-varying analytical model, the multivariate complex normal (MVCN) model, was introduced by Wallace and Jensen in [36]. Extending (4.24), the space–time correlation matrix is

$$\mathbf{R}_{\mathbf{H}}(n, m) = \mathcal{E} \left\{ \text{vec}(\mathbf{H}(nT_s)) \text{vec}(\mathbf{H}(nT_s + mT_s))^H \right\}, \quad (3.34)$$

where  $T_s$  is the sampling interval. To limit the computational complexity, it is assumed that the space–time correlation matrix is separable into space and time, that is, that the  $(ij, k\ell)$ th element of  $\mathbf{R}_{\mathbf{H}}(n, m)$  can be written as

$$R_{\mathbf{H},ij,k\ell}(n, m) = R_{S,ij,k\ell}(n) R_T(n, m), \quad (3.35)$$

where  $\mathbf{R}_S(n)$  is averaged over all time delays, and  $R_T(n, m)$  is averaged over all antenna pairs, that is,

$$\begin{aligned} R_{S,ij,k\ell}(n) &= \sum_m h_{i,j}(nT_s) h_{k,\ell}^*(nT_s + mT_s) \quad \text{and} \\ R_T(n, m) &= \sum_{i,j,k,\ell} h_{i,j}(nT_s) h_{k,\ell}^*(nT_s + mT_s). \end{aligned} \quad (3.36)$$

Now, defining the matrix  $\tilde{\mathbf{R}}_T$  such that the  $(i, j)$ th element is  $R_T(i, j - i)$ , the time-varying channel matrix elements are simulated as

$$h_{i,j}(nT_s) = \sum_{k,\ell} \sum_m X_{S,ij,k\ell}(n) X_{T,n,m} G_{ij}(m), \quad (3.37)$$

where  $\mathbf{X}_T = \tilde{\mathbf{R}}_T^{1/2}$ ,  $\mathbf{X}_S = \mathbf{R}_S^{1/2}$ , and the random variables  $G_{ij}(n)$  are i.i.d. complex Gaussian distributed. This model was parameterized in [36] using measured data from an indoor scenario; the model provided reasonable accuracy over a range of approximately 2–15 wavelengths. The limitations in the model accuracy over shorter distances was attributed to the assumption of space–time separability, which is not likely to be valid in this region. Over larger distances, the spatial structure of  $\mathbf{H}(t)$  changes significantly, such that averaging over time delays to generate  $\mathbf{R}_S(n)$  loses validity.

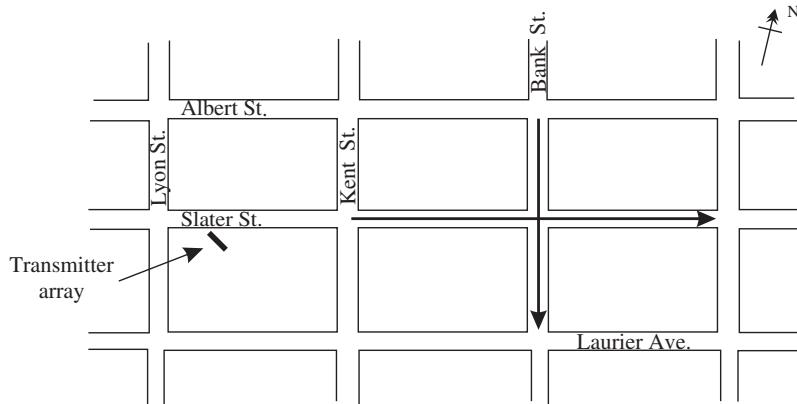
### 3.4 MEASURED CHANNEL CHARACTERISTICS

The models described in the previous section provide convenient descriptions of the MIMO radio channel that can be reliably reproduced. It is important to recognize that models give only partial characterizations and that they will not fully describe real scenarios experienced by MIMO systems. For evaluating expected performance, therefore, the model parameters should be selected to typify conditions that are as realistic as possible. The parameters of the physical channel models (Section 3.3.1) can be manipulated to provide a wide range of channel conditions, and large collaborative measurement campaigns have led to typical characterizations suitable for different environments. In this section, measured data are used to parameterize some of the analytical models discussed in Section 3.3.2 to demonstrate the range of characteristics that can be observed over a distance of several hundred wavelengths.

The MIMO channel measurements presented in this section were obtained in an urban microcellular environment, using the MIMO sounder developed at the Communications Research Centre Canada [45]. The sounder has eight antenna elements at both the transmitter and receiver, which were configured for these measurements in uniform linear arrays with spacing of one-half wavelength at the carrier frequency of approximately 2 GHz. The sounding signals were BPSK pseudonoise sequences with a bandwidth 25 MHz, emitted simultaneously from each of the elements in the array. The signals received at each element of the receiver array were downconverted and sampled at 50 Msamp/s. The signals were processed off-line to extract the complex baseband impulse response of each of the transmitter–receiver pair links. The channel responses were obtained at a rate of 250/s, which is more than twice the maximum Doppler frequency, yielding  $N_t \cdot N_r = 64$  impulse responses every 4 ms. The impulse responses were fast Fourier transformed, and the complex coefficients corresponding to a single frequency were extracted to obtain narrowband channel response matrices  $\tilde{\mathbf{H}}(k)$ , at instants  $kT_s$  for  $T_s = 4$  ms, that can be used in the system model (3.11).

Data series were obtained in downtown Ottawa, with the transmitter located in a window on the fourth floor that was oriented at  $45^\circ$  immediately adjacent to the street. The antenna elements at the transmitter and receiver were quarter-wave monopoles. The receiver was in a van with the antenna elements mounted on the roof. The van was driven at approximately 30 km/h through light weekend traffic, along the routes indicated in Figure 3.3.

As seen from Figure 3.3, each measurement route is about two blocks long, during which time the receiver terminal is not only driven through an intersection but also past buildings with varying frontages, other vehicles, alleys between buildings, and



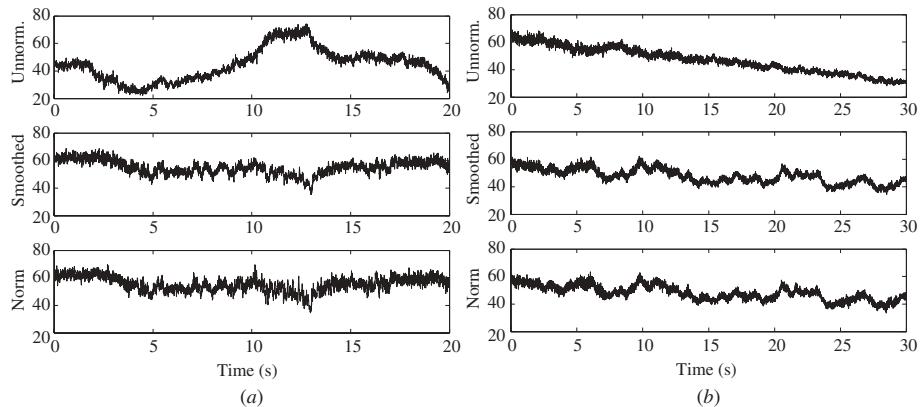
**Figure 3.3** Measurement routes in downtown Ottawa, Canada.

the like. All these large-scale features in the propagation environment cause noticeable changes in the MIMO channel response matrix.

Large-scale changes in the propagation environment pose a challenge for characterizing the channel. When the measured channel response matrices  $\mathbf{H}(k)$  are used in (3.11), some normalization is required in order to set the variance of the noise vector  $\mathbf{n}$  to give the desired SNR. The normalization used depends on the characteristics sought: the impact of different normalization strategies on the estimated channel capacity is illustrated in Figure 3.4 for the two measurement routes.

The top panel in Figure 3.4 shows the capacity (as before, capacity is given per unit bandwidth) of the “unnormalized” channel responses at 30 dB computed using (3.13)—in this case, the power is averaged across the whole measurement run to provide the desired average SNR.

In the second panel, the received power has been normalized using a running window before computing the capacity, again at 30 dB. The rectangular normalization



**Figure 3.4** Measured capacity for unnormalized, smoothed, and normalized  $8 \times 8$  channel response matrices at 30 dB: (a) Bank St.; (b) Slater St.

window was 80 ms long, giving the smoothed channel response matrix:

$$\mathbf{H}_{\text{smooth}}(k) = \frac{\tilde{\mathbf{H}}(k)}{\left[ \frac{1}{N_r N_t K} \sum_{n=k-K/2+1}^{k+K/2} \|\tilde{\mathbf{H}}(n)\|_F^2 \right]^{1/2}}, \quad (3.38)$$

where  $\|\mathbf{H}\|_F = \left[ \sum_{i=1}^{N_r} \sum_{j=1}^{N_t} |H_{ij}|^2 \right]^{1/2}$  is the Frobenius norm,  $H_{ij}$  is the  $(i, j)$ th element of  $\mathbf{H}$ , and  $K = 20$ .

The bottom panels in Figure 3.4 show the capacity when each channel response matrix is normalized to give

$$\mathbf{H}_{\text{norm}}(k) = \frac{\tilde{\mathbf{H}}(k)}{\left[ \frac{1}{N_r N_t} \|\tilde{\mathbf{H}}(k)\|_F^2 \right]^{1/2}}. \quad (3.39)$$

This removes the impact of power levels from the capacity computation altogether, so the remaining variations are due solely to the changing spatial diversity. The normalized capacity changes rapidly over short periods of time, indicating that the features in the environment that impact the channel's spatial structure have a very localized effect.

In Figure 3.4a, the effect of passing through the intersection of Bank St. at Slater St. can clearly be seen in the region 11–13 s, where a strong signal is received that has propagated from the transmitter location along Slater St. This directional signal increases the received SNR, thereby increasing the computed capacity. However, as seen in the lower two panels, when the received power is normalized over short intervals or sample by sample, the computed capacity decreases in this region. This is because the signal component arriving along Slater St. is highly directional and dominates the scattered components from other directions, reducing the effective spatial diversity. The impact of the power increase thus counters the loss in spatial diversity, and the selection of the normalization interval may favor one or the other.

The propagation along Slater St., while non-line-of-sight, is highly directional. The received signal is dominated by a small number of dominant multipath components that arrive from the small angular range defined by the street canyon. The top panel in Figure 3.4b shows the dominating effect of path loss on the capacity with no power normalization. However, it is clear from the bottom panel that there are marked local changes in the spatial structure of the multipath components, which are obscured by the path loss in the top panel.

### 3.4.1 Analytical Model Parameterization

It was seen above that the capacity of the channel varies as the receiver moves along the street, resulting from changes in the spatial correlation as well as the total received power. When channel models such as those discussed in Section 3.3 are applied to assess the performance of communications systems, the range of parameters selected should reflect these different characteristics. Physical channel models, by their nature, incorporate information about the significant features of the propagation environment.

The choice of parameters for the analytical models is less clear; the range of parameters required to describe the MIMO channel over a small area, a one-block radius, will be illustrated here.

The MIMO data measured on Bank and Slater Streets in Ottawa have been used to parameterize the Kronecker, VCR, and Weichselberger models. The channel response matrices were normalized using (3.39). As in [46], no attempt has been made to separate the Ricean and Rayleigh fading components.

The spatial correlation matrices used to fit the Kronecker model are estimated using  $K = 40$  consecutive measurement samples:

$$\hat{\mathbf{R}}_t = \frac{1}{K} \sum_{k=1}^K \mathbf{H}^H(k) \mathbf{H}(k) \quad \text{and} \quad \hat{\mathbf{R}}_r = \frac{1}{K} \sum_{k=1}^K \mathbf{H}(k) \mathbf{H}^H(k). \quad (3.40)$$

These are applied to (3.28), where multiple realizations are obtained using different i.i.d. complex Gaussian matrices  $\mathbf{G}$ .

The estimated coupling matrix for the VCR model is computed using

$$\Omega_v = \left[ \frac{1}{K} \sum_{k=1}^K (\mathbf{F}_{N_r}^H \mathbf{H}(k) \mathbf{F}_{N_t}^*) \odot (\mathbf{F}_{N_r}^T \mathbf{H}^*(k) \mathbf{F}_{N_t}) \right]^{1/2}, \quad (3.41)$$

where  $\mathbf{F}_{N_t}$  and  $\mathbf{F}_{N_r}$  are the  $N_t \times N_t$  and  $N_r \times N_r$  unitary discrete Fourier transform matrices, respectively, with columns given by the steering vector in (3.30), where  $(d/\lambda) \sin \phi_{t,j} = (j - 1)/N_t - 0.5$ ,  $j = 0, \dots, N_t - 1$ , for the  $j$ th column of  $\mathbf{F}_{N_t}$ , and  $(d/\lambda) \sin \phi_{r,i} = (i - 1)/N_r - 0.5$ ,  $i = 0, \dots, N_r - 1$ , for the  $i$ th column of  $\mathbf{F}_{N_r}$ . Thus, the  $(i, j)$ th element of  $\Omega_v$  is

$$\Omega_{v,ij} = \left[ \frac{1}{K} \sum_{k=1}^K |\psi_r^T(\phi_{r,i}) \mathbf{H} \psi_t(\phi_{t,j})|^2 \right]^{1/2}, \quad (3.42)$$

where  $\psi_r(\phi_{r,i})$  and  $\psi_t(\phi_{t,j})$  are the  $i$ th and  $j$ th columns of  $\mathbf{F}_{N_r}$  and  $\mathbf{F}_{N_t}$ , respectively.

The Weichselberger coupling matrix was computed similarly to the VCR coupling matrix, but in this case the Fourier transform matrices were replaced by the estimated eigenvector matrices,  $\hat{\mathbf{U}}_t$  and  $\hat{\mathbf{U}}_r$ , found from the spatial correlation matrix estimates

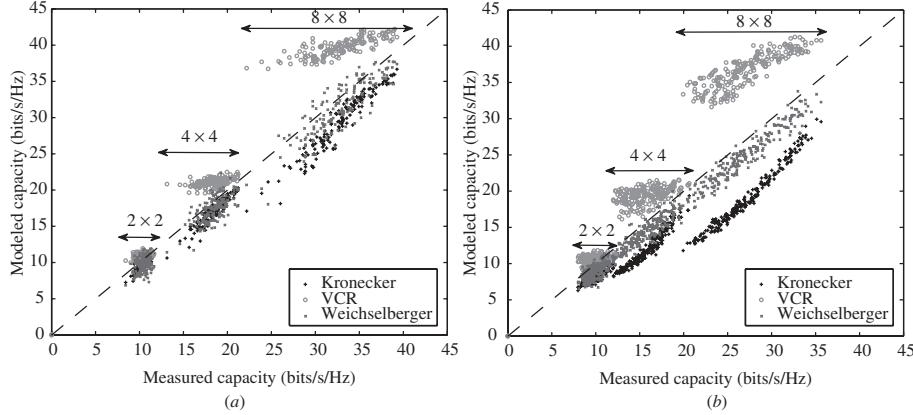
$$\hat{\mathbf{R}}_t = \hat{\mathbf{U}}_t \hat{\Lambda}_t \hat{\mathbf{U}}_t^H \quad \text{and} \quad \hat{\mathbf{R}}_r = \hat{\mathbf{U}}_r \hat{\Lambda}_r \hat{\mathbf{U}}_r^H. \quad (3.43)$$

Then

$$\Omega_w = \left[ \frac{1}{K} \sum_{k=1}^K (\hat{\mathbf{U}}_r^H \mathbf{H}(k) \hat{\mathbf{U}}_t) \odot (\hat{\mathbf{U}}_r^T \mathbf{H}^*(k) \hat{\mathbf{U}}_t^*) \right]^{1/2}. \quad (3.44)$$

As with the Kronecker model, for both the VCR and Weichselberger models, different realizations representing the effects of Rayleigh fading are obtained by using i.i.d. complex Gaussian matrices  $\mathbf{G}$  in (3.31) and (3.33), respectively.

The capacity was computed using (3.13) for each of the  $K$ -normalized measured channel responses and averaged to obtain the mean measured capacity. The corresponding parameterized models were used to generate  $K$ -simulated channel responses, and



**Figure 3.5** Average capacity of simulated vs. measured channel responses at 20 dB: (a) Bank St.; (b) Slater St.

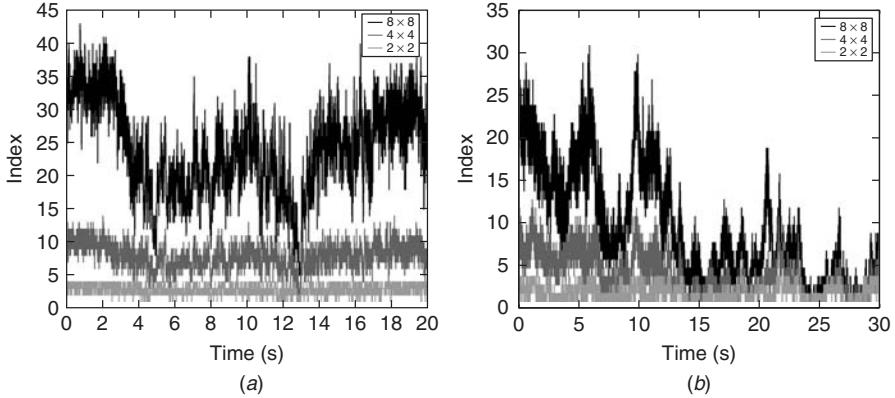
the mean capacity of those was also computed. Figure 3.5 shows the simulated versus measured average capacities for locations along the two measurement routes, using  $8 \times 8$ ,  $4 \times 4$ , and  $2 \times 2$  MIMO configurations, each using half-wavelength antenna element spacing at an SNR of 20 dB. Clearly, the Weichselberger model provides the best emulation of average capacity. The Kronecker model channel realizations consistently yield low capacities, particularly on Slater St. The correlation on this street is generally higher than on Bank St., so these results are consistent with earlier ones showing that the Kronecker model does not reproduce correlation accurately [47]. The VCR model also provides low accuracy in the mean capacity, resulting primarily from the poor resolution obtained from the small array size and the fixed steering vectors, as MPCs falling between the steering directions are not represented adequately in the model.

Capacity is only one metric to evaluate the quality of a model. Other metrics, including multipath richness, diversity, and angular power spectra, have been proposed, for example, by Özcelik et al. in [46]. As the analytical model is not reproducing the actual structure of the physical channel, its accuracy must be considered in terms of the purpose of the model.

### 3.4.2 Temporal Variations

As noted in Section 3.3.2, most analytical models do not incorporate time-varying characteristics. The MVCN model introduced by [36] applies over short distances, as discussed in Section 3.3.2, for which the spatial characteristics of the channel remain fairly constant. To evaluate MIMO systems adequately for mobile applications, a range of model parameters reflecting the impact of the large-scale variations encountered in a typical operational scenario should be considered. For this purpose, the MIMO data measured on Bank and Slater Streets in Ottawa were used to parameterize the VCR and Weichselberger models. The channel response matrices were normalized, as in (3.39), to focus on the channel's spatial properties. Overlapping segments of length  $K = 40$  samples were used; this length corresponds to a quasi-stationary interval, as in [48].

As an indicator of the changing characteristics, the number of significant coupling modes within the Weichselberger model was evaluated for each segment. This was



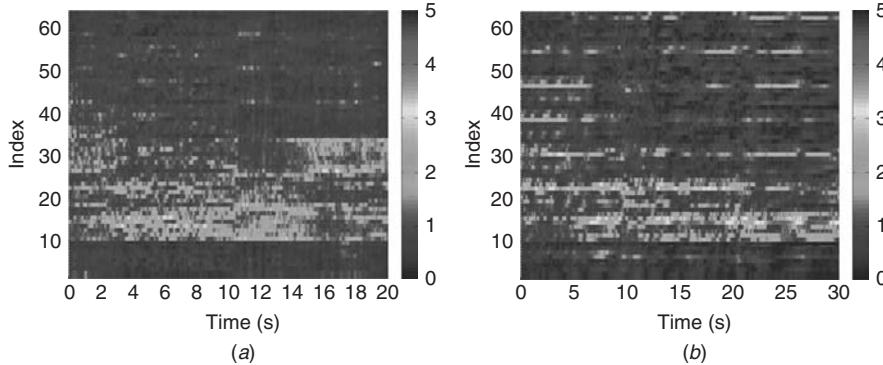
**Figure 3.6** Number of significant coupling modes in the Weichselberger parameterization for measured data: (a) Bank St.; (b) Slater St.

determined by counting the number of elements in  $\Omega_w$ , computed using (3.44), providing 95% of the total power transmission, where power is measured as the squared magnitudes of the elements of  $\Omega_w$ . This is plotted in Figure 3.6 for  $8 \times 8$ ,  $4 \times 4$ , and  $2 \times 2$  channel response matrices, each for half-wavelength antenna element spacings.

On Bank St., the variation in the number of modes shown in Figure 3.6a approximately follows the normalized capacity, shown in the bottom panel of Figure 3.4a. The variation is more marked as the array size increases, as the resolution provided by the spatial filters  $\hat{\mathbf{U}}_t$  and  $\hat{\mathbf{U}}_r$  increases. In the range of 4–8 s, the signal energy is received mainly through a single transmit eigenmode, which suggests that the dominant propagation mechanism is diffraction at the corner into Bank St. and that the scattering occurs mainly near the receiver terminal. This is observed in Figure 3.6 as a small number of significant coupling modes. In the intersection region, 11–13 s, the coupling is often dominated by a single coupling mode due to the near line-of-sight path, although there is considerable variability due to local scattering at the transmitter and receiver. In the urban canyons, farthest from the intersection with Slater St., there is coupling among multiple transmitter and receiver modes.

On Slater St. (Fig. 3.6b), a marked change in the type of coupling observed occurs around 14 s, at which point the receiver terminal is approaching the intersection with Bank St. Before that point, the coupling occurs between multiple transmit and receive modes, indicating scattering at the transmitter and receiver. In the last 18 s, the observations are dominated by a very small number of coupling modes. At this point, there is some local scattering near the receiver, but the wave-guiding effects along the street canyon causes the attenuation of these multipath components to be quite high, resulting in only one or two significant coupling modes.

Although it was seen that the VCR model does not tend to fit the measured data as well as the Weichselberger model, in terms of the average capacity, it does provide insight into the direction of the signal components, and is therefore a useful tool to investigate the time-varying behavior of the MIMO channel. The elements of the  $8 \times 8$  coupling matrix have been computed from the measured data using (3.41), again using overlapping data segments of length  $K = 40$  samples. Figure 3.7 shows  $\text{vec}(\Omega_v)$ , which is the length 64 vector containing the stacked columns of  $\Omega_v$ . This arrangement means that groups of  $N_r = 8$  adjacent blocks in the figure couple from the same transmit



**Figure 3.7** Parameters of virtual channel representation for measured data: (a) Bank St.; (b) Slater St.

steering vector. Recall from (3.30) that the steering vectors do not represent uniformly spaced directions of arrival, rather, the angles corresponding to the steering vectors are closer together near the perpendicular to the array.

Figure 3.7a shows the model parameterization for Bank St. In the first 2 s, the signal energy is received through all the receive steering vectors indicating scattering all around the receiver, but as the mobile moves toward the intersection two or three receive steering vectors begin to dominate, which supports the hypothesis of diffraction into the urban canyon being the prime propagating mechanism. In the intersection, in the region of 10–13 s, the signal energy is now coupled through several transmit steering vectors, into a single receive steering vector, indicating scattering near the transmitter but not the receiver. One of the limitations of the VCR model is seen in this region: The overall coupling is quite weak even though the total receive power is relatively high, as was seen in Figure 3.4. This is because MPCs arriving from the end of the array, perpendicular to the motion of the vehicle, are not within the beams of any of the steering vectors.

On Slater St. (Fig. 3.7b), in the first 5 s the coupling occurs mainly into one of the receive steering vectors from several transmit vectors, indicating that there is scattering mainly near the transmitter. As the vehicle moves farther down the street, the impact of the scattering is reduced and the coupling is primarily from two transmit steering vectors. In the last 10 s, beyond the intersection with Bank St., the coupling is primarily through single transmit and receiver steering vectors.

These parameterizations over the two measurement runs show the type of variation that can be experienced by a MIMO system in a mobile urban environment. Over a relatively short interval, the characteristics of the channel can change quite dramatically. This may necessitate adaptive MIMO signaling strategies to maximize the potential of the spatial channel, and illustrates the importance of evaluating MIMO technologies over a wide range of channel model parameterizations.

### 3.5 STATIONARITY

It is clear from the capacity and channel parameterization results shown previously that the channel structure changes as the receiver moves. This suggests that the statistics of

the channel response matrix change also with time or distance, which means that the data series will be nonstationary. Channel models such as the ring models and analytical models described in Section 3.3 implicitly assume at least wide-sense stationarity, meaning that the mean and covariance do not vary with time. Despite the importance of this assumption, it is rarely tested in a rigorous way. A framework for identifying wide-sense stationary (WSS) segments in measured data was introduced in [48] for narrowband channel responses, based on Priestley's evolutionary spectra [49], and is outlined here.

For  $N_r \times N_t$  MIMO channels, the WSS condition requires that the  $N_r \cdot N_t$  time series of channel responses  $\{h_{n,m}(k)\}$  are jointly WSS. Two time series  $\{x_i(k)\}$  and  $\{x_j(k)\}$ ,  $k = 1, \dots, L$ , are jointly WSS if both  $\{x_i(k)\}$  and  $\{x_j(k)\}$  are WSS, which requires that their means and autocorrelation functions do not change with time and that their cross-correlation function is independent of time. To test the joint WSS hypothesis, the vector time series  $\{\mathbf{x}(k)\}$  is broken into  $N$  segments of length  $K$ , and the means, autocorrelation, and cross-correlation functions of each are computed and tested to determine whether the differences are statistically significant. Testing the second moments requires knowledge of the distributions of the sample autocorrelation and cross-correlation functions to establish how much difference is significant. This is difficult as the correlation functions are not known a priori: A more practical approach is to use estimates of the power spectral density (PSD) and cross-spectral density (CSD), which are the Fourier transforms of the autocorrelation and cross-correlation functions.

As the statistics of  $\{x(k)\}$  are time varying, the PSD and CSD cannot be computed exactly, but estimates can be obtained by smoothing in time and frequency—these are the evolutionary spectrum (ES) and evolutionary cross-spectrum (ECS) estimates. The windowed short-term Fourier transform is first computed for  $\{x_i(k)\}$  as

$$U_i(k, \omega) = \sum_{u=-\infty}^{\infty} g(u) x_i(k-u) e^{-j\omega(k-u)T_s} \quad (3.45)$$

for frequencies  $\omega \in [-\pi, \pi]$ ; then the PSD estimate  $|U_i(k, \omega)|^2$  is approximately unbiased. This estimate needs to be smoothed to reduce the variance, giving the nearly unbiased ES estimate over the neighborhood of  $k$

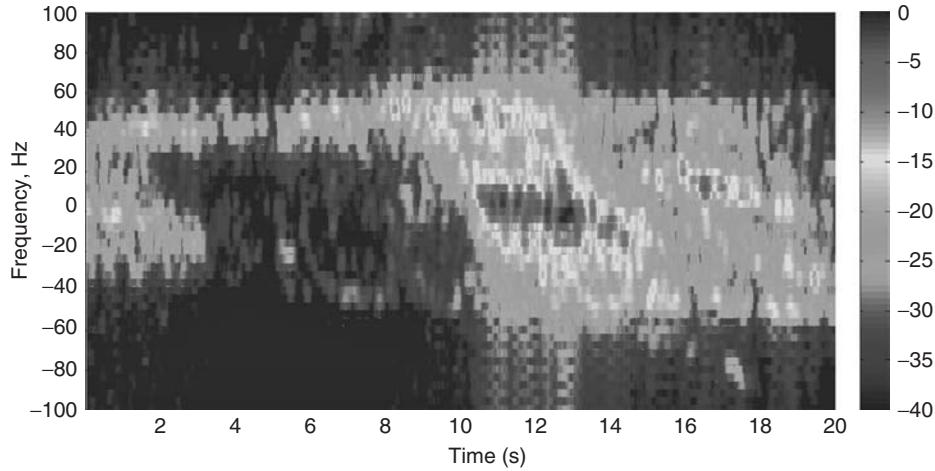
$$\hat{S}_{ii}(k, \omega) = \sum_{v=-\infty}^{\infty} w(v) |U_i(k-v, \omega)|^2. \quad (3.46)$$

The smoothed estimate of the ECS between  $\{x_i(k)\}$  and  $\{x_j(k)\}$  in the same region is

$$\hat{S}_{ij}(k, \omega) = \sum_{v=-\infty}^{\infty} w(v) U_i(k-v, \omega) U_j^*(k-v, \omega). \quad (3.47)$$

The frequency-domain smoothing sequence selected was

$$g(u) = \begin{cases} \frac{1}{2\sqrt{h\pi}}, & |u| \leq h, \\ 0, & |u| > h, \end{cases} \quad (3.48)$$



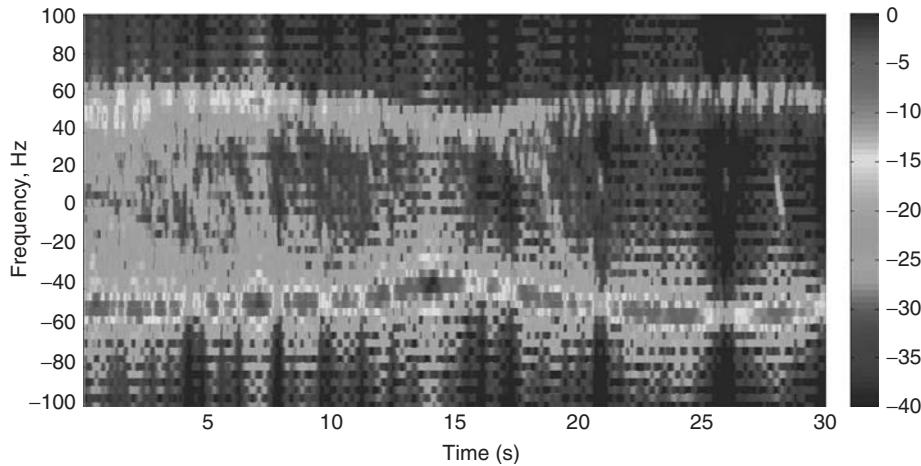
**Figure 3.8** Estimated evolutionary spectrum on Bank St.

where the frequency resolution is proportional to  $1/h$ , and  $h = 13$ . For the time-domain smoothing, the sequence used was

$$w(v) = \begin{cases} \frac{1}{K}, & |v| \leq \frac{K}{2}, \\ 0, & \text{otherwise,} \end{cases} \quad (3.49)$$

which operates over a short interval of  $K$  samples that can be considered to be stationary; in this case  $K = 40$ . The rationale behind the choice of these parameters can be found in [48].

The result of this time- and frequency-domain smoothing is shown in Figure 3.8 for the route along Bank St. and in Figure 3.9 for Slater St. Over each segment of



**Figure 3.9** Estimated evolutionary spectrum on Slater St.

$K + h + 1 = 54$  samples, or 0.216 s, the velocity of the vehicle can be treated as approximately constant, denoted  $r$ . The  $K + h + 1$  samples used in each ES estimate are then collected at locations spaced by  $r T_s$ ; this is equivalent to using a uniform linear array. For a velocity of 30 km/h, the normalized array element spacing is  $d/\lambda = r T_s \approx 0.22$ . Comparing (3.45) with (3.30), it is seen that the exponential terms in (3.45) form a steering vector along the direction given by  $\sin \phi_r = (1/r)(\omega/2\pi)$ , where  $\phi_r = 0$  is perpendicular to the direction of motion, and  $\omega/2\pi$  is the Doppler frequency shown along the vertical axis in Figures 3.8 and 3.9. The assumption that each segment is WSS is then equivalent to having the scattering objects in the far field of the virtual array, that is, that the MPC angles and delays do not change significantly over the interval being considered.

Thus, the changes in the estimated power spectrum reflect the changes in the local environment or in the velocity of the receiver terminal. The maximum Doppler frequency is  $f_D = r \sin \phi_{\max}/\lambda$ , where  $\phi_{\max}$  is the largest angle of arrival. In urban environments, especially in the low traffic density conditions of these measurements, there is generally no significant scattered signal energy arriving directly from the front or rear of the vehicle. The diagonal lines in Figure 3.9 show the changing direction of arrival of multipath components arriving from distinct physical objects such as parked vehicles or buildings.

Testing for WSS is then realized by testing for changes in the means and in the estimated ES and ECS. The tests are applied to  $N$  segments of length  $K$ ; for the results presented below,  $N = 3$  was used, for a total interval length of just over 0.5 s. The statistical tests used are the univariate analysis of variance (ANOVA) and its multivariate counterpart, MANOVA; see, for example, [50, Chapter 5]. These test whether the differences in the means of sets of observations are statistically significant. An outline of the procedure is given here—interested readers are directed to [48] for more detailed information.

The test for wide-sense stationarity of MIMO intervals can be separated into two hypotheses:

$\mathcal{H}_{01}$ : the individual data series,  $\{x_i(k)\}$ ,  $i = 1, \dots, N_r \cdot N_t$ , are WSS.

This has two parts:

$\mathcal{H}_{01a}$ : the means of the  $N$  data segments are equal;

$\mathcal{H}_{01b}$ : the ES density functions of the  $N$  data segments are equal.

$\mathcal{H}_{02}$ : the ECS density functions of  $\{\mathbf{x}(k)\}$  for each data segment are equal.

Hypothesis  $\mathcal{H}_{02}$  need not be tested if  $\mathcal{H}_{01}$  is rejected; similarly if  $\mathcal{H}_{01a}$  is rejected,  $\mathcal{H}_{01b}$  is not necessary.

To test  $\mathcal{H}_{01a}$  for data series  $\{x_i\}$ ,  $i = 1, \dots, N_r \cdot N_t$ , test vectors are generated using the real part of the stacked channel response vector, and a one-way MANOVA [50, Chapter 5] is applied. The test may be repeated using the imaginary parts of the channel responses. The hypothesis is tested using the Pillai–Bartlett trace [51, Chapter 4] with  $df_e = K(N - 1)$  degrees of freedom to generate the statistic  $\hat{F}$ . This is used in an  $F$  test using  $df_1 = N_r N_t (N - 1)$  and  $df_2 = s(df_e - N_r N_t + s)$ , where  $s = \min(N_r N_t, N - 1)$ , to obtain the  $p$  value,  $p = P(\hat{F} | \mathcal{H}_{01a})$ , which is the probability of observing the statistic  $\hat{F}$  if the null hypothesis  $\mathcal{H}_{01a}$  were true. Then  $\mathcal{H}_{01a}$  is rejected at a significance level  $\alpha$  if  $p < \alpha$ .

The MANOVA is based on the assumptions that the data samples  $\{x_i(k)\}$  are normally distributed and that the pairwise covariances are similar. The first assumption can be tested using the Kolmogorov–Smirnov goodness-of-fit test [52, Chapter 3]; it has been found from measured data that this assumption is generally true when the scattering environment is at least moderately rich. Note that processes that are Gaussian and wide-sense stationary are also ergodic. The second assumption is confirmed by acceptance of  $\mathcal{H}_{01b}$ .

For the second moment tests ( $\mathcal{H}_{01b}$  and  $\mathcal{H}_{02}$ ), a log transform is applied to the ES and ECS estimates to reduce the difference in estimation error at different frequencies,  $\omega$ . The log estimates are sampled at intervals  $K$  in time and  $\pi/h$  in frequency to provide approximately independent samples. Furthermore, only the gain of the ECS will be considered, as the phase does not satisfy the normality assumptions required to apply the MANOVA. Thus, for a given pair  $i, j$ , the test samples are

$$y_{mn}^{(i,j)} = \ln |\hat{S}_{ij}(nK, m\pi/h)|, \quad m \in \mathcal{M}, n \in \mathcal{N} \quad (3.50)$$

where  $\mathcal{N} = \{n, n = 1, \dots, N\}$  and  $\mathcal{M} = \{m, m \neq 0, |m/h| \leq f_D T_s/2\}$ . The set  $\mathcal{M}$  of frequency indices excludes frequencies above the maximum Doppler shift, where there is no signal content. The number of frequencies used in the test will be denoted  $M$ .

The test vector for  $\mathcal{H}_{01b}$  is then

$$\mathbf{d}_{mn} = \begin{bmatrix} y_{mn}^{(1,1)} & y_{mn}^{(2,2)} & \dots & y_{mn}^{(N_r \cdot N_t, N_r \cdot N_t)} \end{bmatrix}^T. \quad (3.51)$$

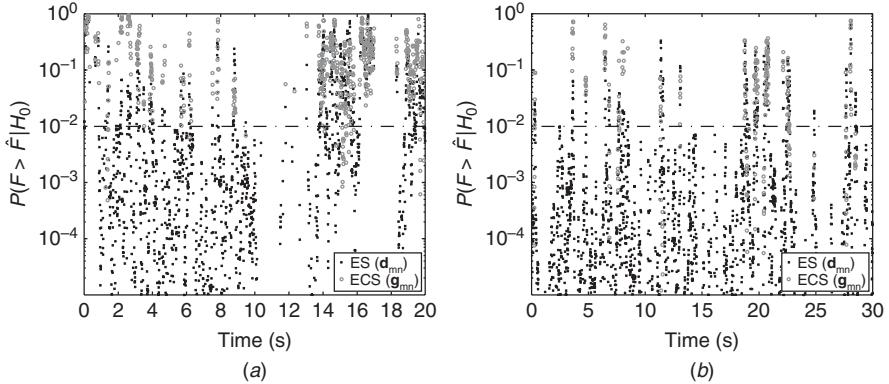
The two-way MANOVA is applied to this vector using  $df_e = (N - 1)(K - 1)$ , to test for variations over time. Finally, if  $\mathcal{H}_{01}$  is accepted,  $\mathcal{H}_{02}$  is tested in a similar way to  $\mathcal{H}_{01b}$ , using the length  $N_r N_t (N_r N_t - 1)/2$  vector

$$\mathbf{g}_{mn} = \begin{bmatrix} y_{mn}^{(1,2)} & y_{mn}^{(1,3)} & \dots & y_{mn}^{(N_r \cdot N_t - 1, N_r \cdot N_t)} \end{bmatrix}^T. \quad (3.52)$$

The Pillai–Bartlett trace statistic is generated as for  $\mathcal{H}_{01b}$ , and the  $p$  value is again computed using the  $F$  test to determine whether the null hypothesis should be accepted.

The results of the second-order tests for wide-sense stationarity applied to the measurements along Bank and Slater Streets are shown in Figure 3.10. Small arrays with  $N_t = N_r = 2$  antenna elements were considered, separated by one-half wavelength at the transmitter and receiver. The tests have not been performed over the intervals in which  $\mathcal{H}_{01a}$  was rejected. The value of  $\alpha = 0.01$  is also shown; recall that the null hypothesis is rejected if  $p < \alpha$ . Only two antenna elements are considered at each terminal, spaced by one-half wavelength. The vectors used in testing  $\mathcal{H}_{01b}$  and  $\mathcal{H}_{02}$  are length  $N_r N_t$  and  $N_r N_t (N_r N_t - 1)/2$ , respectively. For large array sizes, the tests become computationally unwieldy and less reliable for the small sample sizes available in the quasi-stationary intervals.

Figure 3.10 shows that the channel response vectors along both streets are only WSS for short bursts. The second moment is usually the cause of rejecting the null hypothesis. The first moment hypothesis is rejected on Bank St. while the receiver is passing through the intersection (10–13 s) because there is a large portion signal energy that arrives from the side, causing variations in the means of the real and imaginary parts of the channel response. This effect is also observed in the region 17–18 s, where



**Figure 3.10** Second-order test MANOVA  $p$  values using intervals of approximately 0.5 s, for  $N_t = N_r = 2$ , with one-half wavelength spacing: (a) Bank St.; (b) Slater St.

there is an opening along the street that enables a strong specular signal to be received almost perpendicular to the motion of the vehicle. The WSS hypothesis is accepted for the few seconds after the vehicle enters the urban canyon south of the intersection: At this point, the signal is received mainly along the street (see Fig. 3.7a) and thus is a stabilizing feature in the evolution spectrum. For Slater St. (comparing Figs. 3.10b and 3.9), it is seen that the results are consistent with the ES estimates, that is, the responses are concluded to be WSS when there is little variation in the ES estimates. Overall, the WSS hypothesis is accepted in 39% of half-second intervals on Bank St. but only 8% on Slater St.

### 3.6 SUMMARY

The channel response matrix  $\mathbf{H}$  experienced by a MIMO system depends on the physical features of the propagating environment, the spatial arrangement and electromagnetic properties of the transmitter and receiver antenna elements, and the degree of mobility. A wide range of MIMO channel models has been proposed, which are capable of representing some or all of the desired features. Some models are based on physical considerations and provide a realistic representation of typical propagation conditions, while some others rely on unrealistic assumptions, such as the location of scatterers, in order to achieve analytical tractability.

The effects of mobility can be modeled over short distances where the multipath structure changes very little. The main effect observed in this case is multipath fading, which can be simulated with either physical or analytical models. Analytical models can be used to provide many realizations of the channel response matrix  $\mathbf{H}$  that all have the same correlation properties, suitable for simulating quasi-static conditions. On the other hand, physical models are able to provide a time series of channel response matrices,  $\{\mathbf{H}(t)\}$ , reflecting the temporal fading effects.

Over large distances, the path loss and angles of departure and/or arrival of multipath components change, which affects the autocorrelation function of the channel. Intervals in which the autocorrelation function, or equivalently the power-spectral density function, changes are not wide-sense stationary. From the MIMO channel measurements

obtained in urban Ottawa, Canada, it was seen that these nonstationary intervals are the majority. This poses a challenge for testing MIMO systems in real mobile environments as modeling and simulation methods for nonstationary conditions are limited and difficult to validate experimentally.

## REFERENCES

1. A. F. Molisch, H. Asplund, R. Hedbergott, M. Steinbauer, and T. Zwick, "The COST 259 directional channel model—Part I: Overview and methodology," *IEEE Trans. Wireless Commun.*, vol. 5, pp. 3421–3443, Dec. 2006.
2. M. Steinbauer, A. F. Molisch, and E. Bonek, "The double-directional radio channel," *IEEE Antennas Propagat. Mag.*, vol. 43, pp. 51–63, Aug. 2001.
3. J. H. Winters, "On the capacity of radio communication systems with diversity in a Rayleigh fading environment," *IEEE J. Select. Areas Commun.*, vol. 5, pp. 871–878, June 1987.
4. E. Telatar, "Capacity of multi-antenna Gaussian channels," *Eur. Trans. Commun.*, vol. 10, no. 6, pp. 585–596, 1999.
5. G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Commun.*, vol. 6, no. 3, pp. 311–335, 1998.
6. A. Goldsmith, S. A. Jafar, N. Jindal, and S. Vishwanath, "Capacity limits of MIMO channels," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 684–701, June 2003.
7. R. G. Gallager, *Information Theory and Reliable Communication*, New York: Wiley, 1968.
8. T. J. Willink, "Improving power allocation to MIMO eigenbeams under imperfect channel estimation," *IEEE Commun. Lett.*, vol. 9, pp. 622–624, July 2005.
9. T. Yoo and A. Goldsmith, "Capacity and power allocation for fading MIMO channels with channel estimation error," *IEEE Trans. Inform. Theory*, vol. 52, pp. 2203–2214, May 2006.
10. L. M. Correia (Ed.), *Mobile Broadband Multimedia Networks*, New York: Academic, 2006.
11. W. C. Y. Lee, "Correlations between two mobile base-station antennas," *IEEE Trans. Commun.*, vol. 21, pp. 1214–1224, Nov. 1973.
12. R. H. Clarke, "A statistical theory of mobile radio reception," *Bell. Syst. Tech. J.*, vol. 47, pp. 957–1000, Jul./Aug.
13. G. L. Stüber, *Principles of Mobile Communication*, Kluwer Academic, 2001.
14. P. Petrus, J. H. Reed, and T. S. Rappaport, "Geometrical-based statistical macrocell channel model for mobile environments," *IEEE Trans. Commun.*, vol. 50, pp. 495–502, Mar. 2002.
15. J. C. Liberti and T. S. Rappaport, "A geometrically based model for line-of-sight multipath radio channels," in *Proc. IEEE 46th Veh. Tech. Conf.*, Vol. 2, Apr. 1996, pp. 844–848.
16. M. Pätzold and B. O. Hogstad, "A wideband MIMO channel model derived from the geometric elliptical scattering model," *Wireless Commun. Mobile Comput.*, 2007.
17. G. J. Byers and F. Takawira, "Spatially and temporally correlated MIMO channels: Modeling and capacity analysis," *IEEE Trans. Veh. Technol.*, vol. 53, pp. 634–643, May 2004.
18. M. Pätzold, B. O. Hogstad, and N. Youssef, "Modeling, analysis, and simulation of MIMO mobile-to-mobile fading channels," *IEEE Trans. Wireless Commun.*, vol. 7, pp. 510–520, Feb. 2008.
19. A. G. Zajić and G. L. Stüber, "Space-time correlated mobile-to-mobile channels: Modelling and simulation," *IEEE Trans. Veh. Technol.*, vol. 57, pp. 715–726, Mar. 2008.
20. A. Abdi, J. A. Barger, and M. Kaveh, "A parametric model for the distribution of the angle of arrival and the associated correlation function and power spectrum at the mobile station," *IEEE Trans. Veh. Technol.*, vol. 51, pp. 425–434, May 2002.

21. A. F. Molisch, A. Kuchar, J. Laurila, K. Hugl, and R. Schmalenberger, "Geometry-based directional model for mobile radio channels—Principles and implementation," *Eur. Trans. Commun.*, vol. 14, pp. 351–359, 2003.
22. L. Laurila, A. F. Molisch, and E. Bonek, "Influence of the scatterer distribution on power delay profiles and azimuthal power spectra of mobile radio channels," in *Proc. Int. Symp. Spread Spectrum Tech. and Appl.*, Vol. 1, Sept. 1998, pp. 267–271.
23. L. M. Correia (Ed.), *Wireless Flexible Personalised Communications*, New York: Wiley, 2001.
24. "NEWCOM—Network of Excellence in Wireless Communications," available: newcom.ismb.it.
25. "WINNER—Wireless World Initiative New Radio," available: www.ist-winner.org.
26. A. F. Molisch, "A generic model for MIMO wireless propagation channels in macro and microcells," *IEEE Trans. Signal Process.*, vol. 52, pp. 61–71, Jan. 2004.
27. N. Blaunstein, M. Toeltsch, J. Laurila, E. Bonek, D. Katz, P. Vainikainen, N. Tsouri, K. Kalliola, and H. Laitinen, "Signal power distribution in the azimuth, elevation and time delay domains in urban environments for various elevations of base station antenna," *IEEE Trans. Ant. Propag.*, vol. 54, pp. 2902–2916, Oct. 2006.
28. D. Chizhik, G. J. Foschini, and R. A. Valenzuela, "Capacities of multi-element transmit and receive antennas: Correlations and keyholes," *Elect. Lett.*, vol. 36, pp. 1099–1100, June 2000.
29. D. Gesbert, H. Bölcskei, D. A. Gore, and A. J. Paulraj, "Outdoor MIMO wireless channels: Models and performance prediction," *IEEE Trans. Commun.*, vol. 50, pp. 1926–1934, Dec. 2002.
30. H. Asplund, A. A. Glazunov, A. F. Molisch, K. I. Pedersen, and M. Steinbauer, "The COST 259 directional channel model—Part II: Macrocells," *IEEE Trans. Wireless Commun.*, vol. 5, pp. 3434–3450, Dec. 2006.
31. A. A. M. Saleh and R. A. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE J. Select. Areas Commun.*, vol. 5, pp. 128–137, Feb. 1987.
32. P. Petrus, J. H. Reed, and T. S. Rappaport, "Effects of directional antennas at the base station on the Doppler spectrum," *IEEE Commun. Lett.*, vol. 1, pp. 40–42, Mar. 1997.
33. Q. H. Spencer, B. D. Jeffs, M. A. Jensen, and A. L. Swindlehurst, "Modeling the statistical time and angle of arrival characteristics of an indoor multipath channel," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 347–360, Mar. 2000.
34. J. W. Wallace and M. A. Jensen, "Modeling the indoor MIMO wireless channel," *IEEE Trans. Ant. Propag.*, vol. 50, pp. 591–599, May 2002.
35. C.-C. Chong, C.-M. Tan, D. I. Laurenson, S. McLaughlin, M. A. Beach, and A. R. Nix, "A new statistical wideband spatio-temporal channel model for 5-GHz band WLAN systems," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 139–150, Feb. 2003.
36. J. W. Wallace and M. A. Jensen, "Time-varying mimo channels: Measurement, analysis, and modeling," *IEEE Trans. Ant. Propag.*, vol. 54, pp. 3265–3273, Nov. 2006.
37. H. Bölcskei and D. G. A. J. Paulraj, "On the capacity of OFDM-based spatial multiplexing systems," *IEEE Trans. Commun.*, vol. 50, pp. 225–234, Feb. 2002.
38. W. C. Jakes (Ed.), *Microwave Mobile Communications*, New York: Wiley, 1974.
39. C.-N. Chuah, J. M. Kahn, and D. Tse, "Capacity of multi-antenna array systems in indoor wireless environment," in *Proc. Globecom*, Vol. 4, 1998, pp. 1894–1899.
40. D.-S. Shiu, G. J. Foschini, M. J. Gans, and J. M. Kahn, "Fading correlation and its effect on the capacity of multielement antenna systems," *IEEE Trans. Commun.*, vol. 48, pp. 502–513, Mar. 2000.

41. A. G. Burr, "Capacity bounds and estimates for the finite scatterers MIMO wireless channel," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 812–818, June 2003.
42. A. M. Sayeed, "Deconstructing multiantenna fading channels," *IEEE Trans. Signal Process.*, vol. 50, pp. 2563–2579, Oct. 2002.
43. W. Weichselberger, M. Herdin, H. Özcelik, and E. Bonek, "A stochastic MIMO channel model with joint correlation of both link ends," *IEEE Trans. Wireless Commun.*, vol. 5, pp. 90–1001, Jan. 2006.
44. N. Costa and S. Haykin, "A novel wideband MIMO channel model and experimental validation," *IEEE Trans. Ant. Propag.*, vol. 56, pp. 550–562, Feb. 2008.
45. C. Squires, T. Willink, and B. Gagnon, "A flexible platform for MIMO channel characterization and system evaluation," in *WIRELESS 2003—Proc. 15th Conf. on Wireless Commun.*, Calgary, Canada, July 2003.
46. H. Özcelik, N. Czink, and E. Bonek, "What makes a good MIMO channel model?" in *Proc. 61st Veh. Tech. Conf. (VTC Spring 05)*, Vol. 1, Stockholm, Sweden, May 2005, pp. 156–160.
47. H. Özcelik, M. Herdin, W. Weichselberger, J. Wallace, and E. Bonek, "Deficiencies of 'Kronecker' MIMO radio channel model," *Elect. Lett.*, vol. 39, pp. 1209–1210, Aug. 2003.
48. T. J. Willink, "Wide-sense stationarity of mobile MIMO radio channels," *IEEE Trans. Veh. Technol.*, vol. 57, pp. 704–714, Mar. 2008.
49. M. Priestley, *Spectral Analysis and Time-Series*, London: Academic, 1981.
50. D. Morrison, *Multivariate Statistical Methods*, New York: McGraw-Hill, 1976.
51. D. J. Hand and C. C. Taylor, *Multivariate Analysis of Variance and Repeated Measures*, Boca Raton, FL: CRC Press, 1987.
52. L. Sachs, *Applied Statistics*, New York: Springer Verlag, 1982.



## CHAPTER 4

---

# Robustness Issues in Sensor Array Processing

Alex B. Gershman

Technische Universität Darmstadt, Darmstadt, Germany

## 4.1 INTRODUCTION

In the last four decades, sensor array processing has found numerous applications in radar [1–4], sonar [5–9], microphone arrays [10, 11], wireless communications [12–14], navigation [15, 16], seismology [17–19], radio astronomy [20, 21], biomedicine [22–24], automotive processing [25], and other fields [26–29].

Direction-of-arrival (DOA) estimation and adaptive beamforming are two important areas of sensor array processing that will be considered in this chapter. The main objective of DOA estimation is to obtain accurate high-resolution estimates of the source DOAs, whereas the primary goal of adaptive beamforming is to detect and estimate the signal-of-interest waveforms in the presence of interference and noise by means of data-adaptive spatial filtering.

Both these areas have a long history of theoretical research and practical applications, and a variety of advanced DOA estimation and adaptive beamforming methods have been proposed in the last four decades; see [30–37] and references therein.

However, most of the existing array processing methods are entirely based on the assumption of the exact knowledge of the array manifold (i.e., the signal propagation model and antenna array parameters). Moreover, some of these methods are additionally based on quite restrictive assumptions on the signal or interference sources and sensor noise. As a result, such methods can be subject to a severe performance degradation in practical cases when their underlying assumptions on the environment, sensor array, or sources/noise become violated. The main reason for such a degradation is a high sensitivity of high-resolution DOA estimation and adaptive beamforming methods to signal model and array manifold errors. Such errors can be caused, for example, by signal pointing mismatches, distorted or time-varying array shape, imperfect array calibration, unknown multipath propagation and scattering effects, unknown sensor mutual

coupling, unknown ambient noise fields, as well as environmental time variability and fluctuations [5–9, 38–78].

In the present chapter, we will consider robustness issues in narrowband sensor array processing. In Section 4.2, the array signal model is presented and the DOA estimation problem is formulated. In the same section, we discuss the main types of robustness required in direction finding and present an overview of traditional and robust DOA estimation techniques, with a particular emphasis on array self-calibration methods, DOA estimators for partly calibrated and time-varying arrays, source localization techniques in the presence of rapidly moving sources, and DOA estimation methods that are robust against unknown ambient noise fields. In Section 4.3, robust adaptive beamforming methods are considered. In this section, both the traditional (ad hoc) and advanced (worst-case performance optimization based) robust beamformers are discussed in detail.

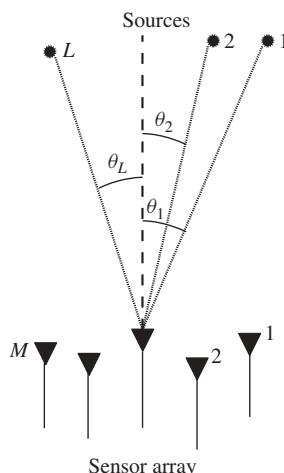
## 4.2 DIRECTION-OF-ARRIVAL ESTIMATION

### 4.2.1 Signal Model

The geometry of one-dimensional DOA estimation problem is shown in Figure 4.1. In the traditional formulation of this problem, it is assumed that an array of  $M$  sensors receives narrowband signals from  $L$  ( $L < M$ ) point signal sources located at the DOAs  $\{\theta_1, \dots, \theta_L\}$ . In this case, the  $M \times 1$  array observation vector at discrete time  $t$  can be modeled as [33, 35]

$$\mathbf{x}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}(t), \quad (4.1)$$

where  $\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1), \dots, \mathbf{a}(\theta_L)]$  is the  $M \times L$  direction matrix,  $\mathbf{a}(\theta)$  is the  $M \times 1$  steering vector,  $\boldsymbol{\theta} = [\theta_1, \dots, \theta_L]^T$  is the  $L \times 1$  vector of unknown source DOAs,  $\mathbf{s}(t) = [s_1(t), \dots, s_L(t)]^T$  is the  $L \times 1$  vector of unknown source waveforms,  $\mathbf{n}(t) = [n_1(t), \dots, n_M(t)]^T$  is the  $M \times 1$  vector of unknown sensor noise, and  $(\cdot)^T$  denotes the transpose.



**Figure 4.1** Geometry of one-dimensional DOA estimation problem.

It is typical for most studies in the field of array processing to consider the ideal case of exactly known array manifold, where the direction matrix  $\mathbf{A}(\boldsymbol{\theta})$  is known up to the unknown parameter vector  $\boldsymbol{\theta}$ . However, in practical cases this assumption can be often violated because of a poor array calibration, an imperfect knowledge of the propagation channel, or unknown array shape distortions [39–51].

Another typical (and somewhat more specific) assumption is that the sources are far field and have plane wavefronts. This assumption is often violated in practice because of wavefront distortions/fluctuations due to unknown propagation effects and because of sources that may be located in the near field of the antenna array [40, 41, 58, 62].

Another overly idealistic assumption that is commonly made when designing DOA estimation methods is that the matrix  $\mathbf{A}(\boldsymbol{\theta})$  is time invariant over a certain observation interval. In practice, however, both the array geometry and the source locations may rapidly vary in time because of source/array motion or vibration [53–57, 63–65].

Using (4.1), the array covariance matrix can be expressed as

$$\mathbf{R} \triangleq E\{\mathbf{x}(t)\mathbf{x}^H(t)\} = \mathbf{A}\mathbf{S}\mathbf{A}^H + \mathbf{Q}, \quad (4.2)$$

where  $\mathbf{S} \triangleq E\{\mathbf{s}(t)\mathbf{s}^H(t)\}$  is the  $L \times L$  full-rank covariance matrix of the source waveforms,  $\mathbf{Q} \triangleq E\{\mathbf{n}(t)\mathbf{n}^H(t)\}$  is the  $M \times M$  full-rank covariance matrix of sensor noise,  $E\{\cdot\}$  denotes the statistical expectation, and  $(\cdot)^H$  stands for the Hermitian transpose. It is quite typical to assume in the literature that sensor noises are temporally and spatially white complex Gaussian random processes, that is,

$$E\{\mathbf{n}(t)\mathbf{n}^H(k)\} = \delta_{tk}\sigma^2\mathbf{I}, \quad (4.3)$$

where  $\sigma^2$  is the noise variance and  $\delta_{tk}$  is the Kronecker delta. In the latter case, (4.2) can be simplified as

$$\mathbf{R} = \mathbf{A}\mathbf{S}\mathbf{A}^H + \sigma^2\mathbf{I}. \quad (4.4)$$

In practice, the external (ambient) sensor noise field can be unknown and may not satisfy the assumption in (4.3); see [69–77] and references therein. In such a case, the presence of colored or spatially nonuniform noise may lead to a severe degradation of high-resolution DOA estimation methods.

In real-world applications, the matrix  $\mathbf{R}$  is unknown. It is usually estimated from the snapshot data as

$$\hat{\mathbf{R}} = \frac{1}{J} \sum_{t=1}^J \mathbf{x}(t)\mathbf{x}^H(t) = \frac{1}{J} \mathbf{X}\mathbf{X}^H, \quad (4.5)$$

where  $J$  is the number of snapshots and  $\mathbf{X} = [\mathbf{x}(1), \dots, \mathbf{x}(J)]$  is the  $M \times J$  data matrix.

#### 4.2.2 Traditional DOA Estimation Techniques

First of all, let us briefly consider traditional DOA estimation techniques that directly or indirectly use the assumptions of the exact knowledge of the array manifold and spatially white noise.

One of the simplest nonparametric DOA estimation methods is commonly referred to as *conventional beamformer*. It is based on scanning the array beam and computing the output power for each beam scan angle [34, 35]. The beamformer output power

for the angle  $\theta$  is given by  $E\{|a^H(\theta)x(t)|^2\} = a^H(\theta)\mathbf{R}a(\theta)$ . In the finite sample case, the conventional beamformer function is given by

$$f_{CB}(\theta) = \frac{1}{J} \sum_{t=1}^J |a^H(\theta)x(t)|^2 = a^H(\theta)\hat{\mathbf{R}}a(\theta). \quad (4.6)$$

The source DOA estimates can be found from the  $L$  main maxima of this function. Although the conventional beamformer is rather limited in its angular resolution, its significant advantages are implementational simplicity and robustness.

To overcome the low-resolution property of the conventional beamformer technique, it was suggested in [79] to estimate the spatial spectrum of multiple sources by means of a spatial filter that maintains a distortionless response toward the signal coming from the direction  $\theta$  while minimizing the total output array power. In the finite sample case, the resulting nonparametric Capon spectral function can be expressed as

$$f_{CAPON}(\theta) = \mathbf{w}^H(\theta)\hat{\mathbf{R}}\mathbf{w}(\theta), \quad (4.7)$$

where the weight vector  $\mathbf{w} = \mathbf{w}(\theta)$  of the spatial filter can be found by solving the following optimization problem [79]:

$$\min_{\mathbf{w}} \mathbf{w}^H \hat{\mathbf{R}} \mathbf{w} \quad \text{subject to} \quad \mathbf{w}^H \mathbf{a}(\theta) = 1. \quad (4.8)$$

Solving (4.8) and substituting the resulting weight vector to the Capon spectral function yields [35, 79]

$$f_{CAPON}(\theta) = \frac{1}{a^H(\theta)\hat{\mathbf{R}}^{-1}a(\theta)}. \quad (4.9)$$

The Capon estimator of (4.9) is known to achieve better resolution than the conventional beamformer technique, and, hence, it belongs to the class of so-called high-resolution methods [33–35]. However, its resolution is still quite limited in the sense that it does not improve when the number of snapshots is increased. Hence, the performance of the Capon technique in scenarios with closely spaced signal sources can be much worse than that predicted by the corresponding deterministic and stochastic Cramér–Rao bounds (CRBs) [34].

To further improve the performance of DOA estimation in closely spaced source scenarios, a number of parametric techniques were proposed in the late 1970s early 1980s see [34–36] and references therein. These techniques are entirely based on the signal parameterization according to model (4.1).

Perhaps the most statistically motivated approach among the parametric techniques is the maximum-likelihood (ML) method, whose essence is to maximize the stochastic or deterministic *likelihood function* over the unknown signal and noise parameters [80, 81]. For example, in the deterministic case the ML technique obtains the source DOAs from the minimum of the following concentrated negative log-likelihood function [34, 35]:

$$\mathcal{L}(\theta) = \text{trace}\{\mathbf{P}_A^\perp(\theta)\hat{\mathbf{R}}\}, \quad (4.10)$$

where

$$\mathbf{P}_A(\theta) = \mathbf{A}(\theta)(\mathbf{A}^H(\theta)\mathbf{A}(\theta))^{-1}\mathbf{A}^H(\theta), \quad (4.11)$$

$$\mathbf{P}_A^\perp(\theta) = \mathbf{I} - \mathbf{P}_A(\theta) \quad (4.12)$$

are the orthogonal projection matrices onto the subspace spanned by the columns of  $\mathbf{A}(\theta)$  and onto the orthogonal complement to this subspace, respectively, and  $\text{trace}\{\cdot\}$  denotes the trace of a matrix.

Both the deterministic and stochastic ML estimators have excellent threshold and asymptotic DOA estimation performances [34, 35]. Moreover, in the single-source case, the ML estimator of (4.10) has a very simple form, as it is equivalent to the conventional beamformer approach of (4.6) [37]. However, a serious drawback of the ML techniques in the multiple source case is that they are based on a highly nonlinear optimization in a high-dimensional parameter space, and, hence, their implementation may be prohibitively expensive if the number of sources is large.

Computationally attractive alternatives to the ML DOA estimation techniques are *subspace* DOA estimation methods.<sup>1</sup> The most popular approach among these methods is the multiple signal classification (MUSIC) approach [82, 83]. It exploits a specific structure of the array covariance matrix (4.4) whose eigendecomposition can be written as [35, 82]

$$\mathbf{R} = \mathbf{E}\Lambda\mathbf{E}^H + \mathbf{G}\Gamma\mathbf{G}^H, \quad (4.13)$$

where the  $M \times L$  matrix  $\mathbf{E}$  contains the  $L$  signal subspace eigenvectors of  $\mathbf{R}$ , and the  $L \times L$  diagonal matrix  $\Lambda$  is built from the corresponding eigenvalues. Similarly, the  $M \times (M - L)$  matrix  $\mathbf{G}$  contains the  $M - L$  noise subspace eigenvectors of  $\mathbf{R}$ , and the  $(M - L) \times (M - L)$  diagonal matrix  $\Gamma$  contains the corresponding eigenvalues. It can be proven that the noise subspace and the column space of  $\mathbf{A}$  are orthogonal, that is,  $\mathbf{G}^H\mathbf{A} = \mathbf{0}$  where  $\mathbf{0}$  is the all-zeros matrix [35, 82, 83]. Using this property, the signal DOAs can be found from the following equation [82]:

$$\mathbf{a}^H(\theta)\mathbf{G}\mathbf{G}^H\mathbf{a}(\theta) = 0. \quad (4.14)$$

It follows from (4.14) that in the finite-sample case, the signal DOAs can be estimated from the locations of the  $L$  highest peaks of the following MUSIC spectrum function [82, 83]:

$$f_{\text{MUSIC}}(\theta) = \frac{1}{\mathbf{a}^H(\theta)\hat{\mathbf{G}}\hat{\mathbf{G}}^H\mathbf{a}(\theta)}, \quad (4.15)$$

where the finite-sample counterpart of (4.13),

$$\hat{\mathbf{R}} = \hat{\mathbf{E}}\hat{\Lambda}\hat{\mathbf{E}}^H + \hat{\mathbf{G}}\hat{\Gamma}\hat{\mathbf{G}}^H \quad (4.16)$$

is used in (4.15).

The MUSIC estimator (4.15) is known to achieve an excellent trade-off between the DOA estimation performance and computational cost [34, 35, 84]. As a result, MUSIC has been commonly accepted in the literature as a benchmark approach. It

<sup>1</sup>These methods are also sometimes referred to as *eigenstructure* techniques.

has stimulated the array processing community to seek for subspace techniques with further performance improvements and reduced computational cost [85–102]. However, MUSIC and most other subspace DOA estimation methods are quite sensitive to model mismatches [32, 42, 43]. Therefore, robust DOA estimation methods are of significant demand.

### 4.2.3 Imperfectly Calibrated Arrays

As the traditional high-resolution DOA estimation methods lack robustness against manifold errors, their performance can be seriously affected by an imperfect array calibration, for example, sensor gain/phase/location uncertainties or unknown sensor mutual coupling effects. A practical approach to alleviate this problem is to use array self-calibration techniques that exploit the array received signals to improve its calibration; see [45–51] and references therein.

To illustrate the principle of self-calibration techniques, let us first formulate the array observation model in the presence of manifold errors. For the sake of simplicity, let us consider only the effect of unknown sensor gains and phases [46, 48]. In the latter case, Eq. (4.1) can be modified as

$$\mathbf{x}(t) = \mathbf{D}\mathbf{A}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}(t), \quad (4.17)$$

where  $\mathbf{D} = \text{diag}\{\gamma_1 e^{j\phi_1}, \dots, \gamma_M e^{j\phi_M}\}$  is the unknown  $M \times M$  diagonal matrix that contains the unknown sensor gains  $\gamma_i$  ( $i = 1, \dots, M$ ) and phases  $\phi_i$  ( $i = 1, \dots, M$ ), and  $j \triangleq \sqrt{-1}$ . Then, the array covariance matrix can be expressed as

$$\mathbf{R} = \mathbf{D}\mathbf{A}\mathbf{S}\mathbf{A}^H\mathbf{D}^H + \sigma^2\mathbf{I}. \quad (4.18)$$

The model of (4.17) and (4.18) can be straightforwardly extended to the case of unknown mutual coupling<sup>2</sup> and to the case of unknown sensor locations.<sup>3</sup>

One popular approach to array calibration is to jointly estimate the DOA and array parameters  $\{\theta_i\}_{i=1}^L$  and  $\{\gamma_i, \phi_i\}_{i=1}^M$  from the array observations using ML techniques that are straightforwardly applicable to the model of (4.17) [47, 48]. However, the ML-based self-calibration techniques involve highly nonlinear optimization over a large number of array and signal parameters and, as a result, are computationally demanding.

Alternatively, several subspace methods based on model (4.17) have been proposed. For example, in the case of small gain and phase errors, the authors of [50] proposed to use the first-order Taylor series expansion to obtain closed-form solutions for the unknown sensor gain and phase parameters. Because of inherent ambiguities in the sensor gain and phase estimates, this method requires the knowledge of at least one source DOA.

Another elegant and simple self-calibration approach has been developed in [46] for uniform linear arrays (ULAs). The key idea of this approach is to estimate the signal part of the array covariance matrix,  $\mathbf{D}\mathbf{A}\mathbf{S}\mathbf{A}^H\mathbf{D}^H$ , using the eigendecomposition of  $\mathbf{R}$ . Then, making use of the property  $|[\mathbf{D}\mathbf{A}\mathbf{S}\mathbf{A}^H\mathbf{D}^H]_{ik}| = |[\mathbf{A}\mathbf{S}\mathbf{A}^H]_{ik}| \gamma_i \gamma_k$  and using the Toeplitz structure of  $\mathbf{A}\mathbf{S}\mathbf{A}^H$  in the ULA case, a system of equations for the gain parameters can be obtained and solved by means of the least-squares (LS) technique. Using the so-obtained estimates of sensor gains, another system of equations can be

<sup>2</sup>In this case, the matrix  $\mathbf{D}$  becomes nondiagonal.

<sup>3</sup>In this case, the matrix  $\mathbf{A}$  should be additionally parameterized in terms of unknown sensor positions.

found for the sensor phases whose LS solution completes the procedure of estimating the unknown sensor gain and phase parameters. There are several types of inherent ambiguities in these estimates and, as a result, the final DOA estimates can only be obtained up to an arbitrary rotation factor. To resolve such rotational DOA ambiguity in the final DOA estimate, the knowledge of the phase difference between any two sensors is required [46].

Another promising approach to the array self-calibration problem in the presence of unknown sensor gains and phases has been developed in [52] for ULAs. It applies the idea of the estimation of signal parameters via rotational invariance technique (ESPRIT) algorithm [89] to the array observation model of (4.17) and achieves an excellent performance-to-complexity trade-off as compared to the other array self-calibration methods.

#### 4.2.4 Partly Calibrated Arrays

An interesting trend in robust array processing is DOA estimation in partly calibrated arrays [49, 102, 103]. For example, in large subarray-based sensor array systems, it is quite typical to have each subarray well calibrated, whereas it may be rather difficult to calibrate the whole array (i.e., to determine all the intersubarray manifold parameters).

There are several methods that are applicable to partly calibrated subarray-based sensor arrays and that do not require any additional self-calibration procedure. One important early example of such methods is the popular ESPRIT approach [89], which is applicable to a specific class of partly calibrated arrays composed of identical and identically oriented two-element subarrays.<sup>4</sup> It is worth mentioning that to estimate the source DOAs, the ESPRIT algorithm does not need any information about the intersubarray parameters. The only information needed is the geometry of any of its two-element subarrays. Unfortunately, the array geometries suitable for the ESPRIT technique are quite specific, and this may limit potential applications of this technique.

Several subspace DOA estimation methods have been developed that are applicable to more general classes of partly calibrated arrays, for example, arrays with multiple invariances [98–101] as well as subarray-based arrays with arbitrary subarray geometries [102–104].

As a representative example of such methods, let us consider the rank reduction (RARE) DOA estimator of [102, 103] that is applicable to the most general case of multiple arbitrary subarrays. Assuming that we have  $K$  arbitrary nonoverlapping subarrays, the actual array steering vector can be modeled as [103]

$$\mathbf{a}(\theta, \alpha) = \mathbf{V}(\theta)\mathbf{h}(\theta, \alpha), \quad (4.19)$$

where the  $M \times K$  matrix  $\mathbf{V}(\theta)$  is defined as

$$\mathbf{V}(\theta) \triangleq \begin{bmatrix} \mathbf{v}_1(\theta) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{v}_2(\theta) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{v}_K(\theta) \end{bmatrix}, \quad (4.20)$$

<sup>4</sup>Another interpretation of such arrays is that they should consist of two groups of sensors such that the second one could be obtained as a result of spatial translation of the first one.

$\mathbf{v}_k(\theta)$  is the  $M_k \times 1$  known steering vector of the  $k$ th subarray,  $M_k$  is the number of sensors of the  $k$ th subarray ( $\sum_{k=1}^K M_k = M$ ),  $\boldsymbol{\alpha}$  is the vector of unknown intersubarray parameters, and  $\mathbf{h}(\theta, \boldsymbol{\alpha})$  is the vector that captures all unknown intersubarray perturbations of the nominal array steering vector. The latter nominal array steering vector can be defined as

$$\mathbf{v} \triangleq [\mathbf{v}_1^T(\theta), \mathbf{v}_2^T(\theta), \dots, \mathbf{v}_K^T(\theta)]^T. \quad (4.21)$$

Note that the matrix  $\mathbf{V}(\theta)$  characterizes the known (calibrated) part of the array manifold, whereas  $\mathbf{h}(\theta, \boldsymbol{\alpha})$  characterizes its unknown (uncalibrated) part. The vector  $\boldsymbol{\alpha}$  may contain unknown intersubarray displacements, intersubarray timing errors, propagation channel mismatches between subarrays, or some combination of these effects. Specific expressions for the vector  $\mathbf{h}$  for different particular types of imperfections can be found in [103].

Substituting (4.19) to the MUSIC equation (4.14), we have

$$\mathbf{h}^H(\theta, \boldsymbol{\alpha}) \mathbf{V}^H(\theta) \mathbf{G} \mathbf{G}^H \mathbf{V}(\theta) \mathbf{h}(\theta, \boldsymbol{\alpha}) = 0. \quad (4.22)$$

This means that for a nonzero  $\mathbf{h}(\theta, \boldsymbol{\alpha})$  Eq. (4.14) can be only satisfied when the matrix

$$\mathbf{C}(\theta) \triangleq \mathbf{V}^H(\theta) \mathbf{G} \mathbf{G}^H \mathbf{V}(\theta) \quad (4.23)$$

drops rank. This fact can be used to find the source DOAs without any knowledge of the vectors  $\mathbf{h}$  or  $\boldsymbol{\alpha}$ . According to (4.22) and (4.23), in the finite-sample case the source DOAs can be estimated from the  $L$  highest peaks of the following RARE spectrum [102, 103]:

$$f_{\text{RARE}}(\theta) = \frac{1}{\det\{\hat{\mathbf{C}}(\theta)\}}, \quad (4.24)$$

where

$$\hat{\mathbf{C}}(\theta) = \mathbf{V}^H(\theta) \hat{\mathbf{G}} \hat{\mathbf{G}}^H \mathbf{V}(\theta) \quad (4.25)$$

is the sample estimate of the matrix (4.23), and  $\det\{\cdot\}$  denotes the determinant of a matrix.

Note that the matrix  $\hat{\mathbf{C}}(\theta)$  is independent of  $\boldsymbol{\alpha}$ . Hence, to estimate the signal DOAs from (4.24), no knowledge of the intersubarray parameters is required. It is also noteworthy that in the fully calibrated array case (i.e., in the case of a single subarray,  $K = 1$ ) the RARE DOA estimator (4.24) reduces to the conventional MUSIC estimator [103].

Following the basic idea of root-MUSIC [86], a computationally efficient (search-free) version of the RARE method has been developed in [102] for a particular class of subarray geometries, where all subarrays have to be linear identically oriented, consist of omnidirectional sensors, and have interelement spacings that are integer multiples of the known shortest baseline  $d$ . In the latter case, each of the subarray steering vectors  $\mathbf{v}_k(\theta)$  always has in its entries integer powers of  $z = e^{j(2\pi/\lambda)d \sin \theta}$  where  $\lambda$  is the wavelength. Hence, the RARE null spectrum can be written as the following polynomial:

$$g_{\text{RARE}}(z) = \det\{\hat{\mathbf{C}}(z)\}, \quad (4.26)$$

where

$$\hat{\mathbf{C}}(z) = \mathbf{V}^T(1/z)\hat{\mathbf{G}}\hat{\mathbf{G}}^H\mathbf{V}(z). \quad (4.27)$$

The essence of the root-RARE algorithm proposed in [102] is to estimate the signal DOAs by rooting the polynomial (4.26). The roots of (4.26) can be then used to obtain the DOA estimates in the same way as in the root-MUSIC technique of [86].

An improved modification of the spectral and root-RARE techniques that has an additional robustness property against subarray orientation errors has been recently proposed in [104]. The essence of this approach is to use a truncated Taylor series expansion to model subarray steering vector distortions caused by unknown orientation errors.

#### 4.2.5 Time-Varying Arrays

In practical problems, the array shape and position can rapidly change in time [54, 55, 102]. In some applications (e.g., in the case of an airborne array), the array moves as a rigid body, that is, the array shape remains unchanged. However, in some applications the array shape may also rapidly change in time. Typical examples of such arrays are sonar time-varying arrays or arrays consisting of subarrays that are mounted on different moving (or vibrating) platforms.

In the case of time-varying arrays, the observation model can be written as [54]

$$\mathbf{x}(t) = \mathbf{A}(t, \theta)\mathbf{s}(t) + \mathbf{n}(t). \quad (4.28)$$

The only difference between (4.1) and (4.28) is that in (4.28) the matrix  $\mathbf{A}$  varies in time within the observation interval. The variations of the array manifold are assumed to be known in (4.28).

Using the model of (4.28), the authors of [54, 55] developed several DOA estimation methods for time-varying arrays. They have shown that the deterministic ML estimator amounts to minimizing the following negative concentrated log-likelihood function:

$$\mathcal{L}(\theta) = \sum_{t=1}^J \mathbf{x}^H(t) \mathbf{P}_A^\perp(t, \theta) \mathbf{x}(t), \quad (4.29)$$

where, in contrast to (4.10), the projection matrix  $\mathbf{P}_A^\perp$  depends on time. Note that (4.29) reduces to (4.10) in the conventional case of time-invariant array.

Another estimator developed in [54, 55] for the time-varying array case is the conventional beamformer that takes the following form:

$$f_{CB}(\theta) = \frac{1}{J} \sum_{t=1}^J |\mathbf{a}^H(t, \theta) \mathbf{x}(t)|^2. \quad (4.30)$$

It represents a straightforward extension of (4.6) and is equivalent to the ML estimator of (4.29) in the single-source case.

Based on the facts that the ML estimator in (4.29) can be computationally very expensive and that the resolution of the conventional beamformer technique of (4.30) is low, the authors of [55] have proposed a framework to extend subspace DOA estimation

methods to time-varying arrays. The key idea of their approach is closely related to that of coherent signal-subspace processing [105]. In particular, it is suggested in [55] to use properly selected preprocessing matrices that transform the actual time-varying array manifold to a virtual (time-invariant) one. These preprocessing transformation matrices have been designed in [55] using the array interpolation and focusing approaches [105, 106]. Then, any of conventional subspace-based DOA estimators (such as MUSIC) can be applied to the virtual array observations.

Although the subspace approach of [55] has an excellent performance when applied to arrays whose shape variations are moderate, it can degrade substantially when applied to arrays with large variations of the array shape. This degradation is mainly caused by the manifold transformation errors and is common for other coherent subspace methods as well [105].

Another computationally efficient subspace approach to DOA estimation in time-varying arrays composed of multiple fixed subarrays has been proposed in [102]. Its basic idea is to divide the observation interval into multiple subintervals so that the whole array remains time invariant within each of such subintervals. Then, it is suggested in [102] to apply the root-RARE algorithm to each of these subintervals and to coherently average the resulting polynomials over the whole observation interval. Although, similarly to the approach of [55], the time-varying RARE method of [102] enjoys the benefits of coherent subspace processing, there is no manifold transformation involved in [102]. Hence, the approach of [102] does not suffer from any sort of manifold transformation errors. However, similar to the time-invariant root-RARE technique, it imposes certain restrictions on the subarray geometry.

#### 4.2.6 Rapidly Moving Sources

The problem of moving-source localization and tracking using sensor arrays has attracted much attention in the literature; see [33, 63] and references therein. However, most of the source-tracking algorithms make little use of the array model and related parametric representation of the moving sources. Below, we will discuss DOA estimation techniques that make an explicit use of the array parametric model to characterize the source motion.

In the case of rapidly moving sources, the array observation model can be expressed as

$$\mathbf{x}(t) = \mathbf{A}(\boldsymbol{\theta}(t))\mathbf{s}(t) + \mathbf{n}(t). \quad (4.31)$$

Although this model is rather similar to model (4.28), the main difference between these models is that in (4.28) the source DOAs remain fixed but the array shape (i.e., the structure of the matrix  $\mathbf{A}$ ) varies in time, whereas in (4.31) the DOAs vary in time but the array shape remains unchanged within the observation interval.

In [64] and [65], it has been proposed to use the following local polynomial parameterization of the source time-varying DOAs through their initial DOAs and angular velocities:

$$\boldsymbol{\theta}(t) = \boldsymbol{\theta}^0 + (t - 1)\boldsymbol{\theta}', \quad t = 1, \dots, J, \quad (4.32)$$

where  $\boldsymbol{\theta}^0 = [\theta_1^0, \dots, \theta_L^0]^T$  is the vector of the initial source DOAs, and  $\boldsymbol{\theta}' = [\theta'_1, \dots, \theta'_L]^T$  is the vector of the source angular velocities.

It has been shown in [64] that using (4.31) and (4.32), the deterministic ML estimator of the DOA and velocity parameters  $\hat{\boldsymbol{\vartheta}} \triangleq [\boldsymbol{\theta}^{0T}, \boldsymbol{\theta}'^T]^T$  can be written in the following form:

$$\hat{\boldsymbol{\vartheta}}_{\text{ML}} = \arg \min_{\boldsymbol{\vartheta}} \mathcal{L}(\boldsymbol{\vartheta}), \quad \mathcal{L}(\boldsymbol{\vartheta}) = \sum_{t=1}^J \mathbf{x}^H(t) \mathbf{P}_A^\perp (\boldsymbol{\theta}^0 + (t-1)\boldsymbol{\theta}') \mathbf{x}(t), \quad (4.33)$$

where the time-varying projector matrix  $\mathbf{P}_A^\perp = \mathbf{P}_A^\perp (\boldsymbol{\theta}^0 + (t-1)\boldsymbol{\theta}')$  is explicitly parameterized in terms of  $\boldsymbol{\theta}^0$  and  $\boldsymbol{\theta}'$ .

Also, it has been shown in [65] that using (4.31) and (4.32), the conventional beamformer function can be extended as

$$f_{\text{CB}}(\boldsymbol{\theta}^0, \boldsymbol{\theta}') = \frac{1}{J} \sum_{t=1}^J |\mathbf{a}^H(\boldsymbol{\theta}^0 + (t-1)\boldsymbol{\theta}') \mathbf{x}(t)|^2, \quad (4.34)$$

where the estimates of  $\boldsymbol{\theta}^0$  and  $\boldsymbol{\theta}'$  can be obtained from (4.34) by means of two-dimensional search over  $\boldsymbol{\theta}^0$  and  $\boldsymbol{\theta}'$ .

Comparing (4.29) and (4.33), we can observe that the deterministic ML techniques in the time-varying array and moving-source cases are mathematically similar to each other. The same nontrivial conclusion can be made about the conventional beamformers (4.30) and (4.34). However, it should be stressed here that an important difference between (4.29)–(4.30) and (4.33)–(4.34) is that the techniques (4.33) and (4.34) use quite a different parameterization as compared to (4.29) and (4.30). In particular, the number of unknown parameters in (4.33) is twice as large as that in (4.29). Moreover, in contrast to (4.30), the conventional beamformer technique in (4.34) is based on a two-dimensional rather than one-dimensional search.

It has been shown in [64] and [65] that the explicit use of both the DOA and angular velocity parameters in the DOA estimators (4.33) and (4.34) substantially improves the estimation accuracy as compared to the conventional source-tracking techniques where the velocity parameters are usually obtained by differentiation of the DOA estimates [63].

As both techniques (4.33) and (4.34) are computationally expensive, it would be meaningful to obtain simpler subspace-based methods that are explicitly based on the local polynomial model (4.32) or its extensions. This problem is still open.

#### 4.2.7 Unknown Noise Fields

Most of the array processing techniques assume that the sensor noise covariance is a scaled identity matrix. This assumption is valid in many practical scenarios where the dominant component of the noise waveform is the thermal noise. However, in some cases [e.g., in radio-frequency (RF) systems operating in the high-frequency (HF) or very high frequency (VHF) bands and in most of sonar systems] the dominant noise

component is ambient noise [75]. In such cases, the sensor noise can be spatially correlated/nonuniform and unknown.

The problem of DOA estimation in unknown noise fields has gained much attention in the literature; see [70–77] and references therein.

In [72] and [75], two methods have been proposed where the noise and DOA parameters are jointly estimated using the ML approach. The technique of [72] parameterizes the unknown noise by an autoregressive (AR) model, while the method of [75] uses a parameterization based on a Fourier series representation of the noise spatial spectrum. A substantial drawback of these ML methods is their high computational complexity and weak robustness against the noise model mismatches. To reduce the number of noise parameters and improve the performance of ML methods in sparse arrays, block-correlated and white nonuniform noise models have been considered in [76] and [77].

Several subspace techniques to estimate source DOAs in unknown noise fields have been proposed as well. For example, a popular *covariance differencing* technique has been developed in [69] and [71]. Assuming that the noise covariance matrix  $\mathbf{Q}$  is symmetric and Toeplitz,<sup>5</sup> the authors of [71] proposed to take advantage of the following property of the noise covariance matrix:

$$\mathbf{Q} = \mathbf{J}\mathbf{Q}\mathbf{J}, \quad (4.35)$$

where  $\mathbf{J}$  is the exchange matrix with ones on its antidiagonal and zeros elsewhere. Taking the difference

$$\mathbf{R}_d = \mathbf{R} - \mathbf{JRJ} \quad (4.36)$$

and using (4.35), it follows that

$$\mathbf{R}_d = [\mathbf{A}, \mathbf{JA}] \begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & -\mathbf{S} \end{bmatrix} [\mathbf{A}, \mathbf{JA}]^H. \quad (4.37)$$

This structure of the covariance difference matrix  $\mathbf{R}_d$  is suitable for subspace-based methods. Therefore, the signal DOAs can be estimated by applying any such methods to the sample covariance difference matrix  $\hat{\mathbf{R}}_d = \hat{\mathbf{R}} - \mathbf{JRJ}$ . However, along with the estimates of the actual source bearings  $\{\theta_l\}_{l=1}^L$ , such DOA estimate will also contain “phantom” bearings  $\{-\theta_l\}_{l=1}^L$ . The authors of [69] have proposed an approach to remove such “phantom” DOAs from the final DOA estimate.

Although the covariance differencing method of [69–71] is computationally simpler than the ML techniques of [72] and [75–77], its main drawback is in that the covariance differencing operation effectively doubles the number of sources and, therefore, the condition  $M > 2L$  (rather than  $M > L$ ) must be satisfied for this method.

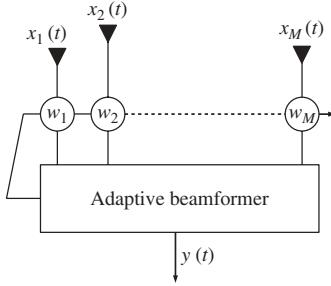
### 4.3 ADAPTIVE BEAMFORMING

#### 4.3.1 Background

The basic scheme of a narrowband adaptive beamformer is shown in Figure 4.2. The beamformer output is given by

$$y(t) = \mathbf{w}^H \mathbf{x}(t), \quad (4.38)$$

<sup>5</sup>This assumption corresponds to the case when the noise is cylindrically or spherically isotropic.



**Figure 4.2** Basic scheme of narrowband adaptive beamformer.

where  $\mathbf{w}$  is the  $M \times 1$  complex beamformer weight vector and  $\mathbf{x}(t)$  is the  $M \times 1$  complex array snapshot vector. This vector can be modeled as [30, 35]

$$\mathbf{x}(t) = \mathbf{x}_s(t) + \mathbf{x}_i(t) + \mathbf{n}(t), \quad (4.39)$$

where  $\mathbf{x}_s(t)$ ,  $\mathbf{x}_i(t)$ , and  $\mathbf{n}(t)$  are the statistically independent signal, interference, and noise components, respectively. If the desired signal is a rank-one source, we have that  $\mathbf{x}_s(t) = s(t)\mathbf{a}_s$ , where  $s(t)$  is the complex signal waveform and  $\mathbf{a}_s$  is the  $M \times 1$  signal steering vector that characterizes the signal spatial signature (array response). In contrast to Section 4.2, we do not parameterize this vector as a function of the signal DOA as no such parameterization is required.

The optimal weight vector can be found by means of maximizing the output signal-to-interference-plus-noise ratio (SINR) [30, 35]:

$$\text{SINR} = \frac{\mathbf{w}^H \mathbf{R}_s \mathbf{w}}{\mathbf{w}^H \mathbf{R}_{i+n} \mathbf{w}}, \quad (4.40)$$

where

$$\mathbf{R}_s \triangleq E \left\{ \mathbf{x}_s(t) \mathbf{x}_s^H(t) \right\}, \quad (4.41)$$

$$\mathbf{R}_{i+n} \triangleq E \left\{ (\mathbf{x}_i(t) + \mathbf{n}(t)) (\mathbf{x}_i(t) + \mathbf{n}(t))^H \right\} \quad (4.42)$$

are the signal and interference-plus-noise covariance matrices, respectively.

In the rank-one signal case,  $\mathbf{R}_s = \sigma_s^2 \mathbf{a}_s \mathbf{a}_s^H$  where  $\sigma_s^2$  is the signal power. In the latter case, Eq. (4.40) can be simplified as [30]

$$\text{SINR} = \frac{\sigma_s^2 |\mathbf{w}^H \mathbf{a}_s|^2}{\mathbf{w}^H \mathbf{R}_{i+n} \mathbf{w}}. \quad (4.43)$$

The optimal weight vector can be found by means of maximizing the SINR in (4.43) or, equivalently, maintaining distortionless response to the desired signal and minimizing the output interference-plus-noise power:

$$\min_{\mathbf{w}} \mathbf{w}^H \mathbf{R}_{i+n} \mathbf{w} \quad \text{subject to} \quad \mathbf{w}^H \mathbf{a}_s = 1. \quad (4.44)$$

Equation (4.44) is equivalent to the Capon spatial filtering problem (4.8). In adaptive beamforming, the Capon method is commonly called the minimum variance distortionless response (MVDR) beamformer [30, 35].

The solution to (4.44) can be expressed in the following well-known form [30, 35]:

$$\mathbf{w}_{\text{opt}} = c \mathbf{R}_{i+n}^{-1} \mathbf{a}_s, \quad (4.45)$$

where  $c = (\mathbf{a}_s^H \mathbf{R}_{i+n}^{-1} \mathbf{a}_s)^{-1}$ , so that the distortionless response constraint of (4.44) is satisfied. However, such a normalization of the optimal weight vector is immaterial from the SINR viewpoint because it does not affect (4.43). Therefore, the constant  $c$  will be omitted below.

In practical scenarios, the exact interference-plus-noise covariance matrix  $\mathbf{R}_{i+n}$  is unavailable, and it has to be estimated from the beamformer received data. Therefore,  $\mathbf{R}_{i+n}$  is usually replaced in (4.44) by the sample covariance matrix (4.5). With such a replacement, (4.45) becomes a popular sample matrix inverse (SMI) beamformer [30, 107]:

$$\mathbf{w}_{\text{SMI}} = \hat{\mathbf{R}}^{-1} \mathbf{a}_s. \quad (4.46)$$

The use of  $\hat{\mathbf{R}}$  in lieu of  $\mathbf{R}_{i+n}$  in (4.46) is known to dramatically affect the beamformer SINR in the case when the signal component is present in the beamformer training data snapshots [60, 108]. Such a performance degradation becomes especially pronounced when the signal spatial signature  $\mathbf{a}_s$  is imperfectly known because the SMI beamformer (4.46) is extremely sensitive even to small signal response errors. In the presence of such errors, it tends to misinterpret the signal components as interference and to suppress these components instead of protecting them [60, 108]. This phenomenon is commonly referred to as *signal self-nulling*.

Another frequent cause of beamformer performance degradation is a nonstationarity of the beamformer training data, which may be caused by fast interferer and antenna motion as well as antenna vibration [53]. The effect of data nonstationarity may severely limit the available training sample size and can lead to a substantial performance degradation even in the case of signal-free training data. This phenomenon is usually called *interference undernulling*.

### 4.3.2 Traditional Robust Beamforming Techniques

A classic technique to improve the robustness of the SMI beamformer against signal response errors and to prevent signal self-nulling is to use additional point or derivative main lobe constraints in the MVDR problem [30, 35]. Although the resulting linearly constrained minimum variance (LCMV) beamformer can efficiently tackle the problem of look direction errors, its serious shortcoming is in that it is restricted by the plane wavefront assumption for the desired signal steering vector.

Another popular approach to improve the robustness of the SMI techniques against signal self-nulling is the *diagonal loading* (DL) method [109–111]. The essence of this method is to regularize the SMI approach by adding a quadratic penalty term to the objective function of the finite-sample MVDR problem. The resulting problem can be expressed as

$$\min_{\mathbf{w}} \mathbf{w}^H \hat{\mathbf{R}} \mathbf{w} + \gamma \mathbf{w}^H \mathbf{w} \quad \text{subject to} \quad \mathbf{w}^H \mathbf{a}_s = 1, \quad (4.47)$$

where  $\gamma$  is the diagonal loading factor. Solving (4.47) and omitting an immaterial scalar, the following loaded SMI (LSMI) beamformer can be obtained [109–111]:

$$\mathbf{w}_{\text{LSMI}} = (\hat{\mathbf{R}} + \gamma \mathbf{I})^{-1} \mathbf{a}_s. \quad (4.48)$$

It can be seen that the LSMI beamformer (4.48) differs from the SMI technique (4.46) only by the scaled identity matrix that is added to  $\hat{\mathbf{R}}$ .

It is well known that diagonal loading substantially improves the output SINR of the SMI technique in scenarios with arbitrary signal array response errors [60, 110]. However, the application of the LSMI technique is limited by the fact that there is no reliable and simple way to properly choose the DL parameter  $\gamma$  because the optimal value of  $\gamma$  is scenario dependent [36].

Another useful approach to prevent signal self-nulling in adaptive arrays is the eigenspace-based beamformer [108]. Its key idea is to improve the signal steering vector by projecting it onto the estimated signal-plus-interference subspace. The weight vector of the eigenspace-based beamformer can be written as

$$\mathbf{w}_{\text{eig}} = \hat{\mathbf{R}}^{-1} \mathbf{P}_{\hat{\mathbf{E}}} \mathbf{a}_s, \quad (4.49)$$

where  $\hat{\mathbf{E}}$  is the matrix that is built from the signal-plus-interference subspace eigenvectors of  $\hat{\mathbf{R}}$ , and  $\mathbf{P}_{\hat{\mathbf{E}}}$  is the projection matrix onto the column space of  $\hat{\mathbf{E}}$ .

If the number of sources is low ( $L \ll M$ ) and their powers are high relative to the noise power, the beamformer (4.49) is known to enjoy a substantially improved robustness against signal array response errors as compared to the SMI beamformer [108]. However, the approach of (4.49) can be subject to a severe degradation when the number of users is comparable with the number of array sensors and when the powers of the signal or some of interferers are low. In the latter case, subspace swap effects may substantially reduce the performance of the eigenspace-based beamformer [36].

In the moving interferer case, several methods have been proposed to improve the beamformer robustness against interference undernulling; see [56, 57, 112–114] and references therein. The key idea of all these methods is to broaden the adaptive beam pattern nulls in unknown directions of interference.

The first class of methods of this type is based on data-driven derivative constraints (DDC) [56, 57]. In the simplest case of a single derivative constraint, the essence of this approach is to replace the matrix  $\hat{\mathbf{R}}$  in the SMI beamformer (or in any other adaptive beamforming technique) by the following modified covariance matrix:

$$\hat{\mathbf{R}}_{\text{DDC}} = \hat{\mathbf{R}} + \zeta \mathbf{B} \hat{\mathbf{R}} \mathbf{B}, \quad (4.50)$$

where  $\mathbf{B}$  is an  $M \times M$  diagonal matrix whose entries depend on the array geometry (see [57] for more details), and a nonnegative parameter  $\zeta$  determines the trade-off between the null depth and width. Under certain mild assumptions, the optimal value of  $\zeta$  can be computed from the known array parameters [57].

The second class of methods based on broadening the beam pattern nulls makes use of data-driven point constraints [112, 113]. These methods exploit *covariance matrix tapering* (CMT) whose essence is to replace the matrix  $\hat{\mathbf{R}}$  in the SMI beamformer (or in any other adaptive beamforming technique) by the tapered covariance matrix

$$\hat{\mathbf{R}}_T = \hat{\mathbf{R}} \odot \mathbf{T}, \quad (4.51)$$

where  $\odot$  stands for the Schur–Hadamard matrix product, and  $\mathbf{T}$  is the  $M \times M$  taper matrix. Proper choices of the matrix  $\mathbf{T}$  are discussed in [112] and [113].

An interesting relationship between the DDC and CMT approaches has been discovered in [114], where it has been shown that the matrix (4.50) can be interpreted as (4.51) with a particular choice of  $\mathbf{T}$ . An efficient online implementation of the CMT approach has been recently reported in [115].

### 4.3.3 Advanced Robust Beamforming Techniques

To avoid the above-mentioned shortcomings of the traditional ad hoc robust beamforming methods, a theoretically rigorous robust MVDR beamforming approach has been recently developed in [116]. The authors of [116] define the array response uncertainty  $\boldsymbol{\delta} \triangleq \tilde{\mathbf{a}}_s - \mathbf{a}_s$  where  $\tilde{\mathbf{a}}_s$  and  $\mathbf{a}_s$  stand for the actual and presumed signal steering vectors, respectively. It is assumed in [116] that the norm of the error vector  $\boldsymbol{\delta}$  is bounded from above by a known constant  $\varepsilon$ . The basic idea of [116] is to incorporate robustness to the conventional MVDR beamforming problem by making use of the worst-case distortionless response constraint that has to be maintained for all the mismatched signal steering vectors in the spherical uncertainty set. Using this constraint, the robust worst-case MVDR beamformer can be found by solving the following problem [116]:

$$\min_{\mathbf{w}} \mathbf{w}^H \hat{\mathbf{R}} \mathbf{w} \quad \text{subject to} \quad |\mathbf{w}^H (\mathbf{a}_s + \boldsymbol{\delta})| \geq 1 \quad \text{for all } \|\boldsymbol{\delta}\| \leq \varepsilon, \quad (4.52)$$

where  $\|\cdot\|$  denotes hereafter the Euclidean norm of a vector or the Frobenius norm of a matrix. In [116], it has been proven that the problem (4.52) can be transformed as

$$\min_{\mathbf{w}} \mathbf{w}^H \hat{\mathbf{R}} \mathbf{w} \quad \text{subject to} \quad \mathbf{w}^H \mathbf{a}_s \geq \varepsilon \|\mathbf{w}\| + 1, \quad (4.53)$$

where the constraint of (4.53) can be shown to be satisfied with equality. The problem (4.53) can be identified as a convex second-order cone programming (SOCP) problem that can be solved with the complexity of  $O(M^3)$  using interior point methods. It has been additionally shown in [116] that (4.53) is equivalent to (4.47) with an *adaptive* choice of the DL factor that is optimally matched to the known amount  $\varepsilon$  of the steering vector uncertainty.

Several useful extensions of the approach of [116] have been recently proposed. In [117], the beamformer of [116] has been extended to the case of ellipsoidal uncertainty. Computationally efficient Newton-type algorithms to solve (4.53) and its extensions have been developed in [117–119], all having the complexity of  $O(M^3)$ . Extensions of the worst-case approach of [116] to the general-rank signal case have been developed in [120].

In [121], the beamformer of (4.53) has been extended to account for interferer nonstationarity in addition to the signal array response errors. The key idea of [121] is, in addition to modeling the uncertainty in the signal steering vector, also to model nonstationarity in the beamformer training data by adding some uncertainty to the data matrix  $\mathbf{X}$ . Let

$$\Delta_x \triangleq \tilde{\mathbf{X}} - \mathbf{X} \quad (4.54)$$

be the uncertainty matrix, where  $\tilde{\mathbf{X}}$  and  $\mathbf{X}$  denote the actual and presumed data matrices, respectively. Note that  $\mathbf{X}$  is the data matrix available to the beamformer, while the actual data matrix  $\tilde{\mathbf{X}}$  can differ from  $\mathbf{X}$  because of nonstationary behavior of the training data

snapshots. The actual sample covariance matrix is given by

$$\hat{\mathbf{R}} = \frac{1}{N} \tilde{\mathbf{X}} \tilde{\mathbf{X}}^H = \frac{1}{N} (\mathbf{X} + \Delta_x) (\mathbf{X} + \Delta_x)^H. \quad (4.55)$$

It has been assumed that the norms of both the spatial signature mismatch  $\delta$  and the data matrix mismatch  $\Delta_x$  are norm bounded by some known constants  $\varepsilon$  and  $\eta$  as

$$\|\delta\| \leq \varepsilon, \quad \|\Delta_x\| \leq \eta. \quad (4.56)$$

Then, the problem of (4.52) can be extended as [121]

$$\min_{\mathbf{w}} \max_{\|\Delta_x\| \leq \eta} \|(\mathbf{X} + \Delta_x)^H \mathbf{w}\| \quad \text{subject to} \quad |\mathbf{w}^H (\mathbf{a}_s + \delta)| \geq 1 \quad \text{for all } \|\delta\| \leq \varepsilon. \quad (4.57)$$

It has been shown in [121] that the latter problem can be converted to an equivalent form

$$\min_{\mathbf{w}} \|\mathbf{X}^H \mathbf{w}\| + \eta \|\mathbf{w}\| \quad \text{subject to} \quad \mathbf{w}^H \mathbf{a}_s \geq \varepsilon \|\mathbf{w}\| + 1. \quad (4.58)$$

Similar to (4.53), the problem of (4.58) is a convex SOCP problem that can be efficiently solved using standard interior point methods.

It has been demonstrated in [116, 118, 121] that the worst-case approaches to robust adaptive beamforming sufficiently outperform the traditional ad hoc robust beamforming techniques.

Several further probabilistic extensions of the deterministic worst-case beamforming methods of [116, 117] have been recently developed in [122]. The rationale behind these extensions is to replace the “hard” (deterministic) worst-case distortionless response constraints by more flexible “soft” beamformer outage probability constraints. Such probabilistic robust beamforming approaches have been considered in [122] both for the cases of circularly symmetric Gaussian and worst-case distributions of the signal steering vector errors. Interestingly, the resulting probability-constrained robust beamforming problems have been shown in [122] to be rather similar to the deterministic worst-case beamformer designs of [116, 117]. However, an important advantage of the probability-constrained beamformers of [122] with respect to their worst-case counterparts is in that they explicitly quantify the parameter  $\varepsilon$  of the uncertainty region in terms of the beamformer outage probability and, therefore, offer a judicious choice of this parameter.

A useful link between adaptive beamforming and signal detection in multiple-access multiple-input multiple-output (MIMO) systems has been established in [123, 124] (see also [37]). Using the results of [123, 124], the application of robust adaptive beamforming methods of [116, 117] to the problem of signal detection in space–time-coded multiple-access MIMO systems has been considered in [125, 126]. Other useful applications of robust adaptive beamforming methods to different practical problems are discussed in [118].

#### 4.4 CONCLUSIONS

We have presented an overview of selected robustness issues in sensor array processing. In summary, the field of robust array processing remains very research active until now.

Current and future trends in this field are focused on designing computationally efficient robust adaptive beamforming and DOA estimation techniques, developing advanced methods that are able to combine different types of robustness, studying theoretical relationships between different robust approaches, and applying robust array processing techniques to a variety of important practical problems.

## ACKNOWLEDGMENTS

This work was supported by the German Research Foundation (DFG) under Grant GE 1881/1-1 and by the European Research Council (ERC) Advanced Investigator Grants Program.

## REFERENCES

1. L. E. Brennan, J. D. Mallett, and I. S. Reed, "Adaptive arrays in airborne MTI radar," *IEEE Trans. Antennas Propag.*, vol. 24, pp. 607–615, Sept. 1976.
2. S. Haykin (Ed.), *Array Processing: Applications to Radar*, New York: Dowden, Hutchinson, & Ross, 1980.
3. S. Haykin, J. Litva, and T. Shepherd (Eds.), *Radar Array Processing*, New York: Springer-Verlag, 1992.
4. P. Stoica and A. L. Swindlehurst, "Maximum likelihood methods in radar array signal processing," *Proc. IEEE*, vol. 86, pp. 421–441, Feb. 1998.
5. H. Cox, "Resolving power and sensitivity to mismatch of optimum array processors," *J. Acoust. Soc. Am.*, vol. 54, pp. 771–758, 1973.
6. D. R. Morgan and T. M. Smith, "Coherence effects on the detection performance of quadratic array processors with application to large-array matched-field beamforming," *J. Acoust. Soc. Am.*, vol. 87, pp. 737–747, Feb. 1988.
7. J. L. Krolik, "The performance of matched-field beamformers with Mediterranean vertical array data," *IEEE Trans. Signal Process.*, vol. 44, pp. 2605–2611, Oct. 1996.
8. A. B. Gershman, V. I. Turchin, and V. A. Zverev, "Experimental results of localization of moving underwater signal by adaptive beamforming," *IEEE Trans. Signal Process.*, vol. 43, pp. 2249–2257, Oct. 1995.
9. E. Y. Gorodetskaya, A. I. Malekhanov, A. G. Sazontov, and N. K. Vdovicheva, "Deep-water acoustic coherence at long ranges: Theoretical prediction and effects on large-array signal processing," *IEEE J. Ocean Eng.*, vol. 24, pp. 156–171, Apr. 1999.
10. Y. Kameda and J. Ohga, "Adaptive microphone-array system for noise reduction," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 34, pp. 1391–1400, Dec. 1986.
11. Q. G. Liu, B. Champagne, and P. Kabal, "A microphone array processing technique for speech enhancement in a reverberant space," *Speech Commun.*, vol. 18, no. 4, pp. 317–334, June 1996.
12. J. Winters, "Spread spectrum in a four-phase communication system employing adaptive antennas," *IEEE Trans. Commun.*, vol. 30, pp. 929–936, May 1982.
13. L. C. Godara, "Application of antenna arrays to mobile communications. II. Beam-forming and direction-of-arrival considerations," *Proc. IEEE*, vol. 85, pp. 1195–1245, Aug. 1997.
14. T. S. Rapaport (Ed.), *Smart Antennas: Adaptive Arrays, Algorithms, and Wireless Position Location*, New York: IEEE, 1998.
15. J. E. Evans, J. R. Johnson, and D. F. Sun, "Application of advanced signal processing techniques to angle of arrival estimation in ATC navigation and surveillance systems," Technical Report, M.I.T. Lincoln Laboratory, Lexington, MA, 1982.

16. M. G. Amin, L. Zhao, and A. R. Lindsey, "Subspace array processing for the suppression of FM jamming in GPS receivers," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, pp. 80–92, Jan. 2004.
17. J. Capon, R. J. Greenfield, and R. J. Kolker, "Multidimensional maximum-likelihood processing for a large aperture seismic array," *Proc. IEEE*, vol. 55, pp. 192–211, Feb. 1967.
18. J. F. Böhme, "Statistical array signal processing of measured sonar and seismic data," in *Proc. SPIE Conf. Advanced Signal Processing Algorithms*, Vol. 2563, pp. 2–20, San Diego, CA, USA, July 1995.
19. D. V. Sidorovitch and A. B. Gershman, "2-D wideband interpolated root-MUSIC applied to measured seismic data," *IEEE Trans. Signal Process.*, vol. 46, pp. 2263–2267, Aug. 1998.
20. U. L. Schwarz, "Mathematical-statistical description of the iterative beam removing technique (method CLEAN)," *Astron. Astrophys.*, vol. 65, pp. 345–356, 1978.
21. A. J. van der Veen, A. Leshem, and A. J. Boonstra, "Array signal processing techniques in radio astronomy," *Exp. Astron.*, vol. 17, nos. 1/3, pp. 231–249, June 2004.
22. Y. Li, J. Razavilar, and K. J. R. Liu, "A high-resolution technique for multidimensional NMR spectroscopy," *IEEE Trans. Biomed. Eng.*, vol. 45, pp. 78–86, Jan. 1998.
23. S. Haykin, "Medical imaging: Perspectives on array signal processing," in *Proc. SPIE Conf. Ultrasonic Imaging and Signal Processing*, Vol. 4687, Feb. 2002.
24. A. Dogandzic and A. Nehorai, "EEG/MEG spatio-temporal dipole source estimation and array design," in *High-Resolution and Robust Signal Processing*, Y. Hua, A. B. Gershman, and Q. Cheng (Eds.), New York: Marcel Dekker, 2004, pp. 393–442.
25. A. M. Zoubir, "Bootstrap multiple tests: An application to optimum sensor location for knock detection," *Appl. Signal Process.*, vol. 1, pp. 120–130, 1994.
26. D. Spielman, A. Paulraj, and T. Kailath, "Eigenstructure approach to directions-of-arrival estimation in IR detector arrays," *Appl. Opt.*, vol. 26, no. 2, pp. 199–202, Jan. 1990.
27. A. Nehorai, B. Porat, and E. Paldi, "Detection and localization of vapor-emitting sources," *IEEE Trans. Signal Process.*, vol. 43, pp. 243–253, Jan. 1995.
28. A. B. Gershman and V. I. Turchin, "Nonwave field processing using sensor array approach," *Signal Process.*, vol. 44, no. 2, pp. 197–210, June 1995.
29. A. Nehorai and A. Jeremic, "Landmine detection and localization using chemical sensor array processing," *IEEE Trans. Signal Process.*, vol. 48, pp. 1295–1305, May 2000.
30. R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Arrays*, New York: Wiley, 1980.
31. B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE Acoust. Speech Signal Process. Mag.*, vol. 5, pp. 4–24, Apr. 1988.
32. S. Haykin (Ed.), *Advances in Spectrum Analysis and Array Processing*, Englewood Cliffs, NJ: Prentice Hall, 1995.
33. D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*, Prentice Hall, Englewood Cliffs, NJ, 1993.
34. H. Krim and M. Viberg, "Two decades of array signal processing research: The parametric approach," *IEEE Signal Process. Mag.*, vol. 13, no. 4, pp. 67–94, July 1996.
35. H. L. Van Trees, *Optimum Array Processing*, New York: Wiley, 2002.
36. Y. Hua, A. B. Gershman, and Q. Cheng (Ed.), *High-Resolution and Robust Signal Processing*, New York: Marcel Dekker, 2003.
37. A. B. Gershman, "Array signal processing," in *Space-Time Wireless Systems—From Array Processing to MIMO Communications*, H. Bölskei, D. Gesbert, C. B. Papadias, and A.-J. van der Veen (Eds.), New York: Cambridge University Press, 2006, pp. 241–260.

38. L. C. Godara, "The effect of phase-shift errors on the performance of an antenna-array beamformer," *IEEE J. Ocean. Eng.*, vol. 10, pp. 278–284, July 1985.
39. N. K. Jablon, "Adaptive beamforming with the generalized sidelobe canceller in the presence of array imperfections," *IEEE Trans. Antennas Propag.*, vol. 34, pp. 996–1012, Aug. 1986.
40. A. Paulraj and T. Kailath, "Direction of arrival estimation by eigenstructure methods with imperfect spatial coherence of wave fronts," *J. Acoust. Soc. Am.*, vol. 83, pp. 1034–1040, Mar. 1988.
41. Y. J. Hong, C.-C. Yeh, and D. R. Ucci, "The effect of a finite-distance signal source on a far-field steering Applebaum array—Two dimensional array case," *IEEE Trans. Antennas Propag.*, vol. 36, pp. 468–475, Apr. 1988.
42. B. Friedlander, "A sensitivity analysis of the MUSIC algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 38, pp. 1740–1751, Oct. 1990.
43. A. L. Swindlehurst and T. Kailath, "A performance analysis of subspace-based methods in the presence of model errors, Part I: The MUSIC algorithm," *IEEE Trans. Signal Process.*, vol. 40, pp. 1758–1774, July 1992.
44. U. Nickel, "On the influence of channel errors on array signal processing methods," *Int. J. Electron. Commun.*, vol. 47, no. 4, pp. 209–219, 1993.
45. Y. Rockah and P. M. Schultheiss, "Array shape calibration using sources in unknown locations—Part 1: Far-field sources," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 35, pp. 286–299, Mar. 1987.
46. A. Paulraj and T. Kailath, "Direction of arrival estimation with unknown sensor gain and phase," in *Proc. ICASSP'85*, Tampa, FL, Apr. 1985, pp. 640–643.
47. A. J. Weiss and B. Friedlander, "Array shape calibration using sources in unknown locations—A maximum likelihood approach," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, pp. 1958–1966, Dec. 1989.
48. D. R. Fuhrmann, "Estimation of sensor gain and phase," *IEEE Trans. Signal Process.*, vol. 42, pp. 77–87, Jan. 1994.
49. A. Weiss and B. Friedlander, "DOA and steering vector estimation using a partially calibrated array," *IEEE Trans. Aerospace Electron. Syst.*, vol. 32, pp. 1047–1057, 1996.
50. B. P. Ng, M. H. Er, and C. Kot, "Array gain/phase calibration techniques for adaptive beamforming and direction finding," *IEE Proc. Radar Sonar Navig.*, vol. 141, pp. 25–29, Feb. 1994.
51. A. Ng and C. M. S. See, "Sensor-array calibration using maximum-likelihood approach," *IEEE Trans. Antennas Propagat.*, vol. 44, pp. 827–835, June 1996.
52. D. Astely, A. L. Swindlehurst, and B. Ottersten, "Spatial signature estimation for uniform linear arrays with unknown receiver gains and phases," *IEEE Trans. Signal Process.*, vol. 47, pp. 2128–2138, Aug. 1999.
53. S. D. Hayward, "Effects of motion on adaptive arrays," *IEE Proc. Radar Sonar Navigat.*, vol. 144, pp. 15–20, Feb. 1997.
54. A. Zeira and B. Friedlander, "Direction finding in time varying arrays," *IEEE Trans. Signal Process.*, vol. 43, pp. 927–937, Apr. 1995.
55. B. Friedlander and A. Zeira, "Eigenstructure-based algorithms for direction finding with time-varying arrays," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, pp. 689–701, Apr. 1996.
56. A. B. Gershman, G. V. Serebryakov, and J. F. Böhme, "Constrained Hung-Turner adaptive beamforming algorithm with additional robustness to wideband and moving jammers," *IEEE Trans. Antennas Propagat.*, vol. 44, pp. 361–367, Mar. 1996.

57. A. B. Gershman, U. Nickel, and J. F. Böhme, "Adaptive beamforming algorithms with robustness against jammer motion," *IEEE Trans. Signal Process.*, vol. 45, pp. 1878–1885, July 1997.
58. A. B. Gershman, C. F. Mecklenbräuker, and J. F. Böhme, "Matrix fitting approach to direction of arrival estimation with imperfect spatial coherence of wavefronts," *IEEE Trans. Signal Process.*, vol. 45, pp. 1894–1899, July 1997.
59. J. Goldberg and H. Messer, "Inherent limitations in the localization of a coherently scattered source," *IEEE Trans. Signal Process.*, vol. 46, pp. 3441–3444, Dec. 1998.
60. A. B. Gershman, "Robust adaptive beamforming in sensor arrays," *Int. J. Electron. Commun.*, vol. 53, pp. 305–314, Dec. 1999.
61. D. Astely and B. Ottersten, "The effects of local scattering on direction of arrival estimation with MUSIC," *IEEE Trans. Signal Process.*, vol. 47, pp. 3220–3234, Dec. 1999.
62. P. Stoica, O. Besson, and A. B. Gershman, "Direction-of-arrival estimation of an amplitude-distorted wavefront," *IEEE Trans. Signal Process.*, vol. 49, pp. 269–276, Feb. 2001.
63. C. R. Sastry, E. W. Kamen, and M. Simaan, "An efficient algorithm for tracking the angles of arrival of moving targets," *IEEE Trans. Signal Process.*, vol. 39, pp. 242–246, Jan. 1991.
64. T. Wigren and A. Eriksson, "Accuracy aspects of DOA and angular velocity estimation in sensor array processing," *IEEE Signal Process. Lett.*, vol. 2, pp. 60–62, Apr. 1995.
65. V. Katkovnik and A. B. Gershman, "Performance study of the local polynomial approximation based beamforming in the presence of moving sources," *IEEE Trans. Antennas Propag.*, vol. 50, pp. 1151–1157, Aug. 2002.
66. O. Besson, F. Vincent, P. Stoica, and A. B. Gershman, "Maximum likelihood estimation for array processing in multiplicative noise environments," *IEEE Trans. Signal Process.*, vol. 48, pp. 2506–2518, Sept. 2000.
67. J. Ringelstein, A. B. Gershman, and J. F. Böhme, "Direction finding in random inhomogeneous media in the presence of multiplicative noise," *IEEE Signal Process. Lett.*, vol. 7, pp. 269–272, Oct. 2000.
68. A. B. Gershman, E. Nemeth, and J. F. Böhme, "Experimental performance of adaptive beamforming in a sonar environment with a towed array and moving interfering sources," *IEEE Trans. Signal Process.*, vol. 48, pp. 246–250, Jan. 2000.
69. A. Paulraj and T. Kailath, "Eigenstructure methods for direction of arrival estimation in the presence of unknown noise fields," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 34, pp. 13–20, Jan. 1986.
70. A. Paulraj and T. Kailath, "Eigenstructure methods for direction of arrival estimation in the presence of unknown noise fields," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 34, pp. 13–20, Feb. 1986.
71. S. Prasad, R. T. Williams, A. K. Mahalanabis, and L. H. Sibul, "A transform-based covariance differencing approach for some classes of parameter estimation problems," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 36, pp. 631–641, May 1986.
72. J. LeCadre, "Parametric methods for spatial signal processing in the presence of unknown colored noise fields," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, pp. 965–983, July 1989.
73. M. Wax, "Detection and localization of multiple sources in noise with unknown covariance," *IEEE Trans. Signal Process.*, vol. 40, pp. 245–249, Jan. 1992.
74. K. M. Wong, J. P. Reilly, Q. Wu, and S. Qiao, "Estimation of the directions of arrival of signals in unknown correlated noise, Part I: The MAP approach and its implementation," *IEEE Trans. Signal Process.*, vol. 40, pp. 2007–2017, Aug. 1992.

75. B. Friedlander and A. J. Weiss, "Direction finding using noise covariance modeling," *IEEE Trans. Signal Process.*, vol. 43, pp. 1557–1567, July 1995.
76. S. A. Vorobyov, A. B. Gershman, and K. M. Wong, "Maximum likelihood direction-of-arrival estimation in unknown noise fields using sparse sensor arrays," *IEEE Trans. Signal Process.*, vol. 53, pp. 34–43, Jan. 2005.
77. M. Pesavento and A. B. Gershman, "Maximum-likelihood direction of arrival estimation in the presence of unknown nonuniform noise," *IEEE Trans. Signal Process.*, vol. 49, pp. 1310–1324, July 2001.
78. K. I. Pedersen, P. E. Mogensen, and B. H. Fleury, "A stochastic model of the temporal and azimuthal dispersion seen at the base station in outdoor propagation environments," *IEEE Trans. Veh. Technol.*, vol. 49, pp. 437–447, Mar. 2000.
79. J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, pp. 2408–2418, Aug. 1969.
80. M. Wax, "Detection and estimation of superimposed signals," PhD dissertation, Stanford University, Stanford, CA, Mar. 1985.
81. J. F. Böhme, "Source parameter estimation by approximate maximum likelihood and nonlinear regression," *IEEE J. Oceanic Eng.*, vol. 10, pp. 206–212, July 1985.
82. R. O. Schmidt, "Multiple emitter location and signal parameter estimation," in *Proc. RADC Spectral Estim. Workshop*, Rome, NY, 1979, pp. 234–258.
83. G. Bienvenu and L. Kopp, "Adaptivity to background noise spatial coherence for high resolution passive methods," *IEEE Proc. ICASSP'80*, Denver, CO, Apr. 1980, pp. 307–310.
84. P. Stoica and A. Nehorai, "MUSIC, maximum likelihood and Cramér-Rao bound," *IEEE Trans. Signal Process.*, vol. 37, pp. 720–741, May 1989.
85. D. H. Johnson and S. R. DeGraaf, "Improving the resolution of bearing in passive sonar arrays by eigenvalue analysis," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 30, pp. 638–647, Aug. 1982.
86. A. J. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *Proc. IEEE ICASSP'83*, Boston, MA, May 1983, pp. 336–339.
87. R. Kumaresan and D. W. Tufts, "Estimating the angles of arrival of multiple plane waves," *IEEE Trans. Aerospace Electron. Syst.*, vol. 19, pp. 134–139, Jan. 1983.
88. D. H. Brandwood, "Noise-space projection: MUSIC without eigenvectors," *IEE Proc. H*, vol. 134, no. 3, pp. 303–309, June 1987.
89. R. Roy and T. Kailath, "ESPRIT—Estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, pp. 984–995, July 1989.
90. S. K. Oh and C. K. Un, "An improved MUSIC algorithm for high resolution array processing," *IEE Electron. Lett.*, vol. 25, no. 22, pp. 1523–1524, Oct. 1989.
91. M. D. Zoltowski, G. M. Kautz, and S. D. Silverstein, "Beamspace root-MUSIC," *IEEE Trans. Signal Process.*, vol. 41, pp. 344–364, Jan. 1993.
92. V. T. Ermolaev and A. B. Gershman, "Fast algorithm for minimum-norm direction of arrival estimation," *IEEE Trans. Signal Process.*, vol. 42, pp. 2389–2394, Sept. 1994.
93. C. P. Mathews and M. D. Zoltowski, "Eigenstructure techniques for 2-D angle estimation with uniform circular arrays," *IEEE Trans. Signal Process.*, vol. 42, pp. 2395–2407, Sept. 1994.
94. P. Stoica, P. Händel, and A. Nehorai, "Improved sequential MUSIC," *IEEE Trans. Aerospace Electron. Syst.*, vol. 31, 1230–1239, Oct. 1995.
95. M. Haardt and J. Nossek, "Unitary ESPRIT: How to obtain increased estimation accuracy with a reduced computational burden," *IEEE Trans. Signal Process.*, vol. 43, pp. 1232–1242, May 1995.

96. A. B. Gershman, "Pseudo-randomly generated estimator banks: A new tool for improving the threshold performance of direction finding," *IEEE Trans. Signal Process.*, vol. 46, pp. 1351–1364, May 1998.
97. M. Pesavento, A. B. Gershman, and M. Haardt, "Unitary root-MUSIC with a real-valued eigendecomposition: A theoretical and experimental performance study," *IEEE Trans. Signal Process.*, vol. 48, pp. 1306–1314, May 2000.
98. A. L. Swindlehurst, B. Ottersten, R. Roy, and T. Kailath, "Multiple invariance ESPRIT," *IEEE Trans. Signal Process.*, vol. 40, pp. 867–881, Apr. 1992.
99. N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, "Parallel factor analysis in sensor array processing," *IEEE Trans. Signal Process.*, vol. 48, pp. 2377–2388, Aug. 2000.
100. A. L. Swindlehurst, P. Stoica, and M. Jansson, "Exploiting arrays with multiple invariances using MUSIC and MODE," *IEEE Trans. Signal Process.*, vol. 49, pp. 2511–2521, Nov. 2001.
101. F. Gao and A. B. Gershman, "A generalized ESPRIT approach to direction-of-arrival estimation," *IEEE Signal Process. Lett.*, vol. 12, pp. 254–257, Mar. 2005.
102. M. Pesavento, A. B. Gershman, and K. M. Wong, "Direction finding using partly calibrated sensor arrays composed of multiple subarrays," *IEEE Trans. Signal Process.*, vol. 50, pp. 2103–2115, Sept. 2002.
103. C. M. S. See and A. B. Gershman, "Direction-of-arrival estimation in partly calibrated subarray-based sensor arrays," *IEEE Trans. Signal Process.*, vol. 52, pp. 329–338, Feb. 2004.
104. S. Abd Elkader, A. B. Gershman, and K. M. Wong, "Rank reduction direction-of-arrival estimators with improved robustness against subarray orientation errors," *IEEE Trans. Signal Process.*, vol. 54, pp. 1951–1955, May 2006.
105. H. Hung and M. Kaveh, "Focusing matrices for coherent signal-subspace processing," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 36, pp. 1272–1281, Aug. 1988.
106. B. Friedlander, "The root-MUSIC algorithm for direction finding with interpolated arrays," *Signal Process.*, vol. 30, pp. 15–25, Jan. 1993.
107. I. S. Reed, J. D. Mallett, and L. E. Brennan, "Rapid convergence rate in adaptive arrays," *IEEE Trans. Aerospace Electron. Syst.*, vol. 10, pp. 853–863, Nov. 1974.
108. D. D. Feldman and L. J. Griffiths, "A projection approach to robust adaptive beamforming," *IEEE Trans. Signal Process.*, vol. 42, pp. 867–876, Apr. 1994.
109. Y. I. Abramovich, "Controlled method for adaptive optimization of filters using the criterion of maximum SNR," *Radio Eng. Electron. Phys.*, vol. 26, pp. 87–95, Mar. 1981.
110. H. Cox, R. M. Zeskind, and M. H. Owen, "Robust adaptive beamforming," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 35, pp. 1365–1376, Oct. 1987.
111. B. D. Carlson, "Covariance matrix estimation errors and diagonal loading in adaptive arrays," *IEEE Trans. Aerospace Electron. Syst.*, vol. 24, pp. 397–401, July 1988.
112. R. J. Mailloux, "Covariance matrix augmentation to produce adaptive array pattern troughs," *IEE Electron. Lett.*, vol. 31, no. 10, pp. 771–772, May 1995.
113. J. R. Guerci, "Theory and application of covariance matrix tapers to robust adaptive beamforming," *IEEE Trans. Signal Process.*, vol. 47, pp. 977–985, Apr. 2000.
114. M. A. Zatman, "Comment on 'Theory and application of covariance matrix tapers for robust adaptive beamforming,'" *IEEE Trans. Signal Process.*, vol. 48, pp. 1796–1800, June 2000.
115. M. Rübsamen, C. Gerlach, and A. B. Gershman, "Low-rank covariance matrix tapering for robust adaptive beamforming," in *Proc. IEEE ICASSP'08*, Las Vegas, Nevada, Apr. 2008.

116. S. Vorobyov, A. B. Gershman, and Z.-Q. Luo, "Robust adaptive beamforming using worst-case performance optimization: A solution to the signal mismatch problem," *IEEE Trans. Signal Process.*, vol. 51, pp. 313–324, Feb. 2003.
117. R. G. Lorenz and S. P. Boyd, "Robust minimum variance beamforming," *IEEE Trans. Signal Process.*, vol. 53, pp. 1684–1696, May 2005.
118. P. Stoica and J. Li (Eds.), *Robust Adaptive Beamforming*, Hoboken, NJ: Wiley, 2006.
119. K. Zarifi, S. Shahbazpanahi, A. B. Gershman, and Z.-Q. Luo, "Robust blind multiuser detection based on the worst-case performance optimization of the MMSE receiver," *IEEE Trans. Signal Process.*, vol. 53, pp. 295–305, Jan. 2005.
120. S. Shahbazpanahi, A. B. Gershman, Z.-Q. Luo, and K. M. Wong, "Robust adaptive beamforming for general-rank signal models," *IEEE Trans. Signal Process.*, vol. 51, pp. 2257–2269, Sept. 2003.
121. S. A. Vorobyov, A. B. Gershman, Z.-Q. Luo, and N. Ma, "Adaptive beamforming with joint robustness against mismatched signal steering vector and interference nonstationarity," *IEEE Signal Process. Lett.*, vol. 11, pp. 108–111, Feb. 2004.
122. S. A. Vorobyov, H. Chen, and A. B. Gershman, "On the relationship between robust minimum variance beamformers with probabilistic and worst-case distortionless response constraints," *IEEE Trans. Signal Process.*, vol. 56, pp. 5719–5724, Nov. 2008.
123. H. Li, X. Lu, and G. B. Giannakis, "Capon multiuser receiver for CDMA systems with space-time coding," *IEEE Trans. Signal Process.*, vol. 50, pp. 1193–1204, May 2002.
124. S. Shahbazpanahi, M. Beheshti, A. B. Gershman, M. Gharavi-Alkhansari, and K. M. Wong, "Minimum variance linear receivers for multiaccess MIMO wireless systems with space-time block coding," *IEEE Trans. Signal Process.*, vol. 52, pp. 3306–3313, Dec. 2004.
125. Y. Rong, S. Shahbazpanahi, and A. B. Gershman, "Robust linear receivers for space-time block coded multi-access MIMO systems with imperfect channel state information," *IEEE Trans. Signal Process.*, vol. 53, pp. 3081–3090, Aug. 2005.
126. Y. Rong, S. A. Vorobyov, and A. B. Gershman, "Robust linear receivers for multi-access space-time block coded MIMO systems: A probabilistically constrained approach," *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 1560–1570, Aug. 2006.

---

## CHAPTER 5

---

# Wireless Communication and Sensing in Multipath Environments Using Multiantenna Transceivers

Akbar M. Sayeed and Thiagarajan Sivanadyan

Wireless Communications Research Laboratory, Department of Electrical and Computer Engineering, University of Wisconsin, Madison, Wisconsin

### 5.1 INTRODUCTION AND OVERVIEW

Multiantenna arrays have emerged as a promising technology for increasing the spectral efficiency and reliability of wireless communication systems by augmenting the traditional signal space dimensions of time and frequency with the spatial dimension [1, 2]. The advantages of such multiple-input multiple-output (MIMO) communication systems are intimately related to the phenomenon of *multipath*—signal propagation over multiple scattering paths—in wireless channels [3–5]. While traditionally considered a detrimental effect due to signal *fading*—wild fluctuations in received signal strength—multipath has emerged as a key source of *diversity* to not only combat the effects of fading for increased reliability but to also increase the information capacity of wireless links [1, 2]. In particular, MIMO systems exploit multipath to establish multiple parallel spatial channels that, in principle, can increase the link capacity in direct proportion to the number of antennas, without increasing the traditional resources of power or bandwidth [6–10]. The advantages of antenna arrays in point-to-point wireless communications have also spurred research in cooperative communication techniques that are aimed at reaping MIMO performance gains in a distributed network setting [11–14].

In this chapter, we review some of the key developments in the last decade aimed at exploitation of multiantenna arrays in wireless communications. Our focus is on the basic structure of wideband MIMO transceivers for optimal communication over multipath wireless channels from the perspective of the spatiotemporal signal space associated with the transceivers. Unlike the traditional additive white Gaussian noise (AWGN) model for point-to-point wireline channels in which the thermal noise at the receiver is the main source of errors, the multipath wireless channel connecting the transmitter and the receiver is best modeled as a *stochastic linear system* due to the large number of physical propagation parameters [1–5]. In particular, the

statistically independent degrees of freedom (DoF) exhibited by the wireless channel, which in turn determine its fundamental performance limits such as capacity, are determined by the *interaction* between the physical multipath propagation environment and the signal space associated with the wireless transceiver. For modern wideband MIMO transceivers this interaction happens in multiple dimensions of time, frequency, and space, and the statistical characteristics of the corresponding multidimensional wireless channel depend on the spatial-temporal-spectral characteristics of the multidimensional waveforms used by the transceivers. We illustrate the key ideas behind multidimensional wireless communication over multipath channels in two contexts: (1) design and analysis of wideband MIMO transceivers for exploitation of multipath diversity in point-to-point links and (2) application of wideband MIMO transceivers for rapid retrieval of information from a network of wireless sensors, emphasizing the importance of multipath diversity and interference suppression in a point-to-multipoint network setting.

In Section 5.2 we review wireless channel modeling in time, frequency, and space. We first develop a physical model for point-to-point wireless channels that explicitly accounts for signal propagation over multiple spatially distributed paths connecting the transmitter and the receiver [3]. Physical models, while accurate, are difficult to incorporate into system design due to their *nonlinear* dependence on a large number of propagation parameters, such as path delays, Doppler shifts, and angles of departure (AoD) at the transmitter and angles of arrival (AoA) at the receiver. The highlight of this section is a *virtual channel representation* of the physical model that captures the interaction between the physical propagation environment and the signal space of the transceivers in time, frequency, and space [15–17]. The virtual representation essentially corresponds to a uniform sampling of the propagation environment in the physical angle–delay–Doppler space commensurate with the spatial-temporal-spectral resolution afforded by the multidimensional signal space afforded by the transceivers. Furthermore, it is *linear* and is characterized by a set of virtual channel coefficients that characterize the statistically independent DoF in the channel. Overall, the virtual channel representation provides a bridge between physical and statistical channel modeling by characterizing the contribution of different propagation paths to the independent DoF in the channel, and greatly facilitates system design and analysis. Our development of physical models and the corresponding sampled virtual representations proceeds along progressively complex scenarios. Section 5.2.1 discusses single-antenna channels, emphasizing channel characteristics in time and frequency. Section 5.2.2 discusses narrowband, static MIMO channels to focus on channel characteristics in space. Section 5.2.3 considers the most general case of time- and frequency-selective MIMO channels to characterize channel characteristics in time, frequency, and space.

In Section 5.3, we discuss transceiver design and analysis for point-to-point wireless links, following the progression in Section 5.2. Our presentation, anchored on the sampled virtual channel representation, is aimed at developing input–output relations for point-to-point wireless links as a function of the spatiotemporal waveforms used for modulation (signaling) at the transmitter and demodulation (matched filtering) at the receiver. Section 5.3.1 considers transceiver structures for single-antenna channels, focusing on exploitation of channel selectivity (variation) in time and frequency. In particular, we develop transceiver structures for two important types of signaling: Fourier signaling used in orthogonal frequency division multiplexing (OFDM) systems,

and spread-spectrum signaling used in code division multiple access (CDMA) systems [3]. We also discuss the concept of orthogonal short-time Fourier (STF) signaling that is a generalization of OFDM signaling adapted to rapidly time-varying channels [18–21]. Section 5.3.2 discusses transceiver structures for nonselective MIMO channels, focusing on the exploitation of spatial channel characteristics. In this context, we emphasize the importance of signaling and reception in *beamspace* for uniform linear arrays (ULAs) of antennas, and its generalization to *eigenspace* signaling and reception for arbitrary array geometries. Section 5.3.3 builds on the previous two sections to develop transceiver structures for the most general case of time- and frequency-selective spatially correlated MIMO channels. In this context, we highlight two types of multidimensional transceiver structures: eigenspace–CDMA transceivers and eigenspace–OFDM/STF transceivers. In general, the design and analysis of wireless transceivers depends on the level of channel state information (CSI) available at the transmitter and/or receiver. Our primary focus in Section 5.3 is on the *coherent case* where *instantaneous CSI* is assumed known perfectly at the receiver to enable coherent reception, whereas only *statistical CSI* is assumed known at the transmitter. Throughout, we illustrate the development of transceiver structures with probability of error and capacity calculations.

In Section 5.4, we discuss an application of the transceiver structures developed in Section 5.3 for information retrieval in wireless sensor networks [22–24]. Wireless sensor networks are an emerging technology that promises an unprecedented ability to monitor the physical environment in a variety of modalities (e.g., acoustic, thermal, chemical) using a network of wireless sensor nodes. Specifically, we discuss a framework for *active wireless sensing* (AWS) in which a wireless information retriever or access point, equipped with a multiantenna array, actively interrogates an ensemble of wireless sensor nodes with wideband (spread-spectrum) space–time waveforms for rapid retrieval of sensor information [25–27]. While originally developed for sensor networks, the AWS framework is also applicable to wireless communication in general point-to-multipoint network settings. We highlight the role of sensor *space–time signatures*, induced by the multipath propagation environment, that greatly facilitate efficient communication between the wireless information retriever (WIR) and the network of sensors. The discussion of AWS also extends the ideas of Section 5.3 to multiuser scenarios, including the discussion of linear techniques for suppressing interference between multiple user (node) transmissions [28]. Finally, the role of time-reversal signaling,<sup>1</sup> a new technique that has been explored recently in the wireless communications [27, 30], is also discussed in the context of multiuser communications.

Finally, in Section 5.5 we provide concluding remarks, including avenues for future research and connections to other chapters in this handbook.

**Notation** Throughout this chapter, we consider complex baseband representations of signals and channels [3]. We use the symbol  $\mathcal{C}$  for the field of complex numbers. Bold-faced letters will be used to represent vectors and matrices; lowercase letters for vectors and uppercase letters for matrices. All vectors are considered column vectors by default. For an  $N$ -dimensional complex vector,  $\mathbf{x} \in \mathcal{C}^N$ ,  $\mathbf{x}^T$  denotes its transpose (a row vector), and  $\mathbf{x}^H = (\mathbf{x}^T)^*$  denotes the Hermitian transpose (transposition and complex

<sup>1</sup>Originally developed in the context of acoustic communications and imaging [29].

conjugation). The same notation is used for matrices as well. Complex Gaussian random variables and vectors will be modeled as *proper (circular) complex Gaussian* [31]. The notation  $x \in \mathcal{CN}(m, \sigma^2)$  is used to denote a complex Gaussian random variable  $x$  with mean  $m$  and variance  $\sigma^2$ . The notation  $\mathbf{x} \in \mathcal{CN}(\mathbf{m}, \Sigma)$  is used to denote a complex Gaussian vector  $\mathbf{x}$  with mean  $\mathbf{m}$  and covariance matrix  $\Sigma = E[(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^H]$ . The notation  $\langle \cdot, \cdot \rangle$  is used to denote the inner product between complex waveforms:  $\langle x, y \rangle = \int x(t)y^*(t) dt$ .

## 5.2 MULTIPATH WIRELESS CHANNEL MODELING IN TIME, FREQUENCY, AND SPACE

Signal propagation over multiple spatially distributed paths, due to scattering from objects in the environment, is a salient feature of wireless channels. The large number of physical multipath propagation parameters necessitates stochastic channel modeling, and wireless channels can be generally modeled as *stochastic linear time-varying* systems [3–5]. The temporal channel variations arise due to the relative motion between the transmitter, receiver, and the multipath scattering environment. In this section, we develop models for capturing statistical characteristics of multipath wireless channels in time, frequency, and space. Traditional models, such as the wide-sense stationary uncorrelated scattering (WSSUS) model [4, 5], are implicitly based on a *rich* scattering environment consisting of infinitely many, densely distributed, propagation paths. In contrast, we will explicitly consider a discrete path model, which is a more accurate reflection of physical reality.

From a communications perspective, we are ultimately interested in characterizing the statistically independent DoF in a wireless channel. The channel DoF characterize: (1) the number of channel parameters to be estimated in practice, (2) the source of fading as well as the level of diversity afforded by the channel, and (3) fundamental channel properties, such as capacity. The channel DoF, in turn, depend on the *interaction* between the physical propagation environment and the signal space of wireless transceivers. For modern wireless transceivers, this interaction happens in multiple dimensions of time, frequency, and space. A highlight of this section is a *virtual modeling framework* for multipath wireless channels that captures this interaction and plays a key role in the subsequent development in this chapter [15–17]. Physically, each propagation path can be represented as a distinct point in angle–delay–Doppler space and the *virtual channel representation* essentially corresponds to a uniform sampling of multipath in angle–delay–Doppler at a resolution commensurate with the spatial-temporal-spectral signal space used by the transceivers. In particular, the sampled virtual channel representation characterizes the contribution of each propagation path to the DoF in the channel: *distinct* virtual channel coefficients are associated with *disjoint* sets of propagation paths that contribute to it, and the number of *dominant* nonvanishing virtual channel coefficients represents the DoF in the channel.

We begin with modeling of single-antenna channels in Section 5.2.1 where the focus is on statistical channel characterization in time and frequency. Section 5.2.2 discusses modeling of slowly time-varying, narrowband MIMO channels to focus on channel statistics in the spatial dimension. Section 5.2.3 discusses the most general case of time-varying, wideband MIMO channels to characterize channel statistics in time,

frequency, and space. In all sections, we first present a physical model for multipath channels, followed by the sampled virtual channel representation to characterize the underlying DoF in the channel. In the context of the virtual representation, we implicitly consider communication using packets of duration  $T$  and (two-sided) bandwidth  $W$ , as in Section 5.3. Thus, the dimension of the temporal signal space is  $N_o \approx TW$ , the time–bandwidth product [32]. For multiantenna channels, we use  $N_T$  and  $N_R$  to denote the number of antennas at the transmitter and the receiver, respectively. Since the focus in this section is on channel characterization, we ignore the issues of noise and interference. The impact of noise and interference is discussed in Sections 5.3 and 5.4.

### 5.2.1 Single-Antenna Channels: Time–Frequency Characteristics

We start with the simplest case of single-antenna channels. In this case, the channel can be generally described as a linear time-varying system [3, 4]:

$$\begin{aligned} r(t) &= \int H(t, f) X(f) e^{j2\pi f t} df = \int_0^{\tau_{\max}} h(t, \tau) x(t - \tau) d\tau \\ &= \int_0^{\tau_{\max}} \int_{-\nu_{\max}/2}^{\nu_{\max}/2} C(\nu, \tau) x(t - \tau) e^{j2\pi \nu t} d\nu d\tau, \end{aligned} \quad (5.1)$$

where  $x(t)$  denotes the transmitted signal,  $X(f)$  is the Fourier transform of  $x(t)$ , and  $r(t)$  is the received signal. The channel is characterized by  $h(t, \tau)$ , the *time-varying impulse response*, or  $H(t, f)$ , the *time-varying frequency response*, or  $C(\nu, \tau)$ , the *delay–Doppler spreading function*. The channel parameters  $\tau_{\max}$  and  $\nu_{\max}$  denote the *delay spread* and (two-sided) *Doppler spread* of the channel:  $\tau_{\max}$  reflects the maximum delay and  $\nu_{\max}/2$  the maximum Doppler shift introduced by the channel. All three channel characterizations are equivalent to each other and related via Fourier transforms. In particular,  $H(t, f)$  and  $C(\nu, \tau)$  are related to  $h(t, \tau)$  as

$$H(t, f) = \int_0^{\tau_{\max}} h(t, \tau) e^{-j2\pi f \tau} d\tau, \quad C(\nu, \tau) = \int h(t, \tau) e^{-j2\pi \nu t} dt. \quad (5.2)$$

Our main interest is in  $H(t, f)$  and  $C(\nu, \tau)$ , which are related through a two-dimensional Fourier transform

$$C(\nu, \tau) = \int \int H(t, f) e^{j2\pi \tau f} e^{-j2\pi \nu t} dt df. \quad (5.3)$$

As noted earlier, the channel is best modeled as a stochastic system due to the underlying multipath propagation. In the WSSUS model [4],  $H(t, f)$  is modeled a two-dimensional WSS process in  $t$  and  $f$  with correlation function

$$R_H(\Delta t, \Delta f) = E[H(t + \Delta t, f + \Delta f) H^*(t, f)]. \quad (5.4)$$

Since  $C(\nu, \tau)$  is related to  $H(t, f)$  via a two-dimensional Fourier transform, it is uncorrelated in both variables:

$$E[C(\nu_1, \tau_1) C^*(\nu_2, \tau_2)] = \Psi_C(\nu_1, \tau_1) \delta(\nu_1 - \nu_2) \delta(\tau_1 - \tau_2), \quad (5.5)$$

where  $\Psi_C(\nu, \tau) \geq 0$  is called the *delay–Doppler scattering function* and can be interpreted as the two-dimensional power spectral density associated with the two-dimensional WSS process  $H(t, f)$ . That is,  $R_H(\Delta t, \Delta f)$  and  $\Psi_C(\nu, \tau)$  are related through a two-dimensional Fourier transform:

$$\Psi_C(\nu, \tau) = \int \int R_H(\Delta t, \Delta f) e^{j2\pi \Delta f \tau} e^{-j2\pi \Delta t \nu} d\Delta t d\Delta f. \quad (5.6)$$

Thus,  $\Psi_C(\nu, \tau)$  is also referred to as the *delay–Doppler power spectrum* associated with the channel, and its support is limited to  $(\nu, \tau) \in [-\nu_{\max}/2, \nu_{\max}/2] \times [0, \tau_{\max}]$ . If the channel is zero mean with (complex) Gaussian statistics, as in Rayleigh fading, then its statistics are completely characterized by the scattering function. Furthermore, we will focus on *underspread* channels for which the channel spread factor  $\tau_{\max} \nu_{\max} \ll 1$ , which is true for virtually all radio frequency wireless channels<sup>2</sup> [3].

**5.2.1.1 Physical Discrete-Path Model** As mentioned earlier, our main focus is on a discrete-path model that explicitly captures channel characteristics in terms of the physical propagation paths. In the discrete-path model,  $H(t, f)$  and  $C(\nu, \tau)$  can be expressed as

$$H(t, f) = \sum_{n=1}^{N_p} \beta_n e^{-j2\pi \tau_n f} e^{j2\pi \nu_n t}, \quad C(\nu, \tau) = \sum_{n=1}^{N_p} \beta_n \delta(\tau - \tau_n) \delta(\nu - \nu_n), \quad (5.7)$$

where  $N_p$  denotes the number of propagation paths from the transmitter to the receiver, and  $\beta_n$ ,  $\tau_n \in [0, \tau_{\max}]$ , and  $\nu_n \in [-\nu_{\max}/2, \nu_{\max}/2]$  denote the complex path gain, delay, and Doppler shift associated with the  $n$ th path, respectively. In particular,  $\beta_n = \alpha_n e^{j\phi_n}$ , where  $\alpha_n \geq 0$  denotes the path amplitude and  $\phi_n$  represents the path phase. Over the (small) time scales of interest, we will assume that  $N_p$  and  $\{\alpha_n, \tau_n, \nu_n\}$  are deterministic but unknown, whereas  $\{\phi_n\}$  are random variables that are all uniformly distributed over  $[-\pi, \pi]$  and are independent for different paths. Thus, the only source of channel randomness is the random path phases,  $\{\phi_n\}$ , the channel variation in time is captured by the path Doppler shifts,  $\{\nu_n\}$ , and the channel variation in frequency is captured by the path delays,  $\{\tau_n\}$ .

For the discrete-path model, the input–output relation in (5.1) becomes

$$r(t) = \sum_{n=1}^{N_p} \beta_n x(t - \tau_n) e^{j2\pi \nu_n t}. \quad (5.8)$$

As evident from (5.1) and (5.8), a multipath wireless channel affects the transmitted signal in both time and frequency.  $C(\nu, \tau)$  reflects this effect in terms of signal *dispersion* in time and frequency: the received signal,  $r(t)$ , is a linear combination of delayed and Doppler-shifted versions of the transmitted signal,  $x(t)$ . The extent of signal dispersion is determined by  $\tau_{\max}$  and  $\nu_{\max}$ . Channel representation in terms of  $H(t, f)$  reflects the effect of the channel in terms of signal *distortion* in time and frequency. In this regard, two measures based on the channel spread parameters are often used to

<sup>2</sup>As opposed to underwater channels based on acoustic communication, which may not be underspread.

capture the *scale* of channel variation in time and frequency [3]:

$$T_{\text{coh}} = \frac{1}{v_{\max}}, \quad W_{\text{coh}} = \frac{1}{\tau_{\max}}, \quad (5.9)$$

where the *coherence time*,  $T_{\text{coh}}$ , represents the duration over which the channel remains strongly correlated in time, whereas the *coherence bandwidth*,  $W_{\text{coh}}$ , represents the bandwidth over which the channel remains strongly correlated in frequency. Note that the inverse relationship between  $(T_{\text{coh}}, W_{\text{coh}})$  and  $(v_{\max}, \tau_{\max})$  is consistent with the Fourier relation in (5.6).

The precise way in which the channel affects the transmitted signal depends on two key signaling parameters,  $T$  and  $W$ , relative to  $v_{\max}$  and  $\tau_{\max}$  (or  $T_{\text{coh}}$  and  $W_{\text{coh}}$ , equivalently). In particular, two composite parameters are important in capturing the interaction between the signal space and the channel:  $W\tau_{\max}$  and  $Tv_{\max}$ . These composite parameters provide four different classifications of the effective channel:

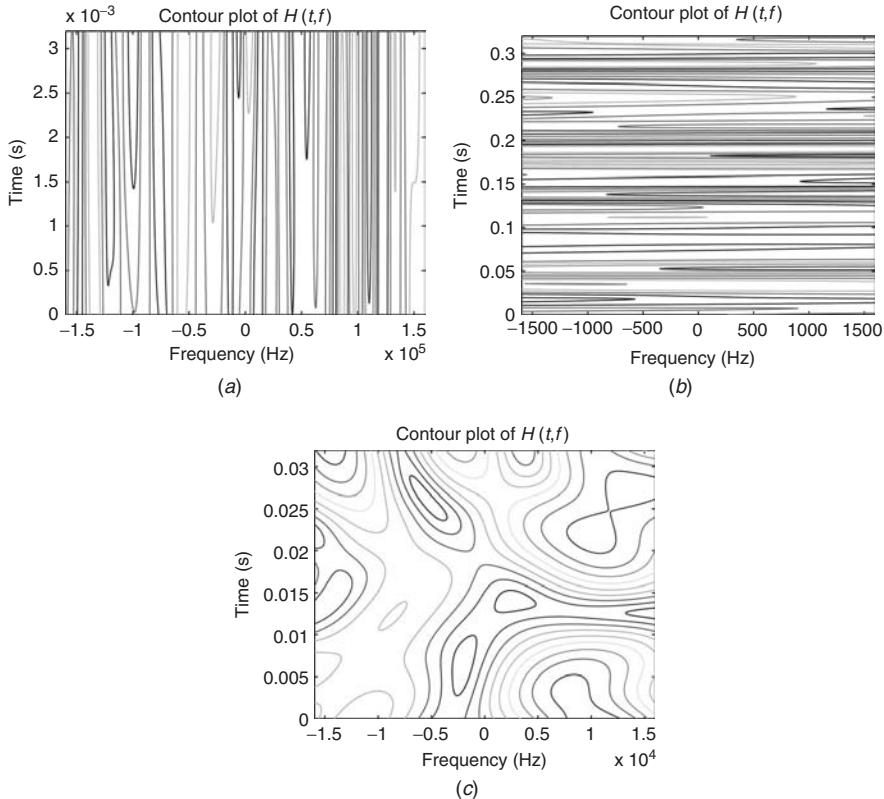
$$\begin{aligned} W\tau_{\max} = \frac{W}{W_{\text{coh}}} &\ll 1 \Leftrightarrow \text{frequency nonselective,} \\ W\tau_{\max} = \frac{W}{W_{\text{coh}}} &> 1 \Leftrightarrow \text{frequency selective,} \end{aligned} \quad (5.10)$$

$$\begin{aligned} T v_{\max} = \frac{T}{T_{\text{coh}}} &\ll 1 \Leftrightarrow \text{time nonselective,} \\ T v_{\max} = \frac{T}{T_{\text{coh}}} &> 1 \Leftrightarrow \text{time selective.} \end{aligned} \quad (5.11)$$

Frequency selectivity means that the signal bandwidth  $W$  is larger than  $W_{\text{coh}}$  so that signal frequencies separated by larger than  $W_{\text{coh}}$  are affected (approximately) independently by the channel. Similarly, time selectivity means that the signal duration  $T$  is larger than  $T_{\text{coh}}$  so that temporal components of the signal, separated by larger than  $T_{\text{coh}}$ , are affected independently by the channel. The level of channel selectivity also reflects the delay or Doppler *diversity* afforded by the channel [15].

Figure 5.1 illustrates the variation in time and frequency in  $H(t, f)$  for different cases of selectivity.  $H(t, f)$  is modeled via (5.7) corresponding to  $N_p = 100$  paths with  $(\tau_n, v_n)$  randomly distributed over a delay spread of  $\tau_{\max} = 0.1$  ms and a Doppler spread of  $v_{\max} = 100$  Hz. Figure 5.1a illustrates a purely frequency-selective channel that exhibits variation only in frequency over the signaling bandwidth  $W$ , 5.1b illustrates a purely time-selective channel that exhibits variation only in time over the signaling duration  $T$ , and 5.1c illustrates a doubly selective channel that exhibits variation in both time and frequency. Different cases correspond to different choices of  $(T, W)$  with a time–bandwidth product of  $N_o = TW = 1024$ .

**5.2.1.2 Virtual Channel Representation: Sampling in Delay–Doppler** The physical discrete-path model (5.8) is nonlinear in the propagation parameters  $(\tau_n, v_n)$ . The key idea behind the sampled channel representation is that the exact values of physical delays and Doppler shifts are not critical from a communication-theoretic perspective—it is the *resolvable* delays and Doppler shifts, at a resolution commensurate with the packet signaling duration  $T$  and bandwidth  $W$ , that are relevant. We assume that  $T \gg \tau_{\max}$  and  $W \gg v_{\max}$  so that interpacket interference in time and frequency is negligible. The sampled virtual representation approximates the physical



**Figure 5.1** Contour plots of  $H(t, f)$  for different cases of selectivity in time and frequency.  $H(t, f)$  is modeled via the physical model with  $N_p = 100$  paths with  $(\tau_n, v_n)$  randomly distributed over a delay spread of  $\tau_{\max} = 10^{-4}$  s and Doppler spread of  $v_{\max} = 100$  Hz, corresponding to  $W_{\text{coh}} = 10^4$  Hz and  $T_{\text{coh}} = 10^{-2}$  s. All cases correspond to signaling with a time–bandwidth product of  $TW = 1024$ . (a) A purely frequency-selective channel corresponding to  $T = 3.2 \times 10^{-3} < T_{\text{coh}}$  and  $W = 3.2 \times 10^5 = 32 \times W_{\text{coh}}$ . (b) A purely time-selective channel corresponding to  $T = 0.32 = 32T_{\text{coh}}$  and  $W = 3.2 \times 10^3 < W_{\text{coh}}$ . (c) A time- and frequency-selective channel with  $T = 3.2 \times 10^{-2} = 3.2T_{\text{coh}}$  and  $W = 3.2 \times 10^4 = 3.2W_{\text{coh}}$ .

model (5.8) with a *linear* channel representation in terms of *resolvable* virtual delays and Doppler shifts [4, 5, 15]:

$$H(t, f) = \sum_{n=1}^{N_p} \beta_n e^{j2\pi v_n t} e^{-j2\pi \tau_n f} \approx \sum_{\ell=0}^{L-1} \sum_{m=-(M-1)}^{M-1} H_v(\ell, m) e^{j2\pi(m/T)t} e^{-j2\pi(\ell/W)f}, \quad (5.12)$$

$$r(t) = \sum_{n=1}^{N_p} \beta_n x(t - \tau_n) e^{j2\pi v_n t} \approx \sum_{\ell=0}^{L-1} \sum_{m=-(M-1)}^{M-1} H_v(\ell, m) x\left(t - \frac{\ell}{W}\right) e^{j2\pi(m/T)t}, \quad (5.13)$$

where  $\Delta\tau = 1/W$  represents the resolution in delay, and  $\Delta\nu = 1/T$  represents the resolution in Doppler afforded by signals of duration  $T$  and bandwidth  $W$ . The sampled representation in (5.13) is a Fourier series representation of  $H(t, f)$  restricted

to  $(t, f) \in [0, T] \times [-W/2, W/2]$  and approximates the physical delays and Doppler shifts,  $(\tau_n, v_n)$ , with uniformly spaced *virtual* delays and Doppler shifts:  $(\hat{\tau}_\ell, \hat{v}_m) = (\ell/W, m/T)$ . Thus, the virtual channel representation in (5.12) and (5.13) is a *linear* representation characterized by the *virtual delay–Doppler channel coefficients*  $\{H_v(\ell, m)\}$ . The approximation by the virtual representation can be made arbitrarily accurate by increasing the number of resolvable delays,  $L$ , and Doppler shifts,  $M$ , included in the summation. However, most of the channel energy is captured by the resolvable delays within the delay and Doppler spreads:

$$L = \lceil W\tau_{\max} \rceil + 1, \quad M = \lceil T v_{\max}/2 \rceil + 1. \quad (5.14)$$

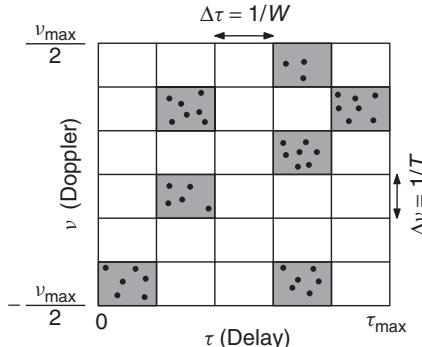
Note that  $L = 1$  for frequency nonselective channels, whereas  $L > 1$  for frequency-selective channels. Similarly,  $M = 1$  for time nonselective channels, whereas  $M > 1$  for time-selective channels.

The virtual channel (Fourier series) coefficients in (5.12) can be computed from  $H(t, f)$  as

$$H_v(\ell, m) = \frac{1}{TW} \int_0^T \int_{-W/2}^{W/2} H(t, f) e^{-j2\pi(m/T)t} e^{j2\pi(\ell/W)f} dt df. \quad (5.15)$$

The virtual representation in delay–Doppler is illustrated in Figure 5.2. Each dot represents a physical propagation path with corresponding  $(\tau_n, v_n)$ , whereas each square represents a *delay–Doppler resolution bin* of size  $\Delta\tau \times \Delta\nu = 1/W \times 1/T$  associated with a virtual channel coefficient  $H_v(\ell, m)$ . The shaded squares represent the *nonzero dominant* virtual coefficients corresponding to resolution bins populated with propagation paths. The nonshaded squares represent resolution bins with no paths contributing to them—the corresponding virtual channel coefficients are nearly zero. This association of virtual coefficients with propagation paths is elaborated next.

**5.2.1.3 Channel Statistics and DoF: Path Partitioning in Delay–Doppler** A key property of the sampled virtual representation is that the  $\{H_v(\ell, m)\}$  partition the



**Figure 5.2** Illustration of the virtual channel representation and path partitioning in delay–Doppler. Each square represents a delay–Doppler resolution bin of size  $\Delta\tau \times \Delta\nu$ , representing a virtual channel coefficient  $H_v(\ell, m)$ . Each shaded square represents a dominant nonzero coefficient with the dots representing the paths contributing to it.

propagation paths into approximately disjoint subsets. Define the following subsets of paths based on their resolution in delay and Doppler:

$$\begin{aligned} S_{\tau,\ell} &= \left\{ n : \frac{\ell}{W} - \frac{1}{2W} < \tau_n \leq \frac{\ell}{W} + \frac{1}{2W} \right\}, \\ S_{v,m} &= \left\{ n : \frac{m}{T} - \frac{1}{2T} < v_n \leq \frac{m}{T} + \frac{1}{2T} \right\}, \end{aligned} \quad (5.16)$$

where  $S_{\tau,\ell}$  denotes the set of paths whose delays,  $\tau_n$ , lie within the  $\ell$ th delay resolution bin in Figure 5.2, and  $S_{v,m}$  is defined similarly in terms of path resolution in Doppler. By substituting the physical model (5.7) in (5.15) it can be shown that the virtual channel coefficients in (5.15) are related to the physical paths as [17]

$$\begin{aligned} H_v(\ell, m) &= \sum_n \beta_n e^{-j\pi(m-v_n T)} \text{sinc} \left[ T \left( \frac{m}{T} - v_n \right) \right] \text{sinc} \left[ W \left( \frac{\ell}{W} - \tau_n \right) \right] \\ &\approx \sum_{n \in S_{v,m} \cap S_{\tau,\ell}} \beta_n, \end{aligned} \quad (5.17)$$

where  $\text{sinc}(x) = \sin(\pi x)/\pi x$  and the last approximation states that  $H_v(\ell, m)$  is approximately the sum of the complex path gains<sup>3</sup> of all paths whose delays and Doppler shifts lie within the *delay–Doppler resolution* of size  $\Delta\tau \times \Delta v$  centered around the  $(m, \ell)$  the virtual delay and Doppler shift, as illustrated in Figure 5.2. It follows that *distinct*  $H_v(\ell, m)$  correspond to approximately<sup>4</sup> *disjoint* subsets of paths, and hence the virtual channel coefficients are approximately statistically independent due to independent path phases. We assume that the virtual coefficients are perfectly independent. Thus, the number of dominant nonvanishing virtual coefficients represents the statistically independent DoF in the channel.

Recall that the only source of randomness in the physical model (5.7) was due to the random path phases. The relationship (5.17) also reveals the rationale for modeling wireless channel coefficients as zero-mean Gaussian random variables (Rayleigh fading): If sufficiently many propagation paths contribute to a virtual channel coefficient  $H_v(\ell, m)$ , then it will exhibit Gaussian statistics due to the central limit theorem.<sup>5</sup> Throughout this chapter we will assume that the virtual channel coefficients exhibit zero-mean complex (proper) Gaussian statistics corresponding to Rayleigh fading [3, 31]. Thus, the channel statistics are characterized by the power in the virtual coefficients:

$$\Psi(\ell, m) = E[|H_v(\ell, m)|^2] \approx \sum_{n \in S_{\tau,\ell} \cap S_{v,m}} E[|\beta_n|^2], \quad (5.18)$$

which represents a sampled version of the delay–Doppler power spectrum in (5.5).

<sup>3</sup>Phase and attenuation factors due to the sinc functions are incorporated into the  $\beta_n$ 's in the approximation in (5.17).

<sup>4</sup>Approximation is due to the finite dimensionality of the signal space and improves with increasing  $T$  and  $W$ .

<sup>5</sup>We also expect from (5.17) that the statistics of  $\{H_v(\ell, m)\}$  will deviate from Gaussian as we increase the signal space resolution, such as through bandwidth as in ultrawideband channels, since very few paths would contribute to a virtual coefficient. This has been observed experimentally for ultrawideband channels; see, for example, [33].

**Special Case: Purely Frequency-Selective Channels** The sampled representation in (5.13) applies to the general case of doubly (time- and frequency-) selective channels. In the special case of purely frequency-selective channels,  $W\tau_{\max} > 1$  and  $T\nu_{\max} \ll 1$ , there is negligible temporal channel variation over the signaling duration,  $H(t, f) \approx H(f)$ , and the physical model and the corresponding virtual representation in (5.13) reduce to

$$r(t) = \sum_{n=1}^{N_p} \beta_n x(t - \tau_n) \approx \sum_{\ell=0}^{L-1} H_v(\ell) x\left(t - \frac{\ell}{W}\right). \quad (5.19)$$

The virtual coefficients resolve the paths only in delay, and the channel statistics are captured by the sampled *delay power spectrum*

$$H_v(\ell) \approx \sum_{n \in S_{\tau,\ell}} \beta_n, \quad \Psi(\ell) \approx \sum_{n \in S_{\tau,\ell}} E[|\beta_n|^2]. \quad (5.20)$$

**Special Case: Purely Time-Selective Channels** In the special case of purely time-selective channels,  $W\tau_{\max} \ll 1$  and  $T\nu_{\max} > 1$ , there is negligible spectral channel variation over the signaling bandwidth,  $H(t, f) \approx H(t)$ , and the physical model and the corresponding virtual representation in (5.13) reduce to

$$r(t) = \sum_{n=1}^{N_p} \beta_n x(t) e^{j2\pi\nu_n t} \approx \sum_{m=-(M-1)}^{M-1} H_v(m) x(t) e^{j2\pi m t/T}. \quad (5.21)$$

The virtual coefficients resolve the paths only in Doppler and the channel statistics are captured by the sampled *Doppler power spectrum*

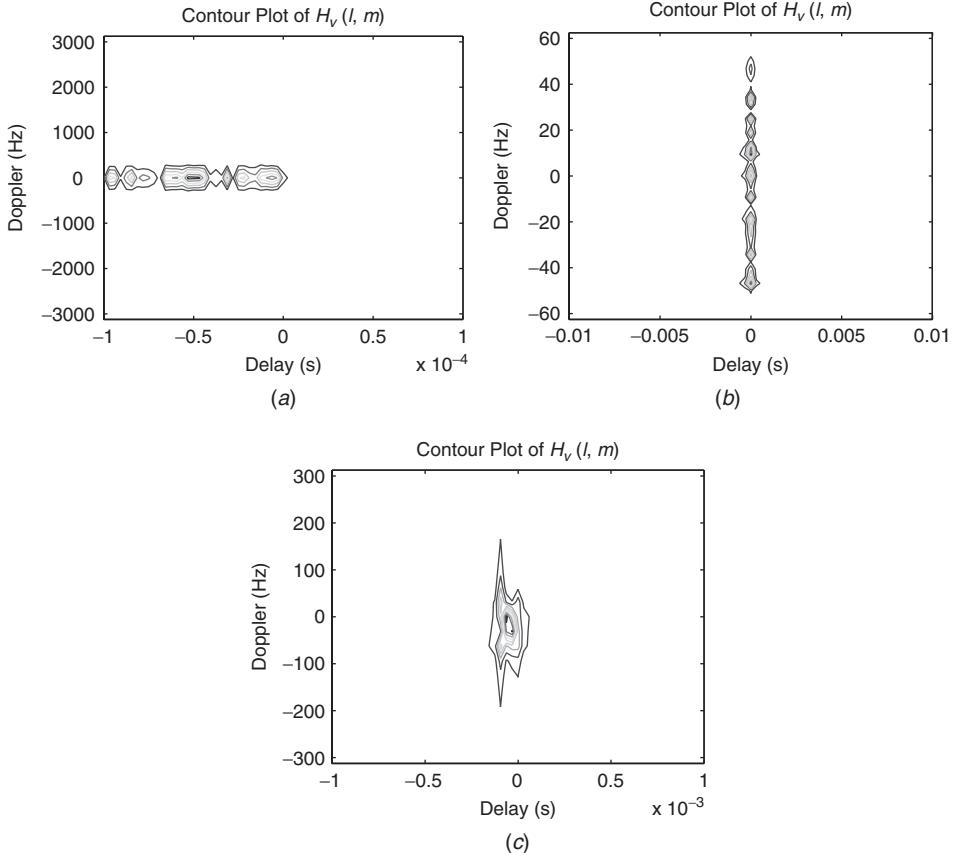
$$H_v(m) \approx \sum_{n \in S_{v,m}} \beta_n, \quad \Psi(m) \approx \sum_{n \in S_{v,m}} E[|\beta_n|^2]. \quad (5.22)$$

Figure 5.3 shows the contour plots of the sampled delay–Doppler coefficients,  $\{H_v(\ell, m)\}$ , for different cases of time and frequency selectivity in  $H(t, f)$  illustrated in Figure 5.1. The sampled coefficients are computed from  $H(t, f)$  using (5.15). Figure 5.3a depicts a purely frequency-selective channel in which the paths are resolvable only in delay, 5.3b depicts a purely time-selective channel in which the paths are resolvable only in Doppler, and 5.3c depicts a doubly selective channel in which the paths are resolvable in both delay and Doppler. Note that the number of resolvable paths, and hence the DoF in the channel, are larger in 5.3a and 5.3b as compared to 5.3c.

### 5.2.2 Nonselective Multiantenna MIMO Channels: Spatial Characteristics

In this section, we focus on spatial modeling of wireless channels. We consider a slowly time-varying, narrowband MIMO channel that is nonselective in time and frequency,  $T\nu_{\max} \ll 1$  and  $W\tau_{\max} \ll 1$ . Consider a system in which the transmitter and receiver are equipped with ULAs with  $N_T$  and  $N_R$  antennas, respectively. The  $N_R$ -dimensional signal vector,  $\mathbf{r}$ , at the receiver is related to the  $N_T$ -dimensional transmitted signal vector,  $\mathbf{x}$ , as

$$\mathbf{r} = \mathbf{Hx}, \quad (5.23)$$



**Figure 5.3** Contour plots of the sampled delay–Doppler channel coefficients  $\{H_v(\ell, m)\}$  corresponding to different cases of selectivity in time and frequency in  $H(t, f)$  depicted in Figure 5.1. (a) Purely frequency-selective channel corresponding to  $T\nu_{\max} = 0.32 < 1$  and  $W\tau_{\max} = 32$  so that a maximum of about  $L = 33$  paths are resolvable in delay. (b) Purely time-selective channel corresponding to  $W\tau_{\max} = 0.32 < 1$  and  $T\nu_{\max} = 32$  so that a maximum of about  $2M = 32$  paths are resolvable in Doppler. (c) A time- and frequency-selective channel corresponding to  $T\nu_{\max} = 3.2$  and  $W\nu_{\max} = 3.2$  so that a maximum of about  $L(2M - 1) = 20$  paths are resolvable in delay and Doppler. As in Figure 5.1, different cases correspond to different choices of  $(T, W)$  with  $TW = 1024$ .

where the  $N_R \times N_T$  matrix  $\mathbf{H}$  represents the spatial channel coupling the transmitter and receiver arrays. Due to multipath fading,  $\mathbf{H}$  is generally modeled as a stochastic matrix. Initial models assumed that the elements of  $\mathbf{H}$  are independent and identically distributed (i.i.d.) zero-mean complex Gaussian random variables, representing a *rich scattering* environment with a large number of propagation paths [6–9]. Since then, it has been realized that physical MIMO channels exhibit spatial correlation, and a number of modeling approaches have been proposed in the literature; the reader is referred to [34] for a recent survey. In this section, we develop the *virtual MIMO*

*channel representation* for ULAs [16]. Its extension to non-ULAs [35, 36] is discussed in Section 5.3.2 when we discuss MIMO transceiver structures.

**5.2.2.1 Physical Model** As in the single-antenna case, a nonselective physical MIMO channel can be accurately modeled as [16]

$$\mathbf{H} = \sum_{n=1}^{N_p} \beta_n \mathbf{a}_R(\theta_{R,n}) \mathbf{a}_T^H(\theta_{T,n}), \quad (5.24)$$

which represents signal propagation over  $N_p$  paths with  $\beta_n$  denoting the complex path gain,  $\theta_{T,n}$  the AoD at the transmitter, and  $\theta_{R,n}$  the AoA at the receiver associated with the  $n$ th path. The vectors  $\mathbf{a}_T(\theta_T)$  and  $\mathbf{a}_R(\theta_R)$  denote the array *steering* and *response* vectors, respectively, for transmitting or receiving a signal in the direction  $\theta_T$  or  $\theta_R$ , respectively:

$$\begin{aligned} \mathbf{a}_T(\theta_T) &= [1, e^{-j2\pi\theta_T}, \dots, e^{-j2\pi\theta_T(N_T-1)}]^T, \\ \mathbf{a}_R(\theta_R) &= [1, e^{-j2\pi\theta_R}, \dots, e^{-j2\pi\theta_R(N_R-1)}]^T, \end{aligned} \quad (5.25)$$

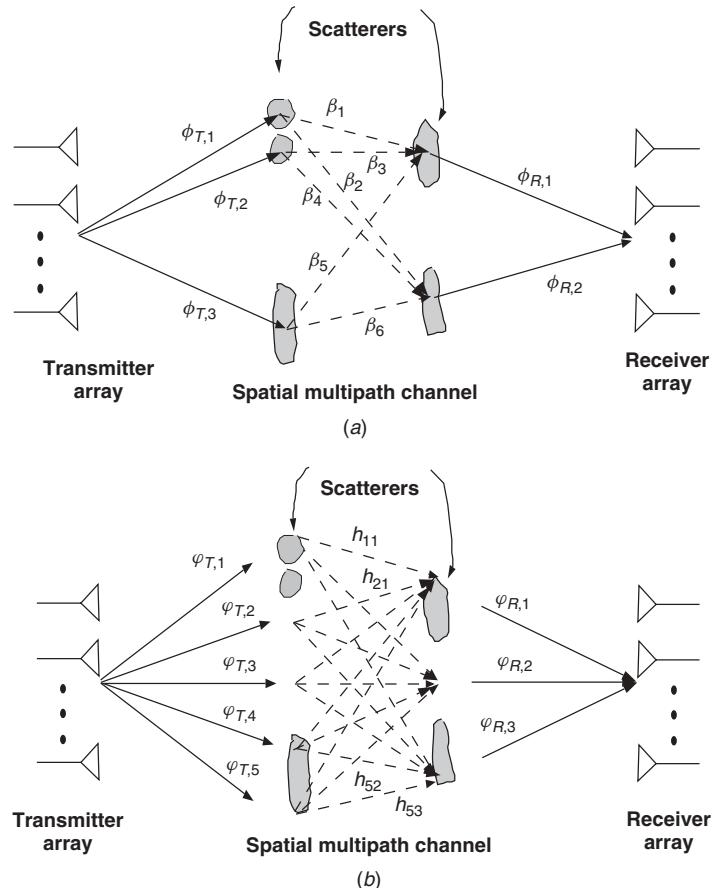
where the normalized angle  $\theta$  is related to the physical angle  $\phi$ , measured relative to broadside, as (see Fig. 5.4a)

$$\theta_T = \frac{d_T}{\lambda} \sin(\phi_T), \quad \theta_R = \frac{d_R}{\lambda} \sin(\phi_R), \quad (5.26)$$

where  $d_T$  and  $d_R$  denote the antenna spacings at the transmitter and receiver, respectively, and  $\lambda$  denotes the wavelength of propagation. We assume maximum angular spreads for AoAs and AoDs,  $\phi \in [-\pi/2, \pi/2]$ , and critical antenna spacing,  $d = \lambda/2$ . In this case,  $\theta \in [-\frac{1}{2}, \frac{1}{2}]$  in (5.25), and there is a one-to-one correspondence between  $\theta$  and  $\phi$ . The effect of antenna spacing on spatial channel characteristics is discussed in [16, 37, 38]. Compared to (5.7), the paths in (5.24) are distinguished by their AoAs and AoDs, and the path delays and Doppler shifts do not come into play due to the nonselective nature of the channel in time and frequency. Physical MIMO channel modeling is illustrated in Figure 5.4a.

**5.2.2.2 Virtual Channel Representation: Sampling in Angle** The physical MIMO channel model in (5.24) depends on the AoAs and AoDs in a nonlinear fashion and thus makes communication-theoretic analysis and design of MIMO communication systems challenging. While the precise knowledge of AoAs and AoDs is important in classical array processing, it is not critical from a communication perspective since the ultimate goal is to reliably communicate over the channel. The virtual MIMO channel representation is based on this motivation and corresponds to *sampling* the spatial scattering environment at uniformly spaced *virtual* AoAs and AoDs:

$$\mathbf{H} = \frac{1}{\sqrt{N_T N_R}} \sum_{i=1}^{N_R} \sum_{k=1}^{N_T} H_v(i, k) \mathbf{a}_R \left( \frac{i}{N_R} \right) \mathbf{a}_T^H \left( \frac{k}{N_T} \right) = \mathbf{A}_R \mathbf{H}_v \mathbf{A}_T^H, \quad (5.27)$$



**Figure 5.4** Illustration of the virtual channel representation in space. (a) Physical channel modeling. The  $n$ th propagation path is associated with a physical angle of departure (AoD),  $\phi_{T,n} = \arcsin(\lambda\theta_{T,n}/d_T)$ , at the transmitter and a physical angle of arrival (AoA),  $\phi_{R,n} = \arcsin(\lambda\theta_{R,n}/d_R)$ , at the receiver. (b) Virtual channel modeling. At the transmitter, the scattering environment is sampled at fixed virtual AoDs,  $\varphi_{T,i} = \arcsin(\lambda(i/N_T)/d_T)$ , and at the receiver the scattering environment is sampled at fixed virtual AoAs,  $\varphi_{R,k} = \arcsin(\lambda(k/N_R)/d_R)$ .

where the matrices  $\mathbf{A}_R$  and  $\mathbf{A}_T$

$$\begin{aligned}\mathbf{A}_R &= \frac{1}{\sqrt{N_R}} \left[ \mathbf{a}_R \left( \frac{1}{N_R} \right), \mathbf{a}_R \left( \frac{2}{N_R} \right), \dots, \mathbf{a}_R(1) \right], \\ \mathbf{A}_T &= \frac{1}{\sqrt{N_T}} \left[ \mathbf{a}_T \left( \frac{1}{N_T} \right), \mathbf{a}_T \left( \frac{2}{N_T} \right), \dots, \mathbf{a}_T(1) \right]\end{aligned}\quad (5.28)$$

are unitary discrete Fourier transform (DFT) matrices whose columns correspond to array response and steering vectors, respectively, at uniformly spaced virtual angles. The spacing between the virtual angles is determined by the array resolutions:  $\Delta\theta_R = 1/N_R$  and  $\Delta\theta_T = 1/N_T$ . As a result, the virtual MIMO channel representation is *linear* and is characterized by the virtual channel matrix  $\mathbf{H}_v$  with elements  $\{H_v(i, k)\}$ . The

virtual MIMO channel representation is a unitarily equivalent representation of  $\mathbf{H}$ , and  $\mathbf{H}_v$  can be computed from  $\mathbf{H}$  via a two-dimensional DFT:

$$\begin{aligned}\mathbf{H}_v &= \mathbf{A}_R^H \mathbf{H} \mathbf{A}_T, \\ H_v(i, k) &= \frac{1}{\sqrt{N_R N_T}} \mathbf{a}_R^H \left( \frac{i}{N_R} \right) \mathbf{H} \mathbf{a}_T \left( \frac{k}{N_T} \right) \\ &= \frac{1}{\sqrt{N_T N_R}} \sum_{\ell=0}^{N_R-1} \sum_{m=0}^{N_T-1} e^{j2\pi \frac{i}{N_R} \ell} H(\ell, m) e^{-j2\pi (k/N_T)m},\end{aligned}\quad (5.29)$$

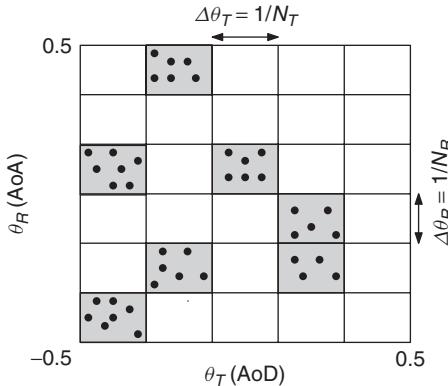
where  $H(\ell, m)$  represent the entries of  $\mathbf{H}$  in the antenna (spatial) domain. The virtual MIMO channel representation is illustrated in Figure 5.4b. As evident from (5.29) and Figure 5.4b, the virtual channel matrix  $\mathbf{H}_v$  is a representation of  $\mathbf{H}$  in beamspace (angle domain).

**5.2.2.3 Channel Statistics and DoF: Path Partitioning in Angle** The virtual MIMO channel representation partitions the propagation paths into approximately disjoint subsets in terms of AoAs and AoDs, as illustrated in Figure 5.5. Define the following subsets of paths associated with  $H_v(i, k)$  based on their resolution in angle:

$$\begin{aligned}S_{\theta_R, i} &= \left\{ n : \frac{i}{N_R} - \frac{1}{2N_R} < \theta_{R,n} \leq \frac{i}{N_R} + \frac{1}{2N_R} \right\}, \\ S_{\theta_T, k} &= \left\{ n : \frac{k}{N_T} - \frac{1}{2N_T} < \theta_{T,n} \leq \frac{k}{N_T} + \frac{1}{2N_T} \right\}.\end{aligned}\quad (5.30)$$

By substituting (5.24) in (5.29), it can be shown that [16]

$$H_v(i, k) \approx \sum_{n \in S_{\theta_R, i} \cap S_{\theta_T, k}} \beta_n, \quad (5.31)$$



**Figure 5.5** Illustration of the virtual channel representation and path partitioning in angle. Each square represents an angle resolution bin of size  $\Delta\theta_R \times \Delta\theta_T$  corresponding to a virtual channel coefficient  $H_v(i, k)$ . Each shaded square represents a dominant nonzero coefficients with the dots representing the paths contributing to it.

which states that each  $H_v(i, k)$  is approximately equal to the sum of the complex gains of all physical paths whose AoAs and AoDs lie within an *angle resolution bin* of size  $\Delta\theta_R \times \Delta\theta_T$  centered around the sample point  $(i/N_R, k/N_T)$  in the  $(\theta_R, \theta_T)$  space. It follows that *distinct*  $H_v(i, k)$ 's correspond to approximately *disjoint* subsets of paths and hence the virtual channel coefficients are approximately statistically independent. For Rayleigh fading, the channel statistics are characterized by the power in the virtual coefficients:

$$\Psi(i, k) = E[|H_v(i, k)|^2] \approx \sum_{n \in S_{\theta_R, i} \cap S_{\theta_T, k}} E[|\beta_n|^2], \quad (5.32)$$

which represents a sampled *angular power spectrum*, and the number of dominant virtual coefficients represents the statistically independent DoF in the channel. Since  $\mathbf{H}$  and  $\mathbf{H}_v$  are unitarily equivalent, it follows that the commonly used i.i.d. model for  $\mathbf{H}$  is equivalent to  $\mathbf{H}_v$  having i.i.d. entries. In general, the entries of  $\mathbf{H}_v$  are independent but not identically distributed and result in correlation in the entries of  $\mathbf{H}$ . In particular, for ULAs it can be shown that the elements of  $\mathbf{H}$  are samples of a two-dimensional stationary process, and the elements of  $\mathbf{H}_v$  correspond to samples of its corresponding spectral representation [16].

### 5.2.3 Time- and Frequency-Selective MIMO Channels: Spatial-Temporal-Spectral Characteristics

We now integrate the development in Sections 5.2.1 and 5.2.2 to consider the most general case of a time- and frequency-selective ( $T\nu_{\max} > 1$ ,  $W\tau_{\max} > 1$ ) spatially correlated MIMO channel corresponding to a transmitter with  $N_T$  antennas and a receiver with  $N_R$  antennas. We assume ULAs of antennas and implicitly consider communication using packets of duration  $T$  and (two-sided) bandwidth  $W$ , as in Section 5.3. In the absence of noise, the transmitted and received signal are related as

$$\mathbf{r}(t) = \int_{-W/2}^{W/2} \mathbf{H}(t, f) \mathbf{X}(f) e^{j2\pi f t} df, \quad 0 \leq t \leq T, \quad (5.33)$$

where  $\mathbf{r}(t)$  is the  $N_R$ -dimensional received signal,  $\mathbf{X}(f)$  is the Fourier transform of the  $N_T$ -dimensional transmitted signal  $\mathbf{x}(t)$ , and  $\mathbf{H}(t, f)$  is the  $N_R \times N_T$  time-varying frequency response matrix that characterizes the channel.

**5.2.3.1 Physical Model** A physical doubly selective MIMO wireless channel can be accurately modeled as [17, 39]

$$\mathbf{H}(t, f) = \sum_{n=1}^{N_p} \beta_n \mathbf{a}_R(\theta_{R,n}) \mathbf{a}_T^H(\theta_{T,n}) e^{j2\pi \nu_n t} e^{-j2\pi \tau_n f}, \quad (5.34)$$

which represents signal propagation over  $N_p$  paths;  $\beta_n$  denotes the complex path gain,  $\theta_{T,n}$  the AoD,  $\theta_{R,n}$  the AoA,  $\tau_n$  the delay, and  $\nu_n$  the Doppler shift associated with the  $n$ th path. The vectors  $\mathbf{a}_T(\theta_T)$  and  $\mathbf{a}_R(\theta_R)$  denote the array steering and response vectors defined in (5.25). As discussed earlier, we assume that  $\tau_n \in [0, \tau_{\max}]$  and  $\nu_n \in [-\nu_{\max}/2, \nu_{\max}/2]$  where  $\tau_{\max}$  and  $\nu_{\max}$  denote the delay spread and the Doppler spread of the channel. We also assume maximum angular spreads,  $(\theta_{R,n}, \theta_{T,n}) \in [-\frac{1}{2}, \frac{1}{2}] \times$

$[-\frac{1}{2}, \frac{1}{2}]$ , at critical antenna spacing. Finally, we assume that over the (small) time scales of interest,  $\{\theta_{T,n}, \theta_{R,n}, \tau_n, v_n\}$  remain fixed; channel variation time is captured by the path Doppler shifts, channel variation in frequency is captured by the path delays, and the channel randomness is captured by the independent complex path gains  $\{\beta_n\}$ .

### 5.2.3.2 Virtual Channel Representation: Sampling in Angle–Delay–Doppler

While accurate (nonlinear) estimation of AoAs, AoD, delays, and Doppler shifts is critical in radar imaging applications, it is not crucial in a communications context. Studying the key communication-theoretic characteristics of time-varying, wideband MIMO channels is greatly facilitated by a linear *virtual representation* of the physical model (5.34) [17, 39]:

$$\mathbf{H}(t, f) \approx \sum_{i=1}^{N_R} \sum_{k=1}^{N_T} \sum_{\ell=0}^{L-1} \sum_{m=-M-1}^{M-1} H_v(i, k, \ell, m) \mathbf{a}_R \left( \frac{i}{N_R} \right) \mathbf{a}_T^H \left( \frac{k}{N_T} \right) e^{j2\pi(m/T)t} e^{-j2\pi(\ell/W)f}, \quad (5.35)$$

which, comparing (5.34) and (5.35), corresponds to sampling the physical angle–delay–Doppler space at uniformly spaced virtual AoAs, AoDs, delays, and Doppler shifts at a resolution determined by the signal space:

$$\Delta\theta_R = \frac{1}{N_R}, \quad \Delta\theta_T = \frac{1}{N_T}, \quad \Delta\tau = \frac{1}{W}, \quad \Delta\nu = \frac{1}{T}. \quad (5.36)$$

In (5.35),  $L = \lceil W\tau_{\max} \rceil + 1$  and  $M = \lceil T\nu_{\max}/2 \rceil + 1$  denote the maximum number of resolvable delays and Doppler shifts within the channel spreads. For maximum angular spreads,  $N_T$  and  $N_R$  reflect the maximum number of resolvable AoDs and AoAs. In essence, the virtual representation in (5.35) is a (linear) Fourier series representation of  $\mathbf{H}(t, f)$  in time, frequency, and space, characterized by the virtual (Fourier) channel coefficients  $\{H_v(i, k, \ell, m)\}$  which can be computed from  $\mathbf{H}(t, f)$  as

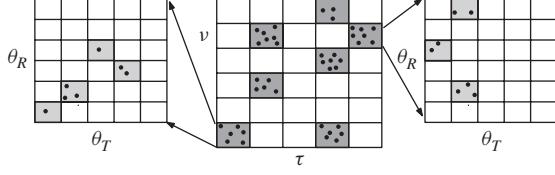
$$H_v(i, k, \ell, m) = \frac{1}{\sqrt{N_R N_T T W}} \int_0^T \int_{-W/2}^{W/2} \mathbf{a}_R^H \left( \frac{i}{N_R} \right) \mathbf{H}(t, f) \mathbf{a}_T \left( \frac{k}{N_T} \right) \times e^{-j2\pi(m/T)t} e^{j2\pi(\ell/W)f} dt df. \quad (5.37)$$

### 5.2.3.3 Channel Statistics and DoF: Path Partitioning in Angle–Delay–Doppler

A key property of the virtual representation is that  $\{H_v(i, k, \ell, m)\}$  partition the propagation paths into approximately disjoint subsets based on their resolution in angle–delay–Doppler, as illustrated in Figure 5.6. By substituting (5.34) in (5.37), it can be shown that [16, 17]

$$H_v(i, k, \ell, m) \approx \sum_{n \in S_{\theta_R, i} \cap S_{\theta_T, k} \cap S_{\tau, \ell} \cap S_{v, m}} \beta_n, \quad (5.38)$$

where  $S_{\tau, \ell}$  and  $S_{v, m}$  are defined in (5.16) and  $S_{\theta_T, i}$  and  $S_{\theta_R, k}$  are defined in (5.30). The above relation states that each  $H_v(i, k, \ell, m)$  is approximately equal to the sum of the complex path gains of all physical paths whose angles, delays, and Doppler shifts lie



**Figure 5.6** Illustration of the virtual channel representation and path partitioning in angle–delay–Doppler. The large box in the center represents the virtual representation in *delay–Doppler* as in Figure 5.2, with each square representing a delay–Doppler resolution bin of size  $\Delta\tau \times \Delta\nu$ , corresponding to a delay–Doppler virtual channel coefficient,  $H_v(\ell, m)$ . Within the large box, each darkly shaded (red) square represents a dominant nonzero delay–Doppler coefficient with the dots representing the paths contributing to it. The smaller boxes on the left and right represent virtual representation in angle, as in Figure 5.5, for two dominant delay–Doppler resolution bins. The smaller squares in these boxes represent angle resolution bins of size  $\Delta\theta_R \times \Delta\theta_T$ . The paths contributing to a particular dominant delay–Doppler coefficient,  $H_v(\ell_o, m_o)$ , in the central box are further resolved in angle to yield the corresponding coefficients in angle–delay–Doppler,  $\{H_v(i, k, \ell_o, m_o)\}$ , represented by smaller squares in the boxes on the left and right. The dominant nonvanishing angle–delay–Doppler coefficients are represented by shaded (green) squares with dots representing the paths contributing to them.

within an *angle–delay–Doppler resolution bin* of size  $\Delta\theta_R \times \Delta\theta_T \times \Delta\tau \times \Delta\nu$  centered around the sample point  $(i/N_R, k/N_T, \ell/W, m/T)$  in the  $(\theta_R, \theta_T, \tau, \nu)$  space. It follows that *distinct*  $H_v(i, k, \ell, m)$ 's correspond to approximately<sup>6</sup> *disjoint* subsets of paths, and hence the virtual channel coefficients are approximately statistically independent due to independent path phases. We assume that the virtual coefficients are perfectly independent. Furthermore, as discussed earlier, we assume that the virtual channel coefficients are zero-mean complex Gaussian (Rayleigh fading) and thus the channel statistics are characterized by the *power* in the virtual coefficients

$$\Psi(i, k, \ell, m) = E[|H_v(i, k, \ell, m)|^2] \approx \sum_{n \in S_{\theta_R, i} \cap S_{\theta_T, k} \cap S_{\tau, \ell} \cap S_{\nu, m}} E[|\beta_n|^2], \quad (5.39)$$

which represents a sampled *angle–delay–Doppler power spectrum*. Throughout the chapter, we assume that  $H_v(i, k, \ell, m) \sim \mathcal{CN}(0, \Psi(i, k, \ell, m))$  corresponding to Rayleigh fading and the different coefficients are statistically independent.

We note that for a fixed  $(\ell_o, m_o)$ , the corresponding set of angle–delay–Doppler virtual channel coefficients,  $\{H_v(i, k, \ell_o, m_o)\}$ , further partitions the paths in  $S_{\tau, \ell_o} \cap S_{\nu, m_o}$ , corresponding to the  $(\ell_o, m_o)$ th delay–Doppler resolution bin, in angle. This is illustrated in Figure 5.6. Thus, as we increase the signal space dimension (by increasing  $T$ ,  $W$ ,  $N_T$ , and/or  $N_R$ ), the paths get resolved at a progressively finer resolution. As a result, some of the virtual channel coefficients may not have any paths contributing to them. This leads to the notion of *dominant* virtual channel coefficients that define the true DoF in the channel.

Let  $\mathcal{S}_D$  denote the set of indices of *dominant* virtual channel coefficients

$$\mathcal{S}_D = \{(i, k, \ell, m) : |\Psi(i, k, \ell, m)| > \epsilon\} \quad (5.40)$$

<sup>6</sup>Approximation is due to the finite dimensionality of the signal space and improves with increasing  $N_T$ ,  $N_R$ ,  $T$ , and  $W$ .

for some appropriately chosen  $\epsilon > 0$ .<sup>7</sup> The number of dominant virtual channel coefficients,  $D = |\mathcal{S}_D|$ , represents the statistically independent DoF in the channel. The CSI of each link is captured by the  $D$  dominant virtual coefficients  $\{H_v(i, k, \ell, m)\}_{\mathcal{S}_D}$ .  $D$  also reflects the level of spatial-temporal-spectral diversity afforded by the channel. Since the virtual coefficients are independent, the *statistical* CSI is captured by the power profile  $\{\Psi(i, k, \ell, m)\}_{\mathcal{S}_D}$  defined in (5.39), whereas the *instantaneous* CSI is captured by the particular realization of  $\{H_v(i, k, \ell, m)\}_{\mathcal{S}_D}$  defined in (5.37) and (5.38). Note that  $D \leq D_{\max} = 2LMN_RN_T$ . In general, the fewer the dominant channel coefficients, the larger the correlation exhibited by the channel in time, frequency, and space.

### 5.3 POINT-TO-POINT MIMO WIRELESS COMMUNICATION SYSTEMS

In this section, we discuss the design and analysis of MIMO transceivers for communication over multipath wireless channels. Our development emphasizes the interaction between the multidimensional signal space and the multipath propagation environment, using insights from the sampled channel representations in Section 5.2. Our development reveals the multidimensional channel structure corresponding to different transceiver configurations; in particular, how channel diversity in angle–delay–Doppler manifests itself and the mechanisms for exploiting it. We begin by discussing single-antenna transceivers in Section 5.3.1, focusing on channel selectivity in time and frequency. We discuss two important forms of temporal signaling: (1) Fourier signaling used in OFDM systems and (2) spread-spectrum signaling used in CDMA systems. In both cases, we discuss both frequency-selective channels and doubly selective channels. In particular, we emphasize the optimality of STF signaling, a generalization of OFDM, for the time- and frequency-selective channels for which the STF or Gabor basis functions serve as approximate eigenfunctions. In Section 5.3.2, we discuss transceiver structures for nonselective spatially correlated MIMO channels to emphasize the role of the spatial dimension in communication. In particular, we emphasize the optimality of beamspace spatial signaling for ULAs, and its generalization, eigenspace signaling, for arbitrary array geometries. Finally, in Section 5.3.3 we integrate the development in the first two sections to discuss multidimensional transceiver structures for the most general case of doubly-selective spatially correlated MIMO channels.

Our focus in this section is on transceiver design for point-to-point communication. We provide pointers for extensions to multiuser settings. Some aspects of multiuser communications and interference are discussed in Section 5.4 in the context of spread-spectrum signaling. Furthermore, our primary focus is on *coherent* reception schemes in which *instantaneous CSI* is assumed known at the receiver. Only *statistical CSI* is assumed known at the transmitter.

#### 5.3.1 Single-Antenna Systems

In this section, we discuss transceiver structures for single-antenna channels. We consider communication using packets of duration  $T$  and bandwidth  $W$  corresponding to a

<sup>7</sup>The choice of  $\epsilon$  is nuanced. An intuitive choice would equal the operating received signal-to-noise ratio (SNR) per dimension—channel coefficients with power below the SNR per dimension do not contribute to the DoF.

temporal signal space with dimension  $N_o \approx TW$  [32]. A key idea in our development is the choice of temporal basis waveforms used for modulation at the transmitter and matched filtering at the receiver [3]. Our focus is on two important classes of modulation waveforms: spread-spectrum (SS) waveforms used in CDMA transceivers and Fourier basis waveforms used in OFDM transceivers. We characterize the input–output relations for the two types of transceivers operating over multipath channels. Our focus is on the practically relevant cases of purely frequency-selective channels, and doubly (time- and frequency-) selective channels. For doubly selective channels, we also introduce the concept of STF signaling, which generalizes the concept of OFDM signaling over frequency-selective channels to doubly selective channels. We first discuss spread-spectrum signaling and then Fourier signaling. Throughout we assume that

$$T \gg \tau_{\max}, \quad W \gg v_{\max} \quad (5.41)$$

so that there is negligible intersymbol interference in time and frequency.

**5.3.1.1 CDMA Transceivers: Spread-Spectrum Signaling** In a CDMA system, each data symbol is modulated onto a spread-spectrum waveform of duration  $T$  of the form [3, 40]

$$q(t) = \sum_{n=1}^{N_o} c[n]v(t - nT_c), \quad (5.42)$$

where  $v(t)$  is chip waveform of duration  $T_c \approx 1/W$ ,  $\{c[n] \in \{-1, 1\} : n = 1, \dots, N_o\}$  is a pseudorandom binary code of length  $N_o = T/T_c \approx TW$ . Thus, there is a one-to-one correspondence between a pseudorandom spreading code  $\{c[n]\}$  and a spread-spectrum waveform  $q(t)$ . Different users in a CDMA system are assigned distinct spreading codes/waveforms. We assume that  $q(t)$  is normalized to have unit energy:  $\int |q(t)|^2 dt = 1$ .

Consider a single-user system. The transmitted CDMA signal takes the form

$$x(t) = \sqrt{\mathcal{E}} \sum_i x_i q(t - iT), \quad (5.43)$$

where  $x_i$  is the  $i$ th data symbol from a given constellation [e.g., binary phase-shift keying (BPSK) or QPSK], and  $\mathcal{E}$  denotes the symbol energy. Under assumption (5.41), symbol-by-symbol detection suffices at the receiver, and we focus on transmission and reception of the 0th symbol, without loss of generality:

$$x(t) = \sqrt{\mathcal{E}} x q(t), \quad 0 \leq t \leq T, \quad (5.44)$$

where  $x$  denotes the transmitted data symbol.

For a purely frequency-selective channel ( $Tv_{\max} \ll 1$ ,  $W\tau_{\max} > 1$ ), the received signal for a single symbol transmission in (5.44) is given by

$$r(t) = \sqrt{\mathcal{E}} x \tilde{q}(t) + w(t), \quad 0 \leq t \leq T + \tau_{\max} \approx T, \quad (5.45)$$

$$\tilde{q}(t) = \sum_{n=1}^{N_p} \beta_n q(t - \tau_n) \approx \sum_{\ell=0}^{L-1} h_{\ell} q\left(t - \frac{\ell}{W}\right), \quad (5.46)$$

where  $\tilde{q}(t)$  in (5.45) denotes the channel-distorted version of  $q(t)$ , and  $w(t)$  is a complex AWGN process. The first equality in (5.46) characterizes  $\tilde{q}(t)$  in terms of the physical model, and the second approximation corresponds to the sampled channel representation in delay [ $\{h_\ell = H_v(\ell)\}$  in (5.19)].

The relation (5.46) states that the received signal is a linear combination of delayed copies of the transmitted spreading waveform  $q(t)$ . The well-known RAKE receiver structure in CDMA systems [3, 40] is based on the sampled channel representation. In coherent reception, assuming that  $\{h_\ell\}$  are known at the receiver, the maximum-likelihood (ML) detector of the transmitted symbol,  $x$ , is based on matched filtering or correlating the received signal  $r(t)$  with  $\tilde{q}(t)$ . Thus, the decision statistic for ML detection of  $x$  is given by

$$\begin{aligned} z &= \langle r(t), \tilde{q}(t) \rangle = \int_0^T r(t) \tilde{q}^*(t) dt = \sum_{\ell=0}^{L-1} h_\ell^* \int_0^T r(t) q^* \left( t - \frac{\ell}{W} \right) dt \\ &= \sum_{\ell=0}^{L-1} h_\ell^* r_\ell = \mathbf{h}^H \mathbf{r} = x \sqrt{\mathcal{E}} \|\mathbf{h}\|^2 + \mathbf{h}^H \mathbf{w}, \end{aligned} \quad (5.47)$$

$$r_\ell = \int r(t) q^* \left( t - \frac{\ell}{W} \right) dt \approx x \sqrt{\mathcal{E}} h_\ell + w_\ell \Leftrightarrow \mathbf{r} = \sqrt{\mathcal{E}} \mathbf{h} x + \mathbf{w}, \quad (5.48)$$

where  $\mathbf{r}$  is the  $L$ -dimensional vector of correlated outputs,  $\{r_\ell\}$ ,  $\mathbf{h}$  is the  $L$ -dimensional vector of sampled channel coefficients,  $\{h_\ell\}$ , and  $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_L)$ . Recall that  $L$  also reflects the level of delay diversity afforded by the channel. The RAKE structure in computing the decision statistic  $z$  corresponds to first correlating  $r(t)$  with delayed versions of  $q(t)$  to compute  $r_\ell$ , and then *coherently* combining the correlator outputs using known values of channel coefficients,  $\{h_\ell\}$ . Relation (5.48) is based on the fact that a spread-spectrum waveform  $q(t)$  is approximately orthogonal to its copies that have been delayed by multiples of chip duration ( $T_c = 1/W$ ) [3]:

$$\int q(t) q^* \left( t - \frac{\ell}{W} \right) dt \approx \delta_\ell. \quad (5.49)$$

The above orthogonality relation is due to the pseudorandom nature of the underlying code  $\{c[n]\}$ , and as a result the noise random variables,  $w_\ell = \langle w(t), q(t - \ell/W) \rangle$ , corrupting the correlator outputs are also approximately independent. Relation (5.48) is based on the assumption of perfect orthogonality between delayed versions of  $q(t)$ .<sup>8</sup>

For a doubly selective channel ( $T\nu_{\max} > 1$ ,  $W\tau_{\max} > 1$ ), the received signal for a single symbol is again given by (5.45), but the channel-distorted waveform  $\tilde{q}(t)$  is now given by

$$\tilde{q}(t) = \sum_{n=1}^{N_p} \beta_n e^{j2\pi\nu_n t} q(t - \tau_n) \approx \sum_{\ell=0}^{L-1} \sum_{m=-(M-1)}^{M-1} h_{\ell,m} q \left( t - \frac{\ell}{W} \right) e^{j2\pi m t / T}, \quad (5.50)$$

<sup>8</sup>Correlation between delayed copies of  $q(t)$  can be readily incorporated by replacing (5.48) with  $\mathbf{r} = x \sqrt{\mathcal{E}} \mathbf{Q} \mathbf{h} + \mathbf{w}$ , where  $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \mathbf{Q})$ , and  $\mathbf{Q}$  is an  $L \times L$  matrix with entries  $\mathbf{Q}_{\ell,\ell'} = \langle q(t - \ell'/W), q(t - \ell/W) \rangle$ ,  $(\ell, \ell') : 0, \dots, L - 1$ . Equation (5.46) implies that  $\mathbf{Q} = \mathbf{I}$ .

where the first equality is based on the physical model, and the second approximation is the sampled representation for doubly selective channels  $\{h_{\ell,m} = H_v(\ell, m)\}$  in (5.13)]. In this case, (5.50) states that the received signal is a linear combination of delayed and Doppler-shifted copies of the transmitted waveform  $q(t)$ . The decision statistic in the ML detector for the transmitted symbol  $x$  is based on a *delay–Doppler RAKE structure* that is a generalization of the delay RAKE structure for doubly selective channels [15]:

$$\begin{aligned} z &= \langle r(t), \tilde{q}(t) \rangle = \int_0^T r(t) \tilde{q}^*(t) dt \\ &= \sum_{\ell=0}^{L-1} \sum_{m=-(M-1)}^{M-1} h_{\ell,m}^* \int_0^T r(t) q^* \left( t - \frac{\ell}{W} \right) e^{-j2\pi mt/T} dt \\ &= \sum_{\ell=0}^{L-1} \sum_{m=-(M-1)}^{M-1} h_{\ell,m}^* r_{\ell,m} = \mathbf{h}^H \mathbf{r} = x \sqrt{\mathcal{E}} \|\mathbf{h}\|^2 + \mathbf{h}^H \mathbf{w}, \\ r_{\ell,m} &= \int r(t) q^* \left( t - \frac{\ell}{W} \right) e^{-j(2\pi mt/T)} dt \approx x \sqrt{\mathcal{E}} h_{\ell,m} + w_{\ell,m} \Leftrightarrow \mathbf{r} = \sqrt{\mathcal{E}} \mathbf{h} x + \mathbf{w}, \end{aligned} \quad (5.51)$$

(5.52)  
where  $\mathbf{r}$  is the  $L(2M-1)$ -dimensional vector of correlated outputs  $\{r_{\ell,m}\}$ ,  $\mathbf{h}$  is the  $L(2M-1)$ -dimensional vector of sampled channel coefficients,  $\{h_{\ell,m}\}$ , and  $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{L(2M-1)})$ . Recall that  $L(2M-1)$  also reflects the delay–Doppler diversity afforded by the channel. The RAKE structure in computing the decision-statistic  $z$  corresponds to first correlating  $r(t)$  with delayed and Doppler-shifted versions of  $q(t)$  to compute  $r_{\ell,m}$ , and then *coherently* combining these correlator outputs using known values of channel coefficients,  $\{h_{\ell,m}\}$ . Relation (5.52) is based on the fact that delayed and Doppler-shifted versions of  $q(t)$  (by multiples of  $\Delta\tau = 1/W$  and  $\Delta\nu = 1/T$ ) are approximately orthogonal to each other [15]:

$$\int q(t) q^* \left( t - \frac{\ell}{W} \right) e^{j2\pi mt/T} dt \approx \delta_\ell \delta_m \quad (5.53)$$

due to the pseudorandom nature of  $q(t)$ .

The transmitted symbol  $x$  in both cases above can be detected using the decision-statistic  $z$  in (5.47) and (5.51). For example,  $x \in \{-1, 1\}$  for BPSK modulation, and the ML detector for  $x$  in both cases is given by

$$\hat{x} = \text{sign}(\text{real}\{z\}) = \text{sign}(\text{real}\{\mathbf{h}^H \mathbf{r}\}), \quad (5.54)$$

and the corresponding probability of error,  $P_e$ , in detecting  $x$  at the receiver is given by

$$P_e(\mathbf{h}) = Q \left( \sqrt{2\mathcal{E} \|\mathbf{h}\|^2} \right), \quad P_e = E[P_e(\mathbf{h})], \quad (5.55)$$

where  $P_e(\mathbf{h})$  denotes the probability of error conditioned on a given realization of  $\mathbf{h}$ ,  $P_e$  denotes the long-term averaged probability of error where the averaging is over the statistics of  $\mathbf{h}$ , and  $Q(\cdot)$  denotes the  $Q$  function representing the tail probability of a

standard Gaussian  $\mathcal{N}(0, 1)$ :

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt. \quad (5.56)$$

In contrast, the  $P_e$  in an AWGN channel with the same average received SNR is given by

$$P_{e,\text{AWGN}} = Q\left(\sqrt{2\mathcal{E}E[\|\mathbf{h}\|^2]}\right). \quad (5.57)$$

Comparing (5.55) and (5.57), we note that  $P_{e,\text{AWGN}}$  corresponds to pushing the expectation inside the argument of the  $Q(\cdot)$  function in (5.55). As a result, the  $P_e$  over a fading channel is always larger than  $P_{e,\text{AWGN}}$  due to fluctuations in instantaneous received SNR

$$\text{SNR}(\mathbf{h}) = \mathcal{E}\|\mathbf{h}\|^2 \quad (5.58)$$

induced by multipath fading.<sup>9</sup>

Recall from our discussion in Section 5.2 that different components of the channel vector  $\mathbf{h}$  are *independent* zero-mean complex Gaussian random variables. It follows that  $\|\mathbf{h}\|^2$  is  $\chi^2$  random variable, and this fact can be used to obtain closed-form expressions for  $P_e$  [3]. We note that for purely frequency-selective channels,  $\|\mathbf{h}\|^2$  is  $\chi^2$  with  $2L$  DoF, representing the  $L$ -fold delay diversity afforded by the multipath channel. For doubly selective channels,  $\|\mathbf{h}\|^2$  is  $\chi^2$  with  $2L(2M - 1)$  DoF, representing the  $L(2M - 1)$ -fold delay–Doppler diversity afforded by the channel [15].

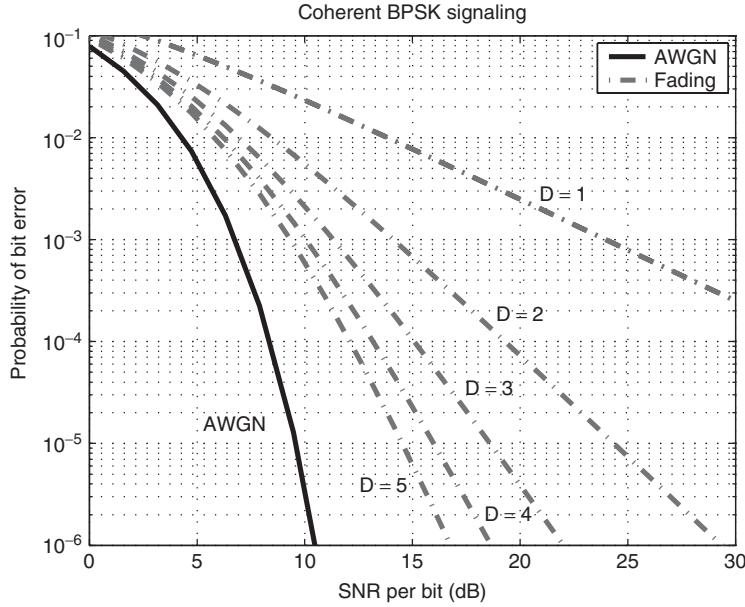
In Figure 5.7,  $P_e$  is plotted as a function of average SNR per bit,  $E[\text{SNR}(\mathbf{h})] = \mathcal{E}$ , for coherent BPSK signaling over a fading channel with different levels of diversity  $D$ . The  $D$  channel coefficients corresponding to the diversity branches are modeled as i.i.d.  $\mathcal{CN}(0, 1/D)$  so that  $E[\|\mathbf{h}\|^2] = 1$ . The total average SNR per bit is kept fixed with increasing values of  $D$ —the average SNR per diversity branch is  $\mathcal{E}/D$ . The performance of BPSK signaling over an AWGN channel is also shown for comparison. Two observations are worth noting. First, there is significant loss in SNR due to fading (the  $D = 1$  plot) compared to an AWGN channel; for example, a loss of about 18 dB in SNR at  $P_e \approx 10^{-3}$ . Second, as we increase the level of diversity,  $D$ ,  $P_e$  approaches the performance over an AWGN channel. In fact, it can be shown that for coherent reception,  $P_e \rightarrow P_{e,\text{AWGN}}$  as  $D \rightarrow \infty$  [3].<sup>10</sup>

From the above development, we conclude that spread-spectrum signaling in CDMA systems facilitates exploitation of multipath diversity at the receiver via the delay RAKE structure in frequency-selective channels and the delay–Doppler RAKE structure in doubly selective channels. Furthermore, using (5.14) we note that the level of diversity increases with increasing  $T$  and  $W$  due to the increased multipath resolution in delay and Doppler, respectively.

We have focused on point-to-point communication in the above discussion. In the multiuser case, the sufficient statistics for detection of symbols of different users are still

<sup>9</sup>The SNR expression corresponds to unit-variance assumption on components of  $\mathbf{w}$ . If  $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I})$ , the SNR expression in (5.58) gets modified to  $\text{SNR}(\mathbf{h}) = \mathcal{E}\|\mathbf{h}\|^2/\sigma^2$ .

<sup>10</sup>In contrast, for noncoherent signaling, such as noncoherent FSK, for a given average SNR per bit,  $\mathcal{E}$ , there is an optimum level of diversity,  $D \approx \mathcal{E}/3$ , that minimizes the  $P_e$ . For larger values of  $D$ , the  $P_e$  starts to increase again [3].



**Figure 5.7**  $P_e$  versus average SNR per bit,  $\mathcal{E}$ , for coherent BPSK signaling over a fading channel with different levels of diversity,  $D$ . The  $D$  independent fading channel coefficients are modeled as i.i.d.  $\mathcal{CN}(0, 1/D)$ . The performance over an AWGN channel with the same average SNR per bit is also shown for comparison.

based on the RAKE receiver structure corresponding to codes of different users. However, different users transmissions interfere and interference suppression techniques need to be applied at the receiver. We refer the reader to [28] for general treatment of multiuser detection techniques and to [41, 42] for multiuser detection techniques based on the sampled delay–Doppler channel representation. Some aspects of multiuser detection in the context of space–time transceivers are discussed in Section 5.4.

**5.3.1.2 ODFM Transceivers: Fourier Signaling** In OFDM signaling, data is modulated onto Fourier basis waveforms [3, 43]. We first develop the system model for purely frequency-selective channels for which Fourier vectors serve as channel eigenfunctions [44]. We then discuss an extension of OFDM, orthogonal STF signaling [21], that is more appropriate for doubly selective channels—STF basis waveforms serve as approximate eigenfunctions for underspread doubly dispersive channels.

Consider a purely frequency-selective single-antenna channel ( $T\nu_{\max} \ll 1$ ,  $W\tau_{\max} > 1$ ). Under assumption (5.41), we again focus on the transmission and reception of a single packet. In OFDM signaling, the transmitted signal for one packet takes the form

$$x(t) = \sqrt{\mathcal{E}} \sum_{m=0}^{N_o-1} x_m \phi_m(t), \quad 0 \leq t \leq T, \quad (5.59)$$

$$\phi_m(t) = \frac{1}{\sqrt{T}} e^{j2\pi \Delta f t} = \frac{1}{\sqrt{T}} e^{j2\pi(m/T)t}, \quad m = 0, \dots, N_o - 1, N_o = TW, \quad (5.60)$$

where  $x_m$  denotes the data modulated onto the  $m$ th Fourier basis waveform,  $\phi_m(t)$ , and the basis functions  $\{\phi_m(t)\}$  in (5.60) form an orthonormal basis for the space of signals of duration  $T$  and bandwidth  $W$ :

$$\langle \phi_m(t), \phi_{m'}(t) \rangle = \int_0^T \phi_m(t) \phi_{m'}^*(t) dt = \delta_{m-m'}. \quad (5.61)$$

The data symbols have normalized average power,  $E[\|\mathbf{x}\|^2] = \sum_m E[|x_m|^2] = 1$ , and  $\mathcal{E}$  denotes symbol energy

$$\int_0^T E[|x(t)|^2] dt = \mathcal{E} \sum_m E[|x_m|^2] = \mathcal{E}. \quad (5.62)$$

The received signal is given by

$$r(t) = \int h(\tau) x(t-\tau) d\tau + w(t) = \sum_m x_m H\left(\frac{m}{T}\right) \phi_m(t) + w(t), \quad (5.63)$$

where  $h(\tau)$  and  $H(f)$  denote the impulse response and frequency response of the multipath channel, and  $w(t)$  denotes a complex AWGN process. At the receiver,  $r(t)$  is projected onto (or correlated with) the Fourier basis functions

$$r_m = \langle r, \phi_m \rangle = \int r(t) \phi_m^*(t) dt = x_m H\left(\frac{m}{T}\right) + w_m, \quad m = 0, \dots, N_o - 1, \quad (5.64)$$

where we have used the orthogonality relation (5.61). Stacking the  $r_m$  into an  $N_o$ -dimensional vector yields the following vector equation for OFDM transceivers

$$\mathbf{r} = \sqrt{\mathcal{E}} \mathbf{H} \mathbf{x} + \mathbf{w}, \quad (5.65)$$

where  $\mathbf{r} \in \mathcal{C}^{N_o}$  is the vector of correlator outputs at the receiver,  $\mathbf{x} \in \mathcal{C}^{N_o}$  is the vector of transmitted OFDM symbols in the packet,  $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_o})$ , and  $\mathbf{H}$  is the  $N_o \times N_o$  stochastic channel matrix that characterizes an OFDM link. As evident from (5.64), the matrix  $\mathbf{H}$  is a diagonal matrix since the Fourier basis functions  $\{\phi_m(t)\}$  serve as eigenfunctions<sup>11</sup> for purely frequency-selective channels [44]:

$$\mathbf{H} = \text{diag}\left(H(0), H\left(\frac{1}{T}\right), \dots, H\left(\frac{N_o-1}{T}\right)\right). \quad (5.66)$$

The advantage of OFDM over purely frequency-selective channels is evident from (5.64), (5.65), and (5.66): OFDM signaling decomposes the multipath channel into  $N_o = TW$  noninterfering parallel channels corresponding to different frequencies associated with the Fourier basis functions. The diagonal entries of  $\mathbf{H}$ , reflecting the impact of the channel on symbols modulated onto different frequencies, are random variables and exhibit a correlated structure. A simple model for capturing the  $L = \lceil W\tau_{\max} \rceil + 1$  level delay diversity (or DoF) afforded by the multipath channel is a *block fading*

<sup>11</sup>Strictly speaking, the Fourier basis functions are asymptotic eigenfunctions, in the limit of large  $T$  [44]. However, in practice, a cyclic prefix is used to essentially yield a diagonal  $\mathbf{H}$  for finite  $T$  [43].

model for  $\mathbf{H}$  based on the concept of *frequency coherence subspaces* induced by the coherence bandwidth [see (5.9)]: The  $N_o$  diagonal entries are partitioned into  $L$  independently fading blocks, where the  $N_o/L$  entries in each block, corresponding to a coherence bandwidth,<sup>12</sup> are assumed to be identical:

$$\mathbf{H} = \text{diag}(h_1, \dots, h_1, h_2, \dots, h_2, \dots, h_L, \dots, h_L). \quad (5.67)$$

Thus, the channel is characterized by the  $L$ -independent zero-mean Gaussian random variables,  $\{h_i\}$ , reflecting the delay diversity in the channel. Furthermore, due to the stationary nature of the channel in frequency, the  $h_i$  are identically distributed as well. Recall that the channel coefficients in the delay-RAKE structure in CDMA systems directly capture the delay diversity. The block fading for OFDM systems in (5.67) is a simple abstraction of delay diversity in the frequency domain and is useful for analyzing system performance.

**5.3.1.3 STF Transceivers: Short-Time Fourier Signaling** Now, let us consider doubly selective channels for which  $T\nu_{\max} > 1$ ,  $W\tau_{\max} > 1$ . In such channels, the orthogonality of the Fourier basis functions,  $\{\phi_m(t)\}$ , is destroyed at the receiver due to the significant temporal channel variation over the packet duration  $T$ . As a result,  $\mathbf{H}$  in (5.65) is no longer diagonal and the off-diagonal entries represent interference between the different OFDM symbols at the receiver. However, appropriately chosen Gabor or STF basis waveforms [18–21], a generalization of Fourier basis waveforms in OFDM, serve as approximate eigenfunctions for underspread ( $\tau_{\max}\nu_{\max} \ll 1$ ) doubly selective channels. In orthogonal STF signaling, the transmitted signal for one packet takes the form [21]

$$x(t) = \sqrt{\mathcal{E}} \sum_{\ell=0}^{N_t-1} \sum_{m=0}^{N_f-1} x_{\ell,m} \phi_{\ell,m}(t), \quad 0 \leq t \leq T, \quad (5.68)$$

$$\phi_{\ell,m}(t) = g(t - \ell T_o) e^{j2\pi m F_o t}, \\ \ell = 0, \dots, N_t - 1, \quad m = 0, \dots, N_f - 1, \quad N_t N_f = N_o = TW, \quad (5.69)$$

where  $x_{\ell,m}$  denotes the data symbol modulated onto the  $(\ell, m)$ th STF basis waveform,  $\phi_{\ell,m}(t)$ . Each basis waveform,  $\phi_{\ell,m}(t)$ , is generated from a prototype pulse,  $g(t)$ , via time and frequency shifts as in (5.69). For appropriate choice of  $g(t)$  with  $T_o F_o = 1$ , the resulting set of STF basis waveforms,  $\{\phi_{\ell,m}\}$ , form an orthonormal basis for the space of signals of duration  $T$  and bandwidth  $W$  [21].<sup>13</sup> As in OFDM, the data symbols have normalized average power,  $E[\|\mathbf{x}\|^2] = \sum_m E[|x_m|^2] = 1$ , and  $\mathcal{E}$  denotes packet energy:

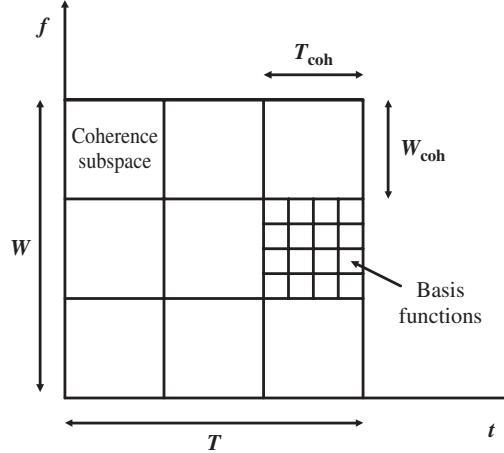
$$\int_0^T E[|x(t)|^2] dt = \mathcal{E} \sum_{\ell,m} E[|x_{\ell,m}|^2] = \mathcal{E}. \quad (5.70)$$

Short-time Fourier basis waveforms are illustrated in Figure 5.8. Each STF basis function has a duration and (essential<sup>14</sup>) bandwidth proportional to  $T_o$  and  $W_o$ ,

<sup>12</sup>  $W_{\text{coh}}/\Delta f = T/\tau_{\max} = TW/W\tau_{\max} \approx N_o/L$ .

<sup>13</sup> Biorthogonal basis functions can also be generated for  $T_o F_o > 1$ , with better interference properties but at the cost of spectral efficiency [20, 21].

<sup>14</sup> Since a strictly time-limited signal cannot have finite bandwidth, and vice versa due to the Fourier uncertainty principle.



**Figure 5.8** Illustration of STF basis functions tiling the time–frequency plane corresponding to a packet of duration  $T$  and bandwidth  $W$ . Each small square represents a STF basis function,  $\phi_{\ell,m}(t)$ . Each bigger square represents a time–frequency coherence subspace corresponding to  $T_{coh} \times W_{coh}$  with  $N_{coh}$  basis functions.

respectively. With appropriate choice of the prototype pulse,  $g(t)$ , and by matching the parameters  $(T_o, F_o)$  to the channel spread parameters [19–21]

$$\frac{T_o}{F_o} \propto \frac{\tau_{\max}}{v_{\max}}, \quad T_o F_o = 1 \quad (5.71)$$

the resulting  $N_o$  STF basis waveforms serve as a set of approximate eigenfunctions for underspread<sup>15</sup> doubly selective channels. We assume that the STF basis is generated with the matching in (5.71).

The received STF signal is given by

$$r(t) = \int h(t, \tau) x(t - \tau) d\tau + w(t) \approx \sqrt{\mathcal{E}} \sum_{\ell, m} x_{\ell, m} H(\ell T_o, m F_o) \phi_{\ell, m}(t) + w(t), \quad (5.72)$$

where  $h(t, \tau)$  denotes the time-varying impulse response and  $H(t, f)$  denotes the time-varying frequency response of the doubly selective multipath channel. The approximation in (5.72) illustrates the approximate eigenproperty of STF basis waveforms—it ignores the relatively small interference between different basis waveforms [21]. The approximation states that, analogous to OFDM, the different STF basis waveforms do not interfere with each other and the  $(\ell, m)$ th STF basis waveform simply gets multiplied by the corresponding value of the time-varying transfer function,  $H(\ell T_o, m F_o)$ , during transmission. At the receiver,  $r(t)$  is projected onto the STF basis waveforms

$$r_{\ell, m} = \langle r, \phi_{\ell, m} \rangle = \int r(t) \phi_{\ell, m}^*(t) dt = \sqrt{\mathcal{E}} x_{\ell, m} H(\ell T_o, m F_o) + w_{\ell, m}, \quad (5.73)$$

<sup>15</sup>The approximate eigenfunction property of the STF basis waveforms holds for  $\tau_{\max} v_{\max}$  as large as 0.01 [21].

and the resulting projections can be stacked into an  $N_o$ -dimensional vector to yield the following vector equation for STF transceivers:

$$\mathbf{r} = \sqrt{\mathcal{E}}\mathbf{H}\mathbf{x} + \mathbf{w}, \quad (5.74)$$

where  $\mathbf{r} \in \mathcal{C}^{N_o}$ ,  $\mathbf{x} \in \mathcal{C}^{N_o}$  is the vector of transmitted STF symbols,  $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_o})$ , and  $\mathbf{H}$  is the  $N_o \times N_o$  channel matrix that characterizes the STF link. As evident from (5.73), the matrix  $\mathbf{H}$  is an approximately diagonal matrix. We will assume that  $\mathbf{H}$  is exactly diagonal<sup>16</sup>:

$$\begin{aligned} \mathbf{H} = \text{diag}(H(0, 0), \dots, H(0, (N_f - 1)F_o), \dots, H((N_t - 1)T_o, 0), \dots, \\ H((N_t - 1)T_o, (N_f - 1)F_o)). \end{aligned} \quad (5.75)$$

Analogous to the block fading model for the OFDM channel matrix in (5.67), the diagonal entries of  $\mathbf{H}$  for STF signaling admit a corresponding intuitive block fading structure in terms of *time–frequency coherence subspaces* as illustrated in Figure 5.8. The  $N_o = TW$  diagonal entries of  $\mathbf{H}$  in (5.75) are partitioned into  $D = L(2M - 1) \approx (\lceil W\tau_{\max} \rceil + 1)(\lceil T\nu_{\max} \rceil + 1) \approx N_o\tau_{\max}\nu_{\max}$  independently fading subspaces corresponding to  $T_{\text{coh}} \times W_{\text{coh}}$ , with each subspace containing  $N_{\text{coh}} = T_{\text{coh}}W_{\text{coh}}/(T_oF_o) = N_o/D$  basis elements:

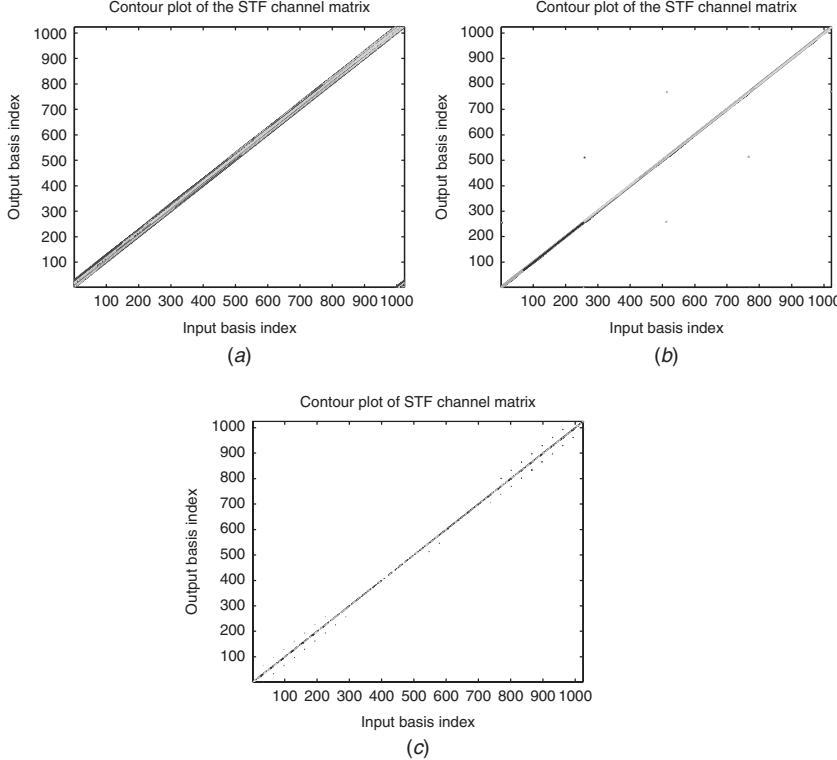
$$\mathbf{H} = \text{diag}(h_1, \dots, h_1, h_2, \dots, h_2, \dots, h_D, \dots, h_D). \quad (5.76)$$

The channel coefficients in each coherence subspace are assumed to be identical, whereas the coefficients in different subspaces are i.i.d. Thus, the STF channel matrix is characterized by  $D$  i.i.d. zero-mean Gaussian random variables,  $\{h_i\}$ , reflecting the  $D$ -level *delay–Doppler* diversity (the statistically independent DoF) afforded by the doubly selective channel. Recall that in CDMA systems, the delay–Doppler diversity was directly exploited by the delay–Doppler RAKE receiver structure.

Figure 5.9 illustrates the diagonal nature of the STF channel matrix corresponding to different cases of selectivity in time and frequency in  $H(t, f)$  depicted in Figure 5.1. Contour plots are shown for the full  $N_o \times N_o$  channel matrix with entries  $\{H(\ell'T_o, m'F_o; \ell T_o, m F_o)\}$ , where the indices  $\{(\ell, m)\}$  correspond to the input STF basis functions used at the transmitter [see (5.68)], and the indices  $\{(\ell', m')\}$  correspond to the output STF basis functions used at the receiver [see (5.73)]. The basis parameters  $(T_o, F_o)$  are matched to the channel spread parameters according to (5.71). All cases depicted in Figure 5.9 correspond to different choices of  $(T, W)$ , resulting in different corresponding choices of  $(N_t, N_f)$ , for a given packet length of  $N_o = TW = 1024$ . Note that the STF channel matrix is very nearly diagonal in all cases of channel selectivity.

**5.3.1.4 Capacity of Single-Antenna Channels** The capacity of single-antenna channels can be easily calculated in the OFDM/STF domain using (5.65) or (5.74). We consider coherent capacity where the channel,  $\mathbf{H}$ , is perfectly known at the receiver and

<sup>16</sup>This assumption is valid for small spread factors, and in general the residual interference can be mitigated at the receiver using a variety of interference cancelation schemes; the reader is referred to [21] for more details.



**Figure 5.9** Contour plots of the STF channel matrix,  $\{H(\ell' T_o, m' F_o; \ell T_o, m F_o)\}$ , including the off-diagonal entries, where  $\{(\ell, m)\}$  correspond to the STF basis indices at the transmitter and  $\{(\ell', m')\}$  correspond to the STF basis indices at the receiver. The plots correspond to different cases of selectivity in time and frequency in  $H(t, f)$  depicted in Figure 5.1 (see also Figure 5.3). The time–frequency support  $(T_o, F_o)$  of the STF basis functions is matched to channel delay and Doppler spreads as in (5.71):  $T_o = 10^{-3}$  s and  $F_o = 10^3$  Hz. Different cases correspond to different choices of  $(T, W)$  for a given packet length  $N_o = TW = 1024$ . (a) Purely frequency-selective channel corresponding to  $T = 3.2$  ms and  $W = 3.2 \times 10^5$  Hz so that  $N_t = 4$  and  $N_f = 256$ . (b) Purely time-selective channel corresponding to  $T = 0.32$  s and  $W = 3.2 \times 10^3$  Hz so that  $N_t = 256$  and  $N_f = 4$ . (c) A time- and frequency-selective channel corresponding to  $T = 32$  ms and  $W = 3.2 \times 10^4$  Hz so that  $N_t = N_f = \sqrt{N_o} = 32$ .

only channel statistics are assumed known at the transmitter. For Rayleigh fading, the capacity-achieving input is zero-mean Gaussian with i.i.d. entries,  $\mathbf{x} \sim \mathcal{CN}(\mathbf{0}, \rho \mathbf{I}/W)$ , where  $\rho = \mathcal{E}/T$  is the average total transmitted power [21]. The coherent ergodic capacity is given by

$$C = \frac{1}{N_o} E \left[ \log \left\{ \det \left( \mathbf{I} + \frac{\mathcal{E}}{N_o} \mathbf{H} \mathbf{H}^H \right) \right\} \right] = \frac{1}{N_o} E \left[ \log \left\{ \det \left( \mathbf{I} + \frac{\rho}{W} \mathbf{H} \mathbf{H}^H \right) \right\} \right] \text{ bits/s/Hz} \quad (5.77)$$

$$= \frac{1}{N_o} \sum_{i=0}^{N_o-1} E \left[ \log \left( 1 + \frac{\rho}{W} |h_i|^2 \right) \right] = E \left[ \log \left( 1 + \frac{\rho}{W} |h_i|^2 \right) \right] \text{ b/s/Hz}, \quad (5.78)$$

where the expectation is over channel statistics, the first equality in (5.78) follows from the diagonal nature of  $\mathbf{H}$ , and the second equality follows from the fact that all diagonal entries,  $\{h_i\}$ , of  $\mathbf{H}$  are identically distributed.

**5.3.1.5 CDMA versus OFDM/STF Signaling** At a fundamental level CDMA and OFDM/STF signaling are no different. However, from a practical perspective there are some advantages and disadvantages to both systems. First of all, in terms of interacting with the channel, in CDMA systems the channel is sampled in the delay–Doppler domain, whereas in STF systems, the channel is sampled in the time–frequency domain. Furthermore, in CDMA systems the transmission strategy remains the same, independent of the selectivity of the channel—only the receiver structure changes as a function of channel selectivity (delay RAKE versus delay–Doppler RAKE). On the other hand, the STF basis waveforms need to be appropriately adapted to the channel spread parameters as in (5.71) for creating noninterfering parallel channels.

Second, as evident from our discussion above, the channel diversity is exploitable in CDMA systems at the receiver only, whereas in OFDM/STF systems it is directly accessible at the transmitter as well. Furthermore, in CDMA systems the full diversity of the channel is exploitable with a single spreading waveform used for transmission, whereas in OFDM/STF systems, the transmitted information must be spread over all basis functions to fully exploit channel diversity. In particular, if a single bit is transmitted over all STF basis functions, it is easy to show that the resulting  $P_e$  in detecting the bit at the receiver can be calculated in a way similar to (5.55) for CDMA systems. The underlying test statistic in both cases involves a  $\chi^2$  random variable with  $2D$  degrees of freedom, where  $D$  is the level of diversity. In CDMA systems, this is directly evident since the channel coefficients directly sample the physical channel in the delay–Doppler domain, whereas in an STF system the level of diversity is revealed by the block fading model in terms of time–frequency coherence subspaces.

Finally, from a multiuser perspective, multiple users are assigned distinct spreading codes in a CDMA system. To allocate different rates to different users, multiple codes could be assigned to certain users. However, due to multipath channel effects, the codes of different users, as well as multiple codes for a particular user, interfere at the receiver. In an OFDM/STF system, different user transmissions can be kept orthogonal (and noninterfering) by assigning different users *disjoint* subsets of OFDM/STF basis functions. However, as noted above, these subsets of basis functions may not be able to maximally exploit channel diversity. In order to fully exploit channel diversity, OFDM/STF transmissions from each user must be spread over *all* basis functions. In this case, the resulting multiuser OFDM/STF transmissions will interfere at the receiver.

### 5.3.2 Nonselective MIMO Systems

Consider a narrowband, nonselective ( $T v_{\max} \ll 1$ ,  $W \tau_{\max} \ll 1$ ) MIMO channel corresponding to a transmitter with  $N_T$  antennas and a receiver with  $N_R$  antennas. The system equation in this case is given by

$$\mathbf{r} = \mathbf{Hx} + \mathbf{w}, \quad (5.79)$$

where  $\mathbf{r} \in \mathcal{C}^{N_R}$  is the received signal vector,  $\mathbf{x} \in \mathcal{C}^{N_T}$  is the transmitted signal vector, and  $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_R})$ . MIMO communication systems equipped with multiantenna

arrays augment the traditional signal space dimensions of time and frequency with the spatial dimension for enhanced communication over multipath channels. The spatial dimension enables exploitation of statistically independent *spatial* DoF afforded by the spatially distributed propagation paths in the multipath channel. These DoF are revealed by the virtual MIMO channel representation for ULAs. However, unlike single-antenna channels in which the channel DoF can be exploited for improving the *reliability* of communication through the concept *diversity*, MIMO systems also enhance the rate of communication over multipath channels by providing a *spatial multiplexing* gain [1, 2, 7, 8]. This is because the MIMO channel matrix  $\mathbf{H}$  is full-rank [ $\text{rank}(\mathbf{H}) = \min(N_R, N_T)$ ] in a rich scattering environment, thereby enabling the creation of  $\min(N_R, N_T)$  parallel spatial channels between the transmitter and the receiver. Thus, a MIMO channel can support  $\min(N_R, N_T)$  simultaneous data streams without any additional consumption of power or bandwidth.

Initial works on exploiting MIMO capacity gains were based on *spatial multiplexing*—transmitting *independent temporally coded* data streams on multiple antennas—inspired by the BLAST space–time communication architecture proposed in [8, 9]. However, it was soon realized that the reliability of MIMO communication could be dramatically enhanced by *space–time coding*—joint coding across spatial and temporal dimensions. The field of space–time coding was launched by the seminal work in [45, 46]. However, initial work on orthogonal space–time block codes [47, 48] and space–time trellis codes [46] emphasized exploitation of *spatial diversity* for improved reliability, at the cost of rate. Subsequent works on space–time coding were aimed at combining the rate advantage of spatial multiplexing and diversity advantage of orthogonal space–time codes, such as linear dispersion codes [49]. The ability to exploit the channel DoF for diversity (reliability) or multiplexing (rate) is governed by a fundamental *diversity versus multiplexing trade-off*, which was characterized in the high SNR regime in [50].

In Section 5.3.2.1, we first extend our discussion on statistical characterization of MIMO channels in Section 5.2.2 to non-ULA geometries and also emphasize fundamental differences in spatial channel characteristics compared to channel characterization in time and frequency. We then discuss MIMO link capacity in Section 5.3.2.2, followed by a discussion of MIMO transceiver structure in Section 5.3.2.3.

**5.3.2.1 Marginal and Joint Channel Statistics** The seminal work of Telatar, Foschini, and Gans [6–9] on the capacity of MIMO channels was based on an i.i.d. model for  $\mathbf{H}$  representing a rich scattering environment—the elements of  $\mathbf{H}$  are modeled as i.i.d.  $\mathcal{CN}(0, 1)$  random variables. However, i.i.d. MIMO channels are the exception rather than the norm in practice. Thus, there has been extensive work in the last decade on modeling and analysis of spatially correlated MIMO channels in which the entries of  $\mathbf{H}$  exhibit a correlated structure [10, 16, 51–54]. The relevant statistics for correlated MIMO channels are transmit, receive, and joint statistics:

$$\Sigma_T = E[\mathbf{H}^H \mathbf{H}] = \mathbf{U}_T \Lambda_T \mathbf{U}_T^H, \quad (5.80)$$

$$\Sigma_R = E[\mathbf{H} \mathbf{H}^H] = \mathbf{U}_R \Lambda_T \mathbf{U}_R^H, \quad (5.81)$$

$$\Sigma_{TR} = E[\mathbf{h} \mathbf{h}^H] = \mathbf{U}_{TR} \Lambda_{TR} \mathbf{U}_{TR}^H, \quad \mathbf{h} = \text{vec}(\mathbf{H}), \quad (5.82)$$

where  $\Sigma_T$  denotes the *transmit covariance matrix*,  $\Sigma_R$  denotes the *receive covariance matrix*, and  $\Sigma_{TR}$  denotes the *joint covariance matrix* of  $\mathbf{h} = \text{vec}(\mathbf{H})$ , where

$\text{vec}(\mathbf{H})$  corresponds to stacking the columns of  $\mathbf{H}$  into one  $N_T N_R$ -dimensional vector  $\mathbf{h}$  [55]. The second equalities in the above equations represent the eigendecompositions of the respective matrices. For example,  $\mathbf{U}_T$  represents the (unitary) matrix of eigenvectors of the transmit covariance matrix, and  $\Lambda_T$  represents the corresponding diagonal matrix of transmit eigenvalues. For Rayleigh fading channels, the channel statistics are characterized by the *joint covariance matrix*  $\Sigma_{TR}$ , and  $\Sigma_T$  and  $\Sigma_R$  can be viewed as *marginal* channel statistics as seen from the transmitter or the receiver side, respectively.

Initial works on correlated MIMO channels were based on the so-called *kronecker* model (see, e.g., [51]), which is entirely based on the marginal transmit and receive statistics:

$$\mathbf{H}_{\text{kron}} = \Sigma_R^{1/2} \mathbf{H}_{\text{iid}} \Sigma_T^{1/2} = \mathbf{U}_R \mathbf{H}_{\text{ind}} \mathbf{U}_T^H, \quad \mathbf{H}_{\text{ind}} = \Lambda_R^{1/2} \mathbf{H}_{\text{iid}} \Lambda_T^{1/2}, \quad (5.83)$$

where  $\mathbf{H}_{\text{iid}}$  represents an i.i.d. channel matrix and  $\mathbf{H}_{\text{ind}}$  has independent but not identically distributed entries. The joint statistics for the kronecker model are given by

$$\mathbf{h}_{\text{kron}} = \text{vec}(\mathbf{H}_{\text{kron}}) = \left[ \Sigma_T^{1/2} \otimes \Sigma_R^{1/2} \right] \mathbf{h}_{\text{iid}} = \left[ \mathbf{U}_T^* \otimes \mathbf{U}_R \right] \mathbf{h}_{\text{ind}}, \quad (5.84)$$

$$\Sigma_{TR,\text{kron}} = E \left[ \mathbf{h}_{\text{kron}} \mathbf{h}_{\text{kron}}^H \right] = \Sigma_T \otimes \Sigma_R = \left[ \mathbf{U}_T^* \otimes \mathbf{U}_R \right] [\Lambda_T \otimes \Lambda_R] \left[ \mathbf{U}_T^* \otimes \mathbf{U}_R \right]^H, \quad (5.85)$$

where  $\otimes$  denotes the kronecker product and we have used the identity  $\text{vec}(\mathbf{ADB}) = [\mathbf{B}^T \otimes \mathbf{A}] \text{vec}(\mathbf{D})$  [55]. The second equality in (5.85) shows that the joint statistics of kronecker model are simply the kronecker product of the marginal statistics, and the third equality is the eigendecomposition of  $\Sigma_{TR,\text{kron}}$ , which is related to the eigendecomposition of the marginal statistics through kronecker products:

$$\mathbf{U}_{TR,\text{kron}} = \mathbf{U}_T^* \otimes \mathbf{U}_R, \quad \Lambda_{TR,\text{kron}} = \Lambda_T \otimes \Lambda_R. \quad (5.86)$$

The separable structure of joint statistics in terms of marginal statistics in the kronecker model is not sufficiently rich to capture realistic channels in practice (see, e.g., [34, 35, 56–58]). The virtual MIMO channel representation for ULAs [16] was the first model for correlated MIMO channels that did not suffer for this limitation. Recall from (27) that the virtual representation decorrelates the MIMO channel matrix  $\mathbf{H}$ :

$$\mathbf{H}_v = \mathbf{A}_R^H \mathbf{H} \mathbf{A}_T \quad (5.87)$$

through the two-dimensional DFT effected by  $\mathbf{A}_R$  and  $\mathbf{A}_T$ . The entries of  $\mathbf{H}_v$  are independent but not identically distributed (as is the case for  $\mathbf{H}_{\text{ind}}$  for the kronecker model), and the correlation in  $\mathbf{H}$  is captured by the power profile,  $\{\Psi(i, k)\}$ , of the entries of  $\mathbf{H}_v$  defined in (5.32). It turns out that for ULAs, the matrices of transmit and receive eigenvectors are independent of the scattering environment and are given by DFT matrices  $\mathbf{A}_R$  and  $\mathbf{A}_T$ :

$$\mathbf{U}_{T,v} = \mathbf{A}_T, \quad \mathbf{U}_{R,v} = \mathbf{A}_R, \quad \mathbf{U}_{TR,v} = \mathbf{A}_T^* \otimes \mathbf{A}_R. \quad (5.88)$$

Furthermore, the corresponding diagonal matrices of eigenvalues for the marginal and joint covariance matrices are determined by the power profile,  $\{\Psi(i, k)\}$ , of  $\mathbf{H}_v$ :

$$\Lambda_{T,v} = E[\mathbf{H}_v^H \mathbf{H}_v], \quad \Lambda_{R,v} = E[\mathbf{H}_v \mathbf{H}_v^H], \quad \Lambda_{TR,v} = E[\mathbf{h}_v \mathbf{h}_v^H], \quad \mathbf{h}_v = \text{vec}(\mathbf{H}_v). \quad (5.89)$$

Comparing (5.86) and (5.89), we note that  $\Lambda_{TR,v}$  is not constrained to have a separable structure as  $\Lambda_{TR,\text{kron}}$ .

The kronecker model is applicable to arbitrary array geometries but is limited to separable statistical channel modeling. The virtual representation, on the other hand, captures joint channel statistics in full generality but is limited to ULAs of antennas. Motivated by these advantages and limitations of the two models, a generalization of the virtual representation—the *eigenbeam model* or the *canonical model*—was proposed in [35] and [36] to capture joint channel statistics for arbitrary array geometries. Specifically, the canonical model replaces the DFT matrices  $\mathbf{A}_T$  and  $\mathbf{A}_R$  in the virtual representation (5.27) with the matrices of transmit and receive eigenvectors in (5.80) and (5.81):

$$\mathbf{H} = \mathbf{U}_R \mathbf{H}_c \mathbf{U}_T^H. \quad (5.90)$$

With this transformation, it is shown in [35, 36] that if all the columns of  $\mathbf{H}$  share the same set of eigenvectors ( $\mathbf{U}_R$ ) and all the columns of  $\mathbf{H}^H$  share the same set of eigenvectors ( $\mathbf{U}_T$ ), the matrix  $\mathbf{H}_c$  has independent but not necessarily identically distributed entries (as in  $\mathbf{H}_v$  and  $\mathbf{H}_{\text{ind}}$ ). Thus, for the canonical model, the joint statistics are characterized by

$$\mathbf{U}_{TR,c} = \mathbf{U}_T^* \otimes \mathbf{U}_R, \quad \Lambda_{TR,c} = E[\mathbf{h}_c \mathbf{h}_c^H], \quad \mathbf{h}_c = \text{vec}(\mathbf{H}_c). \quad (5.91)$$

The canonical model is completely parallel to the virtual representation and corresponds to replacing  $\mathbf{A}_T$  and  $\mathbf{A}_R$  with  $\mathbf{U}_T$  and  $\mathbf{U}_R$ , respectively. The canonical model captures joint channel statistics, not constrained by a separable structure, and is applicable to arbitrary array geometries. However, unlike the virtual representation for ULAs for which the channel eigenvectors are independent of the scattering geometry, the eigenvectors in the canonical model, as in the kronecker model, depend on both the scattering environment and the array characteristics.

**5.3.2.2 Capacity of Nonselective MIMO Channels** We now discuss characterization of the ergodic capacity of MIMO channels—the maximum (long-term) information rate that can be reliably supported by MIMO channels. We assume that perfect CSI is available at the receiver (coherent reception) and only statistical CSI is available at the transmitter. Recall the system equation for a MIMO link:

$$\mathbf{r} = \sqrt{\rho} \mathbf{H} \mathbf{x} + \mathbf{w}, \quad (5.92)$$

where  $\rho$  denotes the total transmit SNR,  $E[\|\mathbf{x}\|^2] = 1$ , and  $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_R})$ . For the coherent case, when  $\mathbf{H}$  is assumed perfectly known at the receiver, the ergodic capacity is given by [1, 2, 7, 10]

$$C = \max_{\mathbf{Q}: \text{trace}(\mathbf{Q}) \leq 1} E [\log \det (\mathbf{I} + \rho \mathbf{H} \mathbf{Q} \mathbf{H}^H)] \text{ bits/s/Hz}, \quad (5.93)$$

which corresponds to using the capacity-achieving Gaussian input,  $\mathbf{x} \sim \mathcal{CN}(\mathbf{0}, \mathbf{Q})$ , where  $\mathbf{Q} = E[\mathbf{x}\mathbf{x}^H]$  denotes the input covariance matrix, and  $\text{trace}(\mathbf{Q}) = E[\|\mathbf{x}\|^2]$  denotes the sum of the diagonal entries of  $\mathbf{Q}$ . For i.i.d. MIMO channels, the optimal input is i.i.d. across different spatial dimensions,  $\mathbf{x}_{\text{opt}} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}/N_T)$ , and the capacity can be approximated at  $C \sim \min(N_T, N_R) \log(1 + \rho)$  representing the multiplexing gain over single-antenna channels. In particular, for i.i.d. channels, the capacity increases in direct proportion to the number of antennas,  $\min(N_T, N_R)$ , without any additional increase in transmit power or bandwidth.

For correlated channels, it has been shown that the capacity-achieving input covariance matrix takes the form [36, 53, 54]

$$\mathbf{Q}_{\text{opt}} = \mathbf{U}_T \Lambda_{\text{opt}} \mathbf{U}_T^H, \quad (5.94)$$

where  $\mathbf{U}_T$  is the matrix of transmit eigenvectors, defined in (5.80), and  $\Lambda_{\text{opt}}$  is determined via the *statistical waterfilling* in the optimization (5.93), and can be determined via iterative numerical algorithms [53, 54]. The above relation states that the capacity-achieving input vector has statistically independent components in the eigendomain corresponding to the diagonal covariance matrix  $\Lambda_{\text{opt}}$ . Unlike i.i.d. channels for which a uniform-power, full-rank input is optimal at all SNRs, for correlated channels the rank of the input is a function of the operating SNR since the transmit eigenvalues are not uniform. In particular, in the limit of high SNR, a uniform-power input is optimal, that is,  $\Lambda_{\text{opt}} = \mathbf{I}/N_T$  as in i.i.d. channels, whereas in the limit of low SNR, a rank-1 *beamforming* is optimal; that is,  $\text{rank}(\Lambda_{\text{opt}}) = 1$  and all the transmit power is concentrated in the eigendirection corresponding to the largest transmit eigenvalue [36, 53, 54].

**5.3.2.3 MIMO Transceivers: Eigenspace/Beamspace Signaling** In this section, we discuss the basic structure of MIMO transceivers using information about channel statistics and capacity optimal signaling. From a signal space perspective, the transmitted signal  $\mathbf{x} \in \mathcal{C}^{N_T}$  and the received signal  $\mathbf{r} \in \mathcal{C}^{N_R}$ . These signals can be represented using any orthonormal bases for the corresponding signal spaces. The transmitted signal can be represented as a linear combination of the transmit basis vectors, and the received signal is first projected onto the receive basis vectors to facilitate further processing at the receiver. While any set of transmit and receive basis vectors can be used in principle, appropriate choice of bases can greatly facilitate system design and analysis. Our discussion on modeling of MIMO channel statistics suggests a natural choice: the set of statistical transmit eigenvectors, the columns of  $\mathbf{U}_T$ , at the transmitter, and the set of statistical receiver eigenvectors, the columns of  $\mathbf{U}_R$ , at the receiver. That is, the transmit signal  $\mathbf{x}$  is represented as a linear combination of transmit eigenvectors and the received signal  $\mathbf{r}$  is projected onto the receive eigenvectors:

$$\mathbf{x} = \mathbf{U}_T \mathbf{x}_c, \quad \mathbf{r}_c = \mathbf{U}_R^H \mathbf{r} \Leftrightarrow \mathbf{r}_c = \sqrt{\rho} \mathbf{H}_c \mathbf{x}_c + \mathbf{w}_c, \quad (5.95)$$

where  $\mathbf{w}_c = \mathbf{U}_R^H \mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$ , and  $\mathbf{x}_c = \mathbf{U}_T^H \mathbf{x}$  and  $\mathbf{r}_c$  are the representations of the transmitted and received signals (in the antenna domain) with respect to the transmit and receive eigenbases. We note that for ULAs for which  $\mathbf{U}_T = \mathbf{A}_T$  and  $\mathbf{U}_R = \mathbf{A}_R$ , signaling and reception in the eigendomain corresponds to the beamspace domain and has an intuitively appealing physical interpretation: The elements of  $\mathbf{x}_c = \mathbf{x}_v$  correspond to signals transmitted in different beam directions, and the elements of  $\mathbf{r}_c = \mathbf{r}_v$

correspond to received signals from different beam directions. Furthermore, as discussion in Section 5.2.2, the independence of different entries of  $\mathbf{H}_v = \mathbf{H}_c$  has an intuitively appealing interpretation due to path partitioning: Distinct entries of  $\mathbf{H}_v$  are associated with disjoint sets of propagation paths.

The above relation shows the benefit of representing the transmitted and received signals in terms of the *statistical* channel eigenvectors (or fixed steering and response vectors in ULAs): The resulting channel matrix coupling  $\mathbf{x}_c$  and  $\mathbf{r}_c$  is the canonical channel matrix,  $\mathbf{H}_c$ , which has independent entries. Thus, transmission and reception along the statistical eigenvectors effectively decorrelates the channel matrix  $\mathbf{H}$ . Furthermore, as in (5.94), the capacity-achieving  $\mathbf{x}_c$  consists of independent Gaussian signals; that is,  $\mathbf{x}_c \sim \mathcal{CN}(\mathbf{0}, \Lambda)$ , where  $\Lambda$  is a *diagonal* covariance matrix, and the capacity characterization in (5.93) can be equivalently expressed as

$$C = \max_{\Lambda: \text{trace}(\Lambda) \leq 1} E [\log \det (\mathbf{I} + \rho \mathbf{H}_c \Lambda \mathbf{H}_c^H)] \text{ bits/s/Hz.} \quad (5.96)$$

As discussed above, uniform power allocation across all eigendirections is optimal at high SNRs, whereas a rank-1 input that excites the dominant transmit eigendirection is optimal at low SNRs. The uncorrelated nature of  $\mathbf{H}_c$  in (5.95) greatly facilitates design and analysis in a variety of aspects, including capacity analysis [36, 37, 52, 53], channel estimation [59], spatial multiplexing [60], and space–time coding [61, 62].

It is worth noting that even though the elements of  $\mathbf{H}_c$  are statistically independent, different transmitted symbols in  $\mathbf{x}_c$  interfere with each other at the receiver due to the nondiagonal entries in  $\mathbf{H}_c$ . This is analogous to multiuser interference in CDMA systems and a variety of interference suppression techniques, originally developed for CDMA systems [28], can be used at the receiver for reliable decoding of the transmitted symbols.

**5.3.2.4 Space–Time Coding** In this section, we review the basic idea of space–time coding to reap the capacity and diversity advantage of MIMO channels [45, 46]. Our focus is on the probability of error analysis of space–time codes that leads to the criteria used in the design of space–time codes [45, 46]. Since the seminal works in [45, 46], a variety of methodologies have been proposed for space–time code design, including space–time trellis codes [46], orthogonal space–time block codes [47, 48], and linear dispersion codes [49]. We consider signaling and reception in the eigenspace (or beamspace for ULAs), as in (5.95), that greatly facilitates analysis for spatially correlated MIMO channels.

The basic idea in space–time coding is to jointly encode information in space and time to exploit spatial diversity for enhanced reliability. Consider a discrete-time model for an  $N_R \times N_T$  MIMO system in the eigenspace

$$\mathbf{r}_c(k) = \sqrt{\rho} \mathbf{H}_c(k) \mathbf{x}_c(k) + \mathbf{w}_c(k), \quad (5.97)$$

where the index  $k$  represents the  $k$ th channel use. We consider a block fading model so that the channel is constant over  $K \geq N_T$  channel uses and changes independently between blocks of  $K$  channel uses. For simplicity of notation, we suppress the subscript  $c$  with the understanding that we are working in the eigenspace so that the elements of  $\mathbf{H}$  ( $= \mathbf{H}_c$ ) are independent but not necessarily identically distributed. Stacking

the received signal vectors over one block of  $K$  channel uses, the system equation becomes

$$\mathbf{R} = \sqrt{\rho} \mathbf{H} \mathbf{X} + \mathbf{W}, \quad (5.98)$$

where  $\mathbf{R} = [\mathbf{r}(1), \dots, \mathbf{r}(K)]$  is the  $N_R \times K$  matrix of received signal vectors,  $\mathbf{X} = [\mathbf{x}(1), \dots, \mathbf{x}(K)]$  is the  $N_T \times K$  matrix of transmitted signal vectors, and  $\mathbf{W} = [\mathbf{w}(1), \dots, \mathbf{w}(K)]$  is the  $N_R \times K$  noise matrix. The transmitted signal matrix (or space-time codeword)  $\mathbf{X}$  satisfies the power constraint  $E[\text{trace}(\mathbf{X}\mathbf{X}^H)] = K$ .

Consider a codebook of  $N$  codewords,  $\mathcal{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_N\}$ . In each block, one of the  $N$  codewords is transmitted. The rate of the space-time code  $\mathcal{X}$  is determined by the size of the codebook and is given by  $R = \log(N)/K$  bits/channel use, which must be less than  $\Delta t C$  where  $\Delta t$  is the sampling interval defining each channel use and  $C$  is the channel capacity at the given SNR ( $\rho$ ) as defined in (5.93) or (5.96). We assume coherent reception and that ML decoding is employed at the receiver to determine which codeword was transmitted. Let  $\mathbf{X}$  denote a transmitted codeword and  $\hat{\mathbf{X}}$  the decoded codeword at the receiver. Assuming that all codewords are equally likely, the conditional error probability, conditioned on a particular realization of  $\mathbf{H}$ , is given by

$$P_e(\mathbf{H}) = \frac{1}{N} \sum_{i=1}^N P_e(\mathbf{X}_i | \mathbf{H}), \quad (5.99)$$

where  $P_e(\mathbf{X}_i | \mathbf{H})$  denotes the conditional error probability when the  $i$ th codeword is transmitted and is given by

$$P_e(\mathbf{X}_i | \mathbf{H}) = P(\cup_{j \neq i} \{\hat{\mathbf{X}} = \mathbf{X}_j\} | \mathbf{X} = \mathbf{X}_i, \mathbf{H}) \leq \sum_{j=1, j \neq i}^N P(\hat{\mathbf{X}} = \mathbf{X}_j | \mathbf{X} = \mathbf{X}_i, \mathbf{H}), \quad (5.100)$$

where the inequality reflects the union bound [3]. Using the notation  $P(\mathbf{X}_i \rightarrow \mathbf{X}_j | \mathbf{H})$  for  $P(\hat{\mathbf{X}} = \mathbf{X}_j | \mathbf{X} = \mathbf{X}_i, \mathbf{H})$ , the conditional error probability in (5.99) can be bounded as

$$P_e(\mathbf{H}) \leq \frac{1}{N} \sum_{i=1}^N \sum_{j=1, j \neq i}^N P(\mathbf{X}_i \rightarrow \mathbf{X}_j | \mathbf{H}), \quad (5.101)$$

and the unconditional error probability can be bounded by averaging  $P_e(\mathbf{H})$  over the statistics of  $\mathbf{H}$ :

$$P_e = E[P_e(\mathbf{H})] \leq \frac{1}{N} \sum_{i=1}^N \sum_{j=1, j \neq i}^N P(\mathbf{X}_i \rightarrow \mathbf{X}_j), \quad (5.102)$$

where  $P(\mathbf{X}_i \rightarrow \mathbf{X}_j) = E[P(\mathbf{X}_i \rightarrow \mathbf{X}_j | \mathbf{H})]$  denotes the pairwise error probability (PEP) of decoding  $\mathbf{X} = \mathbf{X}_i$  as  $\hat{\mathbf{X}} = \mathbf{X}_j$ . Space-time code design boils down to the design of the codebook  $\mathcal{X}$  so that the PEP between any pair of codewords is as small as possible.

We now discuss the calculation of the PEP between an arbitrary pair of codewords to get insight into the design criteria for the codebook  $\mathcal{X}$ . Let  $\mathbf{E} = \mathbf{X} - \hat{\mathbf{X}}$  denote the error codeword matrix when  $\mathbf{X}$  is transmitted and decoded as  $\hat{\mathbf{X}} \neq \mathbf{X}$ . Using

the fact that  $\mathbf{H}$  has independent Gaussian entries, the PEP can be bounded as [45, 46, 60, 62]

$$P(\mathbf{X} \rightarrow \hat{\mathbf{X}}) = E[P(\mathbf{X} \rightarrow \hat{\mathbf{X}}|\mathbf{H})] \leq \left| \mathbf{I}_{N_T N_R} + \frac{\rho}{4} \tilde{\Sigma} (\mathbf{I}_{N_R} \otimes \mathbf{R}_e) \right|^{-1}, \quad (5.103)$$

where  $\mathbf{R}_e = \mathbf{E}\mathbf{E}^H$  is the  $N_T \times N_T$  codeword error correlation matrix and  $\tilde{\Sigma} = E[\text{vec}(\mathbf{H}^T)\text{vec}(\mathbf{H}^T)^H]$  is the  $N_R N_T \times N_R N_T$  (joint) channel covariance matrix of the vector  $\text{vec}(\mathbf{H}^H)$  obtained by stacking the rows of  $\mathbf{H}$ . The bound in (5.103) shows that the PEP depends on the interaction between the channel and the codewords that is captured by the  $N_R N_T \times N_R N_T$  matrix:

$$\Delta = \tilde{\Sigma} (\mathbf{I}_{N_R} \otimes \mathbf{R}_e) = \text{diag}(\tilde{\Sigma}(1)\mathbf{R}_e, \dots, \tilde{\Sigma}(N_R)\mathbf{R}_e), \quad (5.104)$$

where the second equality in terms of the block diagonal matrix follows from the fact that  $\tilde{\Sigma}$  is a diagonal matrix of the form  $\tilde{\Sigma} = \text{diag}(\tilde{\Sigma}(1), \dots, \tilde{\Sigma}(N_R))$  where  $\tilde{\Sigma}(i)$  represents the diagonal covariance matrix of the  $i$ th row of  $\mathbf{H}$  representing the MISO channel coupling the  $N_T$  transmit eigendimensions to the  $i$ th receive eigendimension. Let  $d_i = \text{rank}(\tilde{\Sigma}(i)\mathbf{R}_e) \leq \min(\text{rank}(\tilde{\Sigma}(i), \text{rank}(\mathbf{R}_e)) \leq N_T$ . Using (5.104), the PEP bound in (5.103) simplifies to

$$P(\mathbf{X} \rightarrow \hat{\mathbf{X}}) \leq \prod_{i=1}^{N_R} \left| \mathbf{I}_{N_T} + \frac{\rho}{4} \tilde{\Sigma}(i)\mathbf{R}_e \right|^{-1} = \prod_{i=1}^{N_R} \prod_{j=1}^{d_i} \left( 1 + \frac{\rho}{4} \lambda_j(\tilde{\Sigma}(i)\mathbf{R}_e) \right)^{-1}, \quad (5.105)$$

where  $\lambda_j(\tilde{\Sigma}(i)\mathbf{R}_e)$  denotes the  $j$ th nonzero eigenvalue of  $\tilde{\Sigma}(i)\mathbf{R}_e$ ; the first inequality follows from the fact that the determinant of the block diagonal matrix,  $\Delta$ , in (1.104) is the product of the determinants of the component matrices, and the second equality follows from the fact that the determinant of a matrix equals the product of its eigenvalues. At high SNR ( $\rho \gg 1$ ), the PEP bound in (5.105) can be further simplified to

$$P(\mathbf{X} \rightarrow \hat{\mathbf{X}}) \leq \left( \frac{4}{\rho} \right)^{\sum_{i=1}^{N_R} d_i} \frac{1}{\prod_{i=1}^{N_R} \prod_{j=1}^{d_i} \lambda_j(\tilde{\Sigma}(i)\mathbf{R}_e)}, \quad (5.106)$$

where the first term reflects the *diversity gain*—the rank of  $\Delta$ —and the second term reflects the *coding gain*—the product of the nonzero eigenvalues of  $\Delta$  in (5.104). Thus, the overall goal of space–time code design is to design the codebook  $\mathcal{X}$  so that the resulting  $\Delta$  for each pair of codewords has maximum rank (to maximize the diversity gain), and the size of its nonzero eigenvalues is as large as possible (to maximize the coding gain).

We note that the diversity gain is bounded as  $\sum_{i=1}^{N_R} d_i \leq N_R N_T$  and is limited by both the rank of  $\tilde{\Sigma}(i)$  and  $\mathbf{R}_e$ . For i.i.d. channels,  $\tilde{\Sigma}(i) = \mathbf{I}_{N_T}$ , and  $d_i = d = \text{rank}(\mathbf{R}_e)$ . Thus, for i.i.d. channels, the high SNR PEP bound in (5.106) reduces to

$$P(\mathbf{X} \rightarrow \hat{\mathbf{X}}) \leq \left( \frac{4}{\rho} \right)^{N_R d} \frac{1}{\prod_{i=1}^{N_R} \prod_{j=1}^d \lambda_j(\mathbf{R}_e)}, \quad (5.107)$$

which leads to the well-known “rank” and “determinant” criteria for space–time code design for i.i.d. channels [45, 46]: The codebook  $\mathcal{X}$  should be designed so that the error correlation matrix  $\mathbf{R}_e$  for each pair of codewords is full rank ( $d = N_T$  to ensure

maximum diversity gain  $N_T N_R$ ) and as large a determinant as possible (to maximize the coding gain). In general, the design of the codebook  $\mathcal{X}$  requires numerical methods to optimize the diversity and coding gains.

### 5.3.3 Time- and Frequency-Selective MIMO Systems

In this section, we integrate our development of single-antenna transceivers for doubly selective channels in Section 5.3.1 and MIMO transceivers for nonselective MIMO channels in Section 5.3.2 to develop transceiver structures in the most general case of time- and frequency-selective, spatially correlated MIMO channels. In all cases, as discussed in Section 5.3.2, spatial signaling and reception is with respect to the transmit and receive spatial eigenvectors. In the time–frequency domain, we will consider both OFDM/STF and CDMA signaling. This yields two main classes of transceivers: eigenspace–OFDM/STF transceivers and eigenspace–CDMA transceivers.

**5.3.3.1 Eigenspace–STF Transceivers** Consider a communication link with  $N_T$  antennas at the transmitter and  $N_R$  antennas at the receiver operating over a doubly selective ( $T\nu_{\max} > 1$ ,  $W\tau_{\max} > 1$ ), spatially correlated MIMO channel. Communication of information occurs through packets of duration  $T$  and bandwidth  $W$ . The spatiotemporal signal space is of dimension  $N_{s,T} = N_T T W$  at the transmitter and of dimension  $N_{s,R} = N_R T W$  at the receiver. Spatial modulation at the transmitter is done using the  $N_T$  transmit eigenvectors, the columns of the transmit covariance matrix  $\mathbf{U}_T$ :  $\{\mathbf{u}_{T,i} : i = 1, \dots, N_T\}$ . Spatial demodulation at the receiver is done using the  $N_R$  receive eigenvectors, the columns of the receive covariance matrix  $\mathbf{U}_R$ :  $\{\mathbf{u}_{R,k} : k = 1, \dots, N_R\}$ . Temporal modulation and demodulation is done using the  $N_o = TW$  STF basis waveforms:  $\{\phi_{\ell,m}(t) : \ell = 0, \dots, N_t - 1; m = 0, \dots, N_f - 1\}$ . We consider transmission and reception of a single packet since interpacket interference is negligible under assumption (5.41).

The transmitted signal vector for one packet can be expressed as

$$\mathbf{x}(t) = \sqrt{\mathcal{E}} \sum_{i=1}^{N_T} \sum_{\ell=0}^{N_t-1} \sum_{m=0}^{N_f-1} x_{c,i,\ell,m} \mathbf{u}_{T,i} \phi_{\ell,m}(t) = \sum_{\ell=0}^{N_t-1} \sum_{m=0}^{N_f-1} \mathbf{U}_T \mathbf{x}_{c,\ell,m} \phi_{\ell,m}(t), \quad 0 \leq t \leq T, \quad (5.108)$$

where  $\mathcal{E} = \int E[\mathbf{x}^H(t)\mathbf{x}(t)] dt = \mathcal{E} \sum_{i,\ell,m} E[|x_{c,i,\ell,m}|^2]$  denotes the total transmit packet energy,  $\{x_{c,i,\ell,m}\}$  denote the  $N_{s,T}$  data symbols modulated onto the spatiotemporal basis waveforms, and  $\mathbf{x}_{c,\ell,m}$  denotes the  $N_T$ -dimensional vector of *spatial* data symbols corresponding to the  $(\ell, m)$ th STF basis waveform. Using the spatial eigenvectors instead of the DFT vectors for ULAs, the sampled virtual representation in (5.35) becomes

$$\mathbf{H}(t, f) = \mathbf{U}_R \mathbf{H}_c(t, f) \mathbf{U}_T^H \quad (5.109)$$

$$\approx \mathbf{U}_R \left[ \sum_{\ell=0}^{L-1} \sum_{m=-(M-1)}^{M-1} \mathbf{H}_c(\ell, m) e^{j2\pi(m/T)t} e^{-j2\pi(\ell/W)f} \right] \mathbf{U}_T^H \quad (5.110)$$

$$= \sum_{i=1}^{N_R} \sum_{k=1}^{N_T} \sum_{\ell=0}^{L-1} \sum_{m=-(M-1)}^{M-1} H_c(i, k, \ell, m) \mathbf{u}_{R,i} \mathbf{u}_{T,k}^H e^{j2\pi(m/T)t} e^{-j2\pi(\ell/W)f}, \quad (5.111)$$

where  $\mathbf{H}_c(t, f) = \mathbf{U}_R^H \mathbf{H}(t, f) \mathbf{U}_T$  is the representation of  $\mathbf{H}(t, f)$  with respect to the spatial basis functions, the expansion within the brackets in (5.110) is a sampled delay–Doppler representation of  $\mathbf{H}_c(t, f)$ , and (5.111) is the most explicit version of the sampled representation in terms of the spatial basis functions.

Using (5.108) and (5.109) in (5.33), the received signal vector can be expressed as

$$\mathbf{r}(t) = \sqrt{\mathcal{E}} \sum_{\ell=0}^{N_t-1} \sum_{m=0}^{N_f-1} \int_{-W/2}^{W/2} \mathbf{U}_R \mathbf{H}_c(t, f) \mathbf{x}_{c,\ell,m} \Phi_{\ell,m}(f) e^{j2\pi f t} df + \mathbf{w}(t), \quad (5.112)$$

where  $\Phi_{\ell,m}(f)$  is the Fourier transform of  $\phi_{\ell,m}(t)$ , and  $\mathbf{w}(t)$  denotes an  $N_R \times 1$  vector of independent complex AWGN processes. The received signal vector is first projected onto receive spatial eigenvectors to yield

$$\mathbf{r}_c(t) = \mathbf{U}_R^H \mathbf{r}(t) = \sqrt{\mathcal{E}} \sum_{\ell=0}^{N_t-1} \sum_{m=0}^{N_f-1} \int_{-W/2}^{W/2} \mathbf{H}_c(t, f) \mathbf{x}_{c,\ell,m} \Phi_{\ell,m}(f) e^{j2\pi f t} df + \mathbf{w}_c(t), \quad (5.113)$$

which is then projected onto the STF basis waveforms to yield

$$\begin{aligned} \mathbf{r}_{c,\ell,m} &= \langle \mathbf{r}_c(t), \phi_{\ell,m}(t) \rangle \\ &= \sqrt{\mathcal{E}} \sum_{\ell'=0}^{N_t-1} \sum_{m'=0}^{N_f-1} \left( \int_0^T \int_{-W/2}^{W/2} \mathbf{H}_c(t, f) \Phi_{\ell',m'}(f) \phi_{\ell,m}^*(t) e^{j2\pi f t} dt df \right) \\ &\quad \times \mathbf{x}_{c,\ell',m'} + \mathbf{w}_{c,\ell,m}, \end{aligned} \quad (5.114)$$

$$\approx \sqrt{\mathcal{E}} \mathbf{H}_c(\ell T_o, m F_o) \mathbf{x}_{c,\ell,m} + \mathbf{w}_{c,\ell,m} \quad (5.115)$$

where the term in brackets in (5.114) evaluates to  $\approx \delta_{\ell-\ell'} \delta_{m-m'} \mathbf{H}_c(\ell T_o, m F_o)$ ,  $\mathbf{H}_c(\ell T_o, m F_o) = \mathbf{H}_c(t, f)|_{(t,f)=(\ell T_o, m F_o)}$ , due to the eigenproperty of STF basis functions, resulting in (5.115). Stacking the  $N_R \times 1$  vectors,  $\{\mathbf{r}_{c,\ell,m}\}$ , into a single  $N_{s,R} = N_R T W$  dimensional vector yields the following matrix system equation for eigenspace–STF transceivers:

$$\mathbf{r}_c = \sqrt{\mathcal{E}} \mathbf{H}_c \mathbf{x}_c + \mathbf{w}_c, \quad (5.116)$$

$$\begin{aligned} \mathbf{H}_c &= \text{diag} (\mathbf{H}_c(0, 0), \dots, \mathbf{H}_c(0, (N_f - 1) F_o), \mathbf{H}_c((N_t - 1) T_o, 0), \dots, \\ &\quad \mathbf{H}_c((N_t - 1) T_o, (N_f - 1) F_o)), \end{aligned} \quad (5.117)$$

where  $\mathbf{r}_c \in \mathcal{C}^{N_{s,R}}$ ,  $\mathbf{x}_c \in \mathcal{C}^{N_{s,T}}$ ,  $E[\|\mathbf{x}_c\|^2] = 1$ ,  $\mathbf{w}_c \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_{s,R}})$ , and the  $N_{s,R} \times N_{s,T}$  matrix  $\mathbf{H}_c$  in (5.117) is the representation of  $\mathbf{H}(t, f)$  with respect to the spatial eigenvectors and STF basis waveforms used for transmission and reception. As shown in (5.117),  $\mathbf{H}_c$  has a *block diagonal* structure due to the eigenproperty of STF basis waveforms, and the  $N_R \times N_T$  component matrices on the diagonal are the eigendomain spatial matrices corresponding to different STF dimensions.

**5.3.3.2 Capacity of Doubly Selective MIMO Channels** The system equation (5.116) for eigenspace–STF transceivers can be interpreted as a combination of (5.75)

and (5.95)—each diagonal entry in (5.75) is replaced by a spatial matrix in the eigendomain of the form (5.95). In particular, each of the component spatial matrices in (5.117) has independent but not identically distributed entries due to the spatial transformation into the eigendomain. Consequently, (5.117) also greatly facilitates calculation of the coherent ergodic capacity of doubly selective correlated MIMO channels. The capacity achieving input vector,  $\mathbf{x}_c \in \mathcal{C}^{N_{s,T}}$ , has zero-mean and independent Gaussian entries,  $\mathbf{x}_{c,\text{opt}} \sim \mathcal{CN}(\mathbf{0}, \Lambda_c)$ , and the diagonal covariance matrix  $\Lambda_c$  can be decomposed as

$$\Lambda_c = \text{diag} (\Lambda_{c,0,0}, \dots, \Lambda_{c,0,N_f-1}, \dots, \Lambda_{c,N_t-1,0}, \dots, \Lambda_{c,N_t-1,N_f-1}), \quad (5.118)$$

corresponding to the ordering of the STF dimensions in (5.117), where each component diagonal matrix is an  $N_T \times N_T$  matrix corresponding to a particular  $\mathbf{x}_{c,l,m}$  in (5.108). The coherent ergodic capacity of the link, assuming perfect knowledge of  $\mathbf{H}_c$  at the receiver, can be computed as

$$C = \frac{1}{TW} \max_{\Lambda_c: \text{trace}(\Lambda_c) \leq 1} E [\log \{\det (\mathbf{I}_{N_{s,R}} + \mathcal{E} \mathbf{H}_c \Lambda_c \mathbf{H}_c^H)\}] \text{ b/s/Hz} \quad (5.119)$$

$$\begin{aligned} &= \frac{1}{TW} \sum_{\ell=0}^{N_t-1} \sum_{m=0}^{N_f-1} \max_{\Lambda_{\ell,m}: \text{trace}(\Lambda_{\ell,m}) \leq 1/TW} \\ &\quad \times E [\log \{\det (\mathbf{I}_{N_R} + \mathcal{E} \mathbf{H}_c(\ell T_o, m F_o) \Lambda_{\ell,m} \mathbf{H}_c^H(\ell T_o, m F_o))\}] \quad (5.120) \end{aligned}$$

$$= \max_{\Lambda_c: \text{trace}(\Lambda_c) \leq 1/TW} E \log \det (\mathbf{I}_{N_R} + \mathcal{E} \mathbf{H}_c(0, 0) \Lambda_c \mathbf{H}_c^H(0, 0)) \text{ b/s/Hz}, \quad (5.121)$$

where the second equality follows from the fact that the optimal energy allocation is uniform across different STF dimensions, and the third equality follows from the fact that the spatial statistics of  $\mathbf{H}_c(t, f)$  are invariant with respect to  $t$  and  $f$  due to channel stationarity in time and frequency. Note that  $\Lambda_c$  is of dimension  $N_{s,T} \times N_{s,T}$  in (5.119) and of dimension  $N_T \times N_T$  in (5.121). The optimal spatial power allocation matrix  $\Lambda_c$  in (5.121) is determined via statistical water filling, assuming knowledge of channel statistics at the transmitter, as in Section 5.3.2.2.

A couple of comments about the system equation (5.116) and the corresponding capacity characterization are in order. First, note that energy allocated to different STF dimensions evaluates to

$$\frac{\mathcal{E}}{TW} = \frac{\rho T}{TW} = \frac{\rho}{W}, \quad (5.122)$$

where  $\rho = \mathcal{E}/T$  denotes the total average transmit power [or, equivalently, the total average transmit SNR since the noise in different dimensions is normalized to unit variance in  $\mathbf{w}_c \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_{s,R}})$ ]. Second, for purely frequency-selective channels, the STF basis waveforms reduce to OFDM basis waveforms corresponding to a MIMO–OFDM system [63] represented in the domain of statistical spatial eigenvectors. Furthermore, for ULAs eigendomain is replaced by beamspace for signaling and reception. Finally, while capacity-achieving  $\mathbf{x}_c$  has independent entries, from a reliability (probability of error) perspective, it is advantageous to code across the different spatial-temporal-spectral dimensions in  $\mathbf{x}_c$ , analogous to the

difference between spatial multiplexing and space–time coding in nonselective MIMO systems.

**5.3.3.3 Eigenspace–CDMA Transceivers** In eigenspace–CDMA transceivers the spatial basis functions used at the transmitter and the receiver remain the same, but spread-spectrum waveforms are used for temporal signaling. Let  $q(t)$  denote a unit energy spreading waveform of the form (5.42). The transmitted signal vector for one packet is given by

$$\mathbf{x}(t) = \sqrt{\mathcal{E}} \mathbf{U}_T \mathbf{x}_c q(t) = \sqrt{\mathcal{E}} \sum_{i=1}^{N_T} \mathbf{u}_{T,i} x_{c,i} q(t), \quad (5.123)$$

where  $\mathbf{x}_c \in \mathcal{C}^{N_T}$ ,  $E[\|\mathbf{x}_c\|^2] = 1$ , is the vector of symbols transmitted in different spatial transmit eigendirections, and  $\mathcal{E}$  denotes the total transmit energy ( $\rho = \mathcal{E}/T$  denotes the total transmit power or SNR):  $\int_0^T E[\mathbf{x}^H(t)\mathbf{x}(t)] dt = \mathcal{E} \sum_i E[|x_{c,i}|^2] \int |q(t)|^2 dt = \mathcal{E}$ . Using (5.110), the received signal vector is given by

$$\mathbf{r}(t) = \sqrt{\mathcal{E}} \sum_{\ell=0}^{L-1} \sum_{m=-(M-1)}^{M-1} \mathbf{U}_R \mathbf{H}_c(\ell, m) \mathbf{x}_c q(t - \ell/W) e^{j2\pi mt/T} + \mathbf{w}(t), \quad (5.124)$$

where  $\mathbf{H}_c(\ell, m)$  represents the component of  $\mathbf{H}_c(t, f)$  in (5.109) corresponding to the  $(\ell, m)$ th resolvable delay and Doppler shift. The received signal is first projected onto the receive spatial basis functions to yield

$$\mathbf{r}_c(t) = \mathbf{U}_R^H \mathbf{r}(t) = \sqrt{\mathcal{E}} \sum_{\ell=0}^{L-1} \sum_{m=-(M-1)}^{M-1} \mathbf{H}_c(\ell, m) \mathbf{x}_c q\left(t - \frac{\ell}{W}\right) e^{j2\pi mt/T} + \mathbf{w}_c(t), \quad (5.125)$$

which is then correlated with delayed and Doppler-shifted versions of  $q(t)$  to yield

$$\begin{aligned} \mathbf{r}_{c,\ell,m} &= \int_0^T \mathbf{r}_c(t) q^*\left(t - \frac{\ell}{W}\right) e^{-j2\pi mt/T} dt \\ &\approx \sqrt{\mathcal{E}} \mathbf{H}_c(\ell, m) \mathbf{x}_c + \mathbf{w}_{c,\ell,m}, \quad \ell = 0, \dots, L-1, \quad m = -(M-1), \dots, M-1. \end{aligned} \quad (5.126)$$

Stacking all the delay–Doppler correlator vector outputs in (5.126) into a single  $N_R D = N_R \times L(2M-1)$  dimensional vector yields the system equation for eigenspace–CDMA transceivers:

$$\mathbf{r}_c = \sqrt{\mathcal{E}} \mathbf{H}_c \mathbf{x}_c + \mathbf{w}_c \quad (5.127)$$

where  $\mathbf{r}_c \in \mathcal{C}^{N_R D}$ ,  $\mathbf{x}_c \in \mathcal{C}^{N_T}$ ,  $\mathbf{w}_c \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_R D})$ , and  $\mathbf{H}_c$  is a  $N_R D \times N_T$  channel matrix given by

$$\mathbf{H}_c = [\mathbf{H}_c^T(0, -(M-1)), \dots, \mathbf{H}_c^T(L-1, M-1)]^T \quad (5.128)$$

reflecting the  $D = L(2M-1)$ -level delay–Doppler diversity, in addition to the  $N_R$ -level spatial diversity, exploitable at the receiver.

As in Section 5.3.2.1, if a single BPSK symbol is transmitted,  $\mathbf{x}_c = b\mathbf{x}_{c,o}$ ,  $b \in \{-1, 1\}$  for some unit energy  $\mathbf{x}_{c,o}$ , then the ML estimate for the bit and the corresponding  $P_e$  is given by

$$\hat{b} = \text{sign}(\text{real}\{\mathbf{x}_{c,o}^H \mathbf{H}_c^H \mathbf{r}_c\}), \quad z = \mathbf{x}_{c,o}^H \mathbf{H}_c^H \mathbf{r}_c = b\sqrt{\mathcal{E}} \|\mathbf{H}_c \mathbf{x}_{c,o}\|^2 + \mathbf{x}_{c,o}^H \mathbf{H}_c^H \mathbf{w}_c, \quad (5.129)$$

$$P_e(\mathbf{H}_c) = Q\left(\sqrt{2\mathcal{E}\|\mathbf{H}_c \mathbf{x}_{c,o}\|^2}\right), \quad P_e = E[P_e(\mathbf{H}_c)]. \quad (5.130)$$

Note that the decision-statistic  $z$  in (5.129) involves the  $N_R D$ -dimensional vector  $\mathbf{H}_c \mathbf{x}_{c,o}$ , a linear combination of the columns of  $\mathbf{H}_c$ , which has independent entries. Thus, the decision-statistic involves a  $\chi^2$  random variable with  $2N_R D$  DoF representing the spatial-delay–Doppler diversity exploitable at the receiver.

In this section, we have primarily focused on point-to-point communication using wideband MIMO transceivers. In a multiuser context, in general, there is interference between the signals of different users, especially in CDMA systems, and a variety of multiuser detection techniques can be applied [28]. Multiuser detection techniques based on the sampled channel representation for a multiple access channel where the base station or access point is equipped with an antenna array are discussed in [64]. Interference suppression is also discussed in the following section.

## 5.4 ACTIVE WIRELESS SENSING WITH WIDEBAND MIMO TRANSCEIVERS

In this section, we discuss an application of wideband MIMO transceivers in the area of wireless sensor networks that have emerged as a promising technology for gathering information about the physical environment using a network of wireless sensor nodes (see, e.g., [22, 23]). The sensor nodes can sense the environment in a variety of modalities, including acoustic, seismic, chemical, and biological, and can communicate using wireless front ends. Since the sensor nodes are typically battery powered, energy consumption by the sensors is a key design challenge. Wireless sensor networks are being developed for a variety of applications, including surveillance, environmental monitoring, industrial monitoring, and health care. Most existing proposals for the communication architecture in wireless sensor networks are based on the original vision of *in-network processing*: The measurements collected by the sensors are processed within the network for different application tasks (e.g., detection or classification of an event) via exchange of local information between sensors in the network. However, in-network processing of sensor data can entail excess delay and energy consumption due to the attendant tasks of information routing and coordination between nodes (see, e.g., [24]).

Specifically, we present a framework for information retrieval in wireless sensor networks—*active wireless sensing* (AWS)—in which a wireless information retriever (WIR), equipped with a multiantenna array, actively interrogates an ensemble of (single-antenna) wireless sensors with wideband space–time waveforms for rapid retrieval of sensor information [25–27]. A key motivation for AWS is to reduce the excess delay and energy consumption associated with the distributed communication architecture in in-network processing. Technological advances in agile radio frequency (RF) front ends and reconfigurable antenna arrays provide another motivation for AWS.

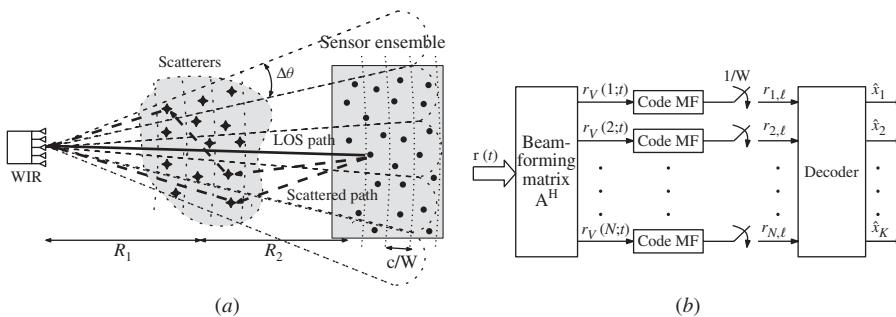
While originally developed for rapid information retrieval in sensor networks, the framework is also applicable to general point-to-multipoint network communication settings.

Active wireless sensing is built on two primary assumptions: (1) The sensor nodes are dumb in that they have limited computational ability but have relatively sophisticated RF front ends, and (2) the WIR is computationally powerful, is equipped with an antenna array, and actively interrogates the sensor ensemble with wideband space-time waveforms. A key idea behind AWS is that the distinct sensors induce *distinct space-time signatures* at the WIR that depend on the sensor locations relative to the multipath scattering environment connecting the sensors to the WIR. The sensor space-time signatures are exploited by the WIR for distinguishing different sensor signals. In the context of the development in Section 5.3, the MIMO transceiver at the WIR is an example of a beamspace-CDMA transceiver that performs spatial processing in the beamspace and uses spread-spectrum waveforms for communication with the sensors.

The next section introduces the basic communication architecture in AWS and develops the space-time system model for communication between the WIR and the sensors through a multipath channel. In Section 5.4.2, we discuss the concept of angle-delay matched filtering for computation of sufficient statistics for information retrieval at the WIR and also discuss the impact of multipath scattering on the DoF in the the *angle-delay signatures* induced by the sensors at the WIR. Section 5.4.3 discusses the uplink communication in AWS from the sensor ensemble to the WIR. In particular, we discuss a linear minimum mean-squared error (MMSE) scheme for suppressing the interference between sensor transmissions, and discuss two important performance metrics: sensing capacity that characterizes the maximum rate of reliable information retrieval from the sensor ensemble, and the probability of error in recovering sensor data. In Section 5.4.4, we focus on downlink communication in AWS from the WIR to the sensor ensemble. In particular, we highlight the potential of *time-reversal techniques* [29, 30] for downlink communication in AWS.

#### 5.4.1 Basic Space-Time Communication Architecture

Consider an ensemble of  $K$  sensors randomly distributed over a region of interest, as illustrated in Figure 5.10a. The WIR equipped with an  $N$ -element ULA, initiates



**Figure 5.10** Active wireless sensing. (a) Basic architecture for communication between the WIR and the sensor ensemble. (b) Computation of sufficient statistics at the WIR through angle-delay matched filtering.

information retrieval by sending a beacon signal for timing and frequency synchronization. The sensors send information to the WIR in packets of duration  $T$  and bandwidth  $W$ , synchronized with respect to timing of the beacon from the WIR. The sensors use a *common* spread-spectrum waveform,  $q(t)$ , of the form (5.42) and known at the WIR, to communicate to the WIR. Unlike traditional multiple-access schemes, such as CDMA, in which distinct nodes are assigned distinct codes or signatures  $\{q_k(t)\}$ , in AWS all sensor nodes use an identical temporal spread-spectrum waveform  $q(t)$ . Their transmissions are distinguished at the WIR via distinct angle–delay signatures induced by the multipath scattering environment.

The transmitted signals from the  $K$  sensors for one packet are given by

$$x_k(t) = \sqrt{\mathcal{E}} q(t)x_k, \quad 0 \leq t \leq T, \quad k = 1, \dots, K, \quad (5.131)$$

where  $x_k(t)$  denotes the transmitted waveform,  $x_k$  the data symbol, and  $\mathcal{E}$  the transmission energy from the  $k$ th sensor. The sensor transmissions pass through a frequency-selective ( $W\tau_{\max} > 1$ ) and time nonselective ( $T\nu_{\max} \ll 1$ ) multipath channel consisting of  $N_p$  scattering paths, as illustrated in Figure 5.10a. The received signal vector at the  $N$ -element ULA of the WIR,  $\mathbf{r}(t) = [r_1(t), r_2(t), \dots, r_N(t)]^T$ , is a superposition of all the sensor transmissions:

$$\begin{aligned} \mathbf{r}(t) &= \sum_{k=1}^K \int_0^{\tau_{\max}} \mathbf{h}_k(t') x_k(t-t') dt' + \mathbf{w}(t) \\ &= \sqrt{\mathcal{E}} \sum_{k=1}^K x_k \sum_{n=1}^{N_p} \beta_{k,n} q(t - \tau_{k,n}) \mathbf{a}(\theta_{k,n}) + \mathbf{w}(t) \end{aligned} \quad (5.132)$$

$$\approx \sqrt{\mathcal{E}} \sum_{k=1}^K x_k \sum_{i=1}^N \sum_{\ell=0}^{L-1} H_{v,k}(i, \ell) q\left(t - \frac{\ell}{W}\right) \mathbf{a}\left(\frac{i}{N}\right) + \mathbf{w}(t), \quad (5.133)$$

$$\mathbf{h}_k(t) = \sum_{n=1}^{N_p} \beta_{k,n} \delta(t - \tau_{k,n}) \mathbf{a}(\theta_{k,n}) \approx \sum_{i=1}^N \sum_{\ell=0}^{L-1} H_{v,k}(i, \ell) \delta\left(t - \frac{\ell}{W}\right) \mathbf{a}\left(\frac{i}{N}\right), \quad (5.134)$$

where  $\mathbf{h}_k(t)$  represents the  $N \times 1$  vector channel impulse response from the  $k$ th sensor to the WIR,  $\mathbf{a}(\theta)$  denotes the  $N \times 1$  array response vector [see (5.25)] of the ULA at the WIR, and  $\mathbf{w}(t)$  denotes a vector AWGN process with independent components. The second equality in (5.132) and the first equality in (5.134) represents a physical model for  $\mathbf{h}_k(t)$ , where  $\tau_{k,n} \in [0, \tau_{\max}]$  denotes the relative delay,  $\theta_{k,n} \in [-\frac{1}{2}, \frac{1}{2}]$  the angle of arrival (AoA), and  $\beta_{k,n}$  the complex path gain of the  $n$ th scattering path associated with the  $k$ th sensor. The third approximation in (5.133) and the second approximation in (5.134) represent the sampled representation of the multipath channel associated with  $k$ th sensor at delay resolution  $\Delta\tau = 1/W$  and angle resolution  $\Delta\theta = 1/N$  ( $L = \lceil W\tau_{\max} \rceil + 1$ ). Thus, each  $\mathbf{h}_k(t)$  is characterized by the  $LN$  sampled angle–delay channel coefficients  $\{H_{v,k}(i, \ell)\}$ .

Without loss of generality, assume that for each sensor, the  $n = 1$  path represents a strong line-of-sight (LOS) component with energy  $E[|\beta_{k,1}|^2] = 1$ , and the remaining  $N_p - 1$  paths are non-line-of-sight (NLOS) with energy  $E[|\beta_{k,n}|^2] = \sigma_s^2 < 1$ ,  $n = 2, \dots, N_p$ , as illustrated in Figure 5.10a. The energy of the scattered paths,  $\sigma_s^2$ ,

is generally smaller than that of the LOS components since the NLOS paths incur additional losses due to multiple reflections and larger propagation distances. We also assume that  $\{\theta_{k,n}, \tau_{k,n}\}$  are fixed during the time scales of interest. The only source of channel randomness are the random and independent phases of the gains  $\{\beta_{k,n}\}$ . The total channel power,  $\sigma_c^2$ , is defined as

$$\sigma_c^2 = \sum_n |\beta_{k,n}|^2 = 1 + (N_p - 1)\sigma_s^2 \quad (5.135)$$

and grows linearly with  $N_p$  since more paths couple more energy transmitted from the sensors to the WIR.

#### 5.4.2 Angle-Delay Matched Filtering

As evident from the sampled representation in (5.133), the received packet signal from each sensor belongs to an  $N_s = LN$ -dimensional spatiotemporal signal subspace spanned by the (approximately<sup>17</sup>) orthonormal space-time basis functions  $\{\mathbf{u}_{i,\ell}(t) = (1/\sqrt{N})\mathbf{a}(i/K)q(t - \ell/W)\}$ . Thus, to retrieve the sensor data for each packet, the WIR performs *angle-delay matched filtering* with respect to  $\{\mathbf{u}_{i,\ell}\}$  as illustrated in Figure 5.10b:  $\mathbf{r}(t)$  is first projected onto  $N$ -fixed beam directions, and each beamformer output is then temporally correlated with uniformly delayed versions of  $q(t)$  to yield the sufficient statistics for information retrieval:

$$\begin{aligned} r_{i,\ell} &= \frac{1}{\sqrt{N}} \int_0^T \mathbf{a}^H \left( \frac{i}{N} \right) \mathbf{r}(t) q^* \left( t - \frac{\ell}{W} \right) dt \\ &\approx \sqrt{\mathcal{E}N} \sum_{k=1}^K x_k H_{v,k}(i, \ell) + w_{i,\ell}, \quad i = 1, \dots, N, \quad \ell = 0, \dots, L-1. \end{aligned} \quad (5.136)$$

Stacking the angle-delay matched filtered (MF) outputs in an  $N_s$ -dimensional vector, the uplink system equation in AWS can be written as [26]

$$\mathbf{r} = \sqrt{N\mathcal{E}} \mathbf{Hx} + \mathbf{w} = \sqrt{N\mathcal{E}} \sum_{k=1}^K x_k \mathbf{h}_k + \mathbf{w}, \quad (5.137)$$

where  $\mathbf{x}$  denotes the  $K$ -dimensional vector of transmitted sensor symbols,  $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_s})$ ,  $\mathbf{h}_k$  is an  $N_s$ -dimensional vector consisting of  $\{H_{v,k}(i, k)\}$ , and  $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_K]$  is the  $N_s \times K$  uplink channel matrix coupling the sensors to the WIR. The factor  $\sqrt{N}$  in (5.137) represents the array gain at the WIR. We note that AWS is equivalent to a semidistributed MIMO system with  $\mathbf{H}$  representing the channel matrix. Successful recovery of sensor data at the WIR requires that  $N_s \geq K$  and the  $\mathbf{h}_k$ 's are linearly independent so that  $\text{rank}(\mathbf{H}) = K$ .

The vector  $\mathbf{h}_k$  represents the *angle-delay signature* induced by the  $k$ th sensor at the WIR, which can be estimated at the WIR using pilot sensor transmissions. Since the components of  $\mathbf{h}_k$  correspond to the sampled channel coefficients, it can be explicitly

<sup>17</sup>The approximation is due to the approximate orthogonality of delayed versions of  $q(t)$ —see (5.49).

related to the physical scattering environment via the concept of path partitioning [25] (see also Section 5.2):

$$\begin{aligned} h_k(i, \ell) &= H_{v,k}(i, \ell) \approx \sum_{n \in S_{i,\ell}(k)} \beta_{k,n}, \\ S_{i,\ell}(k) &= \left\{ n : \left| \theta_{k,n} - \frac{i}{N} \right| < \frac{1}{2N}, \quad \left| \tau_{k,n} - \frac{\ell}{W} \right| < \frac{1}{2W} \right\}, \end{aligned} \quad (5.138)$$

where  $S_{i,\ell}(k)$  denotes the set of all paths, associated with the  $k$ th sensor, whose angles and delays lie within the angle–delay resolution bin of size  $\Delta\theta \times \Delta\tau = (1/N) \times (1/W)$  corresponding to the  $(i, \ell)$ th angle–delay MF output in (5.136). The dominant nonvanishing entries in  $\mathbf{h}_k$  represent its statistically independent DoF since they correspond to disjoint sets of propagation paths (with independent path gains). Using (5.138), we can now characterize the differences in the structure of  $\{\mathbf{h}_k\}$  for LOS and multipath scattering environments, as discussed next.

In a LOS channel ( $N_p = 1$ ), each  $\mathbf{h}_k$  has one DoF—one dominant nonvanishing component—corresponding in a single angle–delay bin determined by the relative physical location of the  $k$ th sensor encoded in  $(\theta_{k,1}, \tau_{k,1})$ . It follows from (5.138) that the sensor signatures  $\{\mathbf{h}_k\}$  are linearly independent ( $\mathbf{H}$  is full rank) if and only if sensors are spaced sufficiently far apart so that their LOS paths lie in distinct angle–delay bins. When sensors are closely spaced, multiple sensors are mapped to the same angle–delay bin. In this case, separation of sensor transmissions at the WIR requires that the sensors associated with the same angle–delay bin be assigned distinct spreading codes, as in traditional CDMA systems.

In a multipath environment with sufficiently many ( $N_p \gg 1$ ) and *spatially distributed* NLOS paths, it follows from (5.138) that each  $\mathbf{h}_k$  exhibits a large number of dominant nonvanishing components or DoF that are statistically independent. It can also be shown that if  $N_p \geq N_s \geq K$ , the entries of  $\mathbf{H}$  are (approximately) statistically independent and hence the different  $\mathbf{h}_k$  are linearly independent<sup>18</sup> almost surely [27]. As a result  $\mathbf{H}$  is full-rank almost surely. Thus, AWS allows for exploitation of multipath scattering in two important aspects. First, the average energy in each signature  $E [\|\mathbf{h}_k\|^2] = \sigma_c^2$  grows linearly with  $N_p$  [see (5.135)], thereby increasing energy efficiency. Second, the presence of multipath scattering increases the DoF in sensor angle–delay signatures, effectively increasing the sensor resolution at the WIR. Thus, AWS over multipath can accommodate finer scale sensing and larger information rates compared to LOS environments.

#### 5.4.3 Uplink Communication: Rate and Reliability of Information Retrieval

In this section, we discuss the performance of uplink communication in AWS from the sensors to the WIR. Under the assumption that  $T \gg \tau_{\max}$ , packet-by-packet decoding suffices. Consider the simultaneous transmission of  $K$  packets from the  $K$  sensors in (5.131) in a single packet duration  $T$ . In the angle–delay MF outputs in (5.137), the angle–delay signatures of different sensors,  $\{\mathbf{h}_k\}$ , interfere with each other since they

<sup>18</sup>Even if two components of distinct  $\mathbf{h}_k$ 's correspond to the same set of propagation paths, as in (5.138), the path phases associated with the common set of paths will be different for the two sensors if the paths are sufficiently distributed in space relative to the sensor separation.

are not orthogonal, in general. This is analogous to interference between multiuser transmissions in a CDMA system and a variety of multiuser detection techniques can be applied [28]. We consider a simple linear MMSE interference suppression scheme [28] that exploits the differences in  $\{\mathbf{h}_k\}$  to suppress the interference between them. The linear MMSE receiver is described by a  $K \times N_s$  matrix,  $\mathbf{G}_{\text{mmse}}$ , that operates on the MF output vector  $\mathbf{r}$  and is given by

$$\mathbf{G}_{\text{mmse}} = \arg \min_{\mathbf{G}} E[\|\mathbf{Gr} - \mathbf{x}\|^2] = \mathbf{H}^H \mathbf{R}^{-1}, \quad (5.139)$$

where  $\mathbf{R} = E[\mathbf{rr}^H] = N\mathcal{E}\mathbf{HH}^H + \mathbf{I}$  is the correlation matrix of the MF outputs. In (5.139),  $\mathbf{R}^{-1}$  suppresses the interference corrupting the MF outputs, and the matrix  $\mathbf{H}^H$  performs angle–delay signature matched filtering on the resulting filtered MF outputs. The  $k$ th filtered decision statistic in  $\tilde{\mathbf{r}} = \mathbf{G}_{\text{mmse}}\mathbf{r}$  can be expressed as

$$\tilde{r}_k = \sqrt{N\mathcal{E}} \mathbf{h}_k^H \mathbf{R}^{-1} \mathbf{h}_k x_k + \sqrt{N\mathcal{E}} \sum_{k' \neq k} \mathbf{h}_k^H \mathbf{R}^{-1} \mathbf{h}_{k'} x_{k'} + \mathbf{h}_k^H \mathbf{R}^{-1} \mathbf{w}, \quad k = 1, \dots, K, \quad (5.140)$$

where  $\mathbf{h}_k^H \mathbf{R}^{-1} \mathbf{h}_k$  represents the filtered *desired* signal from the  $k$ th sensor, and  $\mathbf{h}_k^H \mathbf{R}^{-1} \mathbf{h}_{k'}$  the suppressed interference from the  $k'$ th sensor. Decisions on the transmitted symbols in  $\mathbf{x}$  can then be made from  $\tilde{\mathbf{r}}$  depending on the nature of the symbol constellation.

If the sensors transmit using BPSK symbols,  $\{x_k \in \{-1, +1\}\}$ , the symbol decisions at the WIR take the form

$$\hat{\mathbf{x}}_{\text{mmse}} = \text{sign} \{ \text{real} (\tilde{\mathbf{r}}) \} = \text{sign} \{ \text{real} (\mathbf{G}_{\text{mmse}} \mathbf{r}) \}. \quad (5.141)$$

Using a Gaussian approximation for the interference [28], the instantaneous (conditioned on  $\mathbf{H}$ ) probability of error in detecting the  $k$ th bit stream from the  $k$ th sensor can be expressed in terms of the signal to interference and noise ratio (SINR) as

$$P_{e,k}(\mathbf{H}) = Q \left( \sqrt{2\text{SINR}_k(\mathbf{H})} \right), \quad \text{SINR}_k(\mathbf{H}) = \frac{N\mathcal{E} |\mathbf{h}_k^H \mathbf{R}^{-1} \mathbf{h}_k|^2}{\|\mathbf{h}_k^H \mathbf{R}^{-1}\|^2 + N\mathcal{E} \sum_{k' \neq k} |\mathbf{h}_{k'}^H \mathbf{R}^{-1} \mathbf{h}_{k'}|^2}. \quad (5.142)$$

The long-term averaged  $P_e$  is given by  $P_e = E[Q(\sqrt{2\text{SINR}_k(\mathbf{H})})]$  where the expectation is over the statistics of  $\mathbf{H}$ .

**5.4.3.1 Sensing Capacity** The uplink communication scheme discussed above corresponds to *uncoded* transmissions from each sensor.  $K$  bits of sensor information are retrieved by the WIR in each transmission packet of duration  $T$ , over a bandwidth  $W$ , with energy  $\mathcal{E}$  expended by each sensor. What is the *sensing capacity*—the maximum rate of reliable information retrieval—of AWS for a given packet energy  $\mathcal{E}$ ?

The capacity of the AWS system is governed by the  $N_s \times K$  stochastic matrix  $\mathbf{H}$  in (5.137) under the constraint of independent transmissions from different distributed sensor nodes (components of  $\mathbf{x}$ ). Using results on the capacity of MIMO channels [7, 53], the sensing capacity of AWS, for a given  $\mathbf{H}$ , is given by

$$C(\mathbf{H}) = \frac{1}{TW + L} \log \det (\mathbf{I} + \mathcal{E}N\mathbf{HH}^H) = \frac{1}{TW + L} \sum_{k=1}^{K_{\text{eff}}} \log_2 (1 + \mathcal{E}N\lambda_k) \text{ bits/s/Hz}, \quad (5.143)$$

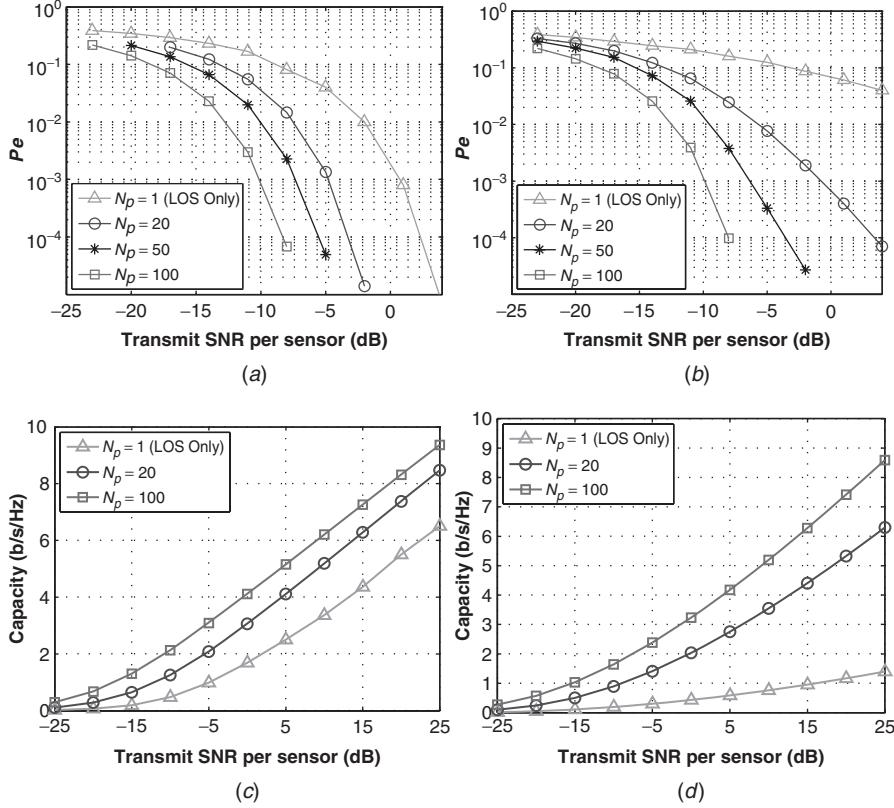
which reflects the mutual information between the  $K$  sensor inputs and  $N_s$  MF outputs at the WIR, under the assumption of equal power and independent Gaussian signaling from the sensors,<sup>19</sup> and  $TW + L = (T + \tau_{\max})W$  is the effective time–bandwidth product for each channel use. The second equality in (5.143) is in terms of the eigenvalues,  $\{\lambda_k\}$ , of the matrix  $\mathbf{H}\mathbf{H}^H$  where  $1 \leq K_{\text{eff}} \leq K$  is the rank of  $\mathbf{H}\mathbf{H}^H$  and represents the number of parallel channels created between the sensor ensemble and the WIR. The long-term ergodic capacity is given by  $C = E[C(\mathbf{H})]$ , where the expectation is over the statistics of  $\mathbf{H}$  in (5.143). The sensing capacity can be achieved via independently *coded* transmissions from the sensors.

**5.4.3.2 Numerical Results** We now illustrate the performance of information retrieval in the AWS uplink with numerical results. The results are generated using a spreading code of dimension  $TW = 127$  for  $q(t)$ ,  $N = 9$  antennas at the WIR,  $K = 108$  sensors, a normalized delay spread of  $L = 12$  delay bins, and energy in the NLOS paths,  $\sigma_s^2 = \frac{1}{8}$ . The scatterers are located half way between the sensor ensemble and the WIR, as illustrated in Figure 5.10a. The distances shown are  $R_1 + R_2 \approx 2R = 100c/W$ , where  $c$  is the speed of wave propagation. We consider two cases of sensor spacing: (1) widely spaced—the LOS paths from the 108 sensors arrive in distinct angle–delay bins at the WIR, or (2) closely spaced—the 108 LOS paths are mapped to only 12 distinct angle–delay bins. The normalized angular spread for the NLOS paths is  $\theta_{k,n} \in [-1/\sqrt{2}, 1/\sqrt{2}]$  corresponding to spatial path dispersion of  $45^\circ$  on either side of broadside direction.

The probability of error performance and sensing capacity in AWS are illustrated in Figure 5.11, which plot the long-term  $P_e$  and  $E[C(\mathbf{H})]$ , averaged across all sensors, as a function of the per-sensor transmit SNR,  $\mathcal{E}$ . In Figure 5.11a, c, the sensors are widely spaced, and hence  $\mathbf{H}$  is always full-rank as discussed in Section 5.4.2, and  $K_{\text{eff}} = K$ . Thus, in this case, the shift in the  $P_e$  curves toward lower SNR and the upward shift in the capacity curves with increasing  $N_p$  primarily reflect the increased energy coupled through the multipath channel [increased size of the eigenvalues in (5.143)]. In Figure 5.11b, d, the sensors are closely spaced, and hence  $\mathbf{H}$  is rank-deficient in the LOS scenario (see Section 5.4.2). In this case, the improved  $P_e$  performance and the increase in the sensing capacity with  $N_p$  are due to two effects. First, the spatially distributed NLOS paths increase the DoF in the channel  $\mathbf{H}$  ( $K_{\text{eff}}$  increases), which is reflected in the increasing *slope* of both the  $P_e$  and capacity curves as  $N_p$  increases, corresponding to an increase in the diversity gain and the multiplexing gain, respectively. Second, the shifts in the  $P_e$  and capacity curves for larger  $N_p$  are also due to higher energy capture as in the previous case.

The  $P_e$  curves correspond to retrieval of  $K = 108$  bits per channel use. For widely spaced sensors in Figure 5.11a,  $P_e = 10^{-3}$  is achieved at a per-sensor SNR of  $-10$  dB for  $N_p = 100$  paths, and at  $-5$  dB for  $N_p = 20$  paths. The corresponding AWS capacity at these SNRs for closely spaced sensors in Figure 5.11c is about  $2.2 \times (TW + L) = 2.2 \times 139 \approx 305$  bits per channel use for either  $N_p = 100$  or  $N_p = 20$  paths. Thus, uncoded transmission achieves about one third of capacity at a  $P_e = 10^{-3}$ . For closely spaced sensors, on the other hand, from Figure 5.11b the same  $P_e$  is achieved at SNRs of  $-10$  and  $-1$  dB for  $N_p = 100$  and  $N_p = 20$ , respectively. The corresponding

<sup>19</sup>We note that since the elements of  $\mathbf{H}$  are independent, the ergodic capacity achieving input  $\mathbf{x}_{\text{opt}} \sim \mathcal{CN}(\mathbf{0}, \Lambda)$ —that is, independent signaling from the sensors is optimal. If  $E[\mathbf{H}^H \mathbf{H}] = c\mathbf{I}$ , then  $\Lambda = c'\mathbf{I}$  as in (5.143).



**Figure 5.11** (a) and (b)  $P_e$  versus per-sensor transmit SNR ( $\mathcal{E}$ ). (c) and (d) Sensing capacity versus per-sensor transmit SNR ( $\mathcal{E}$ ). The sensors are widely spaced in (a) and (c), and closely spaced for (b) and (d).

AWS capacity at these SNRs from Figure 5.11d is about  $1.8 \times 139 = 250$  bits per channel for  $N_p = 100$  and about  $2 \times 139 = 278$  bits per channel use for  $N_p = 20$ . Thus, the spectral efficiency of uncoded transmission is a little higher for closely spaced sensors. Furthermore, it is worth noting that the performance is almost invariant to the spacing of sensors for richer multipath ( $N_p = 100$ ), since a  $P_e = 10^{-3}$  is achieved at the same SNR of  $-10$  dB in both cases. On the other hand, for fewer paths ( $N_p = 20$ ), a higher SNR of  $-1$  dB (compared to  $-5$  dB) is required for closely spaced sensors compared to widely spaced sensors to achieve a  $P_e = 10^{-3}$ .

#### 5.4.4 Downlink: Addressing Sensors with Space–Time Reversal Signaling

An attractive feature of AWS is the ability of the WIR to individually address distinct sensors for “programming” them for different tasks, such as signal estimation or event detection. Furthermore, it is often desirable to shift the computational burden from the (sensor) nodes to the access point (WIR) for energy efficiency. With these goals in mind, in this section we discuss a novel downlink communication scheme for sending dedicated information to distinct sensors using *space–time reversed* (STR) versions of their signatures  $\{\mathbf{h}_k\}$  [65]. The sensor nodes can then retrieve the information

intended for them by simply match-filtering to the (common) spread-spectrum waveform,  $q(t)$ , used in the uplink communication—no channel estimation is required at the sensors. Time reversal techniques, previously successfully used in acoustic communication and imaging applications [29], have been investigated more recently for wireless communications (see, e.g., [30]).

The STR transmitted signal vector from the WIR is given by

$$\mathbf{s}_{\text{tr}}(t) = \sum_{k=1}^K \mathbf{s}_{\text{tr},k}(t), \quad (5.144)$$

$$\begin{aligned} \mathbf{s}_{\text{tr},k}(t) &= \sqrt{\frac{\mathcal{E}}{N\|\mathbf{h}_k\|^2}} s_k \sum_{i=1}^N \sum_{\ell=0}^{L-1} h_k^*(i, \ell) \mathbf{a}^* \left( \frac{i}{N} \right) q^* \left( \tilde{T} - t - \frac{\ell}{W} \right), \\ 0 \leq t \leq \tilde{T}, \quad \tilde{T} &= T + \tau_{\max}, \end{aligned} \quad (5.145)$$

where  $\mathbf{s}_{\text{tr},k}(t)$  denotes the STR signal and  $s_k$  the data symbol intended for  $k$ th sensor with  $E[|s_k|^2] = 1$ , and the normalization ensures that each  $\mathbf{s}_{\text{tr},k}(t)$  has energy  $\mathcal{E}$ . Using the reciprocity of the multipath channel and the sampled channel representation, the received signal at the  $k'$ th sensor is given by

$$z_{k'}(t) = \sum_{i=1}^N \sum_{\ell=0}^{L-1} h_{k'}(i, \ell) \mathbf{a}^T \left( \frac{i}{N} \right) \mathbf{s}_{\text{tr}} \left( t - \frac{\ell}{W} \right) + w_{k'}(t), \quad (5.146)$$

where  $w_{k'}(t)$  denotes the AWGN process at the  $k'$ th sensor. The  $k'$ th sensor filters  $x_{k'}(t)$  with  $q(t)$  and the filter output is given by

$$x_{k'}(t) = \int z_{k'}(t') q(t - t') dt'. \quad (5.147)$$

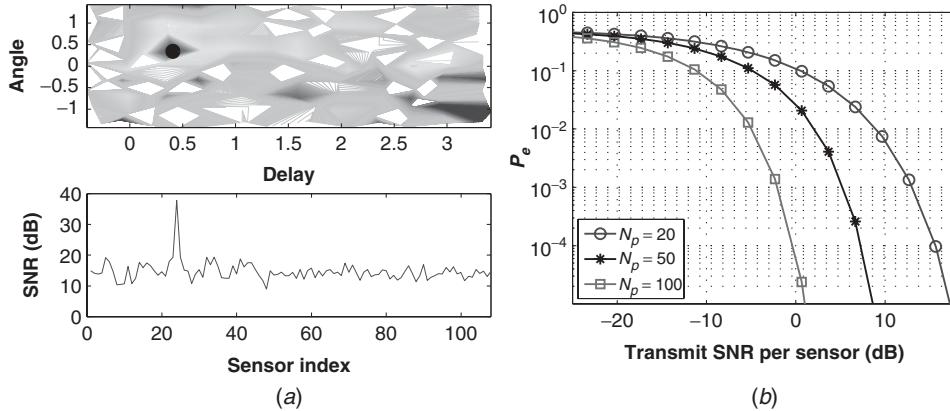
Sampling  $x_{k'}(t)$  at the “optimal” sampling time  $t = \tilde{T}$  yields the decision statistic,  $x_{k'}$ , at the  $k'$ th sensor for detecting the symbol,  $s_{k'}$ , intended for it [65]:

$$x_{k'} = x_{k'}(\tilde{T}) = \sqrt{\mathcal{E}N\|\mathbf{h}_{k'}\|^2} s_{k'} + \sum_{k=1, k \neq k'}^K \sqrt{\frac{\mathcal{E}N}{\|\mathbf{h}_k\|^2}} \mathbf{h}_{k'}^T \mathbf{h}_k^* s_k + w_{k'}, \quad k' = 1, \dots, K, \quad (5.148)$$

where the first term represents the desired signal component, the second term denotes the interference with WIR transmissions intended for other sensors, and  $w_{k'}$  denotes the noise in the decision statistic. The factor  $\sqrt{N}$  reflects the beamforming gain in downlink communication due to the antenna array at the WIR. Stacking the decision statistics into one  $K$ -dimensional vector yields the downlink system equation with STR signaling:

$$\mathbf{x} = \sqrt{\mathcal{E}N} \mathbf{H}^T \tilde{\mathbf{H}}^* \mathbf{s} + \mathbf{w}, \quad \tilde{\mathbf{H}} = \left[ \frac{\mathbf{h}_1}{\|\mathbf{h}_1\|}, \dots, \frac{\mathbf{h}_K}{\|\mathbf{h}_K\|} \right], \quad (5.149)$$

where  $\sqrt{\mathcal{E}N} \tilde{\mathbf{H}}^* \mathbf{s}$  represents the STR signal transmitted from the WIR with total transmit energy  $K\mathcal{E}$ ,  $\mathbf{s}$  denotes the vector of data symbols intended for different sensors with  $E[\|\mathbf{s}\|^2] = K$ ,  $\sqrt{N} \mathbf{H}^T$  represents the downlink channel coupling the WIR to the



**Figure 5.12** (a) The SNR of the decision statistics at different sensors with STR signaling from the WIR intended for a particular sensor. In the (top) image plot, a black circle indicates the target sensor (index 23). The plots correspond to  $N_p = 100$  scattering paths. (b)  $P_e$  vs. per-sensor transmit SNR ( $\mathcal{E}$ ) for different number of scattering paths  $N_p$ . The simulation setup in all plots is identical to the one used for the numerical results in Figures 5.11(b) and (d) corresponding to closely spaced sensors.

vector of sensor decision statistics,  $\mathbf{x}$ , with  $\sqrt{N}$  reflecting the beamforming gain, and  $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_K)$  represents the independent noise corrupting the sensor decision statistics. The diagonal entries of  $\sqrt{\mathcal{E}}\mathbf{N}\mathbf{H}^T\tilde{\mathbf{H}}^*$  represent the desired signal terms and the off-diagonal entries represent the interference in  $\mathbf{x}$  as in (5.148).

Figure 5.12a shows the SNR of the decision statistics,  $\mathbf{x}$ , for all sensors when the WIR transmits information only for a particular sensor (index 23) using its signature. Note that the desired sensor exhibits a significantly higher SNR compared to other sensors due to the STR operation—in general, the richer the multipath, the sharper the ability of STR signaling to focus the signal at the desired sensor (location) while minimizing the interference to other sensors (locations) [29]. Low-complexity linear *precoding* techniques, analogous to linear MMSE interference suppression in the uplink, can also be used at the WIR to further suppress any residual interference between the different sensor transmissions, thereby improving the reliability of down-link communication [65]. Figure 5.12b shows the average  $P_e$  of STR signaling with one such interference suppression scheme as a function of per-sensor transmit SNR ( $\mathcal{E}$ ) when  $K = 108$  sensors are simultaneously addressed. Note the effect of both increased *energy capture* ( $P_e$  curves shift toward lower SNR) and *higher angle-delay diversity* (steeper  $P_e$  curves) with larger number of scattering paths  $N_p$ .

## 5.5 CONCLUDING REMARKS

Multipath propagation and interference are the two most salient features of wireless communications. While the basic theory of point-to-point communication over multipath wireless channels is fairly well developed, theory for optimally dealing with interference in a network setting is still not fully developed [66, 67]. In this chapter, we have discussed basic transceiver structures for optimal communication over multipath

wireless channels. Our development was anchored on a sampled virtual representation of wireless channels that captures the interaction between the physical propagation environment and signal space of the transceivers in time, frequency, and space to reveal the statistically independent DoF available in the channel for communication. While the primary focus of transceiver design and analysis was on point-to-point links, we also discussed an application of the theory in active wireless sensing in which linear techniques for suppression of multiuser interference were considered.

In terms of future work, the development in this chapter lays the foundation for exploiting the advanced capabilities of *agile* wireless transceivers for optimal communication and sensing over multipath wireless channels. Technological advances are affording wireless transceivers with agility in terms of frequency, bandwidth, waveform, and/or array configuration. Design and analysis of waveform-agile wireless transceivers requires multipath channel characterization as a function of the resolution in time, frequency, and space afforded by the wireless transceiver configuration. This interaction is captured by the sampled virtual channel representation. Such sophisticated wireless transceivers also hold great promise in the emerging area of *cognitive radio* [68–70] in which the wireless nodes in a network sense and adapt to the wireless environment to better utilize the limited radio spectrum. The concept of cognitive radio also subsumes two related emerging areas of *dynamic spectrum access* [71, 72] and *waveform diversity* [73]. From the viewpoint of the impact of multipath propagation, a key emerging insight is that wireless channels exhibit a *sparse* multipath structure as the dimension of the signal space increases [74]. Recent research results have shown that agile wireless transceiver configuration can be adapted to the sparsity of multipath for dramatic increases in link capacity and reliability (see, e.g., [38, 75, 76]). From the viewpoint of learning the channel state information, a key element of cognitive radio, the emerging theory of compressed sensing could be fruitfully leveraged for efficient estimation of sparse multipath channels [77].

In the context of other chapters in this handbook, this chapter is most closely related to Chapter 1 on wavefields and Chapter 3 on MIMO radio propagation. Other chapters that have related material include Part 1 on fundamental issues in sensor and array processing, and Chapter 7 on space–time waveform diversity.

## REFERENCES

1. D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, 2005.
2. A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2006.
3. J. G. Proakis, *Digital Communications*, 4th ed., New York: McGraw-Hill, 2002.
4. P. A. Bello, “Characterization of randomly time-variant linear channels,” *IEEE Trans. Commun. Syst.*, vol. CS-11, pp. 360–393, Nov. 1963.
5. R. S. Kennedy, *Fading Dispersive Communication Channels*, New York: Wiley, 1969.
6. E. Telatar, “Capacity of multi-antenna Gaussian channels,” *AT& T-Bell Labs Intern. Tech. Memo.*, June 1995.
7. I. E. Telatar, “Capacity of multi-antenna Gaussian channels,” *Eur. Trans. Telecommun.*, vol. 10, pp. 585–595, Nov. 1999.
8. G. J. Foschini, “Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas,” *Bell Labs Tech. J.*, vol. 1, no. 2, pp. 41–59, 1996.

9. G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Commun.*, vol. 6, pp. 311–335, 1998.
10. A. Goldsmith, S. Jafar, N. Jindal, and S. Vishwanath, "Capacity limits of MIMO channels," *IEEE J. Selec. Areas Commun.* (special issue on MIMO systems), vol. 21, no. 5, pp. 684–702, June 2003.
11. A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity—Part I: System description," *IEEE Trans. Commun.*, pp. 1927–1938, Nov. 2003.
12. A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity—Part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, pp. 1939–1948, Nov. 2003.
13. P. Liu, Z. Tao, Z. Lin, E. Erkip, and S. Panwar, "Cooperative wireless communications: A cross-layer approach," *IEEE Wireless Commun. Mag.*, pp. 84–92, Aug. 2006.
14. A. Chakrabarti, E. Erkip, A. Sabharwal, and B. Aazhang, "Code designs for cooperative communication," *IEEE Signal Process. Mag.*, pp. 16–26, Sept. 2007.
15. A. M. Sayeed and B. Aazhang, "Joint multipath-Doppler diversity in mobile wireless communications," *IEEE Trans. Commun.*, vol. 47, no. 1, pp. 123–132, Jan. 1999.
16. A. M. Sayeed, "Deconstructing multi-antenna fading channels," *IEEE Trans. Signal Process.*, vol. 50, no. 10, pp. 2563–2579, Oct. 2002.
17. A. M. Sayeed, "A virtual representation for time- and frequency-selective correlated MIMO channels," *Proc. 2003 Int. Conf. Acoust. Speech Signal Process.*, vol. 4, pp. 648–651, 2003.
18. D. Gabor, "Theory of communication," *J. IEEE*, vol. 93, pp. 429–457, 1946.
19. W. Kozek, "On the transfer function calculus for underspread LTV channels," *IEEE Trans. Signal Process.*, pp. 219–223, Jan. 1997.
20. W. Kozek and A. F. Molisch, "Nonorthogonal pulseshapes for multicarrier communications in doubly dispersive channels," *IEEE J. Select. Areas Commun.*, vol. 16, no. 8, pp. 1579–1589, Oct. 1998.
21. K. Liu, T. Kadous, and A. M. Sayeed, "Orthogonal time-frequency signaling for doubly dispersive channels," *IEEE Trans. Inform. Theory*, pp. 2583–2603, Nov. 2004.
22. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.
23. *IEEE Signal Processing Magazine* (S. Kumar and F. Zhao and D. Shepherd (Eds.), Special issue on collaborative signal and information processing in microsensor networks, Mar. 2002).
24. D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'01)*, Vol. 4, May 2001, pp. 2033–2036.
25. T. Sivanadyan and A. Sayeed, "Active wireless sensing for rapid information retrieval in sensor networks," in *Proc. 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*, Apr. 2006, pp. 85–92.
26. A. Sayeed and T. Sivanadyan, "Source channel communication protocols and tradeoffs in active wireless sensing," in *Proc. 44th Annual Allerton Conference on Communication, Control and Computing*, Sept. 2006.
27. T. Sivanadyan and A. Sayeed, "Active wireless sensing in multipath environments," in *Proc. IEEE Statistical Signal Processing Workshop (SSP'07)*, Madison, WI, Aug. 2007, pp. 378–382.
28. S. Verdu, *Multiuser Detection*, Cambridge University Press, 1998.
29. M. Fink, "Time reversed acoustics," *Phys. Today*, vol. 50, no. 3, pp. 34–40, Mar. 1997.

30. C. Oestges, J. Hansen, S. M. Emami, A. D. Kim, G. Papanicolaou, and A. J. Paulraj, "Time reversal techniques for broadband wireless communication systems," *Eur. Microwave Week*, pp. 49–66, Oct. 2004.
31. K. S. Miller, *Complex Stochastic Processes*, Reading, MA: Addison-Wesley, 1974.
32. D. Slepian, "On bandwidth," *Proc. IEEE*, vol. 64, no. 3, pp. 292–300, Mar. 1976.
33. A. F. Molisch, "Ultrawideband propagation channels—Theory, measurement and modeling," *IEEE Trans. Veh. Tech.*, Sept. 2005.
34. P. Almers, E. Bonek, A. Burr, N. Czink, M. Debbah, V. Degli-Esposti, H. Hofstetter, P. Kyösti, D. Laurenson, G. Matz, A. F. Molisch, C. Oestges, and H. Özcelik, "Survey of channel and radio propagation models for wireless MIMO systems," *EURASIP J. Wireless Commun. Networking*, vol. 2007, p. 19, 2007.
35. W. Weichselberger, M. Herdin, H. Özcelik, and E. Bonek, "A stochastic MIMO channel model with joint correlation of both link ends," *IEEE Trans. Wireless Commun.*, pp. 90–100, Jan. 2006.
36. J. H. Kotecha and A. M. Sayeed, "Canonical statistical models for correlated MIMO fading channels and capacity analysis," Tech. Rep. ECE-03-05, University of Wisconsin-Madison, Oct. 2003.
37. A. Sayeed, V. Raghavan, and J. Kotecha, "Capacity of space-time wireless channels: A physical perspective," paper presented at the 2004 Information Theory Workshop, San Antonio, TX, Sept. 2004.
38. A. M. Sayeed and V. Raghavan, "Maximizing MIMO capacity in sparse multipath with reconfigurable antenna arrays," *IEEE J. Select. Topics Signal Process.*, June 2007.
39. A. M. Sayeed and V. Veeravalli, "The essential degrees of freedom in space-time fading channels," in *Proc. 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'02)*, Lisbon, Portugal, Sept. 2002, pp. 1512–1516.
40. A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communications*, Reading, MA: Addison-Wesley, 1995.
41. A. M. Sayeed, A. Sendonaris, and B. Aazhang, "Multiuser detection in fast fading multipath environments," *IEEE J. Select. Areas Commun.*, vol. 16, no. 9, pp. 1691–1701, Dec. 1998.
42. T. A. Kadous and A. M. Sayeed, "Decentralized multiuser detection for time-varying multipath channels," *IEEE Trans. Commun.*, vol. 48, pp. 1840–1852, Nov. 2000.
43. A. R. S. Bahai, B. R. Saltzberg, and M. Ergen, *Multi Carrier Digital Communications: Theory and Applications of OFDM*, Springer, 2004.
44. Robert M. Gray, "On the asymptotic eigenvalue distribution of toeplitz matrices," *IEEE Trans. Inform. Theory*, vol. 18, no. 6, pp. 725–730, Nov. 1972.
45. M. P. Fitz, J. Grimm, and J. V. Krogmeier, "Results on code design for transmitter diversity in fading," in *1997 IEEE Intl. Symp. Inform. Th.*, 1997, p. 234.
46. V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criteria and code construction," *IEEE Trans. Inform. Theory*, pp. 744–765, Mar. 1998.
47. V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. Inform. Theory*, pp. 1456–1467, July 1999.
48. E. G. Larsson and P. Stoica, *Space-Time Block Coding for Wireless Communications*, Cambridge: Cambridge University Press.
49. B. Hassibi and B. Hochwald, "High-rate codes that are linear in space and time," *IEEE Trans. Inform. Theory*, pp. 1804–1824, July 2002.

50. L. Zheng and D. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple antenna channels," *IEEE Trans. Inform. Theory*, vol. 49, no. 5, 2003.
51. C-N. Chuah, D. N. C. Tse, J. M. Kahn, and R. A. Valenzuela, "Capacity scaling in MIMO wireless systems under correlated fading," *IEEE Trans. Inform. Theory*, vol. 48, no. 3, pp. 637–650, Mar. 2002.
52. K. Liu, V. Raghavan, and A. M. Sayeed, "Capacity scaling and spectral efficiency in wide-band correlated MIMO channels," *IEEE Trans. Inform. Theory* (special issue on MIMO systems), pp. 2504–2526, Oct. 2003.
53. V. Veeravalli, Y. Liang, and A. Sayeed, "Correlated MIMO wireless channels: Capacity, optimal signaling and asymptotics," *IEEE Trans. Inform. Theory*, June 2005.
54. A. M. Tulino, A. Lozano, and S. Verdú, "Impact of antenna correlation on the capacity of multiantenna channels," *IEEE Trans. Inform. Theory*, vol. 51, no. 7, pp. 2491–2509, July 2005.
55. J. W. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Trans. Circ. Syst.*, vol. 25, no. 9, pp. 772–781, Sept. 1978.
56. A. F. Molisch, M. Steinbauer, M. Toeltsch, E. Bonek, and R. S. Thomä, "Capacity of MIMO systems based on measured wireless channels," *IEEE J. Select. Areas Commun.*, vol. 20, no. 3, pp. 561–569, Apr. 2002.
57. H. Özcelik, M. Herdin, H. Hofstetter, and E. Bonek, "A comparison of measured  $8 \times 8$  MIMO systems with a popular stochastic channel model at 5.2Ghz," in *10th International Conference on Telecommunications (ICT 2003)*, 2003.
58. Z. Yan, M. Herdin, A. Sayeed, and E. Bonek, "Experimental study of mimo channel statistics and capacity via the virtual channel representation," UW Technical Report, Feb. 2007.
59. J. Kotecha and A. M. Sayeed, "Optimal signal design for estimation of correlated MIMO channels," *IEEE Trans. Signal Process.*, pp. 546–557, Feb. 2004.
60. Z. Hong, K. Liu, R. Heath, and A. Sayeed, "Spatial multiplexing in correlated fading via the virtual channel representation," *IEEE J. Select. Areas Commun.* (special issue on MIMO systems), vol. 21, no. 5, pp. 856–866, June 2003.
61. K. Liu and A. M. Sayeed, "Space-time D-block codes via the virtual MIMO channel representation," *IEEE Trans. Wireless Commun.*, May 2004.
62. A. M. Sayeed, J. Kotecha, and Z. Hong, "Capacity-optimal linear dispersion codes for correlated MIMO channels," paper presented at the IEEE Vehicular Technology Conference, Los Angeles, CA, Sept. 2004.
63. H. Bölcseki, "MIMO-OFDM wireless systems: Basics, perspectives and challenges," *IEEE Wireless Commun.*, vol. 13, pp. 31–37, 2006.
64. E. N. Onggosanusi, A. M. Sayeed, and B. D. Van Veen, "Multi-access interference suppression in canonical space-time coordinates: A decentralized approach," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 833–844, May 2002.
65. T. Sivanadyan and A. Sayeed, "Space-time reversal techniques for information retrieval in wireless sensor networks," paper presented at the Sensor, Signal and Information Processing Workshop (SenSIP'08), Sedona, AZ, May 2008.
66. E. Biglieri, J. Proakis, and S. Shamai(Shitz), "Fading channels: Information-theoretic and communications aspects," *IEEE Trans. Inform. Theory*, pp. 2619–2692, Oct. 1998.
67. A. Ephremides and B. Hajek, "Information theory and communication networks: An unconsummated union," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2416–2434, 1998.
68. J. Mitola III, "Cognitive radio: An integrated agent architecture for software defined radio," PhD thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, 2000.
69. S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Select. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

70. *IEEE J. Select Areas Commun.*, special issue on adaptive, spectrum agile and cognitive wireless networks, Apr. 2007.
71. Q. Zhao and B. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Process. Mag.*, May 2007.
72. *IEEE J. Select Topics Signal Processing*, special issue on signal processing and networking for dynamic spectrum access, Apr. 2008.
73. *IEEE J. Select Topics Signal Processing*, special issue on adaptive waveform design for agile sensing and communication, June 2007.
74. A. M. Sayeed, "Sparse multipath wireless channels: Modeling and implications," in *Proc. ASAP*, 2006.
75. V. Raghavan, G. Hariharan, and A. M. Sayeed, "Capacity of sparse multipath channels in the ultra-wideband regime," *IEEE J. Select. Topics Signal Process.*, Oct. 2007.
76. G. Hariharan and A. Sayeed, "Minimum probability of error in sparse wideband channels," paper presented at the 44th Annual Allerton Conference, 2006.
77. E. J. Candès and M. B. Wakin, "People hearing without listening: An introduction to compressive sampling," *IEEE Signal Process. Mag.*, Mar. 2008.

---

**PART II**

---

**NOVEL TECHNIQUES FOR AND  
APPLICATIONS OF ARRAY SIGNAL  
PROCESSING**



## CHAPTER 6

---

# Implicit Training and Array Processing for Digital Communication Systems

Aldo G. Orozco-Lugo<sup>1</sup>, Mauricio Lara<sup>1</sup>, and Desmond C. McLernon<sup>2</sup>

<sup>1</sup> Cinvestav-IPN, México,

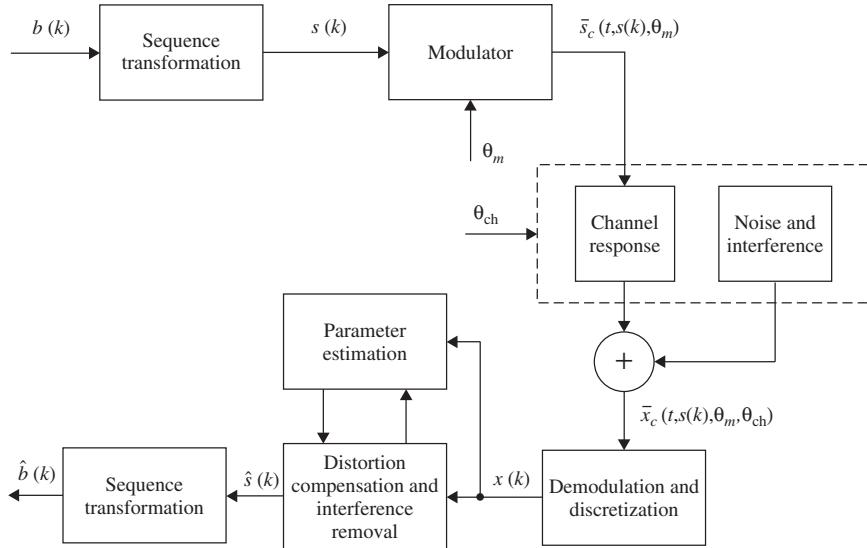
<sup>2</sup> University of Leeds, United Kingdom,

### 6.1 INTRODUCTION

While the term array processing is well understood, the expression implicit training has recently appeared in the digital communications literature and its meaning should first be clearly defined. By implicit training (IT) we mean *a strategy where a special sequence is embedded in the transmitted signal for the purpose of aiding the parameter estimation tasks at the receiver but in such a way that no additional bandwidth is required*. Notice that this definition does not rule out the possibility of employing extra bandwidth for other purposes, as, for example, when channel coding or linear precoding techniques are used to mitigate deleterious channel effects. As the chapter proceeds, it will become apparent that many techniques proposed before under widely different names can now all be grouped under the umbrella definition of IT.

So this chapter deals with the use of IT techniques for parameter estimation and detection problems in digital communications systems that could employ array processing. The chapter starts with a description of the IT methodology and the kind of detection and estimation problems that appear in digital communications systems and that can be solved by the use of IT. We then proceed to give a thorough review of recent research work associated with IT and clearly describe the current state of the art. Once the IT framework is fully established and a historical account of the state of the art has been given, the chapter focuses on array processing based around IT. In particular, digital beamforming and source separation problems are examined for both flat and frequency-selective channels. Our contribution concludes with the most recent and interesting application of IT-based array processing to digital communications—multiple packet reception (MPR). Finally, some open problems and possible future research directions are given at the end of the chapter.

Now, we will focus throughout on digital communications systems that employ linear modulation. A simplified block diagram of such a system is shown in Figure 6.1. The elements of the sequence  $b(k)$  are complex numbers that belong to a predefined



**Figure 6.1** Basic elements of a digital communications system.

constellation such as QPSK or QAM. This way,  $b(k)$  represents the discrete-time complex baseband information data values, which in practical systems are obtained after a suitable manipulation (e.g., source and channel coding) of a binary stream representation of the source. In some systems, the sequence  $b(k)$  is further transformed into a new complex baseband sequence  $s(k)$  of numbers that do not necessarily belong to the original constellation. As an example, in systems that use OFDM, this transformation is an inverse fast fourier transform (FFT). Other possible transformations include the combination of data values and training symbols, and more will be developed later. Let the signal  $s(k)$  be applied to the input of the modulator every  $T_s$  seconds. Essentially, the digital modulator transforms the discrete-time (complex) sequence into a continuous-time bandpass signal waveform  $\bar{s}_c(t)$ .

The channel includes the medium through which the signal travels from transmitter to receiver and the noise and interference that could be present in the system. Note that the implicit assumption of linearity allows the representation of the noise and the interference as an isolated block. At the receiver, the demodulator and discretization block processes the received  $x_c(t)$  bandpass signal and delivers the discrete-time sequence  $x(k)$ . The sequence  $x(k)$  is the complex baseband discrete-time corrupted version of the transmitted sequence  $s(k)$ . Further processing carried out by the subsequent blocks restores an approximate version  $\hat{b}(k)$  of the transmitted information sequence  $b(k)$ .

The crux of the matter is that the transmitted signal suffers distortion, in addition to contamination by noise and interference, while traveling through the channel. In order to correctly recover the transmitted information, an implicit or explicit knowledge of both distortion and interference is needed at the receiver, which is taken into account while processing the received signal to generate  $\hat{b}(k)$ . Distortion and interference can be parameterized by using an appropriate model. Once this model is available, we are

able to use the powerful theoretical tools of estimation theory to obtain an approximate knowledge of its defining parameters.

Let us then examine the kind of parameter estimation and detection problems that appear in digital communications. The digital modulator produces a continuous-time bandpass signal  $\bar{s}_c(t, s(k), \theta_m)$ , which is a function of a certain modulation parameter set  $\theta_m$ , the sequence  $s(k)$ , and, of course, the time variable  $t$ . The continuous-time bandpass channel output can be expressed as  $\bar{x}_c(t, s(k), \theta_m, \theta_{ch})$ , where  $\theta_{ch}$  represents a set of channel-dependent parameters. Normally, the channel-dependent parameters and some of the modulator-dependent parameters are unknown to the receiver, and the problem is that the information sequence  $b(k)$  cannot, in general, be retrieved from the received signal  $\bar{x}_c(t, s(k), \theta_m, \theta_{ch})$  without the knowledge (or at least an approximation) of both  $\theta_m$  and  $\theta_{ch}$ . Therefore, in order to recover the information sequence, it is necessary to estimate both the unknown sets of parameters  $\theta_m$  and  $\theta_{ch}$ , either directly or indirectly. Using these estimates, the distortion correction block produces  $\hat{s}(k)$ , which is an approximation to the sequence  $s(k)$ . Using  $\hat{s}(k)$  an approximation  $\hat{b}(k)$  to the information sequence is obtained after a sequence transformation. Another approach is to perform a joint estimation and detection of  $b(k)$ ,  $\theta_m$  and  $\theta_{ch}$ . This occurs, for example, in iterative timing recovery [1] and turbo equalization [2]. Unfortunately, in most of the cases this joint estimation procedure is extremely difficult to carry out, and, therefore, the first approach is generally adopted in practice.

### 6.1.1 System Parameters and Link Model

In this section we will show how to relate the channel- and modulator-dependent parameters to a discrete-time model of the communications system. Let us first elucidate the nature of the sets  $\theta_m$  and  $\theta_{ch}$ . Typical parameters belonging to the set  $\theta_m$  are the frequency and initial phase of the carrier, the period and initial epoch of the clock, the nonlinearity in the power amplifier, and the band-limiting filter pulse shape. On the other hand, and considering, for example, a multipath propagation channel, typical entries of the set  $\theta_{ch}$  are the attenuation factors, propagation delays, angles of arrival, and Doppler shifts of the different multipath trajectories. Besides, parameters associated with the nature and intensity of noise and interference could also be considered as part of  $\theta_{ch}$ . It is important to mention that the unknown parameters are not restricted to be time-independent, deterministic variables or functions. They can be time-dependent or even in a more complicated situation, random processes.

To clarify matters, let us bring in an example where the dependence of  $\bar{s}_c(t)$  and  $\bar{x}_c(t)$  on the modulator and channel parameters is made explicit. Consider a linear modulator with carrier frequency  $f_M$ , carrier phase  $\phi_M$ , symbol period  $T_s$ , and transmitter shaping filter  $g_{TX}(t)$ . Then the bandpass transmitted signal is given by

$$\bar{s}_c(t) = \operatorname{Re} \left\{ s_c(t) e^{j(2\pi f_M t + \phi_M)} \right\}, \quad (6.1)$$

where  $s_c(t) = r(t) * g_{TX}(t)$  and

$$r(t) = \sum_k s(k) \delta(t - kT_s). \quad (6.2)$$

Now, if  $h_{\text{CH}}(t)$  represents the complex baseband channel impulse response and  $\bar{n}_c(t)$  is bandpass white noise, then the bandpass received signal is given by

$$\bar{x}_c(t) = \operatorname{Re} \{s_c(t)e^{j(2\pi f_M t + \phi_M)}\} * \operatorname{Re} \{h_{\text{CH}}(t)e^{j2\pi f_M t}\} + \bar{n}(t). \quad (6.3)$$

Suppose now that the demodulator operates with a down-converting carrier frequency  $f_D$  and carrier phase  $\phi_D$ , and it is followed by a receiver filter  $g_{\text{RX}}(t)$  matched to  $g_{\text{TX}}(t)$ . Then the complex baseband representation of the continuous-time demodulated signal can be written as [3]

$$x_c(t) = e^{j\phi_R} e^{j2\pi f_R t} [r(t) * h_c(t)] + n_c(t) \quad (6.4)$$

where  $n_c(t)$  is the filtered complex baseband noise, and  $h_c(t) = g_{\text{TX}}(t) * h_{\text{CH}}(t) * g_{\text{RX}}(t)$  is the overall complex baseband continuous-time system impulse response. Also  $\phi_R = \phi_M - \phi_D$  and  $f_R = f_M - f_D$  are, respectively, the carrier phase and frequency offsets. If the signal is further sampled at time instants  $t = t_0 + kT_s$ , then using (6.2), we obtain the sampled version of (6.4):

$$x_c(t_0 + kT_s) = e^{j\phi_R} e^{j2\pi f_R(t_0 + kT_s)} \sum_{m=-\infty}^{\infty} s(m) h_c(t_0 + kT_s - mT_s) + n_c(t_0 + kT_s). \quad (6.5)$$

Finally, assuming a causal channel response limited to  $M$  samples, the discrete-time version becomes

$$x(k) = e^{j\phi_0} e^{j2\pi f_0 k} \sum_{n=0}^{M-1} s(k-n) h(n) + n(k), \quad (6.6)$$

where  $\phi_0 = \phi_R + 2\pi f_R t_0$ ,  $f_0 = f_R T_s$ . Also  $h(k)$  and  $n(k)$  are, respectively, the discrete-time complex baseband versions of the channel impulse response and system noise. The first factor ( $e^{j\phi_0}$ ) on the right-hand side of the above equation, representing the phase offset, is often absorbed by  $h(n)$  in the technical literature.

Let us now consider a particular channel and compute the overall system impulse response. Consider a nominal carrier frequency  $f_M = 900$  MHz and a symbol period  $T_s = 1 \mu\text{s}$ . Let the receiver have a two-element antenna array with element spacing equal to one half-wavelength. The transmitter and the receiver both use root raised cosine shaping filters with a roll-off factor  $\beta = 0.15$ . The transmitted signal is received via two different paths whose amplitudes, delays, and angles of arrival are given in Table 6.1. In this example, we really have two system impulse responses, one per receiving antenna. We can combine these two responses in a system impulse response matrix  $\mathbf{h}(n)$ , as shown below.

Thus, the continuous-time baseband channel impulse response is calculated as [4]

$$\mathbf{h}(t) = [h_{c1}(t), h_{c2}(t)]^T = \sum_{i=0}^1 \alpha_i e^{-j2\pi f_c \tau_i} \mathbf{s}(\theta_i) g(t - \tau_i), \quad (6.7)$$

where  $h_{cq}(t)$  is the overall impulse response from the transmitter to the  $q$ th array element;  $g(t)$  is the combined impulse response of the transmitting and receiving

**TABLE 6.1 Parameters of Transmission Paths**

Path Number	Amplitude ( $\alpha$ )	Delay ( $\tau$ )	Arrival Angle ( $\theta$ )
0	0.95	3.2225	-27.67
1	0.23	3.6884	34.31

*Note:* Delays in microseconds. Arrival angles in degrees with respect to the normal of the array.

filters;  $\alpha_i$ ,  $\tau_i$ , and  $\theta_i$  are the amplitude, delay, and angle of arrival of the  $i$ th path, and the summation is over the two transmission paths. Also,  $s(\theta_i)$  is the steering vector of the array associated with the wavefront coming from direction  $\theta_i$ . For this two-element array, the steering vector is computed as [5]

$$\mathbf{s}(\theta_i) = \left\{ 1, \exp \left[ e^{j \frac{2\pi \chi}{\lambda} \sin(\theta_i)} \right] \right\}^T, \quad (6.8)$$

where  $\lambda$  is the carrier wavelength and  $\chi$  the spacing between the array elements. The discrete-time channel coefficients are obtained as a result of sampling (6.7) at time instants  $t_0 + kT_s$  where  $t_0$  is a given initial sampling time. For the particular values in Table 6.1 and assuming  $t_0 = 1 \mu\text{s}$ , we obtain the discrete-time channel responses given below:

$$\begin{aligned} \text{Re}\{h_1(k)\} &= [-0.0180, 0.0315, -0.0812, -0.1809, 0.0415, -0.0218], \\ \text{Im}\{h_1(k)\} &= [-0.0717, 0.1417, -0.8414, -0.1755, 0.0859, -0.0507], \\ \text{Re}\{h_2(k)\} &= [-0.0817, 0.1592, -0.8835, -0.2798, 0.1096, -0.0631], \\ \text{Im}\{h_2(k)\} &= [-0.0278, 0.0506, -0.1836, -0.2191, 0.0554, -0.0297], \end{aligned} \quad (6.9)$$

where  $h_1(k)$  and  $h_2(k)$  are the discrete-time impulse responses of each of the two channels. The impulse responses have been truncated to six symbol periods due to the very small values outside this range.

The previous development shows how to obtain a discrete-time model (6.6) of the digital communications system for particular channel and modulator-dependent parameters. Similar procedures can be used to determine appropriate discrete-time models in other situations, as, for example, when the receiver does not have an exact knowledge of the symbol time period or when a direct current (dc) offset is present. Note that it is not the objective of the present chapter to give appropriate discrete-time models for all the possible situations but instead to point out that once an appropriate model is available, the task of information transmission can be formulated as a parameter estimation/signal detection problem.

The basic communications system that has just been described is appropriate for a single way link between one source and one destination. Although the communication problem involving multiple simultaneous transmissions is considerably more difficult, the receiver and transmitter structures follow similar configurations. In the general multiuser communication scenario, it is possible to observe that the signal received by any entity is actually dependent on the information sequence and the modulator and channel-dependent parameters of all the active transmitters in the communications environment, and the parameter estimation task is thus much more complicated in this case. Finally note that all the transmitted and received signals could be vector valued (as in the previous example, where the receiver uses two antennas). This accounts

for the possibility that each transmitter and receiver could be sending or receiving information via several channels, as is the case, for example, when array processing is employed.

### 6.1.2 Methods of Parameter Estimation in Digital Communications Systems

We have seen before that in order to obtain reliable transmissions under a channel that introduces distortion, parameter estimation is necessary at the receiver. In the example given in the previous section, the unknown parameters of the model are the channel response  $h(k)$  and the carrier frequency and phase offsets  $f_0$  and  $\phi_0$ . Note that once the estimates of these parameters are available and their effects have been accounted for, the data sequence  $\hat{b}(k)$  can be obtained by means of hypothesis testing. Parameter estimation can in principle be obtained from the received signal  $x(k)$  without precise knowledge of the transmitted signal, with the exception of perhaps some of its statistical properties. This is the blind setup, and it is very attractive from the point of view of the transmitter, but the receiver must absorb all the implementation complexity [6–11]. Therefore, practical digital communications systems have been traditionally designed using the explicit training (ET) paradigm for parameter estimation, also known as pilot symbol-assisted modulation (PSAM) [12] or pilot-assisted transmission (PAT) [13].

In ET-based systems, the transmitter sends a training pattern in either time or frequency slots that are not used by data. The transmitted information is thus recovered, aided by the training pattern known to the receiver. Reference [13] presents an overview and detailed examples of state-of-the-art digital communications systems based on ET sequences. A drawback of the ET approach is that the training sequence consumes bandwidth that could otherwise be used for the transmission of information. But in exchange, it greatly simplifies the reception tasks compared to blind methods, particularly in difficult practical channel conditions. In order to avoid the loss of bandwidth incurred by ET approaches and still be able to identify the channel impairments with manageable complexity (as opposed to blind techniques), digital communication receivers could rely on IT, and this is discussed in the next section. Taking the previous remarks into account, we could distinguish three ways to perform the estimation of  $\theta_m$  and  $\theta_{ch}$ .

- *Explicit Training (ET)* In this method a training (pilot) sequence known to the receiver is inserted in the transmitted signal either in time or in frequency. We will refer to these two variants as time-multiplexed training (TMT) and frequency-multiplexed training (FMT), respectively. The receiver exploits the knowledge of both the training sequence and its placement pattern to carry out the necessary tasks. Note that in ET, bandwidth is wasted because of the multiplexed nature of the transmissions.
- *Implicit Training (IT)* In IT a special sequence is embedded in the transmitted signal for the purpose of aiding the parameter estimation tasks at the receiver, but without compromising the transmission bandwidth, as is the case when ET is used. Using both the knowledge of the embedded sequence and the way that the embedding was performed, the estimates of the unknown system parameters can be obtained. In contrast to ET, IT is bandwidth efficient because it does not require

extra bandwidth to convey training information. However, IT normally implies either a waste of power (that can be made small at the expense of estimation accuracy) or in some cases a phase ambiguity that needs to be eliminated by other means.

- *Blind Method (BM)* In this method, no preprocessing of the transmitted signal is carried out, and the estimation of the unknown parameters is accomplished based only on the received signal  $x_c(t, b(k), \theta_m, \theta_{ch})$ . The dependence of this signal on the information sequence  $b(k)$  is normally (but not always) eliminated using statistical averages. Blind estimation is also known as non-data-aided estimation in the field of synchronization [14, 15]. The advantage of BM is that no training is required and information can be conveyed by the system at all times. A special form of blind estimation is known as decision-directed processing. This method uses the detected sequence  $\hat{b}(k)$  (that will act as training) as a substitute for the true information  $b(k)$ . The output of the channel decoder could also be used under some circumstances instead of  $\hat{b}(k)$ . The main disadvantage of decision-directed processing is that parameter estimation is reliable only when the system is working close to the desired operating point, in other words, when the actual estimates  $\hat{\theta}_m$  and  $\hat{\theta}_{ch}$  are close to their true values [15].

Note that, in general, explicit, implicit, and blind approaches could coexist in the same communications system, and this possibility should be considered whenever their combination gives us a definite advantage. For example, ET and BM can be combined in channel equalization, where after an ET approach has opened up the “eye,” the system then switches to decision-directed equalization.

To clarify matters, we now show how to accomplish channel estimation based on ET and on one particular case of IT. To this end, we consider the model given by (6.6) and assume that carrier frequency and phase offsets are zero. In this case, the unknown parameters are the values  $h(k)$  of the channel response. First, (6.6) is rewritten (for  $k = 0, 1, \dots, N - 1$ ) employing matrix notation, as follows:

$$\mathbf{x} = \mathbf{Sh} + \mathbf{n}, \quad (6.10)$$

where  $\mathbf{x} = [x(0), x(1), \dots, x(N - 1)]^T$  is the received sequence vector,  $\mathbf{n} = [n(0), n(1), \dots, n(N - 1)]^T$  represents the noise vector, and  $\mathbf{h} = [h(0), h(1), \dots, h(M - 1)]^T$  is the vector of the  $M$  unknown channel coefficients. Also matrix  $\mathbf{S}$  (dim.  $N \times M$ ) is given by

$$\mathbf{S} = \begin{bmatrix} s(0) & 0 & \cdots & 0 \\ s(1) & s(0) & \ddots & 0 \\ \vdots & \vdots & \ddots & s(0) \\ \vdots & \vdots & \ddots & \vdots \\ s(N - 1) & s(N - 2) & \cdots & s(N - M) \end{bmatrix}, \quad (6.11)$$

where we assume that transmission starts at time zero. It is possible to see that (6.10) represents a linear model between the observation  $\mathbf{x}$  and the unknown vector  $\mathbf{h}$ . In the case of ET, the transmitted (training) sequence  $s(k)$  is known at the receiver, and

one way of estimating  $\mathbf{h}$  is by the use of a standard linear least-squares approach as follows:

$$\hat{\mathbf{h}} = \mathbf{S}^+ \mathbf{x} \quad (6.12)$$

where  $\mathbf{S}^+$  denotes the pseudoinverse of matrix  $\mathbf{S}$ .

Let us now turn to channel estimation based on IT. In this case, the sequence  $s(k)$  is unknown at the receiver. However, in this example, a known training sequence  $c(k)$  is added to the information sequence  $b(k)$  to produce  $s(k)$ , that is,  $s(k) = b(k) + c(k)$ . Further assume that  $c(k) = c(k + P)$  is a periodic sequence of discrete delta pulses, that is,  $c(k) = 1$  for  $k \bmod P = 0$  and  $c(k) = 0$  otherwise. To keep the explanation simple consider  $M = 2$  and  $P = M$ . Furthermore assume that  $N$  is even. In these conditions we have the following model

$$\mathbf{x} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{h} + \begin{bmatrix} b(0) & 0 \\ b(1) & b(0) \\ b(2) & b(1) \\ b(3) & b(2) \\ \vdots & \vdots \\ b(N-2) & b(N-3) \\ b(N-1) & b(N-2) \end{bmatrix} \mathbf{h} + \mathbf{n}. \quad (6.13)$$

If we now define [with  $N_2 = (N - 2)/2$ ]

$$\mathbf{y} = \begin{bmatrix} \sum_{k=0}^{N_2} x(2k) \\ \sum_{k=0}^{N_2} x(2k + 1) \end{bmatrix} = \frac{N}{P} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{h} + \begin{bmatrix} \sum_{k=0}^{N_2} b(2k) & \sum_{k=0}^{N_2} b(2k - 1) \\ \sum_{k=0}^{N_2} b(2k + 1) & \sum_{k=0}^{N_2} b(2k) \end{bmatrix} \mathbf{h} + \begin{bmatrix} \sum_{k=0}^{N_2} n(2k) \\ \sum_{k=0}^{N_2} n(2k + 1) \end{bmatrix}, \quad (6.14)$$

then for zero mean  $b(k)$  and  $n(k)$ , we can estimate the channel via the unbiased and consistent estimate  $\hat{\mathbf{h}} = (P/N)\mathbf{y}$ —that is,  $E(\hat{\mathbf{h}}) = \mathbf{h}$  and  $\lim_{N \rightarrow \infty} (\hat{\mathbf{h}}) = \mathbf{h}$ . Note that although the previous estimate is only asymptotically exact, good estimates are obtained for relatively small values of  $N$  and small values of training power to data power ratio [16]. Note also that the estimator is very simple, requiring only  $N_2$  additions per channel coefficient. This example just intends to show in a simple manner how IT operates. In what follows we consider IT-based processing in more detail.

## 6.2 CLASSIFICATION OF IMPLICIT TRAINING METHODS

In IT-based systems the sequence transformation block depicted in Figure 6.1 serves to implicitly embed the training information onto the information sequence  $b(k)$ . This

training is thus contained in  $s(k)$ . A general representation that includes all the IT techniques that will be described in this chapter is the affine transformation given by

$$\mathbf{s} = \mathbf{Ab} + \mathbf{c}, \quad (6.15)$$

where matrix  $\mathbf{A}$  is known as the precoding matrix,  $\mathbf{b}$  is the vector whose elements are given by  $b(k)$ ,  $\mathbf{c}$  is a bias vector with elements  $c(k)$ , and  $\mathbf{s}$  is the vector whose components  $s(k)$  represent the complex baseband discrete-time transmitted signal, which in turn will be applied to the pulse shaper and modulator to generate the continuous-time transmitted signal  $\bar{s}_c(t)$ . Note that as well as representing IT, the so-called affine precoding model [17] given in Eq. (6.15) also includes as particular cases important linear modulation methods such as CDMA and OFDM [18].

As just stated, the training information is implicitly included in the transmitted signal  $\mathbf{s}$  in Eq. (6.15). This information is contained in matrix  $\mathbf{A}$  and vector  $\mathbf{c}$ , which are assumed known at the receiver. The following sections will discuss particular cases of IT where matrix  $\mathbf{A}$  is given by a square matrix. This choice corresponds to the case where the output ( $\mathbf{s}$ ) and input ( $\mathbf{b}$ ) vectors have the same length—that is, there is no bandwidth expansion, or equivalently, a loss in date rate does not occur.

### 6.2.1 Superimposed Training

When  $\mathbf{c} \neq \mathbf{0}$  in (6.15), the resulting form of IT is called superimposed training (ST). Usually  $\mathbf{A} = \mathbf{I}$ , the identity matrix, but other choices are possible. ST is bandwidth efficient since it does not occupy additional time and/or frequency resources. Unfortunately, the previous advantage also conveys a penalty. On the one hand, power is taken from data to be assigned to training, and this implies that, for a given total transmitted power, the data signal to noise ratio is reduced as some power is allocated to the training sequence. On the other hand, there is a loss of orthogonality between training and data, which has two important implications. One of them is that data acts as noise when it comes to parameter estimation. The other is that training acts as noise when data detection is carried out. It is worth mentioning that the simple example of channel estimation based on IT introduced in the previous section refers precisely to the ST method being discussed here.

Research on ST is relatively new. The ST technique has been used for the purpose of acquiring synchronization in [19, 20], and to the best of our knowledge, the first work that considered the use of ST to estimate the channel impulse response in a single carrier system is presented in [21]. In this work, the basic ST method is described but a channel estimation performance analysis is absent. Another contribution along the same lines appears in [22]. The work in [22] just adds the possibility of performing direct channel equalization using ST but no performance analysis is given nor is the training sequence synchronization (TSS) problem considered. It is important to mention that although [22] focuses on ST, the other two IT schemes considered later on in this chapter have been already briefly mentioned in [22] where the connection of IT with watermarking is established. In [23] a performance analysis of ST is presented for the first time but only for the special case of a training sequence composed of a train of equally spaced discrete delta functions. Again, TSS is not considered.

It is worth mentioning that [22] and [23] both overlooked the original contribution made in [21]. Later on, in the work presented in [16], several new contributions are introduced. The TSS problem is for the first time seriously considered, tackled, and solved employing fourth-order statistics. Also, a more realistic channel model is contemplated by the introduction of an unknown receiver dc offset. This offset commonly appears in modern direct conversion receivers and cannot be neglected in practice, so it must be included in the model as an unknown parameter. Apart from solving the synchronization problem, the dc offset problem is also solved by a polynomial rooting numerical method. Performance analysis is also carried out for any periodic training sequence and a family of sequences is identified that provide desirable channel estimation properties. In [24] and [25], an alternative approach to estimate the dc offset and the channel is proposed, which is based on frequency domain processing. (A later publication [26] showed that both the approaches of [25] and [16] (using frequency- and time-domain approaches, respectively) are essentially equivalent.) Training sequence synchronization is also considered in [27–30].

An interesting line of development follows from the work in [31]. In this work, a new form of ST, called data-dependent superimposed training (DDST), is introduced. It is interesting to mention that an earlier work that uses a similar idea, but for synchronization purposes, is presented in [146]. This new form of IT has the property of canceling the undesirable interfering effects caused by the information data at the time of estimating the channel. Remember that ST is just a superposition of data and training and so both sequences interfere with each other, as mentioned before. So, due to the fact that when using DDST the interference coming from the data is removed, the channel estimation performance is vastly improved, and this gives perfect estimation under noiseless conditions. On the downside, DDST requires the insertion of a cyclic prefix (that can be exploited for frequency-domain equalization) for its proper operation, but it may operate without it with minimal degradation [32]. It is worth mentioning that in DDST the square precoding matrix  $\mathbf{A}$  is rank deficient [31]. Thus, vector  $\mathbf{b}$  cannot be recovered from  $\mathbf{s}$  by subtracting  $\mathbf{c}$  and inverting  $\mathbf{A}$  (the inverse does not exist). However, exploiting the finite alphabet property of the data, the estimate of data vector  $\mathbf{b}$  can be improved by nonlinear processing [31]. DDST is very attractive for low-order constellations such as QPSK, but it is less adequate when used in conjunction with higher-order constellations such as QAM64. This is so because the rank deficiency of matrix  $\mathbf{A}$  provokes a data identifiability problem that has been concisely reported in [33]. Issues concerning synchronization and dc offset estimation applicable under the DDST framework have already been considered in both [29] and [30].

Other lines of research in ST are as follows. In [34–39] iterative channel estimation procedures are considered. An investigation on the channel estimation improvement, achieved by the use of a bases expansion for the band-limited channel, is presented in [40]. Issues concerning the optimal assignment of power for data and training have been examined in [41]. The application of ST for CDMA is investigated in [42]. The use of ST alleviates the peak-to-average power ratio problem that occurs in OFDM, as is elucidated in [43]. Other works that deal with the application of ST to OFDM are [44–47]. The work in [48] considers the application of ST for ultra-wideband communications, whereas [49] studies the applicability of using ST jointly with space–time coding. In [34, 50–53] issues concerning the use of ST for systems with multiple transmit and multiple receive antennas (MIMO systems) have been presented. The work in [54] explores the possibility of using ST to achieve packet separation based on array

processing. Estimation of double-selective channels is investigated in [55–61a] while direct equalization for this type of channels is explored in [62].

The previous paragraphs show that the theory, applications, and performance analysis associated with ST are reaching a mature state and so it is convenient at this point to start exploring adequate implementation architectures for the practical realization of digital transceivers based on this technique.

### 6.2.2 Time-Varying Transmitted Power

When  $\mathbf{A}$  in (6.15) is a diagonal matrix possessing real positive elements with at least two different values and vector  $\mathbf{c} = \mathbf{0}$ , then we have the IT method that we call time-varying transmitted power (TVTP). In this case, the data  $b(k)$  is multiplied by a sequence with modulus variations to form the transmitted baseband sequence  $s(k)$ . The effect of the multiplication is just an expansion or contraction of the constellation. As with the other two IT techniques discussed in this chapter, TVTP is bandwidth efficient. Moreover, as opposed to ST, power is not taken from data to be assigned to training. However, the fact that TVTP induces modulus variations onto the transmitted signal results in an increase in the uncoded BER compared to the case when no modulus variations are imposed on the data [3, 63]. TVTP is the only IT technique considered here that has the advantage of insensitivity to carrier frequency offsets (CFO). In other words, parameter estimation algorithms can be developed at the receiver without worrying too much about CFO, which can then be compensated once the other tasks have been performed. The main disadvantage of TVTP is that it does not allow us to obtain carrier phase information at the receiver, so forcing us to apply either differential modulation at the transmitter (paired by differential detection at the receiver) or to provide a complementary way to obtain the carrier phase information.

Research work on TVTP can be summarized as follows. The TVTP operation is proposed for the first time in [64] where an almost periodic variation in power is introduced to the source symbols. Based on this amplitude variation, a subspace-based channel estimation procedure is proposed. At about the same time, a multirate filter bank structure at the transmitter is presented [65]. This arrangement allows simple and efficient channel estimation and equalization methods at the receiver. This filter bank introduces redundancy that allows zero-forcing equalization irrespective of the channel zero locations. Before we proceed, let us make a brief comment. Note that in the case of symbol-rate processing, finite impulse response (FIR) equalizers are not in general able to accomplish zero-forcing equalization. When oversampling is employed, zero-forcing FIR equalizers are indeed possible, but their existence depends on the channel zero locations [66]. This way, the scheme proposed in [65] guarantees the existence of zero-forcing FIR equalizers at the cost of a loss in data rate (due to redundancy). Although the framework introduced in [65] includes TVTP as a special case when there is no loss in data rate, the main motivation of the study is on perfect FIR equalization irrespective of channel zero locations, which is only possible when redundancy is introduced.

A periodic TVTP operation is first proposed in [67] where it is exploited to achieve “blind” channel estimation using a correlation matching approach. Independently, and at about the same time, [68] introduces modulation-induced cyclostationarity (MIC), which is in fact periodic TVTP (PTVTP). The focus of [68] is on “blind” channel

estimation, but this time exploiting the induced cyclostationary statistics using subspace methods. The application of TVTP in a single-carrier MIMO spatial multiplexing scenario can be traced back to the work in [69] while the multicarrier case is treated in [63]. Sometime later, under the same framework of [68], [70] finds TVTP sequences possessing certain optimality properties for the channel estimation task. Based on the previous work, [71] tackles the channel identification problem in the context of block transmissions with cyclic prefix and frequency-domain equalization, whereas [72] generalizes the approach to the estimation of MIMO channels. Periodic TVTP in conjunction with array processing is used in [73] with the objective to separate two sources by spatial filtering. To this end, a known modulus algorithm (KMA) is employed, whose cost function is quartic in the beamformer weights and so may exhibit local minima. However, conditions on the characteristics of the periodic IT-TVTP sequence are found that guarantee a cost function free of local minima. Later on, this approach is generalized in [74] to the case of multiple sources and frequency selective channels.

Periodic TVTP has also been used for CDMA systems [75–77] and for channel estimation under carrier offset conditions in [78]. Source separation based on power variations is also analyzed in [79] and [80] where a nonperiodic power variation pattern is exploited. In [79], advantage is taken of two different levels of transmitted power of the sources over two consecutive blocks of symbols. Provided that no two sources change their power by the same proportion, the method asymptotically delivers a zero forcing source separation solution that can be obtained in closed form based only on second-order statistics of the received signal. The work in [79] is later generalized in [80] and [81] to the case where the power variation of sources could be applied in more than two intervals. However, to perform source separation while exploiting the statistics of more than two intervals requires parallel factor analysis, which is a computationally demanding technique. In [82] TVTP is exploited to achieve multipacket reception based on an analytical known modulus algorithm (AKMA). This work is later extended and complemented to achieve joint source separation and synchronization in [83, 84].

### 6.2.3 Constellation Rotation Transmission

For this method, the matrix  $\mathbf{A}$  in (6.15) is also diagonal and vector  $\mathbf{c}$  is also zero as in the previous case. The difference is that the elements of  $\mathbf{A}$  are now complex valued and possess constant modulus. The effect of the multiplication is a time-varying rotation (so the name CRT) of the transmitted constellation points. The use of CRT does not increase the bandwidth and does not imply a waste of data power in favor of training power. Furthermore, due to the fact that rotations do not induce modulus variations, the uncoded BER will not be degraded as happens in the case of TVTP. So it seems that CRT gives us only advantages and that it should be preferred over the previous two approaches. Unfortunately, this is not true. IT-CRT also has the disadvantage of not allowing us to obtain carrier phase information at the receiver, restricting us to noncoherent communication if no alternative way of obtaining carrier phase information is available.

The exploitation of cyclostationarity in communications systems is outlined in [85]. When IT-based techniques use periodic sequences, cyclostationarity is induced onto the transmitted signal. The employment of cyclic statistics to perform interference removal in array processing can be traced back to [86]. This work shows that sources impinging onto an antenna array can be separated based upon their distinct cyclic statistics. Unrelated baud rates and/or carrier frequencies result in different cyclic statistics. The source

separation algorithms proposed in [86], which form a group known as the SCORE family, accept a closed-form solution that can be obtained by linear algebra tools. To simplify computational burden, [87] proposes adaptive versions of the algorithms in the SCORE family. The work in [88] presents algorithms with enhanced estimation capabilities as compared to those of the SCORE family. As mentioned above, the three previous works exploit differences in baud rates and/or carrier frequencies between the sources. However, they are not very specific as to how to guarantee that different sources have indeed unrelated baud rates and/or carrier frequencies. Later on, the contribution in [89] states that carrier frequency offset-based algorithms converge faster than their baud rate counterparts and so proposes to induce different small frequency offsets to the sources. This approach shifts the spectrum of individual sources and so provokes an increase in the overall system bandwidth.

Fortunately, it is possible to still induce cyclostationary statistics without bandwidth expansion or spectrum shifting by the use of CRT. In fact, the complex baseband representation of the received signal is the same in both cases, that is, by introducing small amounts of frequency offsets as in [89] or by using CRT. This way, via the use of CRT and without compromising bandwidth, the algorithms proposed in the four previous works can be efficiently utilized in practice. To the best of our knowledge, CRT is proposed for the first time in [90], where it is used to accomplish MIMO channel estimation for two users where each one is modulated by a different complex exponential sequence. The estimation algorithm exploits second-order conjugate cyclostationary statistics. Later, a complex chirp CRT sequence is proposed to achieve direct equalization in [91]. It is worth mentioning that in this work, CRT is exploited at the receiver not by using cyclostationary statistics but by taking advantage of the strong structural properties of the chirp modulation sequence. In fact, this is a deterministic algorithm that will deliver perfect estimates in the absence of noise, contrary to the methods based on cyclic statistics. This work is later expanded and reported in [92]. Another deterministic approach is proposed in [93], where CRT is employed to achieve source separation through MIMO array processing. Basically, the transmitters use several antennas and the same data is sent through all the antennas but a different CRT modulation is applied to each antenna. In [94] CRT is used as a possible solution to the multipacket reception problem. The CRT modulating sequences are polynomial phase sequences (PPS). The packet separation algorithm is deterministic and the method is shown to outperform the proposals in [88, 89].

#### 6.2.4 Discussion

Although in this chapter we are focusing on IT schemes that do not reduce data rate, it is important to mention that interesting properties are obtained when redundancy is introduced by the use of a tall matrix  $\mathbf{A}$  in Eq. (6.15), as discussed next. Equation (6.15) represents what is known in geometry as an affine transformation [95], which is formed by a linear transformation ( $\mathbf{Ab}$ ) followed by a translation ( $\mathbf{c}$ ). The use of a linear transformation at the transmitter is also known as linear precoding in the communications field [96]. When the channel is unknown at the transmitter, a linear precoder can be used to mitigate the effects of spectral nulls caused by a frequency selective channel [65]. This is achieved at the expense of reducing data rate, in other words, making matrix  $\mathbf{A}$  tall.

Unlike linear precoding, affine precoding was introduced very recently in [17, 97] where it is argued that an affine precoder is more efficient than the combination of a

linear precoder plus explicit time multiplexed training, due to the fact that redundancy is better exploited. In [17, 97, 98], the ST vector  $\mathbf{c}$  is orthogonal to the column space of matrix  $\mathbf{A}$ , in such a way that  $\mathbf{c}^T \mathbf{A} = \mathbf{0}$ , which has the effect of making the data and training orthogonal. Note that in order to make  $\mathbf{c}^T \mathbf{A} = \mathbf{0}$ , when  $\mathbf{A}$  is full column rank, it is necessary for  $\mathbf{A}$  to be a tall matrix, and thus the data rate must be reduced. In ST, matrix  $\mathbf{A}$  is square and full rank, and so it is impossible to make the ST sequence orthogonal to the column space of  $\mathbf{A}$ , with the consequence that data acts as noise during the channel estimation process, as stated before. However, when  $\mathbf{A}$  is rank deficient, vector  $\mathbf{c}$  can indeed be made orthogonal to the column space of  $\mathbf{A}$  even in the case when  $\mathbf{A}$  is square, just as happens in DDST. Indeed, in DDST, matrix  $\mathbf{A} = \mathbf{I} - \mathbf{J}$  where  $\mathbf{I}$  is the identity matrix and  $\mathbf{J} = (1/Q)\mathbf{1}_Q \otimes \mathbf{I}_P$ ,  $\mathbf{1}_Q$  is the  $(Q \times Q)$  matrix of all ones,  $\mathbf{I}_P$  is the  $(P \times P)$  identity matrix, and  $\otimes$  denotes the Kronecker product. Also,  $P$  is the period of the ST sequence and  $Q$  is the number of training sequence periods contained in one transmitted block of symbols [31].

We would like to stress that IT has previously been considered as belonging to the class of “blind” techniques, and many of the previous cited works on IT actually introduce the method as a blind technique. The reason for this relationship is justified by the fact that blind estimation in its broader sense refers to a situation where the system is unknown and a precise knowledge of the input signal is not available, as happens when IT is used. However, blind estimation could be defined in a strict sense as the situation where the system is unknown and the transmitted signal has not been specially manipulated for the purpose of aiding the estimation process, or if it has, the receiver is completely unaware of the manipulation procedure. Provided we adhere to the strict sense definition, IT can be considered as an independent concept, as was indeed done in Section 6.2. This way, we will have explicit, implicit, and blind estimation approaches as three separate entities. This is the point of view that we take in this chapter.

Finally, we are concerned in pointing out that IT is not reduced to affine precoding. In fact, nonlinear transformations that embed the training sequence inside the transmitted signal can also be considered IT techniques. It remains to see if this more general nonlinear framework possesses tractable complexity and gives us additional gains. Note also that although we have considered that IT does not expand the bandwidth (as nonredundant affine precoding), redundant affine precoders can also be considered as part of the IT framework in the case when redundancy is added to mitigate the deleterious effect of the channel and not for aiding parameter estimation tasks at the receiver, so fulfilling our definition of IT. Note that there could be certain cases where a clear-cut line between IT-redundant affine precoders and non-IT-redundant affine precoders does not exist.

### 6.3 IT-BASED ESTIMATION FOR A SINGLE USER

Communication systems are easier to analyze when broken down into small tasks. However, each of these tasks in itself is complex enough to require the intervention of an expert that specializes in a particular process of the system. This explains the fact that the majority of research studies focus on particular aspects of the whole communications problem. This situation in turn might have as a result that the understanding of the complete system and the interrelation between the parts are sometimes difficult to master.

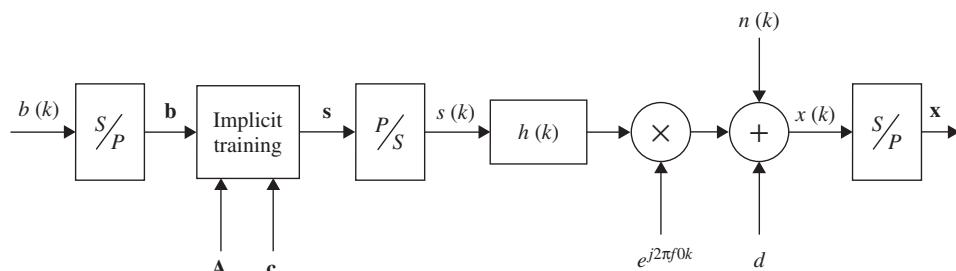
In this section we will endeavor to explain some of the difficulties that are faced when assembling a practical digital communications system. To this end, a discrete-time model for the digital communications system (like the one presented in the introduction) will be used that includes most of the impairments that are likely to be found in actual practice. This will hopefully result in a better understanding of the interrelation that exists between different blocks in the system. In particular, we will focus on the way that IT can be used to build a working system. By doing this, we will be able to verify whether or not a complete solution can be developed based on already published research material on IT, which also has the added advantage of identifying open research problems. The next section introduces the model of the communications system when a single user is transmitting. Later on we will deal with the more complex case of a multiuser scenario where the receiver has multiple antennas. The model is described in discrete-time nomenclature because this way the impact that distinct channel impairments and countermeasures could produce is easier to understand. This is because countermeasures are normally carried out in discrete-time by digital signal processing.

### 6.3.1 System Model

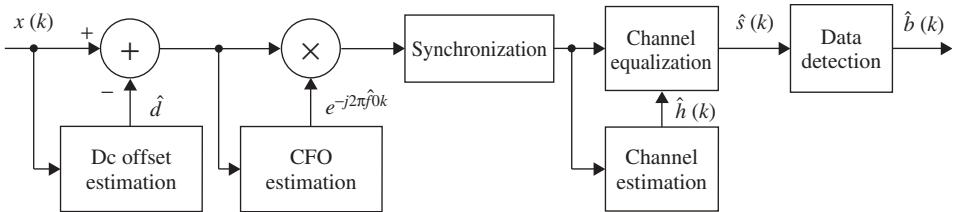
Figure 6.2 shows a block diagram of the discrete-time representation of a digital communications system. It includes IT at the transmitter. The channel impairments represented by the system are the following: channel frequency selectivity is taken into account by the channel impulse response  $h(k)$ ; carrier frequency offset is represented by the term  $e^{j2\pi f_0 k}$ ; carrier phase offset is embedded into the channel impulse response; dc offset given by the complex (in general) variable  $d$ ; and finally additive Gaussian noise is represented by  $n(k)$ . It is also important to highlight that transmission delays are accounted for by  $h(k)$ . Moreover, a time-varying channel impulse response can also be used to model timing differences between the transmitter and the receiver data generating clocks. Considering the system model depicted in Figure 6.2, it is not difficult to show (see also the example introduced in Section 6.1.1) that one possible relation between the transmitted and received signal vectors is [3]

$$\mathbf{x} = \mathbf{D}\mathbf{H}\mathbf{s} + \mathbf{n}, \quad (6.16)$$

where  $\mathbf{s}$  is given in (6.15), the elements of matrix  $\mathbf{D}$  are dependent on the carrier frequency offset, and matrix  $\mathbf{H}$  takes into account the channel impulse response, carrier



**Figure 6.2** Discrete-time representation of a digital communications transmission using IT.



**Figure 6.3** Discrete-time representation of a digital communications receiver exploiting IT.

phase, and timing related problems. Finally, vector  $\mathbf{n}$  includes both the dc offset and additive noise.

Figure 6.3 shows a possible implementation of the receiver. Note that in this case reception tasks are performed sequentially and in a particular order. This is, of course, not the only option, and we will indicate alternative configurations as we proceed. In Figure 6.3, dc offset is estimated and compensated first, followed by CFO estimation and compensation. Then training sequence, frame, and perhaps clock synchronization tasks are carried out. Once the three previous tasks have been completed, we proceed to perform channel estimation and equalization, although direct channel equalization is also possible. Finally, data detection is accomplished.

In the following sections we will provide a review of several research works that address the type of estimation problems that have been exemplified in Figure 6.3. Note that although this section is intended for IT-based parameter estimation in the single-user case, the revision that follows also includes some references that deal with the general multiuser problem.

### 6.3.2 DC Offset Estimation and Compensation

*ST-Based Estimation* The dc offset is difficult to deal with in ST-based systems because in general it interferes with channel identification. The compensation of dc offset is easily accomplished after its estimation by a simple subtraction. In [16] training sequence synchronization and dc offset estimation are carried out jointly using higher-order statistics and polynomial rooting. It is worth mentioning that higher order statistics are necessary because the period of the ST sequence is equal to the channel model order plus one, which represents the smallest value of the period necessary to obtain unique channel estimates. Although this approach is interesting from a theoretical standpoint, it is not very adequate for practical implementation due to its high computational complexity. A simpler dc offset estimation approach is presented in [24, 25]. In these cases, the simplicity results from increasing the period of the ST sequence beyond the minimum required to achieve unique channel identification, as stated above. Thus the channel is estimated in the frequency domain by avoiding the dc component that is (in effect) impaired by the unknown dc offset. Finally, after removing the contribution of the training sequence on the received signal, the dc offset is obtained via simple averaging (exploiting the zero-mean property of the transmitted information). A different approach where the ST sequence is designed to have zero mean is suggested in [99]. In this case, and taking into account the zero-mean property of the data, the unknown dc offset can be estimated by averaging the received signal over a given interval. Another approach for dc offset estimation can be found in

[29, 30] where training sequence synchronization, channel, and dc offset estimation are jointly performed using vector space projections.

*TVTP- and CRT-Based Estimation* For these two approaches dc offset is not a problem when the transmitted information data has zero mean since in this case the dc offset can be simply estimated by averaging the received signal over a given interval.

### 6.3.3 Frequency Offset Estimation and Compensation

*ST-Based Estimation* Superimposed training for CFO estimation under AWGN channels is first proposed in [19, 20], and later on applied to OFDM in [100] and to frequency-hopped OFDM in [101]. For frequency selective channels, the following results have been reported. The approach introduced in [102] is based on DDST. Using an additional data-dependent sequence, certain bins in the frequency domain are nulled. It should be noted that a non-zero carrier frequency offset provokes a cyclic shift in the frequency domain. Therefore, an estimate can be obtained by finding a cyclic shift that restores the nulls to their original positions. In [103], the previous method is enhanced by also exploiting the energy contained in the superimposed sequence  $c(k)$ . The previous two approaches deliver adequate performance but, in order to be successfully applied, they require frame synchronization to determine the boundaries of the transmitted DDST block. Finally, in [104] three CFO estimation procedures are proposed that operate under the ST framework. They are appealing because they can be applied before training sequence synchronization and channel estimation are obtained. CFO estimation based on ST has also been proposed for multicarrier CDMA systems in [105].

*TVTP-Based Estimation* In this case, frequency offset estimation can be carried out after all other estimation tasks have been performed. The reason relies on the fact that synchronization and channel equalization can be made insensitive to CFO when TVTP is exploited. This way, CFO estimation can be accomplished after equalization [78] and removal of the TVTP modulating sequence employing well-established non-data-aided algorithms that operate under the AWGN channel [14, 15].

*CRT-Based Estimation* To the best of the authors' knowledge, no CFO estimation methods are yet available for this case.

### 6.3.4 Frame and Training Sequence Synchronization

*ST-Based Synchronization* It is highlighted in [16, 21] that, in the case of periodic ST, synchronization of the ST sequence at the receiver is important if a circular shift ambiguity of the channel coefficient vector is to be avoided. Note that this type of ambiguity could render proper equalization impossible, and so it becomes mandatory to acquire training sequence synchronization (TSS) for correct channel estimation. Joint dc offset estimation and synchronization based on higher order statistics is introduced in [16]. Another approach is to obtain synchronization based on model fitting, where a certain cost function is minimized when the correct synchronization point is found [27]. In [28] an attractive (low complexity) algorithm based on vector space projections is introduced, which is later extended to deal with dc offset in [29]. When block

transmissions are taking place, frame synchronization is important. The first work addressing frame synchronization for AWGN channels based on ST can be found in [106]. The work in [30] targets frame synchronization for DDST.

**TVTP-Based Synchronization** In the case of AWGN channels, TVTP-based synchronization is almost straightforward to achieve when the TVTP pattern is periodic. This can be accomplished by downsampling the symbol-rate-sampled received sequence by a factor equal to the period (let us say  $P$ ) of the TVTP sequence in order to form  $P$  substreams. The power in each substream can then be estimated and compared to the one impressed onto the transmitted TVTP pattern to obtain the correct synchronization point. In the case of frequency selective channels, the situation is more complicated, and as far as we know no method to acquire synchronization has yet been presented in the open literature. However, instead of acquiring synchronization with the purpose of obtaining correct channel identification, we can use TVTP to obtain direct equalizer estimation. In this case, synchronization and equalization can be jointly obtained because the equalizer will synthesize the delay that coincides with the TVTP pattern; see [74].

**CRT-Based Synchronization** Under appropriate selection of the period of the CRT sequence, synchronization can be jointly accomplished with direct equalization [92]. No channel estimation method for frequency selective channels (and hence no synchronization technique) based on CRT has yet been proposed.

### 6.3.5 Channel Estimation

**ST-Based Estimation** Channel estimation based on ST has been proposed in [16, 21–25] by exploiting first-order cyclostationary statistics. All the previous formulations are formally equivalent, with the differences consisting mainly on the selection of the training sequences and on the domain where the estimation is performed [26].

**TVTP-Based Estimation** Channel estimation methods have been proposed in [64, 67, 68, 70, 71, 78]. A structured subspace method is considered in [64], whereas [68] introduces closed-form linear solutions, subspace, and nonlinear correlation matching approaches. A correlation matching approach is also proposed in [67]. Channel identification and CFO estimation are presented in [78]. Estimation based on channel outer products is introduced in [70]. Channel estimation for block transmissions with frequency-domain equalization is treated in [71].

**CRT-Based Estimation** We are unaware of channel estimation methods.

### 6.3.6 Channel Equalization

Channel equalization can be performed once channel information is available; alternatively, direct equalization is also an option. This section considers only those proposals that use the IT sequence to achieve direct equalization.

**ST-Based Equalization** Direct zero-forcing equalization is considered in [22] while the work in [54] proposed MMSE solutions in a multiple user scenario.

**TVTP-Based Equalization** Direct channel equalization for single or multiple users has been considered in [74] where higher order statistics were implicitly exploited by minimizing a nonquadratic cost function.

**CRT-Based Equalization** Direct equalization based on CRT can be found in [92]. It is shown that chirp sequences are very useful for the solution of this problem.

### 6.3.7 Data Detection

Data detection can be carried out in a similar way as with conventional (where IT is absent) digital communications systems. The only consideration is that the IT sequence must be removed (the affine precoding model must be reversed) before data detection takes place. It is also possible to avoid the removal of the IT sequence provided we take its presence into account during the detection process. Note that in the case of DDST, the affine precoding model must be reversed by nonlinear operations [31]. As was mentioned before, estimation errors and the nonorthogonality between data and training might result in an incomplete removal of the IT sequence. As a consequence, this leads to extra noise in the data detection process.

### 6.3.8 Final Remarks

In this section we exposed several problems that may arise when one intends to build a working digital communications system. A specific receiver architecture was introduced to facilitate the explanation of the main issues. From the exposition of the different estimation tasks and the review of several approaches based on IT to solve them, we conclude that the field is reaching a mature state, and practical implementation of digital communications systems seems now feasible at least in the single-user case.

## 6.4 IT-BASED ESTIMATION FOR MULTIPLE USERS EXPLOITING ARRAY PROCESSING: CONTINUOUS TRANSMISSION

The previous section dealt with IT-based estimation problems that appear in a single-user scenario. The focus now will be on the multiuser case where an array is used at the receiver. When dealing with a multiuser system, we essentially have the same estimation problems as in the single-user case. However, in the multiuser scenario, the parameter estimation tasks become much more involved. This is because the received signal now depends on a greater number of parameters. Therefore, the dimensionality of the problem increases and so does the complexity to compute the estimates. We will consider two possible situations. In this section, the users transmit information in a continuous fashion. In the next section, they send information in the form of packets.

### 6.4.1 Source Separation in Flat Channels

Let us have a closer look at the source separation problem for flat channels. This problem has received considerable attention. Relevant references are [107–118]. In this section we assume that the users transmit their signals in a continuous fashion,

in other words, the signals are present on the channel during relatively long time intervals. A simplified two-user model of the system will be used first. This has the purpose of easing the notation and also simplifies the problem, such that a better initial understanding can be obtained. The next section then relaxes the frequency flat channel condition and treats the general multiuser case.

Let us assume that the transmitter operates according to the TVTP principle and see how TVTP can be exploited at the receiver to separate the two users using the antenna array. The situation is depicted in Figure 6.4, where the information sequences for the users are represented by  $b_1(k)$  and  $b_2(k)$  and each user possesses its own modulating sequence  $a_i(k)$  ( $i = 1, 2$ ). The transmitted sequences are denoted by  $s_1(k)$  and  $s_2(k)$ . The receiver uses an antenna array composed of  $K$  elements and, as interference cancellation structures, two spatial equalizers, one for each user. The coefficients of the spatial equalizers are calculated using two distinct adaptive control units.

The received signal vector  $\mathbf{x}$  at the output of the antenna array can be expressed as

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{n}, \quad (6.17)$$

where  $\mathbf{H}$  is a mixing matrix, vector  $\mathbf{x}$  is given by  $\mathbf{x} = [x_1(k), x_2(k), \dots, x_K(k)]^T$  with  $x_i(k)$  representing the noisy output from antenna  $i$ , vector  $\mathbf{s}$  is given by  $\mathbf{s} = [s_1(k), s_2(k)]^T$ , and the noise vector is  $\mathbf{n} = [n_1(k), n_2(k), \dots, n_K(k)]^T$ , with entry  $n_i(k)$  equal to the noise contribution at antenna  $i$ . So let the mixing matrix be

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & h_{2,1} \\ h_{1,2} & h_{2,2} \\ \vdots & \vdots \\ h_{1,K} & h_{2,K} \end{bmatrix}. \quad (6.18)$$

The first column of  $\mathbf{H}$  is given by the sum of the antenna array steering vectors for every path of user 1, where the second column is similarly defined but now for user 2. The notation  $h_{i,j}$  represents the channel response from user  $i$  to receiving antenna  $j$ . It is clear that vector  $\mathbf{x}$  is a mixture of both transmitted signals. Using spatial equalization

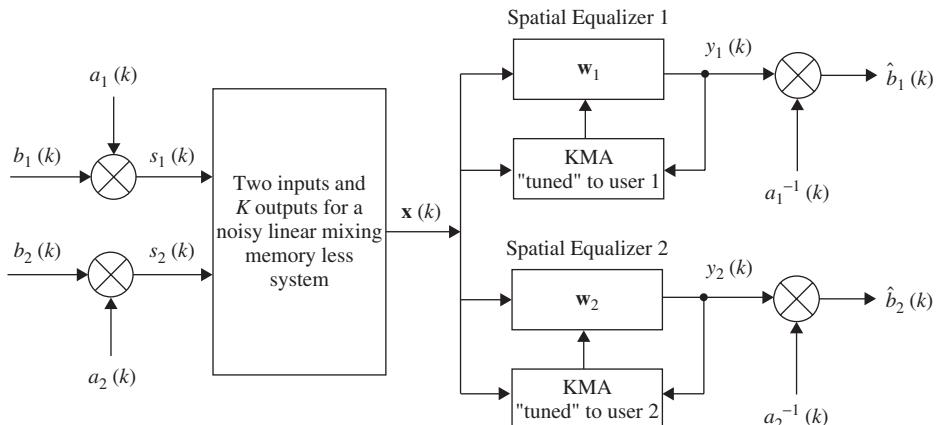


Figure 6.4 Separation of two users based on TVTP-KMA.

it is possible to separate the users. The output of the equalizer for user  $i$  ( $i = 1, 2$ ) can be expressed by

$$y_i(k) = \mathbf{w}_i^T \mathbf{x}, \quad (6.19)$$

where  $\mathbf{w}_i = [w_i(1), w_i(2), \dots, w_i(K)]^T$  is the spatial equalizer coefficient vector for user  $i$  and  $y_i(k)$  is its output. It is desired that  $y_i(k) = s_i(k)$ . In order to achieve this condition the knowledge of the expected value of the instantaneous power of  $s_i(k)$  will be exploited. There are potentially many ways to take advantage of this property. One possibility is to use the known modulus algorithm (KMA), a method first introduced in [119] as a generalization of the well-known constant modulus algorithm (CMA) [120, 121]. Surprisingly this method does not appear to be widely used despite its origins in the popular constant modulus algorithm. The following material is based on the work presented in [73] (see also [3]).

Let us give an empirical explanation as to why we may wish to use the KMA for the scheme at hand. Since the transmitted signal possesses known power variations, the use of the KMA, which shares many properties with the versatile CMA [120, 121], suggests itself as a good candidate. Also, the fact that different users possess different modulating sequences implies that they possess a different power variation pattern. This simple consideration brings us to the conclusion that if the KMA is “tuned” to a particular user, it will try to recover this user and not the interferer. This is so because the cost associated with the extraction of the desired user will be less than either the cost of recovering the interferer or the cost of recovering a mixture of desired user and interferer. Before presenting the TVTP-KMA solution to the current problem, it is of interest to review how the well-studied CMA has been applied to the problem of interference mitigation in antenna array receivers. In this way, we will be in a position to explain why KMA-TVTP has key advantages as compared to other works based on the use of the CMA.<sup>1</sup>

The term CMA is coined in [121]. However, a complete family of algorithms with the CMA as a special case is previously proposed in [120]. This last work employs the CMA to achieve intersymbol interference (ISI) cancellation. From then on, the algorithm has been successfully applied in diverse areas of signal processing. In the context of signal separation using antenna arrays, it has been recognized that the CMA exhibits robustness against imperfect arrays and coherent multipath environments, two disruptions that preclude the use of super resolution techniques like [122] and [123] for direction of arrival estimation. The study of blind beamformers based on the CMA, or variations, can be traced back to [124]. This work studies the capture performance of the CMA when used with constant modulus signals. Later, [125] analyzes the CMA cost function and finds that the CMA algorithm works in an optimal fashion for nonconstant modulus signals as well, provided that their normalized kurtosis<sup>2</sup> is less than two. Some time later, [126] and [127] find the conditions on the number of sensors necessary to cancel intersymbol interference and jammers using spatial or spatiotemporal filters, with emphasis on the CMA capabilities.

From the work in [125], it is known that the CMA recovers one signal and cancels the others. However, it is not known which signal will be recovered. To overcome

<sup>1</sup>The CMA itself possesses several advantages when compared to other procedures, including low computational complexity and insensitivity to carrier frequency offset.

<sup>2</sup>The normalized kurtosis  $\kappa_{b_i(k)}$  of a random source is defined as  $\kappa_{b_i(k)} = \frac{E\{|b_i(k)|^4\}}{\left[E\{|b_i(k)|^2\}\right]^2}$  [134] (page 1944).

this limitation, a multistage canceller based on the CMA (MSC-CMA) is proposed in [128] and later analyzed in [129, 130]. The idea here is to apply the CMA beamformer and extract one signal. Using this extracted signal, it is possible to employ an adaptive canceller to remove it from the mixture. A second CMA is then applied to the remaining mixture and the process repeated. Proceeding this way, it is possible to recover all the signals. In [131] and [132] a different approach is taken and it is called the multiuser CMA (MU-CMA). They propose the joint detection of all the signals simultaneously. Their objective function is a combination of CMA plus an additional term that penalizes the cross correlation between different beamformer outputs. The characteristics of this objective function are analyzed in [131] for the ISI-free case but no global convergence proof is available for the general case. Some results have been presented in [133].

One inherent limitation of the approaches based on the CMA reviewed before (MSC-CMA and MU-CMA) is that it is not known which signal appears in which beamformer. It is obvious that if the number of users is high the uncertainty of distinguishing every desired user increases, resulting in a considerable growth in the complexity of user ordering. The identification of the users needs to be done after the data has been recovered, and thus every user will have to transmit its own identification signature. Moreover, a complication appears when users enter or leave the system. This creates a new condition that brings the possibility of rendering the previous ordering useless because with a new interferer all the beamformers will have to readapt, and this could potentially result in a permutation of users. Also, both CMA-based schemes described before are useful for a base station but not for a stand-alone user who has no interest in the recovery of other users (and perhaps does not have the computational power to do so).

Using TVTP together with KMA permits the recovery of each signal separately in a decentralized manner, therefore avoiding the complications that arise when employing a centralized architecture. Moreover, there is no problem in identifying which signal appears in which beamformer or equalizer because each control unit is “tuned” to a specific user, thus avoiding the postequalization ordering needed for the other approaches. Lastly, it is anticipated that when users enter or leave the system, the adaptation process will be less affected since after the initial loss of signal, the beamformer will readapt to recover the desired user again.

The operation of the TVTP-KMA is as follows. The antenna array vector output is applied to the two spatial equalizers with weight vectors  $\mathbf{w}_1$  and  $\mathbf{w}_2$ . The two equalizers’ outputs are then multiplied by the sequences  $a_1^{-1}(k)$  and  $a_2^{-1}(k)$ , which are no more than the elementwise multiplicative inverse of the transmitted modulating sequences. The resulting signals  $\hat{b}_1(k)$  and  $\hat{b}_2(k)$  will then be an estimate of the transmitted information data, provided the method works. The KMA adjusts the equalizer  $\mathbf{w}_i$  in order to minimize the following cost function:

$$J_i = \left\langle E \left\{ \left[ |y_i(k)|^2 - a_i^2(k) \right]^2 \right\} \right\rangle, \quad (6.20)$$

where  $\langle \cdot \rangle$  stands for time averaging. Using a stochastic gradient approach to minimize (6.20), the update equation (at iteration  $k$ ) for the equalization vector  $\mathbf{w}_i$  is simply

$$\mathbf{w}_i^k = \mathbf{w}_i^{k-1} - \mu \mathbf{x}^* y_i(k) (|y_i(k)|^2 - a_i^2(k)). \quad (6.21)$$

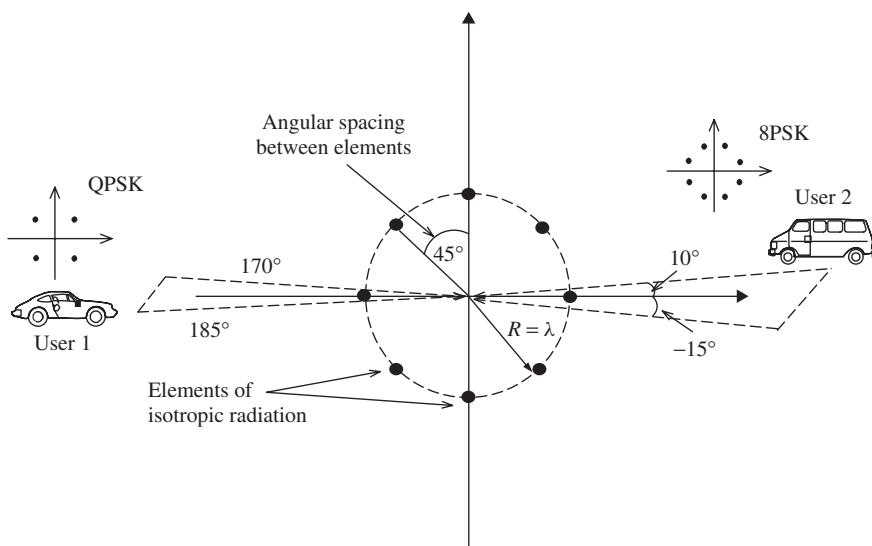
It is shown in [73] that despite the nonquadratic nature of the cost function in (6.20), it does not possess undesired local minima, provided the modulating sequences satisfy

certain sufficient conditions. In other words, the cost function could be molded (see [73] for a three-dimensional graph of the molded cost function) to possess only one minimum, which is of course global, by proper design of the modulating sequences (actually the global minima is not a point, but a family of points that all account for the inherent phase ambiguity of the KMA). This way, convergence of (6.21) to the desired solution will occur irrespective of the initialization point (provided, of course, that the step size factor is adequately chosen).

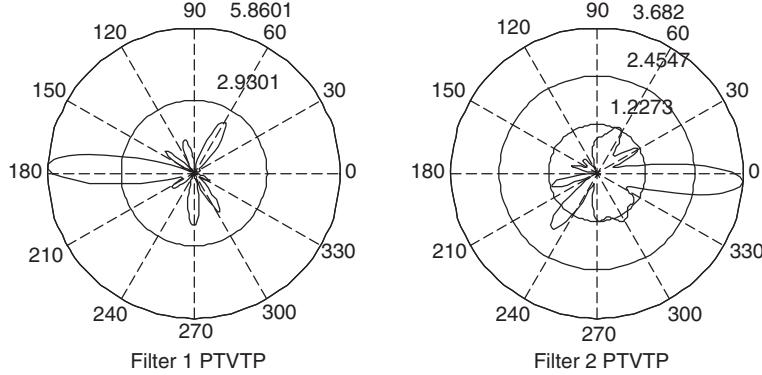
An illustrative example is now presented. A circular array composed of eight elements is employed at the receiver. The radius of the array has been chosen to be equal to one wavelength. The positions of the array elements are as shown in Figure 6.5 (each element is assumed to radiate in an isotropic manner). Each user reaches the receiver via two equal strength paths with very small difference between their propagation delays relative to the symbol period (frequency flat channel). The path's directions of arrival are  $185^\circ$  and  $170^\circ$  for the first user and are  $10^\circ$  and  $-15^\circ$  for the second user with reference to the array elements as shown in the same figure. The first transmitter is using QPSK with an input signal-to-noise ratio (SNR) of  $7.5 \text{ dB}$ . The second transmitter employs 8-PSK with an input SNR =  $17 \text{ dB}$ . The periodic modulating sequences are defined by the following (unique) single periods:  $\{a_1(k)\}_{k=0}^1 = \{1.3, 0.55\}$  and  $\{a_2(k)\}_{k=0}^2 = \{1.4, 0.87, 0.54\}$ . The step size is  $\mu = 5 \times 10^{-3}$ . The synthesized beam patterns after 1000 iterations of the algorithm in (6.21) are shown in Figure 6.6. Note that the responses are such that a significant gain is placed toward the direction of the desired signal and a null toward the interferer.

#### 6.4.2 Source Separation in Frequency-Selective Channels

Frequency-selective channels and multiple users are treated in this section. The objective is now to both equalize and separate the signals. This problem is referred in the literature as the convulsive or dynamic mixture case, and it has been addressed, for



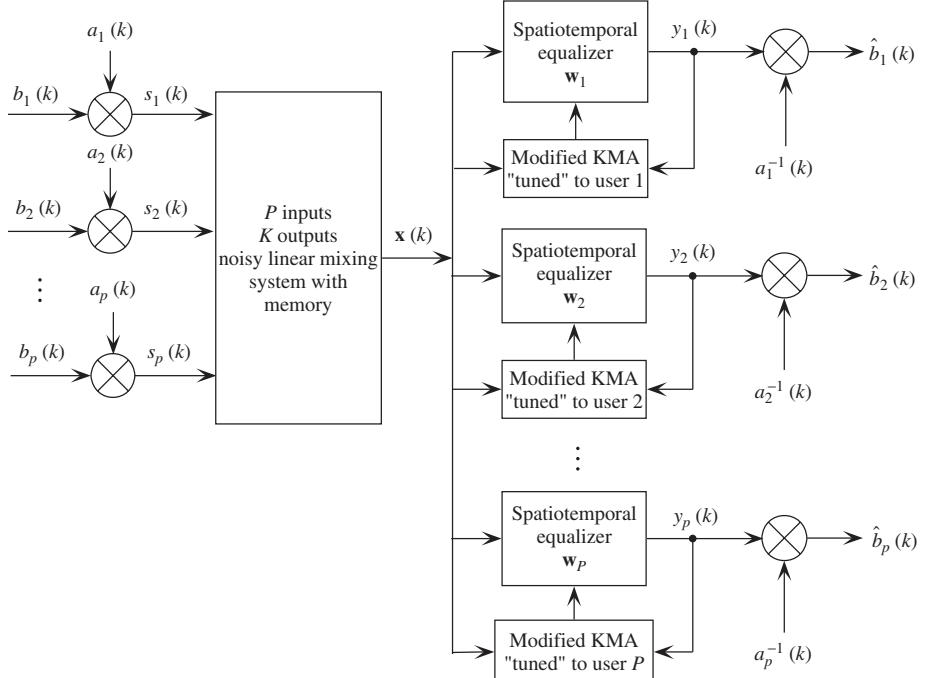
**Figure 6.5** The setup for the example of using the TVTP-KMA for source separation.



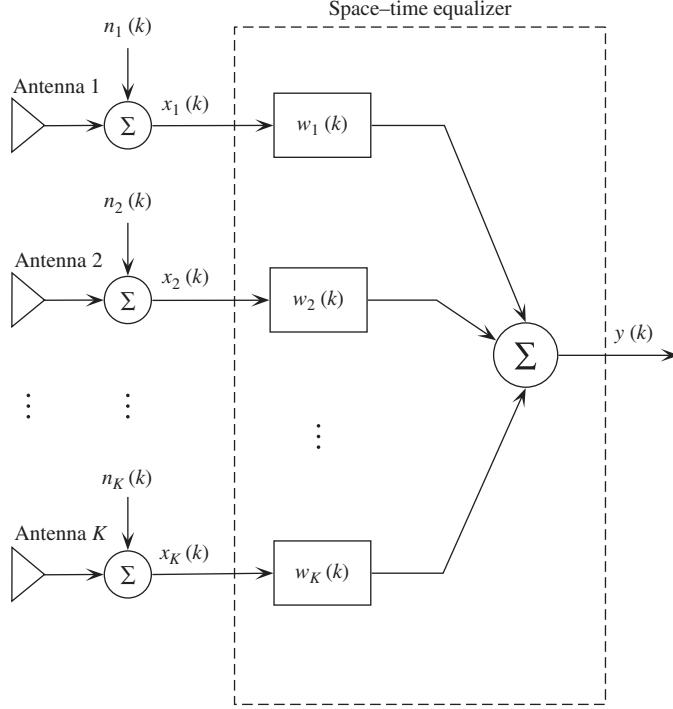
**Figure 6.6** Resulting beampatterns for the source separation setup in Figure 6.5.

example, in [135–139]. The use of TVTP-KMA for this scenario is now discussed with the aid of Figure 6.7. There are  $P$  active users, each one sending data  $b_i(k)$  ( $k = 1, 2, \dots, P$ ) and possessing their own modulating sequence  $a_i(k)$ . The actual transmitted signals  $s_i(k)$  form the input to a MIMO noisy linear system with memory. Each of the system outputs is in general a linear mixture of ISI and multiple-access interference (MAI) coming from all the  $P$  users.

The receiver structure is composed of a bank of spatiotemporal equalizers (like the one presented in Fig. 6.8) driven by a modified KMA to be described next. The



**Figure 6.7** Separation problem (using TVTP-KMA) of  $P$  users with ISI.



**Figure 6.8** A linear space-time equalizer.

method is similar to that presented in the previous section and has been reported in [74]. However, there is one minor modification that allows us to show in a very simple way that the algorithm's cost function is also free of local minima in this general case. The objective function to be considered is now

$$J_i = \left\langle E \left\{ \left[ \left| \frac{y_i(k)}{a_i(k)} \right|^2 - 1 \right]^2 \right\} \right\rangle, \quad (6.22)$$

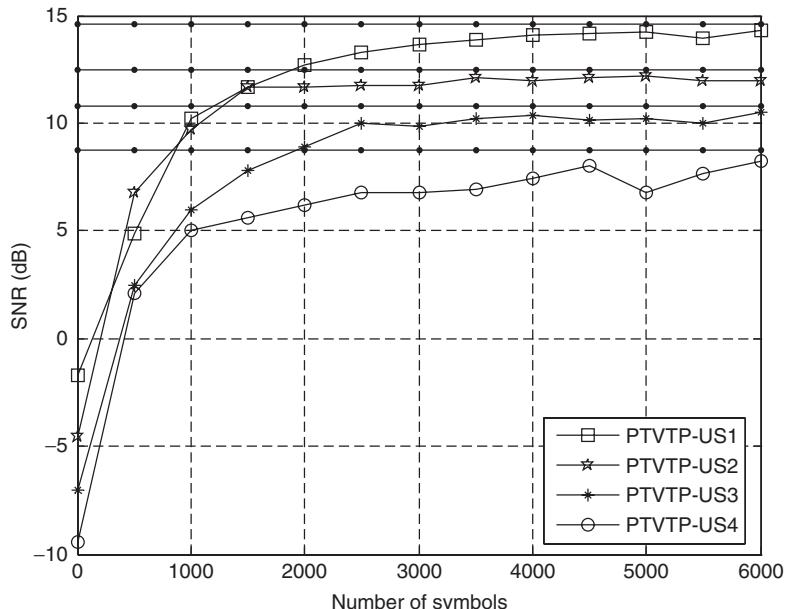
instead of (6.20). The difference, though small, is crucial for finding (in a simple way) the conditions for no local minima to exist. Equation (6.22) will be called the modified KMA cost function. As was said before, the work in [125] shows that in a general scenario with  $P$  transmitting users, the CMA has local minima that correspond to the extraction of any one user (or its delayed replicas) while canceling all the interference. This is true provided that the normalized kurtosis of the signal to be recovered is less than two and there is enough diversity (necessary to achieve the separation of the sources via array processing) in the channel characteristics. The inability of the CMA to recover signals whose kurtosis is bigger than two could be considered a weakness or a limiting factor of the CMA in the sense that it cannot be applied to blindly separate data possessing arbitrary probability density functions. However, this “weakness” is in fact the unique strength that the CMA possesses, and that the modified TVTP-KMA method exploits, to perform global separation of users in the scenario at hand. *The key point is then to make the normalized kurtosis of the interference “appear” bigger than*

two while the kurtosis of the desired signal remains lower than two, which fortunately can be achieved using the TVTP concept. The interested reader is referred to [74] or [3] for a more in-depth explanation of the method.

A simulation is now presented to exemplify the performance of the method, where we have used the same setup as [131]. A five-element uniform linear antenna array with interelement spacing of half wavelength is used at the receiver. Four signals are impinging on the array that belong to four transmitters all using QPSK modulation. The periodic modulating sequences are  $\{a_1(k)\}_{k=0}^1 = \{1.3, 0.55\}$ ,  $\{a_2(k)\}_{k=0}^2 = \{1.4, 0.87, 0.54\}$ ,  $\{a_3(k)\}_{k=0}^4 = \{1.14, 1.48, 0.67, 0.52, 0.87\}$ , and  $\{a_4(k)\}_{k=0}^6 = \{1.1, 1.52, 0.6, 1.2, 0.55, 0.55, 1\}$ . The directions of arrival of the signals with respect to the broadside of the array are  $-50^\circ$ ,  $-20^\circ$ ,  $20^\circ$ , and  $60^\circ$ . The input SNRs are 2, 4, 6, and 8 dB. The step-size parameter is  $\mu = 1 \times 10^{-3}$  for all the beamformers. The receiver is using four spatial filters arbitrarily initialized as  $\mathbf{w}_1 = [1, 0, 0, 0, 0]^T$ ,  $\mathbf{w}_2 = [0, 1, 0, 0, 0]^T$ ,  $\mathbf{w}_3 = [0, 0, 1, 0, 0]^T$ , and  $\mathbf{w}_4 = [0, 0, 0, 1, 0]^T$ . Figure 6.9 shows the SINR evolution for all the users. The optimum MMSE (computed based on the true channels and SNRs) achievable is also drawn in the graphs. Note that the propagation conditions are very severe because the input SNRs are very low for all users, and the initial SINRs are all below 0 dB. Thus, both noise and interference are very strong. Nevertheless, the algorithm does perform satisfactorily and its performance approaches that of the MMSE solution represented by the solid lines with dotted marks in the figure.

#### 6.4.3 Final Remarks

We have considered in this section the application of the TVTP variant of IT to perform source separation based on array processing for continuous transmissions. Note that it



**Figure 6.9** SINR evolution curves for the example of source separation of P users with ISI.

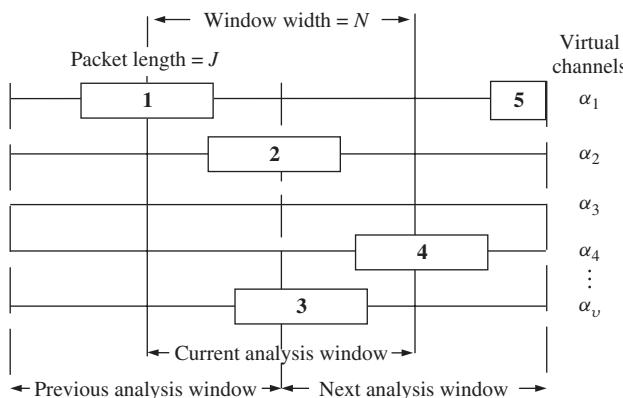
is possible to use the other IT variants (ST and CRT) to perform this task, as has been mentioned in the previous sections. However, due to the limited size of the present chapter, we are not able to give examples for the other variants. The interested reader is directly referred to the original works on the subject.

## 6.5 IT-BASED ESTIMATION FOR MULTIPLE USERS EXPLOITING ARRAY PROCESSING: PACKET TRANSMISSION

In this section we will discuss the application of IT array processing to one of the very promising areas in wireless communications: multiple packet reception in random-access networks. A key issue in these types of networks is the decrease of throughput due to collisions arising from uncoordinated transmitters. It has been typically considered that a collision destroys all the colliding packets, making retransmissions necessary and so decreasing network efficiency. Multiple packet reception (MPR), which has its roots in powerful signal processing techniques, is a possible remedy to the collision problem. This is achieved by providing “smartness” to an antenna array in such a way that it is able to separate the multiple colliding packets. We will now provide an introduction to this emerging field with the purpose of motivating research in this challenging and relatively unexplored topic.

### 6.5.1 Multiple Packet Reception

**6.5.1.1 Multipacket Reception Model** The reception model used in this section corresponds to an uncoordinated ad hoc network. We will use Figure 6.10 as an aid to better understand the model. The packets coming from different users arrive asynchronously at the receiver. Flat and frequency-selective multiuser channels are possible. The transmitted packets could have different lengths, although it is assumed in this chapter that the packet lengths are the same and equal to  $J$  symbol periods. The receiver will process the packets in a window-by-window fashion where the width of each window will be set equal to  $N$  symbol periods. Due to the asynchronous assumption for the arrivals, packets can start at arbitrary times with respect to the beginning of



**Figure 6.10** Packet reception model for MPR problem.

the receiver observation window. In fact, packets can even start before the observation window begins or can finish after the observation window ends, as can be clearly seen in Figure 6.10. In this model we will assume that in order to be decoded, a packet must be completely seen in any one of the observation windows. So, if the receiver wants to give each packet the opportunity to be decoded, the size  $N$  of each observation window should be strictly greater than the packet size  $J$ , consecutive observation windows must be overlapped, and the overlapping should be at least one packet length.

Note that if the overlapping between adjacent observation windows is exactly one packet length, a specific packet will be completely observed either in the current, previous, or next window. As an example consider Figure 6.10, where packets 2 and 3 are completely observed in the current window, packet 1 is completely observed in the previous window, and likewise for packet 4 but for the next observation window. On the other hand, if the overlapping between adjacent windows is greater than one packet length, then a specific packet could be completely observed in more than one observation window. Finally, if the overlapping between adjacent observation windows is less than one packet length, then it is possible that a packet could not be completely observed in any observation window and, if this is so, it will be lost.

**6.5.1.2 Signal Processing for Packet Separation** To accomplish packet separation using a space–time linear equalizer as the one shown in Figure 6.8, it is necessary that the following conditions are all met: (a) The packets must possess different spatial signatures. This way, the spatial diversity of the environment can be exploited by the antenna array. (b) The antenna array should have at least one more element compared to the number of arriving signals if the channel is frequency selective or at least the same number if the multiuser channel is flat [3]. (c) If the separation is based on IT, it must in principle be guaranteed that different packets contain dissimilar training sequences that allow their separation.

Not much work has been done to date in this area. The first contribution can be traced back to [82] where the packet separation problem, from the signal processing point of view, is introduced. In this work, the multiuser channel is assumed to be flat and the receiver is assumed to have the knowledge of both the IT sequence and the arrival time of the desired packet. In this case the TVTP form of IT is used. The separation is carried out based on the analytical KMA (AKMA). AKMA is an algebraic closed-form solution variation of the KMA introduced in the previous section. The work has the merit of introducing the topic but it fails short in the following areas. First, the knowledge of the arrival time of the desired packet is a very strong assumption in the ad hoc network context. Second, and similarly to the arrival time, the knowledge of the IT sequence at the receiver is not realistic. Third, although attractive (insensitive to CFO) and robust, the AKMA possesses high computational complexity. As a consequence, although it can be used under frequency-selective channels, the large number of parameters to be estimated certainly precludes its application in this situation. Extensions of the work in [82] that rescind the synchronization requirement are presented in [83, 84] while theoretical bounds on performance are given in [140].

Another approach to tackle the MPR problem in ad hoc networks is proposed in [94] where some of the restricting assumptions present in [82] are relaxed. This time, CRT-IT instead of TVTP-IT is employed at the transmitters. The assumption of the knowledge of the IT sequences of the packets at the receiver is lifted by introducing the concept of a codebook (with  $D$  elements) of IT sequences. This way, although the

receiver does not know exactly which IT sequence is used for a given packet, the codebook is known. Thus, the transmitters can randomly select a code from the codebook and then use its associated IT sequence for transmission. Thereby a codebook creates the so-called virtual channels, as illustrated in Figure 6.10. The receiver searches through the virtual channels for possible transmitted packets that can be recovered if the conditions (a to c) mentioned above are met. Furthermore, the information about the packet arrival time is unnecessary for this method. In fact, packet synchronization can be accomplished after packet separation and so can be obtained by methods already proposed for the single-user case. Also, the algorithm in [94] can be applied to flat as well as to frequency-selective multiuser channels. However, the method is affected by nonzero CFOs. This drawback is inherited from the CRT-IT training structure.

The MPR problem has also been studied under the ST-IT framework. Relevant works in this area are [54, 141, 142]. The first study deals with the reception model presented before (as well as the other works above). The last two studies consider OFDM transmitters and a different reception model where packets are asynchronous but larger than the observation window.

**6.5.1.3 Throughput versus Complexity Trade-Offs** It is intuitively clear that MPR increases the throughput of the network due to the reduction of collisions. A quantitative study of throughput enhancement is presented in [94]. In the reception model considered here, the throughput depends on the length of the observation windows, on the amount of overlap between windows, on the size of the codebook, and on the separation capability of the array (related to the number of antennas). The formula derived in [94] applies for all observation window lengths (greater than the packet length), codebook sizes (greater than or equal to the number of antennas in the array), and number of antennas in the array. However, it is only valid for the case when the overlap between windows is equal to the size  $J$  of the packets. Some unpublished results (by the authors of this chapter) are available for certain overlap values but the generalization has so far proven to be elusive. Nevertheless, it has been confirmed by simulations that provided the other parameters remain the same, the more the overlap between windows the greater the throughput. However, augmenting the overlap raises the computational burden necessary to process the received signal, as the number of analysis windows per time unit grows. In the limit, when the overlap equals the observation window length, the computational load goes to infinity. This is because adjacent observation windows entirely overlap and so the number of observation windows that are necessary to cover a given time interval is unbounded.

## 6.6 OPEN RESEARCH PROBLEMS

As we have seen in this chapter a good deal of work has been done on IT for single-user links and much less for the general multiuser setup. However, in both cases, some problems remain open. In this section, some possible research directions and open problems will be briefly commented upon.

### 6.6.1 Practical Implementations

The time is right to start the construction of prototypes that incorporate the IT paradigm. Initial attempts that resulted in a working system based on a Texas Instruments DSP are

reported in [143] for the IT-ST-based single-user case. However, this area is wide open to new developments. The multiuser problem still has a lot of theoretical difficulties and is perhaps not yet ready for practical implementations.

### 6.6.2 Parameter Estimation Problems

Although many different approaches exist to obtain the estimation of the necessary parameters based on at least one form of IT, several problems are not yet addressed. Perhaps the most important open problems are those that appear when the link parameters vary with time. Recent results have been presented for time-varying channel estimation but other tracking problems like clock and carrier recovery have not yet been reconsidered under the IT framework.

### 6.6.3 Combination of IT Strategies

So far, publications that use IT have focused on one of the three forms that were presented in Section 6.3. As mentioned there, each form has its own strengths and weaknesses, and perhaps an intelligent combination of several forms could be beneficial for some problems.

### 6.6.4 Multipacket Reception Challenge

From the signal processing viewpoint, multipacket reception could be a tough nut to crack. The asynchronous ad hoc networking model introduced in the previous section imposes considerable technical difficulties. Due to the decentralized and asynchronous nature of the network, the receivers have a very limited knowledge (and control) of the environment as compared to a centrally controlled topology. In particular, the following problems appear. The number of packets in a given observation window is not known beforehand, which for example precludes the use of multiuser detection techniques [144] so attractive for a centrally controlled system. Also, the ignorance of the model orders of the different propagation channels has been a hurdle (even for the single-user case) at least for some of the blind approaches that were proposed in the past, and it is expected that the IT paradigm triumphs over this limitation. In addition, it is not clear how ET approaches should be applied in this multiuser asynchronous scenario since more users imply more training and consequently a higher bandwidth loss. This provokes diminishing returns because in a multiuser channel the need for making efficient use of the bandwidth is even greater. Moreover, ET design of space–time equalizers requires the knowledge of the CFO. In turn, the acquisition of the desired packet CFO may not be easy to accomplish in the multiuser context, especially when power imbalance among packets exist. So we have a chicken and egg type of problem here, *even when ET is used*.

The design of training sequences possessing good cross correlation for multiple shifts is also another issue due to the asynchronous nature of transmissions. The dc offsets at different antennas will not in general be equal and so far work on MPR has not considered this possibility. In the case of OFDM networks, frequency-domain equalization of asynchronous (at the cyclic prefix level) packets is complicated [145]. Another problem is that the conventional approach to automatic gain control could fail in this environment due to asynchronies in the arrival times of the packets. Finally, if the packets have a large number of symbols, time-varying conditions are likely to occur.

This situation has not yet been addressed by any of the previously mentioned references. The problem is enormous in this case due to the need for tracking the time-varying channels, the carrier frequencies, and the data clock periods of all the packets involved in the collision. The authors expect that the material put forward in this chapter will encourage further research on the topics discussed.

## ACKNOWLEDGMENTS

The authors are grateful for the support given by INTEL under research grant INTEL DCIT-2006 for the study of implicitly trained digital communications systems. Thanks are also due to Texas Instruments for their continuous support for the strengthening of the signal processing laboratories at CINVESTAV-IPN that have made possible the first practical implementation of an IT-based digital communications system.

## REFERENCES

1. J. R. Barry, A. Kavcic, S. W. McLaughlin, A. Nayak, and W. Zeng, "Iterative timing recovery," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 89–102, Jan. 2004.
2. R. Koetter, A. C. Singer, and M. Tuchler, "Turbo equalization," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 67–80, Jan. 2004.
3. A. G. Orozco-Lugo, "Blind spatio-temporal processing for communications," PhD thesis, University of Leeds, 2000.
4. A. J. Paulraj and C. B. Papadias, "Space-time processing for wireless communications," *IEEE Signal Process. Mag.*, vol. 14, no. 5, pp. 49–83, Nov. 1997.
5. M. T. Ma, *Theory and Application of Antenna Arrays*, New York: Wiley, 1974.
6. S. Haykin (Ed.), *Blind Deconvolution*, Englewood Cliffs, NJ: Prentice Hall, 1994.
7. S. Haykin (Ed.), *Unsupervised Adaptive Filtering: Blind Source Separation*, Vol. I, New York: Wiley, 2000.
8. S. Haykin (Ed.), *Unsupervised Adaptive Filtering: Blind Source Separation*, Vol. II, New York: Wiley, 2000.
9. Z. Ding and Y. G. Li, *Blind Equalization and Identification*, New York: Marcel Dekker, 2001.
10. G. B. Giannakis, Y. Hua, P. Stoica, and L. Tong, *Signal Processing Advances in Wireless and Mobile Communications: Trends in Channel Estimation and Equalization*, Vol. 1, Englewood Cliffs, NJ: Prentice Hall, 2001.
11. G. B. Giannakis, Y. Hua, P. Stoica, and L. Tong, *Signal Processing Advances in Wireless and Mobile Communications: Trends in Channel Estimation and Equalization*, Vol. 2, Englewood Cliffs, NJ: Prentice Hall, 2001.
12. J. K. Cavers, "An analysis of pilot symbol assisted modulation for Rayleigh fading channels," *IEEE Trans. Vehic. Technol.*, vol. 40, no. 4, pp. 686–693, Nov. 1991.
13. L. Tong, B. M. Sadler, and M. Dong, "Pilot assisted wireless transmissions: General model, design criteria, and signal processing," *IEEE Signal Process. Mag.*, vol. 21, no. 6, pp. 12–25, Nov. 2004.
14. U. Mengali and A. N. D'Andrea, *Synchronization Techniques for Digital Receivers*, New York: Plenum, 1997.
15. H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation and Signal Processing*, New York: Wiley, 1998.
16. A. G. Orozco-Lugo, M. M. Lara, and D. C. McLernon, "Channel estimation using implicit training," *IEEE Trans. Signal Process.*, vol. 52, no. 1, pp. 240–254, Jan. 2004.

17. J. H. Manton, I. Y. Mareels, and Y. Hua, "Affine precoders for reliable communications," in *IEEE ICASSP*, Istanbul, Turkey, June 2000, pp. 2749–2752.
18. A. Vosoughi and A. Scaglione, "Everything you always wanted to know about training: Guidelines derived using the affine precoding framework and the CRB," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 940–954, Mar. 2006.
19. D. Makrakis and K. Feher, "A novel pilot insertion-extraction method based on spread spectrum techniques," in *Miami Technicon*, Miami, FL, Oct. 1987.
20. T. P. Holden and K. Feher, "A spread-spectrum based synchronization technique for digital broadcast systems," *IEEE Trans. Broadcast.*, vol. 36, no. 3, pp. 185–194, Sept. 1990.
21. B. Farhang-Boroujeny, "Pilot-based channel identification: Proposal for semi-blind identification of comm. channels," *IEE Electron. Lett.*, vol. 31, no. 13, pp. 1044–1046, June 1995.
22. F. Mazzenga, "Channel estimation and equalization for M-QAM transmission with a hidden pilot sequence," *IEEE Trans. Broadcasting*, vol. 46, no. 6, pp. 170–176, June 2000.
23. G. T. Zhou, M. Viberg, and T. McKelvey, "Superimposed periodic pilots for blind channel estimation," in *Asilomar CSSC*, Vol. 1, Pacific Grove, CA, Nov. 2001, pp. 653–657.
24. J. K. Tugnait and W. Luo, "On channel estimation using superimposed training and first-order statistics," in *IEEE ICASSP*, Vol. IV, Hong Kong, China, Apr. 2003, pp. 624–627.
25. J. K. Tugnait and W. Luo, "On channel estimation using superimposed training and first order statistics," *IEEE Commun. Lett.*, vol. 7, no. 9, pp. 413–415, Sept. 2003.
26. D. C. McLernon, A. G. Orozco-Lugo, and M. M. Lara, "On the structural equivalence of two recent algorithms for implicitly trained channel estimation," in *IEEE ISSPIT*, Rome, Italy, Dec. 2004, pp. 132–135.
27. J. K. Tugnait and X. Meng, "Synchronization of superimposed training for channel estimation," in *IEEE ICASSP*, Vol IV, Montreal, Canada, May 2004, pp. 853–856.
28. E. Alameda-Hernandez, D. C. McLernon, A. G. Orozco-Lugo, M. Lara, and M. Ghogho, "Synchronization for superimposed training based channel estimation," *IEE Electron. Lett.*, vol. 41, no. 9, pp. 565–567, Apr. 2005.
29. E. Alameda-Hernandez, D. C. McLernon, A. G. Orozco-Lugo, M. Lara, and M. Ghogho, "Synchronization and DC-offset estimation for channel estimation using data-dependent superimposed training," *EUSIPCO*, Antalya, Turkey, Sept. 2005.
30. E. Alameda-Hernandez, D. C. McLernon, A. G. Orozco-Lugo, M. Lara, and M. Ghogho, "Frame/training sequence synchronization and DC-offset removal for (data-dependent) superimposed training based channel estimation," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2557–2569, June 2007.
31. M. Ghogho, D. McLernon, E. Alameda-Hernandez, and A. Swami, "Channel estimation and symbol detection for block transmission using data-dependent superimposed training," *IEEE Signal Process. Lett.*, vol. 12, no. 3, pp. 226–229, Mar. 2005.
32. D. C. McLernon, E. Alameda-Hernandez, A. G. Orozco-Lugo, and M. M. Lara, "Performance of data-dependent superimposed training without cyclic prefix," *IEE Electron. Lett.*, vol. 42, no. 10, pp. 604–606, May 2006.
33. T. Whitworth, M. Ghogho, and D. C. McLernon, "Data identifiability for data-dependent superimposed training," in *IEEE ICC*, Glasgow, Scotland, June 2007, pp. 2545–2550.
34. X. Meng and J. K. Tugnait, "Semi-blind channel estimation and detection using superimposed training," in *IEEE ICASSP*, Vol. IV, Montreal, Canada, May 2004, pp. 417–420.
35. S. M. Moosvi, D. C. McLernon, E. Alameda-Hernandez, A. G. Orozco-Lugo, and M. M. Lara, "A low complexity iterative channel estimation and equalisation scheme for (data dependent) superimposed training," in *EUSIPCO*, Florence, Italy, Sept. 2006.

36. O. Longoria-Gandara, R. Parra-Michel, M. Bazdresch, and A. G. Orozco-Lugo, "Iterative mean removal superimposed training for frequency selective channel estimation," accepted at *WiCom*, Dalian, China, Oct. 2008, pp. 1–5.
37. J. P. Nair, R. Kumar, and V. Ratnam, "An iterative channel estimation method using superimposed training for IEEE 802.16e based OFDM systems," in *IEEE ISCE*, Algarve, Portugal, Apr. 2008, pp. 1–4.
38. L. Chan-Tong, D. D. Falconer, and F. Danilo-Lemoine, "Iterative frequency domain channel estimation for DFT-precoded OFDM systems using in-band pilots," *IEEE J. Select. Areas Commun.*, vol. 26, no. 2, pp. 348–358, Feb. 2008.
39. A. Varma, L. Andrew, C. Athaudage, and J. Manton, "Iterative algorithms for channel identification using superimposed pilots," in *AusCTW*, Brisbane, Australia, Feb. 2005, pp. 195–201.
40. R. Carrasco-Alvarez, A. G. Orozco-Lugo, R. Parra-Michel, and J. K. Tugnait, "Enhanced channel estimation using superimposed training based on universal bases expansion," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1217–1222, March. 2009.
41. J. K. Tugnait and X. Meng, "On superimposed training for channel estimation: Performance analysis, training power allocation, and frame synchronization," *IEEE Trans. Signal Process.*, vol. 54, no. 2, pp. 752–765, Feb. 2006.
42. S.-Y. Jung and D.-J. Park, "Linear MMSE receiver using hidden training sequence in DS/CDMA," *Electron. Lett.*, vol. 39, no. 9, pp. 742–744, May 2003.
43. N. Chen and G. T. Zhou, "Superimposed training for OFDM: A peak-to-average power ratio analysis," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2277–2286, June 2006.
44. R. Dinis, N. Souto, J. Silva, A. Kumar, and A. Correia, "Joint detection and channel estimation for OFDM signals with implicit pilots," in *IEEE MWCS*, Budapest, Hungary, July 2007, pp. 1–5.
45. R. Dinis, N. Souto, J. Silva, A. Kumar, and A. Correia, "On the use of implicit pilots for channel estimation with OFDM modulations," in *IEEE VTC Fall*, Baltimore, MD, Sept.–Oct. 2007, pp. 1077–1081.
46. J. P. Nair and R. V. Raja-Kumar, "Channel estimation and equalization based on implicit training in OFDM systems," in *IEEE IFIP ICWOCN*, Bangalore, India, Apr. 2006.
47. D. Xu and L. Yang, "Channel estimation for OFDM systems using superimposed training," in *IEEE IFIP ICCAI*, Bishkek, Kyrgyzstan, Sept. 2005.
48. X. Luo and G. B. Giannakis, "Low complexity blind synchronization and demodulation for (ultra-) wideband multi-user ad Hoc access," *IEEE Trans. Wireless Commun.*, vol. 5, no. 7, pp. 1930–1941, July 2006.
49. J. Wang and X. Wang, "Superimposed training-based noncoherent MIMO systems," *IEEE Trans. Commun.*, vol. 54, no. 7, pp. 1267–1276, July 2006.
50. M. Ghogho, D. McLernon, E. Alameda-Hernandez, and A. Swami, "SISO and MIMO channel estimation and symbol detection using data-dependent superimposed training," in *IEEE ICASSP*, Vol III, Philadelphia, PA, Mar. 2005, pp. 461–464.
51. M. Qaisrani and S. Lambotharan, "Estimation of doubly selective MIMO channels using superimposed training and turbo equalization," in *IEEE VTC*, Marina Bay, Singapore, May 2008, pp. 1316–1319.
52. W. Yuan and P. Fan, "Implicit MIMO channel estimation without DC-offset based on ZCZ training sequences," *IEEE Signal Process. Lett.*, vol. 13, no. 9, pp. 521–524, Sept. 2006.
53. H. Zhu, B. Farhang-Boroujeny, and Ch. Schlegel, "Pilot embedding for joint channel estimation and data detection in MIMO communication systems," *IEEE Commun. Lett.*, vol. 7, no. 1, pp. 30–32, Jan. 2003.
54. A. G. Orozco-Lugo, G. M. Galvan-Tejada, M. M. Lara, and D. C. McLernon, "A new approach to achieve multiple packet reception for ad hoc networks," in *IEEE ICASSP*, Vol. IV, Montreal, Canada, May 2004, pp. 429–432.

55. H. Shuangchi and J. K. Tugnait, "On doubly selective channel estimation using superimposed training and discrete prolate spheroidal sequences," *IEEE Trans. Signal Process.*, vol. 56, no. 7, part 2, pp. 3214–3228, July 2008.
56. Z. Junruo and Y. Zakharov, "Iterative B-spline estimator using superimposed training in doubly-selective fading channels," in *IEEE Asilomar CSSC*, Pacific Grove, CA, Nov. 2007, pp. 1795–1799.
57. G. T. Zhou and C. Ning, "Superimposed training for doubly selective channel," in *IEEE WSSP*, Saint Louis, MO, Sept.–Oct. 2003, pp. 82–95.
58. J. K. Tugnait and H. Shuangchi, "Doubly-selective channel estimation using data-dependent superimposed training and exponential basis models," *IEEE Trans. Wireless Commun.*, vol. 6, no. 11, pp. 3877–3883, Nov. 2007.
59. Y. Li and L. Yang, "Channel estimation and tracking using implicit training," in *IEEE VTC*, Vol. 1, Los Angeles, CA, Sept. 2004, pp. 72–75.
60. M. Li, W. Zuo, and Z. Liu, "Time-Selective MIMO Channel Estimation Based on Implicit Training," in *IEEE ISISPSCS*, Xiamen, China, Nov.–Dec. 2007, pp. 384–387.
61. M. Ghogho and A. Swami, "Estimation of doubly-selective channels in block transmissions using data-dependent superimposed training," in *EUSIPCO*, Florence, Italy, Sept. 2006.
- 61a. X. Meng and J. K. Tugnait, "Doubly-selective MIMO channel estimation using superimposed training," in *Sensor Array and Multichannel Signal Processing Workshop*, Sitges, Spain, July 2004, pp. 407–411.
62. J. K. Tugnait and S. He, "Direct FIR linear equalization of doubly selective channels based on superimposed training," in *IEEE ICASSP*, Vol. IV, Toulouse, France, May 2006, pp. 589–592.
63. H. Bölcskei, R. W. Heath, Jr., and A. J. Paulraj, "Blind channel identification and equalization in OFDM-based multiantenna systems," *IEEE Trans. Signal Process.*, vol. 50, no. 1, pp. 96–108, Jan. 2002.
64. A. Chevreuil and Ph. Loubaton, "Blind-second order identification of FIR channels: Forced cyclostationarity and structured subspace method," *IEEE Signal Process. Lett.*, vol. 4, no. 7, pp. 204–206, July 1997.
65. G. B. Giannakis, "Filterbanks for blind channel identification and equalization," *IEEE Signal Process. Lett.*, vol. 4, no. 6, pp. 184–187, June 1997.
66. D. T. M. Slock, "Blind fractionally spaced equalization based on cyclostationarity and second-order statistics," in *Proc. ATHOS (ESPRIT Basic Research Group 6620) Workshop on System Identif. and Higher Order Statitics*, Sophia-Antipolis, France, Sept. 1993.
67. A. G. Orozco-Lugo and D. C. McLernon, "An application of linear periodically time-varying digital filters to blind equalisation," in *IEE Colloquium on "Digital Filters: An Enabling Technology"*, London, Apr. 1998, pp. 11/1–11/6.
68. E. Serpedin and G. B. Giannakis, "Blind channel identification and equalization with modulation-induced cyclostationarity," *IEEE Trans. Signal Process.*, vol. 46, no. 7, pp. 1930–1943, July 1998.
69. H. Bölcskei, R. W. Heath, Jr. and A. J. Paulraj, "Blind channel estimation in spatial multiplexing systems using nonredundant antenna precoding," in *IEEE Asilomar CSSC*, Vol. 2, Pacific Grove, CA, Oct. 1999, pp. 1127–1132.
70. C. A. Lin and J. Y. Wu, "Blind identification with periodic modulation: A time-domain approach," *IEEE Trans. Signal Process.*, vol. 50, no. 11, pp. 2875–2888, Nov. 2002.
71. J.-Y. Wu and T.-S. Lee, "Periodic-modulation-based blind channel identification for single-carrier block transmission with frequency-domain equalization," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1114–1130, Mar. 2006.

72. C-A. Lin and Y-S. Chen, "Blind identification for MIMO channels using optimal periodic precoding," *IEEE Trans. Circuits Syst.-I: Reg. Papers*, vol. 54, no. 4, pp. 901–911, Apr. 2007.
73. A. G. Orosco-Lugo and D. C. McLernon, "Blind signal separation for SDMA based on periodically time varying modulation," in *IEE CAP*, York, United Kingdom, Mar.–Apr. 1999, pp. 182–186.
74. A. G. Orosco-Lugo and D. C. McLernon, "Blind ISI and MAI cancellation based on periodically time-varying transmitted power," *IEE Electron. Lett.*, vol. 37, no. 15, pp. 984–986, July 2001.
75. W. Hachem, F. Desbouvries, and Ph. Loubaton, "Blind channel estimation for CDMA systems: An induced cyclostationarity approach," in *IEEE ICASSP*, Vol. 5, Istanbul, Turkey, June 2000, pp. 2477–2480.
76. S. Cao and L. Zhang, "Blind channel estimation for CDMA based on modulation-induced cyclostationarity," *International Conference on Communication Technology*, Vol. 2, Beijing, China, Apr. 2003, pp. 1804–1808.
77. T. Li, J. K. Tugnait, and Z. Ding, "Channel estimation of long-code CDMA systems utilizing transmission induced cyclostationarity," in *IEEE ICASSP*, Vol. IV, Hong Kong, China, Apr. 2003, pp. 105–108.
78. E. Serpedin, A. Crevreuil, G. B. Giannakis, and Ph. Loubaton, "Blind channel and carrier frequency offset estimation using periodic modulation precoders," *IEEE Trans. Signal Process.*, vol. 48, no. 8, pp. 2389–2405, Aug. 2000.
79. M. K. Tsatsanis and C. Kweon, "Blind source separation of non-stationary sources using second-order statistics," in *IEEE Asilomar CSSC*, Pacific Grove, CA, Nov. 1998, pp. 1574–1578.
80. R. Zhang and M. K. Tsatsanis, "A second-order method for blind separation of non-stationary sources," in *IEEE ICASSP*, Salt Lake City, UT, May 2001, pp. 2797–2800.
81. Y. Rong, S. A. Vorobyov, A. B. Gershman, and N. D. Sidiropoulos, "Blind spatial signature estimation via time-varying user power loading and parallel factor analysis," *IEEE Trans. Signal Process.*, vol. 53, no. 5, pp. 1697–1710, May 2005.
82. A.-J. Van der Veen and L. Tong, "Packet separation in wireless ad-hoc networks by known modulus algorithms," in *IEEE ICASSP*, Vol. 3, Orlando, FL, May 2002, pp. 2149–2152.
83. R. Djapic and A.-J. Van Der Veen, "Blind synchronization in asynchronous multiuser packet networks using KMA," in *IEEE SPAWC Workshop*, Rome, Italy, June 2003, pp. 165–169.
84. R. Djapic, A.-J. Van Der Veen, and L. Tong, "Synchronization and packet separation in wireless ad hoc networks by known modulus algorithms," *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 51–63, Jan. 2005.
85. W. A. Gardner (Ed.), *Cyclostationarity in Communications and Signal Processing*, New York: IEEE, 1994.
86. B. G. Agee, S. V. Schell, and W. A. Gardner, "Spectral self coherence restoral: A new approach to blind adaptive signal extraction using antenna arrays," *Proc. IEEE*, vol. 78, no. 4, pp. 753–767, Apr. 1990.
87. S.-J. Yu and J.-H. Lee, "Adaptive array beamforming for cyclostationary signals," *IEEE Trans. Antennas Propagat.*, vol. 44, no. 7, pp. 943–953, July 1996.
88. Q. Wu and K. M. Wong, "Blind adaptive beamforming for cyclostationary signals," *IEEE Trans. Signal Process.*, vol. 44, no. 11, pp. 2757–2767, Nov. 1996.
89. J. Cui, D. D. Falconer, and A. U. H. Sheikh, "Blind adaptation of antenna arrays using a simple algorithm based on small frequency offsets," *IEEE Trans. Commun.*, vol. 46, no. 1, pp. 61–70, Jan. 1998.

90. A. Chevreuil and Ph. Loubaton, "MIMO blind second-order equalization method and conjugate cyclostationarity," *IEEE Trans. Signal Process.*, vol. 47, no. 2, pp. 572–578, Feb. 1999.
91. A. G. Orozco-Lugo and D. C. McLernon, "Blind channel equalization using chirp modulating signals," in *IEEE ICASSP*, Vol. 5, Istanbul, Turkey, June 2000, pp. 2721–2724.
92. A. G. Orozco-Lugo and D. C. McLernon, "Blind channel equalization using chirp modulating signals," *IEEE Trans. Signal Process.*, vol. 52, no. 5, pp. 1364–1375, May 2004.
93. G. Leus, P. Vandaele, and M. Moonen, "Deterministic blind modulation-induced source separation for digital wireless communications," *IEEE Trans. Signal Process.*, vol. 49, no. 1, pp. 219–227, Jan. 2001.
94. A. G. Orozco-Lugo, M. M. Lara, D. C. McLernon, and H. J. Muro-Lemus, "Multiple packet reception in wireless ad hoc networks using polynomial phase-modulating sequences," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2093–2110, Aug. 2003.
95. D. M. Bloom, *Linear Algebra and Geometry*, Cambridge: Cambridge University Press, 1979.
96. M. Vu and A. Paulraj, "MIMO wireless linear precoding," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 86–105, Sept. 2007.
97. D. H. Pham and J. H. Manton, "Orthogonal superimposed training on linear precoding: A new affine precoder design," in *IEEE SPAWC*, New York, June 2005, pp. 445–449.
98. S. Ohno and G. B. Giannakis, "Optimal training and redundant precoding for block transmissions with application to wireless OFDM," *IEEE Trans. Commun.*, vol. 50, no. 12, pp. 2113–2123, Dec. 2002.
99. M. Ghogho and A. Swami, "Improved channel estimation using superimposed training," *IEEE SPAWC Workshop*, Lisbon, Portugal, July 2004, pp. 110–114.
100. F. Tufvesson, M. Faulkner, P. Hoeher, and O. Edfors, "OFDM time and frequency synchronization by spread spectrum pilot technique," in *IEEE Miniconference on Communication Theory in Conjunction with ICC*, Vancouver, Canada, June 1999, pp. 115–119.
101. J. E. Kleider, G. Maalouli, S. Gifford, and S. Chuprun, "Preamble and embedded synchronization for RF carrier frequency hopped OFDM," *IEEE J. Select. Areas Commun.*, vol. 23, no. 5, pp. 920–931, May 2005.
102. S. M. A. Moosvi, D. C. McLernon, A. G. Orozco-Lugo, M. M. Lara, and M. Ghogho, "Carrier frequency offset estimation using data-dependent superimposed training," *IEEE Commun. Lett.*, vol. 12, no. 3, pp. 179–181, Mar. 2008.
103. S. M. A. Moosvi, D. C. McLernon, A. G. Orozco-Lugo, and M. M. Lara, "Improved carrier frequency offset estimation using data-dependent superimposed training," in *IEEE ICEEE*, México City, México, Sept. 2007, pp. 126–129.
104. A. G. Orozco-Lugo, M. M. Lara, S. M. A. Moosvi, E. Alameda-Hernández, and D. C. McLernon, "Frequency offset estimation and compensation using superimposed training," in *IEEE ICEEE*, México City, México, Sept. 2007, pp. 118–121.
105. Y. Ma and R. Tafazolli, "Estimation of carrier frequency offset for generalized MC-CDMA systems by exploiting hidden pilots," *IEEE Signal Process. Lett.*, vol. 12, no. 11, pp. 753–756, Nov. 2005.
106. A. Steingass, A. J. Van-Wijngaarden, and W. G. Teich, "Frame synchronization using superimposed sequences," in *IEEE ISIT*, Ulm, Germany, July 1997, p. 489.
107. J. F. Cardoso and A. Souloumiac, "Blind beamforming for non-gaussian signals," *IEE Proc. F*, vol. 140, no. 6, pp. 362–370, Dec. 1993.
108. L. Tong, Y. Inouye, and R. Liu, "Waveform-preserving blind estimation of multiple independent sources," *IEEE Trans. Signal Process.*, vol. 41, no. 7, pp. 2461–2470, July 1993.
109. P. Comon, "Independent component analysis, a new concept?" *Elsevier Signal Process.*, vol. 36, no. 3, pp. 287–314, Mar. 1994.

110. S. Talwar, M. Viberg, and A. Paulraj, "Blind estimation of multiple co-channel digital signals using an antenna array," *IEEE Signal Process. Lett.*, vol. 1, no. 2, pp. 29–31, Feb. 1994.
111. H. Liu and G. Xu, "Blind estimation of array responses for an asynchronous multiuser system," in *IEEE VTC*, Vol. 2, Chicago, IL, July 1995, pp. 862–865.
112. X. R. Cao and R. Liu, "General approach to blind source separation," *IEEE Trans. Signal Process.*, vol. 44, no. 3, pp. 562–571, Mar. 1996.
113. A.-J. Van der Veen and A. Paulraj, "An analytical constant modulus algorithm," *IEEE Trans. Signal Process.*, vol. 44, no. 5, pp. 1136–1155, May 1996.
114. L. K. Hansen and G. Xu, "A fast sequential source separation algorithm for digital cochannel signals," *IEEE Signal Process. Lett.*, vol. 4, no. 2, pp. 58–61, Feb. 1997.
115. A. Belouchrani, K. Adeb-Meraim, J. F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 434–444, Feb. 1997.
116. V. A. N. Barroso, J. M. F. Moura, and J. Xavier, "Blind array channel division multiple access," *IEEE Trans. Signal Process.*, vol. 46, no. 3, pp. 737–752, Mar. 1998.
117. M. Feng and K. D. Kammeyer, "Blind source separation for communication signals using antenna arrays," in *IEEE ICUPC*, Vol. 1, Florence, Italy, Oct. 1998, pp. 665–669.
118. M. Wax and Y. Anu, "A least squares approach to blind beamforming," *IEEE Trans. Signal Process.*, vol. 47, no. 1, pp. 231–234, Jan. 1999.
119. J. R. Treichler and M. G. Larimore, "New processing techniques based on the constant modulus adaptive algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 33, no. 2, pp. 420–431, Apr. 1985.
120. D. N. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communications systems," *IEEE Trans. Commun.*, vol. 28, no. 11, pp. 1867–1875, Nov. 1980.
121. J. R. Treichler and B. G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 31, no. 2, pp. 459–471, Apr. 1983.
122. R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propagat.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.
123. R. Roy and T. Kailath, "ESPRIT—Estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 7, pp. 984–995, July 1989.
124. R. Gooch and J. Lundell, "The CM array: An adaptive beamformer for constant modulus signals," in *IEEE ICASSP*, Tokyo, Japan, Apr. 1986, pp. 2523–2526.
125. J. Lundell and B. Widrow, "Applications of the constant modulus adaptive beamformer to constant and non-constant modulus signals," in *Asilomar CSSC*, Pacific Grove, CA, 1987, pp. 432–436.
126. S. Mayrargue, "Spatial equalisation of a radio-mobile channel without beamforming using the constant modulus algorithm (CMA)," in *IEEE ICASSP*, Vol. 3, Minneapolis, MN, Apr. 1993, pp. 344–347.
127. S. Mayrargue, "A blind spatio-temporal equaliser for a radio-mobile channel using the constant modulus algorithm (CMA)," in *IEEE ICASSP*, Vol. 3, Adelaide, Australia, Apr. 1994, pp. 317–320.
128. B. J. Sublett, R. P. Gooch, and S. H. Goldberg, "Separation and bearing estimation of co-channel signals," in *IEEE MILCOM*, Boston, MA, Oct. 1989, pp. 629–634.
129. J. J. Shynk and R. P. Gooch, "Convergence properties of the multistage CMA adaptive beamformer," in *IEEE Asilomar CSSC*, Pacific Grove, CA, Nov. 1993, pp. 622–626.

130. J. J. Shynk and R. P. Gooch, "The constant modulus array for cochannel signal copy and direction finding," *IEEE Trans. Signal Process.*, vol. 44, no. 3, pp. 652–660, Mar. 1996.
131. L. Castedo, C. J. Escudero and A. Dapena, "A blind signal separation method for multiuser communications," *IEEE Trans. Signal Process.*, vol. 45, no. 5, pp. 1343–1348, May 1997.
132. C. B. Papadias and A. J. Paulraj, "A constant modulus algorithm for multiuser signal separation in presence of delay spread using antenna arrays," *IEEE Signal Process. Lett.*, vol. 4, no. 6, pp. 178–181, June 1997.
133. S. Lambotharan and J. Chambers, "On the surface characteristics of a mixed constant modulus and cross-correlation criterion for blind equalization of a MIMO channel," *Elsevier Signal Process.*, vol. 74, pp. 209–216, 1999.
134. C. R. Johnson, Jr., P. Schniter, T. J. Endres, J. D. Behm, D. R. Brown, and R. A. Casas, "Blind equalization using the constant modulus criterion: A review," *Proc. IEEE*, vol. 86, no. 10, pp. 1927–1950, Oct. 1998.
135. A.-J. Van der Veen, S. Talwar, and A. Paulraj, "A subspace approach to blind space-time signal processing for wireless communication system," *IEEE Trans. Signal Process.*, vol. 45, no. 1, pp. 173–190, Jan. 1997.
136. J. Xavier and V. Barroso, "Blind source separation, ISI cancellation and carrier phase recovery in SDMA systems for mobile communications," *Wireless Personal Commun.*, vol. 10, pp. 53–76, June 1999.
137. J. M. F. Xavier, V. A. N. Barroso, and J. M. F. Moura, "Closed-form blind channel identification and source separation in SDMA systems through correlative coding," *IEEE J. Select. Areas Commun.*, vol. 16, no. 8, pp. 1506–1517, Oct. 1998.
138. S. Cruces and L. Castedo, "A Gauss-Newton method for blind source separation of convolutive mixtures," in *IEEE ICASSP*, Vol. 4, Seattle, WA, May 1998, pp. 2093–2096.
139. C. Simon, Ph. Loubaton, C. Vignat, C. Jutten, and G. d'Urso, "Blind source separation of convolutive mixtures by maximization of fourth-order cumulants: The non I.I.D. case," in *IEEE Asilomar CSSC*, Vol. 2, Pacific Grove, CA, Nov. 1998, pp. 1584–1588.
140. R. Djapic, G. Leus, and A.-J. Van Der Veen, "The Cramer-Rao bounds for blind and training based packet offset estimation in wireless ad hoc networks," in *IEEE SCVT*, Gent, Belgium. Nov. 2004.
141. V. Venkateswaran and A.-J. Van-der-Veen, "Source separation of asynchronous OFDM signals using superimposed training," in *IEEE ICASSP*, Vol. III, Honolulu, HI, Apr. 2007, pp. 385–388.
142. V. Venkateswaran, A.-J. Van-der-Veen, and M. Ghogho, "Joint source separation and offset estimation for asynchronous OFDM systems using subspace fitting," in *IEEE SPAWC*, Helsinki, Finland, June 2007, pp. 1–5.
143. V. Nájera-Bello, "Design and construction of a digital communications system based on implicit training," MSc. thesis, CINVESTAV-IPN, 2009, to be submitted (in Spanish).
144. S. Verdú, *Multiuser Detection*, Cambridge: Cambridge University Press, 1998.
145. T. Thomas and F. Vook, "Asynchronous interference in broadband CP communications," in *IEEE WCNC*, New Orleans, LA, Mar. 2003, pp. 568–572.
146. L. E. Zegers, "Common Bandwidth Transmission of Information Signals and Pseudonoise Synchronization Waveforms," *IEEE Transactions on Communication Technology.*, Vol. COM-16, no. 6, pp. 796–807, Dec. 1968.

## CHAPTER 7

---

# Unitary Design of Radar Waveform Diversity Sets

Michael D. Zoltowski<sup>1</sup>, Tariq R. Qureshi<sup>1</sup>, Robert Calderbank<sup>2</sup>, and Bill Moran<sup>3</sup>

<sup>1</sup> Purdue University

<sup>2</sup> Princeton University

<sup>3</sup> University of Melbourne

### 7.1 INTRODUCTION

In active sensing systems, the objective is to design a communication system that allows one to learn the environment, which could be one or more moving targets in the case of a radar. In a radar system, the transmitted waveforms are reflected by the target, and the reflected returns are then processed at the receiving end to determine the location and speed of the target. The delay in the received waveforms corresponds to the distance of the target from the radar, and the Doppler shift determines the speed at which the target is moving [1]. Therefore, it is desired to transmit a waveform that provides good resolution in terms of the delay–Doppler properties of the radar returns. This is characterized by the use of ambiguity functions, which measure the delay–Doppler correlation of the received waveforms with the actual transmitted waveform. The ambiguity function [1] of a waveform  $s(t)$  is given by

$$\chi(\tau, v) = \int_{-\infty}^{\infty} s(t)s^*(t - \tau)e^{-j2\pi vt} dt, \quad (7.1)$$

where  $\tau$  and  $v$  are the delay and the Doppler shift, respectively. A perfect radar waveform would have the ambiguity function of the form

$$\chi(\tau, v) = \delta(\tau)\delta(v), \quad (7.2)$$

which means that the spike in the ambiguity function would correspond to the correct delay and Doppler properties of the target. However, waveforms with this kind of ambiguity function do not exist [1]. However, it should be noted that for correct target detection, waveforms with a thumbtack-shaped ambiguity function are not always necessary. In particular, if there is only one target or if there are multiple targets that are

reasonably well separated in the delay–Doppler domain, a waveform with ambiguity function that decays sufficiently fast in delay–Doppler so as to not confuse the nearby targets is sufficient.

In radar systems, the advantages of using closely spaced antennas at the receivers are well known [2–4], some of which are electronic steerability of the antenna array and the ability to use array processing techniques for improved detection. Recently, there has been a lot of interest in radars employing multiple antennas at both the transmitter and the receiver. These radars are commonly known as multiple-input multiple-output (MIMO) radars. MIMO radar [5] offers superior performance over the conventional radar systems in that it provides multiple independent views of the target if the antennas are placed sufficiently far apart. In a MIMO radar, different transmit waveforms are transmitted from the transmitting antennas, and the returns are then processed to determine the target presence, location, and speed. The challenge in MIMO radar, among other things, is waveform separation at the receiver. To this end, a widely studied approach has been to transmit orthogonal waveforms that can be separated at the receiver. However, the relative delay and Doppler shift of the transmitted waveforms might destroy their orthogonality, and, therefore, we need waveforms that remain orthogonal through a certain range of delay and Doppler shifts. The design of such waveforms has been studied extensively in the context of code division multiple access (CDMA) systems [6].

Howard et al. [7] proposed a new multichannel radar scheme employing polarization diversity for getting multiple independent views of the target. In this scheme, Golay pairs [8] of phase-coded waveforms are used to provide synchronization, and Alamouti coding [9] is used to coordinate transmission of these waveforms on the horizontal and vertical polarizations. The combination of Golay complementary sequences and Alamouti coding makes it possible to do radar ambiguity polarimetry on a pulse-by-pulse basis, which reduces the signal processing complexity as compared to distributed aperture radar. This scheme [7] has been shown to provide the same detection performance as the single-channel radar with significantly smaller transmit energy, or provide detection over greater ranges with the same transmit energy as the single-channel radar.

The work done by Howard et al. [7] is based on processing the transmitted waveform matrix at the receiver in a manner that allows us to separate the transmitted waveforms at the receiver and exploit the diversity inherent in an active sensing environment due to its multipath nature. In [7], the waveform separation was achieved through the use of Golay complementary sequences. In this chapter, multiple radar waveforms are simultaneously transmitted from different “virtual antenna” elements where each antenna element is a transceiver. The goal is to process the returns in such a way that the overall ambiguity function is a sum of individual ambiguity functions, such that the sum better approximates the desired thumbtack shape. The use of the term “virtual antenna” here can also include simultaneous beams formed from the same aperture but pointed to different angles, or beams pointed to the same angle but formed from different subapertures. We present an example of a  $4 \times 4$  system with 2 dually polarized transceivers. A  $4 \times 4$  unitary design implies the following features. First, it dictates the scheduling of the waveforms over the 4 virtual antennas over 4 (pulse repetition intervals) (PRIs). Second, it tells us how the matched filtering of the returns over 4 PRIs are combined in such a way so as to achieve both perfect separation (of the superimposed returns) *and* perfect reconstruction. Perfect reconstruction implies that

the sum of the time autocorrelations associated with each of the 4 waveforms is a delta function. The net result of the processing of 4 PRIs over 4 virtual antennas yields 16 cross-correlations all of which ideally exhibit a sharp peak at the target delay. Conditions for both perfect separation and perfect reconstruction are developed, and a variety of waveform sets satisfying both are presented. Third, biorthogonal methods are developed for achieving both perfect reconstruction and perfect separation with adaptive waveforms that are matched to the propagation environment.

## 7.2 2 × 2 SPACE-TIME DIVERSITY WAVEFORM DESIGN

In this section, we introduce a  $2 \times 2$  waveform design presented in [7] that uses complementary sequences and Alamouti coding [9] to coordinate transmission over multiple antennas. Complementary sequences are characterized by the special property that the sum of their autocorrelation functions produces a Dirac delta function [10].

### 7.2.1 Complementary Sequences and Alamouti Coding

A pair of sequences  $x_n$  and  $y_n$  satisfy the Golay property if the sum of their autocorrelation functions satisfies

$$R_{xx}(k) + R_{yy}(k) = \begin{cases} 2N, & \text{if } k = 0, \\ 0, & \text{if } k \neq 0. \end{cases} \quad (7.3)$$

If the elements of these sequences are fourth roots of unity, these are called the complex Golay complementary sequences. We note that the real complementary sequences with elements taking the values  $\pm 1$  are a subclass of the complex complementary sequences.

Alamouti coding is an orthogonal space-time block code (OSTBC) [11] first proposed in [9]. The code is given by a  $2 \times 2$  matrix where the columns represent different time (frequency) slots, and the rows represent different antennas. If  $s_1$  and  $s_2$  are the two different waveforms to be transmitted, the Alamouti code is given by

$$\mathbf{S} = \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix}. \quad (7.4)$$

The received signal can be written as

$$\mathbf{r} = \begin{bmatrix} h_1 & h_2 \\ -h_2^* & h_1^* \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ -n_2^* \end{bmatrix} \quad (7.5)$$

with

$$\mathbf{r} = \begin{bmatrix} r_1 \\ -r_2^* \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} h_1 & h_2 \\ -h_2^* & h_1^* \end{bmatrix}, \quad \mathbf{n} = \begin{bmatrix} n_1 \\ -n_2^* \end{bmatrix}.$$

The matrix  $\mathbf{H}$  is orthogonal since

$$\mathbf{H}^H \mathbf{H} = \begin{bmatrix} |h_1|^2 + |h_2|^2 & 0 \\ 0 & |h_1|^2 + |h_2|^2 \end{bmatrix}. \quad (7.6)$$

If the path gains are known to the receiver, we have

$$\mathbf{H}^H \mathbf{r} = (|h_1|^2 + |h_2|^2) \begin{bmatrix} s_1 \\ -s_2^* \end{bmatrix} + \mathbf{n}', \quad (7.7)$$

and we can detect the signals since  $\mathbf{n}' = \mathbf{H}^H \mathbf{n}$  is still white because  $\mathbf{H}$  is orthogonal [12]. It should be noted that in the case of Alamouti coding, we need to know the channel at the receiver in order to detect the transmitted symbols. In our case, the transmitted waveforms would be known to the radar receiver a priori, and no channel information would be required to detect the target.

### 7.2.2 Polarization Diversity Code Design

In polarization diversity, we can use the two orthogonal polarizations as two independent channels, thus giving us a  $2 \times 2$  system using only one antenna. The matrix  $\mathbf{H}$  in the MIMO system is now replaced by the scattering matrix of the target [7], which is given by

$$\Sigma = \begin{bmatrix} \sigma_{VV} & \sigma_{VH} \\ \sigma_{HV} & \sigma_{HH} \end{bmatrix}, \quad (7.8)$$

where  $\sigma_{VH}$  is the coefficient of the scattering of an incident horizontally polarized field into the vertical polarization. The measurement  $\mathbf{H}$ , however, also depends upon the polarization coupling properties of the transmitting and receiving antennas and is given by

$$\mathbf{H} = \mathbf{C}_{R_x} \Sigma \mathbf{C}_{T_x} = \begin{bmatrix} h_{VV} & h_{VH} \\ h_{HV} & h_{HH} \end{bmatrix}, \quad (7.9)$$

where  $\mathbf{C}_{T_x}$  and  $\mathbf{C}_{R_x}$  are matrices representing the polarization coupling properties of the transmit and receive antennas. Since the transmit and receive antennas are common in most radar systems, the matrices  $\mathbf{C}_{T_x}$  and  $\mathbf{C}_{R_x}$  are conjugate.

The following matrix represents the code matrix that we wish to transmit, with the columns representing different time slots and rows representing two orthogonal polarizations (channels):

$$\mathbf{S} = \begin{bmatrix} s_1[n] & -s_2^*[n] \\ s_2[n] & s_1^*[n] \end{bmatrix}. \quad (7.10)$$

We now define the matrix with which we process the received waveform as

$$\mathbf{S}^* = \begin{bmatrix} s_1^*[-n] & s_2^*[-n] \\ -s_2[n] & s_1[n] \end{bmatrix}. \quad (7.11)$$

These matrices are orthogonal, as given by

$$\mathbf{S} * \mathbf{S}^* = \begin{bmatrix} s_1[n] * s_1^*[-n] + s_2^*[-n] * s_2[n] & s_1[n] * s_2^*[-n] - s_2^*[-n] * s_1[n] \\ s_2[n] * s_1^*[-n] - s_1^*[-n] * s_2[n] & s_2[n] * s_2^*[-n] + s_1^*[-n] * s_1[n] \end{bmatrix}. \quad (7.12)$$

The diagonal elements are zero everywhere except  $n = 0$ , and the off-diagonal elements are zero everywhere because two identical terms are being subtracted. This follows from the fact that convolution is commutative and hence

$$s_1[n] * s_2^*[-n] = s_2^*[-n] * s_1[n]. \quad (7.13)$$

### 7.2.3 Polarization Diversity and Radar Detection

We now present an analysis based on the use of multiple antennas. We consider a system with a single dually polarized transmit and receive antenna. In order to perform target detection, we stack the columns of the  $2 \times 2$  channel coefficient matrix into a vector  $T$ , which is given under different hypotheses as

$$T = \begin{cases} E_t \mathbf{h} + \mathbf{n}, & H_1, \\ \mathbf{n}, & H_0, \end{cases} \quad (7.14)$$

where  $\mathbf{h}$  is a  $4 \times 1$  vector of the channel coefficients. The probability density function (pdf) of  $T$  is given by

$$T = \begin{cases} CN(0, 2N_0 \mathbf{I}_{4 \times 4}) & : H_0, \\ CN(0, (2E_t \sigma^2 + 2N_0) \mathbf{I}_{4 \times 4}) & : H_1, \end{cases} \quad (7.15)$$

where  $CN$  is the complex Gaussian random variable,  $\sigma^2$  is variance of each component of  $\mathbf{h}$ , and  $N_0$  is noise power. The likelihood ratio detector is an energy detector that computes the energy in the received vector under both hypotheses and is given by

$$\|T\|^2 > \gamma, \quad (7.16)$$

where  $\gamma$  is the detection threshold. The probability of false alarm  $P_F$  and probability of detection  $P_D$  require an investigation of the pdf of  $\|T\|^2$ . Let us define

$$T' = \begin{cases} \sqrt{N_0}(\sqrt{2}T'), & H_0, \\ \sqrt{E_t \sigma^2 + N_0}(\sqrt{2}T'), & H_1, \end{cases} \quad (7.17)$$

where

$$T' = CN(0, \mathbf{I}_{4 \times 4}).$$

From [13], we know that  $2\|T'\|^2 \sim \chi_{2n}^2$ . So, the pdf of the test statistic is given by

$$f_{\|T\|^2}(t) = \begin{cases} \frac{t^3 \exp\left[\frac{-t}{2(E_t \sigma^2 + N_0)}\right]}{(2(E_t \sigma^2 + N_0))^4 (3)!} & H_1 \\ \frac{t^3 \exp\left(\frac{-t}{2N_0}\right)}{(2N_0)^4 (3)!} & H_0. \end{cases} \quad (7.18)$$

The probability of false alarm is given by

$$P_F(\gamma) = \sum_{k=0}^3 \left(\frac{\gamma}{2N_0}\right)^k \frac{\exp\left(\frac{-\gamma}{2N_0}\right)}{k!}, \quad (7.19)$$

and the corresponding probability of detection is

$$P_D(\gamma) = \sum_{k=0}^3 \left( \frac{\gamma}{2(E_t\sigma^2 + N_0)} \right)^k \frac{\exp\left[\frac{-\gamma}{2(E_t\sigma^2 + N_0)}\right]}{k!}. \quad (7.20)$$

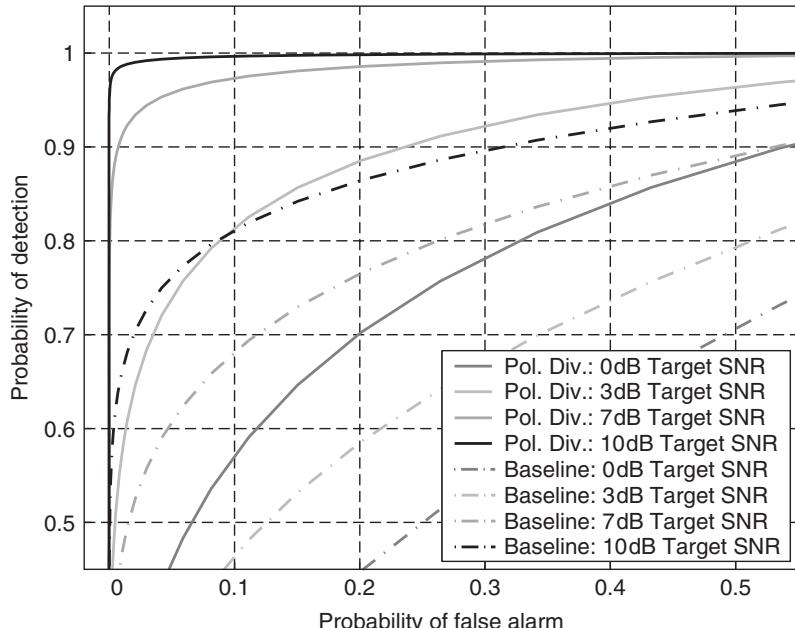
From [14], we know that for a single-channel system, the probabilities of false alarm and detection are given by

$$P_F = \exp\left(-\frac{\gamma}{2N_0}\right), \quad (7.21)$$

$$P_D = \exp\left[\frac{-\gamma}{2(E_t\sigma^2 + N_0)}\right], \quad (7.22)$$

$$P_F = P_D^{(S+1)}. \quad (7.23)$$

A comparison of the receiver operating characteristics (ROC) curves for the baseline system versus the polarization diversity system is given in Figure 7.1, which shows the obvious advantage of using polarization diversity over the baseline system. The use of Alamouti coding combined with Golay pairs and the transmission over orthogonal polarizations allows for the estimation of the polarimetric properties of the target on a pulse-by-pulse basis, thereby simplifying the receiver end signal processing. In the next section, we explore the problem of designing adaptive waveforms for  $4 \times 4$  systems.



**Figure 7.1** Comparison of ROC curves for the  $2 \times 2$  polarization diversity system versus the baseline system.

### 7.3 4 × 4 SPACE-TIME DIVERSITY WAVEFORM DESIGN

In the previous section, we developed a framework for  $2 \times 2$  diversity waveforms achieving perfect reconstruction at the receiver. In this section, we extend those results to the  $4 \times 4$  case.

#### 7.3.1 4 × 4 Waveform Scheduling

For the  $2 \times 2$  case, we saw how the waveforms are scheduled over space-time to achieve perfect reconstruction at the receiving end. We will now look at scheduling a set of waveforms over four antennas. Consider a set of waveforms satisfying

$$s_1[n] * s_1^*[-n] + s_2[n] * s_2^*[-n] + s_3[n] * s_3^*[-n] + s_4[n] * s_4^*[-n] = 4N\delta[n]. \quad (7.24)$$

We schedule these waveforms over space-time as given by the following matrix, where the rows represent spatial dimensions and columns represent temporal dimensions:

$$\mathbf{S} = \begin{bmatrix} s_1[n] & s_2^*[-n] & s_3[n] & s_4^*[-n] \\ -s_2[n] & s_1^*[-n] & -s_4[n] & s_3^*[-n] \\ -s_3[n] & s_4^*[-n] & s_1[n] & -s_2^*[-n] \\ -s_4[n] & -s_3^*[-n] & s_2[n] & s_1^*[-n] \end{bmatrix}. \quad (7.25)$$

These transmitted waveforms are coupled to the receiver through a channel matrix given by

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix}, \quad (7.26)$$

where we have made the assumption that the channel does not change for the duration of the waveforms that constitute the matrix  $\mathbf{E}$ . The received waveform matrix  $\mathbf{R}$ , ignoring the noise component for now, is given by

$$\mathbf{R} = \mathbf{S} * \mathbf{H}, \quad (7.27)$$

where  $*$  denotes convolution. At the receiver, the goal is to process  $\mathbf{R}$  so as to achieve perfect separation of the waveforms. In mathematical terms, we want to process  $\mathbf{R}$  with a matrix  $\mathbf{F}$  such that

$$\mathbf{R} * \mathbf{F} = \alpha \mathbf{I}. \quad (7.28)$$

Let us define  $\mathbf{S}^*$ , which is the time-reversed conjugate of  $\mathbf{S}$ , as

$$\mathbf{S}^* = \begin{bmatrix} s_1^*[-n] & -s_2^*[-n] & -s_3^*[-n] & -s_4^*[-n] \\ s_2[n] & s_1[n] & s_4[n] & -s_3[n] \\ s_3^*[-n] & -s_4^*[-n] & s_1^*[-n] & -s_2^*[-n] \\ s_4[n] & s_3[n] & -s_2[n] & s_1[n] \end{bmatrix}. \quad (7.29)$$

We process the received waveform matrix  $\mathbf{R}$  by  $\mathbf{S}^*$ , that is,

$$\mathbf{U} = \mathbf{R} * \mathbf{S}^*, \quad (7.30)$$

and we want

$$\mathbf{U} = \alpha \mathbf{I}. \quad (7.31)$$

Since the waveforms satisfy (7.24), we have

$$\mathbf{U} = \begin{bmatrix} \delta[n-D] & 0 & -\phi[n-D] & 0 \\ 0 & \delta[n-D] & 0 & \phi[n-D] \\ \phi[n-D] & 0 & \delta[n-D] & 0 \\ 0 & -\phi[n-D] & 0 & \delta[n-D] \end{bmatrix}, \quad (7.32)$$

where

$$\phi[n] = -s_3[n] * s_1^*[-n] + s_4^*[-n] * s_2[n] + s_1[n] * s_3^*[-n] - s_2^*[-n] * s_4^*[-n]. \quad (7.33)$$

We will work with the matrix  $\mathbf{U}$  in the next section to derive conditions for perfect waveform separation and reconstruction.

### 7.3.2 Conditions for Perfect Reconstruction and Separation

Let us define the Key matrix  $\Phi[n]$  as

$$\Phi[n] = \begin{bmatrix} \theta[n] & 0 & -\phi[n] & 0 \\ 0 & \theta[n] & 0 & \phi[n] \\ \phi[n] & 0 & \theta[n] & 0 \\ 0 & -\phi[n] & 0 & \theta[n] \end{bmatrix}. \quad (7.34)$$

We say that waveform design has *perfect reconstruction* property if

$$\theta[n] = \delta[n], \quad (7.35)$$

and if

$$\phi[n] = 0 \quad \forall n, \quad (7.36)$$

we say that the waveform design has *perfect separation* property.

We can see that  $\phi[n]$  is conjugate symmetric, that is,

$$\phi[-n] = -\phi^*[n], \quad (7.37)$$

which implies that if  $\phi[n]$  is real valued, then

$$\phi[0] = 0. \quad (7.38)$$

We can write  $\phi[n]$  as

$$\phi[n] = (-s_3[n] * s_1^*[-n] + s_1[n] * s_3^*[-n]) + (s_4^*[-n] * s_2[n] - s_2^*[-n] * s_4^*[-n]). \quad (7.39)$$

From this, we can see that if all waveforms exhibit conjugate symmetry, that is,

$$s_i[n] = s_i^*[-n] \quad (7.40)$$

for  $i = 1, 2, 3, 4$ , then

$$\phi[n] = 0, \quad (7.41)$$

and we achieve perfect separation. Also, if the waveforms are time-reversed versions of each other, that is,

$$s_1[n] = s_2^*[-n] \quad (7.42)$$

and

$$s_3[n] = s_4^*[-n], \quad (7.43)$$

then we also have

$$\phi[n] = 0. \quad (7.44)$$

Equations (7.40) and (7.42) give us the conditions for perfect waveform reconstruction and separation at the receiver end. It should, however, be noted that this is not an exhaustive list of conditions for perfect reconstruction and separation. Other possible conditions may still exist.

### 7.3.3 4 × 4 Polarization Diversity Radar Detection

We use the ideas developed in the previous section to a  $4 \times 4$  system employing two antennas transmitting and receiving over both the horizontal and the vertical polarization. In a  $4 \times 4$  system, we need four waveforms, but we know that if the waveforms satisfy (7.24) and if they are time-reversed versions of each other, we can achieve perfect separation. Therefore, we use one Golay pair and the time-reversed version of this pair to make four waveforms. The transmitted waveform matrix is given by

$$\tilde{\mathbf{S}}_1 = \begin{bmatrix} \mathbf{S} & -\mathbf{S}^H \\ \mathbf{S} & \mathbf{S}^H \end{bmatrix}. \quad (7.45)$$

The perfect separation is achieved by observing that

$$\tilde{\mathbf{S}}_1 * \tilde{\mathbf{S}}_1^H = \begin{bmatrix} \mathbf{S} * \mathbf{S}^H + \mathbf{S}^H * \mathbf{S} & \mathbf{S} * \mathbf{S}^H - \mathbf{S}^H * \mathbf{S} \\ \mathbf{S} * \mathbf{S}^H - \mathbf{S}^H * \mathbf{S} & \mathbf{S} * \mathbf{S}^H + \mathbf{S}^H * \mathbf{S} \end{bmatrix} = 2\alpha \mathbf{I}_{2 \times 2}. \quad (7.46)$$

The target detection strategy is the same as in Section 7.1 and because the waveform matrices are still orthogonal. The only difference is that the pdf of the test statistic  $\|T\|^2$  is now given by

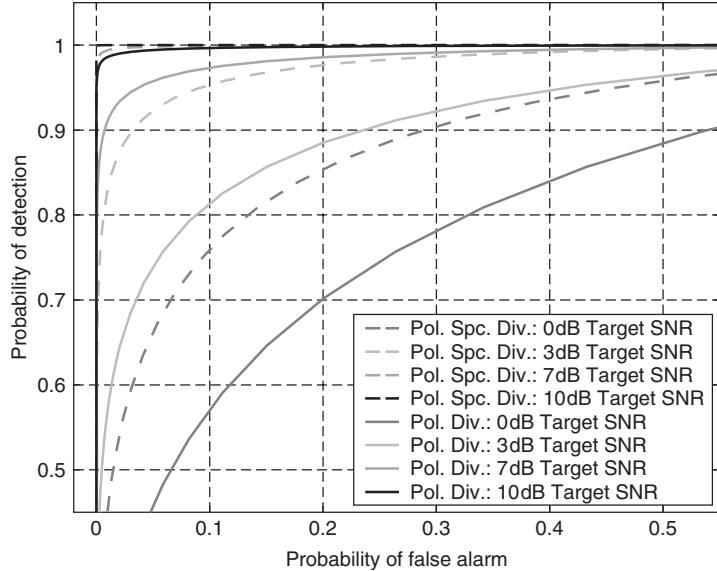
$$f_{\|T\|^2}(t) = \begin{cases} \frac{t^{15} \exp\left[\frac{-t}{2(E_t \sigma^2 + N_0)}\right]}{(2(E_t \sigma^2 + N_0))^{16}(15)!} & : H_1, \\ \frac{t^{15} \exp\left(\frac{-t}{2N_0}\right)}{(2N_0)^{16}(15)!} & : H_0, \end{cases} \quad (7.47)$$

and the corresponding probability of false alarm  $P_F$  and probability of detection  $P_D$  are given by

$$P_F(\gamma) = \sum_{k=0}^1 5 \left( \frac{\gamma}{2N_0} \right)^k \frac{\exp\left(\frac{-\gamma}{2N_0}\right)}{k!} \quad (7.48)$$

and

$$P_D(\gamma) = \sum_{k=0}^1 5 \left( \frac{\gamma}{2(E_t \sigma^2 + N_0)} \right)^k \frac{\exp\left[\frac{-\gamma}{2(E_t \sigma^2 + N_0)}\right]}{k!}. \quad (7.49)$$



**Figure 7.2** Comparison of ROC curves for the  $4 \times 4$  and  $2 \times 2$  polarization diversity system versus the baseline system.

A comparison of the ROC curves for the baseline system, the single-antenna polarization diversity system and the multiple-antenna polarization diversity system is given in Figure 7.2. We can see that the multiple-antenna system provides improved detection performance as compared to the baseline and the single-channel polarization diversity systems.

#### 7.4 WAVEFORM FAMILIES BASED ON KRONECKER PRODUCTS

In this section, we look at the problem of waveform separation using the Kronecker products [10]. Consider two row vectors  $\mathbf{a}$  and  $\mathbf{b}$  given by

$$\mathbf{a} = [ \ a_1 \ a_2 \ a_3 \ ], \quad \mathbf{b} = [ \ b_1 \ b_2 \ b_3 \ ].$$

Their Kronecker product is defined as

$$\mathbf{a} \otimes \mathbf{b} = [ \ a_1b_1 \ a_1b_2 \ a_1b_3 \ a_2b_1 \ a_2b_2 \ a_2b_3 \ a_3b_1 \ a_3b_2 \ a_3b_3 \ ]. \quad (7.50)$$

We will also use a modified Kronecker product, which we denote by  $\otimes_N$ .

For  $N = 2$ , the modified Kronecker product of  $\mathbf{a}$  and  $\mathbf{b}$  is given by

$$\mathbf{a} \otimes_N \mathbf{b} = [ \ a_1b_1 \ a_1b_2 \ a_1b_3 + a_2b_1 \ a_2b_2 \ a_2b_3 + a_3b_1 \ a_3b_2 \ a_3b_3 \ ], \quad (7.51)$$

which is arrived at by observing that

$$\mathbf{a} \otimes_N \mathbf{b} = \left\{ \begin{array}{l} a_1b_1 \quad a_1b_2 \quad a_1b_3 \\ + \quad \quad \quad a_2b_1 \quad a_2b_2 \quad a_2b_3 \\ + \quad \quad \quad a_3b_1 \quad a_3b_2 \quad a_3b_3 \\ = \quad a_1b_1 \quad a_1b_2 \quad a_1b_3 + a_2b_1 \quad a_2b_2 \quad a_2b_3 + a_3b_1 \quad a_3b_2 \quad a_3b_3 \end{array} \right\}. \quad (7.52)$$

We now develop waveform families using these Kronecker products of different sequences.

### 7.4.1 Kronecker Products of Golay Complementary Sequences

Now consider two pairs of complementary Golay sequences:

$$\varepsilon_1[n] * \varepsilon_1^*[-n] + \varepsilon_2[n] * \varepsilon_2^*[-n] = N_1 \delta[n], \quad (7.53)$$

$$\varepsilon_3[n] * \varepsilon_3^*[-n] + \varepsilon_4[n] * \varepsilon_4^*[-n] = N_2 \delta[n]. \quad (7.54)$$

We form the transmitted waveforms of these sequences using the Kronecker product as

$$s_1[n] = \varepsilon_1[n] \otimes \varepsilon_3[n], \quad (7.55)$$

$$s_2[n] = \varepsilon_1[n] \otimes \varepsilon_4[n], \quad (7.56)$$

$$s_3[n] = \varepsilon_2[n] \otimes \varepsilon_3[n], \quad (7.57)$$

$$s_4[n] = \varepsilon_2[n] \otimes \varepsilon_4[n]. \quad (7.58)$$

Consider forming the autocorrelation

$$r[m] = r_{s_1 s_1}[m] + r_{s_2 s_2}[m] + r_{s_3 s_3}[m] + r_{s_4 s_4}[m]. \quad (7.59)$$

Since

$$z[n] = x[n] \otimes_N y[n] \Rightarrow r_{zz}[m] = r_{xx}[m] \otimes_N r_{yy}[m], \quad (7.60)$$

we have that

$$\begin{aligned} r[m] &= (r_{\varepsilon_1 \varepsilon_1}[m] \otimes_N r_{\varepsilon_3 \varepsilon_3}[m]) + (r_{\varepsilon_1 \varepsilon_1}[m] \otimes_N r_{\varepsilon_4 \varepsilon_4}[m]) \\ &\quad + (r_{\varepsilon_2 \varepsilon_2}[m] \otimes_N r_{\varepsilon_3 \varepsilon_3}[m]) + (r_{\varepsilon_2 \varepsilon_2}[m] \otimes_N r_{\varepsilon_4 \varepsilon_4}[m]) \\ &= (r_{\varepsilon_1 \varepsilon_1}[m] \otimes_N (r_{\varepsilon_3 \varepsilon_3}[m] + r_{\varepsilon_4 \varepsilon_4}[m])) \\ &\quad + (r_{\varepsilon_2 \varepsilon_2}[m] \otimes_N (r_{\varepsilon_3 \varepsilon_3}[m] + r_{\varepsilon_4 \varepsilon_4}[m])) \\ &= N_1 (r_{\varepsilon_1 \varepsilon_1}[m] \otimes_N \delta[m]) + N_1 (r_{\varepsilon_2 \varepsilon_2}[m] \otimes_N \delta[m]) \\ &= N_1 ((r_{\varepsilon_1 \varepsilon_1}[m] + r_{\varepsilon_2 \varepsilon_2}[m]) \otimes_N \delta[m]) \\ &= N_1 N_2 (\delta[m] \otimes_N \delta[m])[m] \\ &= N_1 N_2 \delta[m]. \end{aligned} \quad (7.61)$$

Therefore, the Kronecker product preserves the autocorrelation properties of the original sequences in this case. Let us look at the main and off-diagonal sequences of the Key matrix of these waveforms. As an example, consider the real-valued Golay code pairs of length 10,

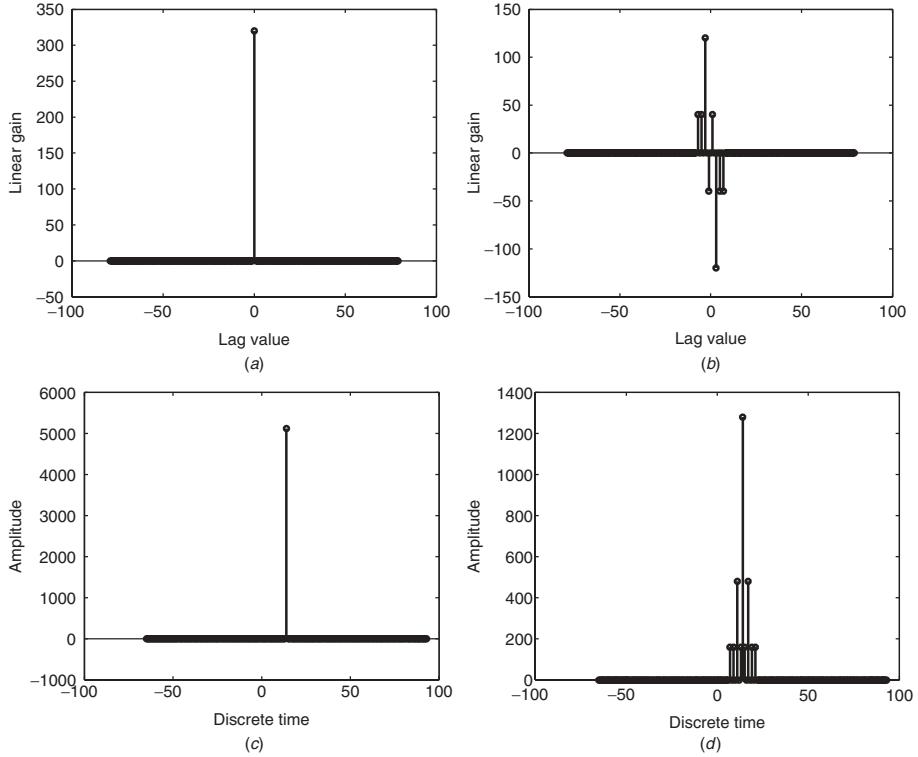
$$\varepsilon_1[n] = \{1, 1, -1, 1, -1, 1, -1, -1, 1, 1\}, \quad (7.62)$$

$$\varepsilon_2[n] = \{1, 1, -1, 1, 1, 1, 1, 1, -1, -1\}, \quad (7.63)$$

and length 8,

$$\varepsilon_3[n] = \{-1, -1, -1, 1, 1, 1, -1, 1\}, \quad (7.64)$$

$$\varepsilon_4[n] = \{-1, -1, -1, 1, -1, -1, 1, -1\}. \quad (7.65)$$



**Figure 7.3** (a) Main diagonal sequence, (b) off-diagonal sequence, (c) coherent sum of all 16 cross correlations, and (d) incoherent sum of all 16 cross correlations.

The main diagonal sequence in the Key matrix is shown in Figure 7.3a, and the off-diagonal sequence  $\phi[n]$  as given in the Key matrix is shown in Figure 7.3b.

From these figures, we see that while the main diagonal term is nonzero only at the correct lag value, we do have residual cross terms in the off-diagonal sequence. We also observe that the off-diagonal sequence is antisymmetric, which means that it does not affect the correlation peak at the true lag value, even though  $\phi[n]$  is not identically zero in this case.

Consider the case of a single-point target at delay  $D$ . The received waveform matrix, ignoring the noise, is given by

$$\begin{bmatrix} h_{11}\delta[l] + h_{13}\phi[l] & h_{12}\delta[l] - h_{14}\phi[l] & h_{13}\delta[l] - h_{11}\phi[l] & h_{14}\delta[l] + h_{12}\phi[l] \\ h_{21}\delta[l] + h_{23}\phi[l] & h_{22}\delta[l] - h_{24}\phi[l] & h_{23}\delta[l] - h_{21}\phi[l] & h_{24}\delta[l] + h_{22}\phi[l] \\ h_{31}\delta[l] + h_{33}\phi[l] & h_{32}\delta[l] - h_{34}\phi[l] & h_{33}\delta[l] - h_{31}\phi[l] & h_{34}\delta[l] + h_{32}\phi[l] \\ h_{41}\delta[l] + h_{43}\phi[l] & h_{42}\delta[l] - h_{44}\phi[l] & h_{43}\delta[l] - h_{41}\phi[l] & h_{44}\delta[l] + h_{42}\phi[l] \end{bmatrix} \quad (7.66)$$

where  $l = n - D$  and  $D$  is the true target delay. Now consider the first and the third term in the first row of the above matrix. If we were able to estimate the channel gains  $h_{11}$  and  $h_{13}$ , we can form

$$h_{11}(h_{11}\delta[l] + h_{13}\phi[l]) + h_{13}(h_{13}\delta[l] + h_{11}\phi[l]) = (h_{11}^2 + h_{13}^2)\delta[l]. \quad (7.67)$$

This shows that if we multiply each correlation output by its corresponding value at its peak lag position, we get an signal-to-noise ratio (SNR) gain similar to MRC (maximal ratio combining). Also, the cross terms due to the off-diagonal term  $\phi[n]$  vanish. Therefore, if we were to estimate the channel, the off-diagonal terms would vanish even when the waveforms do not completely satisfy the conditions of Section 7.3. The main and off-diagonal sequences are shown in Figures 7.3c and 7.3d.

#### 7.4.2 Kronecker Products of Barker Codes

Consider two Barker sequences:

$$\begin{aligned} b_1[n] &= \{1, 1, 1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1\}, \\ b_2[n] &= \{0, 1, 1, 1, -1, -1, -1, 1, -1, -1, 0\}. \end{aligned} \quad (7.68)$$

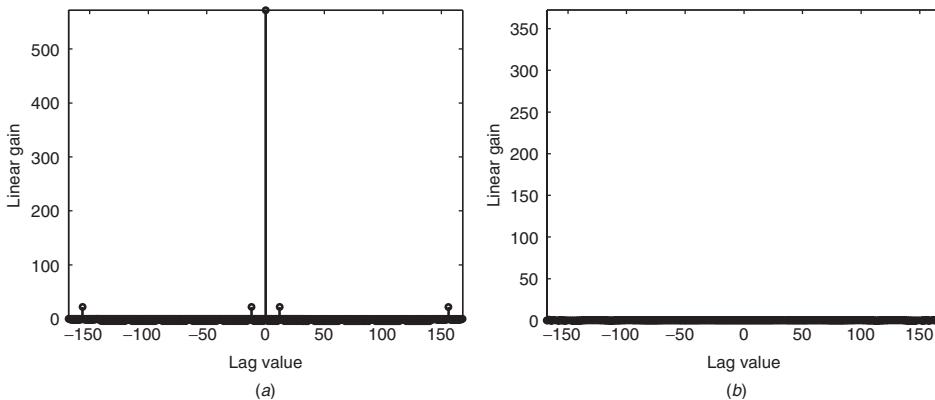
We form the transmitted waveforms as

$$\begin{aligned} s_1[n] &= b_1[n] \otimes b_2[n], \\ s_2[n] &= e_1^*[-n], \\ s_3[n] &= b_2[n] \otimes b_1[n], \\ s_4[n] &= e_3^*[-n]. \end{aligned} \quad (7.69)$$

These waveforms satisfy one of the conditions that leads to perfect separation given in Section 7.3, but their autocorrelation functions do not sum to a delta function. The diagonal and off-diagonal autocorrelation sequences are given in Figures 7.4a and 7.4b. Unlike the Golay complementary codes, these sequences are not identically zero except at the true lag position in the main diagonal. However, the off-diagonal terms are identically zero because of the fact that these waveforms are time-reversed versions of each other.

#### 7.4.3 Conjugate-Symmetric Transmit Waveforms

In this section, we look at waveforms that exhibit conjugate symmetry. An important aspect in the design of conjugate-symmetric waveforms is their DFT (discrete



**Figure 7.4** (a) Main diagonal sequence: some of values around the main lobe are nonzero.  
(b) Off-diagonal sequence is identically zero.

Fourier transform) properties. Since the DFT of a conjugate-symmetric sequence is real valued, the use of conjugate-symmetric sequences enables  $2 \times 2$ ,  $4 \times 4$  and  $8 \times 8$  waveform scheduling according to OSTBC for real designs.

Consider the square root raised cosine (SRRC) quarter-band filter impulse response given by

$$p(t) = \frac{\frac{4\alpha t}{T} \cos\left[\frac{(1+\alpha)\pi t}{T}\right] + \sin\left[\frac{(1-\alpha)\pi t}{T}\right]}{\frac{\pi t}{T} \left[1 - \left(\frac{4\alpha t}{T}\right)^2\right]}. \quad (7.70)$$

In order to make four conjugate-symmetric waveforms using the SRRC, we sample this pulse with a sampling interval  $T_s = T/4$ . This gives us

$$p[n] = \frac{\alpha \cos\left[(1+\alpha)\frac{\pi}{4}n\right] + \sin\left[(1-\alpha)\frac{\pi}{4}n\right]}{n\frac{\pi}{4}(1-(n\alpha)^2)}. \quad (7.71)$$

The response with a roll-off factor of  $\alpha = 0.5$  is shown in Figure 7.5a. This waveform is then modulated by  $\exp(j\pi/4)$ ,  $\exp(-j\pi/4)$ ,  $\exp(3j\pi/4)$ , and  $\exp(-3j\pi/4)$  to get four quarter-band SRRC waveforms, given by

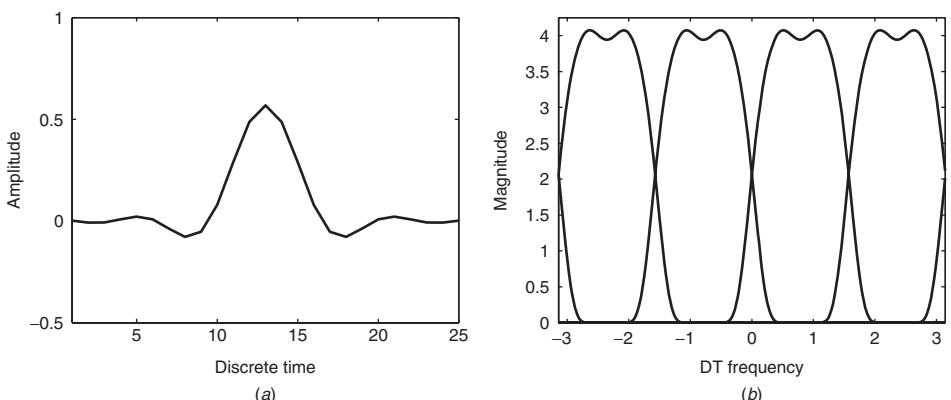
$$s_1 = p[n] \exp\left(\frac{j\pi}{4}\right), \quad (7.72)$$

$$s_2 = p[n] \exp\left(\frac{-j\pi}{4}\right), \quad (7.73)$$

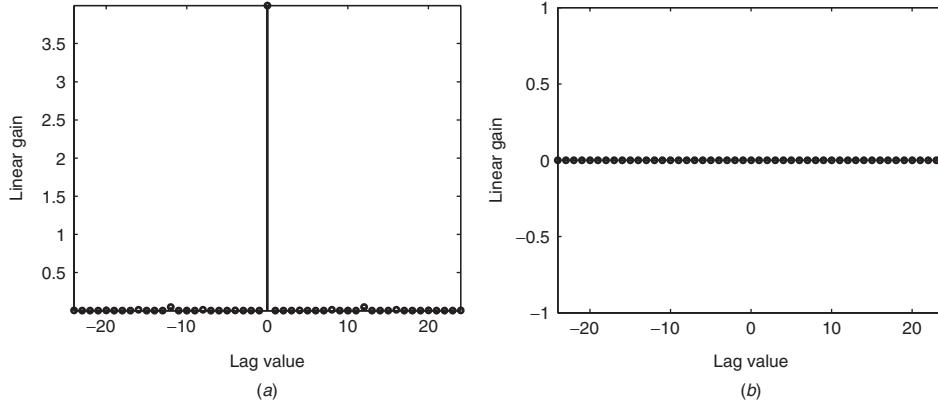
$$s_3 = p[n] \exp\left(\frac{j3\pi}{4}\right), \quad (7.74)$$

$$s_4 = p[n] \exp\left(\frac{-j3\pi}{4}\right). \quad (7.75)$$

The combined frequency response of these waveforms is shown in Figure 7.5b.



**Figure 7.5** (a) Impulse response of the quarter-band filter. (b) Frequency response of the four quarter-band filter waveforms.



**Figure 7.6** (a) Main diagonal sequence is close to a delta function. (b) Off-diagonal sequence is identically zero.

The main and off-diagonal sequences of the Key matrix for these waveforms are given in Figures 7.6a and 7.6b.

We see from these figures that these waveforms are perfectly separable. The small disturbance in the autocorrelation function around the lag values  $\pm 12$  are caused by limiting the infinite SRRC pulse to a finite interval.

#### 7.4.4 Combination of Golay Codes and Half-Band Filters

In the previous section, we created waveforms with quarter-band filters that achieved perfect separation. In this section, we form transmit waveforms through the Kronecker product of Golay codes with half-band SRRC filters. The half-band SRRC waveform is obtained by sampling  $p(t)$  in (7.70) at  $T_s = T/2$ , that is,

$$p[n] = \frac{2\alpha \cos[(1+\alpha)\frac{\pi}{2}n] + \sin[(1-\alpha)\frac{\pi}{2}n]}{n\frac{\pi}{2}[1-(2n\alpha)^2]}. \quad (7.76)$$

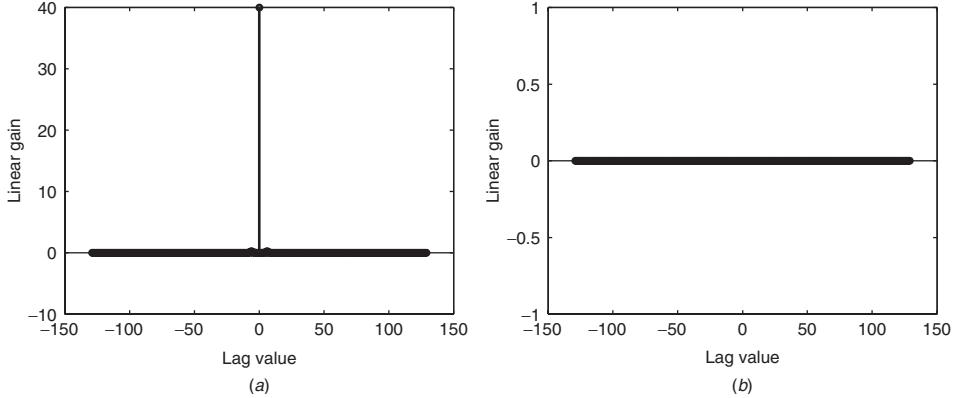
The two half-band SRRC waveforms are then obtained by modulating  $p[n]$  with  $\exp(j\pi/2)$  and  $\exp(-j\pi/2)$ , that is,

$$h_b^{(1)}[n] = p[n] \exp\left(\frac{j\pi}{2}\right), \quad (7.77)$$

$$h_b^{(2)}[n] = p[n] \exp\left(\frac{-j\pi}{2}\right). \quad (7.78)$$

The transmit waveforms are formed by the Kronecker product of these waveforms with the Golay complementary codes  $g_1[n]$  and  $g_2[n]$ , and are given by

$$\begin{aligned} s_1[n] &= g_1[n] \otimes h_b^{(1)}[n], \\ s_2[n] &= g_1[n] \otimes h_b^{(2)}[n], \\ s_3[n] &= g_2[n] \otimes h_b^{(1)}[n], \\ s_4[n] &= g_2[n] \otimes h_b^{(2)}[n], \end{aligned} \quad (7.79)$$



**Figure 7.7** (a) Main diagonal sequence resembles a delta function. (b) Off-diagonal sequence is identically zero.

where  $g_i[n]$  are the Golay codes and  $h_i[n]$  are the half-band filters. The main and off-diagonal sequences of the Key matrix are shown in Figures 7.7a and 7.7b. We see that the waveforms formed from the Kronecker product of Golay sequences with half-band filters possess excellent separation properties.

## 7.5 INTRODUCTION TO DATA-DEPENDENT WAVEFORM DESIGN

So far, we have discussed waveform design without considering the effects of clutter and interference. In this section, we present an overview of the data-dependent waveform design problem. In a physical active sensing environment, we desire to transmit waveforms that depend on the clutter and interference environment, for example, waveforms that are orthogonal to the clutter subspace. This leads to a general waveform design problem that can be expressed as

$$\mathbf{S}_T * \delta[n - D] \mathbf{I} * \mathbf{S}_R \propto \alpha \delta[n - D] \mathbf{I}, \quad (7.80)$$

where  $D$  is the point target delay. Also,

$$\mathbf{S}_T = \begin{bmatrix} s_1[n] & f_2^*[-n] & s_3[n] & f_4^*[-n] \\ -s_2[n] & f_1^*[-n] & -s_4[n] & f_3^*[-n] \\ -s_3[n] & f_4^*[-n] & s_1[n] & -f_2^*[-n] \\ -s_4[n] & -f_3^*[-n] & s_2[n] & f_1^*[-n] \end{bmatrix} \quad (7.81)$$

is the transmitted waveform matrix and

$$\mathbf{S}_R = \begin{bmatrix} f_1^*[-n] & -f_2^*[-n] & -f_3^*[-n] & -f_4^*[-n] \\ s_2[n] & s_1[n] & s_4[n] & -s_3[n] \\ f_3^*[-n] & -f_4^*[-n] & f_1^*[-n] & f_2^*[-n] \\ s_4[n] & s_3[n] & -s_2[n] & s_1[n] \end{bmatrix} \quad (7.82)$$

is the receiver processing matrix. Similar to Section 7.3, we can derive conditions that the waveforms should possess in order to satisfy (7.80), and they are

$$\phi[n] = s_3[n] * f_1^*[-n] + f_4^*[-n] * s_2[n] + s_1[n] * f_3^*[-n] + f_2^*[-n] * s_4[n] = 0, \quad (7.83)$$

$$\theta[n] = s_1[n] * f_1^*[-n] + s_2[n] * f_2^*[-n] + s_3[n] * f_3^*[-n] + s_4[n] * f_4^*[-n] \propto \delta[n]. \quad (7.84)$$

These constraints can be expressed in the form of a matrix equation as

$$\mathbf{S}'\mathbf{F} = \begin{bmatrix} \mathbf{S}_1 & \mathbf{S}_2 & \mathbf{S}_3 & \mathbf{S}_4 \\ -\mathbf{S}_3 & -\mathbf{S}_4 & \mathbf{S}_1 & \mathbf{S}_2 \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{f}_4 \end{bmatrix} = \begin{bmatrix} \delta[n-D] \\ \mathbf{0} \end{bmatrix}, \quad (7.85)$$

where each  $\mathbf{S}_i$  is a  $2N-1 \times N$  convolution matrix in which the rows represent the delayed and flipped waveform sequences. This equation tells us that the waveform correlation should vanish at all delay values except at the true target delay value. Now,  $\mathbf{S}'$  is a  $2(2N-1) \times 4N$  matrix, which means that we have  $2(2N-1)$  equations in  $4N$  unknowns. Since there are more unknowns than the number of equations, this system is underdetermined and there are multiple solutions. Of all the possible solutions, the best solution for our problem is the one that maximizes the correlation between waveforms  $s_i[n]$  and  $f_i[n]$ . Let  $\mathbf{s}_i$  and  $\mathbf{f}_i$  be length  $N$  vectors representing the waveform sequences  $s_i[n]$  and  $f_i[n]$ . Using this notation, the best solution to (7.85) can be represented as

$$\arg \max_{\mathbf{F}} \{\mathbf{S}^T \mathbf{F}\}, \quad (7.86)$$

where

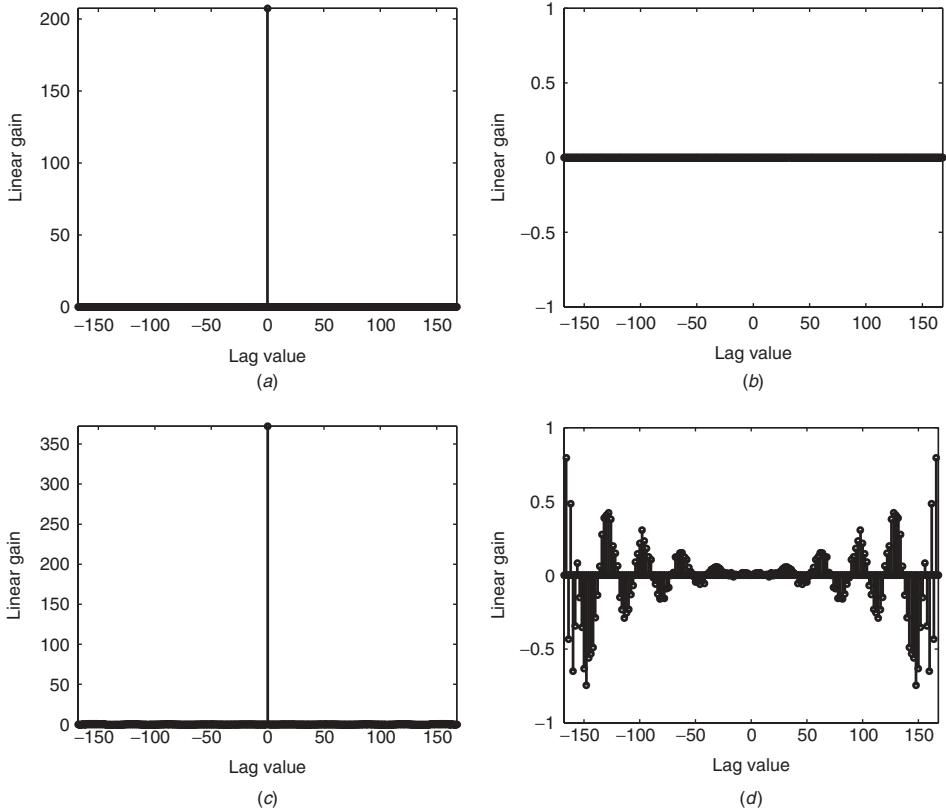
$$\mathbf{S}^T = [\mathbf{s}_1^T \ \mathbf{s}_2^T \ \mathbf{s}_3^T \ \mathbf{s}_4^T], \quad (7.87)$$

$$\mathbf{F} = [\mathbf{f}_1^T \ \mathbf{f}_2^T \ \mathbf{f}_3^T \ \mathbf{f}_4^T]^T. \quad (7.88)$$

In the next section, we apply these concepts to the already known case of waveform design with Barker codes.

### 7.5.1 Reduced Rank Optimization

Consider  $\mathbf{S} = \mathbf{U}\Sigma\mathbf{V}$ , the singular value decomposition (SVD) of  $\mathbf{S}$ , where  $\Sigma$  is the matrix containing the singular values of  $\mathbf{S}$  with  $\leq 4N$  nonzero singular values. If we reduce the rank of this matrix, we are throwing away some degrees of freedom in our solution space. However, by applying a threshold to the singular values and discarding the singular values lower than the threshold allows us to affect a trade-off between the nonzero terms on the main diagonal and off-diagonal and the peak of the main diagonal at the true target delay. To illustrate that, we go back to the Kronecker products of Barker codes, and we form the  $\mathbf{S}$  matrix using the four waveforms we developed based on these codes. The results after applying two different thresholds on the singular values of  $\mathbf{S}$  for this case are shown in Figure 7.8.



**Figure 7.8** (a) Discarding more singular values reduces the peak but removes the unwanted nonzero values. (b) Off-diagonal sequence is zero if we discard more singular values. (c) Discarding fewer singular results in a higher peak compared to (a) and removes the unwanted nonzero values. (d) Off-diagonal sequence is not identically zero if we discard fewer singular values.

We can see from this figure that if we discard a small number of singular values, there is a peak loss of about 1.8 dB (Figure 7.8c) and the off-diagonal terms are slightly higher (Figure 7.8d), but there are no other nonzero terms in the main diagonal. If we discard more singular values, the peak loss increases to about 4.1 dB (Figure 7.8a), but there are no unwanted nonzero terms in either the main diagonal or the off-diagonal (Figure 7.8b). This shows the amount of flexibility we have with the waveform design when we pose this problem in a more general setting.

## 7.6 $3 \times 3$ AND $6 \times 6$ WAVEFORM SCHEDULING

So far, we have focused our attention on antenna arrays with  $2^n$ ,  $n \in \mathbb{Z}^+$  elements. We now look at antenna arrays with three and six elements. Consider the  $3 \times 3$  case. We already know from our previous discussion that the cancelation of the off-diagonal terms occurs in a pairwise fashion. So, in order for the concepts applicable to

antennas with  $2^n$  elements, we extract a  $3 \times 4$  subblock of the transmit matrix and  $4 \times 3$  subblock of the received matrix, that is,

$$\mathbf{S}_T * \delta[n - D] \mathbf{I} * \mathbf{S}_T^* \propto \alpha \delta[n - D] \mathbf{I}, \quad (7.89)$$

where

$$\mathbf{S}_T = \begin{bmatrix} s_1[n] & s_2^*[-n] & s_3[n] & s_4^*[-n] \\ s_2[n] & s_1^*[-n] & s_4[n] & s_3^*[-n] \\ s_3[n] & s_4^*[-n] & s_1[n] & s_2^*[-n] \end{bmatrix} \quad (7.90)$$

is the transmitted waveform matrix and  $\mathbf{S}_T^*$  is defined similar to (7.29). From this, we can see that the conditions for perfect separation and reconstruction that were derived earlier for the  $4 \times 4$  case are applicable to the  $3 \times 3$  case as well.

Proceeding in a similar fashion, we can show that from an  $8 \times 8$  OSTBC real design, we can extract a  $6 \times 8$  subblock from the transmit waveform matrix and  $8 \times 6$  subblock from the received waveform matrix to form a  $6 \times 6$  waveform matrix.

## 7.7 SUMMARY

We have presented an overview of the current and future research trends in diversity waveform design for multichannel radars. We derived some of the conditions for perfect waveform separation and reconstruction at the receiving end. Examples of diversity waveforms for  $2 \times 2$  and  $4 \times 4$  have been provided, and some new waveform designs have been proposed that allow for near perfect separation and reconstruction at the receiver. We saw that Kronecker products of waveform sequences possess some desirable properties that make them suitable for use in radar and active sensing applications. In the end, we introduced the problem of data-dependent waveform design in which waveform designs are adapted according to the clutter and interference properties of the sensing environment.

## REFERENCES

1. N. Levanon, *Radar Principles*, New York: Wiley-interscience, 2001.
2. A. Farina, *Antenna Based Signal Processing Techniques for Radar Systems*, Artech House, 1992.
3. H. Wang, and L. Cai, “On adaptive spatio-temporal processing for airborne surveillance radar systems,” *IEEE Trans. Aerospace Electron. Syst.*, vol. 30, pp. 660–669, 1994.
4. J. Ward, “Cramer-Rao bounds for target Doppler and angle estimation with space-time adaptive processing in radar,” in *Proc. 29th Asilomar Conf. Signals, Syst. Comput.*, 1995, pp. 1198–1202.
5. E. Fishler, A. Haimovich, R. Blum, D. Chizhik, L. Cimini, and R. Valenzuela, “MIMO radar: An idea whose time has come,” in *Proc. IEEE Radar Conference*, 2004, pp. 71–78.
6. H. Schulze and C. Lueders, *Theory and Applications of OFDM and CDMA: Wideband Wireless Communications*, New York: Wiley, 2005.
7. S. D. Howard, A. R. Calderbank, and W. Moran, “A simple polarization diversity scheme for radar detection,” in *Proc. of Second Intl. Conf. on Waveform Diversity and Design*, 2006, pp. 22–27.

8. M. J. E. Golay, "Static multislit spectrometry and its applications to the panoramic display of infrared spectra," *J. Opt. Soc. Am.*, vol. 41, pp. 468–472, 1951.
9. S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1451–1458, 1998.
10. A. H. Roger, and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, 1991.
11. V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1456–1467, 1999.
12. S. M. Kay, *Fundamentals of Statistical Signal Processing*, Vol. 2: *Detection Theory*, Englewood Cliffs, NJ: Prentice Hall, 1993.
13. J. A. Gubner, *Probability and Random Processes for Electrical and Computer Engineers*, Cambridge University Press, 2006.
14. H. V. Trees, *Detection, Estimation and Modulation Theory*, Vol. 3, New York: Wiley, 1971.
15. F. Gini, A. Farina, and M. Greco, "Selected list of references on radar signal processing," *IEEE Trans. Aerospace Electron. Syst.*, vol. 37, pp. 329–359, 2001.

## CHAPTER 8

---

# Acoustic Array Processing for Speech Enhancement

Markus Buck<sup>1</sup>, Eberhard Hänsler<sup>2</sup>, Mohamed Krini<sup>1</sup>, Gerhard Schmidt<sup>1</sup>, and Tobias Wolff<sup>1</sup>

<sup>1</sup>Harman/Becker Automotive Systems, Ulm, Germany

<sup>2</sup>Technische Universität Darmstadt, Darmstadt, Germany

### 8.1 INTRODUCTION

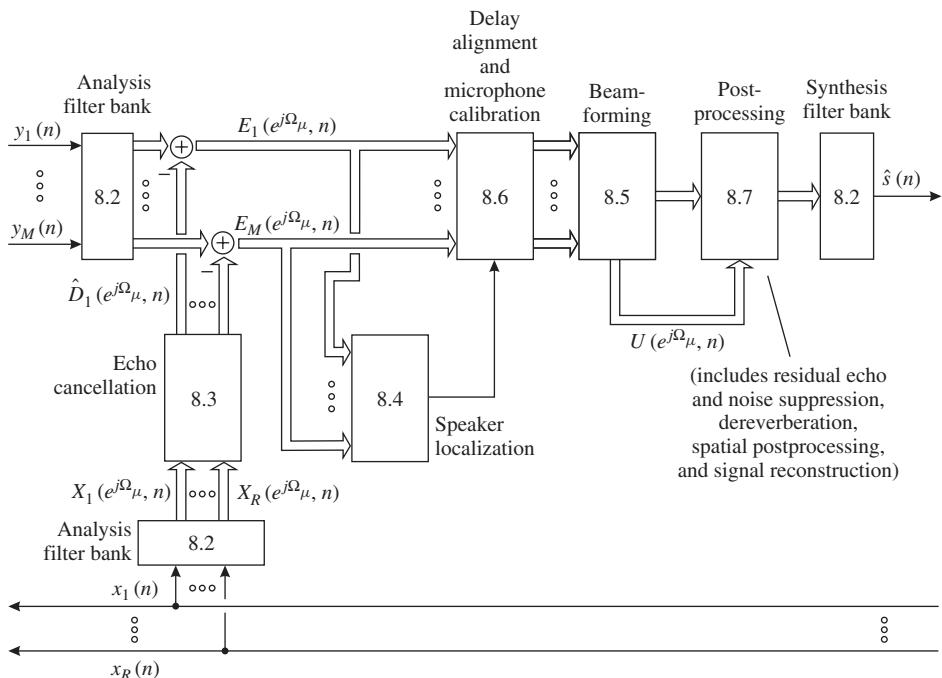
Today, hands-free functionality has become a standard for acoustic front ends of telephone and speech dialog systems. However, these systems are often applied in adverse acoustic environments where ambient noise as well as acoustic couplings of loudspeaker signals superpose the desired speech signal. Furthermore, the level of the desired speech signal is reduced due to the relatively large distance between speaker and microphones. Therefore, the quality of the microphone signals is poor. Methods for controlling noise and echo without degrading speech quality are still subject of intensive research.

The application of array processing has opened new chances in speech and audio signal processing. There are at least two favorable features: reduction of processing power and/or considerable improvement of system performance.

Speech enhancement procedures demand high amounts of computing power even if they work with a single input channel. This attribute stems from the properties of the electroacoustic environment. Splitting the signal into subbands and, for example, adapting cancellation filters for the subband signals reduces the necessary computation power notably [1] and—as an additional benefit—leads to a perceptibly higher performance of the system.

Relying on multiple input channels permits the design of systems with improved quality as compared to single-channel systems. However, it also allows to solve problems such as source localization and tracking or source separation, which cannot be answered if just a single signal is available.

We organize this chapter around the example of a multichannel speech enhancement system (see Fig. 8.1). We discuss solutions to the various tasks such as signal analyzing and synthesizing, echo canceling, speaker localization, delay alignment and microphone calibration, beamforming, residual echo and noise suppression, dereverberation, and signal reconstruction. We also show the interrelations of the corresponding subsystems.



**Figure 8.1** Example of multichannel speech enhancement system. Numbers in the frames refer to the related sections of this chapter.

A look at Figure 8.1 gives an idea of the complexity of a modern speech enhancement system. Only brief discussions are possible in the frame of this handbook. Thus, for further details, we have to refer the reader to the references cited in this chapter.

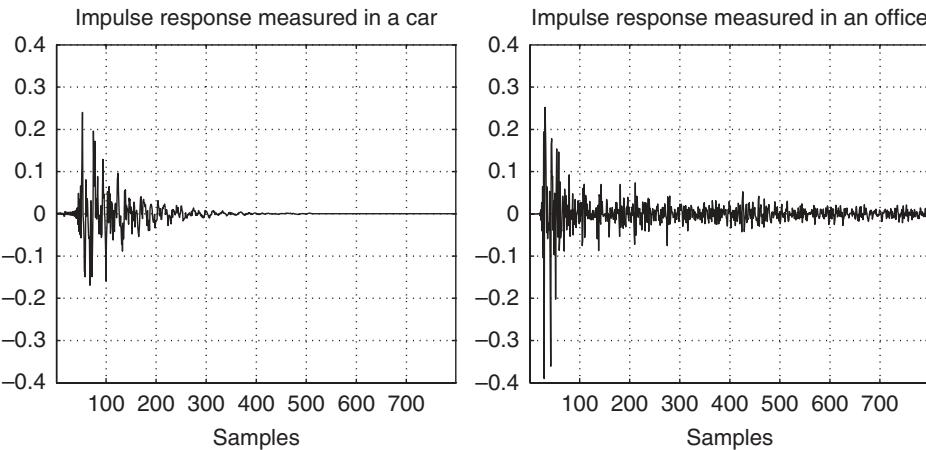
### 8.1.1 Acoustic Environments and Speech Signal Properties

Problems such as acoustic echo cancellation, speech dereverberation, and noise reduction arise whenever loudspeakers and microphones are placed in the same enclosure and the signal sources (speakers) are not close to the microphones. In this case microphones pick up not only the speech signal but also the reverberated signals from the loudspeakers together with environmental noise. Consequently, remote speakers have to listen to their echo, which is delayed by the round trip time of the transmission system.

For low sound pressure and no overload of the converters, loudspeaker–enclosure–microphone (LEM) systems may be modeled with sufficient accuracy as linear systems.

Their impulse responses can be described by a sequence of delayed delta impulses. The delays are associated with the geometrical lengths of related propagation paths. The amplitudes of the impulses depend on the reflection coefficients of the boundaries and on the inverse of the path lengths. As a first-order approximation one can assume that the impulse response decays exponentially.

Figure 8.2 shows the impulse responses of LEM systems measured in a passenger car (left) and in an office (right). They are considerably “longer” than impulse responses of ordinary electrical systems.



**Figure 8.2** Impulse responses measured in a car (left) and in an office (right) (sampling frequency  $f_s = 11\text{ kHz}$ ).

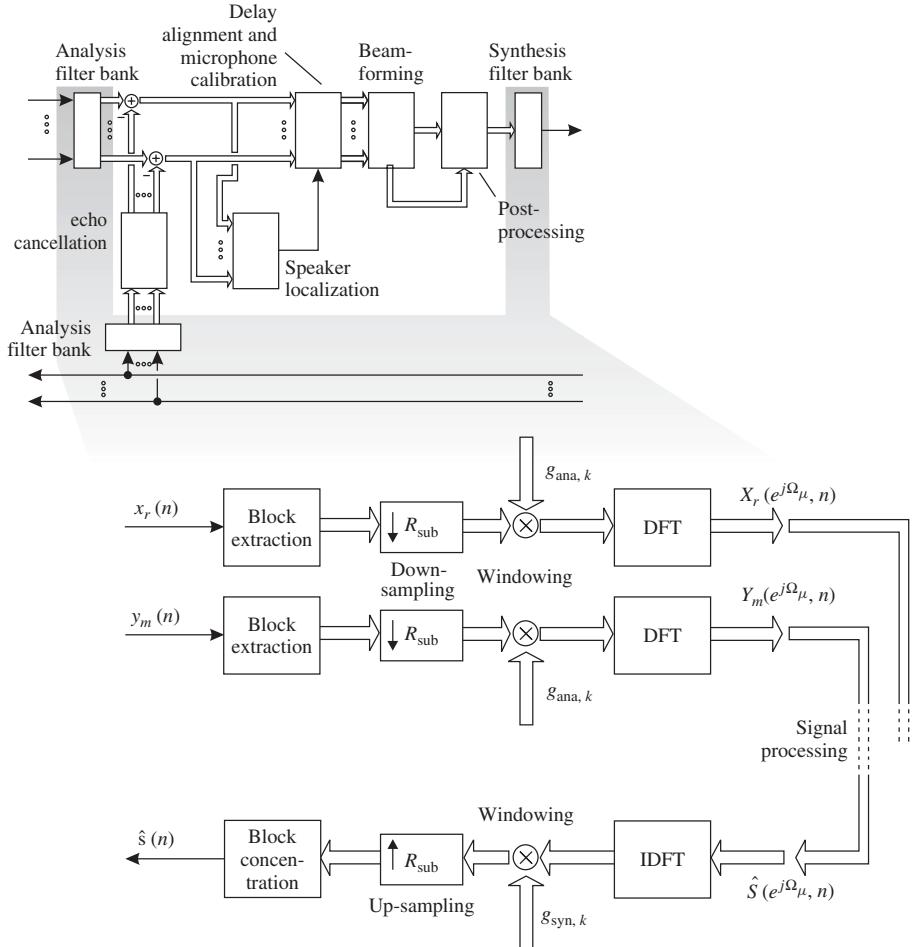
In addition, the impulse responses are highly sensitive to any changes such as the movement of a person within the enclosure (enclosure dislocation). Both properties together explain that high-order adaptive filters are required for echo canceling and noise suppression.

The complexity of speech and audio signal processing is further increased by the fact that most of the time desired signals and noise are available as their sum only and that both occupy the same frequency range. Only sophisticated estimation procedures lead the way out of this difficulty.

Speech signals are characterized by a large bandwidth: The frequency range relevant for signal processing spreads over a broad spectrum reaching from 70 Hz to 4 kHz for telephone applications and to even 10 kHz for broadband systems. Furthermore, the signal-to-noise ratio (SNR) of speech signals varies strongly over time. Whereas this nonstationary characteristic often poses difficulties for signal processing, it also offers the benefit of using speech pauses to analyze the noise.

## 8.2 SIGNAL PROCESSING IN SUBBAND DOMAIN

Processing speech and audio signals in a subband domain offers a good deal of advantages. Therefore, analysis–synthesis schemes are essential parts of processing systems for such signals (see Fig. 8.1). A multitude of structures has been investigated and proposed for individual applications [2, 3]. For speech coding, for example, the preferred choice are analysis–synthesis schemes that conserve the amount of data in the subband or short-term frequency domain they extract from the time domain signal. Perfect reconstruction—the connection of the analysis and the synthesis stage without processing in between—is only a delay—is very important here. Medium or even large aliasing components in each individual subband are tolerable as long as the synthesis stage compensates all aliasing components. For speech processing application, however, large aliasing components are not tolerable as they limit the performance of subband filters, whereas perfect reconstruction is not necessarily required.



**Figure 8.3** Basic building blocks of a DFT-based analysis–synthesis system for loudspeaker signals  $x_r(n)$  and microphone signals  $y_m(n)$ .

The most popular uniform analysis–synthesis scheme applied for speech enhancement periodically performs DFTs and inverse DFTs of overlapping signal segments. The basic structure of such a system is depicted in Figure 8.3. First, a block of the last  $N_{ana}$  samples of the input signals  $y_m(n)$  and  $x_r(n)$ , respectively, is extracted at time  $n$ , with  $m \in \{1, \dots, M\}$  and  $r \in \{1, \dots, R\}$ . Subsequently, appropriate downsampling is applied to obtain the desired rate in the subband domain. Usually, the downsampling factor  $R_{sub}$  is chosen such that successive blocks overlap by 50 or 75%. Each segment is multiplied by a window function  $g_{ana,k}$ . Applying a DFT to the windowed signal vector results in the  $N_{DFT} = N_{ana}$ -point short-term spectrum of the respective block:<sup>1</sup>

$$Y_m(e^{j\Omega_\mu}, n) = \sum_{k=0}^{N_{DFT}-1} y_m(nR_{sub} - k) g_{ana,k} e^{-j\Omega_\mu k}. \quad (8.1)$$

<sup>1</sup>The analysis of the loudspeaker signals  $x_r(n)$  is performed in a similar manner.

For this type of analysis scheme the frequency supporting points  $\Omega_\mu$  are distributed equidistantly over the normalized frequency range:

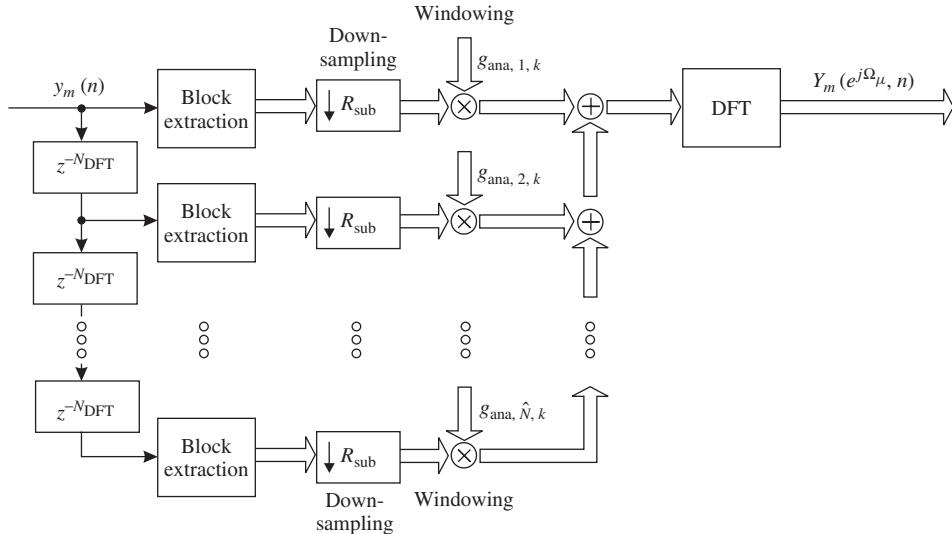
$$\Omega_\mu = \frac{2\pi}{N_{\text{DFT}}} \mu \quad \text{with } \mu \in \{0, \dots, N_{\text{DFT}} - 1\}. \quad (8.2)$$

In order to match the frequency resolution of the filter bank with that of the human auditory system, the frequency resolution can be decreased toward higher frequencies. This can be achieved by replacing the delay elements within the block extraction unit with appropriately designed all-pass filters [4]. However, this feature comes at increased computational cost.

After processing the microphone subband signals  $Y_m(e^{j\Omega_\mu}, n)$  and the reference subband signals  $X_r(e^{j\Omega_\mu}, n)$  with specific signal processing methods, as described in the next sections, enhanced subband signals  $\widehat{S}(e^{j\Omega_\mu}, n)$  are obtained. For synthesizing the output signal, first an inverse DFT is performed. The resulting vector is weighted with the synthesis window function  $g_{\text{syn}, k}$  and overlapping vectors are added (*overlap-add method* [2]) to compute the output signal.

Whenever time-variant processing—such as noise suppression—is applied, the usage of synthesis windowing has the positive effect of reducing distortions at frame boundaries [5], especially when the spectral attenuation has changed significantly between consecutive blocks.

If synthesis windowing should be applied and, at the same time, the frequency selectivity of the analysis should be enhanced, the DFT and its inverse can be extended to a more general structure such as *polyphase filter banks* (see Fig. 8.4).<sup>2</sup> To achieve this, preceding weighted blocks need to be added before the DFT is performed.



**Figure 8.4** Polyphase-based analysis system. Details about synthesis schemes can be found, e.g., in [1].

<sup>2</sup>Note that the basic analysis–synthesis scheme described before is a special case of a polyphase filter bank where just one polyphase component is utilized.

Furthermore, the window function  $g_{\text{ana},k}$  has to be extended by a so-called *prototype low-pass filter* that covers the current as well as previous frames. Using more than one frame for the current spectral analysis leads to a better frequency resolution and lower in-band aliasing properties. However, the inherent disadvantage is that the impulse response of the prototype filter is much longer than the DFT order  $N_{\text{ana}} = \tilde{N} N_{\text{DFT}}$  (with  $\tilde{N}$  being the number of blocks to be added). This results in a reduced time resolution. If the input signal changes its spectral characteristics within the time corresponding to the memory size of the analysis stage of the filter bank, the short-time spectrum is smoothed. For very long prototype filters, that is, the ratio of the filter length and the sampling frequency  $N_{\text{ana}}/f_s$  is larger than about 150 ms, the spectral smoothing results in so-called *postechoes*. They appear at sudden changes of the coefficients of the residual echo and noise suppression filter. Due to a long prototype filter the synthesis stage has a long memory, too. Filling this memory with large amplitudes during a speech sequence and recording very small amplitudes afterwards leads to artifacts during the output of the stored large samples. Anyhow, if the length of the prototype low-pass filter is chosen not too large, polyphase filter banks are a good candidate for analysis–synthesis systems for speech enhancement applications.

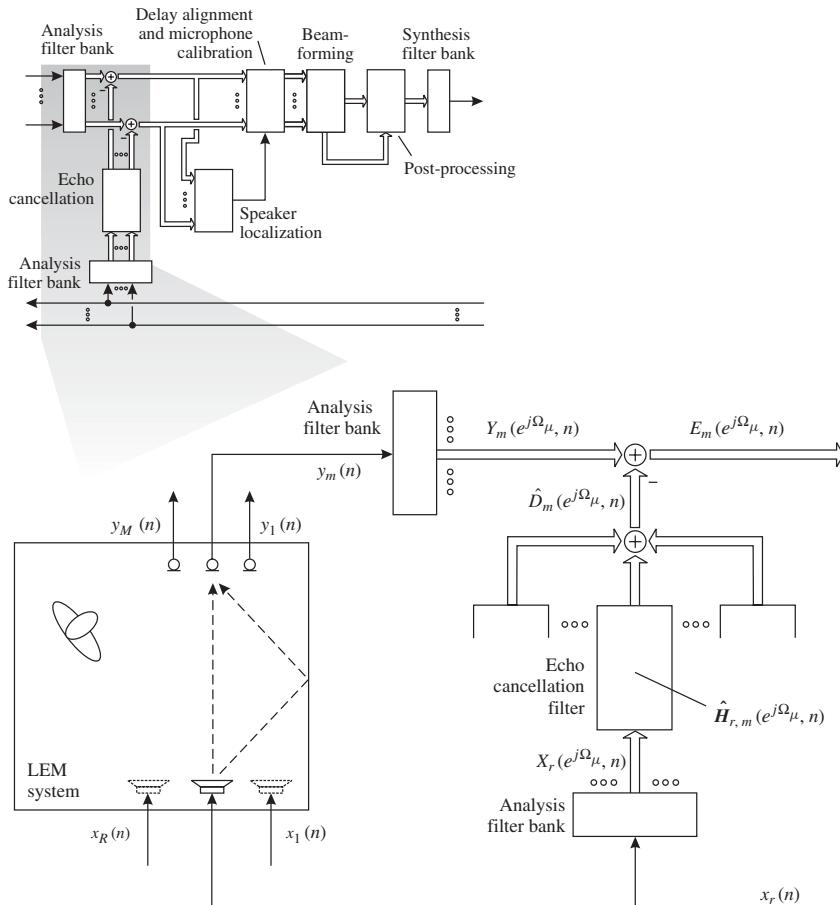
The main advantage of processing in the subband domain (compared to standard time-domain processing) is the reduction of the computational complexity. Depending on the size of the involved filters (e.g., for echo cancellation), the DFT-order  $N_{\text{DFT}}$ , and the subsampling factor  $R_{\text{sub}}$  reductions of about 50–90% can be achieved. However, the required memory usually increases by a factor of 2–4. A further advantage is the orthogonality of the individual subband signals. This allows, for example, for faster filter convergence whenever the time-domain signal is not white (which is definitely true for speech signals). The main drawback of analysis–synthesis systems is the delay that is introduced in the signal path. Especially for hands-free applications this delay has a negative effect on the overall communication quality and thus limitations have been specified (e.g., for hands-free processing in mobile phones a maximum additional delay for signal processing of 39 ms is allowed [6]).

For the rest of this chapter we will use a filter bank with  $N_{\text{DFT}} = 256$  subbands within all examples. For the window function a Hann [7] window of length  $N_{\text{ana}} = N_{\text{DFT}}$  is utilized. Neighboring frames overlap by 75%, resulting from a subsampling factor  $R_{\text{sub}} = 64$ . The time-domain input signals are sampled at  $f_s = 11,025$  Hz, leading to a filter bank delay of about 23 ms.

### 8.3 MULTICHANNEL ECHO CANCELLATION

To avoid the problems of annoying echoes and howling of a communication loop, a set of subband adaptive filters can be placed parallel to the LEM system (see Fig. 8.5). If one succeeds in matching the impulse responses of the filters exactly with the subband impulse responses of the LEM system, the signals  $X_r(e^{j\Omega_\mu}, n)$  and  $E_m(e^{j\Omega_\mu}, n)$  are perfectly decoupled without any disturbing effects to the users of the electroacoustic system.

Since, in real applications, a perfect match (over all times and all situations) cannot be achieved, the remaining subband error signals  $E_m(e^{j\Omega_\mu}, n)$  still contain echo components. For further reduction of these signals an additional filter for residual echo suppression is applied in the transmitting path—as described in Section 8.7.



**Figure 8.5** Basic building blocks of a multichannel subband acoustic echo cancellation system.

The complexity of echo cancellation schemes grows with the number of microphones  $M$  and the number of reference channels  $R$ . For multimicrophone–multireference systems this leads to a huge computational complexity. A reduction can be achieved if echo cancellation is efficiently combined with beamforming. Section 8.5.4 will show details about such combinations.

### 8.3.1 Adaptation Algorithms

The majority of implementations of acoustic echo canceling systems use the *normalized least-mean-square* (NLMS) algorithm to update the adaptive filters. This gradient-type algorithm minimizes the mean-square error [8]. The update equation is given by

$$\begin{aligned} \hat{\mathbf{H}}_{r,m}(e^{j\Omega_\mu}, n+1) &= \hat{\mathbf{H}}_{r,m}(e^{j\Omega_\mu}, n) \\ &+ \mu_{r,m}(e^{j\Omega_\mu}, n) \frac{\mathbf{X}_r(e^{j\Omega_\mu}, n) \mathbf{E}_m^*(e^{j\Omega_\mu}, n)}{\sum_{r=1}^R \mathbf{X}_r^H(e^{j\Omega_\mu}, n) \mathbf{X}_r(e^{j\Omega_\mu}, n)}. \end{aligned} \quad (8.3)$$

The term in the denominator of the update part in Eq. (8.3) represents a normalization according to the energy of all input vectors:

$$\mathbf{X}_r(e^{j\Omega_\mu}, n) = [X_r(e^{j\Omega_\mu}, n), \dots, X_r(e^{j\Omega_\mu}, n - N_{ec} + 1)]^T, \quad (8.4)$$

with  $N_{ec}$  denoting the length of the adaptive filters. This is necessary due to the high nonstationarity of speech signals. The step size of the update is controlled by the factor  $\mu_{r,m}(e^{j\Omega_\mu}, n)$ . The algorithm is stable (in the mean-square sense) for  $0 < \mu_{r,m}(e^{j\Omega_\mu}, n) < 2$ . Reducing the step size is necessary to prevent divergence of the filter coefficients in case of strong local speech signals and/or local background noise that are part of the microphone outputs.

To perform the filter update the subband error signals  $E_m(e^{j\Omega_\mu}, n)$  are computed by subtracting the estimated echo components  $\widehat{D}_m(e^{j\Omega_\mu}, n)$  from the microphone signals  $Y_m(e^{j\Omega_\mu}, n)$ :

$$E_m(e^{j\Omega_\mu}, n) = Y_m(e^{j\Omega_\mu}, n) - \underbrace{\sum_{r=1}^R \widehat{\mathbf{H}}_{r,m}^H(e^{j\Omega_\mu}, n) \mathbf{X}_r(e^{j\Omega_\mu}, n)}_{\widehat{D}_m(e^{j\Omega_\mu}, n)}. \quad (8.5)$$

The NLMS algorithm has no memory, that is, it uses only error signals that are available at the time of the update. The speed of convergence can be improved by extending the NLMS algorithm to the so-called *affine projection* algorithm [9]. This means to consider previous blocks in the update equation and comes at only slightly increased cost. The so-called *recursive least-squares* (RLS) algorithm used occasionally for updating echo cancellation filters minimizes the weighted sum of all previous squared error signals. In order to emphasize recent errors, often an exponentially decaying weighting is applied. For further details on adaptive algorithms we refer the interested reader to, for example, [8] or [10].

### 8.3.2 Adaptation Control

Independent of the specific algorithm used, the update of the coefficients of the echo cancellation filter strongly depends on the subband error  $E_m(e^{j\Omega_\mu}, n)$ . This signal is composed of the *undisturbed* subband error  $E_{u,m}(e^{j\Omega_\mu}, n)$  and the subband signals originating from local interferences such as background noise  $B_m(e^{j\Omega_\mu}, n)$  or local speech  $S_m(e^{j\Omega_\mu}, n)$ :

$$\begin{aligned} E_m(e^{j\Omega_\mu}, n) &= Y_m(e^{j\Omega_\mu}, n) - \sum_{r=1}^R \widehat{D}_{r,m}(e^{j\Omega_\mu}, n) \\ &= S_m(e^{j\Omega_\mu}, n) + B_m(e^{j\Omega_\mu}, n) + \underbrace{\sum_{r=1}^R D_{r,m}(e^{j\Omega_\mu}, n) - \widehat{D}_{r,m}(e^{j\Omega_\mu}, n)}_{E_{u,r,m}(e^{j\Omega_\mu}, n)}, \end{aligned} \quad (8.6)$$

with  $D_{r,m}(e^{j\Omega_\mu}, n)$  being the echo components in microphone  $m$  resulting from the reference channel  $r$ . Only  $E_{u,r,m}(e^{j\Omega_\mu}, n)$  steers the coefficients of the adaptive filter  $\widehat{\mathbf{H}}_{rm}(e^{j\Omega_\mu}, n + 1)$  toward their optimal value. Strictly speaking, this is only true if the

reference signals are mutually uncorrelated. The step-size factor  $\mu_{r,m}(e^{j\Omega_\mu}, n)$  is used to control the robustness of the algorithm. If the filter has converged, the error signal  $E_m(e^{j\Omega_\mu}, n)$  has decreased to a certain value. If suddenly the amplitude of  $E_m(e^{j\Omega_\mu}, n)$  is increased, it may have two reasons that require different actions:

- First, a local speaker became active or a local noise started. In this case the step size has to be reduced to prevent losing the degree of convergence achieved before.
- Second, the impulse response of the LEM system has changed, for example, by the movement of the local talker. Now, the step size has to be increased to its maximal possible value in order to adapt the echo cancellation filter to the new impulse response as fast as possible.

To differentiate those situations the undisturbed error  $E_{u,r,m}(e^{j\Omega_\mu}, n)$  needs to be known in order to control the adaptation process. Another leading point should be mentioned here: The first situation requires immediate action. In the second case, a delayed action causes an audible echo but no divergence of the adaptive filter.

In the literature several schemes have been proposed to deal with the undisturbed error. A popular scheme is to approximate the so-called *pseudo-optimal step size*

$$\mu_{\text{opt},r,m}(e^{j\Omega_\mu}, n) = \frac{E\left\{ |E_{u,r,m}(e^{j\Omega_\mu}, n)|^2 \right\}}{E\left\{ |E_m(e^{j\Omega_\mu}, n)|^2 \right\}}, \quad (8.7)$$

either in a binary (on/off control) or in a continuous manner. Details about those schemes can be found, for example, in [1, 11].

### 8.3.3 Special Problems of Multichannel Systems

For applications such as home entertainment, computer games, or advanced teleconferencing systems, multichannel surround sound is often employed. If the sound source is, for instance, a DVD player with six independent output channels that should be controlled by voice, multichannel echo cancellation is a key technology.

For most types of signals the  $R$  reference channels are sufficiently decorrelated, and channel-independent processing as described in the previous sections works fine. However, if a single speech signal (or a monomusic signal) is played back, the individual reference channels are strongly correlated and no unique solution for the optimization process exist. In this case the resulting misalignments between the adaptive filters and the true impulse responses of the LEM system can be much worse as compared to the uncorrelated case. Whenever the interchannel correlation changes, the filters have to readapt and echoes will be audible during this period.

To reduce the correlation, the reference signals may be processed in a nonlinear and/or time-variant manner before they are played back by the loudspeakers and used as reference signals for the echo cancellation filters. Several approaches such as time-variant all-pass filters [12, 13], insertion of noise [14] in a psychoacoustic motivated manner, or adding a weighted half-way rectified version of the signal to itself [15] have been proposed as well. The degree of nonlinearity or time variability should be controlled in dependence on the convergence state of the filter and the (mutual) correlation properties of the input signals.

## 8.4 SPEAKER LOCALIZATION

Speaker localization is a speech and audio signal processing task that can only be solved in multichannel environments. Main requirements of the solutions are precision, speed, and robustness. Consider, for instance, the problem of steering a beamformer to a moving speaker. For this application the accuracy of the estimated speaker position has to be at least in the range of the beamwidth to prevent signal distortions. For the same reason, the respective algorithm ought to produce reliable results based on very few data samples. In practical scenarios the problem is furthermore hampered by noise and reverberation effects. Especially the latter poses a serious problem for many of the known speaker localization techniques.

The generic acoustic source localization problem, however, addresses the problem of finding the positions of multiple sound sources in the three-dimensional space, whereas the number of sources is generally unknown. To this end, the data from a number of microphones are usually processed in two steps. The first step typically consists of processing a number of direction of arrival (DoA) estimates from several microphone arrays that are set up at different locations. In the second processing step, the individual results are combined to obtain the estimated source location. This second step may be accomplished using triangulation or spherical interpolation, which are both computationally demanding. A computationally efficient spherical interpolation technique has been proposed in [16]. In this section we will focus on some basic notions of the first stage and highlight the most promising current techniques. Some recent developments are described as well.

### 8.4.1 Basic Concepts in DoA Estimation

A very intuitive way to find a talker location is to steer a beamformer to various directions and to choose the angle that maximizes the steered response power. Though this approach is in principle capable of localizing multiple speakers, it lacks accuracy especially in the presence of reverberation, and it is computationally rather expensive [17].

Another class of approaches is based on eigendecomposition of the spatiotemporal covariance matrix of microphone signals. The covariance matrix is Hermitian, and can, therefore, be decomposed into orthogonal eigenvectors with real-valued eigenvalues. The eigenvectors that correspond to the largest eigenvalues form an orthogonal basis of a subspace that is called the *signal subspace*, whereas the remaining set of eigenvectors spans the *noise subspace*. Those orthogonal subspaces can be exploited in different ways. The popular multiple signal classification (MUSIC) algorithm [18], for instance, characterizes vectors that lie in the signal subspace by the norm of their image in the noise subspace, which is ideally zero. The MUSIC method has been extended to general geometries of the array, and frequency-domain implementations have been derived to make it applicable to broadband signals such as speech. This, however, comes at the expense of highly increased computational load. Another method that uses the eigen-decomposition of the spatiotemporal covariance matrix is the MIN-NORM method. It is characterized by its robust angular spectrum [19]. Methods based on eigenanalyses have the advantage of resolving multiple sources. Practical problems arise because the covariance matrix is generally not known and thus has to be estimated. Estimation should only be performed when the source positions are fixed, and all signals can be assumed to be stationary to allow for temporal averaging. Practically, and especially for speech signals, these conditions are, however, hardly met. Additionally, these methods

have been reported to be sensitive against inaccurate sensor positions, deviations from the plane-wave assumption, and reverberation.

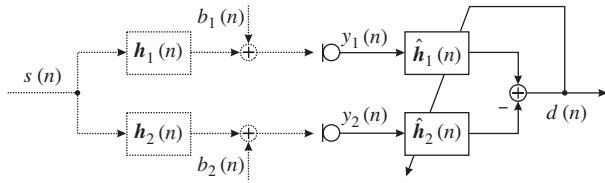
The most widely used methods are those based on time-delay estimation. These approaches attempt to estimate the source location based on the relative time shift between at least two microphone signals, which is translated into an estimate of the spatial angle of incidence. The most prominent time-delay measure is finding the time shift  $\hat{\tau}$  that maximizes the generalized cross-correlation (GCC) function:

$$r_{y_1 y_2}(\tau) = \int_{-\infty}^{\infty} \Psi(f) S_{y_1 y_2}(f) e^{-j2\pi f \tau} df, \quad (8.8)$$

where  $S_{y_1 y_2}(f)$  denotes the cross-power spectral density of the respective time signals  $y_1(t)$  and  $y_2(t)$ . A number of GCC methods are available that differ with respect to the weighting function  $\Psi(f)$ , which is designed to meet specific requirements. For  $\Psi(f) = 1$ , for instance, the pure cross-correlation function results. The GCC method has become so popular because of its simplicity, which is the most significant advantage over the two methods mentioned above. The main problems though are noise and reverberation. The impact of noise has been studied intensively, and optimal maximum-likelihood solutions for  $\Psi(f)$  have been derived and applied successfully [17]. The reverberation problem, however, has been tackled less successful. An approach that has drawn considerable attention is the phase-transform (PHAT) GCC, which strives to mitigate adverse reverberation peaks in the GCC function by putting equal emphasis on all frequencies:  $\Psi(f) = |S_{y_1 y_2}(f)|^{-1}$ . The direct sound peak is thus pronounced, whereas the sensitivity against noise is increased. Further details about GCC methods can be found in [20]. Approaches that attempt to deconvolve the reverberation effects by means of cepstral filtering have also been studied, but neither do they meet the common speed requirements, nor are they effective with speech signals [17]. In contrast to beamformer steering and eigenanalysis methods, the time-delay estimation approaches can only resolve a single sound source. Nevertheless, the vast majority of practical systems makes use of this principle. An attractive remedy to the reverberation problem is the use of adaptive filters.

#### 8.4.2 Adaptive-Filter-Based Methods

The relative time shift between two microphone signals may also be found by comparison of the temporal difference between the maxima of the room impulse responses. These maxima are associated with sound waves propagating on the direct path between the source and each of the microphones. The difference  $\hat{\tau}$  hardly depends on the reverberation, since reverberation effects are represented by the rest of the impulse responses. The goal is thus to (blindly) estimate the impulse responses. This can be accomplished using adaptive FIR filters  $\hat{h}_m(n)$  [1]. The basic two-channel structure is depicted in Figure 8.6. In case that  $b_m(n) = 0$  and excluding the trivial solution  $\hat{h}_1(n) = \hat{h}_2(n) = \mathbf{0}$ , it becomes obvious that the output  $d(n)$  will be zero if  $\hat{h}_1(n) = \mathbf{h}_2(n)$  and vice versa, because for linear systems the convolution is commutative. Hence, minimizing the power of the difference  $d(n)$  between both filter outputs, while avoiding the trivial solution, means estimating the actual room impulse responses  $\mathbf{h}_m(n)$ . The impulse responses can be found using a constraint LMS algorithm. Practically, the search for



**Figure 8.6** Basic structure of a source localization system using adaptive filters.

the optimum filters is not trivial due to the nonstationary characteristics of speech signals and noise. In [21] a time-domain implementation has been presented that outperforms the PHAT-GCC method in the presence of moderate reverberation. It may be argued that this approach requires a significant number of iterations to converge, which makes it infeasible for real-time steering of beamformers. However, complete convergence is not essential since only the positions of the direct sound peaks are of interest. An efficient frequency-domain implementation has been described in [22] that was reported to work reasonably well for real-time steering of a beamformer. As for practical implementations of the GCC method, the spatial resolution depends on the sampling frequency and the microphone distance. Often oversampling is used to interpolate the time series prior to maximum search.

An efficient speaker localization system based on the same principle that works within a subband processing framework has recently been proposed by some of the authors [23]. The proposed localization measure evaluates only a subset of the subband filter coefficients, and the spatial resolution can be chosen as desired for the respective application. Another advantage of the described method is the scalability of the length of the adaptive filters without the need of changing the surrounding signal processing framework. Apart from the fast Fourier transforms (FFTs) needed to compute the input spectra, no additional FFT is required.

## 8.5 BEAMFORMING

Beamforming for speech signal acquisition mostly has to be done in challenging environments. As already pointed out in Section 8.1.1 background noise may deteriorate the quality of the microphone signals, and the desired signal may be reverberated due to room acoustics. Furthermore, the frequency range that is relevant for signal processing spreads over a broad spectrum. On the other hand for consumer products often only a limited number of microphones is available because of hardware costs or because of limited space for integration (e.g., personal navigation devices, notebooks).

### 8.5.1 Basics

The task of a beamformer is to selectively pick up signals impinging from a predefined direction, the so-called *steering direction*. If the position of the speaker is not known in advance, speaker localization as discussed in Section 8.4 has to be employed. Most often the microphones are arranged linearly as *broadside* array where the steering direction is (at least approximately) perpendicular to the array axis. But also *endfire* arrays (the steering direction lies in parallel to the array axis) or multidimensional arrangements are known. Despite the large signal bandwidth, uniform linear arrays are most common for speech applications. One reason for this might be the comprehensive

theory of spatial sampling [24]. In order to avoid spatial aliasing, the microphone spacing  $d_{\text{mic}}$  should be smaller than half of the smallest wavelength  $\lambda_{\text{min}}$ , which depends on the maximum frequency  $f_{\text{max}}$  and the speed of sound  $c$ :

$$d_{\text{mic}} < \frac{\lambda_{\text{min}}}{2} = \frac{c}{2f_{\text{max}}} . \quad (8.9)$$

For example, for a maximum frequency of  $f_{\text{max}} = 5.5 \text{ kHz}$  a spacing of  $d_{\text{mic}} = 3 \text{ cm}$  should not be exceeded. Anyhow the microphone distance is often chosen larger than allowed in order to achieve a higher directionality in the lower frequency range.

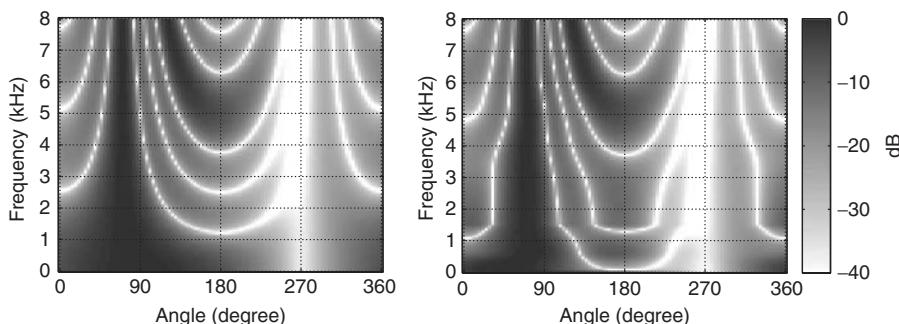
### 8.5.2 Beamformer Structures

A sound wave from a specific spatial direction  $\mathbf{r}$  reaches the different microphones of the array at different times. The relative time difference between the respective signals observed at microphone  $m$  and microphone  $n$  is denoted with  $\tau_{m,n}(\mathbf{r})$ . At the normalized frequency  $\Omega_\mu$  the delay with respect to microphone 1 corresponds to a factor  $P_m(e^{j\Omega_\mu}, \mathbf{r}) = e^{-j\Omega_\mu f_S \tau_{m,1}(\mathbf{r})}$ .

**8.5.2.1 Delay-and-Sum Beamformer** In order to pickup signals from the steering direction  $\mathbf{r}_s$ , the corresponding relative time delays have to be compensated for. If the bandwidths of the frequency subbands are narrow, the time-delay compensation can be realized by a phase shift in each subband. Thus, in the subband framework described in Section 8.2 a delay-and-sum beamformer can be accomplished without the need of fractional delay filters as it is required in the time domain:

$$A_{\text{ds}}(e^{j\Omega_\mu}, n) = \frac{1}{M} \sum_{m=1}^M P_m^*(e^{j\Omega_\mu}, \mathbf{r}_s) E_m(e^{j\Omega_\mu}, n) = \frac{1}{M} \sum_{m=1}^M \tilde{E}_m(e^{j\Omega_\mu}, n) . \quad (8.10)$$

The time-delayed signals are denoted with  $\tilde{E}_m(e^{j\Omega_\mu}, n)$ . The directional characteristics of the delay-and-sum beamformer is strongly dependent on frequency as shown in the example given in Figure 8.7. For low frequencies only a poor directionality can be



**Figure 8.7** Beam pattern for a four-element linear broadside array of  $d_{\text{mic}} = 5 \text{ cm}$  for a delay-and-sum beamformer on the left-hand side and for a constraint minimum variance distortionless response beamformer with  $K = 10$  on the right. The steering direction includes an angle of  $70^\circ$  with the array axis. The microphones themselves have cardioid patterns and are oriented toward  $90^\circ$ . The directional patterns of beamformer and microphones superpose.

achieved, whereas at high frequencies spatial aliasing may even cause so-called *grating lobes* [1, 14, 25].

**8.5.2.2 Filter-and-Sum Beamformer** In a more general structure filters are applied to the input signals before the summation. To obtain a compact notation, the signals are gathered in vectors:

$$\mathbf{W}(e^{j\Omega_\mu}, n) = [\mathbf{W}_1^T(e^{j\Omega_\mu}, n), \dots, \mathbf{W}_M^T(e^{j\Omega_\mu}, n)]^T, \quad (8.11)$$

$$\mathbf{W}_m(e^{j\Omega_\mu}, n) = [W_{m,0}(e^{j\Omega_\mu}, n), \dots, W_{m,N_{\text{bf}}-1}(e^{j\Omega_\mu}, n)]^T, \quad (8.12)$$

$$\tilde{\mathbf{E}}(e^{j\Omega_\mu}, n) = [\tilde{\mathbf{E}}_1^T(e^{j\Omega_\mu}, n), \dots, \tilde{\mathbf{E}}_M^T(e^{j\Omega_\mu}, n)]^T, \quad (8.13)$$

$$\tilde{\mathbf{E}}_m(e^{j\Omega_\mu}, n) = [\tilde{E}_{m,0}(e^{j\Omega_\mu}, n), \dots, \tilde{E}_{m,N_{\text{bf}}-1}(e^{j\Omega_\mu}, n)]^T, \quad (8.14)$$

with  $N_{\text{bf}}$  being the length of the beamformer filters. The filter-and-sum operation can then be written as an inner product:

$$A(e^{j\Omega_\mu}, n) = \mathbf{W}^H(e^{j\Omega_\mu}, n) \tilde{\mathbf{E}}(e^{j\Omega_\mu}, n). \quad (8.15)$$

The beamformer output power can be expressed by a quadratic form:

$$E \left\{ |A(e^{j\Omega_\mu}, n)|^2 \right\} = \mathbf{W}^H(e^{j\Omega_\mu}, n) \mathbf{R}_{\tilde{\mathbf{E}}\tilde{\mathbf{E}}}(\Omega_\mu, n) \mathbf{W}(e^{j\Omega_\mu}, n) \quad (8.16)$$

with the correlation matrix

$$\mathbf{R}_{\tilde{\mathbf{E}}\tilde{\mathbf{E}}}(\Omega_\mu, n) = E \{ \tilde{\mathbf{E}}(e^{j\Omega_\mu}, n) \tilde{\mathbf{E}}^H(e^{j\Omega_\mu}, n) \}. \quad (8.17)$$

For acoustic beamforming the filters usually are designed according to the *linearly constrained minimum variance* (LCMV) criterion. Here, the output power of the beamformer is minimized according to

$$\mathbf{W}_{\text{LCMV}}(e^{j\Omega_\mu}, n) = \underset{\mathbf{W}(e^{j\Omega_\mu}, n)}{\operatorname{argmin}} \left\{ \mathbf{W}^H(e^{j\Omega_\mu}, n) \mathbf{R}_{\tilde{\mathbf{E}}\tilde{\mathbf{E}}}(\Omega_\mu, n) \mathbf{W}(e^{j\Omega_\mu}, n) \right\} \quad (8.18)$$

subject to the constraint

$$\mathbf{C}^H \mathbf{W}(e^{j\Omega_\mu}, n) = \mathbf{f}. \quad (8.19)$$

The constraint matrix  $\mathbf{C}$  has dimension  $MN_{\text{bf}} \times N_{\text{bf}}N_{\text{con}}$  and the response vector  $\mathbf{f}$  has dimension  $N_{\text{bf}}N_{\text{con}} \times 1$ , where  $N_{\text{con}}$  is the number of linear constraints. The optimal solution can be obtained by using Lagrange multipliers [8, 24]:

$$\mathbf{W}_{\text{LCMV}}(e^{j\Omega_\mu}, n) = \mathbf{R}_{\tilde{\mathbf{E}}\tilde{\mathbf{E}}}^{-1}(\Omega_\mu, n) \mathbf{C} \left( \mathbf{C}^H \mathbf{R}_{\tilde{\mathbf{E}}\tilde{\mathbf{E}}}^{-1}(\Omega_\mu, n) \mathbf{C} \right)^{-1} \mathbf{f}. \quad (8.20)$$

Mostly only one directional constraint is applied, which ensures an undistorted response for the steering direction  $\mathbf{r}_s$ . This special case is known as *minimum variance distortionless response* (MVDR) criterion [26]. If furthermore the subband domain  $N_{\text{bf}} = 1$  is chosen, the constraint matrix  $\mathbf{C}$  becomes a vector consisting of the phase shifts  $P_m(e^{j\Omega_\mu}, \mathbf{r}_s)$  corresponding to the steering direction and  $\mathbf{f}$  is equal to 1. In

order to increase the robustness against mismatched conditions, a quadratic constraint is added to the directional constraint of Eq. (8.19). Thus, the amplification by the beamformer filters is limited:

$$\|\mathbf{W}(e^{j\Omega_\mu}, n)\|^2 \leq K. \quad (8.21)$$

A solution to this constraint optimization is obtained by regularization of the spatial correlation matrix [8, 27]:

$$\mathbf{W}_{\text{MVDR}}(e^{j\Omega_\mu}, n, \varepsilon) = \frac{[\mathbf{R}_{\tilde{E}\tilde{E}}(\Omega_\mu, n) + \varepsilon \mathbf{I}]^{-1} \mathbf{C}}{\mathbf{C}^H [\mathbf{R}_{\tilde{E}\tilde{E}}(\Omega_\mu, n) + \varepsilon \mathbf{I}]^{-1} \mathbf{C}}. \quad (8.22)$$

For the regularization parameter  $\varepsilon$  no direct solution exists. However, it has been shown that the array gain increases monotonically with an increasing value of  $\varepsilon$  [28]. The optimal value can thus be calculated iteratively.

Up to now no restrictions have been made for the microphone arrangement and the sound field. Therefore, the theory even holds for near-field acoustics.

**8.5.2.3 Adaptive Beamforming** An adaptive solution to the LCMV optimization problem has been given in [29]. This solution is based on a steepest descent algorithm where the constraints are applied after each adaptation step in order to ensure that the algorithm is robust against error accumulation. In [30] this constrained algorithm has been transformed into an unconstrained one, where the constraint is not implemented in the algorithm but in the structure of the beamformer. Associated to the constraint of Eq. (8.19) a projection matrix  $\mathbf{P}_C = \mathbf{C}[\mathbf{C}^H \mathbf{C}]^{-1} \mathbf{C}^H$  is used for decomposing each filter vector into two orthogonal components. One component that is equal for all weight vectors and fulfills the constraint is the *quiescent weight vector*  $\mathbf{W}_q(e^{j\Omega_\mu}) = \mathbf{P}_C \mathbf{W}_{\text{LCMV}}(e^{j\Omega_\mu}, n) = \mathbf{C}[\mathbf{C}^H \mathbf{C}]^{-1} \mathbf{f}$ . In particular, this vector does not depend on the spatial correlation matrix. The remaining component is orthogonal to the constraint matrix  $\mathbf{C}$  and can be split into a blocking matrix  $\mathbf{B}$  and a filter vector  $\mathbf{W}_a^H(e^{j\Omega_\mu}, n)$  [31]. In an adaptive realization this filter can be adapted without considering the linear constraint [30]. This structure is known as *generalized side-lobe canceler* (GSC) and is depicted in Figure 8.8. In the GSC structure the blocking matrix has to fulfill  $\mathbf{B}^H \mathbf{C} = \mathbf{0}$  [24].

The beamformer output is computed as

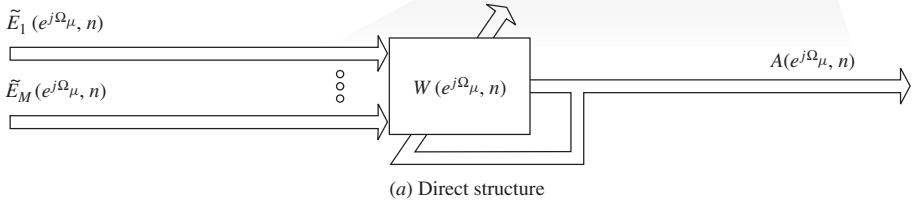
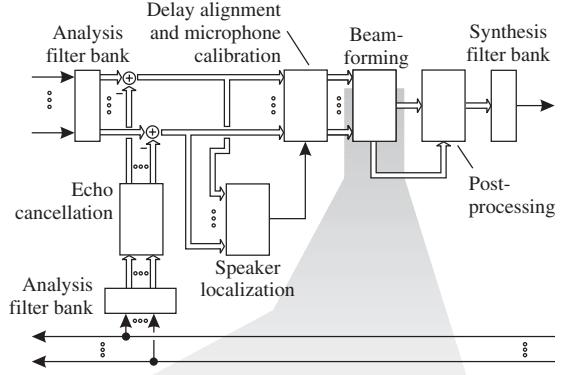
$$A_{\text{GSC}}(e^{j\Omega_\mu}, n) = [\mathbf{W}_q^H(e^{j\Omega_\mu}) - \mathbf{W}_a^H(e^{j\Omega_\mu}, n) \mathbf{B}^H] \tilde{\mathbf{E}}(e^{j\Omega_\mu}, n). \quad (8.23)$$

The blocking matrix does not perform any temporal filtering. Only the current time frame with the input signals  $\tilde{\mathbf{E}}_m(e^{j\Omega_\mu}, n)$  has to be processed. Thus, the dimension of the blocking matrix is reduced to  $(M - N_{\text{con}}) \times M$ . The output signals of the blocking matrix  $U_m(e^{j\Omega_\mu}, n)$ ,  $m = 1, \dots, M - N_{\text{con}}$ , are gathered in the vector

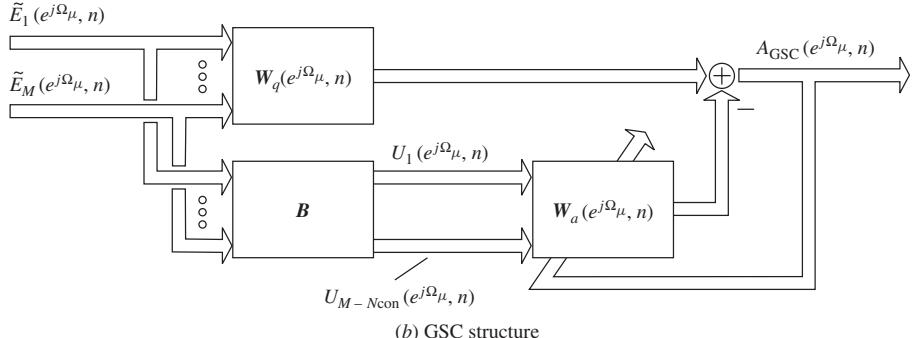
$$\mathbf{U}(e^{j\Omega_\mu}, n) = [\mathbf{U}_1^T(e^{j\Omega_\mu}, n), \dots, \mathbf{U}_{M-N_{\text{con}}}^T(e^{j\Omega_\mu}, n)]^T \quad (8.24)$$

with

$$\mathbf{U}_m(e^{j\Omega_\mu}, n) = [U_m(e^{j\Omega_\mu}, n), \dots, U_m(e^{j\Omega_\mu}, n - N_{\text{bf}} + 1)]^T. \quad (8.25)$$



(a) Direct structure



(b) GSC structure

**Figure 8.8** Beamformer in direct structure and as a general side-lobe canceler.

The NLMS algorithm, for example, can be used to adapt the filter coefficients:

$$\tilde{\mathbf{W}}_a(e^{j\Omega_\mu}, n+1) = \mathbf{W}_a(e^{j\Omega_\mu}, n) + \mu_{bf}(n) \frac{\mathbf{A}_{GSC}^*(e^{j\Omega_\mu}, n) \mathbf{U}(e^{j\Omega_\mu}, n)}{\mathbf{U}^H(e^{j\Omega_\mu}, n) \mathbf{U}(e^{j\Omega_\mu}, n)}. \quad (8.26)$$

The step size  $\mu_{bf}(n)$  is used to control the speed of the adaptation. In order to ensure stability of the adaptive algorithm, the step size has to be chosen as  $0 < \mu_{bf}(n) < 2$ .

In order to increase the robustness of the beamformer, similar to the constraint in Eq. (8.21), the norm of the adaptive filter coefficients can be limited [27, 32]:

$$\mathbf{W}_a(e^{j\Omega_\mu}, n) = \begin{cases} \tilde{\mathbf{W}}_a(e^{j\Omega_\mu}, n), & \text{if } \|\tilde{\mathbf{W}}_a(e^{j\Omega_\mu}, n)\|^2 < K, \\ \frac{\sqrt{K} \tilde{\mathbf{W}}_a(e^{j\Omega_\mu}, n)}{\|\tilde{\mathbf{W}}_a(e^{j\Omega_\mu}, n)\|}, & \text{otherwise.} \end{cases} \quad (8.27)$$

### 8.5.3 Robustness Aspects

The signal model for the LCMV beamformer assumes an ideal wave field as well as perfect microphones. However, in real-world situations the LCMV constraint is weakened because these assumptions are never fulfilled perfectly: There might be deviations at the array, such as microphone tolerances or imprecise positioning. Furthermore, a mismatch in the acoustic environment may occur, such as an uncertain direction of arrival or multipath propagation due to room acoustics. In the GSC structure these mismatched conditions lead to leakage of the desired signal into the noise reference signals  $U_m(e^{j\Omega}, n)$ . Hence, the adaptive filters tend to cancel out the desired speech signal, which results in the so-called *signal cancellation* effect [33].

For speech applications an effective way to circumvent this problem is to control the adaptation step size and permit adaptation only in speech pauses [32, 34]. A robust adaptation control is described in Section 8.6.2.

An effective measure to further increase robustness of an adaptive beamformer is an adaptive blocking matrix [32, 35, 36], where a mismatch of the steering direction can be compensated for. In [37] the relative deviations of the room transfer functions from the desired sound source to the microphones are addressed. A generalized framework that also considers mismatched microphone sensitivities has been presented in [38]. One realization of this framework is presented in Section 8.6. Finally, Table 8.1 shows speech recognition results with utterances (digits) that were recorded in a car at a speed of about 130 km/h. The advantages of adaptive beamforming and of additional microphone calibration as it will be described in Section 8.6 are clearly visible.

### 8.5.4 Combined Echo Cancellation and Beamforming

In the case of full-duplex hands-free systems echo cancellation is required since beamformers are not able to suppress echo components sufficiently in reverberant rooms. As discussed in Section 8.3 the computational complexity for echo cancellation in the multichannel case has the order  $RM$ , where  $R$  is the number of play-back channels and  $M$  is the number of microphones.

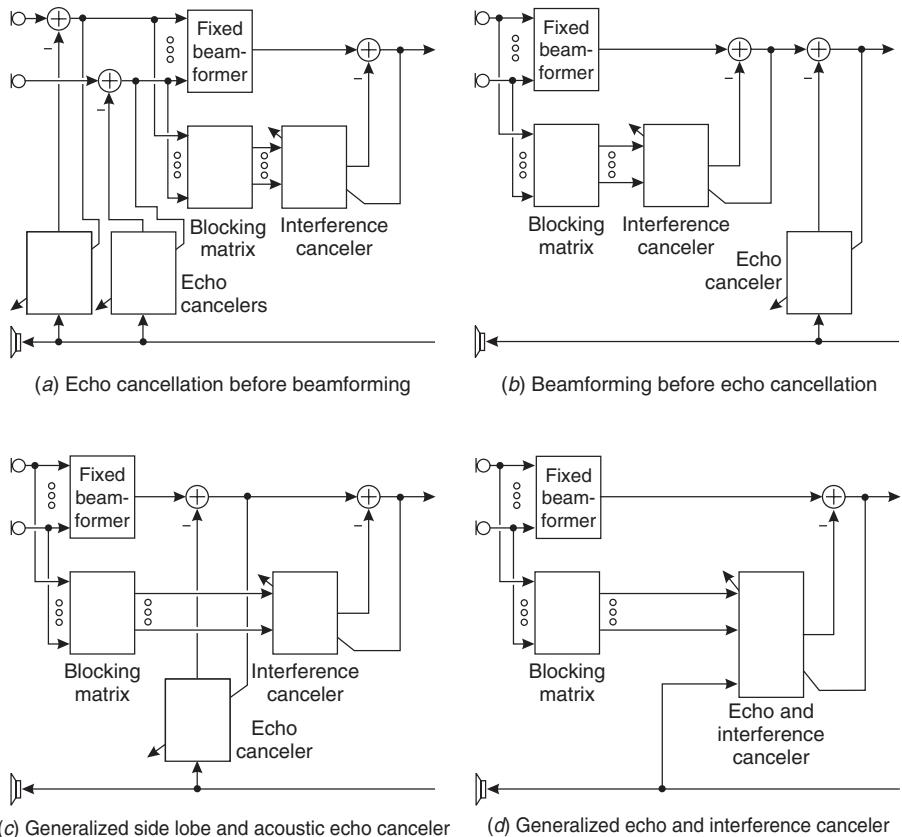
In order to reduce the computational effort various approaches have been proposed for combining echo cancellation and beamforming in an efficient manner:

- Beamforming ahead of echo cancellation (Fig. 8.9b): Combining adaptive beamforming and echo cancellation means cascading two adaptive systems. Since the beamformer has to be considered as part of the echo path, the echo canceler has to track the beamformer's adaptation behavior. Each adaptation of the adaptive beamformer effectively means an enclosure dislocation. Furthermore, the

**TABLE 8.1 Results of Speech Recognition Tests with Digit Loops**

	Unprocessed Single Microphone (%)	Fixed Beamformer (%)	GSC (%)	GSC with Calibration (%)
Word error rate	5.44	3.17	2.81	2.18
Relative reduction of the word error rate	-93.6	-12.8	0	22.4

*Note:* Relative results are given with respect to the generalized side-lobe canceler (GSC).



**Figure 8.9** Different variants for combining echo cancellation and beamforming.

convergence of an adaptive beamformer is usually much faster than that of an echo canceler. Thus, placing the echo canceler behind the beamformer is only reasonable in the case of a fixed beamformer.

- A compromise where the echo cancelers operate within the beamformer has been proposed in [39, 40] (Fig. 8.9c). Here, a beamformer in GSC structure is supposed, and the echo cancellation is performed on the output signal of the fixed beamformer. Thus, only for one channel echo cancellation has to be processed. However, since the adaptive path of the GSC is not subject to any echo cancellation strong echo components may leak over the noise canceling filters into the GSC output signal again. This system has been termed as *generalized side-lobe and acoustic echo canceler* (GSAEC).
- A further approach for combining adaptive echo cancellation and adaptive beamforming is to jointly optimize their adaptive filters on the basis of the same error signal [35, 41, 42] (Fig. 8.9d). Within the GSC structure the loudspeaker signals are treated like additional microphone input signals. This approach has been termed as *generalized echo and interference canceler* (GEIC).

Generally, if the input signals of the noise canceling filters contain echo components in addition to the noise components, two extreme cases can be considered. In the first

case the noise component is much stronger than the echo component, whereas in the second scenario the echo component is dominant. In general, for both cases different optimal solutions result for the adaptive filters. Thus, for the above-mentioned systems the optimal filter coefficients depend on the *echo-to-noise ratio* (ENR). As for most speech applications echo and noise components are nonstationary signals, the ENR may strongly vary over time. As a consequence, the adaptive filters have to follow the quickly changing optimal solution, that is, a frequent misadjustment has to be handled. For this reason it is advantageous to suppress echo components prior to adaptive beamforming (Fig. 8.9a) if there is sufficient computational power available.

## 8.6 SENSOR CALIBRATION

As already pointed out in Section 8.5.3 microphone mismatch in the context of adaptive beamforming causes speech signal cancellation. To reduce that problem the microphones can be preselected with regard to optimal matching or predetermined calibration filters can be applied [43–45]. Both of these solutions require (costly) measurements. Furthermore, the microphones' characteristics usually change over time due to aging effects or environmental influences (e.g., temperature). For that reason, an adaptive matching that tracks these changes is desirable. In the last few years the interest in adaptive methods for self-calibration has increased strongly [38, 46, 47].

In order to examine the deviations among real microphones, we measured the directional properties of 47 microphones. The deviations in dependence of the frequency and the direction of incidence are depicted in Figure 8.10. In [48] it has been shown that it is not possible to compensate the deviations perfectly but only for one direction. For adaptive beamforming the deviations for the steering direction are the most perturbing and should therefore be compensated in order to get identical desired signal components for each microphone.

### 8.6.1 Adaptive Calibration

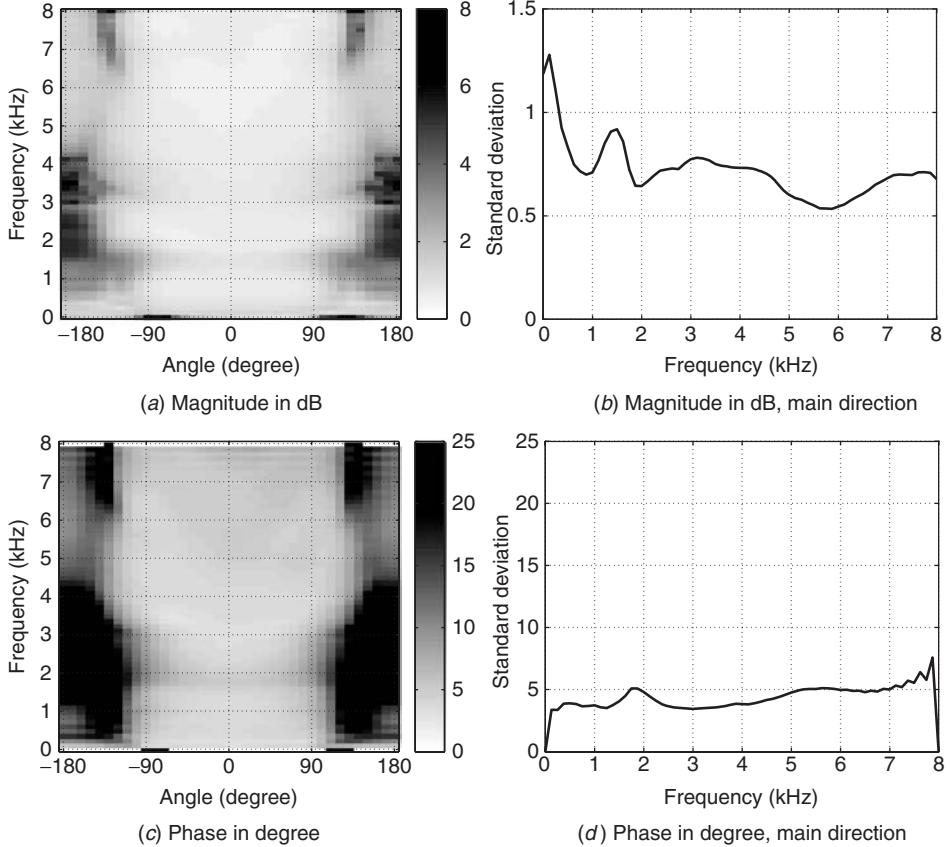
In [38] several solutions for an adaptive calibration have been proposed. The most promising solution is given by a recursive structure, which is depicted in Figure 8.11. For each microphone  $m$  and each subband  $\mu$  the time-aligned input signals are filtered by adaptive calibration filters  $G_{m,l}(e^{j\Omega_\mu}, n)$  with coefficients  $l \in \{0, \dots, N_{\text{cal}} - 1\}$  in order to compensate for the mismatches:

$$\tilde{E}_m^c(e^{j\Omega_\mu}, n) = \sum_{l=0}^{N_{\text{cal}}-1} G_{m,l}^*(e^{j\Omega_\mu}, n) \tilde{E}_m(e^{j\Omega_\mu}, n-l). \quad (8.28)$$

The calibrated input signals  $\tilde{E}_m^c(e^{j\Omega_\mu}, n)$  are fed into a delay-and-sum beamformer as described in Eq. (8.10). The output signal  $A_{\text{ds}}^c(e^{j\Omega_\mu}, n)$  serves as (an enhanced) desired signal for the adaptive filters:

$$A_{\text{ds}}^c(e^{j\Omega_\mu}, n) = \sum_{m=1}^M \tilde{E}_m^c(e^{j\Omega_\mu}, n), \quad (8.29)$$

$$\check{E}_m(e^{j\Omega_\mu}, n) = A_{\text{ds}}^c(e^{j\Omega_\mu}, n) - \tilde{E}_m^c(e^{j\Omega_\mu}, n). \quad (8.30)$$



**Figure 8.10** Deviations of uncalibrated microphones: (a), (b) standard deviation of the logarithmic magnitude; (c), (d) standard deviation of the phase. The nominal directionality of the microphones is a hypercardioid pattern [49].

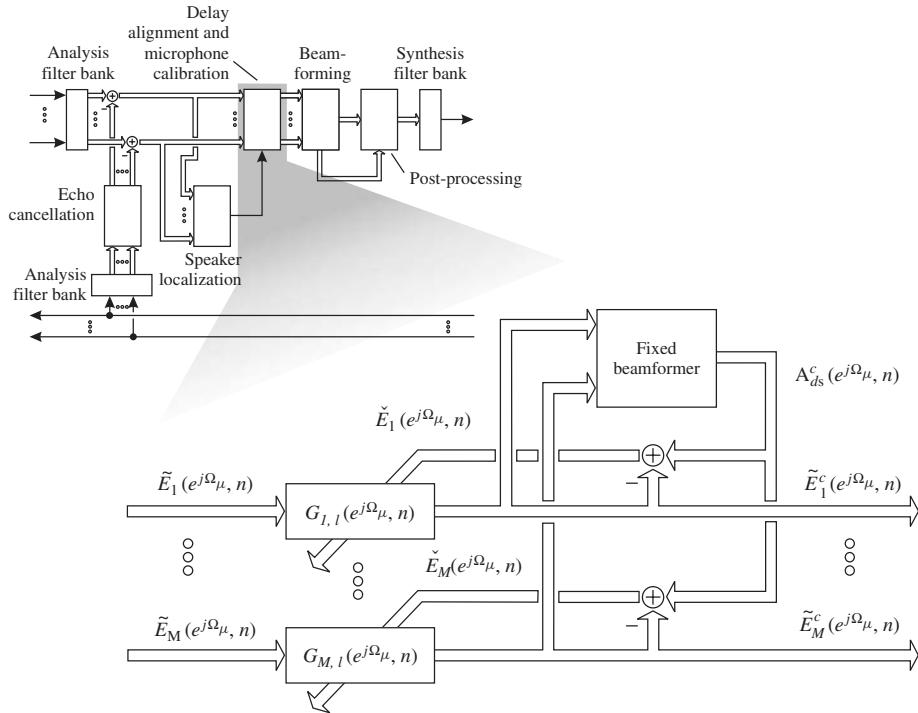
During speech activity an adaptive algorithm has to minimize the mean square of the error signals  $\check{E}_m(e^{j\Omega_\mu}, n)$ . Because of the recursive structure, a constraint according to

$$\sum_{m=1}^M G_{m,l}(e^{j\Omega_\mu}, n) = \begin{cases} M, & \text{for } l = D_{\text{cal}}, \\ 0, & \text{otherwise,} \end{cases} \quad (8.31)$$

with  $0 \leq D_{\text{cal}} < N_{\text{cal}}$ , has to be applied in order to prevent the filters from converging toward zero.

### 8.6.2 Adaptation Control

The adaptation step sizes for both the adaptive beamformer as well as the proposed self-calibration system are controlled by a criterion that detects signal activity from the desired direction. This is accomplished by a ratio of the smoothed signal powers  $\hat{S}_{\text{ads}}(\Omega_\mu, n)$  and  $\hat{S}_{uu}(\Omega_\mu, n)$  of the output signal of a delay-and-sum beamformer



**Figure 8.11** Adaptive self-calibration. The filters are adapted with a recursive structure during speech activity. The output signals may serve as input for a subsequent beamformer.

$A_{ds}(e^{j\Omega_\mu}, n)$  and the output signals of a blocking matrix  $U_m(e^{j\Omega_\mu}, n)$ , respectively, both steered to the desired direction:

$$\widehat{S}_{a_{ds}a_{ds}}(\Omega_\mu, n) = \alpha_q \widehat{S}_{a_{ds}a_{ds}}(\Omega_\mu, n - 1) + (1 - \alpha_q) |A_{ds}(e^{j\Omega_\mu}, n)|^2, \quad (8.32)$$

$$\widehat{S}_{uu}(\Omega_\mu, n) = \alpha_q \widehat{S}_{uu}(\Omega_\mu, n - 1) + (1 - \alpha_q) \sum_{m=1}^{M-N_{\text{con}}} \frac{|U_m(e^{j\Omega_\mu}, n)|^2}{M - N_{\text{con}}}, \quad (8.33)$$

$$Q_{sd}(\Omega_\mu, n) = \frac{\widehat{S}_{a_{ds}a_{ds}}(\Omega_\mu, n)}{W_{eq}(e^{j\Omega_\mu}, n) \widehat{S}_{uu}(\Omega_\mu, n)}. \quad (8.34)$$

The parameter  $\alpha_q$  is a fixed smoothing constant and the adaptive equalization factors  $W_{eq}(e^{j\Omega_\mu}, n)$  serve to normalize this ratio. These factors are adjusted in speech pauses in such a way that the ratio becomes  $Q_{sd}(\Omega_\mu, n) \approx 1$  in periods of stationary background noise. High values of  $Q_{sd}(\Omega_\mu, n)$  indicate signal energy from the steering direction. To further improve the robustness of this criterion, the ratio can be averaged or smoothed across adjacent subbands.

The beamformer filters are adjusted only when  $Q_{sd}(\Omega_\mu, n)$  falls below a predetermined threshold  $Q_{bf}(\Omega_\mu)$ . The calibration filters are adapted only if this ratio exceeds a threshold  $Q_{sc}(\Omega_\mu)$ , with  $Q_{bf}(\Omega_\mu) < Q_{sc}(\Omega_\mu)$ , and the signal power is sufficiently large.

## 8.7 POSTPROCESSING

For most cases the application of acoustic echo cancellation and beamforming as described in the preceding sections is not sufficient. Even if these methods work properly, bothersome residual interferences remain in the output signal.

- Acoustic echo cancelers typically achieve not more than 30–35 dB of echo attenuation in practical applications. Furthermore, enclosure dislocations may lead to a temporarily raised residual echo component in the output signal due to abrupt misadjustments of the adaptive filters.
- Also, the performance of beamformers strongly depends on the acoustic environment. Whereas coherent noise sources such as interfering speakers can be suppressed quite well—diffuse noise such as driving noise in a car only permits for a smaller amount of noise reduction by a beamformer.
- As beamformers are optimized for suppressing noise, in many cases only a moderate attenuation of reverberant signal components can be achieved. However, particularly for speech recognition, reverberation causes a serious drop in performance.

Thus, there is the need to further enhance the beamformer output signal without distorting the desired speech signal considerably. This is accomplished most commonly by postfilters, which weight the spectral signal components with dynamic scalar factors (see Fig. 8.12). Different approaches that address the above-mentioned undesired residual components will be presented in the subsequent sections. If the signal is disturbed to such an extend that postfilter approaches are not able to work properly anymore, signal reconstruction may be applied to replace the noisy signal by a synthesized clean speech signal as will be pointed out in Section 8.7.2. Often it sounds more pleasant if there is a certain amount of additional noise in the output signal. Therefore, a synthetical noise signal, the so-called *comfort noise*, is generated. The reconstructed speech signal as well as the comfort noise are combined with the postfiltered signal by an adaptive mixer.

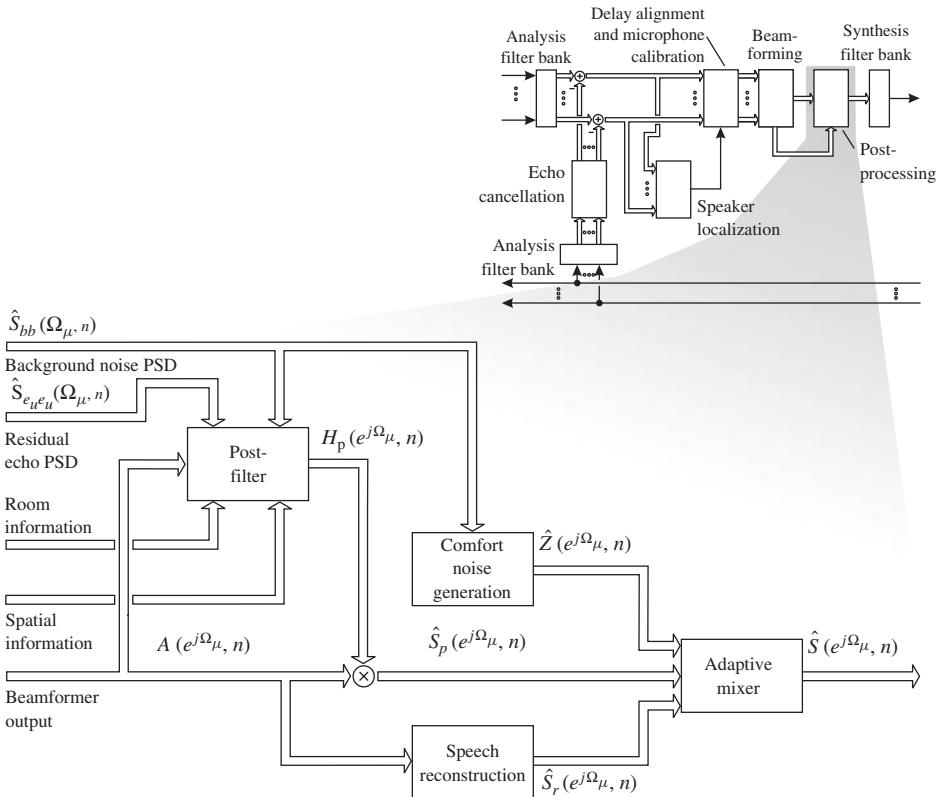
### 8.7.1 Suppression of Residual Interferences

The complex-valued spectrum  $A(e^{j\Omega_\mu}, n)$  at the output of a beamformer is composed of the desired signal portion  $A_s(e^{j\Omega_\mu}, n)$  as well as the residual noise component  $A_n(e^{j\Omega_\mu}, n)$ . The residual noise itself may consist of different kinds of interferences:

- Stationary background noise  $B(e^{j\Omega_\mu}, n)$
- Nonstationary interferences  $I(e^{j\Omega_\mu}, n)$
- Residual echo components  $E_u(e^{j\Omega_\mu}, n)$
- Reverberation  $V(e^{j\Omega_\mu}, n)$

The power of the interfering mixture is to be reduced as far as possible without audible degradation of the desired signal component. To achieve this, a large number of noise reduction methods apply real-valued attenuation factors to the noisy subband signals:

$$\hat{S}_p(e^{j\Omega_\mu}, n) = A(e^{j\Omega_\mu}, n) H_p(e^{j\Omega_\mu}, n). \quad (8.35)$$



**Figure 8.12** Overview of the different stages of postprocessing schemes. PSD stands for power spectral density.

Those methods differ mainly with respect to the function used to determine the filter coefficients  $H_p(e^{j\Omega_\mu}, n)$ . These filter characteristics are designed to meet some optimization criterion, often based on statistical signal models. Therefore, the filter characteristics are mostly a function of the

$$\text{a priori SNR} \quad \xi_a(\Omega_\mu, n) = \frac{S_{aa}(\Omega_\mu, n)}{S_{an}(\Omega_\mu, n)}, \quad (8.36)$$

$$\text{a posteriori SNR} \quad \lambda_a(\Omega_\mu, n) = \frac{S_{aa}(\Omega_\mu, n)}{S_{a_n a_n}(\Omega_\mu, n)}, \quad (8.37)$$

or either one of them. Some widely used characteristics [50–52], for instance, depend on both ratios, whereas other methods use exclusively  $\lambda_a(\Omega_\mu, n)$ . The latter is done in our framework where the signal at the beamformer output is processed according to the following generic characteristic:

$$H_p(e^{j\Omega_\mu}, n) = \left(1 - (\lambda_a^{-1}(\Omega_\mu, n))^\beta\right)^\alpha. \quad (8.38)$$

For  $\alpha = \beta = 1$  the well-known *Wiener* filter weights are obtained. Choosing  $\alpha = \frac{1}{2}$  and  $\beta = 1$  results in the so-called *power subtraction* and by  $\alpha = 1$  and  $\beta = \frac{1}{2}$ , a so-called

*magnitude subtraction* is performed [3, 53]. The derivations of these characteristics assume that the desired signal and the interference are uncorrelated.

To process the filter coefficients, however, the knowledge of the power spectral densities of the disturbed signal  $S_{aa}(\Omega_\mu, n)$ , and the interference  $S_{a_n a_n}(\Omega_\mu, n)$ , respectively, is presumed. Both are generally not known but may be estimated. The main obstacle to estimate  $S_{aa}(\Omega_\mu, n)$  is the nonstationary nature of speech signals, which prohibits temporal averaging. Therefore  $S_{aa}(\Omega_\mu, n)$  is often approximated by the current power spectrum  $|A(e^{j\Omega_\mu}, n)|^2$ . Estimation of the noise power spectral density  $S_{a_n a_n}(\Omega_\mu, n)$  may be by far more complicated and can actually be considered to be the main task to make a noise reduction system work. Depending on the kind of interference, different means have to be taken.

The problem of suppressing a mixture of interferences may be approached in different ways. A very practical way is to first treat all interfering components independently, and to come up with some unified treatment based on available information about the individual components. A possible heuristic way to do so may be to first estimate all noise power spectral densities separately and to calculate the respective filter weights for each component. Afterwards the resulting filter weights may be obtained by choosing the minimum weight for instance. The product of filter weights or even more sophisticated mappings are possible as well. As opposed to this, the power spectral density of the mixture of interferences may also be estimated directly. In this case the filter characteristic has to be applied only once. In the following, noise suppression schemes for different kinds of interferences are considered. At the end of Section 8.7.1 a spatial postfilter is considered that does not distinguish between different kinds of interferences. Nevertheless the respective filter weights may also be combined with coefficients that are processed on a different basis. This may lead to increased robustness.

**8.7.1.1 Background Noise Suppression** The suppression of stationary background noise has been studied intensively and is well understood. Exploiting the stationarity of the noise, estimation of its power spectral density is usually carried out during speech pauses. When the desired signal is active, the spectral noise power estimate is typically changed very slowly or even held constant. This proceeding, however, assumes the availability of voice activity information. When the desired signal is surely not present,  $S_{bb}(\Omega_\mu, n)$  is most commonly estimated by recursive smoothing of the squared magnitudes of the current spectrum. For this, a time-dependent smoothing parameter can be used to control the degree of the temporal smoothing. An overview of control methods is given in [1]. Various alternative methods to estimate the power spectral density of the background noise exist with the method of minimum statistics being the most prominent [54].

Once the estimate is available, it is used to process the filter coefficients according to Eq. (8.38). As mentioned above, the estimate  $\widehat{S}_{aa}(\Omega_\mu, n)$  for the noisy power spectral density typically incorporates much less temporal smoothing than  $\widehat{S}_{bb}(\Omega_\mu, n)$ . As a consequence, both quantities differ during speech pauses. Therefore, the filter coefficients deviate from the “correct” value zero. Negative filter coefficients, however, are not reasonable at all but can be prohibited by introducing a lower bound  $H_{\min}$ , called *spectral floor*. The spectral floor is usually set to a value greater than zero to avoid a zero filter output because such signals are not perceived as pleasant. If the filter coefficients are greater than zero, though no desired signal is present, the background noise will be modulated accordingly, which results in the so-called *musical noise* phenomenon. A heuristic way to prevent the musical noise effect is to simply overestimate

the noise power spectral density by a fixed factor. Too much overestimation should not be applied either since this would result in distortions of the desired signal. A more sophisticated noise reduction method that is robust to musical noise is *recursive Wiener filtering* [53] or the suppression rule derived by Ephraim and Malah [51].

The methods described above work effectively for stationary or slowly varying background noise such as driving noise in a car, for instance. For nonstationary interferences, however, those methods are not suitable since estimation of the noise power spectral density during speech activity is usually very problematic unless some kind of additional information is available that facilitates the estimation process.

**8.7.1.2 Residual Echo Suppression** A further interfering signal component of the beamformer output signal  $A(e^{j\Omega_\mu}, n)$  is the residual echo  $E_u(e^{j\Omega_\mu}, n)$ , which is often referred to as the undisturbed error signal (see Section 8.3.2). Its power spectral density can be estimated from the reference signals  $X_r(e^{j\Omega_\mu}, n)$  and the system mismatches  $H_{\Delta,r}(e^{j\Omega_\mu}, n)$ , which measure the misadjustments of the adaptive echo cancellation filters:<sup>3</sup>

$$\widehat{S}_{e_ue_u}(\Omega_\mu, n) = \sum_{r=1}^R |\widehat{H}_{\Delta,r}(e^{j\Omega_\mu}, n)|^2 S_{x_r x_r}(\Omega_\mu, n). \quad (8.39)$$

The system mismatch is individual for each playback channel and varies in time as the adaptive filters converge. Since the system mismatch is not observable directly, it has to be estimated. Strategies for dynamic estimation of the system mismatch are pointed out in [1, 11]. The residual echo suppression filter coefficients  $H_{\text{res}}(e^{j\Omega_\mu}, n)$  are calculated in the way as indicated in Eq. (8.38) with  $S_{a_n a_n}(\Omega_\mu, n)$  replaced in Eq. (8.37) by  $\widehat{S}_{e_ue_u}(\Omega_\mu, n)$ .

**8.7.1.3 Dereverberation** Whereas the control of background noise and residual echo is well understood, the field of dereverberation is still an open research topic. The classical approaches attempt to dereverberate the signal by inverse filtering or deconvolution. These methods, however, lack robustness since very precise estimates of the unknown channel are required. Inevitable estimation errors can actually cause severe artifacts. Another approach is to suppress reverberation components by dynamic spectral weighting [55–57].

In this chapter a simple but effective subband domain method for suppressing the reverberation component  $V(e^{j\Omega_\mu}, n)$  is presented that uses the filter characteristics given in Eq. (8.38). For most rooms the energy of the late reverberation decays exponentially over time [58]. The same is true for the energy of the subband impulse responses  $G_v(e^{j\Omega_\mu}, n)$ . Estimation of the power spectral density of the reverberation component  $S_{vv}(\Omega_\mu, n)$  is therefore alleviated using the following model:

$$|G_v(e^{j\Omega_\mu}, n)|^2 \approx \begin{cases} 0, & \text{for } n \leq 0, \\ A_v e^{-\gamma_v n}, & \text{for } n > 0. \end{cases} \quad (8.40)$$

The parameter  $\gamma_v$  denotes the steepness of the temporal decay and depends mainly on room parameters such as room size or sound absorption at the walls. The parameter  $A_v$  accounts for the ratio of the reverberation power to the total signal power and depends

<sup>3</sup>As in Section 8.3.2 the individual reference channels  $x_r(n)$  are assumed to be mutually uncorrelated.

mainly on the distance of the speaker and the microphones. Based on this model the power spectral density of the reverberation can be estimated as follows:

$$S_{vv}(\Omega_\mu, n) \approx \sum_{l=D_v}^{\infty} S_{ss}(\Omega_\mu, n-l) A_v e^{-\gamma_v l}. \quad (8.41)$$

The fact that the reverberation may be statistically dependent from the desired speech signal has been neglected here. According to [58] the first 50 ms of a room impulse response increase the speech intelligibility, whereas the opposite is true for the late reverberation. Therefore the first  $D_v$  frames are excluded in Eq. (8.41). To achieve a computationally efficient estimation, Eq. (8.41) can be formulated recursively. Furthermore, since the power spectral density of the clean speech signal  $S_{ss}(\Omega_\mu, n)$  is practically not available, the squared magnitude of the beamformer output signal can be taken as an approximation:

$$\widehat{S}_{vv}(\Omega_\mu, n) = \widehat{S}_{vv}(\Omega_\mu, n-1) e^{-\gamma_v} + |A(e^{j\Omega_\mu}, n-D_v)|^2 A_v e^{-\gamma_v D_v}. \quad (8.42)$$

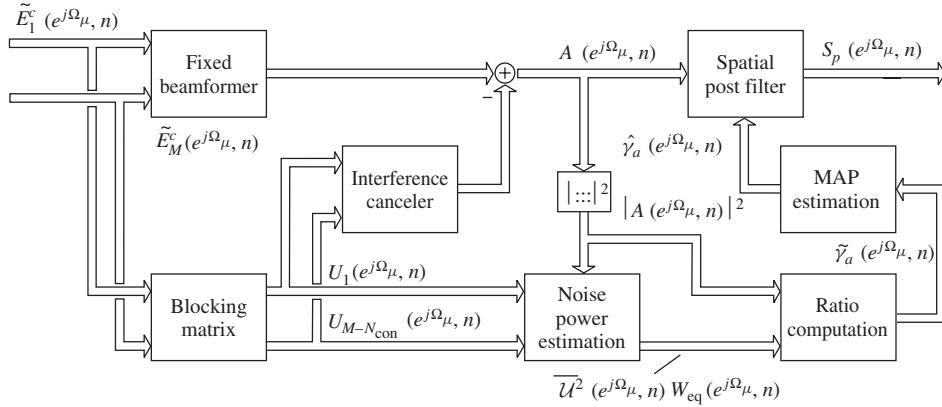
The first term models the decay of the existing reverberation power, whereas the second term accounts for new contributions. The dereverberation filter coefficients  $H_v(e^{j\Omega_\mu}, n)$  are processed according to Eq. (8.38), whereas  $S_{an_a}(\Omega_\mu, n) = \widehat{S}_{vv}(\Omega_\mu, n)$ . Here, the magnitude filter characteristics showed the best results. For details concerning the estimation of the model parameters  $\gamma_v$  and  $A_v$  the interested reader is referred to [59].

The proposed system was evaluated for speech recognition with Lombard speech [60] data and impulse responses measured in different environments. Relative word error rate improvements up to 50% in reverberant environments (about 0.5 s reverberation time) were measured.

**8.7.1.4 Spatial Postfiltering** The methods addressed in the preceding sections either exploit the fact that the respective interference differs from the desired speech signal with respect to temporal signal characteristics, or make use of auxiliary information like the reference signal. In this section a new approach for postfiltering is presented and discussed that does not distinguish between different kinds of interferences. Its structure is depicted in Figure 8.13. The desired signal and the interferences are distinguished by considering *spatial* information from the multichannel part of the system. To simplify the notation, both time and frequency indices will be omitted in some of the following equations.

Known approaches to postfiltering that exploit spatial information consider the spatial correlation [61] or derive a signal absence probability used to control the spectral noise power estimate [62]. Unlike these approaches and those mentioned in the preceding sections, we propose to determine the filter coefficients *in the style* of Eq. (8.38):

$$H_{sp}(e^{j\Omega_\mu}, n) = \max \left\{ 1 - \widehat{\gamma}_a^{-1}(e^{j\Omega_\mu}, n), H_{\min} \right\}, \quad (8.43)$$



**Figure 8.13** Structure of a spatial postfilter system. The abbreviation MAP stands for maximum a posteriori.

whereas the quantity  $\hat{\gamma}_a(e^{j\Omega_\mu}, n)$  is a maximum a posteriori (MAP) estimate for what we call the *instantaneous a posteriori SNR*. Since the filter shall suppress highly time-varying interferences, we choose  $\gamma_a(e^{j\Omega_\mu}, n)$  to be highly time varying as well, and therefore define it as a function of the current magnitude squares:

$$\gamma_a(e^{j\Omega_\mu}, n) = \frac{|A_s(e^{j\Omega_\mu}, n) + A_n(e^{j\Omega_\mu}, n)|^2}{|A_n(e^{j\Omega_\mu}, n)|^2} = \frac{|A(e^{j\Omega_\mu}, n)|^2}{|A_n(e^{j\Omega_\mu}, n)|^2}. \quad (8.44)$$

Hence, the *deterministic* ratio  $\lambda_a(\Omega_\mu, n)$  in Eq. (8.38), is replaced by the *instantaneous* quantity  $\gamma_a(e^{j\Omega_\mu}, n)$ . We consider  $\gamma_a(e^{j\Omega_\mu}, n)$  as a realization of a random variable, which is due to the denominator  $|A_n(e^{j\Omega_\mu}, n)|^2$ , not observable. Statistical modeling allows to derive a maximum a posteriori estimate of the instantaneous a posteriori SNR.

To obtain an estimate for the denominator term of Eq. (8.44), the squared magnitudes at the output of a blocking matrix are at first averaged across the  $M - N_{\text{con}}$  channels:

$$\overline{U^2}(e^{j\Omega_\mu}, n) = \sum_{m=1}^{M-N_{\text{con}}} \frac{|U_m(e^{j\Omega_\mu}, n)|^2}{M - N_{\text{con}}}. \quad (8.45)$$

Subsequently, the temporal average of this signal is matched to that of  $|A_n(e^{j\Omega_\mu}, n)|^2$ , which is done during speech pauses using appropriate equalization factors [similar to Eq. (8.34)]. Hence, a preliminary estimate for the instantaneous a posteriori SNR is given by the ratio:

$$\tilde{\gamma}_a(e^{j\Omega_\mu}, n) = \frac{|A(e^{j\Omega_\mu}, n)|^2}{\overline{U^2}(e^{j\Omega_\mu}, n) W_{eq}(e^{j\Omega_\mu}, n)}. \quad (8.46)$$

Note that this ratio involves no temporal averaging. In the logarithmic domain this estimate is composed of the desired part plus an additive estimation error called  $\Delta(e^{j\Omega_\mu}, n)$ :

$$\begin{aligned}\tilde{\Gamma}_a(e^{j\Omega_\mu}, n) &= 10 \log_{10}(\gamma_a(e^{j\Omega_\mu}, n)) + 10 \log_{10} \left[ \frac{|A_n(e^{j\Omega_\mu}, n)|^2}{\mathcal{U}^2(e^{j\Omega_\mu}, n) W_{\text{eq}}(e^{j\Omega_\mu}, n)} \right] \\ &= \Gamma_a(e^{j\Omega_\mu}, n) + \Delta(e^{j\Omega_\mu}, n).\end{aligned}\quad (8.47)$$

The estimation error causes the *musical noise* phenomenon mentioned above. Here, we employ the MAP principle to optimize the estimate for  $\Gamma_a(e^{j\Omega_\mu}, n)$ . As shown in [63], the *probability density function* (pdf) of  $\Gamma_a$  can be modeled as follows:

$$\hat{f}_{\Gamma_a}(\Gamma_a) = \left( \frac{1}{\sqrt{2\pi\sigma_{\Gamma_a}^2(\xi_a)}} \right) \exp \left[ - \frac{(\Gamma_a - \mu_{\Gamma_a}(\xi_a))^2}{2\sigma_{\Gamma_a}^2(\xi_a)} \right], \quad (8.48)$$

whereas  $\sigma_{\Gamma_a}^2(\xi_a)$  and  $\mu_{\Gamma_a}(\xi_a)$  denote the variance and average, respectively. Both depend on the a priori SNR  $\xi_a(e^{j\Omega_\mu}, n)$ :

$$\mu_{\Gamma_a}(\xi_a) = 10 \log_{10}(\xi_a + 1) \quad \text{and} \quad \sigma_{\Gamma_a}^2(\xi_a) = \sigma_\Phi^2 \left( \frac{\xi_a}{0.5 + \xi_a} \right), \quad (8.49)$$

whereas according to Eq. (8.49), the variance does not exceed its upper limit  $\sigma_\Phi^2$ . The pdf of the observable  $\tilde{\Gamma}_a$ , conditioned on the undisturbed value  $\Gamma_a$ , is modeled according to

$$f_{\tilde{\Gamma}_a}(\tilde{\Gamma}_a | \Gamma_a) = \left( \frac{1}{\sqrt{2\pi\sigma_\Delta^2}} \right) \exp \left[ - \frac{(\tilde{\Gamma}_a - \Gamma_a)^2}{2\sigma_\Delta^2} \right]. \quad (8.50)$$

Here,  $\sigma_\Delta^2$  denotes the variance of the estimation error  $\Delta(e^{j\Omega_\mu}, n)$ . Using these two models together with Bayes' rule, the a posteriori pdf becomes available. To find its maximum, we solve

$$\frac{\partial}{\partial \Gamma_a} \ln(f_{\tilde{\Gamma}_a}(\tilde{\Gamma}_a | \Gamma_a) \cdot \hat{f}_{\Gamma_a}(\Gamma_a)) = 0 \quad (8.51)$$

and obtain the desired estimate in the logarithmic domain

$$\widehat{\Gamma}_a = \frac{\sigma_\Phi^2 \xi_a \tilde{\Gamma}_a + (\xi_a + 0.5) \sigma_\Delta^2 10 \log_{10}(\xi_a + 1)}{\sigma_\Phi^2 \xi_a + (\xi_a + 0.5) \sigma_\Delta^2}. \quad (8.52)$$

Finally, the linearly scaled estimate results from  $\widehat{\gamma}_a = 10^{\widehat{\Gamma}_a/10}$ . Hence, the mapping given in Eq. (8.52) maps the observable  $\tilde{\Gamma}_a(e^{j\Omega_\mu}, n)$ , and the a priori SNR  $\xi_a(\Omega_\mu, n)$  onto the MAP estimate for the logarithmic instantaneous a posteriori SNR  $\Gamma_a(e^{j\Omega_\mu}, n)$ . The variance  $\sigma_\Delta^2$  plays the role of a trade-off factor. Note the two cases

$$\widehat{\Gamma}_a \Big|_{\sigma_\Delta^2=0} = \tilde{\Gamma}_a \quad \text{and} \quad \widehat{\Gamma}_a \Big|_{\sigma_\Delta^2 \rightarrow \infty} = 10 \log_{10}(\xi_a + 1). \quad (8.53)$$

In the first case,  $\tilde{\gamma}_a(e^{j\Omega_\mu}, n)$  is equal to  $\gamma_a(e^{j\Omega_\mu}, n)$  and is therefore passed to the postfilter without further modification. Thus, no temporal averaging is involved in the filtering process. If the random variables associated with  $A_s(e^{j\Omega_\mu}, n)$  and  $A_n(e^{j\Omega_\mu}, n)$  are statistically independent, the second case results in the filter weights of the well-known Wiener filter since then we have  $\xi_a(\Omega_\mu, n) + 1 = \lambda_a(\Omega_\mu, n)$ . In this case the filtering process does involve temporal averaging because  $\xi_a(\Omega_\mu, n)$ , respectively,  $\lambda_a(\Omega_\mu, n)$ , cannot be estimated without exploitation of the dimension time. Thus, the lower the variance of the estimation error  $\Delta(e^{j\Omega_\mu}, n)$ , the fewer temporal averaging is introduced into the MAP estimate. The a priori SNR  $\xi_a(\Omega_\mu, n)$  is estimated using the so-called *decision-directed approach* [51], whereas it is of particular importance that the respective noise power estimate is generated on the basis of  $\bar{\mathcal{U}^2}(e^{j\Omega_\mu}, n)$ , as this introduces spatial information into the estimator. It should be noted, however, that this kind of postfiltering is not restricted to a specific kind of beamforming.

The above-described method has been evaluated using a four-channel real-time implementation with a GSC-type beamformer. Results from speech recognition tests in three different acoustical environments with highly nonstationary noise indicate that the proposed filter achieves significantly lower word error rates than a conventional Wiener postfilter in all acoustical scenarios considered (see Table 8.2). Informal listening tests have also shown that the filter is capable of suppressing highly nonstationary interferences without seriously degrading the desired speech signal. As an example see Figure 8.14 showing four time–frequency analyses. For further details please see [63].

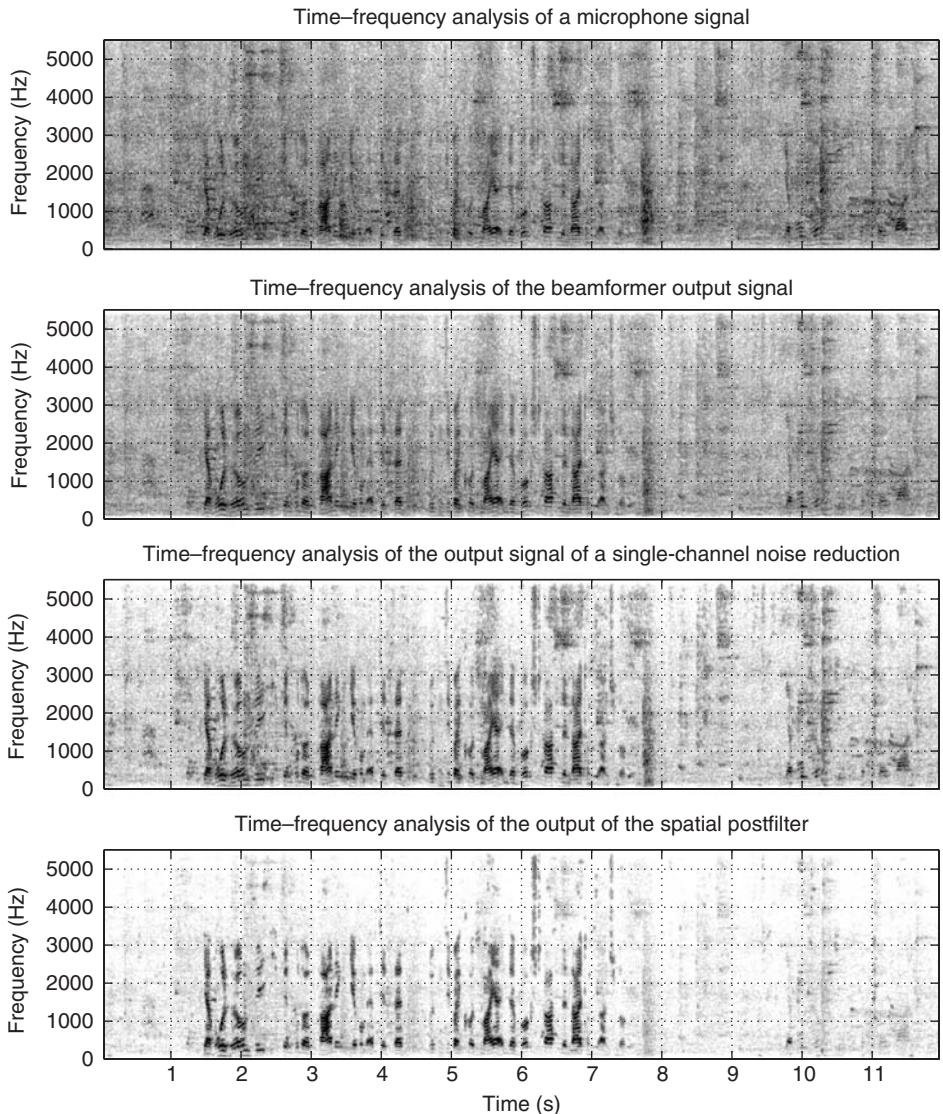
### 8.7.2 Signal Reconstruction

At high noise conditions the speech signal is sometimes distorted to such an extend that postfilter approaches are not able to work successfully. The idea behind signal reconstruction is to detect heavily perturbed speech signal components and replace those parts with *synthetic* speech. In order to generate a synthetic speech signal at least a couple of time–frequency bands with a sufficient SNR are required. However, this condition is fulfilled in the majority of cases such as in vehicles or offices. In [64] a frequency-domain approach for speech synthesis based on a *harmonic-plus-noise model* (HNM) is presented. Unlike this approach, we propose to generate the synthetic speech signal in the time domain based on a modified version of the well-known source–filter model [65].

Figure 8.15 shows an overview of the proposed speech reconstruction scheme. According to the source–filter model for speech generation, first a spectral envelope

**TABLE 8.2 Relative Gains in Terms of Word Error Rate with Respect to (w.r.t.) Different Signals within the Proposed System**

	GSC w. r. t. Single-Channel Wiener Filter (%)	GSC with MAP Postfilter w. r. t. GSC (%)	GSC with MAP Postfilter w. r. t. Single-Channel Wiener Filter (%)
Sidewalk cafe	43.87	56.33	75.49
Train station	47.88	51.10	74.51
Canteen	29.82	28.41	49.75

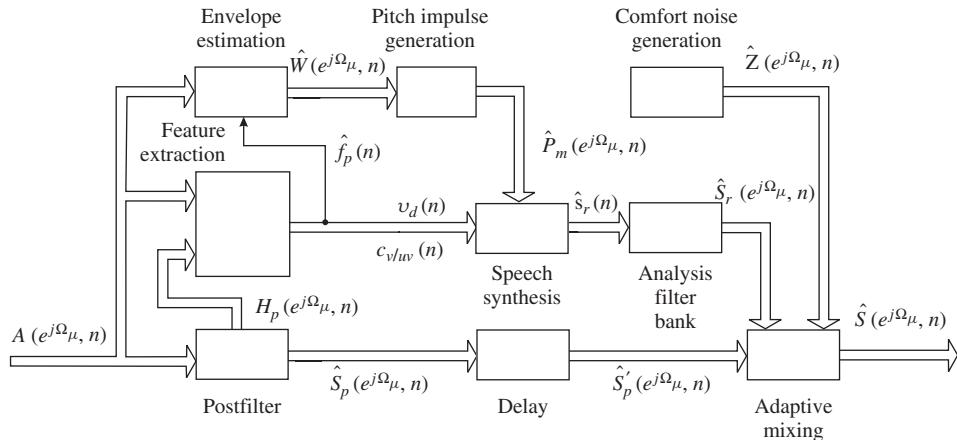


**Figure 8.14** Time–frequency analyses examples of a heavily corrupted microphone signal, the signal after a four-channel beamformer, and two postfilters (single-channel noise reduction and spatial postfiltering).

$\widehat{W}(e^{j\Omega_\mu}, n)$  is extracted from the beamformer signal and subsequently weighted by the spectrum of a generic pitch impulse  $\widehat{P}(e^{j\Omega_\mu})$ :

$$\widehat{P}_m(e^{j\Omega_\mu}, n) = \widehat{W}(e^{j\Omega_\mu}, n) \widehat{P}(e^{j\Omega_\mu}). \quad (8.54)$$

The spectral envelope approximates the behavior of the vocal tract. The spectrum  $\widehat{P}(e^{j\Omega_\mu})$  corresponds to a short impulse that was measured during a voiced section of



**Figure 8.15** Basic structure of the speech reconstruction algorithm.

a real speech signal, whereas the spectral envelope had been removed. We refer to this impulse as the pitch impulse prototype. The synthetic speech signal  $\hat{s}_r(n)$  is generated based on the modified pitch impulse prototype  $\hat{P}_m(e^{j\Omega_\mu n})$  and on extracted features from the input signal. The generated synthetic speech as well as artificial noise are combined with the postfiltered signal by an adaptive mixer.

**8.7.2.1 Envelope Estimation and Pitch Impulse Generation** For estimating the spectral envelope  $\hat{W}(e^{j\Omega_\mu n})$  a broad variety of different methods exist, such as *linear predictive coding* (LPC) or cepstral analysis [65]. An accurate estimation of the clean envelope can in principle be achieved at good SNR, whereas at high noise scenarios these approaches fail. However, better results could be accomplished by applying pitch-specific expert systems (codebooks). For the training of the codebooks the algorithm proposed by Linde–Buzo–Gray (LBG) [66] can be applied. To do that, several pitch-specific databases are utilized. For certain predefined pitch–frequency intervals (e.g., 70–90 Hz, ..., 290–310 Hz) different codebooks are trained individually. The pitch-specific databases consist of a large amount of energy-normalized spectral envelopes extracted from speech signals considering the Lombard effect [60]. Due to the normalization the codebooks become energy independent while the shapes of the envelopes are preserved.

In order to find the desired spectral envelope an estimate for the pitch frequency  $\hat{f}_p(n)$  as well as an estimate for a preliminary spectral envelope  $\hat{W}_s(e^{j\Omega_\mu n})$  are needed. For extracting  $\hat{f}_p(n)$  several algorithms exist, such as the analysis in the cepstral domain, the short-term autocorrelation [67], or harmonic product-spectrum-based schemes [68]. If these methods are applied to highly disturbed speech signals, however, no reliable estimation can be achieved anymore. An enhanced fundamental frequency estimation method is proposed in [69] that allows a reliable operation at low SNR scenarios even for very low fundamental frequencies. An estimate  $\hat{W}_s(e^{j\Omega_\mu n})$  can be determined by smoothing of the magnitudes of the beamformer along frequency and normalizing by the frame energy. From the pitch-specific codebook a spectral envelope entry  $\hat{W}_c(e^{j\Omega_\mu n})$  is chosen that has the best match to  $\hat{W}_s(e^{j\Omega_\mu n})$  at frequencies of

sufficient SNR. Depending on the current estimate for the a priori SNR  $\widehat{\xi}(\Omega_\mu, n)$  at the output of the postfilter, the energy-independent envelope  $\widehat{W}_u(e^{j\Omega_\mu}, n)$  is determined by

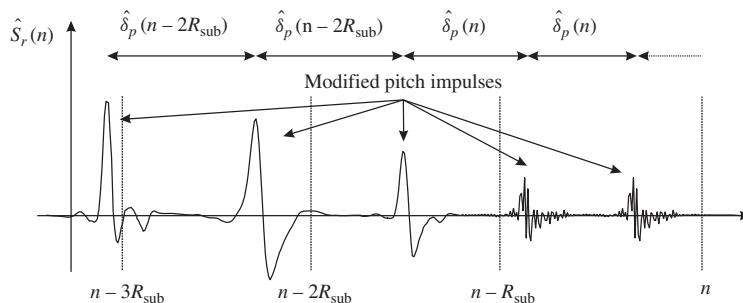
$$\widehat{W}_u(e^{j\Omega_\mu}, n) = \begin{cases} \widehat{W}_s(e^{j\Omega_\mu}, n), & \text{if } \widehat{\xi}(\Omega_\mu, n) > \text{SNR}_0, \\ \widehat{W}_c(e^{j\Omega_\mu}, n), & \text{else,} \end{cases} \quad (8.55)$$

whereas  $\text{SNR}_0$  denotes a suitable predefined level to which the current SNR is compared. The final estimate  $\widehat{W}(e^{j\Omega_\mu}, n)$  is obtained by de-normalizing  $\widehat{W}_u(e^{j\Omega_\mu}, n)$  with the current frame energy.

It is important to note that instead of using an excitation signal, as it is commonly done for speech coding [65], a predefined pitch impulse prototype spectrum is employed in Eq. (8.54). However, instead of using only a generic impulse, one could also use speaker- or pitch-specific impulses for higher improvements. Due to the elementwise weighting by the spectral envelope a modified pitch impulse spectrum  $\widehat{P}_m(e^{j\Omega_\mu}, n)$  results. Thereby, the power is adjusted to that of the input signal.

**8.7.2.2 Speech Synthesis** In order to synthesize a clean speech signal, the modified pitch spectrum is first transformed to the time domain using an IFFT. The resulting modified pitch impulse is added to those obtained from the preceding ones considering an appropriate time shift. The time shift is chosen according to the current pitch frequency:  $\hat{\delta}_p(n) = f_s / \hat{f}_p(n)$ . For the sake of clarity, an example is introduced in Figure 8.16 that shows a section of a synthetic speech signal over three subframes. A train of modified pitch impulses is depicted that, as mentioned before, vary depending on the current spectral envelope.

Furthermore, it should be ensured that no strong variations of the estimated parameters such as spectral envelope or pitch frequency exist. In addition, more features have to be extracted from the postfiltered signal such as voiced/unvoiced classification  $c_{v/u}(n)$  and speech activity detection  $v_d(n)$  [70] to achieve a high-quality speech synthesis. At transitions from voiced to unvoiced parts the synthesized signal amplitudes should be decreased slowly in order to avoid artifacts. For the same reason the pitch frequency should be smoothed, whereas at pitch onset first a maximum search over the postfiltered signal block is accomplished. The corresponding maximum lag is then



**Figure 8.16** Example of a synthesized speech signal over three subframes. Note that in the last subframe two pitch impulses are inserted and therefore  $\hat{\delta}_p(n)$  is used twice.

used to adjust the modified pitch impulse correctly in phase. The synthesized signal vector  $\hat{s}_r(n)$  is subsequently processed by a windowing function, for example, Hann or Hamming window [7], for smoothing of signal parts at the edges of the current frame.

**8.7.2.3 Adaptive Mixing** In order to combine the synthesized and the postfiltered signal adaptively in the frequency domain an FFT is computed to  $\hat{s}_r(n)$ . However, the signal reconstruction inserts a marginal delay in the signal path. Hence, for compensation a delayed version of the postfiltered spectrum  $\hat{S}'_p(e^{j\Omega_\mu}, n)$  has to be used. The mixing is performed such that synthesized parts are used for subbands exhibiting relatively high perturbations and the delayed postfiltered parts are used for the remaining ones:

$$\hat{S}'(e^{j\Omega_\mu}, n) = \alpha_{r,\mu}(n) \hat{S}'_p(e^{j\Omega_\mu}, n) + (1 - \alpha_{r,\mu}(n)) \hat{S}_r(e^{j\Omega_\mu}, n). \quad (8.56)$$

The weighting factors  $\alpha_{r,\mu}(n)$  can be chosen within the range [0,1], depending on the time-smoothed versions of  $\hat{\xi}(\Omega_\mu, n)$  and  $H_p(e^{j\Omega_\mu}, n)$ . These time-smoothed quantities are updated during speech presence only.

Often it sounds more comfortable if there is a certain amount of residual noise in the output signal. For this reason artificial noise called *comfort noise* may be inserted [71, 72]. Whenever the postfiltered signal  $\hat{S}'_p(e^{j\Omega_\mu}, n)$  is smaller than the background noise level  $\hat{S}_{bb}(\Omega_\mu, n)H_{\min}^2$ , the output signal is replaced by comfort noise. If reconstruction is performed, the spectrum  $\hat{S}(e^{j\Omega_\mu}, n)$  is combined with artificial noise. A noise generator is used to produce zero-mean white noise  $\tilde{Z}(e^{j\Omega_\mu}, n)$  with variance  $\sigma_{\tilde{Z},\mu}^2$ . The comfort noise is generated according to

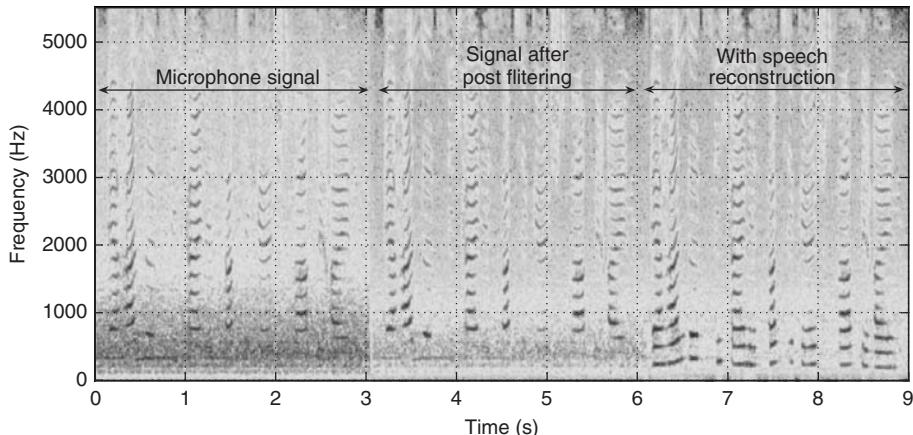
$$\hat{Z}(e^{j\Omega_\mu}, n) = \tilde{Z}(e^{j\Omega_\mu}, n) \sqrt{\frac{\hat{S}_{bb}(\Omega_\mu, n)H_{\min}^2}{\sigma_{\tilde{Z},\mu}^2}}. \quad (8.57)$$

The enhanced output spectrum is finally obtained as follows:

$$\hat{S}(e^{j\Omega_\mu}, n) = \alpha_{z,\mu}(n) \hat{S}'(e^{j\Omega_\mu}, n) + \sqrt{1 - \alpha_{z,\mu}^2(n)} \hat{Z}(e^{j\Omega_\mu}, n), \quad (8.58)$$

where the mixing constants  $\alpha_{z,\mu}(n)$  are determined analog to  $\alpha_{r,\mu}(n)$ . Note that Eq. (8.58) assumes independency of the spectra  $\hat{S}'(e^{j\Omega_\mu}, n)$  and  $\hat{Z}(e^{j\Omega_\mu}, n)$ .

**8.7.2.4 Simulation Example** To show the performance of the proposed method, a time-frequency analysis before and after processing of a noisy speech signal (measured in a car at high speed) is depicted in Figure 8.17. As can be seen, the harmonic structure at lower frequencies of the microphone signal is almost masked completely by the background noise. Note that at these highly disturbed signal parts only a maximum attenuation is applied using a postfilter. By employing the proposed speech reconstruction algorithm, the missing harmonic structure at lower frequencies is regenerated successfully. For that example, the reconstruction has only been performed for frequencies up to 900 Hz. An enhanced speech quality and intelligibility can be accomplished using the proposed method at high noise scenarios.



**Figure 8.17** Time–frequency analysis of a noisy speech signal (left), a postfiltered signal (center), and a postfiltered signal with reconstruction (right).

## 8.8 CONCLUSIONS

In this chapter we concentrate on a speech enhancement system (see Fig. 8.1) as it is currently built and introduced into consumer products. We give short descriptions of the subsystems and the problems that have to be solved. We also show possible alternatives. Most functions are enabled by multichannel/array processing only. The methods presented in this chapter are particularly suitable for practical applications as they comply with relevant requirements such as robustness and computational efficiency.

It is inherent to the class of problems that we have dealt with in speech and audio signal processing that there is not “the only one optimal” solution. Nevertheless, the system described provides considerable comfort to its users. Those are, however, never really satisfied. For example, they ask for higher speech/audio quality, for additional functions, and higher recognition rates. This together with more powerful hardware and software will certainly be the driving force for future developments.

## REFERENCES

1. E. Hänsler and G. Schmidt, *Acoustic Echo and Noise Control: A Practical Approach*, Wiley, Hoboken, NJ, 2004.
2. R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1983.
3. P. Vary, “Noise suppression by spectral magnitude estimation—Mechanism and theoretical limits,” *Signal Process.*, vol. 8, no. 4, pp. 387–400, 1985.
4. P. Vary, “An adaptive filterbank equalizer for speech enhancement,” *Signal Process.*, vol. 86, pp. 1206–1214, June 2006.
5. A. Sugiyama, T. P. Hua, M. Kato, and M. Serizawa, “Noise suppression with synthesis windowing and pseudo noise injection,” *Proc. IEEE ICASSP '02*, pp. 545–548, 2002.
6. “Transmission planning aspects of the speech service in the GSM public land mobile network (PLMS) system,” ETS 300 903 (GSM 03.50), European Telecommunications Standards Institute, France, 1999.

7. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.
8. S. Haykin, *Adaptive Filter Theory*, 4th ed., Prentice Hall, Englewood Cliffs, NJ, 2002.
9. S. Gay and S. Travathia, "The fast affine projection algorithm," *Proc. ICASSP '95*, vol. 3, pp. 3023–3027, 1995.
10. A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley, Hoboken, NJ, 2003.
11. G. Enzner and P. Vary, "Robust and elegant, purely statistical adaptation of acoustic echo canceler and postfilter," *Proc. IWAENC '03*, pp. 43–46, 2003.
12. Y. Joncour, A. Sugiyama, and A. Hirano, "DSP implementations and performance evaluation of a stereo echo canceller with pre-processing," *Proc. EUSIPCO '98*, vol. 2, pp. 981–984, 1998.
13. A. Sugiyama, Y. Joncour, and A. Hirano, "A stereo echo canceller with correct echo-path identification based on an input-sliding technique," *IEEE Trans. Signal Process.*, vol. 49, no. 1, pp. 2577–2587, 2001.
14. A. Gilloire and V. Turbin, "Using auditory properties to improve the behaviour of stereophonic acoustic echo cancellation," *Proc. ICASSP '98*, vol. 6, pp. 3681–3684, 1998.
15. M. M. Sondhi and D. R. Morgan, "Stereophonic acoustic echo cancellation—An overview of the fundamental problem," *IEEE Signal Process. Lett.*, vol. 2, no. 8, pp. 148–151, 1995.
16. Y. Huang, J. Benesty, and G. W. Elko, "Microphone arrays for video camera steering, in *Acoustic Signal Processing for Telecommunication*, S. L. Gay and J. Benesty (Eds.), Kluwer Academic, Boston, 2001, pp. 239–260.
17. J. H. DiBiase, H. F. Siverman, and M. S. Brandstein, "Robust source localization in reverberant rooms," in *Microphone Arrays*, M. S. Brandstein and D. Ward (Eds.), Springer, Berlin, 2001, pp. 157–180.
18. R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propagat.*, vol. AP-34, no. 3, Mar. 1986.
19. R. Kumaresan, "Spectral analysis," in *Handbook for Digital Signal Processing*, S. K. Mitra and J. F. Kaiser (Eds.), Wiley, Hoboken, NJ, 1993, pp. 1143–1242.
20. C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 24, no. 4, pp. 320–327, 1976.
21. J. Benesty, "Adaptive eigenvalue decomposition algorithm for passive acoustic source localization," *J. Acoust. Soc. Am.*, vol. 107, no. 1, pp. 384–391, Jan. 2000.
22. G. Doblinger, "Localization and tracking of acoustical sources," in *Topics in Acoustic Echo and Noise Control*, E. Hänsler and G. Schmidt (Eds.), Springer, Berlin, 2006, pp. 91–122.
23. T. Wolff, M. Buck, and G. Schmidt, "A subband based source localization system for reverberant environments," *Proc. ITG '08*, 2008.
24. D. H. Johnson and D. E. Dudgeon, *Array Signal Processing—Concepts and Techniques*, Prentice Hall, Englewood Cliffs, NJ, 1993.
25. J. L. Flanagan, D. A. Berkley, G. W. Elko, J. E. West, and M. M. Sondhi, "Autodirective microphone systems," *Acustica*, vol. 73, pp. 58–71, 1991.
26. J. Bitzer and K. U. Simmer, "Superdirective microphone arrays," in *Microphone Arrays*, M. Brandstein and D. Ward (Eds.), Springer, Berlin, 2001, pp. 19–38.
27. H. Cox, R. M. Zeskind, and M. M. Owen, "Robust adaptive beamforming," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 35, no. 10, pp. 1365–1375, 1987.
28. E. N. Gilbert and S. P. Morgan, "Optimum design of directive antenna arrays subject to random variation," *Bell Syst. Tech. J.*, vol. 34, pp. 637–663, 1955.
29. O. L. Frost, III, "An algorithm for linearly constrained adaptive array processing," *Proc. IEEE*, vol. 60, no. 8, pp. 926–935, 1972.

30. L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. Antennas Propagat.*, vol. 30, no. 1, pp. 24–34, 1982.
31. C. W. Jim, "A comparison of two LMS constrained optimal array structures," *Proc. IEEE*, vol. 65, no. 12, pp. 1730–1731, 1977.
32. O. Hoshuyama, A. Sugiyama, and A. Hirano, "A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2677–2684, 1999.
33. B. Widrow, K. M. Duvall, R. P. Gooch, and W. C. Newman, "Signal cancellation phenomena in adaptive antennas: Causes and cures," *IEEE Trans. Antennas Propagat.*, vol. 30, no. 3, pp. 469–478, 1982.
34. D. Van Compernolle, "Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings," *Proc. ICASSP '90*, vol. 2, pp. 833–836, 1990.
35. W. Herboldt, S. Nakamura, and W. Kellermann, "Joint optimization of LCMV beamforming and acoustic echo cancellation for automatic speech recognition," *Proc. ICASSP '05*, vol. 3, pp. 77–80, 2005.
36. W. H. Neo and B. Farhang-Boroujeny, "Robust microphone arrays using subband adaptive filters," *Proc. ICASSP '01*, vol. 6, pp. 3721–3724, 2001.
37. S. Gannot, D. Burshstein, and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1614–1626, 2001.
38. M. Buck, T. Haulick, and H.-J. Pfleiderer, "Self-calibrating microphone arrays for speech signal acquisition: A systematic approach," *Signal Process.*, vol. 86, no. 6, pp. 1230–1238, 2006.
39. W. Herboldt and W. Kellermann, "GSAEC—Acoustic echo cancellation embedded into the generalized sidelobe canceller," *Proc. EUSIPCO '00*, vol. 3, pp. 1843–1846, 2000.
40. W. Kellermann, "Acoustic echo cancellation for beamforming microphone arrays," in *Microphone Arrays*, M. Brandstein and D. Ward (Eds.), Springer, Berlin, 2001, pp. 281–306.
41. S. Doclo, M. Moonen, and E. De Clippel, "Combined acoustic echo and noise reduction using GSVD-based optimal filtering," *Proc. ICASSP '00*, vol. 2, pp. 1051–1054, 2000.
42. W. Herboldt, W. Kellermann, and S. Nakamura, "Joint optimization of acoustic echo cancellation and adaptive beamforming," in *Topics in Acoustic Echo and Noise Control*, E. Hänsler and G. Schmidt (Eds.), Springer, Berlin, 2006.
43. Z. Liu, M. L. Seltzer, A. Acero, I. Tashev, Z. Zhang, and M. Sinclair, "A compact multi-sensor headset for hands-free communication," *Proc. WASPAA '05*, pp. 138–141, 2005.
44. S. Nordholm, I. Claesson, and M. Dahl, "Adaptive microphone array employing calibration signals: An analytical evaluation," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 3, pp. 241–252, 1999.
45. X. Zhang and J. H. L. Hansen, "CSA-BF: Novel constrained switched adaptive beamforming for speech enhancement and recognition in real car environments," *Proc. ICASSP 03*, vol. 2, pp. 125–128, 2003.
46. T. P. Hua, A. Sugiyama, and G. Faucon, "A new self-calibration technique for adaptive microphone arrays," *Proc. IWAENC '05*, pp. 237–240, 2005.
47. P. Oak and W. Kellermann, "A calibration algorithm for robust generalized sidelobe cancelling beamformers," *Proc. IWAENC '05*, pp. 97–100, 2005.
48. M. Buck, T. Haulick, and H.-J. Pfleiderer, "Microphone calibration for multi-channel signal processing," in *Topics in Speech and Audio Processing in Adverse Environments*, E. Hänsler and G. Schmidt (Eds.), Springer, Berlin, 2008.

49. G. W. Elko, "Superdirective microphone arrays," in *Acoustic Signal Processing for Telecommunication*, S. L. Gay and J. Benesty (Eds.), Kluwer, Boston, MA, 2000, pp. 181–237.
50. S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 27, no. 2, pp. 113–120, 1979.
51. Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator," *IEEE Trans. Acoustics Speech Signal Process.*, vol. 32, no. 6, pp. 1109–1121, 1984.
52. T. Lotter and P. Vary, "Speech enhancement by map spectral amplitude estimation using a super-Gaussian speech model," *EURASIP J. Appl. Signal Process.*, pp. 1110–1126, July 2005.
53. K. Linhard and T. Haulick, "Spectral noise subtraction with recursive gain curves," *Proc. ICSLP '98*, vol. 4, pp. 1479–1482, 1998.
54. R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 5, pp. 504–512, 2001.
55. E. Habets, Multi-channel speech dereverberation based on a statistical model of late reverberation, *Proc. ICASSP 05*, vol. 4, pp. 173–176, 2005.
56. K. Lebart and J. M. Boucher, "A new method based on spectral subtraction for speech dereverberation," *Acustica*, vol. 87, pp. 359–366, 2001.
57. I. Tashev and D. Allred, "Reverberation reduction for improved speech recognition," *Proc. HSCMA '05*, pp. 18–19, 2005.
58. H. Kuttruff, *Room Acoustics*, 4th ed., Spon Press, London, 2000.
59. M. Buck and A. Wolf, "Model-based dereverberation of single-channel speech signals," *Proc. DAGA '08*, 2008.
60. J. C. Junqua, "The influence of acoustics on speech production: A noise-induced stress phenomenon known as the Lombard reflex," *Speech Commun.*, vol. 20, no. 1, pp. 13–22, 1996.
61. K. U. Simmer, J. Bitzer, and C. Marro, "Post-filtering techniques," in *Microphone Arrays*, M. Brandstein and D. Ward (Eds.), Springer, Berlin, 2001, pp. 39–60.
62. I. Cohen, S. Gannot, and B. Berdugo, "An integrated real-time beamforming and post-filtering system for nonstationary noise environments," *EURASIP J. Appl. Signal Process.*, pp. 1064–1073, Nov. 2003.
63. T. Wolff and M. Buck, "Spatial maximum a posteriori post-filtering for arbitrary beam-forming," *Proc. HSCMA '08*, pp. 53–56, 2008.
64. E. Zavarehei, S. Vaseghi, and Q. Yan, "Noisy speech enhancement using harmonic-noise model and codebook-based post-processing," *IEEE Trans. Speech Audio Process.*, vol. 15, no. 4, pp. 1194–1203, 2007.
65. P. Vary and R. Martin, *Digital Speech Transmission*, Wiley, Hoboken, NJ, 2006.
66. Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, no. 1, pp. 84–95, 1980.
67. W. Hess, *Pitch Determination of Speech Signals*, Springer, Berlin, 1983.
68. M. R. Schroeder, "Period histogram and product spectrum: New methods for fundamental frequency measurements," *J. Acoust. Soc. Am.*, vol. 43, no. 4, pp. 829–834, 1968.
69. M. Krini and G. Schmidt, "Spectral refinement and its application to fundamental frequency estimation," *Proc. IEEE WASPAA '07*, pp. 251–254, 2007.
70. H. Puder and O. Soffke, "An approach for an optimized voice-activity detector for noisy speech signals," *Proc. EUSIPCO 02*, pp. 243–246, 2002.

71. D. Hartmann, "Noise and voice quality in VoIP environments," in *Noise Reduction in Speech Applications*, G. M. Davis (Ed.), CRC Press, Boca Raton, FL, 2002, pp. 277–304.
72. S. J. Leese, "Echo cancellation," in *Noise Reduction in Speech Applications*, G. M. Davis (Ed.), CRC Press, Boca Raton, FL, 2002, pp. 199–216.
73. D. Van Compernolle and S. Van Gerven, "Beamforming with microphone arrays," in *Digital Signal Processing for Telecommunications*, A. R. Figueiras-Vidal (Ed.), Cost 229, 1995, pp. 107–131.
74. G. W. Elko, "Microphone array systems for hands-free telecommunication," *Speech Commun.*, vol. 20, pp. 229–240, 1996.
75. W. Herboldt and W. Kellermann, "Computationally efficient frequency-domain robust generalized sidelobe canceller," *Proc. IWAENC '01*, pp. 51–55, 2001.
76. P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, Englewood Cliffs, NJ, 1992.

## CHAPTER 9

---

# Acoustic Beamforming for Hearing Aid Applications

Simon Doclo<sup>1</sup>, Sharon Gannot<sup>2</sup>, Marc Moonen<sup>3</sup>, and Ann Sprriet<sup>3</sup>

<sup>1</sup>University of Oldenburg, Signal Processing Group, Oldenburg, Germany

<sup>2</sup>Bar-Ilan University, School of Engineering, Ramat-Gan, Israel

<sup>3</sup>Katholieke Universiteit Leuven, Dept. of Electrical Engineering, Leuven, Belgium

## 9.1 INTRODUCTION

Noise reduction algorithms in hearing aids are crucial for hearing-impaired persons to improve speech intelligibility in background noise (e.g., traffic, cocktail party situation). Many hearing aids currently have more than one microphone, enabling the use of multimicrophone speech enhancement algorithms [1]. In comparison with single-microphone algorithms, which can only use spectral and temporal information, multimicrophone algorithms can additionally exploit the spatial information of the sound sources. This generally results in a higher performance, especially when the speech and the noise sources are spatially separated.

Since many hearing impaired have a hearing loss at both ears, they are fitted with a hearing aid at each ear. In a so-called bilateral system, no cooperation between the hearing aids takes place. Current noise reduction algorithms in bilateral hearing aids are not designed to preserve the binaural localization cues, that is, the interaural time difference (ITD) and the interaural level difference (ILD) [2]. These binaural cues play an important role in sound localization and speech segregation in noisy environments [3–7]. In order to achieve true binaural processing, both hearing aids need to cooperate with each other (e.g., through a wireless link) such that a binaural hearing aid can be considered a simple acoustic sensor network. The objective of a binaural signal enhancement algorithm then is not only to selectively extract the useful speech signal and suppress background noise but also to preserve the binaural cues of the sound sources, so as to preserve the auditory impression of the acoustic scene and exploit the binaural hearing advantage.

Section 9.2 gives an overview of several multimicrophone noise reduction techniques for hearing aids. In the remainder of the chapter, we focus on two specific

techniques—adaptive beamforming and multichannel Wiener filtering (MWF)—and their use in monaural and binaural hearing aids. In Section 9.3 monaural beamforming techniques are discussed. We derive the standard generalized side-lobe canceler (GSC) and the transfer function GSC from the minimum-variance distortionless response beamformer, and we present the MWF and its relationship with the transfer function GSC. In Section 9.4 MWF-based techniques for binaural hearing aids are discussed. We present extensions of the standard binaural MWF that specifically aim to preserve the binaural cues of the sound sources.

## 9.2 OVERVIEW OF NOISE REDUCTION TECHNIQUES

In the last decades, several multimicrophone speech enhancement algorithms for monaural and binaural hearing aids have been proposed, for example, based on fixed and adaptive beamforming, MWF, blind source separation (BSS), and computational auditory scene analysis (CASA). Each class of algorithms has its own advantages and limitations.

1. *Fixed Beamforming* Fixed beamformers combine the microphone signals using a time-invariant filter-and-sum operation and are hence data independent. The objective of a fixed beamformer is to obtain spatial focusing on the desired speech source, thereby reducing background noise not coming from the direction of the speech source. Different types of fixed beamformers exist, for example, delay-and-sum beamforming, superdirective beamforming, differential microphone arrays, and frequency-invariant beamforming [8–14]. A specific type of fixed beamformers, namely matched filtering, will be discussed in more detail in Section 9.3.2. For the design of fixed beamformers, the direction of the speech source and the complete microphone configuration need to be known. Hence, fixed beamformers have mainly been used for *monaural* hearing aids [15–17], although for *binaural* hearing aids fixed-beamforming techniques have also been proposed that aim to combine spatial selectivity and noise reduction with the preservation of the binaural cues of the speech source [18–21].

2. *Adaptive Beamforming* In practice, since the background noise is unknown and can change both spectrally and spatially, information about the noise field has to be adaptively estimated. Adaptive beamformers combine the spatial focusing of fixed beamformers with adaptive noise suppression (e.g., adaptively steering a null in the direction of the dominant noise sources [22]). Hence, they generally exhibit a higher noise reduction performance than fixed beamformers.

In a minimum-variance distortionless response (MVDR) beamformer [23, 24], the energy of the output signal is minimized under the constraint that signals arriving from the assumed direction of the desired speech source are processed without distortion. A widely studied adaptive implementation of this beamformer is the GSC [70]. The standard GSC consists of a spatial preprocessor, that is, a fixed beamformer and a blocking matrix, combined with a (multichannel) adaptive noise canceler (ANC). The fixed beamformer provides a spatial focus on the speech source, creating a so-called speech reference; the blocking matrix steers nulls in the direction of the speech source, creating so-called noise references; and the ANC eliminates the noise components in the speech reference that are correlated with the noise references. Due to room reverberation, microphone mismatch, and look direction error, speech components may however leak into the noise references of the standard GSC, giving rise to speech

distortion and possibly signal cancellation. Several techniques have been proposed to limit the speech distortion resulting from this speech leakage by reducing the speech leakage components in the noise references [25–32] and by limiting the distorting effect of the remaining speech leakage components [33–39]. Several of these issues will be discussed in more detail in Section 9.3.

The GSC or one of its more robust variants is a widely used multimicrophone noise reduction technique for monaural hearing aids with an endfire microphone array configuration [22, 40–42, 71]. In an effort to combine adaptive noise reduction with binaural processing, adaptive beamforming techniques producing a binaural output signal have also been proposed. In [43] the frequency spectrum is divided into a low-pass and a high-pass portion, and the low-pass portion is passed through unaltered in order to preserve the ITD cues of the speech source, while adaptive noise reduction is performed only for the high-pass portion. Other algorithms restrict the preservation of the binaural cues to an angular region around the frontal direction while reducing background noise for the other angles [44].

*3. Multichannel Wiener Filtering* In [45] an MWF technique has been proposed that produces a minimum mean-square error (MMSE) estimate of the desired speech component in one of the microphone signals, hence simultaneously performing noise reduction and limiting speech distortion. In addition, the MWF is able to take speech distortion into account in its optimization criterion, resulting in the speech-distortion-weighted MWF (SDW-MWF). The SDW-MWF is uniquely based on estimates of the second-order statistics of the recorded speech signal and the noise signal and hence requires a method that determines time–frequency regions where the desired source is dominant and time–frequency regions where the interference is dominant. The SDW-MWF has been successfully applied as a speech enhancement technique in monaural multimicrophone hearing aids [46, 47]. The monaural SDW-MWF and its relationship with adaptive beamforming will be discussed in more detail in Section 9.3.5.

Since the SDW-MWF produces an estimate of the speech component in the microphone signals and does not make any assumptions regarding the microphone configuration and the room impulse responses relating the speech source and the microphones, it is obviously well suited to combine noise reduction with binaural processing. It was shown in [48, 49] that the binaural MWF perfectly preserves the binaural cues of the speech component but undesirably changes the noise cues to those of the speech component. Several extensions have been proposed to preserve the binaural cues of both the speech and the noise component, either by partial noise estimation [49] or by extending the MWF cost function with terms related to the ITD, ILD, or interaural transfer function [48, 50, 51]. In addition, recently a distributed version of the binaural SDW-MWF has been presented [52]. The binaural MWF and its extensions will be discussed in more detail in Section 9.4.

*4. Blind Source Separation* The signals received at the microphones can essentially be considered a mixture of all sound sources filtered by the respective room impulse responses between the sound source and the microphones. The goal of BSS is to recover all original signals. Many BSS algorithms exploit the independence and the non-Gaussianity of the sources, enabling the use of, for example, independent-component analysis (ICA) techniques [53, 54]. While time-domain ICA-based techniques are well suited to solve the instantaneous mixing problem, they are not able to address the convolutive mixture problem encountered in typical reverberant

environments. By considering the BSS problem in the frequency-domain, the convolutive mixing problem can be transformed into an instantaneous mixing problem for each frequency bin. An inherent permutation and amplitude/phase scaling problem however occurs in frequency-domain BSS approaches, for which several solutions have been proposed [55–57]. Nevertheless, due to its computational complexity the use of BSS techniques in hearing aids has found only limited practical interest.

*5. Computational Auditory Scene Analysis* Algorithms based on CASA aim to perform sound segregation by modeling the human auditory perceptual processing [5, 58–60]. A typical CASA model involves two stages. In the first stage, a time–frequency representation of the incoming mixture of signals is generated, for example, using short-time Fourier transform (STFT) processing or by incorporating a more advanced cochlear model. In the second stage, the resulting time–frequency elements are grouped into separate perceptual streams based on distinctive perceptual cues. As summarized in [5], some cues characterize the monaural acoustic properties of the sources, such as common pitch, amplitude modulation, and onset [61, 62]. In addition, for a binaural system sound sources can also be distinguished based on their spatial direction information, for example, ITD, ILD, and interaural envelope difference [63–65]. A gain factor is applied to each time–frequency element such that regions dominated by the desired sound stream receive a high gain and regions dominated by other streams receive a low gain.

### 9.3 MONAURAL BEAMFORMING

#### 9.3.1 Problem Formulation

Consider an array of  $M$  microphones in a noisy and reverberant environment. The received signals consist of two components: a desired speech signal and an undesired noise signal. The observed signal  $x_m(n)$  at the  $m$ th microphone is hence given by

$$x_m(n) = \sum_{i=0}^{L_h} h_{m,i}(n)s(n-i) + v_m(n) = s_m(n) + v_m(n), \quad m = 1, \dots, M, \quad (9.1)$$

where  $s_m(n)$  and  $v_m(n)$  represent the speech and the noise component at the  $m$ th microphone and  $n$  is the time index. It is assumed that the noise statistics are slowly time varying, while the speech signal is nonstationary. The noise signal may comprise coherent (directional) as well as noncoherent (diffuse and uncorrelated) noise components. The room impulse responses (RIRs)  $h_{m,i}(n)$ , relating the speech source with the  $m$ th microphone, are in general time-varying finite-impulse response (FIR) filters of order  $L_h$ , where  $i = 0, \dots, L_h$  is the filter tap index. We will assume in the remainder of the chapter that the RIRs are slowly time varying, that is,  $h_{m,i}(n) \approx h_{m,i}$ .

The observed signals are analyzed using the STFT, yielding

$$x_m(k, \ell) \approx h_m(k)s(k, \ell) + v_m(k, \ell), \quad (9.2)$$

where  $x_m(k, \ell)$ ,  $s(k, \ell)$ , and  $v_m(k, \ell)$  are the STFT of the respective time-domain signals<sup>1</sup>,  $\ell$  represents the frame index, and  $k = 0, 1, \dots, K - 1$  represents the frequency

<sup>1</sup>We use lowercase symbols to denote both time- and frequency-domain signals. The signals are distinguished by their functional dependency: Time-domain signals depend on the discrete-time index  $n$ , whereas frequency-domain signals depend on the frame index  $\ell$  and the frequency bin index  $k$  (which are frequently omitted).

bin index<sup>2</sup>. Here,  $h_m(k)$ , the Fourier transform of the RIR  $h_{m,i}$ , is denoted the acoustical transfer function (ATF). It is clearly seen that in the STFT representation each frequency bin can be treated separately. Hence, for the compactness of the exposition, the dependency on the frequency bin index  $k$  will be omitted from this point on. In addition, the dependency on the frame index  $\ell$  will be omitted, except for the adaptive algorithms and the relative transfer function estimation in Section 9.3.4.

The equation set (9.2) can be stated in vector form as

$$\mathbf{x} = \mathbf{h}s + \mathbf{v}, \quad (9.3)$$

with

$$\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_M]^T, \quad (9.4)$$

$$\mathbf{h} = [h_1 \ h_2 \ \cdots \ h_M]^T, \quad (9.5)$$

$$\mathbf{v} = [v_1 \ v_2 \ \cdots \ v_M]^T. \quad (9.6)$$

A beamformer, depicted in Figure 9.1, is a system processing each of the microphone signals  $x_m$  by the filters  $w_m^*$  and summing the outputs. Define

$$\mathbf{w}^H = [w_1^* \ w_2^* \ \cdots \ w_M^*], \quad (9.7)$$

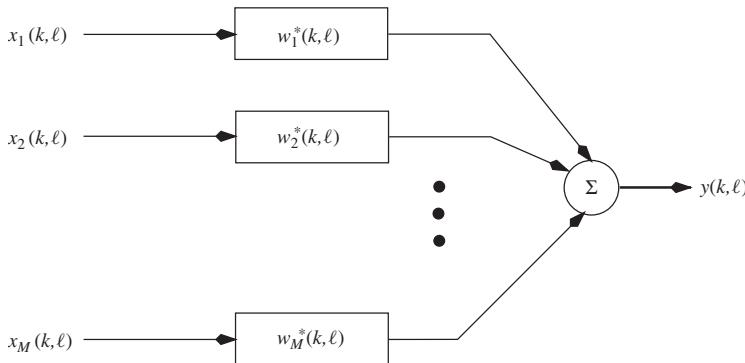
where  $H$  denotes the conjugate transpose. The beamformer output in the STFT domain is then given by

$$y = \mathbf{w}^H \mathbf{x} = \mathbf{w}^H \mathbf{h}s + \mathbf{w}^H \mathbf{v} \triangleq y_s + y_v, \quad (9.8)$$

where  $y_s$  and  $y_v$  represent the speech and noise components at the output of the beamformer. Finally, the output signal in the time-domain  $y(n)$  is reconstructed using the inverse short-time Fourier transform (iSTFT).

The signal-to-noise ratio (SNR) of the beamformer output is defined as

$$\text{SNR}^{\text{out}} \triangleq \frac{E\{|y_s|^2\}}{E\{|y_v|^2\}} = \frac{|\mathbf{w}^H \mathbf{h}|^2 \phi_{ss}}{\mathbf{w}^H \Phi_{vv} \mathbf{w}}, \quad (9.9)$$



**Figure 9.1** Filter-and-sum beamformer.

<sup>2</sup>Equality in (9.2) is only valid for segments that are significantly longer than the RIR length, i.e., the number of frequency bins  $K$  is assumed to be larger than the STFT segment length. Since RIRs tend to be very long, the conditions allowing for this representation to hold cannot be exactly met. We assume, however, that the STFT relation is a reasonable approximation. For a detailed discussion we refer to [106]

with  $\phi_{ss} = E\{|s|^2\}$  the power spectral density (PSD) of the speech signal and  $\Phi_{vv} \triangleq E\{\mathbf{v}\mathbf{v}^H\}$  the correlation matrix of the noise components measured at the microphone signals.

### 9.3.2 Data-Independent Beamformer

The simplest beamformer structure is the *delay-and-sum beamformer*, which first compensates for the relative delay between the microphone signals and then sums the steered signals to form a single output. This beamformer, which is still widely used, can be very effective in mitigating noncoherent (i.e., spatially white) noise sources provided that the number of microphones is relatively high. However, if the noise source is coherent, the noise reduction performance is strongly dependent on the direction of arrival of the noise source. Consequently, the performance of the delay-and-sum beamformer in reverberant environments is often insufficient. The delay-and-sum concept has been extended by introducing the *matched-filter beamformer* (MBF) [12, 66, 67]. This structure, designed for multipath environments such as reverberant enclosures, replaces the simple delay compensator by a matched filter, that is,

$$\mathbf{w}_0 = \frac{\mathbf{h}}{\|\mathbf{h}\|^2}. \quad (9.10)$$

Since the array beam pattern of the delay-and-sum beamformer and the MBF do not depend on the measured signals, they are often referred to as *data-independent* beamformers. It should be noted that the application of data-independent design methods is very limited in dynamic acoustical environments.

### 9.3.3 Minimum-Variance Distortionless Response (MVDR) Beamformer

The MVDR beamformer, proposed by Capon [23], constrains the response of the beamformer in the desired look direction while minimizing the response in all other directions. The MVDR concept was extended to satisfy a set of linear constraints, resulting in the linearly constrained minimum-variance (LCMV) beamformer (for details we refer to [33, 68]). Frost [24] proposed an adaptive form of Capon's beamformer which relies on the assumption that the RIRs, relating the desired source and the microphones, can be uniquely represented by gain and delay values.

In this section, we derive the MVDR beamformer for the general reverberant case and show that under the distortionless response constraint it is equivalent to the maximum signal-to-noise ratio (MSNR) beamformer.

**9.3.3.1 Derivation** Using (9.8), the output PSD of the beamformer is given by

$$E\{|y|^2\} = E\{\mathbf{w}^H \mathbf{x} \mathbf{x}^H \mathbf{w}\} = \mathbf{w}^H \Phi_{xx} \mathbf{w}, \quad (9.11)$$

where  $\Phi_{xx} \triangleq E\{\mathbf{x} \mathbf{x}^H\}$  is the correlation matrix of the received microphone signals. We want to minimize the output PSD subject to the constraint that  $y_s$ , the speech component at the beamformer output, is equal to the speech source signal  $s$ , that is,

$$y_s = \mathbf{w}^H \mathbf{h} s \stackrel{\text{constrain}}{=} s. \quad (9.12)$$

Hence, the constrained minimization problem can be stated as

$$\min_{\mathbf{w}} \mathbf{w}^H \Phi_{xx} \mathbf{w} \quad \text{subject to } \mathbf{w}^H \mathbf{h} = 1. \quad (9.13)$$

To solve (9.13), we first define the complex Lagrangian, that is,

$$\mathcal{L} = \mathbf{w}^H \Phi_{xx} \mathbf{w} + \lambda [\mathbf{w}^H \mathbf{h} - 1] + \lambda^* [\mathbf{h}^H \mathbf{w} - 1], \quad (9.14)$$

where  $\lambda$  is a Lagrange multiplier. Setting the derivative with respect to  $\mathbf{w}^*$  to zero yields

$$\nabla_{\mathbf{w}^*} \mathcal{L} = \Phi_{xx} \mathbf{w} + \lambda \mathbf{h} = 0. \quad (9.15)$$

Imposing the constraint in (9.13), we obtain the MVDR filter

$$\mathbf{w}_{\text{MVDR}} = \frac{\Phi_{xx}^{-1} \mathbf{h}}{\mathbf{h}^H \Phi_{xx}^{-1} \mathbf{h}}. \quad (9.16)$$

**9.3.3.2 Equivalence between MVDR and MSNR Beamformer** It is interesting to show the equivalence between the MVDR solution (9.16) and the MSNR beamformer [33], which is obtained by maximizing the output SNR in (9.9). The well-known solution to this maximization is the (colored noise) matched filter, that is,  $\mathbf{w} \propto \Phi_{vv}^{-1} \mathbf{h}$ . If the array response is constrained to satisfy  $\mathbf{w}^H \mathbf{h} = 1$ , that is, signals impinging the array from the look direction suffer no distortion, we obtain

$$\mathbf{w}_{\text{MSNR}} = \frac{\Phi_{vv}^{-1} \mathbf{h}}{\mathbf{h}^H \Phi_{vv}^{-1} \mathbf{h}}. \quad (9.17)$$

Using (9.3), it is evident that

$$\Phi_{xx} = \phi_{ss} \mathbf{h} \mathbf{h}^H + \Phi_{vv}. \quad (9.18)$$

To show the equivalence of the MVDR beamformer (9.16) and the MSNR beamformer (9.17), we apply the matrix inversion lemma for inverting  $\Phi_{xx}$ , that is,

$$\Phi_{xx}^{-1} = (\phi_{ss} \mathbf{h} \mathbf{h}^H + \Phi_{vv})^{-1} = \Phi_{vv}^{-1} - \frac{\phi_{ss} \Phi_{vv}^{-1} \mathbf{h} \mathbf{h}^H \Phi_{vv}^{-1}}{1 + \rho} \quad (9.19)$$

with

$$\rho = \phi_{ss} \mathbf{h}^H \Phi_{vv}^{-1} \mathbf{h} \quad (9.20)$$

such that

$$(\phi_{ss} \mathbf{h} \mathbf{h}^H + \Phi_{vv})^{-1} \mathbf{h} = \frac{\Phi_{vv}^{-1} \mathbf{h}}{1 + \rho}, \quad (9.21)$$

$$\mathbf{h}^H (\phi_{ss} \mathbf{h} \mathbf{h}^H + \Phi_{vv})^{-1} \mathbf{h} = \frac{\mathbf{h}^H \Phi_{vv}^{-1} \mathbf{h}}{1 + \rho}. \quad (9.22)$$

Dividing (9.21) by (9.22), we obtain

$$\mathbf{w}_{\text{MVDR}} = \frac{\Phi_{\mathbf{v}\mathbf{v}}^{-1} \mathbf{h}}{\mathbf{h}^H \Phi_{\mathbf{v}\mathbf{v}}^{-1} \mathbf{h}} \equiv \mathbf{w}_{\text{MSNR}}. \quad (9.23)$$

Hence, using (9.9), also the output SNR of both beamformers is identical, that is,

$$\text{SNR}_{\text{MVDR}}^{\text{out}} = \text{SNR}_{\text{MSNR}}^{\text{out}} = \phi_{ss} \mathbf{h}^H \Phi_{\mathbf{v}\mathbf{v}}^{-1} \mathbf{h} = \rho, \quad (9.24)$$

which obviously depends on the input SNR and the spatial separation between the speech source and the noise sources. While both MSNR and MVDR beamformers are shown to be equivalent, provided that the ATFs  $\mathbf{h}$  are known, their behavior in case of unknown ATFs is different and has been analyzed in [69].

### 9.3.4 Frequency-Domain Generalized Side-Lobe Canceler

Griffiths and Jim [70] proposed a method to split the constrained minimization (9.13) in the MVDR formulation into two orthogonal operations. Their beamformer structure is referred to as the GSC and consists of two branches. The first branch, satisfying the look direction constraint, is a delay-and-sum beamformer. The second branch, which corresponds to an unconstrained minimization of the beamformer output power, mitigates all signals impinging the array from directions other than the look direction. Breed and Strauss [72] proved that an equivalent GSC structure also exists for the more general LCMV beamformer.

In this section, we reformulate the derivation of Griffiths and Jim in the frequency-domain for arbitrary ATFs relating the speech source and the microphones. A more detailed description of the resulting transfer function GSC (TF-GSC) can be found in [27]. This GSC structure is also used in [73, 107] to deal with the more involved problem, in which additional non-stationary interference signals are received by the array. By imposing additional linear constraints on the beamformer output, the resulting beamformers enhance the interference cancellation.

**9.3.4.1 Derivation** Consider the  $M \times M$ -dimensional full-rank matrix  $\mathbf{U} = [\mathbf{h} \ \mathbf{B}]$ , where the columns of the matrix  $\mathbf{B}$  span the null space of  $\mathbf{h}$ , that is,

$$\mathbf{h}^H \mathbf{B} = 0, \quad \text{rank } \{\mathbf{B}\} = M - 1. \quad (9.25)$$

The matrix  $\mathbf{B}$  is usually denoted a blocking matrix (BM). Define

$$\begin{bmatrix} g_1 \\ \mathbf{g} \end{bmatrix} = -\mathbf{U}^{-1} \mathbf{w} \quad (9.26)$$

such that the filter  $\mathbf{w}$  can be uniquely split as

$$\mathbf{w} = -g_1 \mathbf{h} - \mathbf{B} \mathbf{g} \quad (9.27)$$

with  $\mathbf{g}$  an  $(M - 1)$ -dimensional filter. By imposing the constraint  $\mathbf{w}^H \mathbf{h} = 1$ , we obtain  $g_1 = -1/\|\mathbf{h}\|^2$  such that

$$\mathbf{w} = \mathbf{w}_0 - \mathbf{B} \mathbf{g} \quad (9.28)$$

with  $\mathbf{w}_0$  the matched beamformer in (9.10) which depends on the ATFs. The criterion in (9.13) can now be solved by a simple unconstrained minimization of the cost function with respect to  $\mathbf{g}$  rather than  $\mathbf{w}$ .

The output of the constrained beamformer is given by

$$y = \mathbf{w}_0^H \mathbf{x} - \mathbf{g}^H \mathbf{B}^H \mathbf{x} \triangleq y_{\text{MBF}} - y_{\text{ANC}}. \quad (9.29)$$

In addition, we define the  $M - 1$  output signals of the blocking matrix

$$\mathbf{u} \triangleq \mathbf{B}^H \mathbf{x} = \mathbf{B}^H (\mathbf{h}\mathbf{s} + \mathbf{v}) = \mathbf{B}^H \mathbf{v}, \quad (9.30)$$

which are referred to as *noise reference signals*, as they only contain noise components.

The beamformer output  $y$  is the difference of two terms, both operating on the input signal  $\mathbf{x}$ . The first term,  $y_{\text{MBF}}$ , is the output of the matched beamformer. The MBF coherently sums the desired speech components while in general destructively summing the noise components. Hence, it is expected that the SNR at the MBF output is higher than the input SNR (for a comprehensive discussion of this issue, we refer to [74]). The second term,  $y_{\text{ANC}}$ , is obtained by filtering the noise reference signals  $\mathbf{u}$  with the ANC filters  $\mathbf{g}$ .

The residual noise term in  $y_{\text{MBF}}$  can be reduced by properly adjusting the filters  $\mathbf{g}$ , by solving the unconstrained minimization problem

$$\min_{\mathbf{g}} \left\{ [\mathbf{w}_0 - \mathbf{B}\mathbf{g}]^H \Phi_{\mathbf{vv}} [\mathbf{w}_0 - \mathbf{B}\mathbf{g}] \right\}. \quad (9.31)$$

The solution is given by (see also [75, 76])

$$\mathbf{g} = [\mathbf{B}^H \Phi_{\mathbf{vv}} \mathbf{B}]^{-1} \mathbf{B}^H \Phi_{\mathbf{vv}} \mathbf{w}_0. \quad (9.32)$$

In practice, this unconstrained minimization problem can be adaptively solved using, for example, the normalized least-mean squares (NLMS) algorithm [77], that is,

$$\mathbf{g}(k, \ell + 1) = \begin{cases} \mathbf{g}(k, \ell) + \frac{\lambda}{p_{\text{est}}(k, \ell)} \mathbf{u}(k, \ell) y^*(k, \ell) & \text{inactive speech periods,} \\ \mathbf{g}(k, \ell) & \text{otherwise,} \end{cases} \quad (9.33)$$

where

$$p_{\text{est}}(k, \ell) = \alpha p_{\text{est}}(k, \ell - 1) + (1 - \alpha) \|\mathbf{u}(k, \ell)\|^2 \quad (9.34)$$

represents the PSD estimate of the noise reference signals,  $\lambda$  is a step size that regulates the convergence rate, and  $\alpha$  is a smoothing parameter in the PSD estimation process. To allow for the use of the STFT, we further constrain the ANC filters  $\mathbf{g}$  to have a (time-varying) FIR structure  $g_{m,i}(n) = 0$ ,  $i \notin \{-L_{g,\text{left}}, \dots, L_{g,\text{right}}\}$ . Note that the impulse responses are typically taken to be noncausal to allow for relative delays between the MBF and the ANC branches. The various filtering (multiplications in the STFT domain) operations involved in the algorithm can be realized using the overlap-save method [78, 79].

**9.3.4.2 Suboptimal GSC** The frequency-domain GSC derived in the previous section is solely determined by the ATFs  $\mathbf{h}$ . Due to the large order of the respective RIRs, estimating these ATFs generally is a cumbersome task. Alternatively, the constraint for a distortionless response can be replaced by constraining the output

speech component to be equal to the speech component at an arbitrarily chosen microphone  $r$ . Hence, the constraint in (9.12) becomes

$$y_s = \mathbf{w}^H \mathbf{h} s \stackrel{\text{constrain}}{=} s_r = sh_r, \quad (9.35)$$

which is equivalent to  $\mathbf{w}^H \tilde{\mathbf{h}} = 1$ , where the relative transfer function (RTF)  $\tilde{\mathbf{h}}$  is defined as

$$\tilde{\mathbf{h}} \triangleq \frac{\mathbf{h}}{h_r} = \left[ \begin{array}{cccccc} \frac{h_1}{h_r} & \frac{h_2}{h_r} & \dots & 1 & \dots & \frac{h_M}{h_r} \end{array} \right]^T. \quad (9.36)$$

Note that all ATFs, and in particular  $h_r$ , may have zeros outside the unit circle, as they are not necessarily minimum-phase systems. Hence, to ensure the stability of the RTFs, we allow for noncausal systems and model the impulse responses of the RTFs  $\tilde{\mathbf{h}}$  as noncausal FIR filters. Explicitly, for  $m = 1, \dots, M, m \neq r$   $\tilde{h}_{m,i}(n) = 0$ ,  $i \notin \{-L_{\tilde{h},\text{left}}, \dots, L_{\tilde{h},\text{right}}\}$ , and for  $m = r$ ,  $\tilde{h}_m(n) = \delta(n)$ . It was shown experimentally that the RTFs are usually much shorter than the corresponding ATFs [27], justifying the FIR assumption.

By substituting the ATFs  $\mathbf{h}$  with the RTFs  $\tilde{\mathbf{h}}$ , the modified MVDR and MSNR beamformers are obtained, that is,

$$\tilde{\mathbf{w}}_{\text{MVDR}} = h_r^* \mathbf{w}_{\text{MVDR}} = \frac{\Phi_{\mathbf{xx}}^{-1} \tilde{\mathbf{h}}}{\tilde{\mathbf{h}}^H \Phi_{\mathbf{xx}}^{-1} \tilde{\mathbf{h}}} = \frac{\Phi_{\mathbf{vv}}^{-1} \tilde{\mathbf{h}}}{\tilde{\mathbf{h}}^H \Phi_{\mathbf{vv}}^{-1} \tilde{\mathbf{h}}} = \tilde{\mathbf{w}}_{\text{MSNR}}. \quad (9.37)$$

Similarly, a (suboptimal) modified frequency-domain GSC, referred to as TF-GSC [27], can be constructed in which the distortionless response property is sacrificed for the simplicity of the structure. In particular, the MBF is given by

$$\tilde{\mathbf{w}}_0 = \frac{\tilde{\mathbf{h}}}{\|\tilde{\mathbf{h}}\|^2} \quad (9.38)$$

such that, using (9.3) and (9.29), the MBF output is equal to

$$y_{\text{MBF}} = h_r s + \frac{\tilde{\mathbf{h}}^H}{\|\tilde{\mathbf{h}}\|^2} \mathbf{v}. \quad (9.39)$$

Hence, when using  $\tilde{\mathbf{w}}_0$  rather than  $\mathbf{w}_0$ , the speech component in  $y_{\text{MBF}}$  becomes  $s_r = h_r s$ , as expected. Note that all microphone signals are still added coherently.

The RTFs can also be used for constructing the blocking matrix. To show this, define the noise reference signals  $u_m$ ,  $m = 1, 2, \dots, M, m \neq r$ , as

$$u_m = x_m - \tilde{h}_m x_r = h_m s + v_m - \frac{h_m}{h_r} (h_r s + v_r) = v_m - \tilde{h}_m v_r, \quad (9.40)$$

which contain only noise components, provided that  $\tilde{\mathbf{h}}$  correctly models the RTFs. The elements of the blocking matrix can be readily extracted from  $\mathbf{u} = \mathbf{B}^H \mathbf{x}$  (assuming the first microphone is the reference microphone):

$$\mathbf{B} = \left[ \begin{array}{cccc} -\tilde{h}_2^* & -\tilde{h}_3^* & \dots & -\tilde{h}_M^* \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{array} \right]. \quad (9.41)$$

The consequences of using the RTFs rather than the ATFs in the GSC structure are more evident now. On the one hand, since the output speech component is equal to the speech component at the reference microphone, the proposed suboptimal TF-GSC [27] does not aim at speech dereverberation. On the other hand, as no speech leaks to the noise reference signals, no speech self-cancellation occurs and no annoying speech distortion artifacts are expected. This is not the case when using the delay-only model for the RIRs, as in the regular GSC [70], typically leading to severe distortion effects. It was shown in [108] that sacrificing speech dereverberation may improve the noise reduction ability of the MVDR structure.

**9.3.4.3 Relative Transfer Function Estimation** This section discusses how the RTFs  $\tilde{h}$  can be estimated in practice using the system identification technique proposed in [80] and later used in the context of microphone arrays [27]. This method relies on the assumptions that the background noise is stationary, that the desired signal  $s(n)$  is nonstationary, and that the support of the relative impulse responses relating the microphones is finite and slowly time varying.

Rearranging the terms in (9.40), we have

$$x_m = \tilde{h}_m x_r + u_m. \quad (9.42)$$

Collecting  $L_i$  (not necessarily consecutive) STFT frames  $\ell_i, \dots, \ell_i + L_i - 1$  for  $i = 1, \dots, I$ , we obtain  $I$  different estimates of the time-varying cross-PSDs between the microphone signals (for each frequency bin). Using (9.42), we have

$$\phi_{x_m x_r}^{(i)} = \tilde{h}_m \phi_{x_r x_r}^{(i)} + \phi_{u_m x_r}, \quad i = 1, \dots, I, \quad (9.43)$$

where  $\phi_{x_m x_r}^{(i)}$  is the cross-PSD between  $x_m$  and  $x_r$  and  $\phi_{x_r x_r}^{(i)}$  is the PSD of  $x_r$  during the  $i$ th segment. The cross-PSD  $\phi_{u_m x_r}$  between  $u_m$  and  $x_r$  is assumed to be independent of the segment index  $i$  due to the noise stationarity. Let  $\hat{\phi}_{x_r x_r}^{(i)}$ ,  $\hat{\phi}_{x_m x_r}^{(i)}$ , and  $\hat{\phi}_{u_m x_r}^{(i)}$  be estimates of  $\phi_{x_r x_r}^{(i)}$ ,  $\phi_{x_m x_r}^{(i)}$ , and  $\phi_{u_m x_r}$  obtained by replacing expectations with averages. Note that (9.43) still holds for the estimated values. Let  $\varepsilon_m^{(i)} = \hat{\phi}_{u_m x_r}^{(i)} - \phi_{u_m x_r}$  denote the estimation error of the cross-PSD between  $u_m$  and  $x_r$  during the  $i$ th segment. Hence,

$$\hat{\phi}_{x_m x_r}^{(i)} = \tilde{h}_m \hat{\phi}_{x_r x_r}^{(i)} + \phi_{u_m x_r} + \varepsilon_m^{(i)}, \quad i = 1, \dots, I. \quad (9.44)$$

If the noise reference signals  $u_m$  were uncorrelated with  $x_r$ , then the standard system identification estimate

$$\hat{\tilde{h}}_m = \frac{\hat{\phi}_{x_m x_r}}{\hat{\phi}_{x_r x_r}} \quad (9.45)$$

could be used to obtain an unbiased estimate of  $\tilde{h}_m$ . Unfortunately, by (9.40)  $u_m$  and  $x_r$  are in general correlated. Hence, it is proposed in [80] to obtain an unbiased estimate of  $\tilde{h}_m$  by applying a weighted least-squares procedure to the set of overdetermined equations (9.44), where a separate set of equations is used for each  $m = 1, \dots, M, m \neq r$ .

Shalvi and Weinstein [80] used a diagonal weight matrix in which the weights are proportional to  $L_i$ , the number of STFT frames used for estimating the PSDs, yielding

$$\hat{\tilde{h}}_m = \frac{\langle \hat{\phi}_{x_m x_r} \hat{\phi}_{x_r x_r} \rangle - \langle \hat{\phi}_{x_m x_r} \rangle \langle \hat{\phi}_{x_r x_r} \rangle}{\langle \hat{\phi}_{x_r x_r}^2 \rangle - \langle \hat{\phi}_{x_r x_r} \rangle^2} \quad (9.46)$$

with the averaging operation defined as

$$\langle \varphi \rangle \triangleq \frac{\sum_{i=1}^I L_i \varphi^{(i)}}{\sum_{i=1}^I L_i}. \quad (9.47)$$

Special attention should be given to choosing  $L_i$ . On the one hand, it should be longer than the correlation length of  $x_m(n)$ , which must be longer than the length of the corresponding RIR. On the other hand, it should be short enough for the filter time invariance and the noise stationarity assumptions to hold.

**9.3.4.4 Summary** In this section, we surveyed the TF-GSC [27], which extends the original GSC [70] for dealing with arbitrary ATFs. Figure 9.2 depicts a block diagram of the TF-GSC which comprises three parts: an MBF  $\tilde{\mathbf{w}}_0$ , which aligns the desired speech component at each microphone; a BM  $\mathbf{B}$ , which blocks the desired speech component resulting in the noise reference signals  $\mathbf{u}$ ; and a multichannel ANC  $\mathbf{g}$ , which eliminates the residual noise component at the MBF output. The steps involved in the computation are summarized in Algorithm 1, where the operation  $\xleftarrow{\text{FIR}}$  imposes an FIR structure on the adaptive filters to allow for application of the overlap-save procedure [79].

### 9.3.5 Multichannel Wiener Filter

Multichannel Wiener Filter techniques for noise reduction have been proposed that produce an MMSE estimate of the speech source signal [81, 82], the (reverberant) speech component in one of the microphone signals [45, 83–86], or a reference speech signal [83, 84, 87]. In this section, we will focus on MWF techniques that produce an estimate of the speech component in one of the microphone signals [45, 85]. Similarly

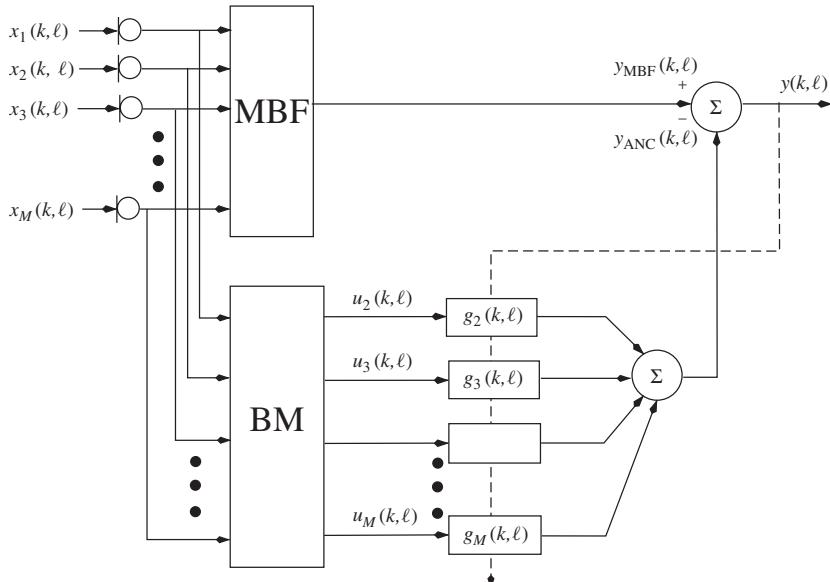


Figure 9.2 Linearly constrained adaptive beamformer.

**Algorithm 1: Summary of TF-GSC Algorithm**

1. Matched beamformer:

$$y_{MBF}(k, \ell) = \tilde{\mathbf{w}}_0^H(k, \ell) \mathbf{x}(k, \ell)$$

where  $\tilde{\mathbf{w}}_0$  is defined in (9.38) and the RTFs are estimated using (9.46).

2. Noise reference signals:

$$\mathbf{u}(k, \ell) = \mathbf{B}^H(k, \ell) \mathbf{x}(k, \ell)$$

where  $\mathbf{B}$  is constructed using (9.41) and the RTFs are estimated using (9.46).

3. Output signal:

$$y(k, \ell) = y_{MBF}(k, \ell) - \mathbf{g}^H(k, \ell) \mathbf{u}(k, \ell)$$

4. Filter update:

$$\begin{aligned} \tilde{\mathbf{g}}(k, \ell + 1) &= \mathbf{g}(k, \ell) + \frac{\lambda}{p_{est}(k, \ell)} \mathbf{u}(k, \ell) y^*(k, \ell) \\ \mathbf{g}(k, \ell + 1) &\xleftarrow{\text{FIR}} \tilde{\mathbf{g}}(k, \ell + 1) \end{aligned}$$

where  $p_{est}(k, \ell) = \alpha p_{est}(k, \ell - 1) + (1 - \alpha) \|\mathbf{u}(k, \ell)\|^2$ .

5. Keep only non-aliased samples [79].

to the TF-GSC, these MWF techniques require neither a priori information about the microphone configuration nor the position of the desired speech source, making it an appealing approach from a robustness point of view [45, 47]. These MWF techniques are uniquely based on estimates of the second-order statistics of the speech and the noise signal.

**9.3.5.1 Concept: MMSE Criterium** The  $M$ -dimensional MWF  $\mathbf{w}_{MWF}$  minimizes the mean-square error between the (unknown) speech component  $s_r = h_r s$  in the  $r$ th microphone signal and the beamformer output:

$$\mathbf{w}_{MWF} = \underset{\mathbf{w}}{\operatorname{argmin}} E\{|s_r - y|^2\} = \underset{\mathbf{w}}{\operatorname{argmin}} E\{|s_r - \mathbf{w}^H \mathbf{x}|^2\}. \quad (9.48)$$

This is illustrated in Figure 9.3. Using (9.3) and assuming the speech and the noise components to be uncorrelated, the solution of (9.48) is<sup>3</sup>

$$\mathbf{w}_{MWF} = \Phi_{\mathbf{xx}}^{-1} E\{\mathbf{x}s_r^*\} = (\phi_{ss} \mathbf{h}\mathbf{h}^H + \Phi_{\mathbf{vv}})^{-1} \phi_{ss} \mathbf{h}h_r^*. \quad (9.49)$$

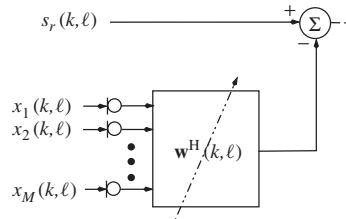


Figure 9.3 Multichannel Wiener filter.

<sup>3</sup>In (9.49), a rank-1 speech correlation matrix is assumed. However, the MWF can also be defined more generally for a full-rank speech correlation matrix [46].

**9.3.5.2 Speech-Distortion-Weighted MWF** The MMSE criterion (9.48) can be easily generalized to allow for a trade-off between noise reduction and speech distortion [34, 38]. We refer to this generalization as speech-distortion-weighted MWF (SDW-MWF). The residual energy  $E\{|s_r - \mathbf{w}^H \mathbf{h}s|^2\}$  of the MWF can be decomposed into

$$E\{|s_r - \mathbf{w}^H \mathbf{h}s|^2\} + E\{|\mathbf{w}^H \mathbf{v}|^2\}, \quad (9.50)$$

where  $E\{|s_r - \mathbf{w}^H \mathbf{h}s|^2\}$  represents the speech distortion energy and  $E\{|\mathbf{w}^H \mathbf{v}|^2\}$  represents the residual noise energy. Hence, the design criterion (9.48) of the MWF can be easily generalized by incorporating a weighting factor  $\mu$  [45]:

$$\mathbf{w}_{\text{SDW}} = \underset{\mathbf{w}}{\operatorname{argmin}} E\{|s_r - \mathbf{w}^H \mathbf{h}s|^2\} + \mu E\{|\mathbf{w}^H \mathbf{v}|^2\}. \quad (9.51)$$

The solution of (9.51) is given by

$$\mathbf{w}_{\text{SDW}} = (\phi_{ss} \mathbf{h} \mathbf{h}^H + \mu \Phi_{vv})^{-1} \phi_{ss} \mathbf{h} h_r^*. \quad (9.52)$$

The factor  $\mu \in [0, \infty]$  allows a trade-off between speech distortion and noise reduction, hence, the name SDW-MWF: The smaller  $\mu$ , the smaller the resulting speech distortion. If  $\mu = 1$ , the MMSE criterion (9.48) is obtained. If  $\mu > 1$ , the residual noise level will be reduced at the expense of increased speech distortion. By setting  $\mu$  to  $\infty$ , all emphasis is put on noise reduction and speech distortion is completely ignored, resulting in  $\mathbf{w} = 0$ . On the other hand, setting  $\mu$  to 0 results in  $\mathbf{w} = \mathbf{e}_r$ , with the  $M$ -dimensional vector  $\mathbf{e}_r$  equal to the  $r$ th canonical vector, that is,

$$\mathbf{e}_r = [ \begin{array}{cccccc} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{array}]^T, \quad (9.53)$$

and hence no noise reduction.

Using the matrix inversion lemma, (9.52) can be written as

$$\mathbf{w}_{\text{SDW}} = \frac{\phi_{ss} \Phi_{vv}^{-1} \mathbf{h}}{\mu + \phi_{ss} \mathbf{h}^H \Phi_{vv}^{-1} \mathbf{h}} h_r^*, \quad (9.54)$$

and, using (9.9), the output SNR is equal to

$$\text{SNR}_{\text{SDW}}^{\text{out}} = \phi_{ss} \mathbf{h}^H \Phi_{vv}^{-1} \mathbf{h} = \rho. \quad (9.55)$$

Note that the output SNR is independent of the trade-off parameter  $\mu$  and is equal to the SNR obtained by the MVDR beamformer.

**9.3.5.3 Implementation** In practice, the speech correlation matrix  $\phi_{ss} \mathbf{h} \mathbf{h}^H$  in (9.52) is unknown. During speech periods, the microphone signals consist of speech plus noise, that is,  $x_m = h_m s + v_m$ . During noise periods, only the noise component  $v_m$  is observed. The second-order statistics of the noise are assumed to be sufficiently stationary so that they can be estimated during periods of noise only. Assuming that the speech and noise signals are uncorrelated,  $\phi_{ss} \mathbf{h} \mathbf{h}^H$  can then be estimated as

$$\phi_{ss} \mathbf{h} \mathbf{h}^H \approx \Phi_{xx} - \Phi_{vv}, \quad (9.56)$$

where  $\Phi_{xx}$  is estimated during speech-plus-noise periods and  $\Phi_{vv}$  during periods of noise only. Similarly, as for the GSC, a robust speech detection is thus needed.

Different recursive implementations have been proposed for the SDW-MWF. In [45, 85], recursive matrix-decomposition-based implementations were presented,

which are computationally quite expensive. In [39], cheaper (time-domain and frequency-domain) stochastic gradient algorithms were proposed. These algorithms however require large circular data buffers, resulting in a large memory requirement. In [34], a frequency-domain criterion for the SDW-MWF was formulated, leading to frequency-domain stochastic gradient implementations using frequency-domain correlation matrices. The latter reduce the memory requirement and the computational complexity of the SDW-MWF.

Instead of estimating the speech correlation matrix  $\phi_{ss}\mathbf{h}\mathbf{h}^H$  online using (9.56), it has been proposed to use a predetermined estimate of  $\phi_{ss}\mathbf{h}\mathbf{h}^H$  [83, 88]. In [83], this estimate is derived from clean speech recordings measured during an initial calibration phase. Additional recordings of the speech source signal allow to produce an estimate of the nonreverberant speech source signal instead of an estimate of the reverberant speech component in one of the microphone signals. However, since the room acoustics, the position of the desired speaker, and the microphone characteristics may change over time, frequent recalibration is required. In [88], a mathematical estimate of the correlation matrix and the correlation vector of the non-reverberant speech is exploited in which some signal model errors are taken into account.

**9.3.5.4 Relationship between MWF and TF-GSC** Simmer et al. [82] addressed the general problem of single-channel postfiltering. First, they derived the multichannel Wiener filter for estimating the speech source signal  $s$  from the microphone signals  $\mathbf{x}$  given in (9.3). Then, they showed that this Wiener filter can be factorized into a multiplication of the MVDR beamformer in (9.16) and a single-channel Wiener filter that depends on the speech and the noise components of the MVDR output. Balan and Rosca [89] showed that these results can be generalized to other multichannel Bayesian cost functions, such as the short-time spectral amplitude (STSA) and the log spectral amplitude (LSA) proposed by Ephraim and Malah [90, 91].

Here, we generalize the results in [82] to the SDW-MWF estimating the speech component at the  $r$ th microphone. The SDW-MWF in (9.54) can be rewritten as

$$\mathbf{w}_{\text{SDW}} = \left( \frac{\phi_{ss}}{\phi_{ss} + \mu(\mathbf{h}^H \Phi_{vv}^{-1} \mathbf{h})^{-1}} \right) \frac{\Phi_{vv}^{-1} \mathbf{h}}{\mathbf{h}^H \Phi_{vv}^{-1} \mathbf{h}} h_r^*. \quad (9.57)$$

Using  $\mathbf{h} = h_r \tilde{\mathbf{h}}$  and  $\phi_{ss} = \phi_{s_r s_r} / |h_r|^2$ , we obtain

$$\mathbf{w}_{\text{SDW}} = \left( \frac{\phi_{s_r s_r}}{\phi_{s_r s_r} + \mu(\tilde{\mathbf{h}}^H \Phi_{vv}^{-1} \tilde{\mathbf{h}})^{-1}} \right) \frac{\Phi_{vv}^{-1} \tilde{\mathbf{h}}}{\tilde{\mathbf{h}}^H \Phi_{vv}^{-1} \tilde{\mathbf{h}}}. \quad (9.58)$$

The reader can easily identify the second term in this expression as the modified MVDR beamformer  $\tilde{\mathbf{w}}_{\text{MVDR}}$  in (9.37) which uses the RTFs rather than the ATFs. We now turn to the first term. Since by definition  $y_s = s_r$ , the speech PSD at the output of the modified MVDR beamformer is equal to

$$\phi_{y_s y_s} = \phi_{s_r s_r}. \quad (9.59)$$

The noise PSD at the output of the modified MVDR beamformer is equal to

$$\phi_{y_v y_v} = (\tilde{\mathbf{w}}_{\text{MVDR}})^H \Phi_{vv} \tilde{\mathbf{w}}_{\text{MVDR}} = \left( \tilde{\mathbf{h}}^H \Phi_{vv}^{-1} \tilde{\mathbf{h}} \right)^{-1}. \quad (9.60)$$

Using (9.59) and (9.60), the components of the first multiplicative term in the SDW-MWF can be identified as the speech distortion-weighted single-channel Wiener filter (SDW-SWF) applied to the modified MVDR beamformer output:

$$\mathbf{w}_{\text{SDW}} = \underbrace{\frac{\phi_{y_s y_s}}{\phi_{y_s y_s} + \mu \phi_{y_v y_v}}}_{\text{SDW-SWF postfilter}} \times \underbrace{\frac{\Phi_{vv}^{-1} \tilde{\mathbf{h}}}{\tilde{\mathbf{h}}^H \Phi_{vv}^{-1} \tilde{\mathbf{h}}}}_{\text{MVDR beamformer}} . \quad (9.61)$$

**9.3.5.5 Mitigating Leakage Problem of GSC** Due to room reverberation, microphone mismatch, look direction error, and spatially distributed sources, speech components may leak into the noise references of the standard GSC, giving rise to speech distortion and possibly signal cancellation. To limit the distortion effect caused by the speech leakage, the ANC of the GSC is typically adapted during periods of noise only [71, 30, 32, 92]. When used in combination with small-sized arrays, for example, in hearing aid applications, an additional robustness constraint [30, 33, 35, 37, 93] is required to guarantee performance in the presence of small errors in the assumed model, such as microphone mismatch [47]. A commonly used solution is to impose a quadratic inequality constraint (QIC) on the ANC filters  $\mathbf{g}$ :

$$\mathbf{g}^H \mathbf{g} \leq \beta^2. \quad (9.62)$$

The QIC avoids excessive growth of the ANC filters, hence reducing the undesired speech distortion when speech leaks into the noise references.

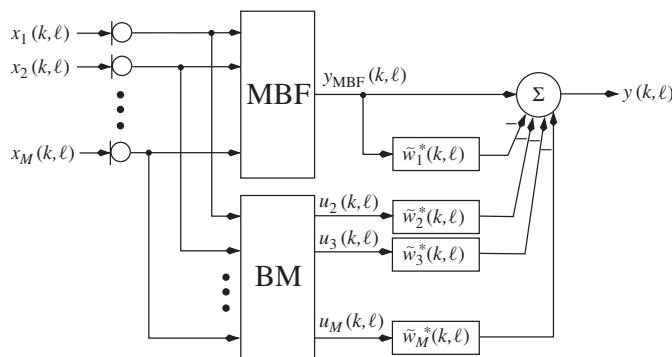
An alternative solution to limit speech distortion is to use an SDW-MWF instead of an ANC for the adaptive stage in the GSC, as depicted in Figure 9.4 [38]. Define the speech and noise components of the noise reference signals of the GSC as  $\mathbf{u}_s$  and  $\mathbf{u}_v$ , respectively, and define the speech and noise components in the speech reference signal as  $y_{s,\text{MBF}}$  and  $y_{v,\text{MBF}}$ , respectively. In contrast to the ANC that minimizes the output noise power, that is,

$$E\{|y_{v,\text{MBF}} - \mathbf{g}^H \mathbf{u}_v|^2\}, \quad (9.63)$$

the  $(M - 1)$ -dimensional SDW-MWF  $\tilde{\mathbf{w}}^H = [\tilde{w}_2^* \tilde{w}_3^* \cdots \tilde{w}_M^*]$  takes speech distortion due to speech leakage explicitly into account in the design criterion, that is,

$$\tilde{\mathbf{w}} = \underset{\tilde{\mathbf{w}}}{\operatorname{argmin}} \frac{1}{\mu} E\{\|\tilde{\mathbf{w}}^H \mathbf{u}_s\|^2\} + E\{|y_{v,\text{MBF}} - \tilde{\mathbf{w}}^H \mathbf{u}_v|^2\}, \quad (9.64)$$

where the parameter  $\mu \in [0, \infty]$  trades off noise reduction and speech distortion.



**Figure 9.4** Application of SDW-MWF to mitigate the speech leakage problem of the GSC.

Since the SDW-MWF takes speech distortion explicitly into account in its design criterion, it is possible to include an extra filter  $\tilde{w}_1$  on the speech reference (see Figure 9.4). Define the  $M$ -dimensional extended multichannel filter  $\tilde{\mathbf{w}}_e$  as

$$\tilde{\mathbf{w}}_e^H = [ \tilde{w}_1^* \quad \tilde{\mathbf{w}}^H ]. \quad (9.65)$$

The optimization criterion then becomes

$$\tilde{\mathbf{w}}_e = \underset{\tilde{\mathbf{w}}_e}{\operatorname{argmin}} \frac{1}{\mu} E \left\{ \left| \tilde{\mathbf{w}}_e^H \begin{bmatrix} y_{s,\text{MBF}} \\ \mathbf{u}_s \end{bmatrix} \right|^2 \right\} + E \left\{ \left| y_{v,\text{MBF}} - \tilde{\mathbf{w}}_e^H \begin{bmatrix} y_{v,\text{MBF}} \\ \mathbf{u}_v \end{bmatrix} \right|^2 \right\}.$$

Depending on the setting of the trade-off parameter  $\mu$  and the presence/absence of the filter  $\tilde{w}_1$  on the speech reference, different algorithms are obtained [38]:

- Without a filter  $\tilde{w}_1$ , we obtain the *speech distortion regularized GSC* (SDR-GSC), where the standard optimization criterion of the ANC is supplemented with the regularization term

$$\frac{1}{\mu} E \{ |\tilde{\mathbf{w}}^H \mathbf{u}_s|^2 \}. \quad (9.66)$$

For  $\mu = \infty$ , speech distortion is completely ignored, which corresponds to the standard GSC. Compared to the QIC, the SDR-GSC is less conservative, since the regularization term is proportional to the actual amount of speech leakage in the noise references. In [38, 39], it has been shown that in comparison with the QIC the SDR-GSC achieves a better noise reduction performance for small model errors while guaranteeing robustness against large model errors.

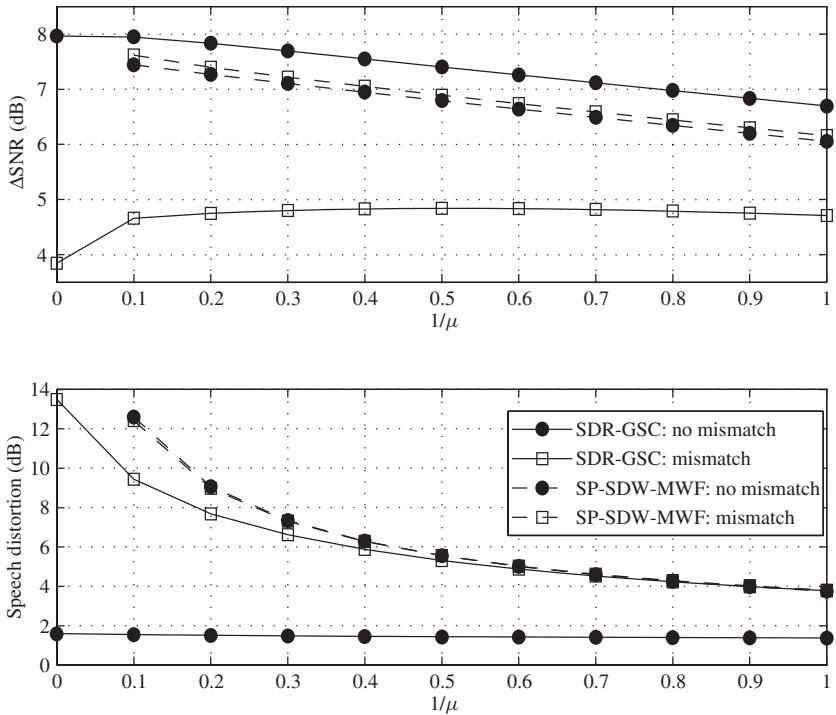
- With a filter  $\tilde{w}_1$ , we obtain the *spatially preprocessed speech-distortion-weighted multichannel Wiener filter* (SP-SDW-MWF). In [38], it has been shown that for infinitely long filter lengths the SP-SDW-MWF is not affected by microphone mismatch.

**9.3.5.6 Illustration for Hearing Aid Application** Figure 9.5 illustrates the noise reduction and speech distortion of the frequency-domain implementation of the SP-SDW-MWF for a hearing aid application [34]. In the implementation, a (non-perfect) energy-based voice activity detector (VAD) has been used [94]. A three-microphone behind-the-ear hearing aid was mounted on the right ear of a dummy head in an office room with a reverberation time  $T_{60\text{dB}} \approx 700$  ms. The desired speech source and noise sources are positioned at a distance of 1 m from the head: the speech source in front of the head, the noise sources at an angle  $\theta$  with respect to the speech source. The speech signal consists of sentences from the HINT-database [95] spoken by a male speaker. The noise scenario consists of five spectrally nonstationary multitalker babble noise sources at  $75^\circ$ ,  $120^\circ$ ,  $180^\circ$ ,  $240^\circ$ , and  $285^\circ$ . The input SNR of the microphone signals is 0 dB. For evaluation purposes, the speech and noise signals were recorded separately.

To assess the performance, the (broadband) *intelligibility weighted SNR improvement* [96] between the output signal and the speech reference signal is computed,

$$\Delta \text{SNR} = \sum_{k=0}^{K-1} F(k) [10 \log_{10} \text{SNR}^{\text{out}}(k) - 10 \log_{10} \text{SNR}^{\text{MBF}}(k)], \quad (9.67)$$

where  $F(k)$  expresses the importance of the  $k$ th frequency bin for speech intelligibility [97]. The intelligibility weighted speech distortion is defined similarly.



**Figure 9.5** Increased robustness of the SP-SDW-MWF for a hearing aid application.

Figure 9.5 depicts the noise reduction and the speech distortion for the SDR-GSC and the SP-SDW-MWF as a function of  $1/\mu$ . This figure also demonstrates the effect of a gain mismatch of 4 dB applied to the second microphone.

- **SDR-GSC:** *In the absence of microphone mismatch*, the amount of speech leakage into the noise references is limited. Hence, the amount of speech distortion of the GSC (i.e., the SDR-GSC with  $1/\mu = 0$ ) is low. Since there is still a small amount of speech leakage due to reverberation, the amount of noise reduction and speech distortion slightly decreases for increasing  $1/\mu$ . *In the presence of microphone mismatch*, the amount of speech leakage into the noise references grows. For  $1/\mu = 0$  (GSC), the speech gets significantly distorted. Setting  $1/\mu > 0$  improves the performance of the GSC in the presence of microphone mismatch, that is, the speech distortion decreases and the degradation in noise reduction becomes smaller.
- **SP-SDW-MWF:** The noise reduction and speech distortion also decrease for increasing  $1/\mu$ . Compared to the SDR-GSC, the distortion is larger due to spectral filtering, but both the noise reduction and speech distortion are hardly affected by microphone mismatch.

## 9.4 BINAURAL BEAMFORMING

The main difference between monaural and binaural beamforming is the fact that we now consider two sets of microphones, that is, one at the left hearing aid and one

at the right hearing aid, and the fact that the binaural beamformer needs to produce two (distinct) output signals (cf. Fig. 9.6). In this section we will consider a binaural system where both hearing aids are cooperating with each other, for example, through a wireless link. Although currently available binaural hearing aids with a wireless link have very limited data rates and are only able to transmit data in order to coordinate parameter settings, we anticipate that future binaural hearing aids will also allow for transmission of (coded) audio signals [98] such that we may assume that all microphone signals are simultaneously available at the left and right hearing aids. As already mentioned in Section 9.1, the objective of a binaural signal enhancement algorithm is not only to selectively extract the useful signal and to suppress the background noise but also to preserve the binaural auditory cues (interaural time and level difference) of the speech and noise sources since these cues play a major role in the preservation of spatial awareness and in an improved speech understanding in noisy environments [2–7].

Section 9.4.1 discusses the binaural microphone configuration and notations. Section 9.4.2 discusses the binaural MWF, where it is shown that the binaural cues of the speech source are preserved but the binaural cues of the noise component are distorted. Sections 9.4.3 and 9.4.4 then discuss two extensions of the binaural MWF that aim to also preserve the binaural cues of the noise component. In Section 9.4.3 a parameter allows to trade off noise reduction performance and preservation of the binaural noise cues, whereas in Section 9.4.4 the binaural MWF cost function is extended with terms related to the interaural transfer function (ITF). Section 9.4.5 presents experimental results for the discussed binaural noise reduction algorithms.

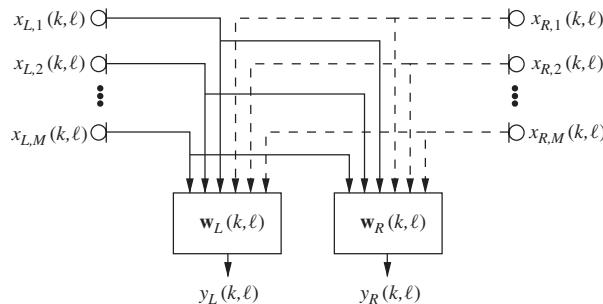
#### 9.4.1 Configuration and Notation

**9.4.1.1 Microphone Signals and Output Signals** Similar to the monaural configuration in Figure 9.1, the binaural hearing aid configuration in Figure 9.6 now consists of two microphone arrays each having  $M$  microphones. The  $m$ th microphone signal on the left hearing aid,  $x_{L,m}$ , can be written as

$$x_{L,m} = h_{L,m}s + v_{L,m}, \quad m = 1 \dots M,$$

with  $h_{L,m}$  the  $m$ th element of the ATF for the left hearing aid. Similarly, the  $m$ th microphone signal on the right hearing aid is equal to  $x_{R,m} = h_{R,m}s + v_{R,m}$ . Hence, the signal model in (9.3) remains valid, that is,

$$\mathbf{x} = \mathbf{hs} + \mathbf{v}, \quad (9.68)$$



**Figure 9.6** General binaural processing scheme.

where  $\mathbf{x}$ ,  $\mathbf{v}$ , and  $\mathbf{h}$  are now  $2M$ -dimensional vectors defined as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_L \\ \mathbf{x}_R \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}_L \\ \mathbf{v}_R \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} \mathbf{h}_L \\ \mathbf{h}_R \end{bmatrix}, \quad (9.69)$$

with

$$\begin{aligned} \mathbf{x}_L &= [x_{L,1} \ x_{L,2} \ \cdots \ x_{L,M}]^T, & \mathbf{h}_L &= [h_{L,1} \ h_{L,2} \ \cdots \ h_{L,M}]^T, \\ \mathbf{x}_R &= [x_{R,1} \ x_{R,2} \ \cdots \ x_{R,M}]^T, & \mathbf{h}_R &= [h_{R,1} \ h_{R,2} \ \cdots \ h_{R,M}]^T. \end{aligned}$$

We will use the  $r_L$ -th microphone on the left hearing aid and the  $r_R$ -th microphone on the right hearing aid as the reference microphones for the speech enhancement algorithms. Typically, the front microphones are used as reference microphones. For conciseness, the reference microphone signals  $x_{L,r_L}$  and  $x_{R,r_R}$  on the left and right hearing aids are denoted as  $x_L$  and  $x_R$ :

$$x_L = \mathbf{e}_L^H \mathbf{x} = h_{LS} + v_L, \quad x_R = \mathbf{e}_R^H \mathbf{x} = h_{RS} + v_R, \quad (9.70)$$

where  $\mathbf{e}_L$  and  $\mathbf{e}_R$  are  $2M$ -dimensional canonical vectors [cf. (9.53)] with  $\mathbf{e}_L(r_L) = 1$  and  $\mathbf{e}_R(M+r_R) = 1$ .

The output signals  $y_L$  and  $y_R$  on the left and right hearing aids are obtained by filtering and summing all microphone signals from both hearing aids, that is,

$$y_L = \mathbf{w}_L^H \mathbf{x}, \quad y_R = \mathbf{w}_R^H \mathbf{x}, \quad (9.71)$$

where  $\mathbf{w}_L$  and  $\mathbf{w}_R$  represent the  $2M$ -dimensional beamformer filter coefficients for the left and right hearing aids. Similar to (9.8), the output signals can be decomposed as

$$y_L = \mathbf{w}_L^H \mathbf{h}s + \mathbf{w}_L^H \mathbf{v} \triangleq y_{sL} + y_{vL}, \quad (9.72)$$

$$y_R = \mathbf{w}_R^H \mathbf{h}s + \mathbf{w}_R^H \mathbf{v} \triangleq y_{sR} + y_{vR}. \quad (9.73)$$

The  $4M$ -dimensional stacked weight vector  $\mathbf{w}$  is defined as

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_L \\ \mathbf{w}_R \end{bmatrix}. \quad (9.74)$$

**9.4.1.2 Performance Measures** Similar to (9.9), the SNR of the beamformer output on the left and right hearing aids is defined as

$$\text{SNR}_L^{\text{out}} = \frac{E\{|y_{sL}|^2\}}{E\{|y_{vL}|^2\}} = \frac{|\mathbf{w}_L^H \mathbf{h}|^2 \phi_{ss}}{\mathbf{w}_L^H \Phi_{vv} \mathbf{w}_L}, \quad \text{SNR}_R^{\text{out}} = \frac{E\{|y_{sR}|^2\}}{E\{|y_{vR}|^2\}} = \frac{|\mathbf{w}_R^H \mathbf{h}|^2 \phi_{ss}}{\mathbf{w}_R^H \Phi_{vv} \mathbf{w}_R}. \quad (9.75)$$

Since for binaural beamformers we are concerned not only with noise reduction but also with binaural cue preservation, we define the ILD and the ITD of the speech and noise components both on the (input) reference microphones and on the beamformer output. The ILD for the speech and noise components is expressed as the power ratio between the left and right hearing aids:

$$P_v^{\text{in}} = \frac{E\{|v_L|^2\}}{E\{|v_R|^2\}} = \frac{\mathbf{e}_L^H \Phi_{vv} \mathbf{e}_L}{\mathbf{e}_R^H \Phi_{vv} \mathbf{e}_R}, \quad P_v^{\text{out}} = \frac{E\{|y_{vL}|^2\}}{E\{|y_{vR}|^2\}} = \frac{\mathbf{w}_L^H \Phi_{vv} \mathbf{w}_L}{\mathbf{w}_R^H \Phi_{vv} \mathbf{w}_R}, \quad (9.76)$$

$$P_s^{\text{in}} = \frac{|h_L|^2}{|h_R|^2}, \quad P_s^{\text{out}} = \frac{E\{|y_{sL}|^2\}}{E\{|y_{sR}|^2\}} = \frac{|\mathbf{w}_L^H \mathbf{h}|^2}{|\mathbf{w}_R^H \mathbf{h}|^2}. \quad (9.77)$$

The ITD is expressed as the (phase of) the cross-correlation:

$$c_v^{\text{in}} = E\{v_L v_R^*\} = \mathbf{e}_L^H \Phi_{vv} \mathbf{e}_R, \quad c_v^{\text{out}} = E\{y_{vL} y_{vR}^*\} = \mathbf{w}_L^H \Phi_{vv} \mathbf{w}_R, \quad (9.78)$$

$$c_s^{\text{in}} = \phi_{ss} h_L h_R^*, \quad c_s^{\text{out}} = E\{y_{sL} y_{sR}^*\} = \phi_{ss} \mathbf{w}_L^H \mathbf{h} \mathbf{h}^H \mathbf{w}_R. \quad (9.79)$$

Combining the ITD and ILD,<sup>4</sup> the *interaural transfer function (ITF)* is defined as the ratio of the components at the left and right hearing aids:

$$\text{ITF}_v^{\text{in}} = \frac{v_L}{v_R}, \quad \text{ITF}_v^{\text{out}} = \frac{y_{vL}}{y_{vR}} = \frac{\mathbf{w}_L^H \mathbf{v}}{\mathbf{w}_R^H \mathbf{v}}, \quad (9.80)$$

$$\text{ITF}_s^{\text{in}} = \frac{h_L}{h_R}, \quad \text{ITF}_s^{\text{out}} = \frac{y_{sL}}{y_{sR}} = \frac{\mathbf{w}_L^H \mathbf{h}}{\mathbf{w}_R^H \mathbf{h}}. \quad (9.81)$$

#### 9.4.2 Binaural Multichannel Wiener Filter

**9.4.2.1 Cost Function** Similar to the monaural MWF discussed in Section 9.3.5, the binaural MWF produces an MMSE estimate of the speech component on both hearing aids [49]. The MSE cost function for the filter  $\mathbf{w}_L$  estimating the speech component  $s_L = h_{LS}$  in the left reference microphone signal and the filter  $\mathbf{w}_R$  estimating the speech component  $s_R = h_{RS}$  in the right reference microphone signal is equal to

$$J_{\text{MSE}}(\mathbf{w}) = E \left\{ \left\| \begin{bmatrix} s_L - y_L \\ s_R - y_R \end{bmatrix} \right\|^2 \right\} = E \left\{ \left\| \begin{bmatrix} s_L - \mathbf{w}_L^H \mathbf{x} \\ s_R - \mathbf{w}_R^H \mathbf{x} \end{bmatrix} \right\|^2 \right\}. \quad (9.82)$$

The cost function for the binaural SDW-MWF, allowing a trade-off between speech distortion and noise reduction, is equal to

$$J_{\text{SDW}}(\mathbf{w}) = E \left\{ \left\| \begin{bmatrix} s_L - \mathbf{w}_L^H \mathbf{h}s \\ s_R - \mathbf{w}_R^H \mathbf{h}s \end{bmatrix} \right\|^2 + \mu \left\| \begin{bmatrix} \mathbf{w}_L^H \mathbf{v} \\ \mathbf{w}_R^H \mathbf{v} \end{bmatrix} \right\|^2 \right\}, \quad (9.83)$$

where  $\mu$  trades off noise reduction and speech distortion. Similar to (9.52) and (9.54) for the monaural case, the binaural SDW-MWF solution is equal to [48, 99]

$$\mathbf{w}_{\text{SDW},L} = (\phi_{ss} \mathbf{h} \mathbf{h}^H + \mu \Phi_{vv})^{-1} \phi_{ss} \mathbf{h} h_L^* = \frac{\phi_{ss} \Phi_{vv}^{-1} \mathbf{h}}{\mu + \rho} h_L^*, \quad (9.84)$$

$$\mathbf{w}_{\text{SDW},R} = (\phi_{ss} \mathbf{h} \mathbf{h}^H + \mu \Phi_{vv})^{-1} \phi_{ss} \mathbf{h} h_R^* = \frac{\phi_{ss} \Phi_{vv}^{-1} \mathbf{h}}{\mu + \rho} h_R^*, \quad (9.85)$$

<sup>4</sup>It can be easily verified by the reader that the phase of the ITF is equal to the ITD and the squared norm of the ITF is equal to the ILD.

with  $\rho = \phi_{ss}\mathbf{h}^H\Phi_{vv}^{-1}\mathbf{h}$ . Hence, it can be shown that the output SNR at both hearing aids is identical and equal to  $\rho$ .

**9.4.2.2 Cue Preservation** From (9.84) and (9.85), it can be seen that the binaural MWF vectors for the left and right hearing aids are parallel:

$$\mathbf{w}_{SDW,L} = ITF_s^{in,*} \mathbf{w}_{SDW,R} \quad (9.86)$$

with  $ITF_s^{in}$  defined in (9.81). Hence, the ITFs of the output speech and noise components are both equal to  $ITF_s^{in}$ , implying that the binaural speech cues are perfectly preserved, but the binaural noise cues are distorted. As all output components are perceived as coming from the speech direction, the auditory perception of the acoustic scene is therefore typically not preserved by the binaural MWF.

### 9.4.3 Partial Noise Estimation (SDW-MWF- $\eta$ )

**9.4.3.1 Cost Function** An extension of the binaural MWF in (9.82) that aims to preserve the binaural noise cues (in addition to preserving the binaural speech cues) has been proposed in [49], where the objective is to produce an MMSE estimate of the sum of the speech component and a scaled version of the noise component in the reference microphone signals, that is,

$$J_{MSE\eta}(\mathbf{w}) = E \left\{ \left\| \begin{bmatrix} (s_L + \eta v_L) - \mathbf{w}_L^H \mathbf{x} \\ (s_R + \eta v_R) - \mathbf{w}_R^H \mathbf{x} \end{bmatrix} \right\|^2 \right\} \quad (9.87)$$

with the scaling parameter  $0 \leq \eta \leq 1$ . When  $\eta = 0$ , this cost function reduces to the standard binaural MWF cost function in (9.82). When  $\eta = 1$ , the optimal filters are obviously equal to  $\mathbf{w}_{MSE\eta,L} = \mathbf{e}_L$  and  $\mathbf{w}_{MSE\eta,R} = \mathbf{e}_R$ , resulting in perfect preservation of the binaural speech and noise cues but no noise reduction at all. This partial noise estimation procedure is in fact an extension of single-channel Wiener filters that have been presented in [100, 101].

Similar to the SDW-MWF cost function in (9.83), it is also possible to combine partial noise estimation while trading off noise reduction and speech distortion:

$$J_{SDW\eta}(\mathbf{w}) = E \left\{ \left\| \begin{bmatrix} s_L - \mathbf{w}_L^H \mathbf{h}_S \\ s_R - \mathbf{w}_R^H \mathbf{h}_S \end{bmatrix} \right\|^2 + \mu \left\| \begin{bmatrix} \eta v_L - \mathbf{w}_L^H \mathbf{v} \\ \eta v_R - \mathbf{w}_R^H \mathbf{v} \end{bmatrix} \right\|^2 \right\}. \quad (9.88)$$

The binaural filters minimizing  $J_{SDW\eta}(\mathbf{w})$  are equal to

$$\begin{aligned} \mathbf{w}_{SDW\eta,L} &= (\phi_{ss}\mathbf{h}\mathbf{h}^H + \mu\Phi_{vv})^{-1}(\phi_{ss}\mathbf{h}\mathbf{h}^H + \eta\mu\Phi_{vv})\mathbf{e}_L \\ &= (1 - \eta)\mathbf{w}_{SDW,L} + \eta\mathbf{e}_L = (1 - \eta)\frac{\phi_{ss}\Phi_{vv}^{-1}\mathbf{h}}{\mu + \rho}h_L^* + \eta\mathbf{e}_L, \end{aligned} \quad (9.89)$$

$$\begin{aligned} \mathbf{w}_{SDW\eta,R} &= (\phi_{ss}\mathbf{h}\mathbf{h}^H + \mu\Phi_{vv})^{-1}(\phi_{ss}\mathbf{h}\mathbf{h}^H + \eta\mu\Phi_{vv})\mathbf{e}_R \\ &= (1 - \eta)\mathbf{w}_{SDW,R} + \eta\mathbf{e}_R = (1 - \eta)\frac{\phi_{ss}\Phi_{vv}^{-1}\mathbf{h}}{\mu + \rho}h_R^* + \eta\mathbf{e}_R. \end{aligned} \quad (9.90)$$

Hence, the SDW-MWF with partial noise estimation corresponds to mixing the output signal of the standard SDW-MWF with the reference microphone signals.

It has been shown in [99] that the SNR improvement at the right hearing aid is equal to

$$\Delta \text{SNR}_R = \Delta \text{SNR}_R^o \frac{[(\eta\mu + \rho)/(\mu + \rho)]^2}{[(\eta\mu + \rho)/(\mu + \rho)]^2 + (\Delta \text{SNR}_R^o - 1)\eta^2}, \quad (9.91)$$

where  $\Delta \text{SNR}_R^o$  represents the SNR improvement at the right hearing aid for the standard binaural SDW-MWF in (9.85). Obviously, if  $\eta = 0$ , the SNR improvement is equal to  $\Delta \text{SNR}_R^o$ , whereas if  $\eta = 1$ , no SNR improvement is obtained, that is,  $\Delta \text{SNR}_R = 1$ . Since the SNR improvement  $\Delta \text{SNR}_R^o$  of the standard SDW-MWF is always larger than or equal to 1 [102], it can be easily shown that

$$1 \leq \Delta \text{SNR}_R \leq \Delta \text{SNR}_R^o, \quad (9.92)$$

that is, the SNR improvement is always smaller when using partial noise estimation. Similar expressions can be derived for the left hearing aid, where it should also be noted that the output SNR at both hearing aids is now not necessarily identical.

**9.4.3.2 Cue Preservation** From (9.89) and (9.90), it can be seen that the binaural MWF filters when using partial noise estimation are in general not parallel such that the ITF of the output speech and noise components are typically different. Using (9.81), the ITF of the output speech component is equal to

$$\text{ITF}_s^{\text{out}} = \frac{\mathbf{w}_{\text{SDW}\eta,L}^H \mathbf{h}}{\mathbf{w}_{\text{SDW}\eta,R}^H \mathbf{h}} = \frac{(1-\eta)[\rho/(\mu + \rho)]h_L + \eta h_L}{(1-\eta)[\rho/(\mu + \rho)]h_R + \eta h_R} = \frac{h_L}{h_R} = \text{ITF}_s^{\text{in}} \quad (9.93)$$

such that the binaural speech cues are preserved. Using (9.76) and (9.77), it can be shown that the ILD of the output noise component is a weighted sum of the ILDs of the input speech and noise component [99], that is,

$$P_v^{\text{out}} = P_s^{\text{in}} + \frac{\eta^2}{\psi \text{SNR}_R^{\text{in}} + \eta^2} (P_v^{\text{in}} - P_s^{\text{in}}) \quad (9.94)$$

with

$$\psi = \frac{(1-\eta)(2\eta\mu + \rho + \eta\rho)}{(\mu + \rho)^2}, \quad (9.95)$$

and  $\text{SNR}_R^{\text{in}}$  representing the input SNR on the right hearing aid. If  $\eta = 0$ , the ILD of the output noise component is equal to  $P_s^{\text{in}}$ , whereas if  $\eta = 1$ ,  $\psi = 0$  such that the ILD of the output noise component is equal to  $P_v^{\text{in}}$ . Similarly, using (9.78) and (9.79), it can be shown that the cross-correlation (related to the ITD) of the output noise component is a weighted sum of the cross-correlations of the input speech and noise component [99]:

$$c_v^{\text{out}} = \psi c_s^{\text{in}} + \eta^2 c_v^{\text{in}}. \quad (9.96)$$

Hence, the parameter  $\eta$  trades off noise reduction and preservation of the binaural noise cues, as will also be shown in the experimental results in Section 9.4.5.

#### 9.4.4 ITF Cost Function (SDW-MWF-ITF)

**9.4.4.1 Cost Function** In order to control the binaural cues of the speech and the noise component, it is also possible to extend the quadratic SDW-MWF cost function in (9.83) with terms related to the ITF of the speech and noise components, as has been proposed in [48, 51, 103]. Since the aim is to preserve the binaural speech and noise cues, the desired output ITFs  $\text{ITF}_v^{\text{des}}$  and  $\text{ITF}_s^{\text{des}}$  are equal to the input ITFs defined in (9.80) and (9.81), where for the noise component the input ITF is assumed to be constant (as is, e.g., the case for a single noise source) such that it can be estimated in the least-squares sense using the noise correlation matrix, that is,

$$\text{ITF}_v^{\text{des}} = \frac{E\{v_L v_R^*\}}{E\{v_R v_R^*\}} = \frac{\mathbf{e}_L^H \Phi_{vv} \mathbf{e}_R}{\mathbf{e}_R^H \Phi_{vv} \mathbf{e}_R}, \quad \text{ITF}_s^{\text{des}} = \frac{h_L}{h_R}. \quad (9.97)$$

The (nonlinear) *ITF cost function* for preserving the binaural cues of the noise component is then defined as [48]

$$J_{\text{ITF},1}^v(\mathbf{w}) = E\left\{\left|\frac{\mathbf{w}_L^H \mathbf{v}}{\mathbf{w}_R^H \mathbf{v}} - \text{ITF}_v^{\text{des}}\right|^2\right\}, \quad (9.98)$$

which, in the case of a single noise source, is equal to

$$J_{\text{ITF},1}^v(\mathbf{w}) = \frac{E\{|\mathbf{w}_L^H \mathbf{v} - \text{ITF}_v^{\text{des}} \mathbf{w}_R^H \mathbf{v}|^2\}}{E\{|\mathbf{w}_R^H \mathbf{v}|^2\}} = \frac{\mathbf{w}^H \Phi_{vv}^t \mathbf{w}}{\mathbf{w}^H \Phi_{vv}^1 \mathbf{w}} \quad (9.99)$$

with

$$\Phi_{vv}^t = \begin{bmatrix} \Phi_{vv} & -\text{ITF}_v^{\text{des},*} \Phi_{vv} \\ -\text{ITF}_v^{\text{des}} \Phi_{vv} & |\text{ITF}_v^{\text{des}}|^2 \Phi_{vv} \end{bmatrix}, \quad \Phi_{vv}^1 = \begin{bmatrix} \mathbf{0}_{2M} & \mathbf{0}_{2M} \\ \mathbf{0}_{2M} & \Phi_{vv} \end{bmatrix}.$$

We also introduce a simplified *quadratic ITF cost function* by assuming the denominator in (9.99) to be constant, that is [51, 103],

$$J_{\text{ITF},2}^v(\mathbf{w}) = E\{|\mathbf{w}_L^H \mathbf{v} - \text{ITF}_v^{\text{des}} \mathbf{w}_R^H \mathbf{v}|^2\} = \mathbf{w}^H \Phi_{vv}^t \mathbf{w}. \quad (9.100)$$

The ITF cost function for the speech component is defined similarly as the ITF cost function for the noise component, by replacing the noise correlation matrix  $\Phi_{vv}$  with the speech correlation matrix  $\phi_{ss} \mathbf{h} \mathbf{h}^H$  and the desired noise ITF with the desired speech ITF. The *total cost function* trading off noise reduction, speech distortion, and binaural cue preservation is then defined as

$$J_{\text{SDW-ITF}}(\mathbf{w}) = J_{\text{SDW}}(\mathbf{w}) + \gamma J_{\text{ITF}}^s(\mathbf{w}) + \delta J_{\text{ITF}}^v(\mathbf{w}), \quad (9.101)$$

where the parameters  $\gamma$  and  $\delta$  enable us to put more emphasis on binaural cue preservation for the speech and the noise component. When using the nonlinear ITF cost function in (9.99), no closed-form expression is available for the filter minimizing  $J_{\text{SDW-ITF}}(\mathbf{w})$  such that we have to use iterative optimization techniques [48]. On the

other hand, when using the quadratic ITF cost function in (9.101), the filter minimising  $J_{\text{SDW-ITF}}(\mathbf{w})$  is equal to

$$\mathbf{w}_{\text{SDW-ITF}} = (\Phi + \gamma \delta \Phi_{\text{ss}}^t + \delta \Phi_{\text{vv}}^t)^{-1} \mathbf{r}_x \quad (9.102)$$

with

$$\Phi = \begin{bmatrix} \phi_{\text{ss}} \mathbf{h} \mathbf{h}^H + \mu \Phi_{\text{vv}} & \mathbf{0}_{2M} \\ \mathbf{0}_{2M} & \phi_{\text{ss}} \mathbf{h} \mathbf{h}^H + \mu \Phi_{\text{vv}} \end{bmatrix}, \quad \mathbf{r}_x = \begin{bmatrix} \phi_{\text{ss}} \mathbf{h} h_L^* \\ \phi_{\text{ss}} \mathbf{h} h_R^* \end{bmatrix}. \quad (9.103)$$

For  $\gamma = 0$ , it can be shown that the filter  $\mathbf{w}_{\text{SDW-ITF}}$  is equal to<sup>5</sup> [99, 104]

$$\mathbf{w}_{\text{SDW-ITF},L} = \frac{\phi_{\text{ss}}}{\mu + \rho} [h_L^* - \xi(h_L^* - \text{ITF}_v^{\text{des},*} h_R^*)] \Phi_{\text{vv}}^{-1} \mathbf{h}, \quad (9.104)$$

$$\mathbf{w}_{\text{SDW-ITF},R} = \frac{\phi_{\text{ss}}}{\mu + \rho} [h_R^* + \xi \text{ITF}_v^{\text{des}} (h_L^* - \text{ITF}_v^{\text{des},*} h_R^*)] \Phi_{\text{vv}}^{-1} \mathbf{h}, \quad (9.105)$$

with

$$\xi = \frac{\delta}{\mu + \rho + \delta(1 + |\text{ITF}_v^{\text{des}}|^2)}. \quad (9.106)$$

Hence, the filter  $\mathbf{w}_{\text{SDW-ITF}}$  is equal to  $\mathbf{w}_{\text{SDW}}$  in (9.84) and (9.85) plus an extra term due to the ITF extension. Remarkably, it can be shown that the output SNR at both hearing aids is again identical and equal to  $\rho$ .

**9.4.4.2 Cue Preservation** From (9.104) and (9.105), it follows that the filters  $\mathbf{w}_{\text{SDW-ITF},L}$  and  $\mathbf{w}_{\text{SDW-ITF},R}$  are parallel such that the ITFs of the output speech and noise components are identical and equal to

$$\text{ITF}^{\text{out}} = \frac{\text{ITF}_s^{\text{in}} - \xi(\text{ITF}_s^{\text{in}} - \text{ITF}_v^{\text{des}})}{1 + \xi \text{ITF}_v^{\text{des},*} (\text{ITF}_s^{\text{in}} - \text{ITF}_v^{\text{des}})}. \quad (9.107)$$

This however implies that typically the binaural cues of neither the speech nor the noise component are perfectly preserved. If the weight  $\delta = 0$ , then  $\xi = 0$  such that the output ITF is equal to the input ITF of the speech component. On the other hand, if  $\delta \rightarrow \infty$ , then  $\xi = 1/(1 + |\text{ITF}_v^{\text{des}}|^2)$  such that the output ITF is equal to  $\text{ITF}_v^{\text{des}}$ , that is, the desired ITF for the noise component. It is important to note that the output ITF in (9.107) is also linked to the output SNR  $\rho$  through the factor  $\xi$ . Hence, for frequencies with a large output SNR, that is, large  $\rho$  and small  $\xi$ , the output ITF is close to the ITF of the speech component, whereas for frequencies with a small output SNR the output ITF is close to the ITF of the noise component [99, 104]. This advantageous perceptual effect has been observed in [51].

## 9.4.5 Experimental Results

**9.4.5.1 Setup and Performance Measures** Two hearing aids with  $M = 2$  omnidirectional microphones each have been mounted on a dummy head in a room with a reverberation time  $T_{60\text{dB}} \approx 140$  ms. The distance between the microphones on each hearing aid is approximately 1 cm. The desired speech source and the noise sources

<sup>5</sup>Similar expressions can be derived when  $\gamma \neq 0$  [99].

are positioned at a distance of 1 m from the head, and different configurations have been considered:

- Speech source in front of the head ( $\theta_s = 0^\circ$ ) and several noise configurations  $\theta_v$ : a single noise source at  $60^\circ$ ,  $90^\circ$ ,  $120^\circ$ ,  $180^\circ$ ,  $270^\circ$ , or  $300^\circ$ ; two noise sources at  $[-60^\circ 60^\circ]$ ,  $[-120^\circ 120^\circ]$ , or  $[120^\circ 210^\circ]$ ; and four noise sources at  $[60^\circ 120^\circ 180^\circ 210^\circ]$  (condition N4a) or  $[60^\circ 120^\circ 180^\circ 270^\circ]$  (condition N4b)
- Speech source at the right side ( $\theta_s = 90^\circ$ ) and a noise source at  $\theta_v = 180^\circ$

The speech signal consists of sentences from the HINT database [95], and the noise source is multitalker babble noise. For all configurations, the input broadband SNR is 0 dB in the front microphone signal on the left hearing aid. For evaluation purposes, the speech and noise signals were recorded separately.

The binaural noise reduction techniques presented in Sections 9.4.2–9.4.4 are applied in each frequency bin. Using a perfect voice activity detector (VAD), the noise correlation matrix  $\Phi_{vv}$  in each frequency bin is computed during noise-only periods, the correlation matrix  $\Phi_{xx}$  in each frequency bin is computed during speech-and-noise periods, and similarly as for the monaural case in (9.56), the speech correlation matrix is estimated as  $\phi_{ss}\mathbf{h}\mathbf{h}^H \approx \Phi_{xx} - \Phi_{vv}$ . The FFT size used for STFT processing is equal to  $K = 128$  and the trade-off parameter  $\mu = 5$ .

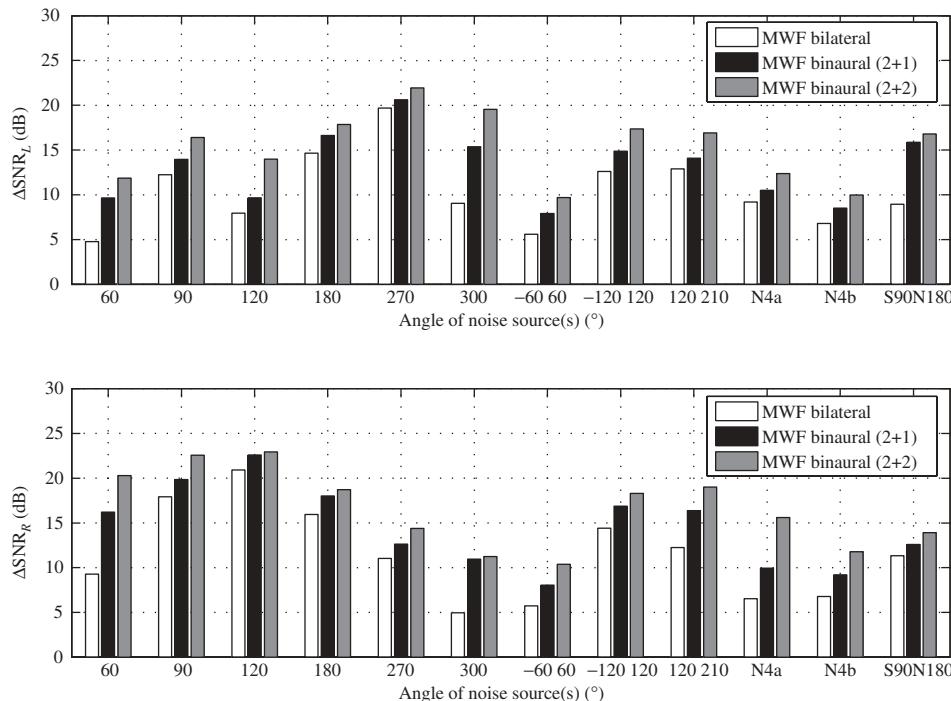
To assess the performance of the different algorithms, the broadband intelligibility weighted SNR improvement between the output signal and the reference microphone signal is computed for the left and right hearing aids as in (9.67). Since it is known that the ITD cue only plays a role in sound localization at low frequencies [4, 6, 7], the *broadband ITD error* is computed as the sum of the ITD errors for frequencies below 1500 Hz, that is, for the noise component

$$\Delta\text{ITD}_v^b = \sum_{k=0}^{K_{\text{ITD}}} |\angle c_v^{\text{out}}(k) - \angle c_v^{\text{in}}(k)|. \quad (9.108)$$

On the other hand, since ILD errors at each frequency are able to contribute to incorrect localization, the *broadband ILD error* is computed as the sum of the ILD errors at each frequency (without weighting), that is, for the noise component

$$\Delta\text{ILD}_v^b = \sum_{k=0}^{K-1} [10 \log_{10} P_v^{\text{out}}(k) - 10 \log_{10} P_v^{\text{in}}(k)]. \quad (9.109)$$

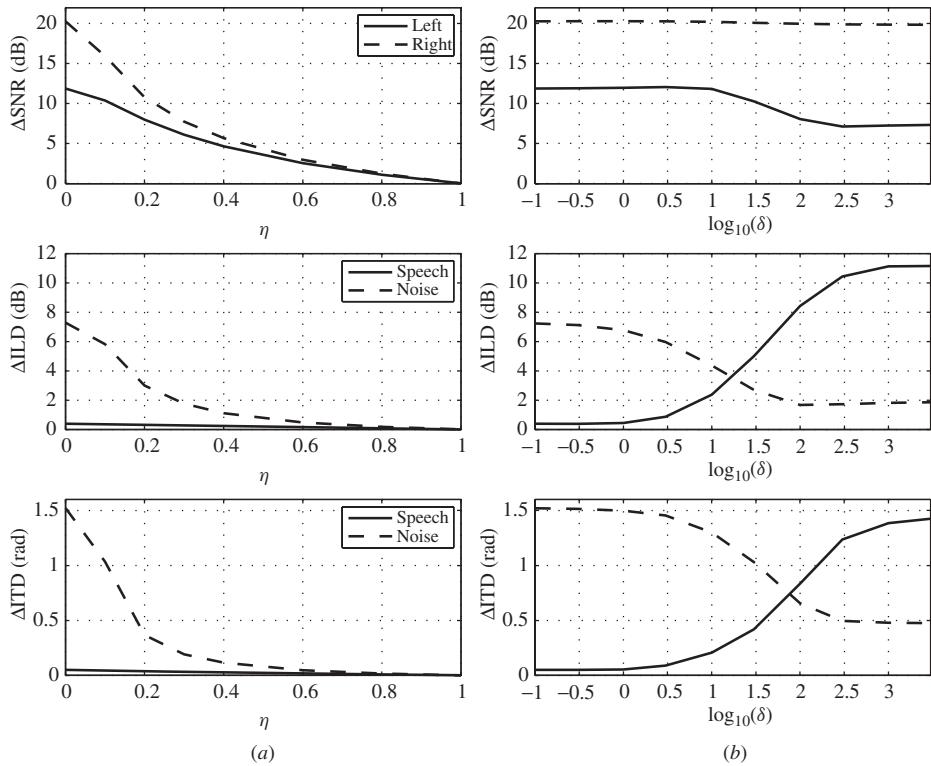
**9.4.5.2 Performance Comparison** For different speech and noise configurations, Figure 9.7 depicts the SNR improvement at the left and right hearing aids for three noise reduction algorithms: (a) a bilateral MWF, that is, two independent MWF algorithms on the left and right hearing aids using two microphones, (b) a binaural MWF using two microphones on the left (right) hearing aid and the front microphone on the right (left) hearing aid, and (c) a binaural MWF using all four microphones from both hearing aids. In general, for all algorithms the SNR improvement is larger when the speech source and the noise source(s) are spatially more separated, with the largest SNR improvement occurring in the hearing aid where the input SNR is lower; for example, for  $\theta_v = 60^\circ$  the SNR improvement is larger in the right hearing aid than in the left hearing aid. Not surprisingly, using more microphones leads to a larger SNR improvement, and this is



**Figure 9.7** SNR improvement at left and right hearing aids for different speech and noise configurations for bilateral MWF (independent processing) and binaural MWF using three and four microphones.

even more pronounced when the speech source and the noise source(s) are close to each other (e.g.,  $\theta_v = 60^\circ$ ) or when multiple noise sources are present.

For the configuration  $\theta_s = 0^\circ$  and  $\theta_v = 60^\circ$ , Figure 9.8 depicts the SNR improvement (left and right hearing aids) and the ILD and ITD error (speech and noise components) for two binaural noise reduction algorithms using all four microphones: (a) the partial noise estimation SDW-MWF- $\eta$  as a function of the parameter  $\eta$  and (b) the SDW-MWF-ITF as a function of the parameter  $\delta$  (we assume  $\gamma = 0$ ). First, it can be observed that for the standard binaural MWF (i.e.,  $\eta = 0$  or  $\delta = 0$ ) the ILD and ITD errors for the speech component are very small, whereas the ILD and ITD errors for the noise component are large, that is, the binaural speech cues are preserved and the binaural noise cues are distorted. When increasing  $\eta$  for the SDW-MWF- $\eta$ , the ILD and ITD errors for the noise component decrease, but the SNR improvement also decreases substantially. However, perceptual experiments have shown that, for example, for  $\eta = 0.2$  it is possible to improve the sound localization performance while hardly reducing the speech intelligibility [105, 109]. When increasing  $\delta$  for the SDW-MWF-ITF, the ILD and ITD errors for the noise component also decrease and the effect on the SNR improvement is smaller than for the SDW-MWF- $\eta$ , but the ILD and ITD errors for the speech component increase. However, initial perceptual experiments still show an advantageous effect [51], which may be due to the frequency dependence of the output ITF on the output SNR, as explained in Section 9.4.4.



**Figure 9.8** SNR improvement and ILD and ILD error for (a) SDW-MWF- $\eta$  and (b) SDW-MWF-ITF ( $\theta_s = 0^\circ$ ,  $\theta_v = 60^\circ$ ).

## 9.5 CONCLUSION

In this chapter we reviewed several multimicrophone beamforming techniques that can be used for noise reduction in monaural and binaural hearing aids. For monaural noise reduction we focused on the transfer function GSC and the multichannel Wiener filter, which are derived from different cost functions but are in fact closely related. For binaural noise reduction we presented several extensions of the standard binaural MWF that aim to preserve the binaural cues of the speech and the noise sources.

## REFERENCES

- V. Hamacher, J. Chalupper, J. Eggers, E. Fischer, U. Kornagel, H. Puder, and U. Rass, "Signal processing in high-end hearing aids: State of the art, challenges, and future trends," *EURASIP J. Appl. Signal Process.*, vol. 18, pp. 2915–2929, 2005.
  - T. Van den Bogaert, T. J. Klasen, M. Moonen, L. Van Deun, and J. Wouters, "Horizontal localisation with bilateral hearing aids: Without is better than with," *J. Acoust. Soc. Am.*, vol. 119, no. 1, pp. 515–526, Jan. 2006.
  - R. Beutelmann and T. Brand, "Prediction of speech intelligibility in spatial noise and reverberation for normal-hearing and hearing-impaired listeners," *J. Acoust. Soc. Am.*, vol. 120, no. 1, pp. 331–342, July 2006.

4. J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localisation*, MIT Press, Cambridge MA, 1983.
5. A. S. Bregman, *Auditory Scene Analysis*, MIT Press, Cambridge MA, 1990.
6. A. W. Bronkhorst and R. Plomp, "The effect of head-induced interaural time and level differences on speech intelligibility in noise," *J. Acoust. Soc. Am.*, vol. 83, no. 4, pp. 1508–1516, Apr. 1988.
7. M. L. Hawley, R. Y. Litovsky, and Colburn, "Speech intelligibility and localization in a multisource environment," *J. Acoust. Soc. Am.*, vol. 105, no. 6, pp. 3436–3448, June 1999.
8. J. Bitzer and K. U. Simmer, "Superdirective microphone arrays," in *Microphone Arrays: Signal Processing Techniques and Applications* M. S. Brandstein and D. B. Ward (Eds.), Springer-Verlag, 2001, pp. 19–38.
9. S. Doclo and M. Moonen, "Design of broadband beamformers robust against gain and phase errors in the microphone array characteristics," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2511–2526, Oct. 2003.
10. S. Doclo and M. Moonen, "Design of far-field and near-field broadband beamformers using eigenfilters," *Signal Process.*, vol. 83, no. 12, pp. 2641–2673, Dec. 2003.
11. G. Elko, "Superdirectional microphone arrays," in *Acoustic Signal Processing for Telecommunication*, S. L. Gay and J. Benesty (Eds.), Kluwer Academic, Boston, 2000, pp. 181–237.
12. E. E. Jan and J. Flanagan, "Sound capture from spatial volumes: Matched-filter processing of microphone arrays having randomly-distributed sensors," in Proc. *IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Atlanta GA, May 1996, pp. 917–920.
13. M. Kajala and M. Hämäläinen, "Broadband beamforming optimization for speech enhancement in noisy environments," in Proc. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, Oct. 1999, pp. 19–22.
14. D. B. Ward, R. A. Kennedy, and R. C. Williamson, "Theory and design of broadband sensor arrays with frequency invariant far-field beam patterns," *J. Acoust. Soc. Am.*, vol. 97, no. 2, pp. 91–95, Feb. 1995.
15. J. M. Kates, "Superdirective arrays for hearing aids," *J. Acoust. Soc. Am.*, vol. 94, no. 4, pp. 1930–1933, Oct. 1993.
16. J. M. Kates and M. R. Weiss, "A comparison of hearing-aid array-processing techniques," *J. Acoust. Soc. Am.*, vol. 99, no. 5, pp. 3138–3148, May 1996.
17. W. Soede, A. J. Berkhout, and F. A. Bilsen, "Development of a directional hearing instrument based on array technology," *J. Acoust. Soc. Am.*, vol. 94, no. 2, pp. 785–798, Aug. 1993.
18. M. R. Bai and C. Lin, "Microphone array signal processing with application in three-dimensional hearing," *J. Acoust. Soc. Am.*, vol. 117, no. 4, pp. 2112–2121, Apr. 2005.
19. J. G. Desloge, W. M. Rabinowitz, and P. M. Zurek, "Microphone-array hearing aids with binaural output—Part I: Fixed-processing systems," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 6, pp. 529–542, Nov. 1997.
20. T. Lotter, "Single and multimicrophone speech enhancement for hearing aids," PhD thesis, RWTH Aachen, Germany, Aug. 2004.
21. I. L. D. M. Merks, M. M. Boone, and A. J. Berkhout, "Design of a broadside array for a binaural hearing aid," in Proc. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, Oct. 1997.
22. F. L. Luo, J. Y. Yang, C. Pavlovic, and A. Nehorai, "Adaptive null-forming scheme in digital hearing aids," *IEEE Trans. Signal Process.*, vol. 50, no. 7, pp. 1583–1590, 2002.
23. J. Capon, "High resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, no. 7, pp. 1408–1418, Aug. 1969.

24. O. L. Frost III, "An algorithm for linearly constrained adaptive array processing," *Proc. IEEE*, vol. 60, no. 8, pp. 926–935, Aug. 1972.
25. I. Claesson and S. Nordholm, "A spatial filtering approach to robust adaptive beaming," *IEEE Trans. Antennas Propagat.*, vol. 40, no. 9, pp. 1093–1096, Sept. 1992.
26. I. Cohen, S. Gannot, and B. Berdugo, "An integrated real-time beamforming and postfiltering system for non-stationary noise environments," *EURASIP J. Appl. Signal Process.*, vol. 11, pp. 1064–1073, Oct. 2003.
27. S. Gannot, D. Burshtein, and E. Weinstein, "Signal enhancement using beamforming and non-stationarity with applications to speech," *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1614–1626, Aug. 2001.
28. S. Gannot and I. Cohen, "Adaptive beamforming and postfiltering," in *Springer Handbook of Speech Processing*, Springer, 2007.
29. W. Herboldt and W. Kellermann, "Adaptive beamforming for audio signal acquisition," in *Adaptive Signal Processing: Applications to Real-World Problems*, J. Benesty and Y. Huang (Eds.), Springer-Verlag, 2003, pp. 155–194.
30. O. Hoshuyama, A. Sugiyama, and A. Hirano, "A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2677–2684, Oct. 1999.
31. S. Nordebo, I. Claesson, and S. Nordholm, "Adaptive beamforming: Spatial filter designed blocking matrix," *IEEE J. Ocean. Eng.*, vol. 19, no. 4, pp. 583–590, Oct. 1994.
32. D. Van Compernolle, "Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Albuquerque NM, Apr. 1990, pp. 833–836.
33. H. Cox, R. M. Zeskind, and M. M. Owen, "Robust adaptive beamforming," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 35, no. 10, pp. 1365–1376, Oct. 1987.
34. S. Doclo, A. Spriet, J. Wouters, and M. Moonen, "Frequency-domain criterion for speech distortion weighted multichannel Wiener filter for robust noise reduction," *Speech Commun.*, vol. 49, nos. 7/8, pp. 636–656, Jul.–Aug. 2007.
35. M. W. Hoffman and K. M. Buckley, "Robust time-domain processing of broadband microphone array data," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 3, pp. 193–203, May 1995.
36. O. Hoshuyama, B. Begasse, and A. Sugiyama, "A new adaptation-mode control based on cross correlation for a robust adaptive microphone array," *IEICE Trans. Fund.*, vol. E84-A, no. 2, pp. 406–413, Feb. 2001.
37. N. K. Jablon, "Adaptive beamforming with the Generalized Sidelobe Canceller in the presence of array imperfections," *IEEE Trans. Antennas Propagat.*, vol. 34, no. 8, pp. 996–1012, Aug. 1986.
38. A. Spriet, M. Moonen, and J. Wouters, "Spatially pre-processed speech distortion weighted multi-channel Wiener filtering for noise reduction," *Signal Process.*, vol. 84, no. 12, pp. 2367–2387, Dec. 2004.
39. A. Spriet, M. Moonen, and J. Wouters, "Stochastic gradient-based implementation of spatially preprocessed speech distortion weighted multichannel Wiener filtering for noise reduction in hearing aids," *IEEE Trans. Signal Process.*, vol. 53, no. 3, pp. 911–925, Mar. 2005.
40. M. Kompis and N. Dillier, "Performance of an adaptive beamforming noise reduction scheme for hearing aid applications," *J. Acoust. Soc. Am.*, vol. 109, no. 3, pp. 1123–1143, 2001.
41. J.-B. Maj, J. Wouters, and M. Moonen, "Noise reduction results of an adaptive filtering technique for dual-microphone behind-the-ear hearing aids," *Ear Hearing*, vol. 25, no. 3, pp. 215–229, June 2004.

42. J. Vanden Berghe and J. Wouters, "An adaptive noise canceller for hearing aids using two nearby microphones," *J. Acoust. Soc. Am.*, vol. 103, no. 6, pp. 3621–3626, June 1998.
43. D. P. Welker, J. E. Greenberg, J. G Desloge, and P. M. Zurek, "Microphone-array hearing aids with binaural output—Part II: A two-microphone adaptive system," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 6, pp. 543–551, Nov. 1997.
44. R. Nishimura, Y. Suzuki, and F. Asano, "A new adaptive binaural microphone array system using a weighted least squares algorithm," In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Orlando FL, May 2002, pp. 1925–1928.
45. S. Doclo and M. Moonen, "GSVD-based optimal filtering for single and multimicrophone speech enhancement," *IEEE Trans. Signal Process.*, vol. 50, no. 9, pp. 2230–2244, Sept. 2002.
46. J.-B. Maj, M. Moonen, and J. Wouters, "SVD-based optimal filtering technique for noise reduction in hearing aids using two microphones," *EURASIP J. Appl. Signal Process.*, vol. 4, pp. 432–443, Apr. 2002.
47. A. Spriet, M. Moonen, and J. Wouters, "Robustness analysis of multi-channel Wiener filtering and generalized sidelobe cancellation for multi-microphone noise reduction in hearing aid applications," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 4, pp. 487–503, July 2005.
48. S. Doclo, T. J. Klasen, T. Van den Bogaert, J. Wouters, and M. Moonen, "Theoretical analysis of binaural cue preservation using multi-channel Wiener filtering and interaural transfer functions," in *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, Paris, France, Sept. 2006.
49. T. J. Klasen, T. Van den Bogaert, M. Moonen, and J. Wouters, "Binaural noise reduction algorithms for hearing aids that preserve interaural time delay cues," *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1579–1585, Apr. 2007.
50. S. Doclo, R. Dong, T. J. Klasen, J. Wouters, S. Haykin, and M. Moonen, "Extension of the multi-channel Wiener filter with localisation cues for noise reduction in binaural hearing aids," in *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, Eindhoven, The Netherlands, Sept. 2005, pp. 221–224.
51. T. Van den Bogaert, S. Doclo, M. Moonen, and J. Wouters, "Binaural cue preservation for hearing aids using an interaural transfer function multichannel Wiener filter," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu HI, Apr. 2007, pp. 565–568.
52. S. Doclo, T. Van den Bogaert, M. Moonen, and Jan Wouters, "Reduced-bandwidth and distributed MWF-based noise reduction algorithms for binaural hearing aids," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 17, no. 1, pp. 38–51, Jan. 2009.
53. A. J. Bell and T. J. Sejnowski, "An information-maximisation approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, no. 6, pp. 1004–1034, 1995.
54. P. Comon. Independent component analysis, a new concept? *Signal Processing*, vol. 36, no. 3, pp. 287–314, Apr. 1994.
55. H. Buchner, R. Aichner, and W. Kellermann, "A generalization of blind source separation algorithms for convolutive mixtures based on second-order statistics," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 1 pp. 120–134, Jan. 2005.
56. S. Makino, "Blind source separation of convolutive mixtures of speech," in *Adaptive Signal Processing: Applications to Real-World Problems*, Springer-Verlag, 2003.
57. L. Parra and C. Spence, "Convulsive blind separation of non-stationary sources," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 3, pp. 320–327, May 2000.
58. G. J. Brown and D. L. Wang, "Separation of speech by computational auditory scene analysis," in *Speech Enhancement*, Springer-Verlag, 2005, pp. 371–402.

59. D. Ellis, "Prediction-driven computational auditory scene analysis," PhD thesis, MIT, Cambridge, MA, 1996.
60. D. F. Rosenthal and H. G. Okun, *Computational Auditory Scene Analysis*, Lawrence Erlbaum Associates, 1998.
61. G. N. Hu and D. L. Wang, "Monaural speech segregation based on pitch tracking and amplitude modulation," *IEEE Trans. Neural Networks*, vol. 15, no. 5, pp. 1135–1150, Sept. 2004.
62. B. Kollmeier and R. Koch, "Speech enhancement based on physiological and psychoacoustical models of modulation perception and binaural interaction," *J. Acoust. Soc. Am.*, vol. 95, no. 3, pp. 1593–1602, Mar. 1994.
63. T. Nakatani and H. G. Okuno, "Harmonic sound stream segregation using localisation and its application to speech stream segregation," *Speech Commun.*, vol. 27, nos. 3/4, pp. 209–222, Apr. 1999.
64. N. Roman, D. L. Wang, and G. J. Brown, "Speech segregation based on sound localization," *J. Acoust. Soc. Am.*, vol. 114, no. 4, pp. 2236–2252, Oct. 2003.
65. T. Wittkop, "Two-channel noise reduction algorithms motivated by models of binaural interaction," PhD thesis, Universität Oldenburg, Germany, Mar. 2001.
66. E. Jan and J. Flanagan, "Microphone arrays for speech processing," in *Proc. URSI International Symposium on Signals, Systems, and Electronics*, San Francisco CA, Oct. 1995, pp. 373–376.
67. D. Rabinkin, R. Renomeron, J. Flanagan, and D. F. Macomber, "Optimal truncation time for matched filter array processing," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Seattle WA, May 1998, pp. 3260–3272.
68. B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Mag.*, Apr. 1988, pp. 4–24.
69. H. Cox, "Resolving power and sensitivity to mismatch of optimum array processors," *J. Acoust. Soc. Am.*, vol. 54, no. 3, pp. 771–785, Sept. 1973.
70. L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. Antennas Propagat.*, vol. 30, no. 1, pp. 27–34, Jan. 1982.
71. J. E. Greenberg and P. M. Zurek, "Evaluation of an adaptive beamforming method for hearing aids," *J. Acoust. Soc. Am.*, vol. 91, no. 3, pp. 1662–1676, Mar. 1992.
72. B. R. Breed and J. Strauss, "A short proof of the equivalence of LCMV and GSC beamforming," *IEEE Signal Process. Lett.*, vol. 9, no. 6, pp. 168–169, June 2002.
73. G. Reuven, S. Gannot, and I. Cohen, "Dual-source transfer-function generalized sidelobe canceller," *IEEE Trans. Audio Speech Lang. Process.*, vol. 16, no. 4, pp. 711–727, May 2008.
74. S. Gannot, D. Burshtein, and E. Weinstein, "Analysis of the power spectral deviation of the general transfer function GSC," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 1115–1121, Apr. 2004.
75. J. Bitzer, K.-D. Kammeyer, and K. U. Simmer, "An alternative implementation of the superdirective beamformer," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz NY, Oct. 1999, pp. 7–10.
76. S. Nordholm, I. Claesson, and P. Eriksson, "The broadband Wiener solution for Griffiths-Jim beamformers," *IEEE Trans. Signal Process.*, vol. 40, no. 2, pp. 474–478, Feb. 1992.
77. S. Haykin, *Adaptive Filter Theory*, 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, 1996.
78. R. E. Crochiere, "A weighted overlap-add method for short-time Fourier analysis/synthesis," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 28, no. 1, pp. 99–102, Feb. 1980.

79. J. J. Shynk, "Frequency-domain and multirate and adaptive filtering," *IEEE Signal Process. Mag.*, vol. no. 1 pp. 14–37, Jan. 1992.
80. O. Shalvi and E. Weinstein, "System identification using nonstationary signals," *IEEE Trans. Signal Process.*, vol. 44, no. 8, pp. 2055–2063, Aug. 1996.
81. C. Marro, Y. Mahieux, and K. U. Summer, "Analysis of noise reduction and dereverberation techniques based on microphone arrays with postfiltering," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 3, pp. 240–259, May 1998.
82. K. U. Simmer, J. Bitzer, and C. Marro, "Post-filtering techniques," in *Microphone Arrays: Signal Processing Techniques and Applications*, M. S. Brandstein and D. B. Ward (Eds.), Springer-Verlag, May 2001, pp. 39–57.
83. S. Nordholm, I. Claesson, and M. Dahl, "Adaptive microphone array employing calibration signals: An analytical evaluation," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 3, pp. 241–252, May 1999.
84. S. Nordholm, I. Claesson, and N. Grbíć, "Optimal and adaptive microphone arrays for speech input in automobiles," 14 in *Microphone Arrays: Signal Processing Techniques and Applications*, M. S. Brandstein and D. B. Ward (Eds.), Springer-Verlag, May 2001, pp. 307–326.
85. G. Rombouts and M. Moonen, "QRD-based unconstrained optimal filtering for acoustic noise reduction," *Signal Process.*, vol. 83, no. 9, pp. 1889–1904, Sept. 2003.
86. A. Spriet, M. Moonen, and J. Wouters, "The impact of speech detection errors on the noise reduction performance of multi-channel Wiener filtering and Generalized Sidelobe Cancellation," *Signal Process.*, vol. 85, no. 6, pp. 1073–1088, June 2005.
87. R. Aichner, W. Herbordt, H. Buchner, and W. Kellermann, "Least-squares error beamforming using minimum statistics and multichannel frequency-domain adaptive filtering," in *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, Kyoto, Japan, Sept. 2003, pp. 223–226.
88. N. Grbíć and S. Nordholm, "Soft constrained subband beamforming for hands-free speech enhancement," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Orlando FL, May 2002, pp. 885–888.
89. R. Balan and J. Rosca, "Microphone array speech enhancement by Bayesian estimation of spectral amplitude and phase," in *Proc. Sensor Array and Multichannel Signal Processing Workshop*, Aug. 2002, pp. 209–213.
90. Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 32, no. 6, pp. 1109–1121, Dec. 1984.
91. Y. Ephraim and D. Malah, "Speech Enhancement Using a Minimum Mean-Square Error Log-Spectral Amplitude Estimator. *IEEE Trans. Acoust. Speech Signal Process.*", vol. 33, no. 2, pp. 443–445, Apr. 1985.
92. W. Herbordt and W. Kellermann, "Frequency-domain integration of acoustic echo cancellation and a Generalized Sidelobe Canceller with improved robustness," *Eur. Trans. Telecommun.*, vol. 13, no. 2, pp. 123–132, Mar.–Apr. 2002.
93. Z. Tian, K. L. Bell, and H. L. Van Trees, "A recursive least squares implementation for LCMP beamforming under quadratic constraint *IEEE Trans. Signal Process.*", vol. 49, no. 6, pp. 1138–1145, June 2001.
94. S. Van Gerven and F. Xie, "A comparative study of speech detection methods," in *Proc. EUROSPEECH*, vol. 3, Rhodos, Greece, 1997, pp. 1095–1098.
95. M. Nilsson, S. D. Soli, and A. Sullivan, "Development of the Hearing in Noise Test for the measurement of speech reception thresholds in quiet and in noise," *J. Acoust. Soc. Am.*, vol. 95, no. 2, pp. 1085–1099, Feb. 1994.

96. J. E. Greenberg, P. M. Peterson, and P. M. Zurek, "Intelligibility-weighted measures of speech-to-interference ratio and speech system performance," *J. Acoust. Soc. Am.*, vol. 94, no. 5, pp. 3009–3010, Nov. 1993.
97. Acoustical Society of America, "American national standard methods for calculation of the speech intelligibility index," ANSI S3.5-1997, June 1997.
98. A. Boothroyd, K. Fitz, J. Kindred, S. Kochkin, H. Levitt, B. C. J. Moore, and J. Yantz, "Hearing aids and wireless technology," *Hearing Rev.*, vol. 14, no. 6, pp. 44–48, June 2007.
99. B. Cornelis, S. Doclo, T. Van den Bogaert, J. Wouters, and M. Moonen, "Theoretical analysis of binaural multi-microphone noise reduction techniques," *IEEE Trans. Audio Speech Lang. Process.*, in press.
100. J. Benesty, J. Chen, A. Huang, and S. Doclo, "New insights into the noise reduction wiener filter," *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 4, pp. 1218–1234, July 2006.
101. B. de Vries and R. A. J. de Vries, "An integrated approach to hearing aid algorithm design for enhancement of audibility, intelligibility and comfort," in *Proc. of the IEEE Benelux Signal Processing Symposium (SPS2004)*, Hilvarenbeek, The Netherlands, Apr. 2004, pp. 65–68.
102. S. Doclo and M. Moonen, "On the output SNR of the speech-distortion weighted multichannel Wiener filter," *IEEE Signal Process. Lett.*, vol. 12, no. 12, pp. 809–811, Dec. 2005.
103. T. J. Klasen, S. Doclo, T. Van den Bogaert, M. Moonen, and J. Wouters, "Binaural multi-channel Wiener filtering for hearing aids: Preserving interaural time and level differences," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, May 2006, pp. 145–148.
104. B. Cornelis, S. Doclo, T. Van den Bogaert, M. Moonen, and J. Wouters, "Analysis of localization cue preservation by multichannel Wiener filtering based binaural noise reduction in hearing aids," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Lausanne, Switzerland, Aug. 2008.
105. T. Van den Bogaert, S. Doclo, M. Moonen, and J. Wouters, "The effect of multimicrophone noise reduction systems on sound source localization by users of binaural hearing aids," *J. Acoust. Soc. Am.*, vol. 124, no. 1, pp. 484–497, July 2008.
106. R. Talmon, I. Cohen, and S. Gannot, "Relative Transfer Function Identification Using Convolutional Transfer Function Approximation," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 17, no. 4, pp. 546–555, May 2009.
107. S. Markovich, S. Gannot, and I. Cohen, "Multichannel Eigenspace Beamforming in a Reverberant Noisy Environment With Multiple Interfering Speech Signals," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 17, no. 6, pp. 1071–1086, Aug. 2009.
108. E.A.P. Habets, J. Benesty, I. Cohen, and S. Gannot, "On a tradeoff between dereverberation and noise reduction using the MVDR beamformer," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 3741–3744.
109. T. Van den Bogaert, S. Doclo, J. Wouters, and Marc Moonen, "Speech enhancement with multichannel Wiener filter techniques in multimicrophone binaural hearing aids," *J. Acoust. Soc. Am.*, vol. 125, no. 1, pp. 360–371, Jan. 2009.

# **Underdetermined Blind Source Separation Using Acoustic Arrays**

Shoji Makino, Shoko Araki, Stefan Winter, and Hiroshi Sawada

NTT Communication Science Laboratories

## **10.1 INTRODUCTION**

People can engage in comprehensible conversations at a noisy cocktail party. This is the well-known “cocktail-party effect,” whereby our ears can extract what a person is saying under such conditions. The aim of blind source separation (BSS) [1] for speech applications is to provide computers with this ability, thus enabling them to determine individual speech waveforms from mixtures.

Blind source separation has already been applied to various problems including the wireless communication and biomedical fields. However, as speech signal mixtures in a natural (i.e., reverberant) environment are generally convolutive mixtures, they involve a task that is structurally much more challenging than instantaneous mixtures, which are prevalent in many other applications. BSS is an approach for estimating source signals using only information about their mixtures observed at each sensor. The estimation is performed without possessing information on individual sources, such as their location, frequency characteristics, and how they are mixed. The BSS technique for speech dealt with in this chapter has many applications including hands-free teleconference systems and automatic meeting minute generators.

Classical approaches for (over)determined BSS, where the number of sensors  $M$  is equal to or more than the number of sources  $N$ , are based on independent component analysis (ICA) [2]. They rely solely on the assumption that the source signals are mutually independent. Many ICA methods have been proposed for the convolutive BSS problem [2–9]. ICA works well even under reverberant conditions.

The area of underdetermined BSS, that is, BSS with fewer sensors  $M$  than sources  $N$  ( $M < N$ ), has attracted increasing attention [10–32]. Underdetermined BSS is a challenging problem because of the inherently adverse conditions, that is, the mixing system is not invertible and so ICA cannot be used. We cannot obtain the source signals simply by inverting the mixing matrix. Therefore, even if we know the mixing matrix exactly, we cannot recover the exact values of the source signals. This is because information is lost in the mixing process. This problem can be understood in

a comprehensive way in that we have  $N$  unknowns, while we have only  $M$  equations ( $M < N$ ), that is, the simultaneous equations become underdetermined and therefore cannot be solved.

The additional effort needed to separate the mixtures in underdetermined BSS requires more refined source models and further assumptions. Here, the sparseness of speech sources is very useful; we can utilize time–frequency diversity, where sources are active in different regions of the time–frequency plane. The sparseness means that only a few samples of the original signals have a value significantly different from zero [32]. The sparseness-based approaches are attractive because they can handle the underdetermined problem. A few recent approaches consider convolutive mixtures [14, 16, 33–35]. These approaches are based on algorithms originally designed for instantaneous mixtures, which are adapted to convolutive mixtures by binwise BSS in the time–frequency domain. The goal of this chapter is to provide an overview of underdetermined BSS for convolutive mixtures of speech. We also assess the sparseness and anechoic assumptions upon which the underdetermined BSS approaches rely for speech signals under different reverberant conditions.

The sparseness-based underdetermined BSS approaches can be divided into two main categories. One method extracts each signal with a time–frequency binary mask [10, 36–38], and the other is based on maximum a posteriori (MAP) estimation, where the sources are estimated after mixing matrix estimation [11–14, 39–41]. Both methods can be easily employed for complex-valued mixture samples that occur when BSS is performed in the time–frequency domain. Furthermore, neither method requires sensor location information or limits the usable number of sensors. This makes it easy for us to employ freely arranged multiple sensors. Therefore, both methods can separate signals that are distributed two or three dimensionally.

The former method is the binary mask approach. With this approach, we assume that signals are sufficiently sparse, and therefore, we can also assume that at most one source is dominant at each time–frequency point. If the sparseness assumption holds, and if we can assume an anechoic situation, we can estimate the geometrical information about the dominant source at each time–frequency point. The geometrical information is estimated by using the level ratio and phase differences between sensors. When we consider this information for all time–frequency points, the points can be grouped into  $N$  clusters [10, 36, 38]. Because an individual cluster corresponds to an individual source, we can separate each signal by selecting the observation signal at time–frequency points in each cluster with a binary mask. The best-known approach may be the degenerate unmixing estimation technique (DUET) [10, 36, 42]. It has the advantage of being implemented in real time [42]. This chapter introduces an extended binary mask approach called MENUET (multiple sensor duet), which employs the  $k$ -means clustering algorithm. The level ratios and phase differences between multiple observations are employed as clustering features. To employ the  $k$ -means algorithm successfully, the variances of the level ratios and phase differences are set at a comparable level.

The latter method is a MAP-based two-stage approach. The method includes two different problems of mixing matrix estimation and  $l_1$ -norm minimization in the frequency domain (i.e., for complex numbers). In blind system identification (BSI), which is the first stage, we estimate the mixing matrix. We employed hierarchical clustering for this task. In the second stage, namely blind source recovery (BSR), we separate the mixtures using the estimated mixing matrix from the first stage. We assumed

statistical independence for the sources. This led to a constrained  $l_1$ -norm minimization with complex numbers as a result of the time–frequency domain approach. The MAP approach can be understood that it implicitly removes  $N - M$  sources from the mixture and separates the remaining  $M$  sources. We also investigated on *explicit*  $N - M$  source removal approach. We show that the performance of the  $N - M$  source removal approach is comparable to or even better than that of the second-order cone programming (SOCP) solution. In addition, the  $N - M$  source removal approach has the advantage of being faster for underdetermined BSS problems with low input–output dimensions.

For the sake of completeness, computational auditory scene analysis (CASA) [43, 44] should also be mentioned. It has similarities with sparseness-based approaches and is also able to separate sound source mixtures without any knowledge of the mixing system. It is not usually considered to be a BSS technique. For this reason, it will not be further considered here.

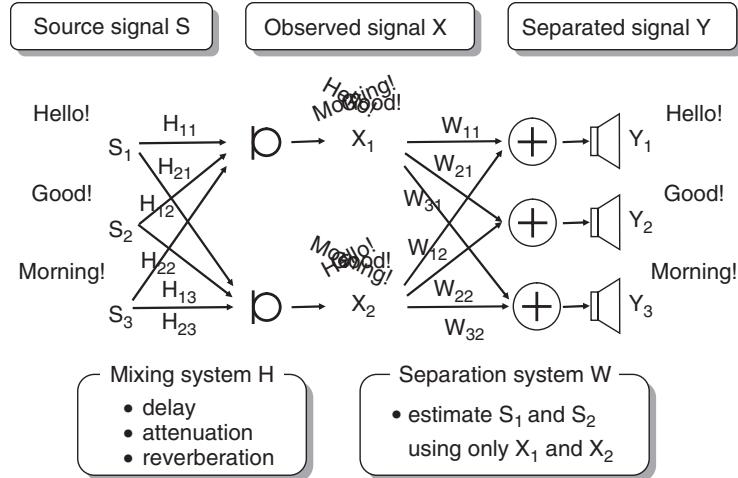
The organization of this chapter is as follows. Section 10.2 introduces the underlying model and objective of underdetermined BSS of speeches in reverberant environments. In Section 10.3, we investigate the relevance of the sparseness of the speech sources and anechoic assumptions upon which the underdetermined BSS relies. Section 10.4 presents the basic framework of the binary mask-based underdetermined BSS method. We describe features for clustering, and test how these features will be clustered by the  $k$ -means clustering algorithm. The MENUET method with  $k$ -means clustering and multiple sensors is described in this section. In Section 10.5, we present the MAP-based two-stage approach. The method consists of two stages of mixing matrix estimation and  $l_1$ -norm minimization in the frequency domain. The  $l_1$ -norm minimization can be solved by  $N - M$  source removal approach and SOCP approach. Section 10.6.1 reports experimental comparison results obtained with nonlinearly arranged sensors in underdetermined scenarios. Even when the sources and sensors were distributed two or three dimensionally, we obtained good separation results with both methods for each scenario under reverberant conditions ( $RT_{60} = 128$  and 300 ms). The final section concludes this chapter.

## 10.2 UNDERDETERMINED BLIND SOURCE SEPARATION OF SPEECHES IN REVERBERANT ENVIRONMENTS

In our treatment of underdetermined BSS, we assume a linear mixture model with negligible noise. In the following, we introduce the underlying model and objective of underdetermined BSS.

### 10.2.1 Discrete Time-Domain Representation

For speech source separation, several sensor microphones are placed in different positions so that each records a mixture of the original source signals at a slightly different time and level. In the real world where the source signals are speech and the mixing system is a room, the signals that are picked up by the microphones are affected by reverberation. With  $t$  denoting discrete time,  $s_k(t)$  denoting the  $k$ th unobservable source signal ( $1 \leq k \leq N$ ), and  $h_{jk}(l)$  ( $1 \leq l \leq L$ ) denoting the  $L$ -tap impulse response from



**Figure 10.1** Underdetermined BSS system configuration.

source  $k$  to sensor  $j$  ( $1 \leq j \leq M$ ),  $N$  sources  $s_1, \dots, s_N$  are convolutively mixed and observed at  $M$  sensors  $x_1, \dots, x_M$ :

$$x_j(t) = \sum_{k=1}^N \sum_{l=1}^L h_{jk}(l)s_k(t-l), \quad j=1, \dots, M. \quad (10.1)$$

This chapter assumes that  $N$  and  $M$  are known, and that the sensor spacing is sufficiently small to avoid the spatial aliasing problem. The goal for BSS is to obtain separated signals  $y_k(t)$  that are estimations of  $s_k(t)$  solely from  $M$  observations. In this chapter, we focus particularly on an underdetermined situation where there are fewer sensors  $M$  than sources  $N$  ( $M < N$ ) (Fig. 10.1).

### 10.2.2 Discrete Fourier Transform Domain Representation

Instead of solving the problem in the time domain, we switch to the discrete Fourier transform (DFT) domain by applying a short-time Fourier transform (STFT) to the mixtures  $x_j(t)$ . The time-domain signals

$$\mathbf{s}(t) = [s_1(t), \dots, s_N(t)]^\top \quad (10.2)$$

and

$$\mathbf{x}(t) = [x_1(t), \dots, x_M(t)]^\top \quad (10.3)$$

are converted into frequency-domain time series

$$\mathbf{s}(f, t) = [s_1(f, t), \dots, s_N(f, t)]^\top \quad (10.4)$$

and

$$\mathbf{x}(f, t) = [x_1(f, t), \dots, x_M(f, t)]^\top \quad (10.5)$$

by a  $T$ -point STFT, respectively. Thereby  $f = 0, f_s/T, \dots$ , and  $f_s(T-1)/T$  ( $f_s$ : sampling frequency;  $t$ : time dependence). Let us define  $\mathbf{H}(f)$  as a matrix whose elements are transformed impulse responses. We call the column vectors  $\mathbf{h}_k(f) = [h_{1k}, \dots, h_{Mk}]^T$  ( $k = 1, \dots, N$ ) mixing vectors and approximate the mixing process by

$$\mathbf{x}(f, t) = \mathbf{H}(f)\mathbf{s}(f, t) = [\mathbf{h}_1(f), \dots, \mathbf{h}_N(f)]\mathbf{s}(f, t). \quad (10.6)$$

This reduces the BSS problem from convolutive to instantaneous mixtures in each frequency bin  $f$ .

### 10.2.3 Objective of Underdetermined BSS

The ultimate objective of underdetermined BSS is the estimation of signals  $y_k(t)$  ( $1 \leq k \leq N$ ) that is an estimation of the original signals  $s_k(t)$  as closely as possible up to arbitrary permutation and scaling, when only the mixtures  $x_j(t)$  are available. In contrast to (over)determined BSS, here a knowledge or estimation of the impulse responses  $h_{jk}(t)$  or their frequency-domain representation  $\mathbf{H}(f)$  is not sufficient. Even if the impulse responses are available, estimating the source signals from the observed signals itself poses a problem as follows.

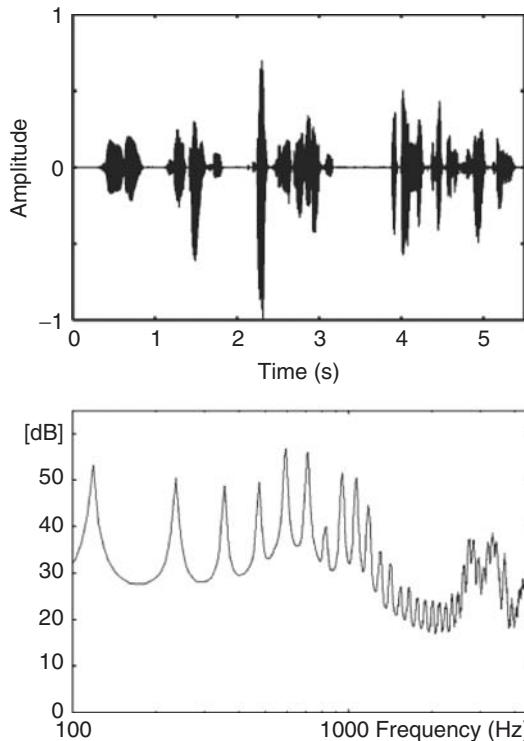
With (over)determined BSS, the mixing matrix  $\mathbf{H}(f)$  is quadratic ( $M = N$ ). Therefore, it can be inverted assuming that it is nonsingular, and (10.6) can be solved for  $\mathbf{s}(f, t)$ . On the other hand, the mixing matrix  $\mathbf{H}(f)$  is not quadratic with underdetermined BSS ( $M < N$ ) but has more columns than rows, that is, the left-side inverse does not exist. As a consequence, (10.6) cannot be solved for  $\mathbf{s}(f, t)$ . Instead,  $\mathbf{s}(f, t)$  can only be approximated based on a knowledge of the mixtures  $\mathbf{x}(f, t)$  and the mixing matrix  $\mathbf{H}(f)$ . The exact result depends on the assumptions made for the approximation.

## 10.3 SPARSENESS OF SPEECH SOURCES

Most approaches to underdetermined BSS are based on the assumption of disjoint sparse sources [16, 45–53]. It is then assumed that the sparse signal representations exhibit little or no overlap in the time–frequency plane, so that mixtures considered in small areas of the time–frequency plane are actually not underdetermined. Different properties such as amplitude ratio or phase difference are used to assign each sample of the mixtures to a limited number of sources. The result can be used in different ways: (1) either the sources can be picked out directly or (2) the result is utilized to estimate the mixing matrix, which in turn is used for estimating the source signals.

First, let us investigate the sparseness of speech signals. Switching to the time–frequency domain offers the additional advantage of facilitating the exploitation of the time–frequency sparseness of speech signals [11]. In general, the sparseness of a signal means that only a few samples have a value significantly different from zero.

The sparseness of speech signals in the time–frequency domain results from the fact that natural speech signals always include pauses over time and even during voice activity (Fig. 10.2, left). Some spectral regions exhibit only low intensity following the harmonic structure of speech spectra with energy concentrated in formants and at multiples of the pitch frequency (Fig. 10.2, right). This means that in the time–frequency



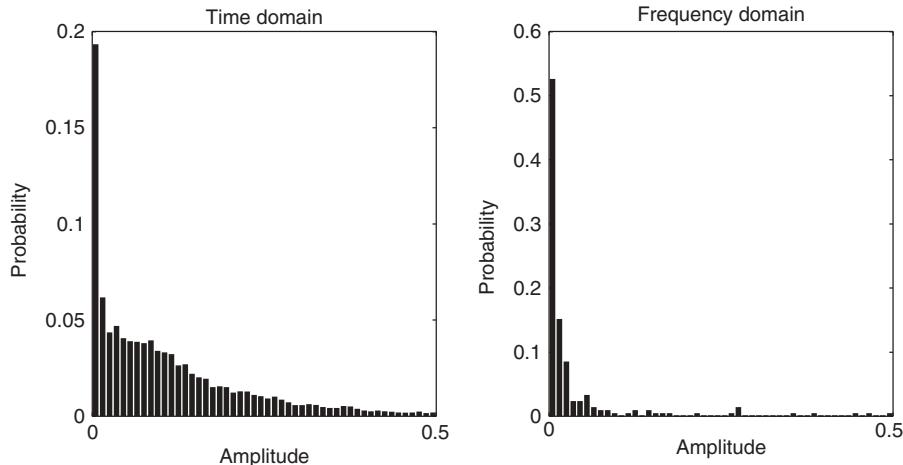
**Figure 10.2** Speech signal in (left) time domain and (right) frequency domain.

domain, only a few frequency bins have significant energy at each time  $t$ , while most frequency bins have values close to zero.

Note that sparseness itself is not yet sufficient to be helpful in underdetermined BSS. For example, if two speech signals are sparse but have their energy concentrated at identical time–frequency points, the two signals would still overlap. Therefore, the more precise concept of disjoint sparseness has been introduced, which in addition to sparseness requires that the involved signals do not have energy concentrated at identical time–frequency points [46]. Speech signals fit the concept of disjoint sparseness since the pitch frequency depends on the speaker and also changes over time.

Using a sparse signal representation is very important as regards ensuring good separation performance since the separation is built on the assumption of sparse source signals. Intuitively, we can expect less overlap between sources as the representation for each source becomes sparser. Therefore, it is essential that we have a signal representation that maximizes sparseness with minimum loss of signal content.

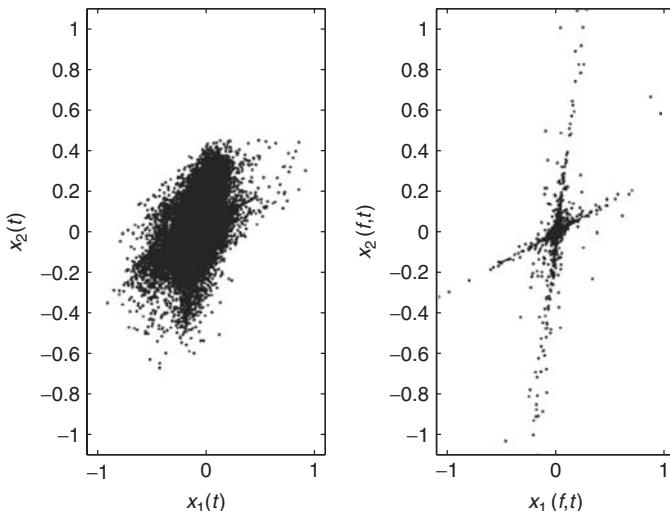
It has already been shown that speech sources are sparser in the time–frequency domain than in the time domain [11, 37]. Figure 10.3 shows an example histogram of signal amplitudes for a time-domain speech source signal (5 s of male speech) and its time–frequency domain representation at a frequency  $f$  of 1 kHz. We can see that the distribution of the time-domain signal has a heavier tail than that of the time–frequency domain signal. This means that the possibility of a time–frequency domain signal being close to zero is much higher than that of a time-domain signal.



**Figure 10.3** Amplitude histogram for (left) a time-domain signal and (right) a time–frequency domain signal at  $f = 1\text{ kHz}$ . The frame length for STFT was  $T = 512$  at 8 kHz sampling. Outliers are deleted for better visualization.

Figure 10.4 provides another illustration showing that the time–frequency domain representation is sparser than the time-domain representation. This is a scatter plot for two speech mixtures. The generative mixing model takes the form

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = [\mathbf{h}_1 \quad \mathbf{h}_2] \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}. \quad (10.7)$$



**Figure 10.4** Scatter plot for (left) a time-domain signal and (right) a time–frequency domain signal at  $f = 1\text{ kHz}$ . The frame length for STFT was  $T = 512$  at 8 kHz sampling.

The horizontal axis is the observation at sensor 1:  $x_1 = h_{11}s_1 + h_{12}s_2$ , the vertical axis shows the sensor 2 observation:  $x_2 = h_{21}s_1 + h_{22}s_2$ . Here,  $s_1$  and  $s_2$  were two female speech signals of 5 s each, and

$$\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} 1.0 & 0.2 \\ 0.3 & 0.9 \end{bmatrix} \quad (10.8)$$

was utilized for simplicity. For our purpose, we can obtain a useful formulation of the mixing system by decomposing the mixing matrix into its column  $\mathbf{h}_k$  and expanding it for every data point

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} h_{11} \\ h_{21} \end{bmatrix} s_1 + \begin{bmatrix} h_{12} \\ h_{22} \end{bmatrix} s_2 = \mathbf{h}_1 s_1 + \mathbf{h}_2 s_2. \quad (10.9)$$

Then, in the  $M$ -dimensional mixture space, the mixing vectors  $\mathbf{h}_k$  define the spatial information of the sources. From (10.9), it is obvious that if only one source is nonzero and the other source is zero, for example  $s_1 \neq 0$  and  $s_2 = 0$ , then the resultant “mixture” would be

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} h_{11} \\ h_{21} \end{bmatrix} s_1 = \mathbf{h}_1 s_1, \quad (10.10)$$

therefore, the points on the scatter plot of  $x_1$  versus  $x_2$  show a clear tendency to cluster along the direction of the mixing vector  $\mathbf{h}_1$ . When sources are sufficiently sparse, they rarely overlap, and therefore, we can see two slopes in the time–frequency representation (Fig. 10.4, right). By contrast, no clear structure can be seen in the time domain (Fig. 10.4, left). These lines correspond to the columns  $\mathbf{h}_k$  of the mixing matrix  $\mathbf{H}$ . Therefore, the essence of the mixing matrix estimation is to find the direction of the maximum data density from the observed data [47].

Now let us look more closely at the overlap of speech signals at each time–frequency point. Here we employ  $l_\epsilon^0$ -norm to investigate the signal overlap. The  $l_\epsilon^0$ -norm is defined as [54, 55]

$$\|s(f, t)\|_{0, \epsilon(f)} = \#\{i, |s_k(f, t)| \geq \epsilon(f)\}. \quad (10.11)$$

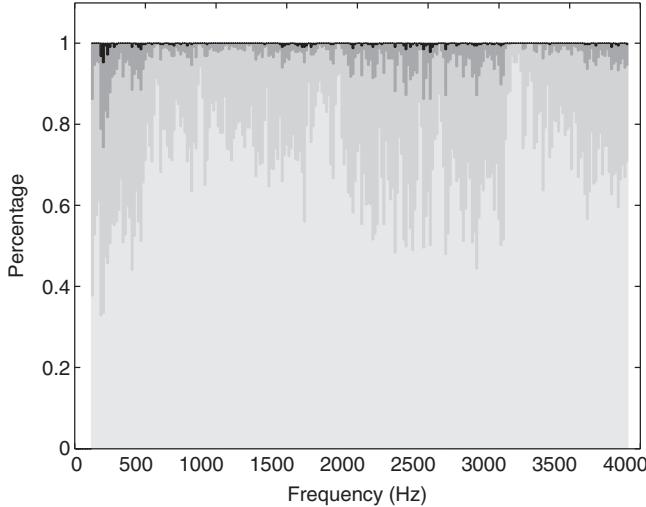
That is,  $l_\epsilon^0$ -norm means the number of sources that have a larger amplitude than a given threshold  $\epsilon$ . Although  $\epsilon$  is set at the noise level in the original  $l_\epsilon^0$ -norm definition, the  $\epsilon$  utilized here was

$$\epsilon(f) = \frac{1}{10} \max_k \max_t |s_k(f, t)|,$$

which gives us  $-20$  dB down from the maximum value at each frequency bin.

Figure 10.5 shows the percentage of frames where  $\|s(f, t)\|_{0, \epsilon(f)} = 0, \dots, N$  for  $N = 3$  sources. At half of the frames, the speech sources are less than  $\epsilon$ . Even in the time frames where the source(s) is (are) active, the frames with only one source are dominant. Only around 10% of the time frames have more than two active sources for  $N = 3$  sources. We can conclude that the speech sources are sufficiently sparse.

From the spectrograms of speech signals (e.g., see Fig. 10.10a below), we can also see intuitively that the speech components seldom overlap.



**Figure 10.5** Percentage of frames where  $l_\epsilon^0$  is  $\|s(f, t)\|_{0, \epsilon(f)} = 0, \dots, N$  for  $N = 3$  source signals. Light gray:  $\|s(f, t)\|_{0, \epsilon(f)} = 0$ , gray:  $\|s(f, t)\|_{0, \epsilon(f)} = 1$ , dark gray:  $\|s(f, t)\|_{0, \epsilon(f)} = 2$ , black:  $\|s(f, t)\|_{0, \epsilon(f)} = 3$ .

### 10.3.1 Sparsest Representation with STFT

For sparseness-based BSS, we should use as sparse a representation as possible. The STFT is the most commonly used transformation for obtaining sparse signal representations.

Let us look for a sparser representation with the STFT. When the frame size  $T$  for the STFT is too large, then the time resolution becomes small and the possibility of several signals existing in a frame increases. On the other hand, if the frame size  $T$  is too small, then the frequency resolution becomes small and a frequency bin contains a wide-band component. As a result, the possibility of multiple sources in each frequency bin again increases. Therefore, there should be an optimal frame size  $T$ .

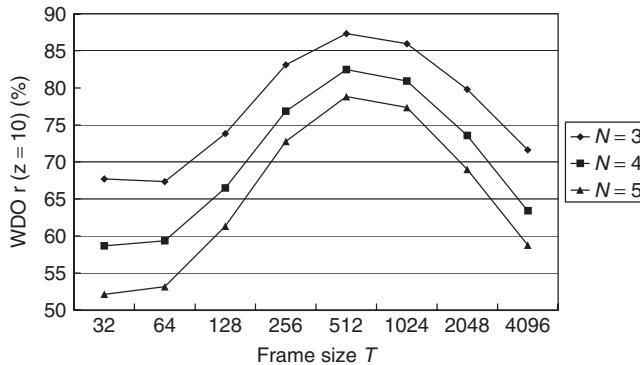
For the sparseness evaluation, this chapter employs a measure called the approximate W-disjoint orthogonality (WDO) [15]:

$$r_k(z) = \frac{\sum_{(f,t)} |\Phi_{(k,z)}(f, t)x_{Jk}(f, t)|^2}{\sum_{(f,t)} |x_{Jk}(f, t)|^2} \times 100[\%]. \quad (10.12)$$

In (10.12),  $x_{Jk}(f, t)$  means the short-time Fourier transformed source  $k$  observed at a certain selected sensor  $J$ :  $x_{Jk}(f, t) = \text{STFT}\left[\sum_l h_{Jk}(l)s_k(t - l)\right]$ . Moreover, in (10.12),  $\Phi_{(k,z)}$  is a time–frequency binary mask that has a parameter  $z$ :

$$\Phi_{(k,z)}(f, t) = \begin{cases} 1, & 20 \log(|x_{Jk}(f, t)|/|\hat{x}_{Jk}(f, t)|) > z, \\ 0, & \text{otherwise,} \end{cases} \quad (10.13)$$

where  $\hat{x}_{Jk}(f, t)$  is the sum of the interference components at sensor  $J$ :  $\hat{x}_{Jk}(f, t) = \text{STFT}\left[\sum_{i=1, i \neq k}^N x_{Ji}(t)\right]$ . The approximate WDO  $r_k(z)$  indicates the percentage of the



**Figure 10.6** Approximate WDO for each frame size  $T$ .  $z = 10$ . Anechoic.

energy of source  $k$  for time–frequency points where it dominates the other sources by  $z$  decibels. Actually,  $r_k(z)$  is named as the preserved-signal ratio in [10]. That is, the approximate WDO  $r_k(z)$  also measures the signal overlap at each time–frequency point. A larger approximate WDO  $r_k(z)$  means more sparseness, and vice versa.

Figure 10.6 is the WDO for different frame sizes  $T$  for  $N = 3, 4$ , and  $5$  for an anechoic case. The sampling frequency was 8 kHz in the investigation. We found that a frame size  $T$  of 512 or 1024 (64 or 128 ms) provides us with the sparsest representation. This result is the same as that reported in previous research [37].

### 10.3.2 Sparseness of Reverberant Speech

It is difficult for sparseness to hold when there is reverberation. In this section, sparseness is investigated for three reverberation conditions:  $RT_{60} = 0, 128$ , and  $300$  ms and three different distances  $R$  between the sensors and sources (see Fig. 10.7). The room impulse responses  $h_{jk}$  were measured in the room depicted in Fig. 10.7 and convolved with real speech signals.

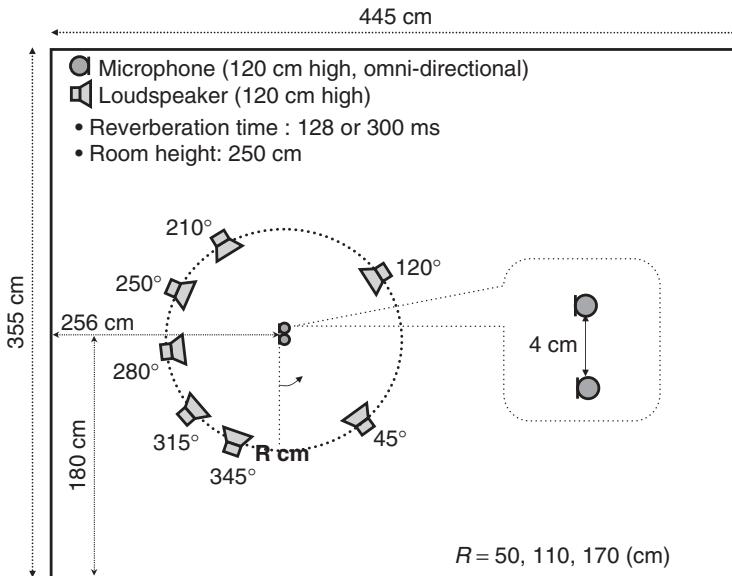
Figure 10.8 is the approximate WDO under certain reverberant conditions. The sources were set at  $45^\circ, 120^\circ$ , and  $315^\circ$  when  $N = 3$ ,  $45^\circ, 120^\circ, 210^\circ$ , and  $315^\circ$  when  $N = 4$ , and  $45^\circ, 120^\circ, 210^\circ, 280^\circ$ , and  $345^\circ$  when  $N = 5$ . We used eight combinations of four male and four female speeches as the sources for each  $N$  condition. As seen in Figure 10.8, the sparseness decreases with increases in both reverberation and distance  $R$ . That is, for reverberant and distant signals, it becomes hard for sparseness to hold.

## 10.4 BINARY MASK APPROACH TO UNDERDETERMINED BSS

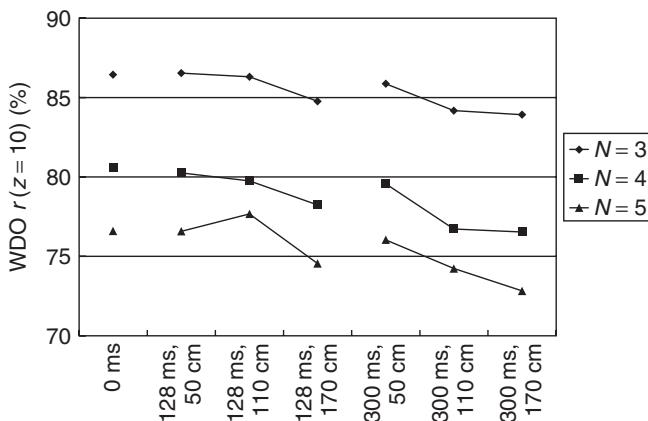
After providing a detailed explanation of the problem of underdetermined BSS and the sparseness of the speech sources, this section offers an overview of existing strategies for solving it. This section describes the procedures used with the binary mask-based method. Figure 10.9 shows the flow of the binary mask approach.

### 10.4.1 Step 1: Signal Transformation to Time–Frequency Domain

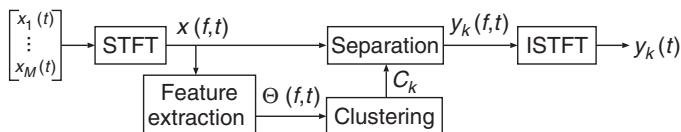
The binary mask approach usually employs a time–frequency domain representation. First, time-domain signals  $x_j(t)$  (10.1) sampled at frequency  $f_s$  are converted into



**Figure 10.7** Room setup ( $M = 2$ ).



**Figure 10.8** Approximate WDO for some reverberant conditions. The  $x$  axis, e.g., 128 ms, 50 cm, indicates that the reverberation time was 128 ms, and the distance between the sources and sensors was 50 cm.  $z = 10$ .



**Figure 10.9** Basic scheme of binary mask approach.

frequency-domain time-series signals  $x_j(f, t)$  with a  $T$ -point short-time Fourier transform (STFT):

$$x_j(f, t) \leftarrow \sum_{r=-T/2}^{T/2-1} x_j(r + tS) \text{win}(r) e^{-j2\pi fr}, \quad (10.14)$$

where  $f \in \{0, \frac{1}{T}f_s, \dots, (T - 1/T)f_s\}$  is a frequency,  $\text{win}(r)$  is a window that tapers smoothly to zero at each end,  $t$  is a new index representing time, and  $S$  is the window shift size. We utilized a Hanning window  $\frac{1}{2}(1 - \cos(2\pi r/T))$  ( $r = 0, \dots, T - 1$ ) as the window  $\text{win}(r)$ .

There are two advantages to working in the time–frequency domain. First, convolutive mixtures (10.1) can be approximated as instantaneous mixtures at each frequency:

$$x_j(f, t) \approx \sum_{k=1}^N h_{jk}(f) s_k(f, t), \quad (10.15)$$

or in a vector notation,

$$\mathbf{x}(f, t) \approx \sum_{k=1}^N \mathbf{h}_k(f) s_k(f, t), \quad (10.16)$$

where  $h_{jk}(f)$  is the frequency response from source  $k$  to sensor  $j$ , and  $s_k(f, t)$  is a frequency-domain time-series signal of  $s_k(t)$  obtained by the same operation as (10.14),  $\mathbf{x} = [x_1, \dots, x_M]^T$ , and  $\mathbf{h}_k = [h_{1k}, \dots, h_{Mk}]^T$  is a mixing vector that consists of the frequency responses from source  $s_k$  to all sensors.

As we have assessed the sparseness of speech signals in the previous section, the second advantage is that the sparseness of a source signal becomes prominent in the time–frequency domain [10, 37], if the source is colored and nonstationary such as speech. The possibility of  $s_k(f, t)$  being close to zero is much higher than that of  $s_k(t)$ . When the signals are sufficiently sparse in the time–frequency domain, we can assume that the sources rarely overlap and, (10.15) and (10.16), respectively, can be approximated as

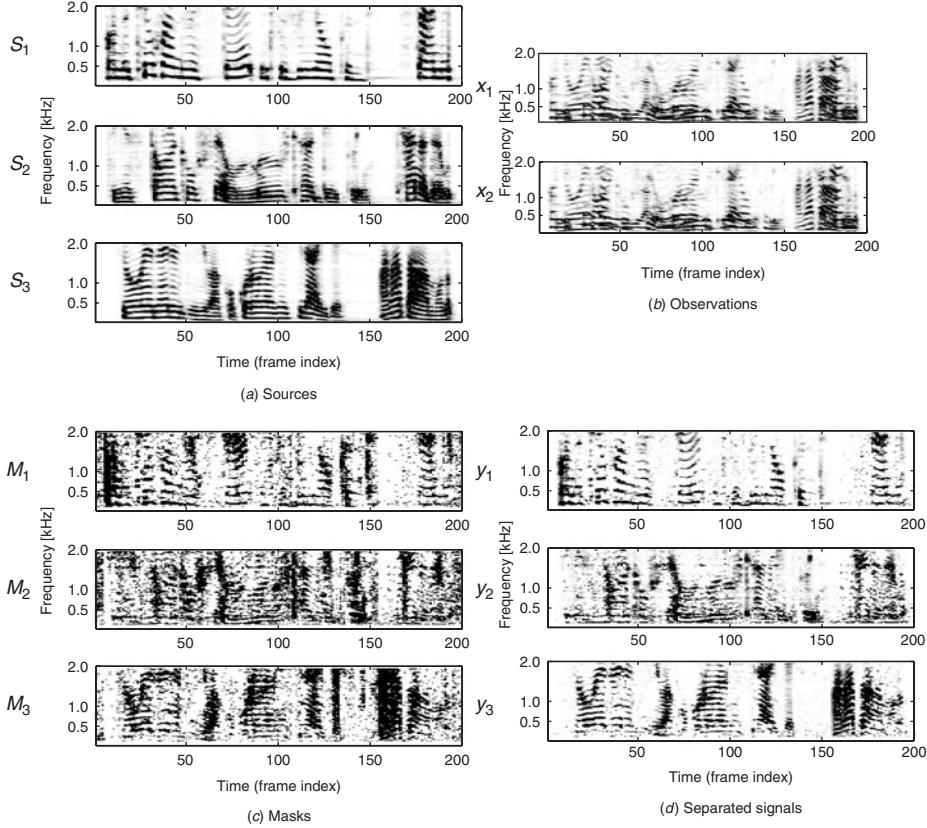
$$x_j(f, t) \approx h_{jk}(f) s_k(f, t), \quad \exists k \in \{1, \dots, N\}, \quad (10.17)$$

$$\mathbf{x}(f, t) \approx \mathbf{h}_k(f) s_k(f, t), \quad \exists k \in \{1, \dots, N\}, \quad (10.18)$$

where  $s_k(f, t)$  is a dominant source at the time–frequency point  $(f, t)$ . For instance, this is approximately true for speech signals [10, 14]. Figure 10.10a shows example spectra of three speech sources in which we can see their temporal/frequency sparseness.

### 10.4.2 Step 2: Feature Extraction

If the sources  $s_k(f, t)$  are sufficiently sparse, separation can be realized by gathering the time–frequency points  $(f, t)$  where only one source  $s_k(f, t)$  is estimated to be dominant. To estimate such time–frequency points, some features  $\Theta(f, t)$  are calculated by using the frequency-domain observation signals  $\mathbf{x}(f, t)$ .



**Figure 10.10** Example spectra of (a) speech sources, (b) observations, (c) masks, and (d) separated signals ( $N = 3$ ,  $M = 2$ ).

Most existing methods utilize the level ratio and/or phase difference between *two* observations as their features  $\Theta(f, t)$ . The features can be summarized as

$$\Theta(f, t) = \left[ \frac{|x_2(f, t)|}{|x_1(f, t)|}, \frac{1}{2\pi f} \arg \left[ \frac{x_2(f, t)}{x_1(f, t)} \right] \right]^T. \quad (10.19)$$

In the phase difference term, the frequency dependence is normalized by dividing it by  $f$ . Thanks to the frequency normalization, we can handle all time–frequency points simultaneously in the next step. If we do not use such frequency normalization, we have to solve the permutation problem among frequencies after the clustering step [40, 41].

Such features (10.19) represent geometric information on sources and sensors if the sources are sufficiently sparse. Let us assume that the mixing process is expressed solely by the attenuation  $\lambda_{jk} \geq 0$  and time delay  $\tau_{jk}$  from source  $k$  to sensor  $j$ :

$$h_{jk}(f) \approx \lambda_{jk} \exp [-j2\pi f \tau_{jk}]. \quad (10.20)$$

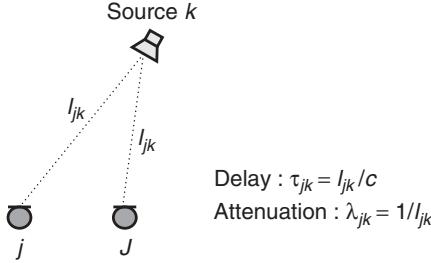


Figure 10.11 Mixing process model.

If there is no reverberation (i.e., an anechoic situation),  $\lambda_{jk}$  and  $\tau_{jk}$  are determined solely by the geometric condition of the sources and sensors (Fig. 10.11). That is, the binary mask approach also assumes an *anechoic* situation. We check the relevance of the anechoic model (10.20) in Section 10.6.3.

If the sources are sparse (10.17), the feature vector (10.19) becomes

$$\boldsymbol{\Theta}(f, t) = \begin{bmatrix} \frac{\lambda_{2k}}{\lambda_{1k}}, -(\tau_{2k} - \tau_{1k}) \end{bmatrix}^T, \quad \exists k. \quad (10.21)$$

That is, the features  $\boldsymbol{\Theta}(f, t)$  contain spatial information on the dominant source  $s_k$  at each time–frequency point  $(f, t)$ ; the level ratio corresponds to the distance, and the phase difference corresponds to the angle.

Figure 10.12 shows an example histogram of the features (10.19) for all time–frequency points of two 5-s speech signals in a weak reverberant condition. In Figure 10.12, the features are well localized, and we can see two clusters, which correspond to each source. Therefore, we can separate signals by picking out the time–frequency points in each cluster. This is the basic idea of the binary mask approach.

**10.4.2.1 Feature Vectors for  $k$ -Means Clustering** In this section, we show that we need appropriate normalization for the feature components (10.19) in order to utilize the  $k$ -means algorithm. This is because the  $k$ -means algorithm assumes that each cluster has a multivariate isotropic variance (see Section 10.4, Step 3).

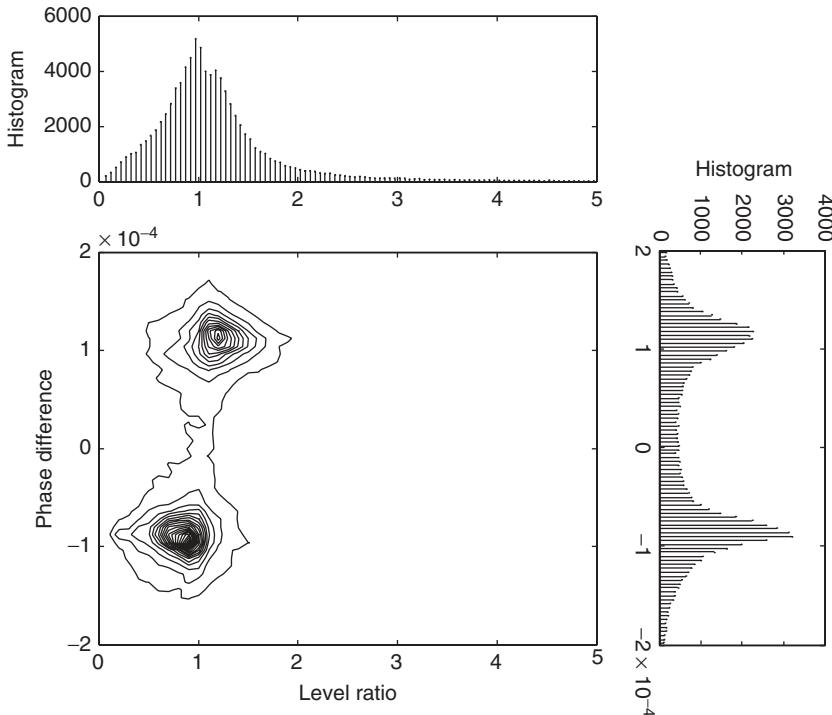
However, the features (10.19) illustrated by Figure 10.12 do not form clusters of isotropic variance, and, therefore, the phase difference is far smaller than the level ratio. Such features cannot be clustered well by the  $k$ -means algorithm.

With the following features (10.22),

$$\left[ \frac{|x_1(f, t)|}{A(f, t)}, \frac{|x_2(f, t)|}{A(f, t)}, \frac{1}{2\pi f c^{-1} d} \arg \left[ \frac{x_2(f, t)}{x_1(f, t)} \right] \right]^T, \quad (10.22)$$

where  $A(f, t) = ||\mathbf{x}(f, t)|| = \sqrt{\sum_{j=1}^M |x_j(f, t)|^2}$ , and the phase is divided by  $2\pi f c^{-1} d$ , the phase difference becomes larger, and the centroids are estimated precisely.

When we normalize the level ratios as seen in the features (10.22), they become  $\leq 1$  and prevent outliers from occurring. Therefore, the features (10.22) provide better performance than (10.19). Moreover, we found that features (10.22), where both the level

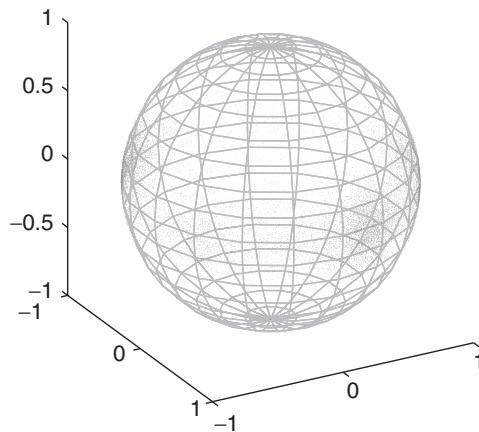


**Figure 10.12** Example histogram of (10.19) when  $N = 2$  and  $M = 2$ . Top: histogram of the level ratio; bottom left: the contour plot of the 2D histogram; bottom right: histogram of the phase difference. Sources were set at  $45^\circ$  and  $120^\circ$ , the reverberation time  $RT_{60} = 128$  ms and the distance between sources and sensors was  $R = 50$  cm (see Fig. 10.7). Outliers are deleted for better visualization.

ratios and phase difference are normalized appropriately, achieve good performance with the  $k$ -means algorithm.

**10.4.2.2 Feature Vectors for Multiple Sensors** In the previous section, we showed that clustering is successfully and effectively executed when we use normalized level ratios and phase differences such as features (10.22). However, thus far, we have discussed the features of a stereo  $M = 2$  system. Several previously proposed algorithms also extract features such as the direction of arrival (DOA) or exploit the amplitude ratio represented by histograms and developed for only  $M = 2$  sensors [10, 14–17]. In both cases, only two sensors can contribute, no matter how many sensors are available. Only a few authors have generalized [40, 41, 56] a method for more than two sensors. A stereo system and a linear sensor array limit the separation ability on a two-dimensional half-plane, for example, they cannot separate sources placed in a mirror image arrangement. To allow the free location of sources, we need more than three sensors arranged two or three dimensionally. This chapter employs more than three sensors arranged two or three dimensionally, which could have a nonlinear/nonuniform alignment (Fig. 10.13).

Therefore, we expand the features to a multiple sensor version based on the result described in the previous section. If we can utilize more than three sensors arranged



**Figure 10.13** Three-dimensional clustering with four sensors.

nonlinearly, we can separate signals located two or three dimensionally. The method can be considered an extension of DUET [10, 36, 42], and so we call it multiple sensor duet: MENUET.

**10.4.2.3 Features in MENUET** The features in MENUET also employ normalized level information and phase differences between multiple observations:

$$\boldsymbol{\Theta}(f, t) = [\boldsymbol{\Theta}^L(f, t), \boldsymbol{\Theta}^P(f, t)]^T, \quad (10.23)$$

where

$$\boldsymbol{\Theta}^L(f, t) = \left[ \frac{|x_1(f, t)|}{A(f, t)}, \dots, \frac{|x_M(f, t)|}{A(f, t)} \right] \quad (10.24)$$

is the observation level information and

$$\boldsymbol{\Theta}^P(f, t) = \left[ \frac{1}{\alpha_1 f} \arg \left[ \frac{x_1(f, t)}{x_J(f, t)} \right], \dots, \frac{1}{\alpha_M f} \arg \left[ \frac{x_M(f, t)}{x_J(f, t)} \right] \right] \quad (10.25)$$

is the phase difference information with respect to the phase of the  $J$ th observation. In the above equations,  $A(f, t) = \sqrt{\sum_{j=1}^M |x_j(f, t)|^2}$ ,  $J$  is the index of one of the sensors, and  $\alpha_j$  ( $j = 1, \dots, M$ ) is a positive weighting constant. By changing  $\alpha_j$ , we can control the weights for the level and phase difference information of the observed signals; a larger  $\alpha_j$  value adds weight to the level and a smaller value emphasizes the phase difference. This is a direct extension of features (10.22).

The normalized level information has the property  $0 \leq \Theta_j^L(f, t) \leq 1$ , where  $\Theta_j^L$  is the  $j$ th component of  $\boldsymbol{\Theta}_L$ . This can prevent the occurrence of outliers. An appropriate value for the phase weight is  $\alpha_j = \alpha = 4\pi c^{-1} d_{\max}$ , where  $c$  is the propagation velocity and  $d_{\max}$  is the maximum distance<sup>1</sup> between sensor  $J$  and sensor  $\forall j \in \{1, \dots, M\}$ .

<sup>1</sup>If we do not have an accurate value for  $d_{\max}$ , we may use a rough positive constant.

**10.4.2.4 Modified Features** We can modify the extended features (10.23) by using the complex representation,

$$\Theta_j(f, t) = \Theta_j^L(f, t) \exp[J\Theta_j^P(f, t)], \quad (10.26)$$

where  $\Theta_j^L$  and  $\Theta_j^P$  are the  $j$ th components of (10.24) and (10.25), respectively. This modification can also be realized by [33, 57]

$$\overline{\Theta}_j(f, t) = |x_j(f, t)| \exp\left[J \frac{\arg[x_j(f, t)/x_J(f, t)]}{\alpha_j f}\right], \quad (10.27)$$

$$\Theta(f, t) \leftarrow \frac{\overline{\Theta}(f, t)}{||\overline{\Theta}(f, t)||}, \quad (10.28)$$

where  $\overline{\Theta}(f, t) = [\overline{\Theta}_1(f, t), \dots, \overline{\Theta}_M(f, t)]^T$ . Features (10.28) are modified features, where the phase difference information is held in the argument term (10.27), and the level information is normalized by the vector norm normalization (10.28). The weight parameter  $\alpha_j$  has the same property as (10.23), however,  $\alpha = 4c^{-1}d_{\max}$  should be the lower limit for successful clustering.

Now the normalized vectors  $\Theta(f, t)$  (10.28) are  $M$ -dimensional complex vectors, and therefore the features will be clustered in an  $M$ -dimensional complex space. The unit-norm normalization (10.28) facilitates the distance calculation in the clustering (10.29), because it projects the vector on a hyper unit sphere. If the features  $\Theta(f, t)$  and the cluster centroid  $\bar{c}_k$  are on the unit sphere, that is,  $||\Theta(f, t)|| = ||\bar{c}_k|| = 1$ , the square distance  $||\Theta(f, t) - \bar{c}_k||^2 = 2(1 - \text{Re}(\bar{c}_k^H \Theta(f, t)))$ . That is the minimization of the distance  $||\Theta(f, t) - \bar{c}_k||^2$  is equivalent to the maximization of the real part of the inner product  $\bar{c}_k^H \Theta(f, t)$ , whose calculation is less computationally complex.

### 10.4.3 Step 3: Clustering

As shown in Figure 10.12, if we can find clusters, each cluster corresponds to an individual source. Therefore, the next step is the clustering of the features  $\Theta(f, t)$ . With an appropriate clustering algorithm, the features  $\Theta(f, t)$  are grouped into  $N$  clusters  $C_1, \dots, C_N$ , where  $N$  is the number of possible sources.

**10.4.3.1 *k*-Means Clustering Algorithm** Features  $\Theta(f, t)$  can be clustered by many methods. For example, the authors [10, 36] first clustered them manually, [38] used kernel density estimation, and [42] applied a maximum-likelihood (ML) based gradient method. Gaussian mixture model (GMM) fitting can also be employed [57] if the number of Gaussians and the initial values for the mean and variance are set appropriately.

In contrast, this chapter employs the *k*-means clustering algorithm [58], which can both automate and accelerate the clustering. The *k*-means clustering algorithm is a well-known and very efficient clustering method. The implementation is easy: for example, MATLAB has the function `kmeans`. The clustering criterion of *k*-means is to minimize the total sum  $\mathcal{J}$  of the squared distances between cluster members and their centroids  $\bar{c}_k$ :

$$\mathcal{J} = \sum_{k=1}^M \mathcal{J}_k, \quad \mathcal{J}_k = \sum_{\Theta(f, t) \in C_k} ||\Theta(f, t) - \bar{c}_k||^2. \quad (10.29)$$

After setting appropriate initial centroids  $\bar{\mathbf{c}}_k$  ( $k = 1, \dots, N$ ),  $\mathcal{J}$  can be minimized by the following iterative updates:

$$C_k = \{\boldsymbol{\Theta}(f, t) \mid k = \operatorname{argmin}_k \|\boldsymbol{\Theta}(f, t) - \bar{\mathbf{c}}_k\|^2\}, \quad (10.30)$$

$$\bar{\mathbf{c}}_k \leftarrow E[\boldsymbol{\Theta}(f, t)]_{\boldsymbol{\Theta} \in C_k} \quad (10.31)$$

where  $E[\cdot]_{\boldsymbol{\Theta} \in C_k}$  is a mean operator for the members of a cluster  $C_k$ . That is, the  $k$ -means clustering algorithm calculates (10.30) and (10.31) until the algorithm converges. The members of each cluster are determined by (10.30). If the features  $\boldsymbol{\Theta}(f, t)$  are properly chosen, then the  $k$ -means works well and each cluster corresponds to an individual source.

Here, it should be noted that  $k$ -means clustering utilizes the squared distance  $\|\boldsymbol{\Theta}(f, t) - \bar{\mathbf{c}}_k\|^2$ , not the Mahalanobis distance  $(\boldsymbol{\Theta}(f, t) - \bar{\mathbf{c}}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\Theta}(f, t) - \bar{\mathbf{c}}_k)$ , where  $\boldsymbol{\Sigma}_k$  is the covariance matrix of the cluster  $k$ . That is, the  $k$ -means algorithm assumes clusters of a multivariate isotropic variance  $\boldsymbol{\Sigma}_k = \mathbf{I}$  for all  $k$ , where  $\mathbf{I}$  denotes an identity matrix.

#### 10.4.4 Step 4: Separation

Next, the separated signals  $y_k(f, t)$  are estimated based on the clustering result. We design a time–frequency domain binary mask that extracts the time–frequency points of each cluster:

$$M_k(f, t) = \begin{cases} 1 & \boldsymbol{\Theta}(f, t) \in C_k \\ 0 & \text{otherwise.} \end{cases} \quad (10.32)$$

Example binary mask spectra are shown in Figure 10.10c. Then, applying the binary masks (Fig. 10.10c) to one of the observations (Fig. 10.10b)  $x_J(f, t)$ , we obtain separated signals (Fig. 10.10d):

$$y_k(f, t) = M_k(f, t)x_J(f, t),$$

where  $J$  is a selected sensor index.

Note that the permutation problem is automatically solved with the spatial information of the sources. We can use the source signal information to solve the permutation problem and achieve better separation performance in the reverberant situation [59]. Note also that we do not have the scaling problem here because we simply “picked out” the target source with the binary mask.

#### 10.4.5 Step 5: Separated Signal Reconstruction

At the end of the flow (Fig. 10.9), we obtain outputs  $y_k(t)$  by employing an inverse STFT (ISTFT) and the overlap-add method [60]:

$$y_k(t) = \frac{1}{A} \sum_{l=0}^{S-1} y_k^{m+l}(t), \quad (10.33)$$

where  $A = \frac{1}{2}(T/S)$  is a constant for the Hanning window case,

$$y_k^m(t) = \begin{cases} \sum_{f \in \{0, \frac{1}{T}f_s, \dots, \frac{T-1}{T}f_s\}} y(f, m) e^{j2\pi f r} & (mS \leq t \leq mS + T - 1) \\ 0 & (\text{otherwise}), \end{cases}$$

and  $r = t - mS$ .

## 10.5 MAP-BASED TWO-STAGE APPROACH TO UNDERDETERMINED BSS

In this section, we present another solution for underdetermined BSS of convulsive speech mixtures based on two stages, namely system identification and source separation. They are denoted by blind system identification (BSI) [22, 61–63] and blind source recovery (BSR) [18, 64, 65], respectively. The approach is based on the MAP principle in connection with a Laplacian source model [66, 67]. They are based on sparseness and Bayesian inference.

In the first stage, we estimate the mixing matrix by employing hierarchical clustering. Based on the estimated mixing matrix, the source signals are estimated in the second stage. The solution for the second stage utilizes the common assumption of independent Laplacian sources, which leads to  $l_1$ -norm minimization.

Both stages are performed in the time–frequency domain to reduce the convulsive mixtures to instantaneous mixtures and increase sparseness. The  $l_1$ -norm minimization has to deal with complex numbers. We employ second-order cone programming (SOCP) and an  $N - M$  source removal approach suitable for dealing with complex numbers. The results of the latter approach are comparable to or even better than the SOCP solution. The advantage is a lower computational cost for problems with low numbers of sources and sensors.

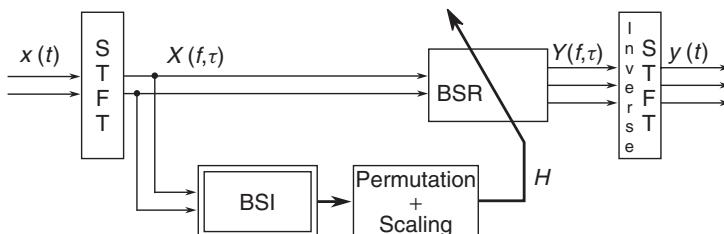
Independent component analysis can be used after the problem has been reduced to (over)determined BSS by, for example,  $N - M$  source removal with a sparseness-based approach [16].

### 10.5.1 Separation Procedures

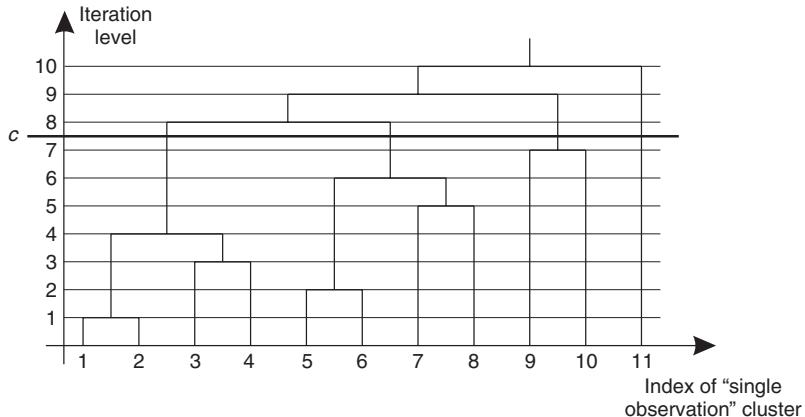
Distinguishing between the two stages of BSI and BSR leads to the common two-stage structure of the separation system in Figure 10.14 for  $N = 3$  source signals and  $M = 2$  sensors where the mixing matrix  $H(f)$  estimated by BSI is used for BSR.

### 10.5.2 Step 1: Blind System Identification – Hierarchical Clustering

In BSI as the first stage, we employ hierarchical clustering to estimate the mixing matrix. This method operates directly on the complex-valued samples. This method does not limit the number of usable sensors, and it prevents convergence problems. Among the most important advantages of the hierarchical clustering algorithm is the fact that it operates directly on the sample data in any vector space of arbitrary dimensions. Therefore, it can easily be applied to complex-valued mixture samples that occur in time–frequency domain convulsive BSS. Furthermore, it does not limit the usable



**Figure 10.14** Basic scheme of MAP-based two-stage approach.



**Figure 10.15** Linking closest clusters.

number of sensors unlike histogram-based methods [10, 14–16], which utilize features such as DOA or exploit the amplitude ratio between two sensors.

Hierarchical clustering is an unsupervised clustering technique that does not depend on initial conditions. It relies solely on a distance measure that corresponds to the dissimilarities between disjoint observation groups [68]. With hierarchical clustering, we use the bottom-up strategy. With the bottom-up strategy, the starting point is single observation samples, which are considered to be clusters containing only one object. Clusters are then combined, so that the number of clusters decreases while the average number of observations per cluster increases.

The combination of clusters into new clusters with the bottom-up strategy is an iterative process and based on the distance between the current clusters. Starting from the “single observation” clusters, the distance between each pair of clusters is calculated, resulting in a distance matrix. At each level of the iteration, the two clusters that are the closest together are combined and form a new cluster (Fig. 10.15). This process is called linking and is repeated until the number of clusters has decreased to a predetermined value  $c$ ,  $N \leq c \leq F$ , where  $F$  denotes the total number of samples.

As mentioned above, most of the samples will cluster around the mixing vectors  $\mathbf{h}_k$ , depending on the degree of disjoint sparseness of the original signals. Special attention must be paid to the remaining samples (outliers), which are randomly scattered in the spaces between the mixing vectors due to the nonideal disjoint sparseness of the sources, reverberation, ambient noise level, and the distance from the sources to the sensors. Usually they are far away from other samples and will be combined with other clusters only at higher levels of the clustering process (i.e., when only a few clusters are left). The value  $c$  is, in general, related to the sparseness of the sources and to the acoustic environment. A smaller  $c$  value can be chosen as the signals become sparser or the acoustic environment less reverberant because of the smaller number of outliers that must be avoided. Experiments have shown that the choice for parameter  $c$  is highly insensitive as long as it is above a certain limit that would combine desired clusters. This suggests that the final cluster number  $c$  should be high at

$$c \gg N. \quad (10.34)$$

By doing so, we prevent these outliers from being linked with the clusters around the mixing vectors  $\mathbf{h}_k$ , which would usually lead to distortions of the estimated mixing vector. This results in more robustness and has a similar effect to garbage models. More important, however, is the fact that we avoid combining desired clusters. Since the outliers are often far away from other clusters, desired clusters may be closer to each other than to outliers. Experiments showed that the exact value of  $c$  does not matter as long as it is above 60 for  $N \in \{3, 4, 5\}$  for the conditions given in Section 10.6.4.

Assuming that the clusters around the mixing vectors  $\mathbf{h}_k$  include the largest numbers of samples, we finally choose  $N$  clusters with the largest numbers of samples as those representing  $N$ . Thereby the number of sources  $N$  must be known. To obtain the mixing vectors, we average all the samples of each cluster

$$\mathbf{h}_k = \frac{1}{|C_k|} \sum_{x \in C_k} x, \quad 1 \leq i \leq N, \quad (10.35)$$

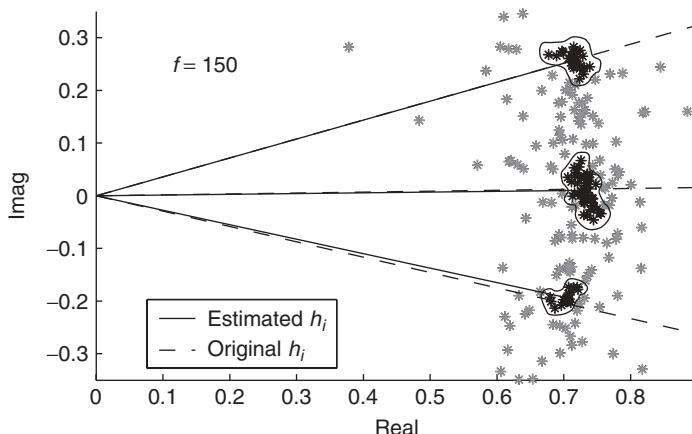
where  $|C_k|$  denotes the cardinality of cluster  $C_k$ . Thus we assume that the contribution of other sources has a zero mean.

An example of the resulting clusters is shown in Figure 10.16. Here we chose  $c = 100$  as in the experiments described in Section 10.6.4. Further experimental details are given in Section 10.6.4. No initial values are required for the mixing vectors  $\mathbf{h}_k$ . This means, in particular, that if the assumption of clusters with many samples around the mixing vectors is true, then the algorithm converges to those clusters.

While the considered signals must have some degree of sparseness in the time–frequency plane, to ensure the assumption that the influence of the other signals can be disregarded for each cluster, they do not have to be statistically independent at this point to obtain useful clusters.

### 10.5.3 Step 2: Blind Source Recovery

In BSR, which is the second stage, we separate the mixtures using the estimated mixing matrix from the first stage. We assume statistical independence for the sources [69]. Utilizing a Laplacian source model leads to a constrained  $l_1$ -norm minimization.



**Figure 10.16** Estimation of mixing vectors.

By assuming sparse sources, each sample in the time–frequency plane can be assigned to few sources or ideally to only one source. Therefore, the problem is in fact reduced to (over)determined BSS. However, the separation filters are highly time variant and therefore difficult to estimate. The assignment of each sample to its source can be supported by the previously identified mixing matrix [70].

The BSR approaches are based on the different principles from BSI. The most general approaches for estimating the unknown source signals are based on Bayesian inference [71, 72]. MAP approaches are widely used special cases of Bayesian interference. They yield the posterior distribution of the desired parameters (here: source signals) by accounting for their prior distribution and the likelihood of the observed data (here: mixed signals). The generality of Bayesian inference stems from the fact that the prior distribution can contain all available knowledge about the desired parameters. As such, sparseness is often encoded by appropriate probability density functions (PDFs).

Sparseness can also be modeled by appropriate PDFs such as the Laplacian PDF where values close to zero have a high probability density and values away from zero have a low probability density. Therefore, sparseness can be dealt with within the framework of Bayesian inference [52]. The disadvantage of Bayesian inference-based approaches is that they can easily lead to analytically nontractable equations requiring computationally expensive numerical methods.

We outline the way in which this principle can lead to constrained  $l_1$ -norm minimization. If we assume an STFT that represents the mixture signals, we have to recover the sources from  $M$ -dimensional complex-valued observation vectors for each DFT bin.

**10.5.3.1 Sparseness-Based Source Model** According to the MAP principle and Bayes' rule [73], we obtain an estimation  $\mathbf{y}$  of the source signals  $\mathbf{s}$  by

$$\mathbf{y} = \arg \max_{\mathbf{s}} P(\mathbf{s}|\mathbf{x}, \mathbf{H}) = \arg \max_{\mathbf{s}} P(\mathbf{x}|\mathbf{s}, \mathbf{H})P(\mathbf{s}), \quad (10.36)$$

once the mixing matrix  $\mathbf{H}$  is known.  $P(\mathbf{s}|\mathbf{x}, \mathbf{H})$  denotes the conditional a posteriori PDF of the source signals  $\mathbf{s}$  given the mixed signals  $\mathbf{x}$  and the mixing matrix  $\mathbf{H}$ .  $P(\mathbf{x}|\mathbf{s}, \mathbf{H})$  denotes the likelihood, which for the noiseless mixture model (10.6) is given by a Dirac impulse  $\delta(\mathbf{x} - \mathbf{H}\mathbf{s})$ . It requires the maximum of the conditional a posteriori PDF to fulfill  $\mathbf{x} = \mathbf{H}\mathbf{s}$ , which essentially turns (10.36) into the constrained problem

$$\max_{\mathbf{s}} P(\mathbf{s}) \quad \text{s.t.} \quad \mathbf{x} = \mathbf{H}\mathbf{s}. \quad (10.37)$$

where s.t. is such that. We further assume mutually independent source signals  $s_k$  whose spectral components have Laplacian distributions:

$$P(\mathbf{s}) = \prod_{k=1}^N P(s_k) \propto \prod_{k=1}^N \exp(-|s_k|), \quad (10.38)$$

whereby  $|s_k|$  denotes the amplitude of  $s_k$ . Since  $\arg \max_{\mathbf{s}} P(\mathbf{s}) = \arg \max_{\mathbf{s}} \log(P(\mathbf{s}))$ , (10.36) leads eventually to the minimization problem

$$\min_{\mathbf{s}} \sum_{k=1}^N |s_k| \quad \text{s.t.} \quad \mathbf{x} = \mathbf{H}\mathbf{s}, \quad (10.39)$$

for each time  $t$  and frequency bin  $f$ . In other words, (10.39) describes a constrained  $l_1$ -norm minimization problem.

By contrast, if the source signals  $s$  have a Gaussian distribution, the optimum solution is given by the Moore–Penrose generalized inverse of the mixing matrix  $H$ ,

$$\mathbf{y} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{x}, \quad (10.40)$$

which is well-known to be the constrained  $l_2$ -norm minimization solution.

**10.5.3.2 Constrained  $l_1$ -Norm Minimization** For constrained optimization problems such as (10.39), techniques developed for linear programming (LP) for real-valued  $l_1$ -norm minimization, we can use second-order cone programming (SOCP) for complex-valued  $l_1$ -norm minimization. Once optimization problems are transformed into the standard form, efficient software packages such as SeDuMi [74] can be used to determine the optimal parameters.

**10.5.3.3 Real-Valued  $l_1$ -Norm Minimization** Although powerful algorithms for LP exist, they are still time consuming. Depending on the dimensions of the problem, we can obtain a faster  $N - M$  source removal approach if we use a certain property of the solution. It can be shown [19, 75] that the  $N$ -dimensional vector  $\mathbf{y}$ , which solves (10.39), contains at least  $N - M$  zeros. This is an interesting phenomenon that results from the Laplacian distribution. The MAP estimation gives  $M$  nonzero sources and  $N - M$  zero sources. The mixing matrix can be invertible for these sources. If we can determine which  $N - M$  components are likely to be zero, and then remove the  $N - M$  components and invert the remainder, we can solve the problem. This is done implicitly in the MAP estimation method.

The lower limit for the number of zeros can be considered a constraint on the solution of (10.39) and can easily be fulfilled by setting  $N - M$  elements of the solution to zero. Then we only have to determine the remaining  $M$  elements. Assuming that we know where to place the zeros, the remaining elements are found by multiplying the inverse of the quadratic matrix built by the remaining mixing vectors  $\mathbf{h}_k$  with the constraining vector  $\mathbf{x}$ :

$$[\mathbf{h}_{i_1} \dots \mathbf{h}_{i_M}]^{-1} \mathbf{x}, \quad i_1, \dots, i_M \in \{1, \dots, N\}. \quad (10.41)$$

For example, the correct zero placement can be determined by combinatorially testing all possibilities and accepting the one with the smallest  $l_1$ -norm. As a simple example let us consider

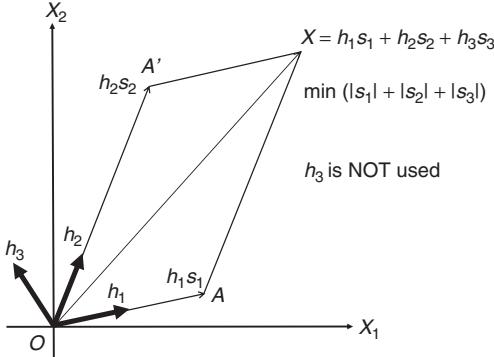
$$\mathbf{H} = \begin{bmatrix} 1 & 0.6 & -0.6 \\ 0 & 0.8 & 0.8 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}. \quad (10.42)$$

According to the dimensions of the problem, at least one element of the solution  $\mathbf{y}$  must be zero. The  $l_1$ -norm of the possible solutions are

$$\left\| \begin{bmatrix} \begin{bmatrix} 1 & 0.6 \\ 0 & 0.8 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} \\ 0 \end{bmatrix} \right\|_1 = 1.25, \quad (10.43)$$

$$\left\| \begin{bmatrix} 0 & \begin{bmatrix} 1 & -0.6 \\ 0 & 0.8 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} \\ 0 \end{bmatrix} \right\|_1 = 2, \quad (10.44)$$

$$\left\| \begin{bmatrix} 0 & \\ \begin{bmatrix} 0.6 & -0.6 \\ 0.8 & 0.8 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} & 0 \end{bmatrix} \right\|_1 = 1.6. \quad (10.45)$$



**Figure 10.17** Shortest path from the origin  $O$  to the data point  $\mathbf{x}$  is  $O\text{-}A\text{-}x$  (or  $O\text{-}A'\text{-}x$ ). Therefore,  $\mathbf{x}$  decomposes as  $O\text{-}A$  along direction  $\mathbf{h}_1$  and as  $O\text{-}A'$  along direction  $\mathbf{h}_2$ .

The notation of (10.44) reflects the above description where one element is set to zero and the remaining quadratic matrix inverted. The chosen solution would be the one corresponding to (10.43).

#### 10.5.3.4 Shortest Path Algorithm and $N - M$ Source Removal Approach

The  $N - M$  source removal approach is further supported geometrically by the shortest path algorithm [19] and  $l_0$ -norm, which basically counts the number of nonzero elements.

Even when the mixing matrix  $\mathbf{H}$  is known, since the system in (10.6) is underdetermined, its solution is not unique. The usual approach to sparse underdetermined BSS consists of finding the solution that minimizes the  $l_1$ -norm, as in (10.39). In this case, the optimal representation of the data point

$$\mathbf{x}(f, t) = \mathbf{h}_1(f)s_1(f, t) + \cdots + \mathbf{h}_N(f)s_N(f, t), \quad (10.46)$$

that minimizes  $\sum_{k=1}^N |s_k|$  is the solution of the corresponding linear programming problem. Geometrically, for a given feasible solution, each source component is a segment of length  $|s_k|$  in the direction of the corresponding mixing vector  $\mathbf{h}_k$  and, by concatenation, their sum defines a path from the origin to  $\mathbf{x}$ . Minimizing  $\sum_{k=1}^N |s_k|$  amounts therefore to finding the *shortest path* to  $\mathbf{x}$  over all feasible solutions. Note that, with the exception of singularities, since mixture space is  $M$ -dimensional,  $M$  (independent) mixing vectors  $\mathbf{h}_k$  will be required for a solution to be feasible (i.e., to reach  $\mathbf{x}$  without error). For the two-dimensional case (see Fig. 10.17), the shortest path is obtained by choosing the mixing vectors  $\mathbf{h}_1$  and  $\mathbf{h}_2$  that enclose  $\mathbf{x}$ , but  $\mathbf{h}_3$  is not used.

Let  $\mathbf{H}_r = [\mathbf{h}_1 \mathbf{h}_2]$  be the reduced square matrix that includes only the selected mixing vectors; let  $\mathbf{W}_r = \mathbf{H}_r^{-1}$ ; and let  $\mathbf{s}_r$  be the decomposition of the target point along  $\mathbf{h}_1$  and  $\mathbf{h}_2$ . The components of the sources are then obtained as

$$\mathbf{s}_r = \mathbf{W}_r \mathbf{x}, \quad s_k = 0 \quad \text{for } k \neq 1, 2. \quad (10.47)$$

#### 10.5.3.5 Complex-Valued $l_1$ -Norm Minimization

If complex numbers are involved, then  $l_1$ -norm minimization problems (10.39) can be transformed into an SOCP problem [76], which can be solved numerically for example with SeDuMi [74].

In contrast to the real-valued  $l_1$ -norm minimization problem where a minimum number of zeros can be guaranteed theoretically in the optimal solution, the number of zeros cannot be predicted with complex-valued problems. Once complex numbers are involved, their imaginary part results in an inherent  $l_2$ -norm, which leads to smooth slopes as they appear with second-order or higher polynomials. There the  $l_1$ -norm is changed from the sum of absolute values of real numbers to the sum of the  $l_2$ -norms of the real and imaginary part. The introduction of the  $l_2$ -norm explains the different behavior of complex-valued  $l_1$ -norm minimization compared with its real counterpart.

#### 10.5.4 Permutation Indeterminacy

The disadvantage of BSS in the time–frequency domain is the internal permutation indeterminacy. It describes the problem whereby the order of the output signals is arbitrary and cannot be determined by the separation process. While this is often a minor problem as regards instantaneous mixtures, it becomes a very serious problem if convolutive mixtures are involved and separation is performed in the time–frequency domain. In this case, the output signals are not automatically aligned across the frequency bins, and several principles, including clustering DOA and correlation, can be used to accomplish this.

In our framework, we use a clustering-based method to mitigate the permutation indeterminacy [77, 78]. To this end, the mixing vectors are normalized such that they become independent of frequency. As in the discussion in the section about hierarchical clustering for estimating the mixing vectors, the mixing vectors that correspond to the same source are assumed to form clusters. These clusters are determined and their members reordered accordingly so that their order in the mixing matrix reflects the cluster to which they belong.

#### 10.5.5 Scaling Indeterminacy

As a consequence of the scaling indeterminacy, each output signal in each frequency bin can be multiplied independently by an arbitrary factor. In order to mitigate the effect of the scaling indeterminacy, we apply a normalization technique that can be derived by the minimal distortion principle (MDP) [79]. By minimizing the distortion induced by the separation system, this principle also prevents the separation system from whitening the signals.

The idea is to modify the separation system so that its output is as close as possible to the input that would be observed if only the source signal corresponding to the considered output was active. Note that the MDP does not dereverberate the mixed signals but only avoids linear distortion by the separation system.

Originally, the MDP was introduced for (over)determined BSS where each output was linked to one specific sensor. As a result, the normalization resulted in a separation system that minimizes the expected value

$$E \left\{ \sqrt{\sum_{k=1}^N (y_k - x_k)^2} \right\} \quad (10.48)$$

after the permutation indeterminacy has been solved. The expectation refers to time frames  $t$ .

We extend it to underdetermined BSS by allowing an arbitrary sensor to serve as a reference resulting in the minimization of

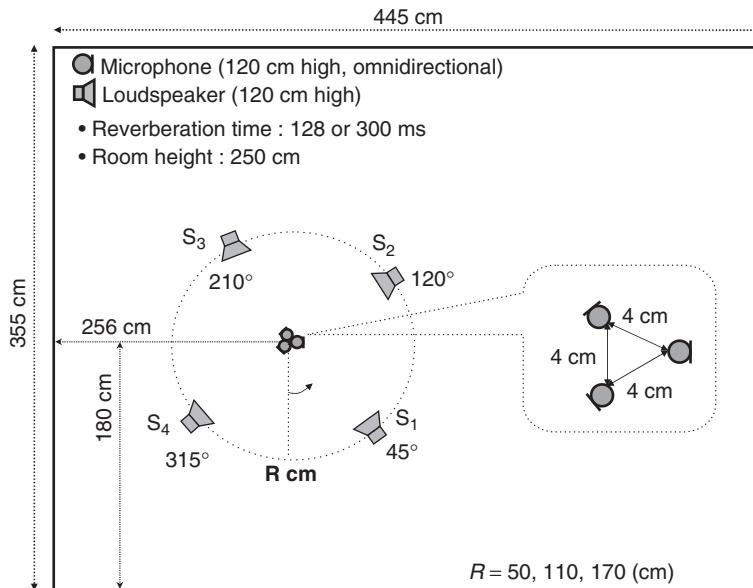
$$E \left\{ \sqrt{\sum_{k=1}^N (y_k - x_{r_k})^2} \right\}, \quad r_k \in \{1, \dots, M\} \quad (10.49)$$

after the permutation indeterminacy has been solved. As a special case only one sensor is necessary as a reference for all output signals, that is,  $r_k = r \in \{1, \dots, M\} \forall k$ . As a consequence, each output signal is multiplied by the component of the estimated mixing matrix that corresponds to the considered output signal and, for example, to the first sensor. As an alternative, an average of components that correspond to several sensors can be used instead of a single sensor. In both cases, the scaling factors can be applied directly to the estimated mixing matrix instead of the output signals.

## 10.6 EXPERIMENTAL COMPARISON WITH BINARY MASK APPROACH AND MAP-BASED TWO-STAGE APPROACH

### 10.6.1 Experimental Conditions with Binary Mask Approach

We performed experiments with measured impulse responses  $h_{jk}(l)$  in a room as shown in Figures 10.18 and 10.19. The room reverberation times  $RT_{60}$  were 128 and 300 ms. We used the same room for both reverberation times but changed the wall condition. We also changed the distance  $R$  between the sensors and sources. The distance variations were  $R = 50, 110$ , and  $170$  cm (see Figs. 10.18 and 10.19). Mixtures were made by convolving the measured impulse responses in the room and 5-s English speeches.



**Figure 10.18** Room setup ( $N = 4, M = 3$ ).

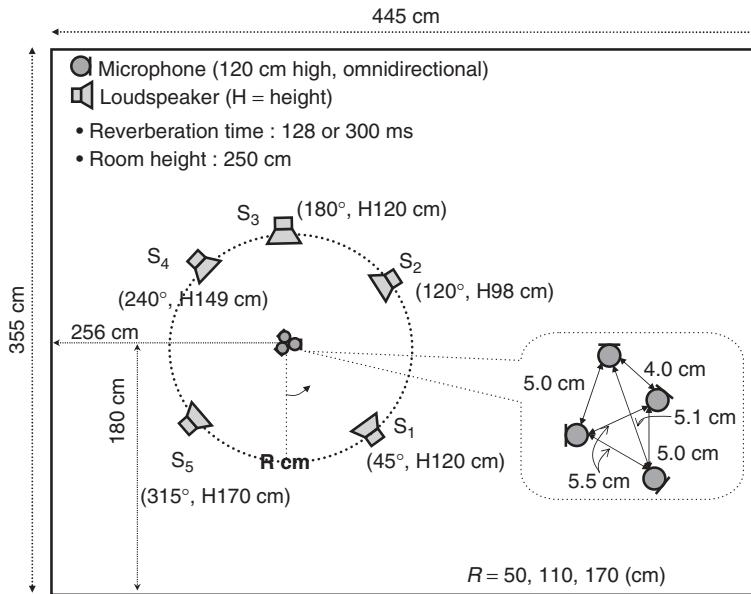


Figure 10.19 Room setup ( $N = 5, M = 4$ ).

For the anechoic test, we simulated the mixture by using the anechoic model (10.20) and the mixture model (10.1). The sampling rate was 8 kHz. The STFT frame size  $T$  was 512 and the window shift  $S = T/4$ .

Unless otherwise noted, we utilized modified features (10.28) with  $\alpha_j = \alpha = 4c^{-1}d_{\max}$  for the features because the computational cost of distance calculation is low (see Section 10.4.2). We utilized the  $k$ -means algorithm for the clustering, where the number of sources  $N$  was given. We set the initial centroids of the  $k$ -means using a far-field model. The initialization method is described in Section 10.7.1.

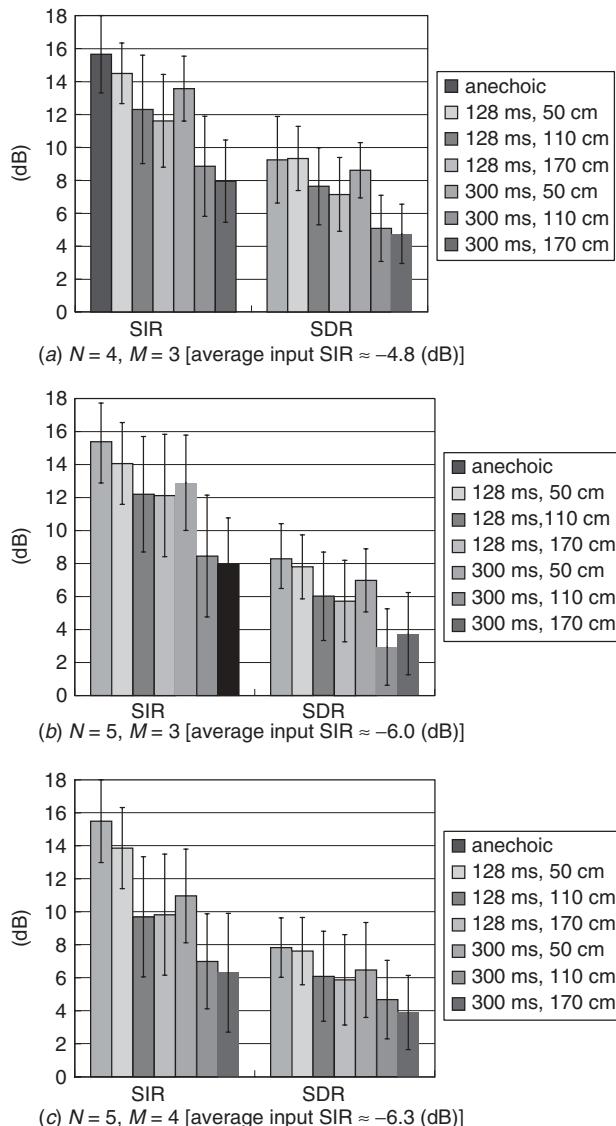
The separation performance was evaluated in terms of the signal-to-interference ratio (SIR) improvement and the signal-to-distortion ratio (SDR). Their definitions are given in Section 10.7.2.

## 10.6.2 Separation Results with Binary Mask Approach

**10.6.2.1 With Two Sensors** First, we experimented using two sensors under the condition described in Section 10.4.2. The features (10.22) achieved an SIR improvement of 12.4 dB and the modified features (10.28) achieved an SIR improvement of 12.2 dB. A comparison with the MAP approach can be found in [40]. A comparison with GMM fitting is provided in [57]. Note that two sensors/linear arrays do not work when the sources are placed at axisymmetrical locations with respect to the microphone array because they have the equivalent features in (10.19).

**10.6.2.2 With Three 2-D Sensors** Here we show the separation results obtained with three sensors arranged two-dimensionally (Fig. 10.18). Note that sources were also distributed two-dimensionally.

Figure 10.20a shows the separation result when  $N = 4$  and  $M = 3$ . We can see that the binary mask approach achieved good separation performance with the nonlinear



**Figure 10.20** Average SIR improvement and SDR for each condition. Error bar shows the standard deviations for all outputs and combinations.

sensor arrangement. We also evaluated the performance for  $N = 5$  and  $M = 3$ , where the source positions were  $45^\circ$ ,  $120^\circ$ ,  $210^\circ$ ,  $280^\circ$ , and  $345^\circ$  and obtained good performance (Fig. 10.20b).

**10.6.2.3 With Four Sensors** The MENUET method can also be applied to a three-dimensional sensor array arranged nonuniformly. The setup is shown in Figure 10.19. Here, the system knew only the maximum distance  $d_{\max}$  (5.5 cm) between the reference microphone and the others. To avoid the spatial aliasing problem, we utilized frequency bins of up to 3100 Hz in this setup. Figure 10.20c

shows the separation result when  $N = 5$  and  $M = 4$ . Figure 10.20c shows that MENUET can be applied to such three-dimensional microphone array systems.

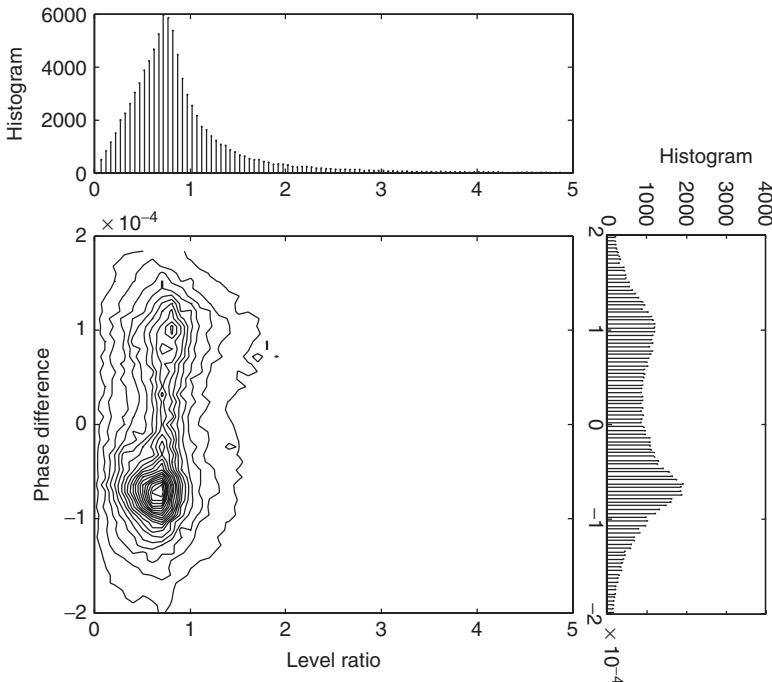
### 10.6.3 Validity of Sparseness Assumption and Anechoic Assumption

In this section, we assess the relevance of two assumptions, related to the effects of reverberation and distance, using real speech signals and measured room impulse responses. The binary mask approach utilizes two assumptions:

1. Sparseness assumption (10.17):  $x_j(f, t) \approx h_{jk}(f)s_k(f, t)$ ,  $\exists k \in \{1, \dots, N\}$ .
2. Anechoic assumption (10.20):  $h_{jk}(f) \approx \lambda_{jk} \exp[-j2\pi f\tau_{jk}]$ .

In the experimental results, Figure 10.20 also showed the performance tendency under reverberant conditions. The performance degrades as the reverberation time  $RT_{60}$  increases. Moreover, performance degradation was observed as the distance  $R$  became large. This is because, under long reverberation and/or large  $R$  conditions, it becomes difficult for the source sparseness (10.17) and anechoic assumptions (10.20) to hold as discussed in Sections 10.3.2 and 10.6.3.

Figure 10.21 shows an example histogram of features (10.19) for two 5-s speech signals under a stronger reverberant condition than in Figure 10.12. In Figure 10.21 the reverberation time was  $RT_{60} = 300$  ms and the distance  $R$  between sensors and



**Figure 10.21** Example histogram of (10.19) when  $N = 2$  and  $M = 2$ . Top: histogram of the level ratio; bottom left: contour plot of the 2D histogram; bottom right: histogram of the phase difference. Sources were set at  $45^\circ$  and  $120^\circ$ ,  $RT_{60} = 300$  ms,  $R = 110$  cm (see Fig. 10.7). Outliers are deleted for better visualization.

**TABLE 10.1** Averaged Standard Deviations of Level Ratio and Phase Difference in (10.21) for Each Cluster

Standard Deviation	RT <sub>60</sub> 128 ms			RT <sub>60</sub> 300 ms		
	R = 50 cm	R = 110 cm	R = 170 cm	R = 50 cm	R = 110 cm	R = 170 cm
$\sigma_L$	0.60	1.05	2.06	0.87	1.68	2.79
$\sigma_P$ (ms)	0.07	0.09	0.10	0.08	0.11	0.12

sources was 110 cm, whereas in Figure 10.12 RT<sub>60</sub> = 128 ms and R = 50 cm. The other conditions are the same as those in Figure 10.12. Compared with Figure 10.12, each histogram in Figure 10.21 is broadened owing to the reverberation.

Table 10.1 shows the standard deviations  $\sigma_L$  and  $\sigma_P$  of the level ratio and the phase difference in (10.19), respectively. The table shows average values for two clusters and eight speaker combinations. Table 10.1 shows that the variance of the clusters becomes large as reverberation and distance increase. This means that it becomes difficult for the sparseness assumption (10.17) and the anechoic assumption (10.20) to hold. Therefore, clustering and separation become difficult in reverberant situations and the binary mask approach has to handle such cases in a real separation.

It is also important to mention nonlinear distortion in separated signals. There is nonlinear distortion (musical noise) in the outputs with our method, due to the winner-take-all property of the binary mask. The results of subjective tests with 10 listeners can be found in [33]. Some sound examples can be found on the Internet ([http://www.kecl.ntt.co.jp/icl/signal/araki/xcluster\\_fine.html](http://www.kecl.ntt.co.jp/icl/signal/araki/xcluster_fine.html)).

In this chapter, we described the sparse source separation procedure. In addition, using centroids estimated by clustering, we can also achieve the source localization of sparse sources. This is because each centroid is the expectation value of the geometric information (10.21) of each source. As sparse source localization requires the sensor locations, it is not a blind process. The procedures for sparse source localization are detailed in [80]. As the method is based on MENUET, it allows us to employ an arbitrary sensor arrangement, and estimate the directions of sparse sources distributed three dimensionally.

#### 10.6.4 Experimental Conditions with MAP-Based Two-Stage Approach

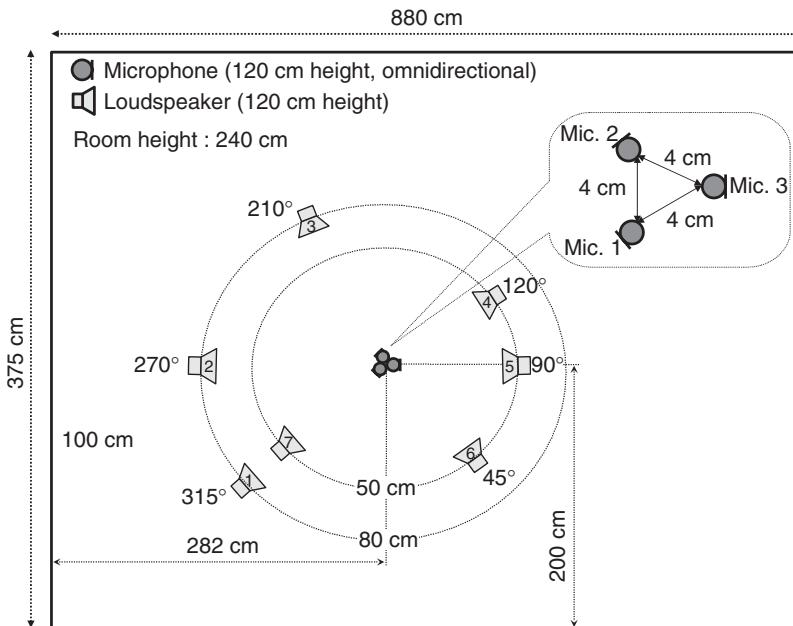
In our experiments, we separated mixtures that we obtained from clean speech signals and recorded room impulse responses. We tested both approaches with the estimated mixing matrix with different numbers of sources ( $N \in \{3, 4, 5\}$ ) and sensors ( $M \in \{2, 3\}$ ). We performed four experiments for each scenario. Each of the four experiments had a different combination of speakers selected from six male and female English speakers. Further experimental conditions are summarized in Table 10.2 and Figure 10.22. For comparison, we also applied a time–frequency masking approach to the same mixtures [33].

#### 10.6.5 Comparison of Separation Results

The results are shown in Figure 10.23. A subjective evaluation of the separated sources supports the results.

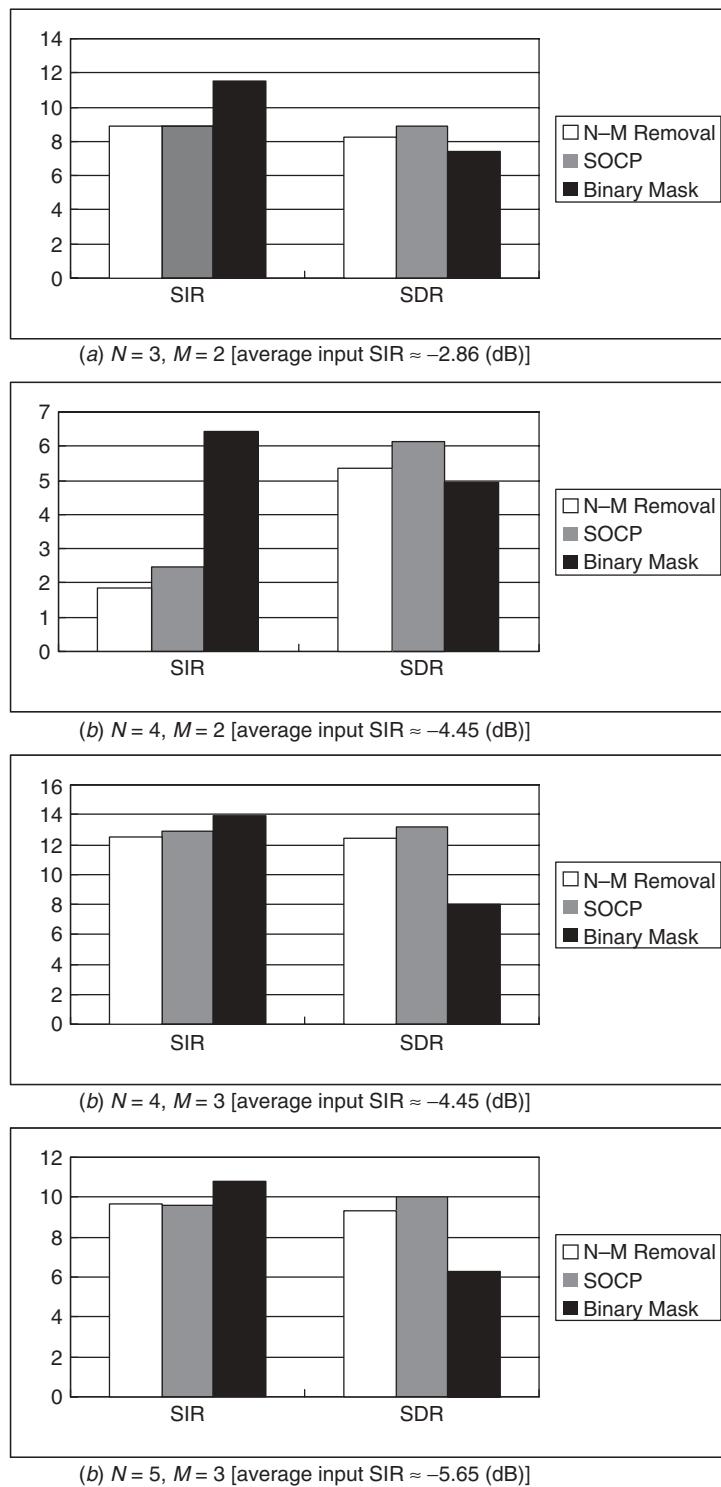
**TABLE 10.2 Experimental Conditions**

Sensor distance	4 cm
Source signal length	7 s
Reverberation time $T_R$	120 ms
Sampling frequency $f_s$	8 kHz
Window type	Hanning
Filter length	1024 points
Shifting interval	256 points
Number of clusters $c$	100

**Figure 10.22** Room setup.

The SOCP and  $N - M$  source removal approach solutions of the MAP-based approach yield similar results with the estimated mixing matrix. Although the difference in performance quality is negligible in practical applications with estimated mixing matrices, the computational complexity reveals great differences. The  $N - M$  source removal approach has a low computational complexity. On the other hand, the SOCP solution has a high computational complexity.

One reason for the big difference in the computational complexity can be found in the reusability of previous results. For underdetermined BSS in the time–frequency domain, the minimum  $l_1$ -norm solution must be calculated several times with the same mixing matrix. The  $N - M$  source removal approach solution is built on the inverses of selected mixing vectors. Once they are calculated, they can be reused as long as the mixing matrix does not change. In contrast, SOCP cannot profit from the reuse of earlier results due to its algorithmic nature.



**Figure 10.23** Average SIR improvement and SDR for each condition.

The time–frequency masking approach yields better separation in terms of the SIR than the MAP-based methods. This is because the time–frequency masking approach uses only time–frequency points that originate from a single source with high confidence. In contrast, the MAP-based methods do not evaluate the confidence regarding the origin of a time–frequency point but use all points for separation in a uniform way. On the other hand, by using all time–frequency points, the MAP-based methods result in fewer artifacts, as expressed by a higher SDR.

## 10.7 CONCLUDING REMARKS

After providing a detailed examination of the sparseness of speech sources, we presented two underdetermined BSS methods for convolutive mixtures. Both methods were performed in the time–frequency domain to reduce the convolutive mixtures to instantaneous mixtures and increase sparseness.

The first approach is the binary mask approach named MENUET. Its features employ normalized level information and phase differences between multiple sensors. The features are clustered by the  $k$ -means clustering algorithm. This method can operate in real time.

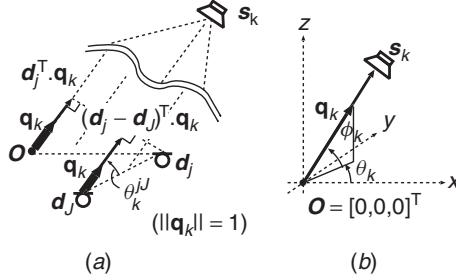
The second approach is the MAP-based approach. The method employs two stages, namely system identification and source separation. We employ the hierarchical clustering algorithm in the first stage. In the second stage, the MAP estimation in connection with a Laplacian source model leads to the constrained  $l_1$ -norm minimization problem.

The MAP estimation gives  $M$  nonzero sources and  $N - M$  zero sources. If we can determine which  $N - M$  components are likely to be zero, and then remove the  $N - M$  components and invert the remainder, we can solve the problem. This is accomplished implicitly in the MAP estimation method. We also examined an *explicit*  $N - M$  source removal approach.

Both methods make it easy to employ multiple sensors arranged in a nonlinear/nonuniform way to separate two- and three-dimensionally distributed speech sources. Experimental results confirmed that the assumption of sparseness in time–frequency and space and, therefore, clusters around the mixing vectors, is sufficiently fulfilled for convolutively mixed speech signals in the time–frequency domain. We obtained promising experimental results in a room with weak reverberation with ( $M \in \{2, 3, 4\}$ ) sensors and ( $N \in \{3, 4, 5\}$ ) sources. We also reported the performance under some reverberant conditions, where the sparseness and anechoic assumptions were deteriorating. The results showed that the direct-to-reverberant ratio is important for sparse signal processing. The underdetermined sparse source separation in reverberant conditions is still an open problem.

### 10.7.1 Initial Values for $k$ -Means Clustering

The  $k$ -means algorithm is sensitive to the initial values of the centroids especially when the number of sources  $N$  is large and the reverberation time is long. Therefore,



**Figure 10.24** (a) Far-field model; (b) definition of source direction.

we designed the initial centroids by using the far-field model, where the frequency response  $h_{jk}(f)$  is given as

$$h_{jk}(f) \approx \exp[-j2\pi f c^{-1} \mathbf{d}_j^\top \mathbf{q}_k],$$

and using the same normalization as each feature. Here,  $c$  is the propagation velocity of the signals, and the three-dimensional vectors  $\mathbf{d}_j$  and  $\mathbf{q}_k$  represent the location of sensor  $j$  and the direction of source  $k$ , respectively (see Fig. 10.24 and [80]).

When designing the initial centroids, the sensor locations  $\mathbf{d}_j$  ( $j = 1, \dots, M$ ) were on almost the same scale in each setup, and the initial directions  $\mathbf{q}_k$  were set so that they were as scattered as possible. Concretely, in the experiments, we utilized the sensor vector  $\mathbf{q}_k = [\cos \theta_k \cos \phi_k, \sin \theta_k \cos \phi_k, \sin \phi_k]^\top$  (see Fig. 10.24). The azimuth of the  $k$ th source was set at  $\theta_k = 2\pi/N \times k$  ( $k = 1, \dots, N$ ) for  $M \geq 3$ , and  $\theta_k = \pi/N \times k$  ( $k = 1, \dots, N$ ) for  $M = 2$ . The elevation  $\phi_k = 0$  for all sources  $k$ . Note that these initial values of  $\mathbf{d}_j$  and  $\mathbf{q}_k$  were not exactly the same in each setup.

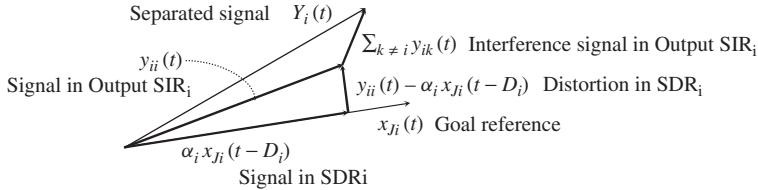
### 10.7.2 Performance Measures

The separation performance is evaluated in terms of signal-to-interference ratio (SIR) improvement and signal-to-distortion ratio (SDR). A larger number represents a better result for both criteria. To calculate these numbers, we need the individual source observations  $x_{Jk}$  defined by  $x_{Jk}(t) = \sum_l h_{Jk}(l)s_k(t-l)$ , which are not available with the BSS procedure. The SIR improvement for output  $i$  is calculated by Output SIR $_i$  – Input SIR $_i$ . These two types of SIRs are defined by the power ratio between the components related to the target sources and interference sources, at a specific microphone  $J$  and at the output  $i$ :

$$\text{Input SIR}_i = 10 \log_{10} \frac{\sum_t |x_{Ji}(t)|^2}{\sum_t |\sum_{k \neq i} x_{Jk}(t)|^2} \quad (\text{dB}),$$

$$\text{Output SIR}_i = 10 \log_{10} \frac{\sum_t |y_{ii}(t)|^2}{\sum_t |\sum_{k \neq i} y_{ik}(t)|^2} \quad (\text{dB}),$$

where  $y_{ik}$  is the component of  $s_k$  that appears at output  $y_i$ . The signal  $y_{ik}$  is calculated by applying the same separation operation to the individual source observations  $x_{1k}, \dots, x_{Mk}$  instead of the mixtures  $x_1, \dots, x_M$ . Such signals are decomposed components of the separated signal:  $y_i(t) = \sum_{k=1}^N y_{ik}(t)$ .



**Figure 10.25** Graphical interpretation of SIR and SDR definitions.

The SDR for output  $i$  is defined by the power ratio between the individual source observation  $x_{Ji}$  at a microphone  $J$  and the distortion in  $y_{ii}$ :

$$\text{SDR}_i = 10 \log_{10} \frac{\sum_t |\alpha_i x_{Ji}(t - \delta_i)|^2}{\sum_t |y_{ii}(t) - \alpha_i x_{Ji}(t - \delta_i)|^2} \quad (\text{dB}).$$

The distortion is defined in the denominator and is minimized by adjusting scalars  $\delta_i$  and  $\alpha_i$  for time and amplitude differences. The optimal  $\delta_i$  is obtained by maximizing the cross correlation

$$\delta_i = \operatorname{argmax}_{\delta} \sum_t y_{ii}(t) x_{Ji}(t - \delta).$$

Or, if the time difference between  $x_J$  and  $y_i$  is known based on the operations of the separation system, this information can be used simply for  $\delta_i$ . In either case, the optimal  $\alpha_i$  is then calculated by a least-mean-square estimator

$$\alpha_i = \frac{\sum_t y_{ii}(t) x_{Ji}(t - \delta_i)}{\sum_t |x_{Ji}(t - \delta_i)|^2}.$$

Figure 10.25 shows a graphical interpretation of Output SIR $_i$  and SDR $_i$ .

## REFERENCES

1. S. Haykin (Ed.), *Unsupervised Adaptive Filtering*, Vol. I: *Blind Source Separation*. Wiley, New York, 2000.
2. A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, Wiley, New York, 2001.
3. H. Buchner, R. Aichner, and W. Kellermann, “Blind source separation for convolutive mixtures: A unified treatment,” in *Audio Signal Processing for Next-Generation Multimedia Communication Systems*, Y. Huang and J. Benesty (Eds.), Kluwer Academic, Feb. 2004, pp. 255–293.
4. H. Sawada, R. Mukai, S. Araki, and S. Makino, “Frequency-domain blind source separation,” in *Speech Enhancement*, J. Benesty, S. Makino, and J. Chen (Eds.), Springer, Mar. 2005, pp. 299–327.
5. S. Amari, S. Douglas, A. Cichocki, and H. Yang, “Multichannel blind deconvolution and equalization using the natural gradient,” in *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications*, Apr. 1997, pp. 101–104.
6. P. Smaragdis, “Blind separation of convolved mixtures in the frequency domain,” *Neurocomputing*, vol. 22, pp. 21–34, 1998.

7. L. Parra and C. Spence, "Convulsive blind separation of non-stationary sources," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 3, pp. 320–327, May 2000.
8. J. Anemüller and B. Kollmeier, "Amplitude modulation decorrelation for convulsive blind source separation," in *Proc. ICA2000*, June 2000, pp. 215–220.
9. S. Araki, R. Mukai, S. Makino, T. Nishikawa, and H. Saruwatari, "The fundamental limitation of frequency domain blind source separation for convulsive mixtures of speech," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 2, pp. 109–116, 2003.
10. Ö. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Trans. Signal Process.*, vol. 52, no. 7, pp. 1830–1847, July 2004.
11. P. Bofill and M. Zibulevsky, "Blind separation of more sources than mixtures using sparsity of their short-time Fourier transform," in *Proc. ICA2000*, June 2000, pp. 87–92.
12. L. Vielva, D. Erdogmus, C. Pantaleon, I. Santamaría, J. Pereda, and J. Principe, "Underdetermined blind source separation in a time-varying environment," in *Proc. ICASSP2002*, Vol. 3, May 2002, pp. 3049–3052.
13. P. Bofill, "Underdetermined blind separation of delayed sound sources in the frequency domain," *Neurocomputing*, vol. 55, nos. 3/4, pp. 627–641, Oct. 2003.
14. A. Blin, S. Araki, and S. Makino, "Underdetermined blind separation of convulsive mixtures of speech using time-frequency mask and mixing matrix estimation," *IEICE Trans. Fund.*, vol. E88-A, no. 7, pp. 1693–1700, 2005.
15. S. Rickard and Ö. Yilmaz, "On the approximate W-disjoint orthogonality of speech," in *Proc. ICASSP2002*, Vol. I, May 2002, pp. 529–532.
16. S. Araki, S. Makino, A. Blin, R. Mukai, and H. Sawada, "Underdetermined blind separation for speech in real environments with sparseness and ICA," in *Proc. ICASSP2004*, Vol. III, May 2004, pp. 881–884.
17. L. Vielva, I. Santamaría, C. Pantaleon, J. Ibáñez, and D. Erdogmus, "Estimation of the mixing matrix for underdetermined blind source separation using spectral estimation techniques," in *Proc. EUSIPCO2002*, Vol. 1, Sept. 2002, pp. 557–560.
18. K. Waheed and F. Salem, "Algebraic overcomplete independent component analysis," in *Proc. ICA2003*, 2003, pp. 1077–1082.
19. F. Theis, "Mathematics in independent component analysis," PhD dissertation, University of Regensburg, 2002.
20. A. Ferréol, L. Albera, and P. Chevalier, "Fourth-order blind identification of underdetermined mixtures of sources (FOBIUM)," *IEEE Trans. Signal Process.*, vol. 53, no. 5, pp. 1640–1653, May 2005.
21. L. D. Lathauwer and J. Castaing, "Second-order blind identification of underdetermined mixtures," in *Proc. ICA2006*, Mar. 2006, pp. 40–47.
22. L. Albera, P. Comon, P. Chevalier, and A. Ferréol, "Blind identification of underdetermined mixtures based on the hexacovariance," in *Proc. ICASSP2004*, Vol. II, May 2004, pp. 29–32.
23. P. Bofill and E. Monte, "Underdetermined convoluted source reconstruction using LP and SOCP, and a neural approximator of the optimizer," in *Proc. ICA2006*, Mar. 2006, pp. 569–576.
24. Y. Deville, J. Chappuis, S. Hosseini, and J. Thomas, "Differential fast fixed-point BSS for underdetermined linear instantaneous mixtures," in *Proc. ICA2006*, Mar. 2006, pp. 48–56.
25. C. Wei, L. Khor, W. Woo, and S. Dlay, "Post-nonlinear underdetermined ICA by Bayesian statistics," in *Proc. ICA2006*, Mar. 2006, pp. 773–780.
26. S. Lesage, S. Krstulović, and R. Gribonval, "Under-determined source separation: Comparison of two approaches based on sparse decompositions," in *Proc. ICA2006*, Mar. 2006, pp. 633–640.

27. C. Févotte and S. Godsill, "Blind separation of sparse sources using jeffrey's inverse prior and the em algorithm," in *Proc. ICA2006*, Mar. 2006, pp. 593–600.
28. P. Comon and M. Rajih, "Blind identification of under-determined mixtures based on the characteristic function," in *ICASSP2005*, Vol. IV, Mar. 2005, pp. 1005–1008.
29. L. Albera, A. Ferreol, P. Comon, and P. Chevalier, "Blind identification of Overcomplete MixturEs of sources (BIOME)," *Linear Algebra Applications, Special Issue on Linear Algebra in Signal and Image Processing*, vol. 391C, pp. 3–30, Nov. 2004.
30. L. D. Lathauwer, "Simultaneous matrix diagonalization: The overcomplete case," in *Proc. ICA2003*, Apr. 2003, pp. 821–825.
31. L. D. Lathauwer, B. D. Moor, J. Vandewalle, and J.-F. Cardoso, "Independent component analysis of largely underdetermined mixtures," in *Proc. ICA2003*, Apr. 2003, pp. 29–34.
32. L. D. Lathauwer, P. Comon, B. D. Moor, and J. Vandewalle, "ICA algorithms for 3 sources and 2 sensors," in *Proc. IEEE Signal Processing Workshop on Higher-Order Statistics*, 1999, pp. 116–120.
33. S. Araki, H. Sawada, R. Mukai, and S. Makino, "A novel blind source separation method with observation vector clustering," in *Proc. IWAENC2005*, Sept. 2005, pp. 117–120.
34. R. Olsson and L. Hansen, "Blind separation of more sources than sensors in convolutive mixtures," in *Proc. ICASSP2006*, May 2006.
35. M. Pedersen, D. Wang, J. Larsen, and U. Kjems, "Separating underdetermined convolutive speech mixtures," in *Proc. ICA2006*, Mar. 2006, pp. 674–681.
36. A. Jourjine, S. Rickard, and Ö. Yilmaz, "Blind separation of disjoint orthogonal signals: Demixing N sources from 2 mixtures," in *Proc. ICASSP2000*, Vol. 5, pp. 2985–2988, June 2000.
37. M. Aoki, M. Okamoto, S. Aoki, H. Matsui, T. Sakurai, and Y. Kaneda, "Sound source segregation based on estimating incident angle of each frequency component of input signals acquired by multiple microphones," *Acoust. Sci. Technol.*, vol. 22, no. 2, pp. 149–157, 2001.
38. N. Roman, D. Wang, and G. J. Brown, "Speech segregation based on sound localization," *J. Acoust. Soc. Am.*, vol. 114, no. 4, pp. 2236–2252, Oct. 2003.
39. F. Theis, E. Lang, and C. Puntonet, "A geometric algorithm for overcomplete linear ICA," *Neurocomputing*, vol. 56, pp. 381–398, 2004.
40. S. Winter, W. Kellermann, H. Sawada, and S. Makino, "MAP-based underdetermined blind source separation of convolutive mixtures by hierarchical clustering and  $l_1$ -norm minimization," *EURASIP J. Adv. Signal Process.*, vol. 2007, Article ID 24717, pp. 1–12, Jan. 2007.
41. J. M. Peterson and S. Kadamb, "A probabilistic approach for blind source separation of underdetermined convolutive mixtures," in *Proc. ICASSP2003*, Vol. VI, Apr. 2003, pp. 581–584.
42. S. Rickard, R. Balan, and J. Rosca, "Real-time time-frequency based blind source separation," in *Proc. ICA2001*, Dec. 2001, pp. 651–656.
43. D. Ellis, "Prediction-driven computational auditory scene analysis," PhD dissertation, MIT, Cambridge, MA, 1996.
44. J. Burred and T. Sikora, "On the use of auditory representations for sparsity-based sound source separation," in *Proc. IEEE Fifth Int. Conf. on Information, Communications and Signal Processing (ICICS)*, Dec. 2005.
45. R. Saab, O. Yilmaz, M. McKeown, and R. Abugharbieh, "Underdetermined sparse blind source separation with delays," in *Signal Processing with Adaptive Sparse Structured Representations Workshop (SPARS)*, 2005.
46. N. Linh-Trung, A. Belouchrani, K. Abed-Meraim, and B. Boashash, "Separating more sources than sensors using time-frequency distributions," *EURASIP J. Appl. Signal Process.*, vol. 2005, no. 17, pp. 2828–2847, 2005.

47. M. Zibulevsky and B. Pearlmutter, "Blind source separation by sparse decomposition," *Neural Comput.*, vol. 13, no. 4, pp. 863–882, 2001.
48. S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," Technical Report, Department of Statistics, Stanford University, Stanford, CA, 1995.
49. P. G., D. Nuzillard, and A. Ralescu, "Sparse deflations in blind signal separation," in *Proc. ICA2006*, Mar. 2006, pp. 807–814.
50. Y. Luo, W. Wang, J. Chambers, S. Lambotharan, and I. Proudler, "Exploitation of source nonstationarity in underdetermined blind source separation with advanced clustering techniques," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2198–2212, June 2006.
51. C. Chang, P. C. Fung, and Y. S. Hung, "On a sparse component analysis approach to blind source separation," in *Proc. ICA2006*, Mar. 2006, pp. 765–772.
52. B. A. Pearlmutter and V. K. Potluru, "Sparse separation: Principles and tricks," *Proc SPIE*, vol. 5102, pp. 1–4, Apr. 2003.
53. I. Gorodnitsky and B. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 600–616, Mar. 1997.
54. J. Karvanen and A. Cichocki, "Measuring sparseness of noisy signals," in *Proc. ICA2003*, Apr. 2003, pp. 125–130.
55. S. Rickard, "Sparse sources are separated sources," in *Proc. EUSIPCO2006*, Sept. 2006.
56. R. Balan, J. Rosca, and S. Rickard, "Non-square blind source separation under coherent noise by beamforming and time-frequency masking," in *Proc. ICA2003*, Apr. 2003, pp. 313–318.
57. S. Araki, H. Sawada, R. Mukai, and S. Makino, "Underdetermined blind sparse source separation for arbitrarily arranged multiple sensors," *Signal Process.*, vol. 87, pp. 1833–1847, Feb. 2007.
58. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed., Wiley, New York, 2000.
59. H. Sawada, S. Araki, and S. Makino, "A two-stage frequency-domain blind source separation method for underdetermined convolutive mixtures," in *Proc. WASPAA2007*, Oct. 2007, pp. 139–142.
60. S. Araki, S. Makino, H. Sawada, and R. Mukai, "Reducing musical noise by a fine-shift overlap-add method applied to source separation using a time-frequency mask," in *Proc. ICASSP2005*, Vol. III, Mar. 2005, pp. 81–84.
61. P. Comon, "Blind channel identification and extraction of more sources than sensors," in *Proc. SPIE*, 1998, pp. 2–13.
62. A. Taleb, "An algorithm for the blind identification of N independent signal with 2 sensors," in *Proc. ISSPA2001*, Aug. 2001, pp. 5–8.
63. J.-F. Cardoso, "Super-symmetric decomposition of the fourth-order cumulant tensor blind identification of more sources than sensors," in *Proc. ICASSP91*, Vol. V, 1991, pp. 3109–3112.
64. F. Theis and E. Lang, "Formalization of the two-step approach to overcomplete BSS," in *Proc. of SIP2002*, 2002, pp. 207–212.
65. K. Waheed, "Blind source recovery: State space formulations," Technical Report, Department of Electrical and Computer Engineering, Michigan State University, Sept. 2001.
66. S. Winter, H. Sawada, S. Araki, and S. Makino, "Overcomplete BSS for convolutive mixtures based on hierarchical clustering," in *Proc. ICA2004*, Sept. 2004, pp. 652–660.
67. S. Winter, H. Sawada, and S. Makino, "On real and complex valued  $l_1$ -norm minimization for overcomplete blind source separation," in *Proc. WASPAA2005*, Oct. 2005, pp. 86–89.
68. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, Springer, 2002.

69. L. Vielva, D. Erdogmus, and J. C. Principe, "Underdetermined blind source separation using a probabilistic source sparsity model," in *Proc. ICA2001*, Dec. 2001, pp. 675–679.
70. D. Donoho and M. Elad, "Optimally-sparse representation in general (non-orthogonal) dictionaries via  $l_1$  minimization," *Proc. Natl. Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, Mar. 2003.
71. C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
72. A. Gelman, J. Carlin, H. Stern, and D. Rubin, *Bayesian Data Analysis*, Chapman & Hall, 1995.
73. A. Papoulis and S. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed., McGraw-Hill, New York, 2002.
74. J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11/12, pp. 625–653, 1999, special issue on Interior Point Methods.
75. I. Takigawa, M. Kudo, and J. Toyama, "Performance analysis of minimum  $l_1$ -norm solutions for underdetermined source separation," *IEEE Trans. Signal Process.*, vol. 52, no. 3, pp. 582–591, Mar. 2004.
76. A. Pruessner, M. Bussieck, S. Dirkse, and A. Meeraus, "Conic programming in GAMS," in *INFORMS Annual Meeting*, Atlanta, Oct. 2003, pp. 19–22.
77. H. Sawada, S. Araki, R. Mukai, and S. Makino, "Blind extraction of a dominant source signal from mixtures of many sources," in *Proc. ICASSP2005*, Vol. III, Mar. 2005, pp. 61–64.
78. H. Sawada, R. Mukai, S. Araki, and S. Makino, "A robust and precise method for solving the permutation problem," *IEEE Trans. Speech Audio Process.*, vol. 12, pp. 530–538, Sept. 2004.
79. K. Matsuoka, "Independent component analysis and its applications to sound signal separation," in *Proc. IWAENC2003*, Sept. 2003, pp. 15–18.
80. S. Araki, H. Sawada, R. Mukai, and S. Makino, "DOA estimation for multiple sparse sources with normalized observation vector clustering," in *Proc. ICASSP2006*, Vol. 5, May 2006, pp. 33–36.



---

## CHAPTER 11

---

# Array Processing in Astronomy

Douglas C.-J. Bock

Combined Array for Research in Millimeter-Wave Astronomy, University of California,  
Berkeley, California

### 11.1 INTRODUCTION

Most array processing in astronomy is in radio astronomy. Arrays are used to obtain resolution and image quality unobtainable with single reflectors. The high image quality facilitates quantitative analysis and comparison with images at other wavelengths. Radio astronomy arrays can be divided into two main classes, beamforming arrays and correlation arrays. The beamforming arrays produce instantaneous summed array beams from a direction of interest. Correlation arrays provide images over the entire single-element primary beam pattern, computed off-line from records of all the possibles correlations between the antennas, pairwise. The next generation of instruments will use techniques from both types of instruments.

This chapter will focus primarily on correlation arrays, including their theory, design, and processing, but also mention aperture-plane phased arrays, focal-plane phased arrays, and array processing at optical and infrared wavelengths. It will follow the conventions usual in radio astronomy, although the results are applicable to (and increasingly used in) optical astronomy. The first part of array processing in radio astronomy occurs before the instrument is even built: the design of the antenna configuration for high-quality imaging of complex structures, in the presence of significant thermal noise and instrumental errors. This implies a much lower side-lobe level than can be obtained with an array (beamforming or correlated) having regularly spaced antennas. This chapter will cover the design of the antenna configuration in some detail.

### 11.2 CORRELATION ARRAYS

Correlation arrays are arrays of antennas (commonly parabolic reflectors) that are analyzed by forming the cross correlation between each pair of antennas (each interferometer). The cross correlation is implemented by digital or analog multiplication after equalizing the path lengths to each antenna in the interferometer. These arrays have been used widely in radio astronomy to allow increased angular resolution over

what could be obtained with a single reflector. They allow imaging of the entire field of view of the individual antenna at a resolution set by the overall extent of the array. The use of the cross correlation of pairs of antennas allows imaging with antennas having arbitrary position in the aperture plane. This section presents the correlation array and describes its use for astronomical imaging.

### 11.2.1 Aperture Synthesis

Aperture synthesis in radio astronomy is the procedure of using a sparsely sampled aperture distribution of antennas to estimate the entire aperture field. Since astronomical sources are complex, the array must be designed with an excellent point spread function (often called the *synthesized beam*) in order to avoid confusion between the source and the side-lobe structure. And since astronomical sources vary widely in size, there is a diversity of scales, both within arrays and between arrays. Although images can often be made near instantaneously, it is common to accumulate data for periods of up to several hours in order to obtain sufficient signal to noise and to improve the sampling of the aperture plane *in the frame of the source*, and thus the response of the instrument to complex structures. This technique is called *Earth rotation aperture synthesis* [1] and will be described shortly. Aperture synthesis arrays vary from those just a few meters in size, for measuring structures of order degrees in the cosmic microwave background, to arrays of a few antennas distributed sparingly across the globe, and in Earth orbit that can resolve superluminal jets in active galactic nuclei at distances of billions of light years. The problem of designing an antenna configuration will be discussed in Section 11.2.3.

We begin the description of aperture synthesis by considering the response of the two-element interferometer to a distant source. This section follows the notation of [2]. See this reference for additional details: We can provide only the main results here.

Consider a source distribution  $I(\mathbf{s})$  on the celestial sphere, where  $\mathbf{s}$  is a unit vector (Fig. 11.1). We require the source to be spatially incoherent (noiselike) and in the far field of the array. The source is viewed by a pair of antennas with baseline in wavelengths  $\mathbf{D}_\lambda$  and antenna pattern (effective collecting area)  $A(\mathbf{s})$ . A portion of the source,  $d\Omega$ , contributes power  $\frac{1}{2}A(\mathbf{s})I(\mathbf{s})\Delta\nu d\Omega$  at each of the antennas, where  $\Delta\nu$  is the bandwidth of the receiving system. The signals at the two antennas are multiplied in a correlator, introducing a fringe (i.e., sinusoidally varying) pattern with fringe frequency  $\nu\tau_g$ , where  $\tau_g = \mathbf{D}_\lambda \cdot \mathbf{s}/\nu$  is the geometric delay. Then we can write the correlator output as

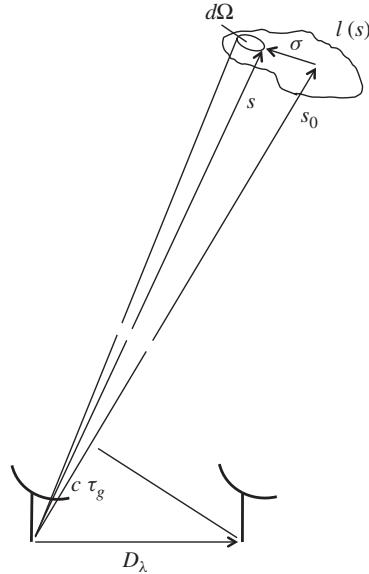
$$dr = A(\mathbf{s})I(\mathbf{s})\Delta\nu d\Omega \cos 2\pi\nu\tau_g. \quad (11.1)$$

Integrating to get the response of the interferometer to the entire source, we obtain

$$r = \Delta\nu \int_S A(\mathbf{s})I(\mathbf{s}) \cos 2\pi\mathbf{D}_\lambda \cdot \mathbf{s} d\Omega. \quad (11.2)$$

The interferometer measures the complex visibility, integrated over some period, commonly seconds. The visibility is defined as

$$\mathcal{V} = |\mathcal{V}|e^{j\phi_v} = \int_{4\pi} A_N(\boldsymbol{\sigma})I(\boldsymbol{\sigma})e^{-j2\pi\mathbf{D}_\lambda \cdot \boldsymbol{\sigma}} d\Omega, \quad (11.3)$$



**Figure 11.1** Positions vectors used to describe the interferometer response to a source on the celestial sphere.

where we have normalized the antenna pattern to the response at the beam center, that is,  $A_N(\sigma) = A(\sigma)/A_0$ , and referred to the phase center of the observation,  $\mathbf{s}_0$ , that is,  $\mathbf{s} = \mathbf{s}_0 + \sigma$  (Fig. 11.1).

By separating out the real and imaginary parts of (11.3) and comparing to (11.2), we can express the output of the correlator as a fringe pattern compared to the field center,  $\mathbf{s}_0$ , at which the fringe frequency would be zero since the interferometer has been phased up in that direction:

$$r = A_0 \Delta v |\mathcal{V}| \cos(2\pi \mathbf{D}_\lambda \cdot \mathbf{s}_0 - \phi_v). \quad (11.4)$$

The fundamental technique of aperture synthesis, is, then, to record  $r$  as function of the visibility amplitude  $|\mathcal{V}|$  and phase  $\phi_v$ , and to use these to estimate the source structure  $I(\sigma)$ . The way this is done in practice is to choose coordinate systems so that the inversion can be expressed as a two-dimensional Fourier transform, evaluated either as a direct Fourier transform (DFT) or a fast Fourier transform (FFT).

Next, we consider how the source of interest can be described in terms of the visibilities that the array measures. The terrestrial coordinate system used is right handed and has components  $(u, v, w)$  of the baseline  $\mathbf{D}_\lambda$  projected onto the plane normal to the direction to the source, where  $u$  is measured to the east,  $v$  to the north, and  $w$  toward the source. The celestial coordinate system has components on the unit sphere  $(l, m)$  where  $l$  and  $m$  are directional cosines of  $u$  and  $v$ , respectively. In this formalism, a third celestial coordinate is not given, as the sources are assumed to be distant. Calculation of  $\mathbf{D}_\lambda \cdot \sigma$  using these coordinate systems [2] provides a two-dimensional Fourier relation for the case when  $|l|$  and  $|m|$  are small enough that

the products with  $w$  can be neglected, that is,

$$\mathcal{V}(u, v, w) \simeq \mathcal{V}(u, v, 0) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{A_N(l, m) I(l, m)}{\sqrt{1 - l^2 - m^2}} e^{-j 2\pi(u l + v m)} dl dm. \quad (11.5)$$

Thus we can invert the transform to obtain

$$\frac{A_N(l, m) I(l, m)}{\sqrt{1 - l^2 - m^2}} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathcal{V}(u, v) e^{j 2\pi(u l + v m)} du dv. \quad (11.6)$$

This is the monochromatic version of the van Cittert–Zernicke theorem, which states that for a monochromatic incoherent source, the spatial coherence function is related to the source intensity distribution by the Fourier transform: The image is represented by the Fourier transform of the visibilities. A comprehensive description of the theorem is given by [3].

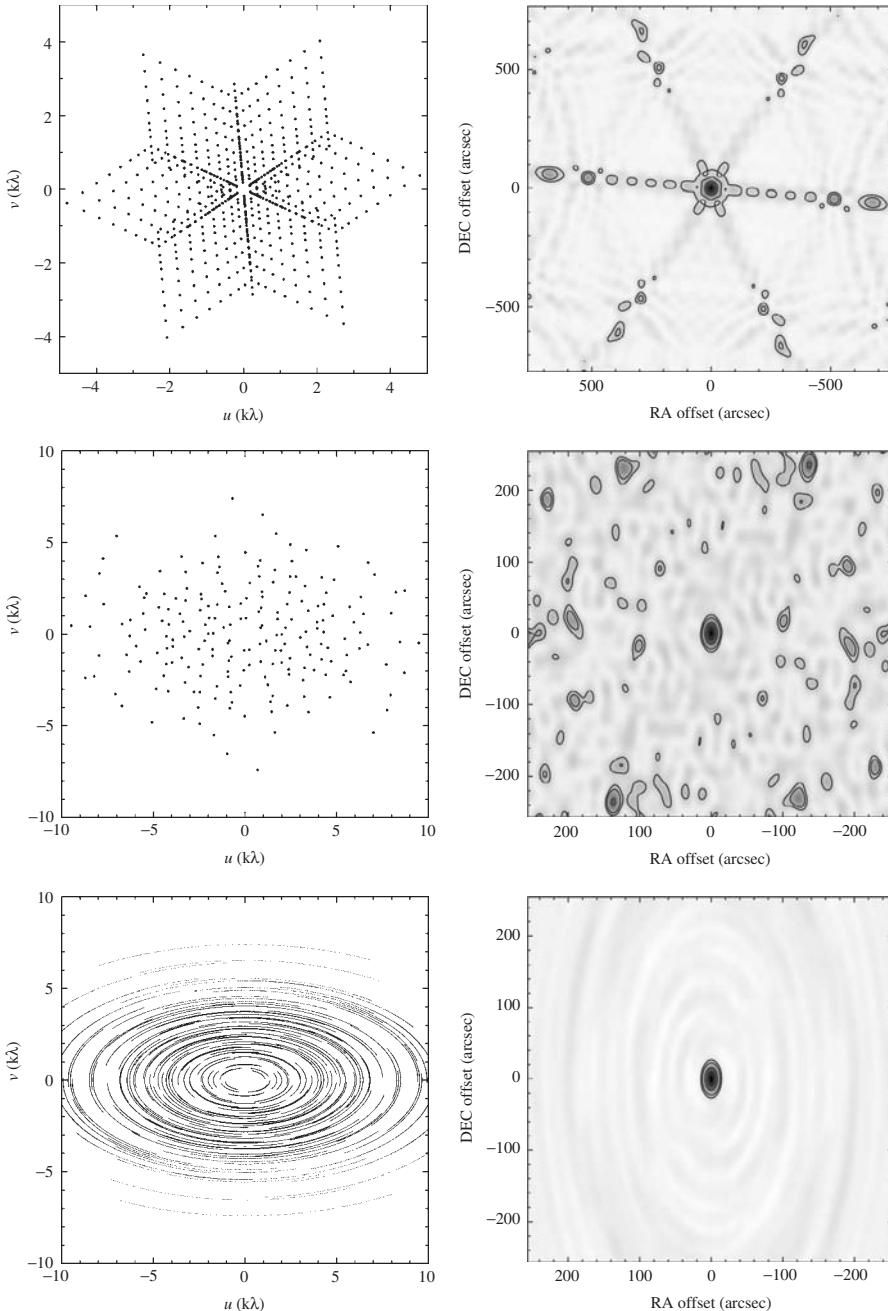
In describing the response of an aperture synthesis observation to a source, it is useful to consider the distribution of interferometer pairs (baselines) in the array. There are  $N(N - 1)/2$  baselines between  $N$  antennas. This is done in the Fourier plane, often referred to as the *uv plane*. The plane contains the *uv* coordinates of each visibility measured. Since the *uv* coordinates describe the baseline projected in the plane perpendicular to the direction to the source, the magnitude and argument represent the length and orientation of the baseline. Examples of the *uv* plane are given in Figure 11.2. The points in the outer portion of the plane represent observations with long baselines and the measurement of high-resolution information. The points near the center of the plane correspond to short baselines and the sensitivity to extended structure. In a long observation, the projected baseline varies with the rotation of the Earth, tracing out an ellipse. The act of filling the *uv* plane in this manner is known as Earth rotation aperture synthesis. The increased *uv* coverage arises as the array images the same sky at many different orientations. Image quality is improved with increasing coverage of the *uv* plane.

The limitations in  $w$ ,  $l$ , and  $m$  that allowed the approximation in (11.5) imply that for an array whose baselines vary in  $w$  (are noncoplanar) imaging is limited to small  $l$  and  $m$ , that is, to small fields. We note that an array with coplanar antennas will not in general have coplanar baselines during an Earth rotation synthesis. However, a linear east–west array of antennas can have coplanar baselines if  $l$  and  $m$  are chosen in a tangent plane to the north or south celestial pole. The *uv* plane will be filled with ellipses centered on the origin. Good *uv* coverage is obtained with a 12-h observation.

To quantify the limitation, we consider low observing elevations, where  $w$  becomes comparable to  $\mathbf{D}_\lambda$ . A condition that no phase errors exceed 0.1 radians implies that  $\theta_f < \frac{1}{3}\sqrt{\theta_b}$ , where  $\theta_f$  and  $\theta_b$  are the widths of the synthesized field and beam in radians, respectively. For example, this condition implies that an array with arcsecond resolution should be used to image over a field of size less 2.5 arcminutes. Although much radio astronomy imaging is done in this regime, it has been necessary to use compute-intensive routines that avoid the restriction for large-area surveys. See [2] for more details.

### 11.2.2 Data Acquisition and Correlation

A parabolic reflector is the typical antenna in a correlation array operating between 500 MHz and 900 GHz, while dipole-class antennas are used at the lowest frequencies.



**Figure 11.2** Fourier ( $uv$ ) plane coverage and point-spread function (synthesized beam) for the VLA D configuration snapshot (top), CARMA D configuration snapshot (middle), and CARMA 4-h Earth rotation synthesis (bottom). The beam contours are at 10, 20, 50, and 90% of the peak. Note how the Earth rotation synthesis causes the  $uv$  points from the snapshot to trace out ellipses. The greater filling of the  $uv$  plane aids the recovery of complex structure, which is equivalent to noting that the synthesized beam is of higher quality.

We will not discuss antennas nor the feeds, mixers, and amplifiers that follow. A comprehensive review may be found in [2] and other standard texts. The defining characteristic is that sensitivity is almost always a limiting factor, and thus designs optimizing system temperature and collecting area are predominant. Here, we will focus on the array processing aspects of the correlation radio astronomy array.

The signals from the antennas are brought back to a central location either at the sky frequency or at some intermediate frequency. It is becoming common to transport signals back at the sky frequency where they are  $\lesssim 10$  GHz, in order that the full bandwidth may be available for current or future wide-band instrumentation. This also avoids the need to transport a local oscillator signal to the antennas. The choice of whether to digitize at the antenna or at the central laboratory depends on the balance between desired bandwidth and technology, with a preference for digitizing at the antenna to reduce nonidealities in the signal transport. In many wide-band systems, the intermediate frequency signal is further down-converted into several bands before digitization.

As mentioned above, the radio astronomy cross correlator produces visibilities, which are typically integrated for several seconds in a manageable number (up to tens of thousands) of frequency channels. Any processing that is required on shorter time scales (e.g., delay correction on long baselines, certain types of radio frequency interference mitigation) must be done online. The visibility data are stored for later analysis and reanalysis. For example, future analyses could vary calibration, data editing, imaging, and deconvolution choices. For later processing, the observing parameters of each visibility (projected baseline, frequency) are assumed to be fixed for the duration of the integration. For some future instruments the overwhelming number of visibilities may require that multiple spectral-plane images, rather than the visibility data, are the archived quantity.

Where sources are time varying on rapid time scales, they are understood to be compact (no larger than the light-crossing time) and therefore not resolved by the typical imaging array.<sup>1</sup> Therefore, it is often preferable to choose an instrument with a beamformer (or a large single dish) for more cost-effective observations of such sources.

### 11.2.3 Antenna Configuration

The design of the antenna configuration is a complex optimization based on the expected future scientific utility of the instrument. Since many radio telescopes are general-purpose instruments, with lifetimes upward of 30 years, it is often preferable to make the configurations as generally useful as possible: sensitive to a wide variety of spatial scales and perhaps reconfigurable. With careful attention to the design it is possible to get an excellent point-spread function, even with significant constraints on the antenna positions, since it is the distributions of the antenna spacings in the Fourier ( $uv$ ) plane that matter. The general goal is to get as much power (and thus sensitivity) as possible into a compact and round point-spread function. To first order, this point-spread function is constant across the entire image.

<sup>1</sup>Even at the nearest stars ( $>1$  parsec, or 3.26 light years distant) a light-second subtends only 6 milliarcseconds, smaller than the resolving power of all but the very long baseline arrays. However, future special-purpose instruments for imaging rapid variations, such as in the relatively nearby sun, will make images on time scales of tens of milliseconds [4].

The first problem is to determine the size and number of antennas. For  $N$  antennas of diameter  $D$ , point-source sensitivity goes as  $ND^2$  (the total collecting area), while wide-area imaging speed (i.e., for areas larger than the field of view of the individual element) goes as  $ND$ . This implies that a survey instrument should comprise many small antennas, while an instrument to observe compact sources could be made of fewer larger antennas. Even when observing small fields, imaging performance may indicate a preference for tens or hundreds of antennas, especially if instantaneous (“snapshot”) imaging performance is important. The exact choice will depend on antenna construction techniques and the relative affordability of those parts of the system that go as  $N$  (antennas and the precorrelator signal path) and those that go as  $N(N - 1)/2$ , that is,  $O(N^2)$  (the correlator and facilities for dealing with its data). Both costs will be functions of the system bandwidth, so the choice for an instrument primarily designed for narrowband spectral line imaging may be different from the choice for continuum imaging over very wide (many gigahertz) bandwidths.

If a wide range of spatial frequencies is to be imaged, then a reconfigurable array may be considered. Reconfigurable arrays have used railroad tracks or roads to facilitate antenna movement. Custom transporters are required in either case. Arrays of large numbers of antennas may need to keep observing during near-continuous reconfiguration to be efficient. The Atacama Large Millimeter/submillimeter Array (ALMA, Section 11.2.7), currently under construction, will use this technique.

Configuration designs fall broadly into those generated by some kind of rule, such as a geometric progression of stations on a linear track, and those produced by numerical optimization. In either case, there should be a target distribution of  $uv$  samples (visibilities) in the  $uv$  plane. A Gaussian distribution of samples will lead to a Gaussian point-spread function, which is optimum for high-fidelity imaging. A uniform distribution improves the resolution, at the expense of higher side lobes. The higher side lobes due to the sharp cutoff at the edge of the uniform distribution (which is a sampled form of a two-dimensional rectangular function) become more of a concern at large  $N$ , where the large number of antennas provides the opportunity for very low side lobes in a more tapered distribution. It is also possible to weight the visibility samples in the  $uv$  plane to optimize the imaging for a particular problem (Section 11.2.4), at some cost to sensitivity.

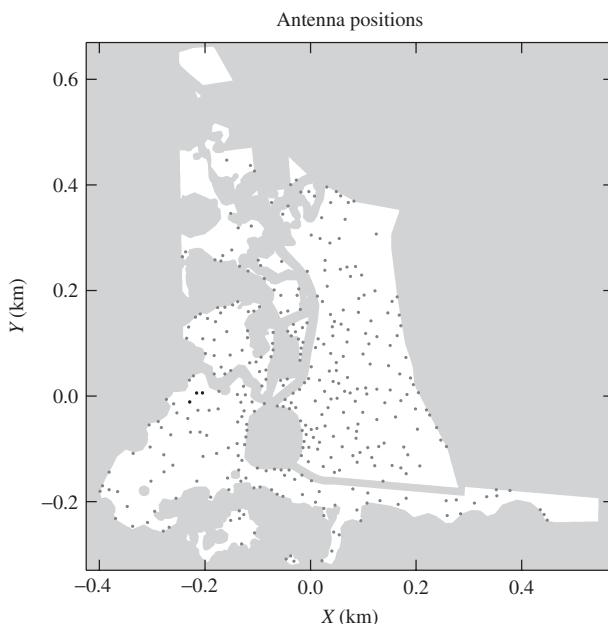
The simplest designs, such as regular rectangular grids, have been avoided for imaging arrays owing to the strong grating response and the excess of redundant spacings. More sparse arrays designed by rule tend to have geometric patterns designed to provide as few redundant antenna pairs as possible. Figure 11.2 shows the  $uv$  coverage and synthesized beam of the 27-antenna Very Large Array (VLA, Section 11.2.7). The instantaneous distribution of the  $uv$  samples shows signs of the three arms that were used to minimize antenna transport costs. The antennas are reconfigured with transporters on railway tracks. In this example, the threefold symmetry can clearly be seen in the synthesized beam. The beam quality can be substantially improved by Earth rotation synthesis. The Australia Telescope Compact Array and Westerbork Synthesis Radio Telescope (Section 11.2.7) have similar designs, but the antennas of each array lie on a single east–west line, relying on Earth rotation aperture synthesis for imaging. For comparison, Figure 11.2 also shows an array of fewer (15) antennas, the Combined Array for Research in Millimeter-wave Astronomy (CARMA, Section 11.2.7). The antenna transporter is unconstrained by railway tracks, and the antenna configuration has been optimized to a target  $uv$  distribution. Thus the snapshot

distribution produces a high-quality beam, which is further improved by a 4-h Earth rotation synthesis.

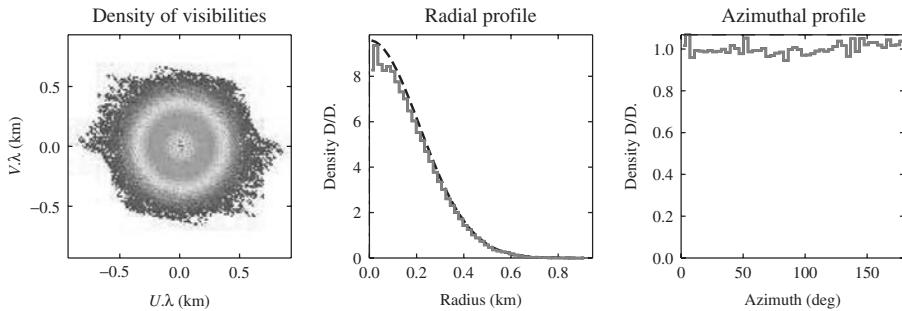
Modern arrays of large numbers of antennas have commonly been designed by computer optimization to take into account geographic constraints while conforming to a desired  $uv$  plane coverage. Several algorithms have been used. One method [5] uses a Kohonen self-organized neural network algorithm to optimize the instantaneous zenith performance of an array, under the assumption that a uniform Fourier distribution is preferred. As mentioned above, this distribution provides maximum resolution for a given available area, but at the cost of increased side lobes. This algorithm is computationally relatively expensive.

Another method [6] evaluates the response of the array in the image plane and optimizes antenna positions to reduce the level of the positive side lobes. This relies on the result [7] that the level of *negative* side lobes in the response function of a pseudorandom  $N$ -element array is proportional to  $1/(N - 1)$ , irrespective of the actual array configuration. This method produces an excellent result in the design of compact configurations, by avoiding grating responses. However, since it does not constrain the  $uv$  coverage, it is of limited applicability in the design of a more sparse configuration.

A very flexible method [8, 9] has been constructed for the design of recent larger- $N$  arrays such as the Allen Telescope Array (ATA) and the Atacama Large Millimeter/submillimeter Array (Section 11.2.7). This method allows arbitrary topographical constraints (e.g., Fig. 11.3) and target  $uv$  distributions, including distributions from long (Earth rotation synthesis) observations. In each cycle of the algorithm the  $uv$  distribution of the current antenna distribution is compared to the target distribution, to calculate the over- or underdensity throughout the  $uv$  plane (Fig. 11.4). To improve

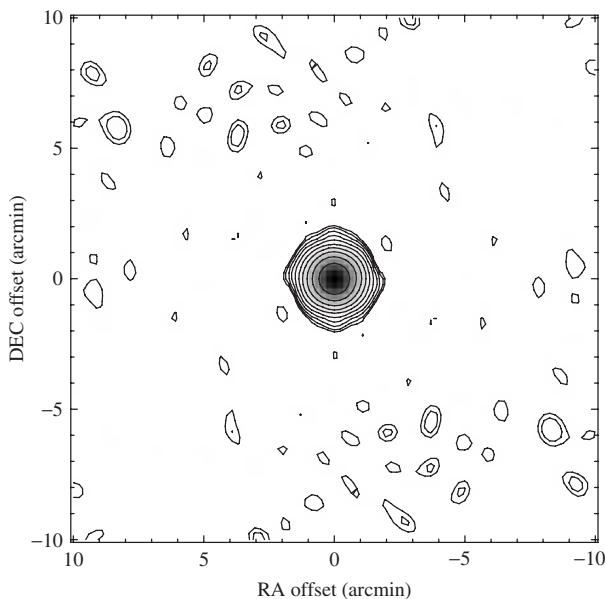


**Figure 11.3** Antenna positions overlaid on the site mask used for the Allen Telescope Array (ATA) configuration design.

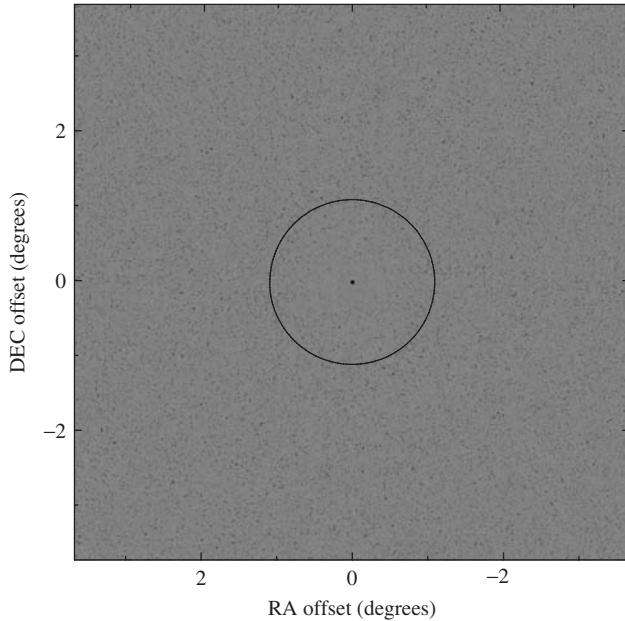


**Figure 11.4** Fourier ( $uv$ ) plane analyses of the antennas in Figure 11.3. Despite severe physical constraints placed on the antenna locations, it is possible to have a very close agreement to the target Gaussian distribution of  $uv$  samples.

the array, the  $uv$  distribution of samples due to one antenna (i.e., in pairwise combination with each other antenna) is compared to the difference between the model and actual density over the entire  $uv$  plane. A gradient toward a position that reduces this difference is calculated and the antenna position is changed appropriately. The algorithm is of order  $N^2$  and thus has greater applicability to large- $N$  designs than many slower algorithms. The impact of local minima is reduced by introducing random positional errors every few cycles. The algorithm can easily incorporate fixed antennas. The results of this algorithm can be an excellent response despite severe site constraints (Fig. 11.5). The algorithm provides very low side lobes in the inner area



**Figure 11.5** Inner region of the synthesized beam (point-spread function) for the ATA configuration in Figure 11.3. The contours are logarithmic starting at  $\pm 0.5$ ,  $\pm 0.9$ ,  $\pm 1.6\%$ . The response closely approximates a Gaussian, with low side lobes.



**Figure 11.6** Synthesized beam over the area of the primary beam of the 6-m ATA antennas at 1.4 GHz [full width at half maximum (FWHM) shown as a circle], with the gray scale saturated at  $\pm 5\%$ . Note the uniform, noiselike side lobes.

of the array response (Fig. 11.6), approximately the inner  $N$  resolution elements, as expected from the  $N$ -independent parameters (the antennas). Beyond this reason, the side lobe’s “noise” is  $0.3\%$  root mean square (rms),  $\sim 1/N$ . Individual side lobes can be much higher, up to several percent. Generally, these will fall outside the primary response of the antenna and be attenuated in the final image. If desired, a limited number of these side lobes can be reduced by fine-tuning the antenna distribution [10].

#### 11.2.4 Imaging

The goal of the imaging system is to produce an accurate representation of the sky from the measured visibilities. The typical imaging system consists of general-purpose computers capable of performing the FFT with appropriate weighting, mosaicking multiple fields, and performing a deconvolution. The computing requirements can become substantial for large mosaicked images with many channels. Most systems are offline, although the tendency in the coming large- $N$  regime will be toward online or pipeline imaging in order to limit stored data volumes. A more detailed discussion will be found in [11].

In Section 11.2.1, we introduced the basic measurable of the correlator array, the complex visibility (11.3). In practice, only a finite number  $M$  of visibilities are actually measured. The sampling function,  $S(u, v)$  (expressed in terms of the two-dimensional Dirac delta function) denotes this:

$$S(u, v) = \sum_{k=1}^M \delta(u - u_k, v - v_k). \quad (11.7)$$

Thus, from (11.6),

$$I^D(l, m) \equiv \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} S(u, v) \mathcal{V}(u, v) e^{j 2\pi (ul + vm)} du dv. \quad (11.8)$$

where  $I^D$  is the so-called dirty image and  $\mathcal{V}$  denotes the corruption of  $V$  by noise. We have renormalized from (11.6). We can now write the sampled visibility function as

$$\mathcal{V}^S(u, v) \equiv \sum_{k=1}^M \delta(u - u_k, v - v_k) \mathcal{V}(u_k, v_k). \quad (11.9)$$

If we write the Fourier transform operator as  $\mathcal{F}$ , then from (11.8),

$$\begin{aligned} I^D = \mathcal{F}\mathcal{V}^S &= \mathcal{F}(S\mathcal{V}) \\ &= \mathcal{F}S * \mathcal{F}\mathcal{V}, \end{aligned} \quad (11.10)$$

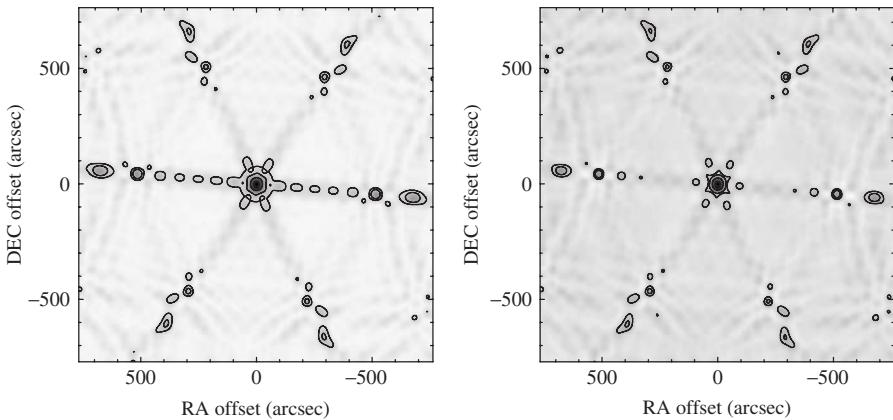
where the  $*$  denotes convolution. That is, the observed brightness distribution is the point-spread function of the array, the synthesized beam (the Fourier transform of the sampling function), convolved with the true brightness distribution (estimated by the Fourier transform of the measured visibilities). *Deconvolution* is needed to recover the brightness distribution.

**11.2.4.1 Weighting** A weighting of the  $uv$  samples is often performed before imaging. The data are weighted in three main ways: to recognize the data reliability, to modify the  $uv$  density, and to *taper* the data. The purpose is to arbitrarily redefine the aperture before Fourier transforming back to the image.

The most common decision the astronomer has to make when choosing weighting functions is whether to favor imaging quality or signal to noise. In order to weight the reliability of the data, the integration time, bandwidth, and system temperature will be taken into account. This weighting will minimize the noise in the image and produces the best sensitivity. At high radio frequencies, where system temperatures ( $T_{\text{sys}}$ ) are often dominated by atmospheric emissions, a weighting is commonly applied to take into account the variation in sensitivity with weather and elevation. Note that this is separate from the earlier step of calibrating the system gain variation with system temperature. For an observation of a very faint source, or a program to detect a source of unknown strength, the usual choice would be *natural* weighting, in which each  $uv$  sample is given equal weight. Other weighting schemes will tend to lead to an image with higher rms noise.

In cases where the need for accurate imaging of complex structure dominates the sensitivity requirements, uniform weighting may be used. In this scheme, visibilities are weighted in inverse proportion to their density in the  $uv$  plane. Figure 11.7 shows the improvement uniform weighting makes in the beam of a snapshot observation with the VLA. Depending on the details of the observation, the signal-to-noise cost can be up to 50%, or approximately twice the observing time to reach a given sensitivity limit. There are additional weighting schemes that provide improved imaging quality with a lesser increase in the noise level [11].

Finally, data may be weighted by applying a smooth (usually radially symmetric) *tapering* function that down weights data in the outer portion of the  $uv$  plane. The



**Figure 11.7** Synthesized beam for the VLA D configuration snapshot with natural weighting (left; same as Fig. 11.2) and uniform weighting (right). The uniformly weighted beam has a better synthesized beam at the expense of higher noise and side lobes.

purpose of the down weighting is to optimize the sensitivity of the image at a particular spatial scale, by removing the noise contributions of baselines too long to be sensitive to that scale. It is also used to facilitate comparison to other images.

**11.2.4.2 Gridding** For calculation by a fast Fourier transform (the most common method), the visibility data must be gridded onto a regular rectangular grid. The standard method is to convolve the samples by a smoothing function so that the normalized sum of the visibilities can be determined at each regular grid point. A variety of convolution functions are in use. Ideally, one would use a function whose Fourier transform drops off rapidly at the edge of the image, to avoid aliasing other sources outside the field. In practice, this requires sufficiently unconfined convolution functions that the computing becomes expensive (typical convolution functions in use have size  $\lesssim 10$  cells). So aliasing needs to be avoided by choice of the image and cell sizes and, preferably, by including all features of interest in the field of view of the observation.

**11.2.4.3 Deconvolution** The role of deconvolution is to estimate the source distribution  $I = \mathcal{F}V$  by removing its convolution with the array response (11.10) from the image. This amounts to estimating  $V$  for the entire plane, from those measurements actually made. The problem cannot be solved by application of a linear process since this would allow any estimate of unmeasured spatial frequencies (known as “invisible distributions”), even those not plausible given the known information. These spatial frequencies correspond to the holes in the  $uv$  plane, and to extrapolation beyond the edge of the measured samples (i.e., at higher resolution). A nonlinear process is required.

In radio astronomy, the most common nonlinear algorithms are variants of CLEAN (devised for radio astronomy by Högbom [12]) or the maximum entropy method. Detailed surveys of deconvolution in radio astronomy may be found in [13, 14]. This process is often the most computer-intensive part of the imaging process, and considerable attention has been given to optimizing the procedures.

The CLEAN algorithm represents the entire image by a distribution of point sources. The CLEAN procedure is easy to perform but not well understood theoretically. It is done as follows. The brightest pixel is found, and the point-spread function at this

point is subtracted from the image, usually with some gain factor  $<1$ . The pixel's magnitude and position is recorded. This is called a *CLEAN component*. The process is repeated until some cutoff (set above the thermal noise) is reached, leaving a residual image of noise and low-level structure. Next, each CLEAN component is convolved with an “ideal” restoring function, such as a two-dimensional Gaussian, and added back into the residual image. The result is an image with approximately the noise and low-level errors of the original, but with the bright structure represented by the collection of Gaussian components. The algorithm works well for sources that are not very extended. Problems can become noticeable at short interferometer spacings, since extended structures have more power there. In addition, invisible distributions such as *CLEAN stripes* (modulations across the image at unsampled frequencies) can sometimes appear, particularly if the CLEANing is carried down into the thermal noise of the image.

Basic CLEAN can be excessively slow for large images. There are variants that speed up computing, reduce effects of gridding and aliasing, take into account varying beamshapes (e.g., in mosaics), and the like. One of the newest is *multiscale CLEAN* in which the CLEAN component size is allowed to vary, that is, the subtracted components can be simple but non-point-like objects convolved with the beam. For simple surveys of pointlike sources, imaging and deconvolution can easily be automated. However, for complex sources the user must typically adjust algorithm details (gains, regions to CLEAN, etc.) interactively to produce the best result for the individual problem. It is an area of active research and development to produce automatic imaging pipelines that can take data from a wide variety of observational conditions and produce publication-quality images most of the time.

The maximum entropy method (MEM) is a deconvolution method that finds a model image with maximum entropy that fits the data to within the noise level. Here, the term maximum entropy has led to some confusion. For this purpose, it is taken simply to mean maximizing one of a series of functions found to be useful. A measure in common use [14] is

$$\mathcal{H} = - \sum_k I_k \ln \frac{I_k}{M_k e}, \quad (11.11)$$

where the sum is over each of the  $k$  pixels in the image, and  $M_k$  is a “default” image that allows a priori information, such as a lower resolution image to be used as a starting point. Various algorithms are used to converge on an image that maximizes entropy within the noise constraint:

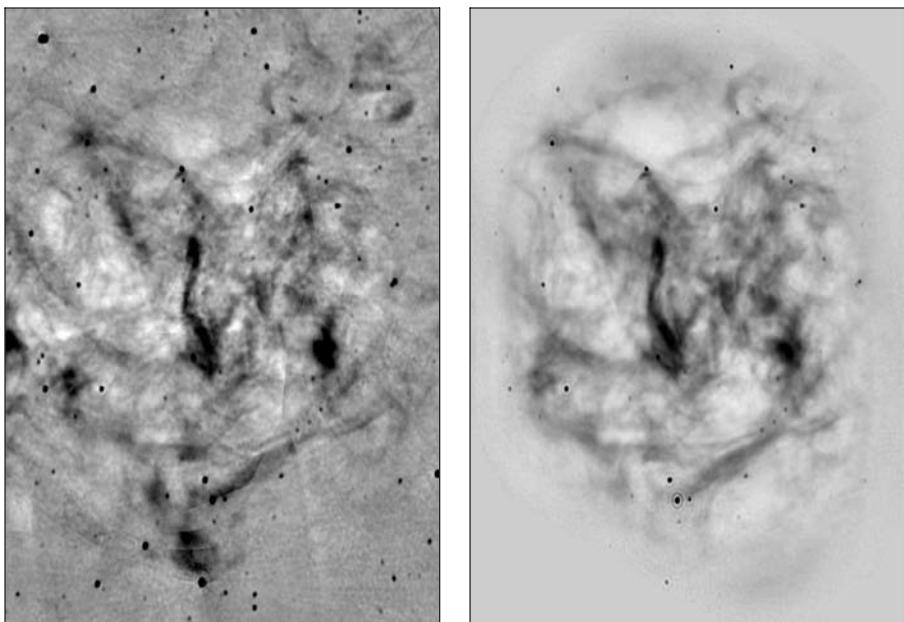
$$\chi^2 = \sum_k \frac{|\mathcal{V}_k^{\text{meas}} - \mathcal{V}_k^{\text{model}}|^2}{\sigma_k^2}. \quad (11.12)$$

The MEM does not typically constrain the total power in the image. Therefore, it is useful to provide a total estimate of the integrated flux from the object, if available, to avoid the image forming on a plateau of unphysical emission. This estimate could be from a single-dish map, which may also be the default image. Although MEM is often a better choice for imaging extended structure, it has the disadvantage that it is poor at handling unresolved features, and that the final resolution can vary across the image. The latter feature could make quantitative analysis of the point sources difficult. If a field has both point sources and extended structure, then CLEAN can be used to

remove the compact sources first. In cases where the  $uv$  data do not provide sufficient information reliably to image complex structure, and polarization data are available, it may be an advantage to jointly deconvolve maps of the four Stokes parameters, using an appropriate constraint [15]. The measure in (11.11), which enforces positivity, would not be appropriate for the Stokes  $U$  or  $V$  maps. However, other choices are available.

**11.2.4.4 Mosaicking** Sometimes the source to be imaged will be much larger than the primary beam of the antennas and thus not able to be imaged in a single pointing. In this case, a form of mosaicking is typically used: The antennas dwell on each of several pointings in turn. It is important to cycle through all the mosaic pointings often during a long observation (at a small penalty in time lost due to antenna drive and settling times). In this way, good  $uv$  coverage at each mosaic position can be maintained, by Earth rotation synthesis. The mosaic positions are spaced so as to allow sufficient overlap that the entire mosaic has approximately uniform sensitivity. Spacing the pointings at less than or equal to the Nyquist interval also aids in the recovery of extended structure if a joint deconvolution of the fields in the mosaic is performed [16].

According to the details of the source structure to be imaged, the deconvolution step might be performed on each field before the mosaic is made, or on a mosaic of “dirty” images. In the latter case, a joint deconvolution (typically MEM) allows the recovery of extended structure across many fields. Should the fields consist mainly of compact sources, it may be appropriate (and computationally less expensive) to mosaic images that have already been CLEANed. Figure 11.8 compares the benefits of the two



**Figure 11.8** MOST (left) and ATCA + Parkes Telescope images of a region of the Vela supernova remnant [15, 20]. The MOST image is a linear mosaic of several CLEAN deconvolved images, while the ATCA image was made with a joint MEM deconvolution of  $\sim 30$  fields. See text.

approaches for one complex field. In the observation with the Molonglo Observatory Synthesis Telescope (MOST), a linear mosaic was made of the images deconvolved with the CLEAN algorithm. The ATCA image used an MEM deconvolution over many fields; this algorithm allowed the inclusion of single-dish data to measure the total power in the image. The MOST image<sup>2</sup> has more artifacts in the extended structure since the individual images overlap with differing recovery of the extended structure. However, the point sources are better imaged in the MOST map since the CLEAN algorithm was used.

### **11.2.5 Calibration**

Calibration of radio interferometer data [17] makes use of both online and offline techniques, which here we define as those techniques that are carried out before and after the steps of correlation and integration. Online calibration tasks include measurement of and correction for antenna pointing, signal path delay, and receiver sensitivity (system temperature). Some measurements (e.g., round-trip phase of the analog signal path, atmospheric delay) may be made during the observation, even if they are applied later as corrections to the correlated data. Offline calibration makes use of observations of astronomical sources to correct for effects that vary more slowly than a correlator integration. Offline calibrations include determination of each antenna's position, gain, bandpass, and polarization leakage. Typically, they also include a check of the absolute flux density scale. Some measurements that need long integration times, such as determining the antenna position and bandpass, are obtained from the visibility data collected in long observations and applied online for future observations. The atmospheric contribution to gain may not be separable from the antenna gain, even though atmospheric effects may dominate, especially at higher frequencies.

The basic interferometric calibration that must be applied to every observation is the determination of the varying gain of the antenna signal path (including the atmosphere). To do this, calibration sources of known brightness and position (usually compact, extragalactic sources) are observed regularly. These allow the determination of empirical gain correction factors that compensate for gain variation that cannot easily be measured or corrected online. The time scale for reobservation depends on instrumental or atmospheric fluctuations. At centimeter and millimeter wavelengths this usually varies from a few to tens of minutes. Next-generation millimeter- and submillimeter-wave antennas are being designed to switch much more rapidly in order to track atmospheric fluctuations on time scales of a fraction of a minute. The calibration is usually an iterative process that relies on phase closure between each triangle of three antennas to arrive at gains that match the data to the model calibration source.

One technique commonly used where the field of view contains a bright source is *self-calibration* ([18] and references therein). In this process the structure of the bright source may not be known in advance. The procedure is similar to deconvolution in that the visibility data are interpreted using some plausible assumptions about the source structure [19]. To do this, both the visibility gains and the model source structure are varied until the Fourier transform of the model source agrees with the now-calibrated visibility data. The easiest implementation is to separate the modeling of the source

<sup>2</sup>Note that the MOST is a beamforming instrument rather than a correlation array. However, the deconvolution process is similar.

structure from the calibration. This is done iteratively by using a deconvolution step (CLEAN or MEM) to represent the data as a point source that can separately be calibrated using the standard phase closure technique. For self-calibration to work, there must be a reasonable a priori model of the source, which could be obtained by imaging with an initial calibration or from an image made previously.

### 11.2.6 Mitigation of Radio Frequency Interference

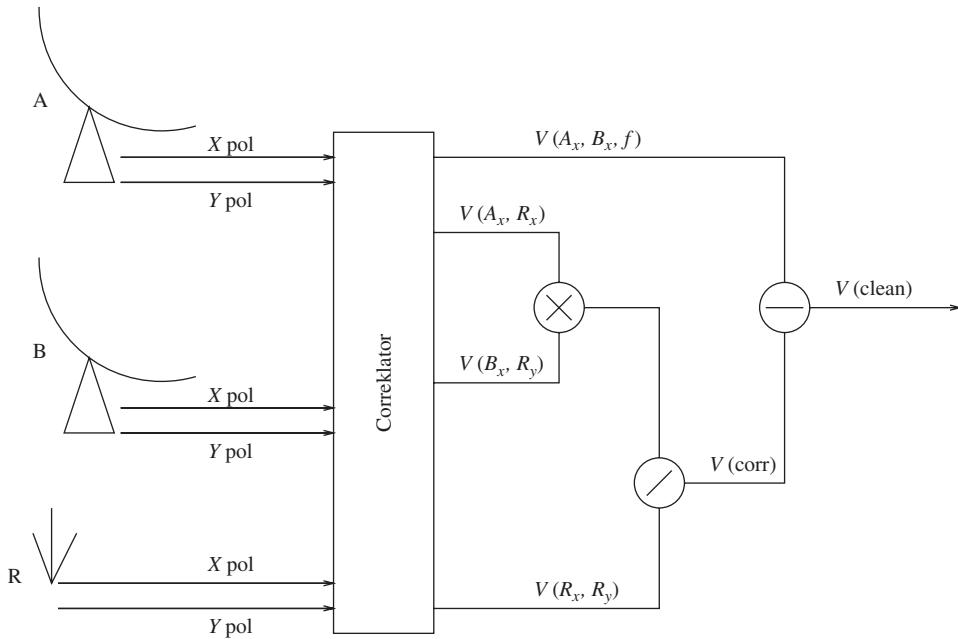
Radio frequency interference (RFI) is a significant consideration in the use of radio telescopes, and in the design of new systems. A recent review may be found in [21]. Although most countries have regulatory frequency allocations for radio astronomy and other passive services, these are typically very narrow in frequency. For example, the immediate vicinity of the 1427-MHz line of neutral hydrogen (HI), which dominates the universe, is well-protected in countries that are members of the International Telecommunications Union. Yet the frequency allocation is only 27 MHz wide. Meanwhile, extragalactic searches for this emission cover the entire spectrum down to  $\sim$ 100 MHz, while measurements of the broadband continuum emission (from synchrotron radiation) make use of bandwidths well in excess of 100 MHz. Several new low-frequency instruments (Section 11.2.7) are being designed to operate in the heavily used 30- to 200-MHz part of the spectrum.

Arrays are naturally resistant to RFI for two reasons. First, where the antennas are far apart, a local interferer may be strong at only one antenna, and therefore not correlated in the visibility. Nevertheless, the dynamic range at each antenna must be adequate to avoid saturation in the signal path. Second, the array is phased up on an astronomical source, and most interferers are terrestrial or elsewhere on the sky. Thus, they have a nonzero fringe rate and are “washed out” over the integration time. The antenna pattern (reflector shape) can also help reject out-of-field RFI. Several strategies are being used to mitigate the RFI effects: locating the arrays in radio-quiet places (western Australia, southern Africa), the use of high dynamic range receiving systems, and various RFI excision and cancellation schemes.

The simplest active strategy in common use is frequency and time excision. Since astronomers are imaging incoherent sources, they are generally concerned only with a statistical description of the source (e.g., the spatial power spectrum); any non-noise-like signal in the  $uv$  data is suspect.<sup>3</sup> This contrasts with communications systems, where the underlying signal is the relevant quantity. In many cases, manual data inspection reveals the RFI and the data are flagged to be ignored in subsequent processing. If only a few channels or time periods are flagged, the impacts on sensitivity and image quality are negligible. However, the process is labor intensive. Since future arrays will have many antennas, and often not store the raw visibilities, manual inspection is likely to become impractical. In well-characterized data, like those from an instrument devoted primarily to surveys at a fixed frequency, automated noise characterizations can do this flagging in real time. To date, this has not been widely implemented on correlation arrays, although there are some notable examples in beamforming interferometers such as the MOST (Section 11.2.7).

A relatively new development in RFI mitigation for correlator arrays is cancellation using a reference signal from the interferer [22]. This is equivalent to a real-time

<sup>3</sup>A notable exception is in searches for fast variable sources, such as pulsars, or in SETI (Search for Extraterrestrial Intelligence).



**Figure 11.9** An array-based postcorrelation adaptive filter. (From [21]; permission obtained from author.)

adaptive filter but may be applied *after correlation* to the time-averaged visibility data. Figure 11.9 depicts the arrangement. Two antennas, A and B, are pointed at the astronomical source, with a third antenna, R, directed at the interfering source. Consider the three cross products between the X polarization of each astronomical antenna and the two polarizations of the RFI receptor.  $\langle V_A V_B^* \rangle$  denotes the time-integrated correlation of signals  $V_A$  and  $V_B$ . It can be shown that a function of the cross correlations gives an estimate of the interfering signal, *as received by the astronomy antennas*. This correction,

$$V_{\text{corr}} = \frac{\langle V_{A_x} V_{R_x}^* \rangle \langle V_{R_y} V_{B_x}^* \rangle}{\langle V_{R_y} V_{R_x}^* \rangle}, \quad (11.13)$$

can easily be subtracted from the corrupted visibility to obtain a clean astronomical correlation. This method has a key advantage over its real-time kin: It works well at low interference-to-noise ratio (INR) levels. The reason is that a  $\sqrt{\text{bandwidth}} \times \text{time}$  reduction in noise takes place because the measurement is done after correlation and integration. This scheme has been tested at the Australia Telescope Compact Array on RFI from a television transmission tower. Substantial processing power will be needed to implement this scheme for large arrays with multiple sources. Observing with RFI is one of the main challenges for the next generation of radio telescopes.

### 11.2.7 Examples

In this section we describe briefly some significant correlation arrays in radio astronomy.

**11.2.7.1 Centimeter-Wave Arrays of Parabolic Reflectors** The Very Large Array (VLA) is an array of twenty-seven 25-m antennas in New Mexico that operates up to 43 GHz [23]. The antennas may be moved along three sets of railroad-type tracks in a Y configuration. Five scaled configurations are in common use, with the smallest having baselines from 35 m to 1 km and the largest having baselines from 680 m to 36 km. The VLA operates in several bands in the centimeter-wave range. The VLA is the most powerful of several similar general-purpose instruments, which also include the Australia Telescope Compact Array (six 22-m antennas) [24] and the Westerbork Synthesis Radio Telescope (fourteen 25-m antennas in the Netherlands) [25].

The Very Long Baseline Array (VLBA) is an array of ten 25-m antennas spread across the entire United States, from Hawaii to the U.S. Virgin Islands [26]. It operates at up to 96 GHz. Data are stored on media that are shipped to a correlator at a central location. This technique is known as Very Long Baseline Interferometry (VLBI). VLBI allows the highest resolution imaging of all astronomy. There are several international collaborations implementing VLBI between telescopes across the world and to satellites [27]. MERLIN is a connected array of up to 7 antennas with baselines up to 200 km, in the United Kingdom [28]. It has a unique imaging capability with resolution intermediate to the more compact connected arrays and the VLBI arrays. MERLIN is currently undergoing an upgrade (e-MERLIN) that will allow up to 2 GHz bandwidth to be correlated in real time. Next-generation VLBI arrays may be connected by the Internet.

The Allen Telescope Array (ATA; Fig. 11.10) is an array of 6-m offset Gregorian reflectors at Hat Creek, northern California [29]. The array operates in the range 0.5–10 GHz. The array has been constructed initially with 42 antennas; expansion



**Figure 11.10** The Allen Telescope Array consists presently of forty-two 6-m reflectors, with plans to increase to 350 antennas.

to 350 antennas is planned. As the first “large-*N*” array, the ATA is a precursor to the Square Kilometer Array (Section 11.4.2). Its instantaneous *uv* coverage is already superior to any comparable instrument and will improve with future expansion.

**11.2.7.2 Low-Frequency Arrays** A class of arrays operating at lower frequencies is currently under construction. These share the common characteristic that they will be constructed of dipoles or phased arrays of dipoles. All will need to overcome challenging problems in wide-field imaging and to cope with extreme radio-frequency interference and significant ionospheric propagation effects. The Low-Frequency Array (LOFAR), under construction in the Netherlands and adjoining countries, will operate in the range of 30–240 MHz, excluding the frequency modulation (FM) band [30]. The Murchison Widefield Array (MWA) will operate from 80 to 300 MHz in Western Australia [31]. The Long Wavelength Array (LWA) will operate from 10 to 88 MHz in New Mexico [32]. Each of these arrays is a large project in itself, with thousands of elements spread across hundreds of kilometers.

**11.2.7.3 Millimeter and Submillimeter Arrays** Several millimeter and submillimeter arrays are presently operating, and a large one, the Atacama Large Millimeter/submillimeter Array (ALMA; [33]) is under construction. When complete, ALMA, at a 5000-m altitude site in Chile, will consist of around 66 antennas (12 will be 7 m diameter, the rest will be 12 m diameter). The antennas will be capable of observing at wavelengths ranging from 0.3 to 3 mm, on baselines up to 18 km. All millimeter and submillimeter arrays consist of parabolic reflectors with very high surface accuracy. The Combined Array for Research in Millimeter-wave Astronomy (CARMA) is a millimeter-wave interferometer that operates in the 3-mm, 1-cm, and 1.3-mm bands in California [34]. It consists of 23 antennas ranging in diameter from 3.5 to 10.4 m on baselines up to 2 km. A comparable instrument, the Institut de Radio Astronomie Millimetrique Plateau de Bure Interferometer, in France, consists of six 15-m antennas that operate in the 1.3-, 2-, and 3-mm bands on baselines up to 760 m [35]. The Submillimeter Array is in Hawaii and consists of eight 6-m antennas operating on baselines up to 0.5 km and with receivers planned up to 900 GHz [36].

## 11.3 APERTURE PLANE PHASED ARRAYS

Phased arrays in radio astronomy have employed many of the same techniques used by other phased arrays. However, like radio astronomy correlation arrays, they have usually been sparse and extended in order to obtain high resolution, and to make use of aperture synthesis, including Earth rotation aperture synthesis (Section 11.2.1). Many phased array beams are typically produced simultaneously in order to image a wide field of view. To reduce side lobes that would limit the dynamic range, a regularly spaced phased array must have many more elements than a correlation array.

An interesting example of the class is the Molonglo Observatory Synthesis Telescope (MOST; Fig. 11.11) [37]. MOST is an east–west array of 7744 elements on a line feed in a cylindrical paraboloid reflector. In the north–south direction pointing and directivity are achieved with the paraboloid. In the east–west direction, the MOST steers two 778-m phased arrays (“arms”). Each arm has 3872 circularly polarized dipoles at the focus. These are phased up in waveguides into 176 sections. Low-noise amplifiers



**Figure 11.11** The Molonglo Observatory Synthesis Telescope is a cylindrical paraboloid  $1.6 \text{ km} \times 12 \text{ m}$ , with a phased array line feed consisting of 7744 elements at the focus.

(LNAs) follow the line feed at each section, after which each set of 4 sections is phased into a single signal using a microstrip beamformer. The 44 signals from each arm are phased up in a resistive matrix with many fixed phase gradients to form 64 fan beams. Each fan beam from the west array is then multiplied with the corresponding fan beam from the east array. This produces a set of sixty-four  $44' \times 2^\circ$  fan beams. A differential phase gradient may be rapidly altered to timeshare the 64 beams into 448 beams covering  $2^\circ$ . Thus an image of diameter  $2^\circ$  may be formed from a 12-h observation. Presently, the system is being altered to correlate outputs from the individual LNAs at a wider bandwidth. This is possible as the bandwidths of the line feeds and LNAs are relatively wide ( $\sim 100 \text{ MHz}$ ). The new system will provide data that are easier to calibrate, since slowly varying effects at the level of individual sections may be measured and corrected after the observation.

Some correlation arrays may be used as phased arrays. This is usually done where sensitivity in one or a few interferometer beam areas is the main consideration, and where a full-bandwidth data stream with the sensitivity of the entire array is desired. For example a phased array of reflectors might be used as part of a VLBI experiment (Section 11.2.7) or as an input to a specialized processing system for measuring or searching for transient or time-variable phenomena, such as a system to search for extraterrestrial intelligence.

## 11.4 FUTURE DIRECTIONS

### 11.4.1 Focal Plane Arrays

The term *focal plane array* can describe any array of detectors at the focal plane of a telescope, including photographic plates or charge coupled devices (CCDs) in

optical astronomy, bolometers at millimeter and submillimeter wavelengths, or arrays of heterodyne receivers at the focal plane of large centimeter-wave antennas. These all operate essentially as independent detectors: No array processing is required. An advantage of heterodyne arrays over bolometric arrays is the high-resolution spectroscopy possible; it often comes at the expense of overall bandwidth. Recently, radio astronomers have begun to consider interlinked arrays of elements to improve the imaging and survey efficiency of radio telescopes.

The Australian Square Kilometer Array Pathfinder (ASKAP; [38]) is currently being designed as a major step toward the Square Kilometer Array (Section 11.4.2). It will consist of thirty 12-m parabolic antennas having baselines up to 2 km, for a total collecting area of more than  $3000\text{ m}^2$ . The goal of the project is to enable imaging in fields of view even larger than the primary beam of the antenna. It will operate in the range of 700–1800 MHz. The plan includes a focal plane phased array with 30 beams and a field of view of  $30\text{ deg}^2$  at 1.8 GHz. The development and use of the focal plane array (FPA) will likely be the main technical challenge of the project. The focal plane array is specified to have efficiency after beamforming of 60–75%, system temperature after beamforming (including LNA contribution) of 50 K. In addition to the challenges of using a FPA at wide bandwidth, discussed above, there are likely to be new calibration and imaging procedures needed to allow interferometry with phase array feeds.

#### 11.4.2 Square Kilometer Array

The Square Kilometer Array (SKA) [39, 40] is planned to be the principal next-generation radio telescope. Spread over thousands of kilometers, with cost an order of magnitude less than existing instruments, this array will resolve milliarcsecond structures. Yet it will retain half its collecting area (i.e., many short baselines) in the central 5 km, for excellent sensitivity to extended structures. Its specification calls for a sensitivity orders of magnitude above current instruments over a frequency range from 70 MHz to  $\sim 25$  GHz. This is expressed as  $A_e/T_{\text{sys}} = 10,000\text{ m}^2\text{ K}^{-1}$  [41], where  $A_e$  is the effective area of the antennas and  $T_{\text{sys}}$  is the system temperature. For system temperatures in the region of 50 K, this corresponds to approximately a square kilometer of physical collecting area. Other key specifications include the survey performance (“speed”) proportional to  $(A_e/T_{\text{sys}})^2 \times$  field of view (FOV). To satisfy these specifications across the desired frequency range, it may be necessary to use several technologies. At frequencies above about 500 MHz, parabolic reflectors will suffice, with phased array feeds becoming useful to increase the survey performance. At the lowest frequencies aperture arrays of fixed antenna elements may be practical, allowing instantaneous all-sky coverage limited only by postprocessing. Primarily motivated by a desire to minimize RFI, the international group designing the SKA has selected southern Africa and western Australia as the two candidate sites. Nevertheless, careful attention is being given to incorporating techniques for RFI mitigation into the design.

To cope with the enormous data volume (potentially millions of cross correlations at gigahertz bandwidth), it will be necessary to combine techniques learned from correlation arrays with beamforming techniques. Current designs incorporate a limited number (hundreds) of clusters of collecting elements. Each cluster would have a broadband beamformer with multiple phased outputs within the primary beam of the individual

element. Some of the beam thus formed could be further combined within a central beamformer, for example, to image time-varying but unresolved radio sources. Others would be passed to a conventional correlator for imaging the entire field of view of the cluster. The combination of techniques (and consequent processing requirements) will depend on the details of individual science projects. Careful attention will be paid to the antenna configuration design within each cluster, and of the array overall. However, the problems are solved, the SKA is likely to push the limits of techniques developed for beamformers and for correlation arrays.

### 11.4.3 Array Processing at Optical and Infrared Wavelengths

Optical aperture synthesis, which began with the Michelson interferometer, has been limited by the available electronics. In the meantime, the main techniques have been developed for radio astronomy. However, the techniques discussed in Section 11.2 are equally applicable at optical wavelengths. Labeyrie et al. [42] give a comprehensive overview of optical interferometry in astronomy.

Most optical interferometry to date has used optical beamsplitters to direct beams toward fringe-measuring devices. Delay lines at the optical frequency enable phasing-up on astronomical sources. The sensitivity is limited by the need to split the beams before any amplification takes place. There are several multielement optical interferometers in operation [42]. An alternative technique is to produce an interferometer that operates just like a radio interferometer. This has been done in the mid-infrared, with the Berkeley Infrared Spatial Interferometer [43], which uses lasers locked together on a single LO. Downconversion, amplification, correlation, and imaging proceed as for a radio correlation interferometer. This method has two advantages. First, heterodyne detection allows the delay lines to be much less precisely made. The path lengths have to be equalized only to the size set by the length of the coherent wavegroup,  $c/\delta\nu$ . For a bandwidth of 1 GHz, this is 30 cm. Second, high-resolution spectroscopy is enabled since radio-frequency filters can project arbitrarily narrow bands more conveniently than at optical wavelengths. However, the quantum limit is significant for these bandwidths at optical/infrared wavelengths. The uncertainty principle states that the uncertainties in the phase and intensity are related. When only a few ( $n$ ) photons are received during the integration period, the uncertainty in the phase ( $\phi$ ) will be high, that is,  $\delta\phi \cdot \delta n \geq \frac{1}{2}$ . To date, this has limited heterodyne interferometry to the brightest stars.

## 11.5 CONCLUSION

This chapter has reviewed techniques that were largely developed for, and applied to, arrays of up to a few dozen antennas. Such instruments often require sophisticated data processing techniques and hardware; but these aspects do not usually dominate the construction cost. Science cases for next-generation large radio telescope arrays require large fields of view, high sensitivity, and excellent imaging fidelity. Optimizing this combination for minimal cost leads to *large-N* arrays, with many relatively small antennas. An example at one extreme of the radio spectrum is the Atacama Large Millimeter/submillimeter Array (under construction), with its 66 antennas optimized for high sensitivity and mosaicing. At lower frequencies, the Square Kilometer Array will

have thousands of antennas. In the lower frequency range, the sensitivity requirement translates to an additional requirement to have sufficient dynamic range that confusing background sources can be separated from the sources of interest.

In the large- $N$  regime, data distribution and processing costs become significant. Optimization of the antenna size and placement depends on a good understanding of the cost, which in turn depends on observing and calibration strategies. Hence it becomes important to understand these costs early in the design process and include them in the design equation. A significant challenge for radio astronomy with new arrays thus is understanding enough about the necessary calibration techniques for instrument design, often before the calibration algorithms can even be implemented.

Perhaps the most significant improvement in radio astronomy array calibration is likely to be better modeling of the primary antenna element. This will add complexity to the computing. At present, each antenna in the array is usually assumed to have an identical antenna pattern, typically constant with time, and often idealized (e.g., as a truncated Gaussian). The side-lobe pattern has generally not been modeled. In the regime of high dynamic range this will not be adequate, especially for polarimetric imaging. Furthermore, when focal plane arrays are placed at each antenna to optimize instruments for wide-field imaging, the antenna pattern will vary with each feed. A detailed antenna model will be part of routine data analysis.

Meanwhile, radio astronomy signals will continue to be affected by unwanted interference. Even at the remotest sites on Earth, satellite and airplane emissions contaminate the data. As communication signals become spread spectrum, they will be more easily confused with the noiselike signals of radio astronomy. Corruption of these noiselike signals will be a difficult form of interference to remove. Although new techniques will be developed to address this challenge, the ultimate instruments of the future may have to be built on the back of the moon or elsewhere so that they are shielded from the Earth's emissions.

## REFERENCES

1. M. Ryle, 1962, *Nature*, 194, 517
2. A. R. Thompson, J. M., Moran, and G. W., Swenson, Jr., *Interferometry and Synthesis in Radio Astronomy*, 2nd ed., Wiley, New York, 2001.
3. M. Born and E. Wolf, *Principles of Optics*, 7th ed., Cambridge University Press, Cambridge, 1999.
4. T. S. Bastian, *Adv. Space Res.*, vol. 32, p. 2705, 2003.
5. E. Keto, *Astrophys. J.*, vol. 475, p. 843, 1997.
6. L. Kogan, "Optimization of an array configuration minimizing side lobes," MMA memo 171, available: <http://www.almal.nrao.edu/memos/>, 1997.
7. L. Kogan, *Publ. Astron. Soc. Pacific*, vol. 111, p. 510, 1999.
8. F. Boone, *Astron. Astrophys.*, vol. 377, p. 368, 2001.
9. F. Boone, *Astron. Astrophys.*, vol. 386, p. 1160, 2002.
10. D. Woody, "Interferometer array point spread functions II. Evaluation and optimization," ALMA memo 390, available: <http://www.almal.nrao.edu/memos/>, 2001.
11. D. S. Briggs, F. R. Schwab, and R. A. Sramek, in *Synthesis Imaging in Radio Astronomy*, Vol. II, Astronomical Society of the Pacific, San Francisco, 1999, p. 127.
12. J. A. Högbom, *Astron. Astrophys. Suppl. Ser.*, vol. 15, p. 417, 1974.

13. R. J. Sault and T. A. Oosterloo, in *Review of Radio Science 1993-1996*, W. R. Stone (Ed.), Oxford University Press, Oxford, 1996, p. 883.
14. T. Cornwell, R. Braun, and D. S. Briggs, in *Synthesis Imaging in Radio Astronomy*, Vol. II, G. B. Taylor, C. L. Carilli, and R. A. Perley (Eds.), Astronomical Society of the Pacific, San Francisco, 1999, p. 151.
15. R. J. Sault, D. C.-J. Bock, and A. R. Duncan, *Astron. Astrophys. Suppl. Ser.*, vol. 139, p. 387, 1999.
16. T. J. Cornwell, *Astron. Astrophys.*, vol. 202, p. 316, 1988.
17. E. B. Fomalont and R. A. Perley, in *Synthesis Imaging in Radio Astronomy*, Vol. II, G. B. Taylor, C. L. Carilli, and R. A. Perley (Eds.), Astronomical Society of the Pacific, San Francisco, 1999, p. 79.
18. T. J. Pearson and A. C. S. Readhead, *Annu. Rev. Astron. Astrophys.*, vol. 22, p. 97, 1984.
19. T. Cornwell, and E. B. Fomalont, in *Synthesis Imaging in Radio Astronomy*, Vol. II, G. B. Taylor, C. L. Carilli, and R. A. Perley (Eds.), Astronomical Society of the Pacific, San Francisco, p. 187, 1999.
20. D. C.-J. Bock, A. J. Turtle, and A. J. Green, *Astron. J.*, vol. 116, p. 1886, 1998.
21. M. Kesteven, *Radio Sci. Bull.*, vol. 322, p. 9, 2007.
22. F. H. Briggs, J. F. Bell, and M. J. Kesteven, *Astron. J.*, vol. 120, p. 3351, 2000.
23. A. R. Thompson, B. G. Clark, C. M. Wade, and P. J. Napier, *Astrophys. J. Suppl. Ser.*, vol. 44, p. 151, 1980.
24. G. J. Nelson, *J. Electric. and Electron. Eng. Australia*, vol. 12, p. 112, 1992.
25. J. W. M. Baars, J. F. van der Brugge, J. L. Casse, J. P. Hamaker, L. H. Sondaar, J. J. Visser, and K. J. Wellington, *IEEE Proc.*, vol. 61, p. 1258, 1973.
26. P. J. Napier, D. S. Bagri, B. G. Clark, A. E. E. Rogers, J. D. Romney, A. R. Thompson, and R. C. Walker, *IEEE Proc.*, vol. 82, p. 658, 1994.
27. R. C. Walker, in *Synthesis Imaging in Radio Astronomy*, Vol. II, G. B. Taylor, C. L. Carilli, and R. A. Perley (Eds.), Astronomical Society of the Pacific, San Francisco, 1999, p. 433.
28. S. T. Garrington et al., in *Ground-Based Telescopes*, J. M., Oschmann, Jr. (Ed.), *Proc. SPIE*, vol. 5489, p. 332, 2004.
29. D. Deboer et al., *Exp. Astron.*, vol. 17, p. 19, 2004.
30. H. D. Falcke et al., *Highlights Astron.*, vol. 14, p. 386, 2007.
31. J. D. Bowman et al., *Astron. J.*, vol. 133, p. 1505, 2007.
32. G. B. Taylor, *Highlights Astron.*, vol. 14, p. 388, 2007.
33. M. Tarenghi, *Astrophys. Space Sci.*, vol. 313, p. 1, 2008.
34. D. C.-J. Bock et al., in *Ground-Based and Airborne Telescopes*, L. M. Stepp (Ed.), *Proc. SPIE*, vol. 6267, p. 626713–1, 2006.
35. S. Guilloteau et al., *Astron. Astrophys.*, vol. 262, p. 624, 1992.
36. P. T. P. Ho, J. M. Moran, and K. Y. Lo, *Astrophys. J.*, vol. 616, p. L1, 2004.
37. B. Y. Mills, *Proc. Astron. Soc. Australia*, vol. 4, p. 156, 1981.
38. S. Johnston et al., *Publ. Astron. Soc. Australia*, vol. 24, p. 174, 2007.
39. C. Carilli and S. Rawlings, *Science with the Square Kilometre Array*, Elsevier, Amsterdam, 2004.
40. P. J. Hall, *The Square Kilometre Array: An Engineering Perspective*, Springer, Dordrecht, 2005.
41. C. Schilizzi et al., “Preliminary specifications for the SKA,” available: <http://www.skatelescope.org>, 2007.
42. A. Labeyrie, S. G. Lipson, and P. Nisenson, *An Introduction to Optical Stellar Interferometry*, Cambridge University Press, Cambridge, 2006.
43. D. D. S. Hale et al., *Astrophys. J.*, vol. 537, p. 998, 2000.

## CHAPTER 12

---

# Digital 3D/4D Ultrasound Imaging Array

Stergios Stergiopoulos  
DRDC Toronto, Toronto, Ontario

### 12.1 BACKGROUND

The fully digital three-dimensional (3D)/(4D: 3D + time) ultrasound system technology of this chapter consists of an advanced beamforming structure that allows the implementation of adaptive and synthetic aperture signal processing techniques in ultrasound systems deploying multidimensional arrays of sensors. The aim with this fully digital ultrasound beamformer is to address the fundamental image resolution problems of current ultrasound systems and to provide suggestions for its implementation into existing 2D and/or 3D ultrasound systems as well as develop a complete stand-alone 3D ultrasound solution. This development has received grant support from the Defence R&D Canada (DRDC) and from the European Commission IST Program (i.e., ADUMS project: EC-IST-2001-34088).

To fully exploit the advantages of the present fully digital adaptive ultrasound technology, however, its implementation in a commercial ultrasound system requires that the system has a fully digital design configuration consisting of analog-to-digital converters (A/DC) and digital-to-analog converters (D/AC) peripherals that have the capability to fully digitize the ultrasound probe time series, to optimally shape the transmitted ultrasound pulses through a D/A peripheral and to integrate linear and/or planar phase array ultrasound probes.

Thus, the digital ultrasound beamforming technology of this chapter can replace the conventional (i.e., time-delay) beamforming structure of ultrasound systems with an adaptive beamforming processing configuration. The results of this development demonstrate that adaptive beamformers improve significantly (at very low cost) the image resolution capabilities of an ultrasound imaging system by providing a performance improvement equivalent to a deployed ultrasound probe with double aperture size. Furthermore, the portability and the low-cost characteristics of the present 3D adaptive ultrasound technology can offer the options to medical practitioners and family physicians to have access of diagnostic imaging systems readily available on a daily basis. As a result, a fully digital ultrasound technology can revise the signal processing

configuration of ultrasound devices to move them away from the traditional hardware and implementation software requirements.

In summary, implementation of an adaptive beamformer is a software installation on a personal computer (PC)-based ultrasound computing architecture with sufficient throughput for 3D and 4D ultrasound image processing. Moreover, the PC-based ultrasound computing architecture, which is introduced in this chapter, can accommodate the processing requirements of the “traditional” linear array 2D scans as well as the advanced matrix-arrays performing volumetric scans.

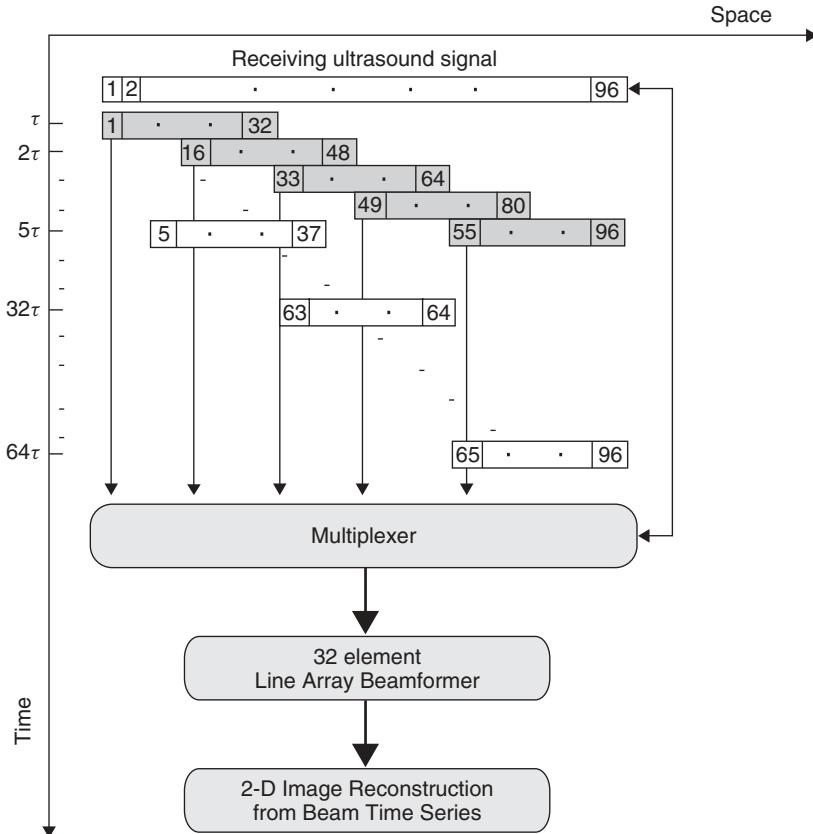
### **12.1.1 Limitations of 2D Ultrasound Imaging Technology**

It has been well established [1–3] that the existing limitations of medical ultrasound imaging systems in poor image resolution is the result of the very small size of deployed arrays of sensors and the distortion effects by the influence of the human body’s nonlinear propagation characteristics. In particular, some of the limitations (e.g., resolution) of ultrasound imaging are related to the fundamental physical aspects of ultrasound transducers and the interaction of ultrasound with tissues (e.g., aberration effects). In addition to these fundamental limitations are restrictions related to the display of the ultrasound images in an efficient manner allowing the physician to extract relevant information accurately and reproducibly. Specifically, the current state of the art of 3D ultrasound imaging technology attempts to address the following limitations:

- Conventional ultrasound images are 2D, hence, the physician must mentally integrate multiple images to develop a 3D impression of the anatomy/pathology during procedures. This practice is time consuming, inefficient, and requires a highly skilled operator, all of which can potentially lead to incorrect diagnostic and therapeutic decisions.
- Often the physician requires accurate estimation of tumor and organ volume. The variability in ultrasound imaging and volume measurements using a conventional 2D technique is high because current ultrasound volume measurement techniques assume an idealized elliptical shape and use only simple measures of the width in two views [4, 5].
- It is difficult to localize the thin 2D ultrasound image plane in the organ and difficult to reproduce a particular image location at a later time, making 2D ultrasound a limited imaging modality for monitoring of disease progression/regression and follow-up patient studies.

**12.1.1.1 Limitations of Current Beamforming Structure of Ultrasound Imaging Systems** A state-of-the-art transducer array of a commercial ultrasound system is either linear or curvilinear depending on the application; and for each deployed array (i.e., ultrasound probe), the number of transducers is in the range of 96–256 elements. However, only a small number of transducers of a given array are beamformed coherently to reconstruct a 2D tomography image of interest.

A typical number of transducers that may be included in the beamforming structure of an ultrasound system is in the range of 32–128 elements. Thus, the array gain of the beamformer may be smaller by approximately  $10 \times \log_{10}(3) \approx 5$  dB than that available by the deployed ultrasound probe. Figure 12.1 illustrates the basic processing steps associated with the ultrasound beamforming process. For the sake of simplicity



**Figure 12.1** Typical beamformers for linear arrays of ultrasound systems. The shaded subapertures indicate the selected spatial locations for the formation of synthetic aperture according to ETAM algorithm.

and without any loss of generality, the array in Figure 12.1 is considered to be linear with 96 elements that can be used to transmit and receive the ultrasound energy. As shown in Figure 12.1, for each active transmission, the ultrasound beamformer processes coherently the received signal of 32 elements only, which is a subaperture of the 96-element deployed array. The active transmission takes place approximately every  $\tau = 0.3$  ms, depending on the desired penetration depth in the body. The beam steering process is at the broadside.

When an active transmission is completed, the receiving 32-element subaperture is shifted to the left by one element, as shown in Figure 12.1. Thus, to make use of all the 96 elements of the deployed probe, the 32-element beamforming process is repeated 64 times, generating 64 broadside beams. In other words, it takes approximately  $64 \times 0.3$  ms  $\approx 20$  ms to reconstruct a 2D tomography image of interest. As a result, the resolution characteristics of the reconstructed image are defined by the array gain of the beamformer and the temporal sampling of the beam time series, for analog or digital beamformers, respectively. In the specific case of Figure 12.1, the pixel resolution along the horizontal  $x$  axis of a reconstructed tomography image is defined by the angular resolution along azimuth of the 32-element beamformer. This resolution is

usually being improved by means of interpolation, which defines the basic difference between beamformers of different ultrasound systems.

The pixel resolution along the vertical  $y$  axis of the reconstructed image is defined by the temporal sampling rate, which is always very high, and it is not a major concern in ultrasound system applications. Thus, improvements of image resolution in ultrasound system applications requires mainly higher angular resolution or very narrow beamwidth, which means longer arrays and longer subapertures for the beamforming process with consequent technical and operational implications of hardware complexity and higher system manufacturing cost.

The main advantages of this simplified beamforming structure are the following:

- A broadside beamformer allows the use of frequency regimes that are higher than the corresponding spatial-aliasing frequency of the sensor spacing of the ultrasound probe. This results from the fact that side-lobe artifacts due to spatial aliasing are insignificant for beams with broadside beam steering. Furthermore, this kind of simplicity in the analog beamforming structure allows for analog high-speed hardware design for the beamformers. Then, the A/DC peripherals are used to digitize the beam time series.
- The advantage (i.e., suppression of spatial-aliasing artifacts) provided by the broadside beam-steering process has been used effectively by various types of illumination techniques using higher order harmonics to achieve deeper penetration with corresponding higher image resolution along the temporal axis.
- The field of view of the probe may be wider than the aperture size required by the broadside beamformer. This approach eliminates the hardware complexity of an A/DC peripheral and it does not require digitization of the probe time series. Instead, it uses a multiplexer combined with an analog beamformer to control the data acquisition process for a probe with a larger number of sensors (e.g., 96 channels in Fig. 12.1) than those being used by the broadside focus beamformer (e.g., 32 channels in Fig. 12.1).

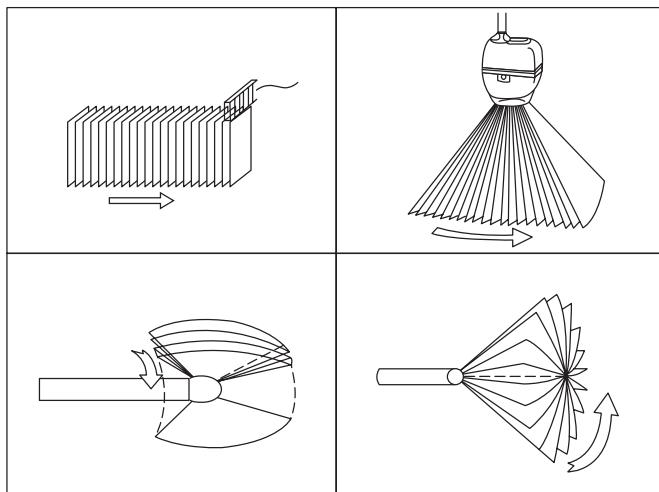
Until recently, the above beamforming concept (see Fig. 12.1) has served well the ultrasound system requirements by providing practical alternatives to technical problems that were due mainly to limitations in the maximum number of channels deployed by A/DC units and the limited capabilities of computing architectures. Presently, these kind of technology limitations do not exist. Thus, new advanced technology options have become available to exploit the vast experience from phase array beamformers that have been advanced by the sonar and radar research communities [1, 6]. In fact, the introduction of linear phase array probes for cardiac applications is the first successful step toward this direction.

The use, however, of linear arrays introduces another major problem in terms of false targets, a problem that has been identified by both the ultrasound and sonar researchers using towed sonar arrays [1, 6, 7]. In particular, a linear array provides angular resolution within the tomography plane (B scan) that the beam steering is formed. The angular resolution, however, of the beam-steering vectors of linear arrays is omnidirectional in the plane perpendicular to the B-scan plane. Thus, reflections from surrounding organs cannot be spatially resolved by the steered beams of a line array; and they appear as false targets in towed array sonars or false components of a reconstructed image by a linear ultrasound probe.

To address the problem of false components in the reconstructed image, the 1.5D and 1.75D ultrasound array probes have been introduced that consist of linear arrays stacked as partially planar arrays. In particular, the GE 1.75D ultrasound array probe consists of 8 linear arrays with 128 sensors each and with 0.2-mm sensor spacing. The linear array spacing is 1.5 mm. Thus, the steered beams are three dimensional and have the property to resolve the angular components of ultrasound-reflected signals along azimuth and elevation. Although the 3D beams of 1.75D arrays may be viewed as the first step for 3D ultrasound imaging, they do not have sufficient angular resolution capabilities along elevation to generate 3D ultrasound volumes. The 3D beamforming structure and the relevant 3D ultrasound experimental system development that is based on a  $16 \times 16$  planar array probe, which is presented in this chapter, attempts to address the above limitations. At this point, however, it is considered appropriate to briefly review the current state of the art in 3D ultrasound technology.

**12.1.1.2 Current Technology Concept of 3D Visualization Methods for Ultrasound Systems** Current 3D ultrasound imaging systems have three components: image acquisition, reconstruction of the 3D image, and display [1, 4, 5, 8–14]. The first component is crucial in ensuring that optimal image quality is achieved. In producing a 3D image, the conventional line array transducer is moved over the anatomy while 2D images are digitized and stored in a microcomputer, as shown in Figure 12.2. To reconstruct the 3D geometry without geometric distortion, the relative position and angulation of the acquired 2D images must be known accurately. Over the years there are numerous developments and evaluating techniques for obtaining 3D ultrasound images using the following two approaches: mechanical scanning and freehand scanning [1, 4, 5, 8–14].

**Mechanical Scanning** Based on earlier work, Fraunhofer has developed systems [1] in which the ultrasound transducer is mounted in a special assembly, which can be driven by a motor to move in a linear fashion over the skin or tilted in equal angular



**Figure 12.2** Mechanical scanning of ultrasound probes for image acquisition of 2D B scans to obtain 3D ultrasound images through volume rendering.

steps. The movement can be continuous, cardiac, and/or respiratory [4, 9]. In addition, the spatial-sampling frequency of the image acquisition can be adjusted based on the elevational resolution of the transducer and the depth of the region of interest. For linear scanning, they collect 140 images ( $336 \times 352$  pixels each) at 0.5-mm intervals in a time that depends on the ultrasound machine frame rate and whether cardiac gating is used. All scanning parameters can be adjusted depending on the experiment and type of acquisition. For example, for 3D B mode, they typically use 2 or 3 focal zones resulting in about 15 frames/s and a total 3D scanning time of 9 s for 140 images.

**Freehand Scanning** Although the mechanical scanning approach produces accurate 3D ultrasound images, the mechanical assembly is bulky and not convenient for the operator. In addition, the mechanical constraint does not permit its use in imaging larger structures such as the liver or fetus. An alternative freehand technique has been proposed [1] that maintains the flexibility of the 2D exam yet produces a 3D image. We are investigating a freehand scanning system in which a magnetic positioning and orientation measurement (POM) device is mounted on the transducer [10–14]. To produce a 3D image, the operator manually moves the handheld transducer, while the POM device transfers the position and orientation coordinates of the transducer to a microcomputer. At the same time, 2D images are digitized by the same computer and associated with the appropriate coordinates. After the necessary number of 2D images are acquired (typically 60–160), the computer reconstructs the 3D image. Care is taken to scan the patient sufficiently slowly so that the region of interest is scanned with no gaps. Typically, the scan lasts 4–11 s while the patient holds his or her breath. Although this technique does produce useful images, it still suffers from major limitations that precludes its use for general diagnostic procedures. Most importantly, the manual scanning of the 3D space with a linear array does not eliminate the false components of the reconstructed B-scan images that were discussed in the previous section.

## 12.2 NEXT-GENERATION 3D/4D ULTRASOUND IMAGING TECHNOLOGY

The experimental 3D/4D ultrasound developments, discussed in this chapter, demonstrate an imaging technology that can lead to next-generation high-resolution diagnostic ultrasound imaging systems. The main components of this technology include:

- Synthetic aperture processing to accommodate digitization requirements for the sensor time series of large-size 2D phase array probes for 3D phase array beam-forming
- 3D adaptive beamforming for the full aperture of the deployed probe to effectively maximize the available array gain and improve angular resolution
- PC-based computing architecture capable to accommodate the computationally intensive signal processing and data acquisition requirements of fully digital 3D/4D ultrasound imaging systems deploying planar arrays
- Experimental fully digital ultrasound system with a  $16 \times 16$  sensor phase planar array with uniform sensor spacing
- Integration of Fraunhofer's 3D and 4D visualization schemes with the image reconstruction process of the 3D ultrasound beamformer

The following sections provide technical details for the above components.

### 12.2.1 Synthetic Aperture Processing for Digitizing Large-Size Planar Arrays

Integration of a synthetic aperture processing in the data acquisition structure of a fully digital ultrasound system can accommodate the highly demanding requirements to digitize the channels of a large-size planar array by using a single A/DC peripheral, which may have fewer A/D channels than those of the planar array probe. More specifically, let us assume that a planar array probe consists of  $N \times N$  sensors and the A/DC peripheral has the capability to digitize  $2N$  channels. Then, a digitization process of the  $N \times N$  channels of the planar array probe requires the use of a multiplexer for  $N/2$  successive acquisitions [i.e.,  $N \times N/(2N) = N/2$ ] of equal size (i.e.,  $2N$ -channels) of  $N/2$  subapertures of the planar array by the  $2N$ -channel A/DC peripheral. Then each set of  $2N$  digitized time series of each subaperture will be integrated with the remaining digitized time series of the  $N/2$  subapertures to form a complete set of digitized  $N \times N$  sensor time series representing a snapshot of the illuminated ultrasound field. However, this successive digitization process of the  $N \times N$  channels of the planar array by a  $2N$ -channel A/DC peripheral may not be fast enough to ensure that there are no motion artifacts between the successively digitized subapertures. To minimize potential motion artifacts during the digitization process of large-size planar arrays, the following synthetic aperture process is recommended.

Shown in Figure 12.1 is the proposed experimental implementation of a synthetic aperture algorithm [15] (i.e., ETAM) for ultrasound applications in terms of the subaperture line array size and sensor positions as a function of time and space. Between two successive positions of the 32-sensor subaperture there are a number of sensor pairs of space samples of the acoustic field that have the same spatial information, their difference being a phase factor [15] related to the time delay these measurements were taken. The optimum overlap size, which is related to the variance of the phase correction estimates, has been shown [15] to be equal to the half size of the deployed subaperture. For the particular example of Figure 12.1, the spatial overlap size will be 16 sensors. Thus, by cross correlating the 16-sensor pairs of the sensor time series that overlap, the desired phase correction factor is derived, which compensates for the time delay between these measurements and the phase fluctuations caused by the variability and nonisotropic propagation characteristics of the human body; this is called the overlap correlator. The key parameters in the ETAM algorithm is the time increment  $\tau$  between two successive sets of measurements. This may be the interval of 0.3 ms between two active ultrasound transmissions. Then, the total number of sets of measurements required by the 32-sensor subaperture to achieve an extended aperture size equal to the deployed array (i.e., 96-sensor array) is five.

Thus, if we consider the subaperture acquisition process in Figure 12.1, the proposed synthetic aperture processing will coherently synthesize the spatial measurements derived from the 32-element subapertures of the ultrasound receiving array into a longer aperture equivalent to the 96-sensor deployed array using only 5 subaperture measurements instead of 64. In this way, the required hardware modifications of an ultrasound system will be minimized since the A/DC will remain the same. Moreover, the time required to reconstruct a tomography image will be reduced from the current 20-ms time interval to  $5 \times 0.3$  ms  $\approx 1.5$  ms. In parallel to this improvement, there will be an increase in the beamformer's array gain by  $10 \times \log_{10}(3) \approx 5$  dB, with improvement also in angular resolution of the ultrasound beamforming structure by a factor of 3 (i.e.,  $\Delta\theta = \lambda/L$  for the synthetic aperture  $\Delta\theta = \lambda/(3L)$ , as described in [3]).

However, the experience gained from the development of the experimental fully digital ultrasound design, reported in this chapter, suggests the following: When the allocated period for the digitization process of the subapertures of a large-size planar array is very fast and of the order of a few milliseconds, then there is no need for synthetic aperture processing, and the best approach is to piece together the subapertures in order to form the fully populated physical aperture for beamforming and to treat the digitized data of the subapertures as samples that have been acquired instantly.

In particular, during the experimental ultrasound development reported in this chapter, a triggering mechanism was used to attempt to keep a constant time delay between the subaperture firings and a constant period for the digitization process of subapertures. This also ensured that there was a high correlation between the overlapping segments of successive subapertures. If the triggering mechanism can be designed to be very accurate—meaning the correlation coefficient becomes exactly one, then there is no need to perform the correlation—phase correction process of the ETAM algorithm because every subaperture is already coherent with respect to the first subaperture. Furthermore since all of the subapertures are coherent, and no correction is needed, there is no need to accumulate overlapping subapertures. This argument is supported also by experimental studies [2] that have shown that when there are no motion effects, the correlation coefficients between subaperture data sets were computed to be very close to 1.0, consistently. This proved that the triggering mechanism is accurate and eliminates the need to reconstruct a synthetic aperture using the ETAM software processing, suggested in this section. Instead with each firing, a section of the aperture is collected and these subapertures are pieced together to create a full synthetic aperture. In the example of Figure 12.1, with the first fire, sensors 1–32 are collected; with the second fire, sensors 33–64 are collected, and sensors 64–96 are collected with the third fire. These three segments are pieced together to create the full 96-element aperture. This is the approach used in both the 2D/3D and 3D/4D ultrasound systems reported in this chapter. In particular, for the linear phase array probe with 64 elements, the A/DC peripheral included 16 channels and, therefore, the full 64-sensor aperture was digitized from 4 successive subaperture acquisitions. In the case of the planar array system with  $16 \times 16 = 256$  elements, the A/DC peripheral included 64 channels and the full 256-channel aperture was digitized from 4 successive subaperture acquisitions.

The validity of this approach was assessed also from the reconstructed images. With a coherent aperture a clear consistent image is created. With any loss of coherence between subapertures, objects in the image are blurred and ghosting appears.

In conclusion, implementation of the synthetic aperture processing by means of the ETAM algorithm may not be needed when the acquisition period of subapertures is well synchronized, is very fast, and of the order of a few milliseconds. In this case, the simplest approach for the acquisition and digitization process of a large-size phase array is to piece together the subapertures in order to form the fully populated physical aperture for beamforming. This approach minimizes the number of firings by 50%, and it reduces the computational requirements of the overlap correlator.

### 12.2.2 Ultrasound Beamforming Structure for Line and Planar Arrays

Deployment of planar arrays by ultrasound medical imaging systems has been gaining increasing popularity because of its advantage to provide real 3D images of organs

under medical examination. The details of a 3D beamforming structure for planar arrays are provided in [3]. However, commercial ultrasound systems deploying planar arrays are not yet available. Moreover, if we consider that a state-of-the-art line phase array ultrasound system consists of 128 sensors, then a planar array ultrasound system should include at least  $128 \times 128 = 16,384$  sensors in order to achieve the angular resolution performance of a line array system and the additional 3D image reconstruction capability provided by the elevation beam steering of a planar array. Thus, increased angular resolution in azimuth and elevation beam steering for ultrasound systems means larger sensor arrays, with consequent technical and higher cost implications. As it will be shown in this chapter, the alternative is to implement adaptive beamforming in ultrasound systems that deploy a planar array with  $32 \times 32 = 1024$  sensors, which consist of 32 line arrays with 32 sensors each. Then, the anticipated array gain improvements by an adaptive beamformer, as defined in the next section, will be equivalent to those provided by a 64-sensor line array for azimuth beam steering and a 64-sensor vertical line array for elevation beam steering for real 3D ultrasound imaging. In summary, the array gain improvements for an adaptive 1024-sensor planar array will be equivalent to those that could be provided by a conventional  $64 \times 64 = 4096$  sensor planar array. This is because for line arrays, a quantitative assessment [3, 16] shows that the image resolution improvements of the proposed adaptive beamformers will be equivalent to a two- to three-time longer physical aperture.

To achieve an effective implementation of the above adaptive ultrasound beamforming concept and to allow for a flexible system design for line and or planar array ultrasound probes, the discussion in this chapter suggests that the advance beamforming structures, defined in [3, 16], address the above requirements for both line and planar array ultrasound probes.

### 12.2.3 Adaptive Beamforming Structure for Ultrasound Systems

Details on the implementation of adaptive beamformers [17, 18] for ultrasound imaging applications have been presented elsewhere [3, 16]. These adaptive beamformers are characterized as dual-use technologies, they have been tested in operational active sonars [16], and they are defined in detail in [3]. Real data results have shown that they provide array gain improvements for signals embedded in anisotropic noise fields, as is the case in the human body.

Despite the geometric differences between the line and planar arrays, the underline beamforming processes for these arrays are time-delay beamforming estimators, which are basically spatial filters. However, optimum beamforming requires the beamforming filter coefficients to be chosen based on the covariance matrix of the received data by the  $N$ -sensor array in order to optimize the array response [16]. For ultrasound applications, the outputs of the adaptive algorithms are required to provide coherent beam time series to facilitate the postprocessing of the image reconstruction processes. This means that these algorithms should exhibit near-instantaneous convergence and provide continuous beam time series that have sufficient temporal coherence to correlate with the reference replica in matched filter processing [16]. In what follows, the adaptive beamforming structure and its implementation in frequency domain, will be redefined in Sections 12.2.4 and 12.2.5 to address system design requirements for fully digital ultrasound imaging technologies. In particular, Section 12.2.4 will introduce the design of a fully digital multifocus active beamformer, and Section 12.2.5 will analyze

the complex structure of a multifocus receiving beamformer for linear and planar phase array probes, respectively.

#### 12.2.4 Multifocus Transmit Beamformer for Linear and Planar Phase Arrays

In ultrasound imaging, the transmitted signals consist of steered beams that illuminate a 2D or a 3D field in the case of a linear array or a planar array, respectively. The beamforming design, discussed in this section, defines beam illuminations that include a set of simultaneously transmitted multifocused beams to cover various depths (ranges) and are characterized by a wide angular width with low side lobes. To illuminate a specific region, several beam transmissions may be required where each transmission illuminates a certain angular sector. The preference in the present design is for a wide angle beamwidth to illuminate a wide region per beam so as to minimize the number of transmissions. Furthermore, the beam's side lobe structure should be downsized in order to minimize the signal power distributed outside the intended illuminated region that may contribute noisy interference in the reflected signals; and its performance should be assessed from the beam power pattern of the illuminated field.

##### 12.2.4.1 Multifocus Transmit Beamformer for Linear Phase Array

Figure 12.3 depicts the configuration of a linear phase array probe in Cartesian coordinates with angle  $\theta$  showing the angular direction of the active beam steering for the transmitted pulse. The illuminated field is the  $x$ - $y$  plane along the positive  $y$  axis with  $0^\circ \leq \theta \leq 180^\circ$ . The broadband characteristics of the transmitted pulse are defined by a linear frequency-modulated (FM) signal, defined by Eq. (12.1) below and with a time-domain response shown in Figure 12.4.

$$p(t) = \sin [2\pi(f_0 t + k_t^2)], \quad (12.1)$$

where  $f_0$  = lower cut-off frequency of the modulated FM signal

$k$  = linear frequency sweep rate [ $k = \text{BW}/(2 \times T_0)$ ]

$\text{BW}$  = bandwidth of the modulated FM signal

$T_0$  = pulse duration

Then, the temporal characteristics of the transmitted signal by the active sensors of the phase array probe are defined below by the time series of Eq. (12.2):

$$\begin{aligned} S_{\text{ref}}(\theta_i, R_s, m) &= \sin \left[ 2\pi \left( f_0 + k \cdot \frac{m}{f_s} \right) \right] \cdot \frac{m}{f_s} \cdot w_p(m - s_{\text{pos}}), \\ &\quad \text{when } s_{\text{pos}} \leq m \leq (s_{\text{pos}} + p_{\text{dur}}) \\ S_{\text{ref}}(\theta_i, R_s, m) &= 0, \quad \text{otherwise,} \end{aligned} \quad (12.2)$$

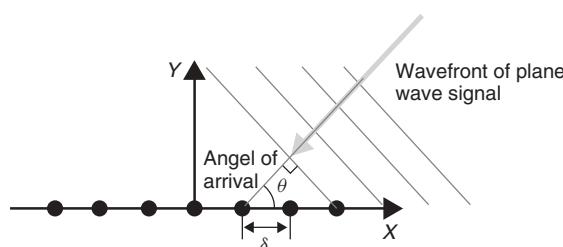
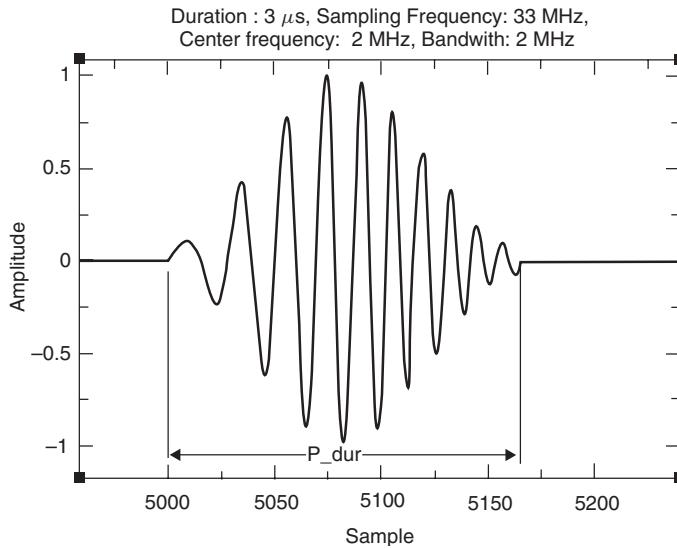


Figure 12.3 Geometric configuration and coordinate system for a line array of sensors.



**Figure 12.4** Linear FM pulse with a temporal window.

where  $f_s$  = sampling frequency  
 $f_0 = f_0(R_s, \theta_i)$ , which is the lower cut-off frequency specified for the focal range  $R_s$  and steering angle  $\theta_i$   
 $S_{\text{ref}}$  = transmitted ultrasound pulse signal  
 $s_{\text{pos}}$  = starting temporal location in the time series, which can be set to zero, as discussed in Section 12.2.4.2  
 $p_{\text{dur}}$  = duration of the active transmission, defined by the values of focal range  $R_s$  and steering angle  $\theta_i$ , as discussed in Section 12.2.4.2  
 $w_p(m - s_{\text{pos}})$  = temporal window of length  $p_{\text{dur}}$   
 $m$  = index for the temporal samples,

The temporal window  $w_p(m - s_{\text{pos}})$  reduces the spectral leak in the transmitted pulse, as discussed in [3], and it can be a Hamming or Kaiser window function. The frequency characteristics of the discrete-time series, such as the lower cut-off frequency,  $f_0$ , and the bandwidth are selected according to the focal range  $R_s$  and steering angle  $\theta_i$  of the transmitted pulse.

Then, for the  $n$ th transducer of the active aperture of the linear phase array probe, the time series  $S_{\text{ref}}(\theta_i, R_s, m)$ , of Eq. (12.1), are modified as follows:

$$S_m(\theta_i, R_s, m) = S_{\text{ref}}(\theta_i, R_s, m + \tau_n)w_n, \quad (12.3)$$

where  $n$  = transducer index

$\tau_n$  = time delay for the  $n$ th transducer of the active component of the line array probe steered at direction  $\theta_i$  and focused at range  $R_s$

$w_n$  = spatial window value for the  $n$ th transducer

and  $\tau_n$  is defined by

$$\tau_n = \left[ \left( R_s^2 + \delta_n^2 - 2R_s \delta_n \cos(\theta_i) \right)^{1/2} - R_s \right] \frac{f_s}{c}, \quad (12.4)$$

where  $\delta_n$  is the location of the  $n$ th transducer with respect to the linear array coordinate system.

To allow for multifocus at different focal ranges and along the same steering angle by using a single transmission, the transmitted time series are modified as follows:

$$S_n(\theta_i, m) = \sum_{\text{all } R_s | \theta = \theta_i} S_n(\theta_i, R_s, m). \quad (12.5)$$

To achieve a wide-angle illumination coverage with a single transmission, the beamwidth of the radiated pulses need to be as wide as possible, and this can be achieved with a spatial window applied as a weighting function along the active transducers of the linear phase array probe. There are two types of windows that have the above-desired characteristics, and these are the Gaussian and 4-term Blackman–Harris windows. More specifically, the 4-term Blackman–Harris window function is defined by the following equation:

$$w_n = a_0 - a_1 \cos\left(2\pi \frac{n}{N-1}\right) + a_2 \cos\left(4\pi \frac{n}{N-1}\right) - a_3 \cos\left(6\pi \frac{n}{N-1}\right), \quad (12.6)$$

where  $n = 0, 1, 2, \dots, N-1$

$a$  = 4-term Blackman–Harris window coefficients:  $a_0 = 0.35875$ ,  $a_1 = 0.48829$ ,  $a_2 = 0.14128$ ,  $a_3 = 0.01168$

$N$  = window length or the number of transducers in the line array probe

The Gaussian window is expressed by

$$w(n) = \exp\left[-\frac{1}{2}\left(\alpha \frac{n - N/2}{N/2}\right)^2\right], \quad (12.7)$$

where  $n = 0, 1, 2, \dots, N-1$ , with  $N$  being the number of the active elements of the array, which can be different than the corresponding  $N$  in 1.6

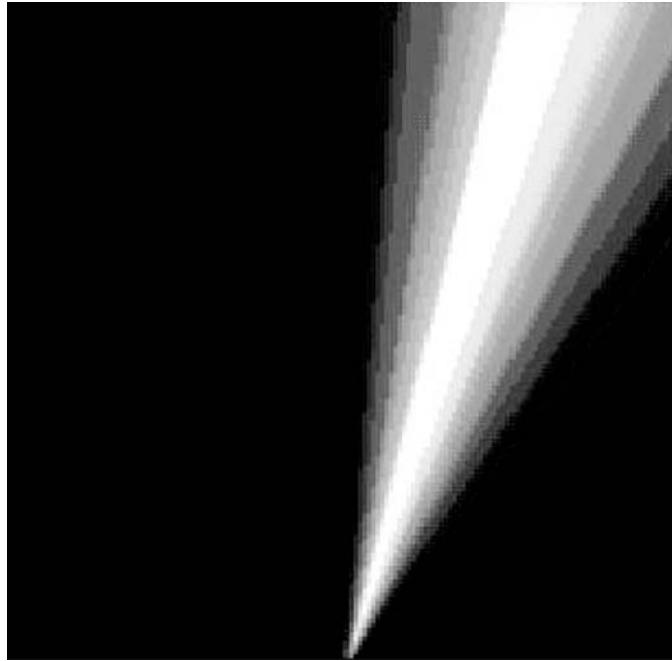
$\alpha$  = alpha parameter,  $\alpha \geq 2$

and  $\alpha$  is the Gaussian window parameter that controls the width of the Gaussian window. The width of the Gaussian window is the inverse of the beamwidth of the radiated pulse and the size of the side lobes with respect to the beam peak magnitude.

Then, the time series defined by Eq. (12.5) are converted into voltage levels through a D/AC peripheral to excite the transducers of the linear phase array probe. This kind of excitation generates illumination patterns, and their beam power pattern distribution is defined in the next section. Figure 12.5 shows the beam power plot according to Eq. (12.5) and the relations (12.1)–(12.4) for a Gaussian spatial window with  $\alpha = 3.4$ .

As shown in Figure 12.5, there is a considerable 3-dB beamwidth increase (about  $12^\circ$ ) due to the implementation of a spatial window. Moreover, since the Gaussian and the 4-point Blackman–Harris spatial windows show similar beamwidth improvements, the Gaussian window is preferable due to its flexibility in changing the  $\alpha$  parameter to obtain a wider selection of beamwidths.

**12.2.4.2 Beam Power Pattern for Multifocus Transmit Beamformer for Linear Phase Array** The image of the beam power pattern of the multifocus illumination is obtained by computing the total power of the signals that arrive at each of the pixels in the coordinate system of the linear phase array. The arriving signal at each



**Figure 12.5** 2D beam power pattern distribution of an illuminated area with size ( $10\text{ cm} \times 10\text{ cm}$ ) covering the angular sector  $0^\circ \leq \Theta \leq 180^\circ$  for radiated pulses by a linear phase array probe with 12 transducers having 0.4-mm sensor spacing. The beam steering is at  $75^\circ$  and focused at 5 cm. The center frequency and bandwidth of the transmitted signal are 2 and 2 MHz, respectively.

pixel is the added combinations from all active transducers radiated signals that are time delayed according to Eq. (12.5). Equation (12.8) expresses the time-delay factor in terms of temporal samples and defines the difference between the distance from the pixel to the center of the array and the distance from the pixel to the transducer,

$$\zeta_n(p_x, p_y, \delta_n) = \left[ \left( p_x^2 + p_y^2 \right)^{1/2} - \left( (p_x - \delta_n)^2 + p_y^2 \right)^{1/2} \right] \frac{f_s}{c}, \quad (12.8)$$

where  $p_x$  and  $p_y$  are the  $x$  and  $y$  locations of the pixel, respectively. Thus, the combined signals from all transducers arriving at a pixel are defined by

$$b(p_x, p_y, \theta_i, m) = \sum_n s_n(\theta_i, m + \zeta_n), \quad (12.9)$$

and the total power level of the combined signals that arrive at a pixel location  $(x, y)$  is

$$P(p_x, p_y, \theta_i) = \sum_m |b(p_x, p_y, \theta_i, m)|^2. \quad (12.10)$$

The beam power pattern image is finally obtained from Eq. (12.11):

$$P(p_x, p_y) = \sum_{\text{all } \theta_i} P(p_x, p_y, \theta_i), \quad (12.11)$$

which includes the summation of the radiated power from all steering angles. The values of  $P(p_x, p_y)$  are further scaled to fit an available gray scale range for image presentation. However, the computations as expressed by Eqs. (12.10) and (12.11) require only a small portion of the transducer signals. The temporal starting point of the pulse,  $s_{\text{pos}}$ , as defined in Eq. (12.2), can be arbitrarily set to *zero*. Then, this starting point  $s_{\text{posth}}$  of the active pulse is delayed or advanced by a specific time delay, as defined in Eq. (12.4), in order to generate the directional and focusing characteristics of the transmitted signals for a specific depth and angular direction. Furthermore, excitation of these signals by the transducers of the active aperture of the probe includes time delays according to Eq. (12.8) to get the intensity of the illumination at a specific pixel. The two time delays according to Eqs. (12.4) and (12.8), determine the total time-delay boundaries of how far the pulse on the reference signal in Eq. (12.2) might be shifted over time. Beyond these boundaries, the transducer signals have zero values and can be neglected in the signal power computations. These time-delay boundaries determine also the temporal length of the segment of the signal that needs to be taken into account for processing. The lower boundary can also be computed as

$$s_{\min} = s_{\text{pos}} - \tau_{\max}, \quad (12.12)$$

where

$$\tau_{\max} = 2\left(\frac{N}{2} + 0.5\right)\delta \frac{f_s}{c} \quad (12.13)$$

and  $s_{\min}$  is the lower boundary or the starting temporal sample of the signal in (12.2) that needs to be taken into account for processing,  $\tau_{\max}$  is the maximum time delay, and  $\delta$  is the transducer spacing in the linear array. The multiplication factor of 2 in Eq. (12.13) corresponds to the two-way propagation in Eqs. (12.4) and (12.8).

The beam output of the transmitted pulse by a linear phase array probe is defined by the following equation:

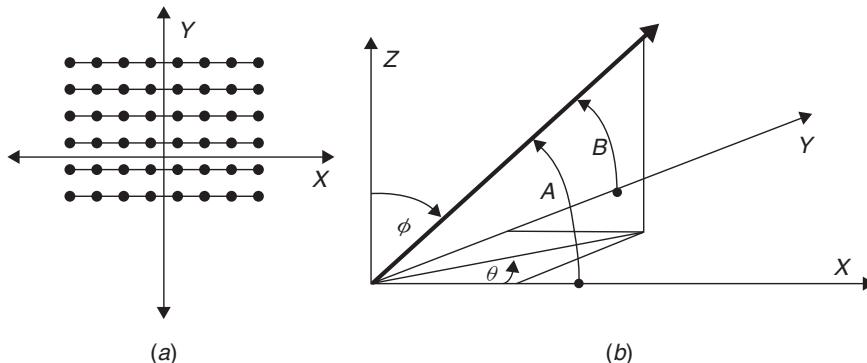
$$B(\theta_i, f) = \sum_n S_n(\theta_i, f) e^{j2\pi f \tau_n}, \quad (12.14)$$

where  $S_n(\theta_i, f)$  is the Fourier transform of the  $n$ th transducer signal  $s_n(\theta_i, m)$ . Equation (12.14) is identical to Eq. (12.25) in Section 12.2.4.4. The time delay  $\tau_n$  is a function of the transducer location and the steering angle  $\theta_i$  and was defined before by Eq. (12.4).

**12.2.4.3 Multifocus Transmit Beamformer for Planar Phase Array** The left-hand side of Figure 12.6 depicts the configuration of a linear phase array probe in Cartesian coordinates. The same coordinate system located at the center of the planar array defines the parameters (i.e.,  $A, B, \theta, \phi$ ) of the 3D beamformer at the right-hand side of Figure 12.6. The broadband characteristics of the transmitted pulses are the same as in the case of the linear phase array and are expressed by Eq. (12.1), and the time series defining transmitted pulses for the active aperture of the phase array planar array include also time delays to account for their range focusing and angular steering characteristics. These time series are expressed by

$$S_{l,n}(\theta_i, \phi_q, R_s, m) = S_{\text{ref}}(\theta_i, \phi_q, R_s, m + \tau_{l,n}) w_{l,n}, \quad (12.15a)$$

where  $S_{\text{ref}}(\theta_i, \phi_q, R_s, m + \tau_{l,n})$  is defined by Eq. (12.2) with  $\tau_{l,n}(A, B, R_s)$  being the time-delay steering defined by the angles ( $A, B$ ) and the focus range  $R_s$  and  $(l, n)$  are



**Figure 12.6** (a) Coordinate system and configuration of a planar array transducer symmetrically located on the  $x$ - $y$  plane. (b) Coordinate system for 3D beamforming process for the planar array shown in (a).

the indexes for a transducer located at the ( $l$ th row,  $n$ th column) of the planar array,  $\tau_{l,n}$  is the shift of the reference signal needed to create the signal for the ( $l, n$ )th transducer to achieve focus at  $R_s$  and  $(\theta_i, \phi_q)$  steering angle in 3D space, defined in Figure 12.6,  $w_{l,n}$  is the spatial window value for the ( $l, n$ )th transducer, and  $\delta_{l,n_x}$  and  $\delta_{l,n_y}$  are the  $x$  and  $y$  locations, respectively, of the ( $l, n$ )th transducer. Estimates of  $\tau_{l,n}(A, B, R_s)$  are provided from the following expression (12.15b):

$$\tau_{l,n}(A, B, R_s) = \frac{\sqrt{R_s^2 + \delta_{l,n_x}^2 + \delta_{l,n_y}^2 - 2\delta_{l,n_x} \cos A - 2\delta_{l,n_y} \cos B} - R_s}{c}. \quad (12.15b)$$

As in the case of the linear phase array active beamformer, the transmitted signals focused at different ranges,  $R_s$ , but along the same steering angle, can be summed up to be fired by the probe under a single transmission as shown below:

$$S_{l,n}(\theta_i, \phi_q, m) = \sum_{\text{all } R_s | \theta_i, \phi_q} S_{l,n}(\theta_i, \phi_q, R_s, m). \quad (12.16)$$

Implementation of a spatial window  $w_{l,n}$  on the planar array requires the application of the decomposition process of the 2D planar array beamformer into two line array beamforming processes that have been introduced in [3]. Thus, implementation of 3D Gaussian and 4-term Blackman–Harris windows to adjust the beamwidth of the 3D beam steering is reduced into a simple linear spatial window as part of the above decomposition process.

An alternative approach to the above decomposition process is to use an approximate implementation of the 2D Gaussian spatial window on the active transducers of the planar array. However, this approximation requires that the planar array is square with equal sizes along the  $x$  and  $y$  coordinates. This approximation includes the 2D Gaussian window expressed by

$$W(n) = \exp \left( -\frac{1}{2} \left( \alpha \frac{n - N/2}{N/2} \right)^2 \right), \quad (12.17)$$

where  $n = 0, 1, 2, \dots, N$ , and  $N$  is the window length with  $\alpha \geq 2$ .

To compute the window values for the transducers in the planar array, a 2D Gaussian window that spans over the diagonal of the planar array is formed. The 2D diagonal Gaussian window is used as a reference to describe the relationship between a transducer's window value versus the distance from the transducer to the center of the planar array. In forming the 2D diagonal Gaussian window, the parameter  $N$  needs to be sufficiently large in order to have good window resolution. In Eq. (12.18)  $D$  describes the diagonal length of the planar array:

$$D = \sqrt{2}(N - 1)d, \quad (12.18)$$

where  $N$  is the size of the one dimension of a square planar array,  $N \times N$ , and  $d$  is the sensor spacing along both the  $x$  and  $y$  coordinates of the equally spaced transducers of the planar array.

Let us denote  $\delta_{l,n}$  as the distance of the  $(l, n)$ th transducer from the center of the planar array:

$$\delta_{l,n} = \sqrt{\delta_{l,n_x}^2 + \delta_{l,n_y}^2}. \quad (12.19)$$

Then, the spatial window values,  $w_{l,n}$  for the  $(l, n)$ th transducer, can be obtained from the 2D Gaussian window  $W(n)$ :

$$w_{l,n} = W\left(\text{round}\left(N \cdot \frac{D/2 - \delta_{l,n}}{D}\right)\right), \quad (12.20)$$

where the function  $\text{round}(x)$  rounds the  $x$  value to the nearest integer.

Estimates of the spatial window weights for a rectangular planar array of size  $N \times M$  can be derived from the decomposition of the planar array beamformer into two sets of linear array beamformers as defined in [3].

**12.2.4.4 Beam Power Pattern for Multifocus Transmit Beamformer for Planar Phase Array** The image of the beam power pattern of the multifocus illumination is obtained by computing the total power of the signals that arrive at each of the pixels in the coordinate system of the planar phase array. The arriving signal at each pixel is the added combinations from all active transducer radiated signals that are time delayed according to Eq. (12.21):

$$\begin{aligned} & \xi_{l,n}(\delta_{l,n_x}, \delta_{l,n_y}, p_x, p_y, p_z) \\ &= \left[ \left( p_x^2 + p_y^2 + p_z^2 \right)^{1/2} + \left( (p_x - \delta_{l,n_x})^2 + (p_y - \delta_{l,n_y})^2 + p_z^2 \right)^{1/2} \right] \frac{f_s}{c}, \end{aligned} \quad (12.21)$$

where  $p_x, p_y, p_z$  are the  $x, y, z$  locations, respectively, of the pixel, and  $\xi_{l,n}$  denotes the difference between the distance from the pixel to the center of the array and the distance from the pixel to the  $(l, n)$ th transducer.

As a result of the time delay of Eq. (12.21), the combined signal radiated from all the active transducers of the planar array that arrives at a pixel  $(x, y)$  can be computed by adding the time-shifted transducer signals as follows:

$$b(p_x, p_y, p_z, \theta_i, \phi_q, m) = \sum_l^N \sum_n^N S_{l,n}(\theta_i, \phi_q, m + \xi_{l,n}). \quad (12.22)$$

Then, the intensity of illumination of the pixel located at  $(x, y)$  can then be expressed by

$$P(p_x, p_y, p_z, \theta_i, \phi_q) = \sum_m \left| \sum_l^N \sum_n^N S_{l,n}(\theta_i, \phi_q, m + \zeta_{l,n}) \right|^2, \quad (12.23)$$

and the 3D beam power pattern image is obtained from

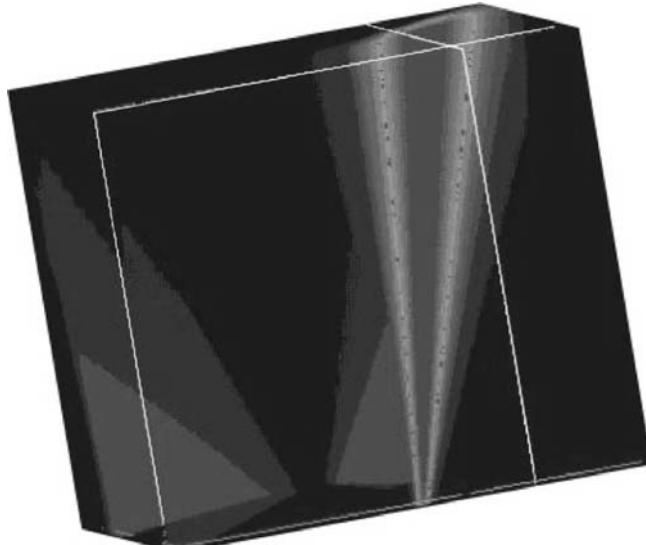
$$P(p_x, p_y, p_z) = \sum_{\text{all } \theta_i, \phi_q} P(p_x, p_y, p_z, \theta_i, \phi_q), \quad (12.24)$$

which represents the 3D image of the summation of all the illuminations of the 3D steering angles.

Figure 12.7 shows the 3D beam power pattern distribution from the illumination of a volume of size  $(10 \text{ cm} \times 10 \text{ cm} \times 10 \text{ cm})$  by a planar array with  $12 \times 12$  transducers having 0.4-mm transducer spacing. The beam steering with multiple range focusing is at a 3D angle  $(75^\circ, 75^\circ)$  as defined by the parameters depicted in the coordinate system of Figure 12.6. The center frequency and bandwidth of the transmitted signal are 2 and 2 MHz, respectively.

As in the case of the linear array active beamformer, the maximum temporal length of the active beam time series of the planar array can be estimated from

$$\tau_{\max} = 2 \left( \frac{N}{2} + 0.5 \right) \sqrt{2} \delta \frac{f_s}{c}, \quad (12.25)$$



**Figure 12.7** 3D beam power pattern distribution from the illumination of a volume of size  $(10 \text{ cm} \times 10 \text{ cm} \times 10 \text{ cm})$  by a planar array with  $12 \times 12$  transducers having 0.4-mm transducer spacing. The beam steering with multiple range focusing is at a 3D angle  $(75^\circ, 75^\circ)$  as defined by the parameters depicted in the coordinate system of Figure 12.6. The center frequency and bandwidth of the transmitted signal are 2 and 2 MHz, respectively.

where the multiplication factor  $\sqrt{2}$  takes into account the transducer spacing along the diagonal direction. Then the lower boundary or the starting temporal sample of the synchronized active transducer signals is

$$s_{\min} = s_{\text{pos}} - \tau_{\max}. \quad (12.26)$$

Finally, the beam output of the transmitted pulse by an active planar array probe is defined by

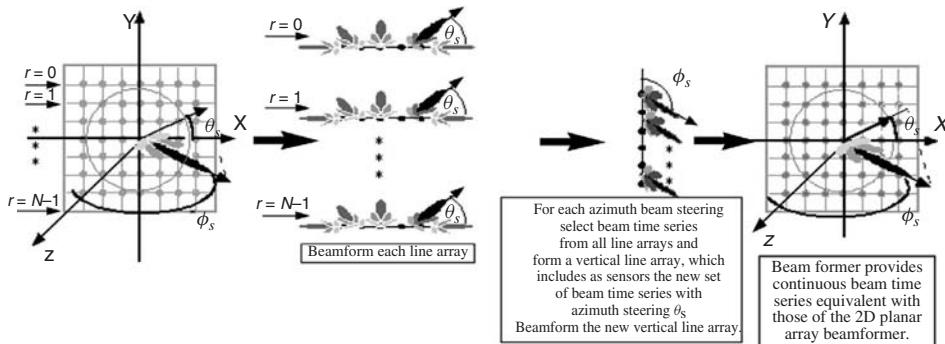
$$B(\theta_i, \phi_q, R_s, f) = \sum_l \sum_m S_{l,n}(\theta_i, \phi_q, f) e^{j2\pi f \tau_{l,n}(\theta_i, \phi_q, R_s)}, \quad (12.27)$$

where  $S_{l,n}(\theta_i, \phi_q, f)$  is the Fourier transform of the  $(l, n)$ th transducer signal  $s_{l,n}(\theta_i, \phi_q, m)$ . The details of the 3D planar array beamformer and its decomposition process into two line array beamformers are defined in [3].

### 12.2.5 Multifocus Receiving Beamformer for Linear and Planar Phase Arrays

The main functions of a receiving ultrasound beamformer is to beamform the reflections and backscattering fields that result from the illumination of a field of view (i.e., body organs) with ultrasound beam-steering waves, as defined in the previous two sections. Thus, following the steered transmission of directional ultrasound pulses in a specific region, the operation of an ultrasound system reverts to the receiving mode. During the receiving mode, a fully digital ultrasound system digitizes the sensor time series of an ultrasound linear or planar array probe that senses the reflections and backscattering effects of the illuminated field of view. Then, a digital receiving beamformer will beamform the received sensor time series, and the output of this process will provide beam time series that will form the basis for the reconstruction of the tomography 2D (B scan) or 3D volumetric image of the illuminated field of view. In this case, the theoretical analysis on beamforming [2, 3, 16–19] can be used by system engineers to design a receiving 2D or 3D ultrasound beamformer. However, the receiving ultrasound beamformer is more complex than the corresponding transmit (i.e., active) beamformer, and the following two sections will provide details in terms of their multifocus configuration and their time-series concatenation to prepare them for the image reconstruction process by taking into consideration the theory in [3].

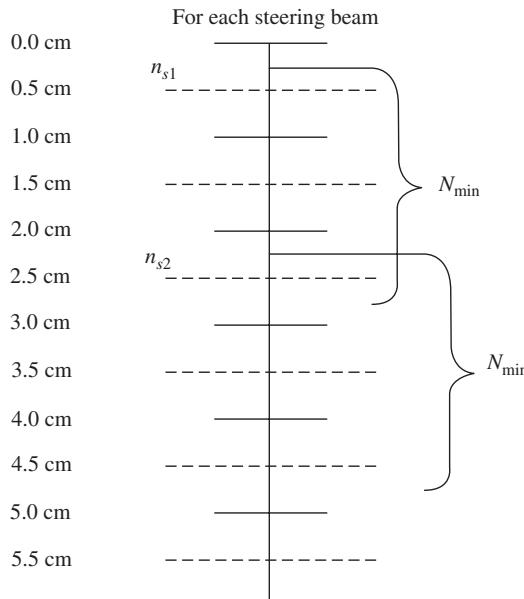
**12.2.5.1 Receiving Ultrasound Beamformer for Linear Phase Array** The coordinate system of a receiving beamformer for linear phase arrays is the same as in the case of the corresponding transmit beamformer. The receiving line array is chosen to be located along the  $x$  axis with the array center at  $x = 0$ , as depicted in Figure 12.8. The receiving beams are steered in the 2D plane along the positive  $y$  axis,  $0^\circ \leq \theta \leq 180^\circ$ . Their beam width is defined by Eq. (1.29) in [3] and forms the image resolution characteristics of the reconstructed image. Thus, each receiving ultrasound beam  $b(\theta_i, r_s, m)$  is characterized by its angular direction  $\theta_i$  (i.e., steering angle) and focus range  $r_s$ . In terms of angular resolution, the image characteristics can be improved by either using interpolation techniques or generating adaptive beams of intermediate fine angular directions between two consecutive steerings  $\theta_i$  and  $\theta_{i+1}$ . In



**Figure 12.8** Coordinate system and geometric representation of the concept of decomposing a planar array beamformer. The sensor planar array beamformer consists of  $N$  linear arrays with  $M$  being the number of sensors in each linear array.

terms of pixel resolution along the temporal axis (depth), the reconstructed image can be improved by dividing the field of view into focal zones, as shown in Figure 12.9. In other words, for a specific angular direction  $\theta_i$  (i.e., steering), the receiving beamformer generates a number of different beams focused at different ranges,  $b(\theta_i, r_s, m)$ , with  $s = 1, 2, 3, \dots$ . Thus, the focal zones in units of temporal samples can be expressed as

$$r_s = 2 \times r_{\text{meter}} \times \frac{f_s}{c}, \quad (12.28)$$



**Figure 12.9** Blocks show the focal zones, each of size 1.0 cm, along a specific azimuth direction, which is the temporal axis denoting range or depth.  $N_{\min}$  denotes the smallest amount of temporal samples for the beamforming process.

where

$$\begin{aligned} 0 \leq r_{s_i} &\leq r_s(i + \frac{1}{2}), & i = 1, \\ r_s(i - \frac{1}{2}) &\leq r_{s_i} \leq r_s(i + \frac{1}{2}), & i > 1 \end{aligned}$$

defines each focal zone by its center, as schematically depicted in Figure 12.9. The factor of 2 in Eq. (12.28) is due to the round trip travel time of the received ultrasound signal.

Although a decrease in the size of a focal zone may improve the image resolution along the axis of range, this choice will increase the number of focal zones and therefore the generation of number of focused receiving beams, a process that will increase the computational load. In other words, for  $I$  number of steering angles and  $S$  number of focal zones, the beamforming process has to be repeated  $I \times S$  times. In the experimental system discussed in this chapter, the focal zone size was set to 1 cm, with angular beamwidth of  $0.5^\circ$ .

It is apparent from the above discussion that the temporal samples of the sensor or beam time series represent range (or depth), and this relationship is defined by

$$r_m = \frac{m}{f_s} c \frac{1}{2}, \quad (12.29)$$

where  $r_m$  represents range (or depth) in units of temporal samples and  $m$  is the sample index of the input sensor time series. The factor of  $\frac{1}{2}$  in (12.29) takes into account the round-trip travel time of the received signal as denoted by Eq. (12.28).

Since the temporal samples of the sensor and beam time series are related to range, a substantial saving in processing time can be achieved by using the smallest possible set of data samples in the beamforming process. This smallest amount of temporal samples,  $M_{\min}$ , that can be considered in the beamforming process should be sufficient to cover at least one focal zone plus twice the maximum time delay of the beam-steering process as defined below:

$$M_{\min} \geq r_m + 2 \left| \frac{1}{2}(N - 1)\delta \frac{f_s}{c} \right|, \quad (12.30)$$

where  $M_{\min}$  denotes the lowest number of temporal samples for the beamforming process and  $N$  is the number of sensors in the linear array probe.

Let us consider now a beamforming process with  $M_{\min}$  temporal samples for the  $s$ th focal zone. Then, the sensor time series and their data segment of temporal samples associated with the specific focal zone  $r_s$  are defined by

$$\begin{aligned} x_n(m) &= y_n(m + r_{s_i}), & 0 \leq m \leq M_{\min} - 1, 0 \leq m + r_{s_i} \leq M, \\ x_n(m) &= 0, & 0 \leq m \leq M_{\min} - 1, (m + r_{s_i} \leq 0, \text{ or, } m + r_{s_i} \geq M), \end{aligned} \quad (12.31)$$

where  $y_n(m)$  is the  $n$ th sensor signal of the linear array probe with  $M$  being the total number of temporal samples in the digitized sensor time series of the  $n$ th sensor. The Fourier transform of  $x_n(m)$  with a filtering operation are defined by,

$$X_n(f_m) = \text{FFT}[x_n(m)] \quad (12.32)$$

$$S_n(f_m) = X_n(f_m) \cdot H_{r_s}(f_n, r_s), \quad (12.33)$$

where  $m = l, l + 1, \dots, l + L$  and  $H_{r_s}(f_n, r_s)$  are the bandpass filter coefficients designated for each focal zone, and FFT is fast Fourier transform. The filtering process

of Eq. (12.33) is a critical step in the focus-receiving beamformer. In particular, high-frequency regimes of ultrasound signals provide better image resolution than lower frequency regimes. However, the associated high propagation losses allow only for very short ranges of ultrasound penetration to achieve effective imaging. On the other hand, lower frequency regimes in the ultrasound signals provide deeper penetration at the expense of poor image resolution.

Ultrasound system designers have used the above propagation characteristics of the ultrasound energy in the human body to structure the broadband frequency spectrum of transmitted pulses illuminating a medium of interest. Thus, the high-frequency regime of a received ultrasound signal is being used for short ranges (i.e., short-range focal zones), while the lower part of the spectrum is being allocated for the deeper focal zones. This kind of design approach requires very wide broadband ultrasound pulses to allow for segmentation of their wideband frequency spectrum into smaller frequency regimes that can be activated by the receiving focus beamformer through the filtering process of Eq. (12.33). Moreover, this filtering process can use a small number of frequency bins  $\{X_n(f_m), m = l, l+1, \dots, l+L\}$  of the sensor signals that represent the frequency regime associated with the focal zone of the receiving beamformer. As a result, the same  $X_n(f_m)$  sensor time series can be reused for as many times in the receiving focus beamforming process as is the number of focal zones. The design characteristics for the filter coefficients,  $H_{r_s}(f_n, r_s)$  can be Hamming or Kaiser FIR filters. An additional advantage of the filtering process defined in Eq. (12.33) is that it minimizes the computational load by minimizing the summation processes of the receiving beamformer [i.e., see Eq. (12.34)] to be equal with the number,  $L$ , of the frequency bins that are considered in the filtering process in (12.33).

The beamforming for each focal zone and steering angle is then performed by phase shifting the sensor signals  $S_n(f_m)$  in the frequency domain through multiplications with the steering vector  $\bar{D}(f, r, \theta)$  with its  $n$ th element being  $d_n(f_m, r_s, \theta_i)$  and summations as defined by Eq. (12.25) in [3], which are rewritten below:

$$B(f_m, r_s, \theta_i) = \sum_n S_n(f_m) d_n(f_m, r_s, \theta_i) w_n. \quad (12.34)$$

The  $n$ th steering element  $d_n(f_m, r_s, \theta_i)$  of the steering vectors is defined by

$$d_n(f_m, r_s, \theta_i) = \exp \left[ j2\pi \frac{(m-1)f_s}{M} \frac{\tau_n(r_s)}{c} \right] \quad (12.35)$$

and

$$\tau_n(r_s) = \sqrt{r_s^2 + \delta_n^2 - 2r_s \delta_n \cos \theta_i}, \quad (12.36)$$

where  $\delta_n$  is the location of the  $n$ th transducer.

For the image reconstruction process, however, the beams in (12.34), which are in the frequency domain, need to be converted into beam time series as follows:

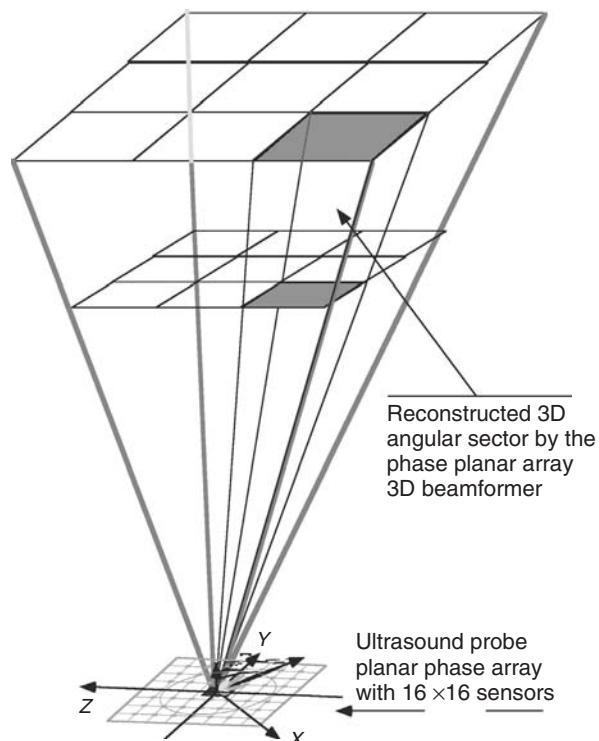
$$b(\theta_i, r_s, m) = \text{IFFT}\{B(f_m, r_s, \theta_i)\}. \quad (12.37)$$

Furthermore, the beam time series  $b(\theta_i, r_s, m)$  of different focal zones ( $r_s, s = 1, 2, 3, \dots$ ), but along the same steering angle  $\theta_i$ , need to be concatenated to reconstruct a new single-beam time series for the steering angle  $\theta_i$  and with a total

number of temporal samples covering the full range (i.e., depth) of the image reconstruction process. For a given focus range, the minimum number of temporal samples from the output of the focus receiving beamformer have already been defined by Eq. (12.30). Thus, the concatenation process is a simple operation of cut and pasting the time samples of  $b(\theta_i, r_s, m)$  from each  $s$ th focal zone. This process will generate a new beam time series  $b(\theta_i, m)$  with typical maximum number of temporal samples  $M$ , equal to the number of samples in the sensor time series  $x_n(m)$  of the line array probe. To comply also with display requirements, the beam time series  $b(\theta_i, m)$  may need to be compressed up to a certain length. The compression is done through an Oring operation as expressed by Eq. (12.38):

$$b_l(\theta_s, m) = (b_{l-1}^6(\theta_i, 2m) + b_{l-1}^6(\theta_i, 2m + 1))^{1/6}. \quad (12.38)$$

**12.2.5.2 Receiving Ultrasound Beamformer for Planar Phase Array** The focus in the design concept of the energy transmission or reception beamforming module for planar arrays is to illuminate or receive the entire volume of interest with a few firings. This is shown in Figure 12.10. Here the volume is illuminated in  $3 \times 3$  sectors, which includes a total of 9 firings. The transmitted signals are all broadband FM (chirp) signals as discussed in the previous sections. They are fired with interelement



**Figure 12.10** Volume 3D/4D digital scanning for ultrasound applications using a planar phase array probe.

delays to allow the transmitted energy to be focused at specific regions in space, which is the space highlighted by the square shaded areas of Figure 12.10. The beamforming energy transmission is done through the  $12 \times 12$  elements at the center of the array, while the receiving 3D beamformer through the  $32 \times 32$  elements of the full phase array. The receiving data acquisition unit digitizes the sensed ultrasound reflections via the A/D/C peripheral of the unit, under a similar arrangement as was defined in the previous sections. The angular subsectors depicted in Figure 12.10, are arranged in column-row configuration. Each angular subsector occupies the region bounded by the  $A$  and  $B$  angles, as defined in Figure 12.6. In Figure 12.10, the angular sector  $70^\circ \leq A \leq 110^\circ, 70^\circ \leq B \leq 110^\circ$  is shown to be divided into 9 angular subsectors consisting of 3 rows and 3 columns. Each subsector occupies  $10^\circ$  of  $A$  angle and  $10^\circ$  of  $B$  angle, with their coordinate system defined in Figure 12.6. As discussed in the previous sections, each subsector is illuminated by separate transmissions. There are, therefore, the same number of sets of received signals as the number of subsectors. The image of each angular subsector is also reconstructed separately using the corresponding set of received data. Volumetric reconstruction of each subsector is derived from the 3D beamforming process applied on all the received sensor time series of the planar array. The number of beams to be formed by the receiving beamformer is specified by the size of the angular subsector. For example, the receiving beamformer is specified to form 10 beams in the angular direction  $A$  and 10 beams in the angular direction  $B$ . This means that 100 ( $10 \times 10$ ) beams will be used to fill and reconstruct the image of each angular subsector.

As in the case of the receiving linear array beamformer, the temporal samples of the sensor and beam time series are related to range (i.e., depth of ultrasound penetration) and therefore a substantial saving in processing time can be achieved by using the smallest possible set of temporal samples in the beamforming process. This smallest amount of temporal samples,  $M_{\min}$  that can be considered in the beamforming process should be sufficient to cover at least one focal zone plus twice the maximum time delay of the beam-steering process as defined before by Eq. (12.30).

Thus, Eq. (12.31) for the linear array is modified for the planar array as follows:

$$\begin{aligned} x_{l,n}(m) &= y_{l,n}^{v,w}(m + r_{s_i}), \quad 0 \leq m \leq M_{\min} - 1, 0 \leq m + r_{s_i} \leq M, \\ x_{l,n}(m) &= 0, \quad 0 \leq m \leq M_{\min} - 1, (m + r_{s_i} \leq 0, \text{ or, } m + r_{s_i} \geq M), \end{aligned} \quad (12.39)$$

where  $l, n$  = transducers indexes ( $l$ th row and  $n$ th column) of the array

$v, w$  = angular subsector index ( $v$ th row and  $w$ th column)

$y_{l,n}^{v,w}(m)$  = input signal of the  $(l, n)$ th transducer of the  $(v, w)$ th angular subsector

$r_{s_i}$  = starting sample for the  $S_i$ th focal zone, as defined by Eqs. (12.40) and (12.28)

$M$  = number of temporal samples of the input sensor time series,  $y_{l,n}^{v,w}(m)$ , which is the length of one snapshot

and

$$M_{\min} \geq r_{s_i} + 2\sqrt{2} \left| \frac{1}{2}(N-1)\delta \frac{f_s}{c} \right|. \quad (12.40)$$

Then, the 3D beamforming process implemented in the frequency domain for the digitized sensor time series  $x_{l,n}(t_m)$ ,  $l = 1, 2, \dots, N, n = 1, 2, \dots, N, m = 1, 2, \dots, M$  of the  $N \times N$  detectors of the phase planar array with  $M$  time samples for each sensor

collected is given below by (12.41), which is identical with (12.27) for the transmit beamformer, with the coordinates defined in Figures 12.6 and 12.10:

$$B(f_m, A, B, r_s) = \sum_{l=0}^{N-1} \sum_{n=0}^{N-1} X_{l,n}(f_m) H_s(f_m, r_s) S_{l,n}(f_m, A, B, r_s), \quad (12.41)$$

where  $X_{l,n}(f_m) = \text{FFT}[x_{l,n}(t_m)]$  and  $H_s(f_m, r_s)$  are the FIR filter coefficients as in (12.33) with the steering vector expressed by  $S_{l,n}(f_m, A, B, r_s) = \exp[j2\pi f_m \tau_{l,n}(A, B, r_s)]$  and the interelement time delays defined by (12.42):

$$\tau_{l,n}(A, B, r_s) = \frac{\sqrt{r_s^2 + \delta_{l,n_x}^2 + \delta_{l,n_y}^2 - 2\delta_{l,n_x} \cos A - 2\delta_{l,n_y} \cos B} - r_s}{c}, \quad (12.42)$$

where the sensor element  $(l, n)$  is located at position  $(\delta_{l,n_x}, \delta_{l,n_y})$ .

Equation (12.42) indicates that for each beam  $(A, B)$  and focal depth  $r_s$  there needs to be  $N \times N$  complex steering vectors computed for each frequency bin of interest. Furthermore, each set of steering vectors  $S_{l,n}(f_m, A, B, r_s)$  is unique, with each of the four function variables independent and not separable. Because of this independence, it is not possible to decompose this beamformer in an efficient manner.

Reference [3] introduces a decomposition process for the 3D planar array beamformer that presents an alternative to the beamformer in (12.41) [17, 18]. This decomposition process allows the beamforming equation in (12.41) to be divided and, hence, decomposed, which in turn allows for it to be easily implemented on a parallel architecture, discussed in Section 12.3. However, it is important to note that this decomposition process is a very close approximation of (12.41), resulting in a simplified two-stage linear beamforming implementation. For plane-wave arrivals (i.e.,  $r_s \rightarrow \infty$ ), the approximation in (12.43) can be directly derived from (12.41), and the derivation is exact. The approximation in the decomposition process for the time-delay parameter in (12.42) is defined by

$$\tau_{l,n}(A, B, r_s) = \frac{\sqrt{r_s^2 + \delta_{l,n_x}^2 - 2\delta_{l,n_x} r_s \cos A} - r_s}{c} + \frac{\sqrt{r_s^2 + \delta_{l,n_y}^2 - 2\delta_{l,n_y} r_s \cos B} - r_s}{c}. \quad (12.43)$$

This approximation leads to the decomposition of the 3D beamforming into two linear steps, expressed by

$$B(f_m, A, B, r_s) = \sum_{l=0}^{N-1} S_l(f_m, B, r_s) \cdot \left[ \sum_{n=0}^{N-1} X_{l,n}(f_m) \cdot H_s(f_m, r_s) \cdot S_n(f_m, A, r_s) \right] \quad (12.44)$$

with the two separated steering vectors expressed as

$$S_l(f_m, A, r_s) = \exp \left\{ j2\pi f_m \left( \frac{\sqrt{r_s^2 + \delta_{l,n_x}^2 - 2\delta_{l,n_x} r_s \cos A} - r_s}{c} \right) \right\},$$

$$S_n(f_m, B, r_s) = \exp \left\{ j2\pi f_m \left( \frac{\sqrt{r_s^2 + \delta_{l,n_y}^2 - 2\delta_{l,n_y} r_s \cos B} - r_s}{c} \right) \right\}.$$

In (12.44), the summation in square brackets is equal to a line array beamformer along the  $x$  axis of the coordinate system in Figure 12.6. This term is a vector that can be denoted as  $B_n(f_m, A, r_s)$ . Then (12.43) can be rewritten as follows:

$$B(f_m, A, B, r_s) = \sum_{n=0}^{N-1} B_n(f_m, A, r_s) \cdot S_n(f_m, B, r_s), \quad (12.45)$$

which defines a linear beamforming along the  $y$  axis, with the beams  $B_n(f_m, A, r_s)$  treated as the input time series. This kind of two-stage implementation is easily parallelized and implemented on a multinode system, as discussed in Section 12.3 and schematically illustrated by Figure 12 in [3]. In this approximate implementation [e.g., expressions (12.41) compared with (12.44) and (12.45)], the error introduced at angles  $A$  and  $B$  close to broadside is negligible. Side-by-side comparisons show that there is no degradation in image quality over the exact implementation for planar array ultrasound application.

The rest of the beamforming and image reconstruction processes for the planar array ultrasound system are identical with those defined for the linear array receiver beamformer in the previous section and expressed by (12.32), (12.33) and (12.37), (12.38). Thus, the formation of beam time series, according to (12.37),  $b(A_i, B_i, r_s, m)$  of different focal zones ( $r_s, s = 1, 2, 3, \dots$ ), but along the same steering angle ( $A_i, B_i$ ) need to be concatenated (e.g., cut-and-paste operations) to reconstruct a new single multifocus beam time series for the steering angle ( $A_i, B_i$ ) and with a maximum total number of temporal samples to be equal with the temporal length of the input sensor time series, covering the full range (i.e., depth) of the image reconstruction process.

To comply also with display requirements, the beam time series  $b(A_i, B_i, r_s, m)$  may need to be compressed up to a certain length. The compression is done through an Oring operation defined by (12.38); and when the formation of the multifocus beam time series is complete, the 3D volumetric ultrasound image, as depicted in Figure 12.10, can be reconstructed. The volume is divided into pixels. A pixel value is determined through an interpolation process of the beam time samples, as follows. First, the radial distance  $r_p$  of a pixel,  $(x_p, y_p, z_p)$  from the center of the array, is determined. The angular location  $(A_p, B_p)$  of the pixel is also computed from

$$r_p = \sqrt{x_p^2 + y_p^2 + z_p^2}, \quad (12.46a)$$

$$A_p = \tan^{-1} \frac{z_p}{x_p}, \quad (12.46b)$$

$$B_p = \tan^{-1} \frac{z_p}{y_p}. \quad (12.46c)$$

Then, the following eight beam time samples

$$\begin{array}{ll} b(r_{p-}, A_{p-}, B_{p-}), & b(r_{p+}, A_{p-}, B_{p-}), \\ b(r_{p-}, A_{p-}, B_{p+}), & b(r_{p+}, A_{p-}, B_{p+}), \\ b(r_{p-}, A_{p+}, B_{p-}), & b(r_{p+}, A_{p+}, B_{p-}), \\ b(r_{p-}, A_{p+}, B_{p+}), & b(r_{p+}, A_{p+}, B_{p+}), \end{array}$$

are used to interpolate each of the pixel values, where

$$r_{p-} \leq r_p \leq r_{p+}, \quad (12.47a)$$

$$A_{p-} \leq A_p \leq A_{p+}, \quad (12.47b)$$

$$B_{p-} \leq B_p \leq B_{p+}, \quad (12.47c)$$

with  $r_{p-}$  and  $r_{p+}$  representing the two closest beam time samples to the pixel location  $r_p$ . Similarly,  $(A_{p-}, A_{p+})$  are the two closest angles to the angular location,  $A_p$  and  $(B_{p-}, B_{p+})$  are the two closest angles to the angular location,  $B_p$ . If the pixel is located outside the angular sector being illuminated, its value is set to zero.

### 12.3 COMPUTING ARCHITECTURE AND IMPLEMENTATION ISSUES

Implementation of a fully digital 3D adaptive beamforming structure in ultrasound systems is a nontrivial issue. In addition to the selection of the appropriate algorithms, success is heavily dependent on the availability of suitable computing architectures.

Past attempts to implement matrix-based signal processing methods, such as adaptive beamformers, were based on the development of systolic array hardware because systolic arrays allow large amounts of parallel computation to be performed efficiently since communications occur locally. None of these ideas are new. Unfortunately, systolic arrays have been much less successful in practice than in theory. The fixed-size problem for which it makes sense to build a specific array is rare. Systolic arrays big enough for real problems cannot fit on one board, much less one chip, and interconnects have problems. A 2D systolic array implementation will be even more difficult. So, any new computing architecture development should provide high throughput for vector as well as matrix-based processing schemes.

A fundamental question, however, that must be addressed at this point is whether it is worthwhile to attempt to develop a dedicated architecture that can compete with a multiprocessor using stock microprocessors. However, the experience gained from sonar computing architecture developments [16] suggests that a cost-effective approach in that direction is to develop a PC-based computing architecture that will be based on the rapidly evolving microprocessor technology of the central processing units (CPUs) of PCs. Moreover, the signal processing flow of advanced processing schemes that include both scalar and vector operations should be very well defined in order to address practical implementation issues. When the signal processing flow is well established, such as in Figures 13, 19, 20, and 21 in Reference [3], then distribution of this flow in a number of parallel CPUs will be straightforward. In the following sections, we address the practical implementation issues by describing the current effort of developing an experimental fully digital 3D/4D ultrasound system deploying a planar array to address the requirements of the Canadian forces for noninvasive portable diagnostic devices deployable in fields of operations.

#### 12.3.1 Technological Challenges for Fully Digital Ultrasound System Architecture

The current state of the art in high-resolution, digital, 3D ultrasound medical imaging faces two main challenges:

1. The ultrasound signal processing structures are computationally demanding. Traditionally, specialized computing architectures and hardware have been used to provide the levels of performance *and* input/output (I/O) throughput required, resulting in high system design and ownership costs. With the emergence of high-end workstations and low-latency, high-bandwidth interconnects [20], it now becomes interesting and timely to investigate if such technologies can be used in building low-cost, high-resolution, 3D ultrasound medical imaging systems.
2. Although beamforming algorithms in digital configuration have been studied in the context of other applications [16], little is known about their computational characteristics with respect to ultrasound-related processing and medical applications in general. It is not clear which parts of these algorithms are the most demanding in terms of processing or communication and how exactly they can be mapped on modern parallel PC-based architectures. In particular, although the algorithmic complexity of different sections can be calculated, little has been done in terms of actual performance analysis on real systems. The lack of such knowledge inhibits further progress in this area since it is not clear how these algorithms should evolve to lead to applicable solutions in the area of ultrasound medical imaging.

Reference [20] addresses both these two issues by introducing a design of a parallel implementation of advanced 3D beamforming algorithms [3] and studying its behavior and requirements on a generic computing architecture that consists of commodity components. This design concept provides an efficient, all-software, sequential implementation that shows considerable advantages over hardware-based implementations of the past. It provides also an efficient parallel implementation of advanced 3D beamforming [3] for a cluster of high-end PCs connected with a low-latency, high-bandwidth interconnection network that allows also for an analysis of its behavior. The emphasis in this design has been placed also on the identification of parameters that critically affect both the performance and cost of ultrasound system.

The end result [20] reveals a number of interesting characteristics leading to conclusions about the prospect of using commodity architectures for performing all related processing in ultrasound imaging medical applications. A brief summary of these findings suggests the following:

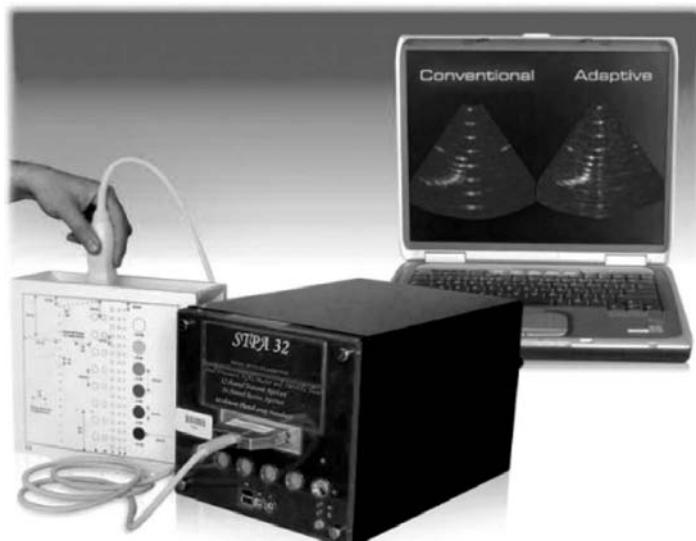
- A PC-based multiprocessor system today can achieve close to real-time performance for high-end ultrasound image quality and is certainly expected to do so in the near future.
- The major components of a digital 3D ultrasound beamforming signal processing structure [16, 20] consists of:
  - (a) Eighty-five to 98% of the time is spent in FFT and beam-steering functions.
  - (b) The communication requirements in the particular implementation are fairly small, localized, and certainly within the capabilities of modern low-latency, high-bandwidth interconnects.
  - (c) The results in Reference [20] provide an indication of the amount of processing required for a given level of ultrasound image quality and number of channels in a probe that can be used as a reference in designing computing architectures for ultrasound systems.

## 12.4 EXPERIMENTAL PLANAR ARRAY ULTRASOUND IMAGING SYSTEM

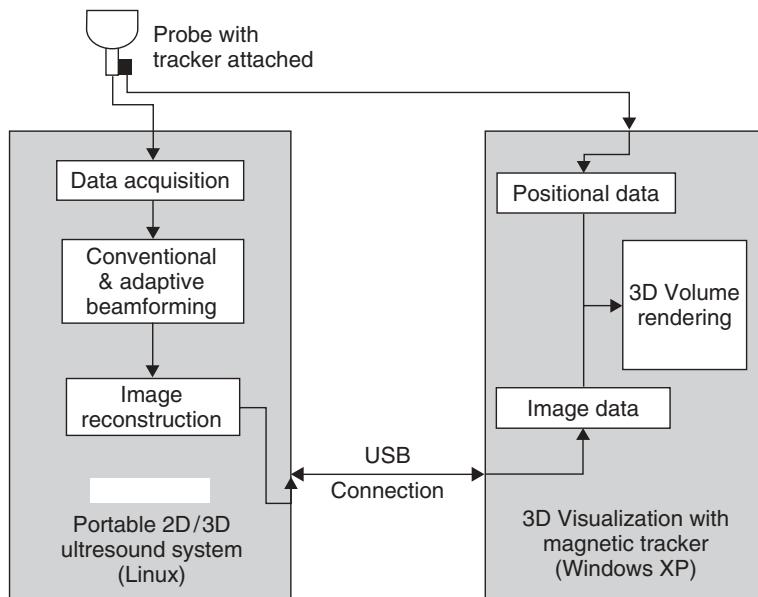
### 12.4.1 System Overview

The experimental configuration of the fully digital ultrasound imaging concept that was used to assess the fully digital 3D ultrasound beamforming structure of this chapter included two versions. The first was configured to be integrated with a linear phase array ultrasound probe, and it is depicted in Figure 12.11. This is a fully digital ultrasound system that includes the linear array probe (64 elements) and the two-node data acquisition unit including two mini-PCs that control the transmitting and receiving functions. This linear array ultrasound system was configured also to provide 3D images through volume rendering of the B-scan outputs. Figure 12.12 shows this 3D configuration, which is defined by the integration of the experimental linear phase array ultrasound system with a portable PC and a tracking device. A USB communication protocol allowed the transfer of the B scans (2D digital images) as inputs to the visualization software for 3D volume rendering installed in the portable PC. More specifically, the experimental linear array ultrasound system can provide 3D volumetric images from a series of B-scan (2D) outputs. The magnetic tracker system, shown schematically in Figure 12.12, provides the coordinates of the probe for each of the acquired image frames (B scans). This tracker provides translational ( $x$ ,  $y$ , and  $z$ ) as well as rotational coordinates with respect to the  $x$ ,  $y$ , and  $z$  axes.

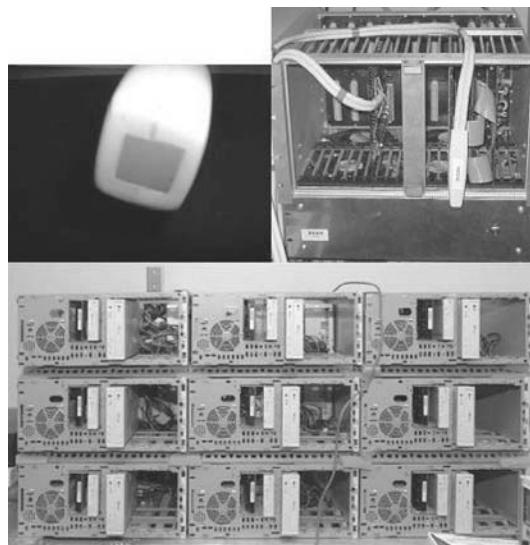
The second configuration is a fully digital planar phase array volumetric ultrasound imaging system, depicted in Figure 12.13. The multinode computing cluster that allows for an effective implementation of the 3D beamforming structure is shown at the lower part of this figure. The top left image in Figure 12.13 shows the planar phase array probe, and the top right image presents the data acquisition unit with the A/D and D/A peripherals controlled by the multinode cluster.



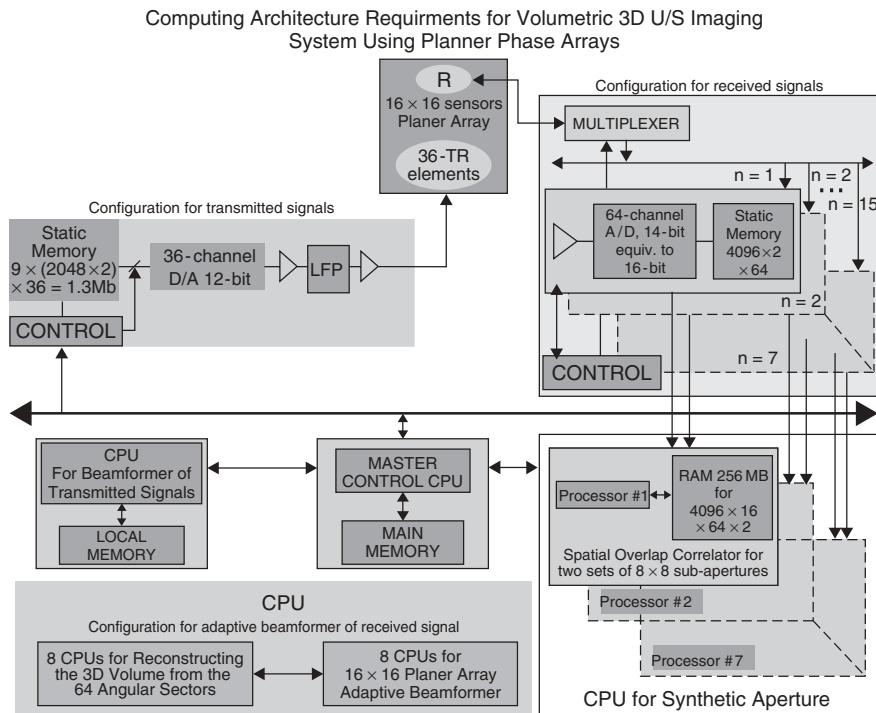
**Figure 12.11** Experimental linear phase array (64-element) ultrasound imaging system integrated with a laptop computer to provide visualization functionalities.



**Figure 12.12** Integration of the experimental linear phase array ultrasound system with a portable PC, a tracking device, and a USB communication protocol for 2D digital image transfers provided as inputs to Fraunhofer's visualization software for 3D volume rendering.



**Figure 12.13** Multinode computing cluster that allows the implementation of the parallel beam-forming structure of the experimental planar phase array ultrasound imaging system is shown at the lower part of this figure. The top left image shows the planar phase array probe and the top right image depicts the data acquisition unit with the A/D and D/A peripherals (with probe attached) controlled by the multinode cluster.



**Figure 12.14** Structure of the PC-based computing schematic representation of the main components including the data acquisition units of a fully digital real-time planar array ultrasound imaging system.

Implementation of the 3D beamforming structure and communication requirements, relevant with the system configuration of Figure 12.13, have been discussed already in [20].

Figure 12.14 shows a schematic representation of the main components of the fully digital real-time planar array ultrasound imaging system that summarizes the developments that have been presented in the previous sections. It depicts a 256-( $16 \times 16$ ) element phased array probe, an A/DC with 64-channel data acquisition unit that through multiplexing acquires time-series signals for the 256 channels and a computing architecture to process the acquired time series into ultrasound volumes, shown also in Figure 12.13. In addition the system uses a 36-channel D/AC to excite the center ( $6 \times 6$ ) transducers of the planar array during the illumination process. The transmit functionality that addresses the pulse design to illuminate at various depths simultaneously is addressed in a subsequent section. The interelement spacing of the probe is 0.4 mm in both directions. This combination forms the front end of the 3D ultrasound system that will support the transmit functions and the receiving functions required for the 3D beamforming. The probe is attached to the data acquisition unit via an interface card. This card provides the means of data flow into and out of the probe through the data acquisition system.

The computing cluster that implements the 3D beamformer software has already been introduced in [3, 20]. This is the multinode cluster that was designed to allow for easy implementation of the 3D beamformer algorithms—both conventional and

adaptive. The integrated hardware platform in Figures 12.13 and 12.14 brings together the planar array probe, the data acquisition unit for the planar array probe, and the multinode PC cluster.

The A/DC is well grounded and capable to sample the 64 channels with an equivalent 14-bit resolution and 33-MHz sampling frequency per channel. Moreover, the unit has dedicated memory and separate bus lines. The D/AC is capable to drive 36 channels with 12-bit resolution and 33-MHz sampling frequency. The period between two consecutive active transmissions is in the range of 0.2 ms. Moreover, the local memory of the D/AC unit has the capability to store the active beam time series with total memory size of 1.35 Mb, being generated by the main computing architecture for each focus depth and transferred to the local D/AC memory when the transmission–acquisition process begins.

The digitization process of the  $16 \times 4$  subapertures by the 14-bit 64-channel A/D unit provides the signals to a system of pin connectors–cables with suppressed cross-talk characteristics (minimum 35 dB). The sampling frequency is 33 MHz for each of the channels associated with a receiving single sensor. The multiplexer associated with the A/DC allows the sampling of the  $16 \times 4$  sensors of the planar array in four consecutive active transmissions to be able to digitize the  $16 \times 16$  planar array channels.

The computing architecture, discussed in [20], includes sufficient data storage capabilities for the sensor time series. The A/DC and signal conditioning modules of the data acquisition process and the communication interface are controlled through S/W drivers that form an integral part of the computing architecture.

It has been assessed that the ultrasound adaptive 3D beamforming structure, defined in [3], provides an effective beam-width size, which is equivalent to that of a two to three times longer aperture along azimuth and elevation of the deployed planar array. Thus, for the deployed receiving  $16 \times 16$  planar array, the adaptive beamformer's beamwidth characteristics will be equivalent with those of a  $(16 \times 2) \times (16 \times 2)$  size planar array. For example, the beam width of a receiving  $16 \times 16$  planar array with element spacing of 0.5 mm for a 3-MHz center frequency, is approximately  $7.4^\circ$ , with effective angular resolution by the adaptive beamformer in terms of beam-width size, to be less than  $3.7^\circ \times 3.7^\circ$ . As a result, the receiving adaptive beams along azimuth will have the following image resolution capabilities:

- For C scan and for depth of 10 cm, the  $3.7^\circ \times 3.7^\circ$  angular resolution sector corresponds to a  $(0.64 \text{ cm}) \times (0.64 \text{ cm}) = 0.41 \text{ cm}^2$  size of tissue resolution, or for depth of 5 cm to a  $(0.32 \text{ cm}) \times (0.32 \text{ cm}) = 0.1024 \text{ cm}^2$  size of tissue resolution.
- For B scan the line resolution will be equivalent to the wavelength of the transmitted center frequency, which is 0.5 mm.
- Thus, the volume resolution of the 3D adaptive beamforming structure will be equivalent to  $(10.24 \text{ mm}^2) \times (0.5 \text{ mm}) = 5.12 \text{ mm}^3$ , in 3D tissue size at a depth of 5 cm.

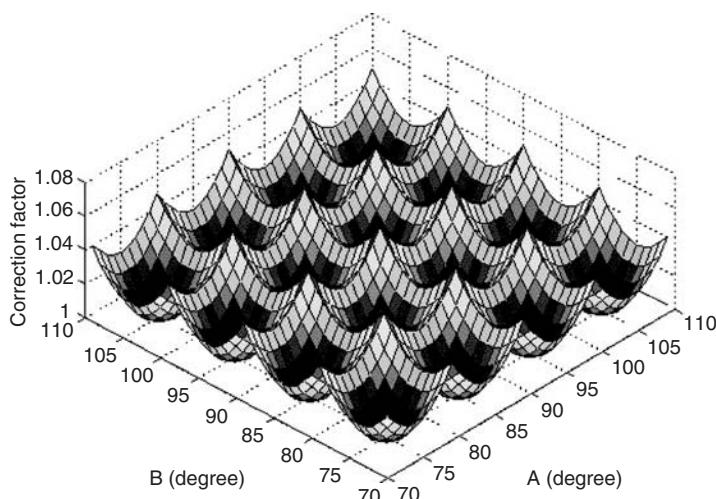
The concept of the energy transmission module to illuminate the entire volume of interest with a few firings has already been depicted in Figure 12.10. Here the volume is illuminated in subsectors. The transmitted signals are all broadband FM (chirp) signals. They are fired with interelement delays to allow the transmitted energy to be focused at specific regions in space (e.g., the space highlighted by the square shaded

areas of Figure 12.10). The energy transmission is done through the  $6 \times 6$  elements at the center of the array. The transmit patterns are loaded into the memory of the data acquisition unit and delivered to the probe via the D/AC portion of the unit when a trigger signal is received. In addition, FM pulses that occupy different nonoverlapping frequency regimes may be coded together to illuminate different focal depths with a single firing, as defined in Section 12.2.5.1. This means that it can be arranged so that one frequency regime can focus and illuminate the lower shaded square, and a second frequency regime the upper shaded square in Figure 12.10.

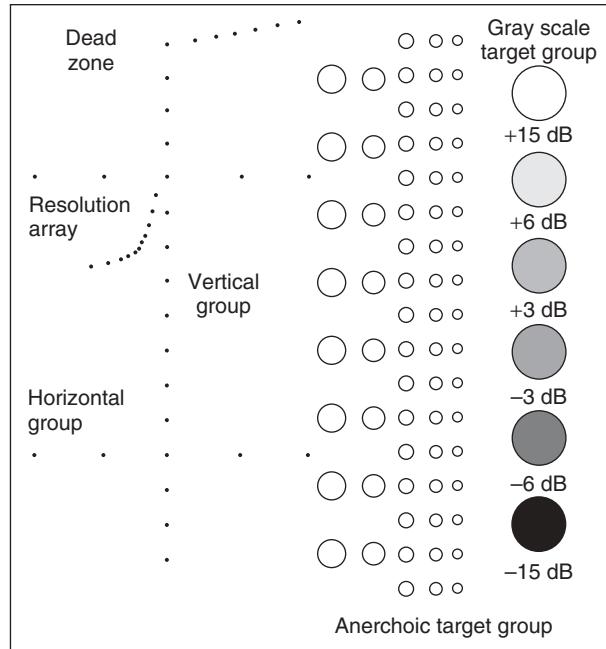
Suppose three focal depths  $d_1 < d_2 < d_3$  are desired. Then, three separate sets of transmit patterns are created. The first set of transmit patterns with the appropriate delay profiles that occupy the lowest frequency band are designed to focus at depth  $d_3$ . A second set of transmit patterns with delay profiles to focus at depth  $d_2$  are designed to occupy a second frequency band higher than the first and with no frequency overlap. Similarly, a third set of transmit patterns are designed to focus at focal depth  $d_1$ . This third set of patterns is designed to occupy the highest frequency band since they will illuminate the shallow regions of the medium of interest. When the design of the three sets of transmit patterns is complete, they are superimposed to create a single transmit pattern. This composite transmit pattern is used to provide the illumination as described in Figure 12.10. However, the requirement to use the smallest possible number of illumination beams leads to a nonuniform energy distribution in space. This requires the application of a linearization function to correct for this type of nonuniformity. A correction function is derived from the illuminating beam shapes and is used later for linearization of the results of the beamformer. An example of a linearization function is shown in Figure 12.15. This figure shows the correction function that would be applied to the output of a  $4 \times 4$  subsector illumination.

#### 12.4.2 Performance Results

Presented in this section are the image output results from the linear phase array (e.g., 64 elements, 2D/3D) and the planar phase array probes (e.g.,  $16 \times 16$ , 3D/4D). While



**Figure 12.15** Correction function for  $4 \times 4$  sector illumination pattern, as depicted in Figure 21.10.



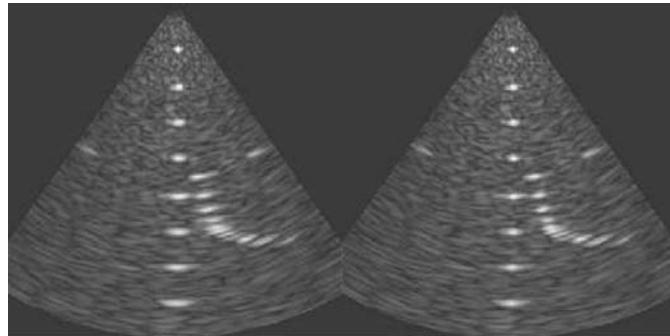
**Figure 12.16** Cross-sectional view of the experimental phantom.

numerous experiments were carried out, only a few typical image/volume outputs for both the adaptive and conventional image results are presented here. All these image results are from the standard ultrasound test target called “phantom,” the cross section of which is shown in Figure 12.16.

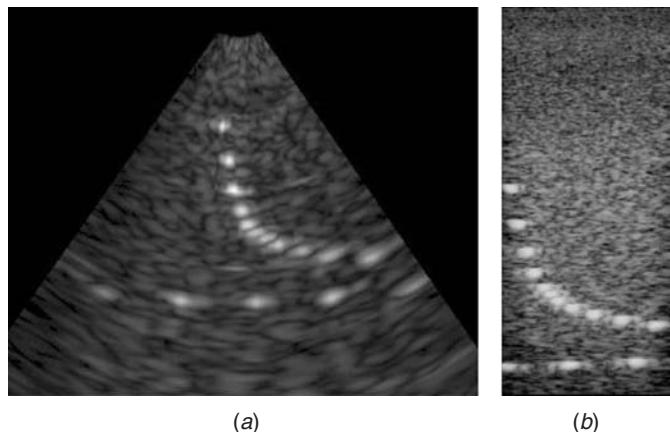
#### 12.4.2.1 Portable 2D/3D Experimental System with Linear Phase Array Probe

**B-Scan Results** Figure 12.17 shows two typical images from the portable 2D/3D system operating in B-scan mode. The left-hand image shows the image obtained using the conventional beamforming technique, and the right-hand image shows the image output from the adaptive beamformer. Both images are obtained by placing the probe on the top of the phantom just above the “dead zone” label. The phantom’s vertical row of reflectors and the curved arrangement are depicted in both images that show no “ghosting” or blurring of these strong reflectors. This assessment is consistent with the triggering mechanism characteristics being sufficient to create the synthetic aperture processing without having the need to implement the overlap–correlation software synthetic aperture technique, as this was substantiated in Section 12.2.2.

Figure 12.18 provides a comparison of the B-scan outputs from a commercial ultrasound imaging system and the experimental prototype of this investigation depicted in Figure 12.11. The images for both systems are obtained by placing the probes on the left side of the phantom just beside the “resolution array” label. Both systems use the same probe (64 channels). However, the commercial system operates at 4 MHz, while the experimental system in Figure 12.11 operates at 2 MHz. The left-hand side image (Fig. 12.18a) shows the output of the adaptive beamformer, while Figure 12.18b shows the image output from the commercial 4-MHz ultrasound system.



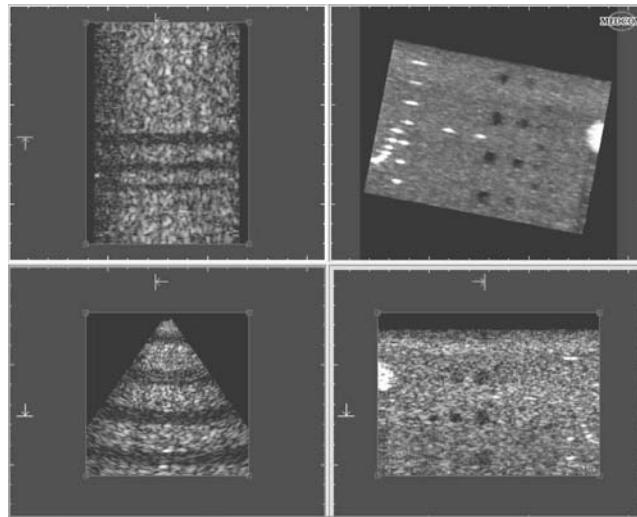
**Figure 12.17** B-scan image results for both the conventional (left-hand-side image) and adaptive (right-hand-side image) beamformers of the experimental linear phase array (64-element) ultrasound imaging system, depicted in Figure 21.11.



**Figure 12.18** (a) Shows the B-scan output of the adaptive beamformer for the same phantom as in Figure 21.17, illuminated at 2 MHz. (b) Shows the output of a commercial system using the same probe at 4 MHz.

It can be seen from the image results of Figure 12.18 that the adaptive beamforming scheme of this investigation performs quite as good, if not even better, with respect to the ultrasound commercial system, in terms of both resolution of the strong scatterers and noise in the bulk image. It should furthermore be considered that the images obtained with the prototype of this investigation have been acquired using a 2-MHz probe, whereas a 4-MHz probe has been employed for the image output (Fig. 12.18b) of the commercial system.

The transmission/acquisition schemes adopted by the experimental 2D/3D prototype system allow to cover a wide scanning angle ( $90^\circ$ – $120^\circ$ ), a characteristic which can be very useful in cardiac imaging applications. Furthermore, the deployment of low frequencies in the range of 2 MHz can achieve the deep penetration depths required in cardiologic ultrasound applications.

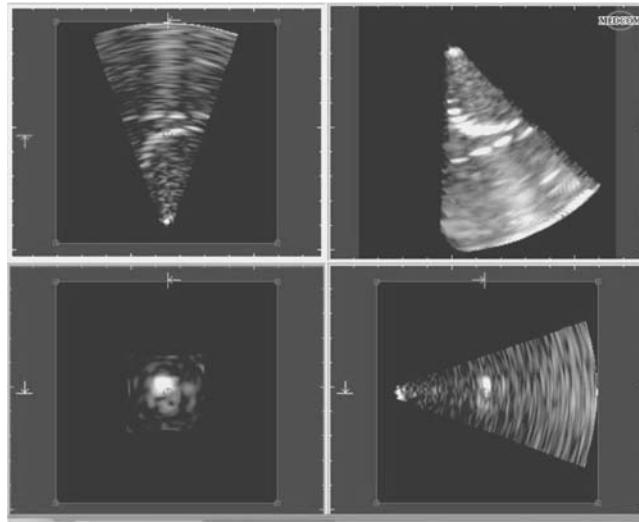


**Figure 12.19** 3D volume is taken along the top surface of the phantom shown in Figure 21.16. The top right panel shows the full reconstructed volume. The remaining three panels show three orthogonal views of the volume scanned.

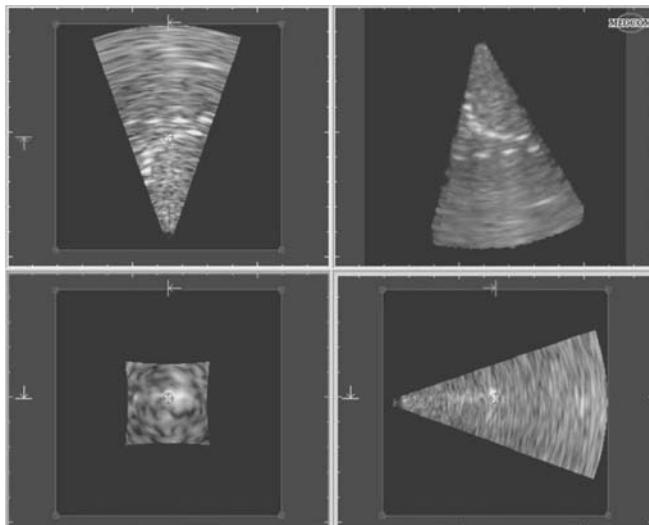
**Volumetric 2D/3D Imaging** The volumetric images created from the 2D/3D system are shown in Figure 12.19. This figure shows cross sections of the 3D output from volume redressing of B-scan images using the techniques discussed in [1]. In particular, Figure 12.19 shows the volume obtained using the standard ultrasound phantom of Figure 12.16. In this experiment the volume is taken along the top surface of the phantom. The top right panel in Figures 12.20 shows the full reconstructed volume. The remaining three panels show three orthogonal views of the volume scanned. Although the experimental 2D/3D system in Figure 12.11 includes a fully digital ultrasound technology with an advanced beamforming structure implemented in frequency domain, this system has been reduced to a portable size, compared to nowadays commercial ultrasound units.

**12.4.2.2 3D/4D Experimental System with Planar Phase Array Probe** The volumes created from the 3D/4D system, deploying the planar ( $16 \times 16$ ) phase array probe, are shown in Figures 12.20 and 12.21. Figure 12.20 shows the volume derived from the 3D conventional beamformer and Figure 12.21 presents the output of the 3D adaptive beamformer. The volumes in both figures are obtained by placing the probe on the left side of the phantom just beside the “resolution array” label. The top right panel in Figures 12.20 and 12.21 show the full reconstructed volume in each case. The remaining three panels show three orthogonal views of the volume scanned.

Like the portable ultrasound 2D/3D unit, the experimental 3D/4D system deploying a planar phase array probe has the capability of performing both conventional as well as adaptive beamforming, and both the modules are integrated into the parallel processing scheme discussed in this chapter. Furthermore, the 3D/4D system is also capable of frequency coding for multizone focusing and of applying deblurring algorithms for enhancing the image quality. Both experimental systems (e.g., 2D/3D, 3D/4D) show good performances in terms of scanning angle apertures, penetration depth, and image



**Figure 12.20** Top right panel shows the full reconstructed volume from the 3D conventional beamformer implemented on the  $16 \times 16$  planar array probe. The remaining three panels show three orthogonal views of the volume scanned, which represent reconstructed B scans of cross sections of the 3D output of the volumetric data of the planar array beamformer, discussed in Section 21.2. The reconstructed volume was for the standard ultrasound phantom of Figure 21.16. In this experiment the volume is taken along the top surface of the phantom.



**Figure 12.21** Results are for the 3D adaptive beamformer implemented on the same planar array time series being used also in Figure 21.20. The top right panel shows the full reconstructed volume from the 3D adaptive beamformer implemented on the  $16 \times 16$  planar array probe. The remaining three panels show three orthogonal views of the volume scanned, which represent reconstructed B scans of cross sections of the 3D output of the volumetric data of the planar array beamformer, discussed in Section 21.2. The reconstructed volume was for the standard ultrasound phantom of Figure 21.16. In this experiment the volume is taken along the top surface of the phantom.

quality. As expected, the adaptive beamforming scheme implemented into the 3D parallel architecture seems to allow for a higher image quality with respect to conventional beamforming for what concerns the axial and contrast resolutions.

## 12.5 CONCLUSION

The fully digital ultrasound system technology discussed in this chapter consists of a set of unique adaptive ultrasound beamformers [17, 18], a PC-based computing architecture, and a set of visualization tools [1], addressing the fundamental image resolution problems of current 3D ultrasound systems. The results of this development can be integrated into existing 2D and/or 3D ultrasound systems or they can be used to develop a complete stand-alone 3D ultrasound system solution.

It has been well established [1–3, 16, 19] that the existing limitations of medical ultrasound imaging systems in poor image resolution is the result of the very small size of deployed arrays of sensors and the distortion effects by the influence of the human body's nonlinear propagation characteristics. The ultrasound technology, discussed in this chapter, replaces the conventional (time-delay) beamforming structure of ultrasound systems with an adaptive beamforming processing configuration that has been developed for sonar array systems. The results of this development [2, 19] have demonstrated that these novel adaptive beamformers improve significantly (at very low cost) the image resolution capabilities of an ultrasound imaging system by providing a performance improvement equivalent to a deployed ultrasound probe with double aperture size. Furthermore, the portability and the low cost for the 3D ultrasound systems offer the options to medical practitioners and family physicians to have access of diagnostic imaging systems readily available on a daily basis.

At this point, however, it is important to note that in order to fully exploit the advantages of this digital adaptive ultrasound technology, its implementation in a commercial ultrasound system requires that the system has a fully digital design configuration consisting of A/DC and D/AC peripherals that would fully digitize the ultrasound probe time series, they will optimally shape the transmitted ultrasound pulses through a D/A peripheral, and they will use phase array linear or matrix ultrasound probes.

This kind of fully digital ultrasound configuration revises the system architecture of ultrasound devices and moves it away from the traditional hardware and implementation software requirements. Thus, implementation of the adaptive beamformer is a software installation on a PC-based ultrasound computing architecture with sufficient throughput for 3D and 4D ultrasound image processing.

In addition, the use of adaptive ultrasound beamformers provides significantly better image resolution than the traditional time-delay-based beamformers. Thus, a good image resolution can be achieved with less aperture size and sensors, thus decreasing the hardware costs of an ultrasound system.

In summary, the PC-based ultrasound computing architecture of this chapter, its adaptive 2D and 3D ultrasound beamforming structure, and the set of visualisation tools allow for a flexible cost-to-image quality adjustment. The resulting product can be upgraded on a continuous base at very low cost by means of software improvements and by means of hardware by taking advantage of the continuous upgrades and CPU performance improvements of the PC-based computing architectures. Thus, for a specific image resolution performance, a complete redesign or product upgrade can be

achieved by means of software improvement since the digital hardware configuration would remain the same.

## REFERENCES

1. G. Sakas, G. Karangelis, and A. Pommert, "Advanced applications of volume visualisation methods in medicine," in *Handbook on Advanced Signal Processing for Sonar, Radar and Medical Imaging Systems*, S. Stergiopoulos (Ed.), CRC Press, Boca Raton, FL, Mar. 2000.
2. A. Dhanantwari, S. Stergiopoulos, F. Bertora C. Parodi, P. Pellegratti, and A. Questa, "An efficient 3D beamformer implementation for real-time 4D ultrasound systems deploying planar array probes," in *Proceedings of the IEEE UFFC'04 Symposium*, Montreal, Canada, Aug. 2004.
3. S. Stergiopoulos, "Advanced beamformers," in *Handbook on Advanced Signal Processing for Sonar, Radar and Medical Imaging Systems*, S. Stergiopoulos (Ed.), CRC Press, Boca Raton, FL, Mar. 2000.
4. S. Tong, D. B. Downey, H. N. Cardinal, and A. Fenster, "A three-dimensional ultrasound prostate imaging system," *Ultrasound Med. Biol.*, vol. 22, pp. 735–746, 1996.
5. T. L. Elliot, D. B. Downey, S. Tong, C. A. Mclean, and A. Fenster, "Accuracy of prostate volume measurements in vitro using three-dimensional ultrasound," *Academic Radiol.*, vol. 3, pp. 401–406, 1996.
6. J. A. Jensen, "Field: A program for simulating ultrasound systems," *Med. Biol. Eng. Comput.*, vol. 34, pp. 351–353, 1996.
7. F. Lu, E. Milius, S. Stergiopoulos, and A. Dhanantwari, "A new towed array shape estimation scheme for real time sonar systems," *IEEE J. Oceanic Eng.*, vol. 28, no. 3, pp. 552–563, 2003.
8. T. R. Nelson, D. Downey, D. H. Pretorius, and A. Fenster, *Three-Dimensional Ultrasound*, Lippincott, Williams and Wilkins, Philadelphia, 1999.
9. P. A. Picot, D. W. Rickey, R. Mitchell, R. N. Rankin, and A. Fenster, "Three-dimensional colour Doppler imaging," *Ultrasound Med. Biol.*, vol. 19, pp. 95–104, 1993.
10. P. R. Detmer, G. Bashein, T. Hodges, K. W. Beach, E. P. Filer, D. H. Burns, and D. E. Strandness, "3D ultrasonic image feature localization based on magnetic scanhead tracking: Vitro calibration and validation," *Ultrasound Med. Biol.*, vol. 20, pp. 923–936, 1994.
11. S. Sherebrin, A. Fenster, R. Rankin, and D. Spence, "Freehand three-dimensional ultrasound: Implementation and applications," *SPIE: Phys. Med. Imag.*, vol. 2708, pp. 296–303, 1996.
12. S. W. Hughes, T. J. D. Arcy, D. J. Maxwell, W. Chiu, A. Milner, R. J. Saunders, and J. E. Shepperd, "Volume estimation from multiplanar 2D ultrasound images using a remote electromagnetic position and orientation," *Ultrasound Med. Biol.*, vol. 22, pp. 561–572, 1996.
13. D. F. Leotta, P. R. Detmer, and R. W. Martin, "Performance of a miniature magnetic position sensor for three-dimensional ultrasound imaging," *Ultrasound Med. Biol.*, vol. 23, pp. 597–609, 1997.
14. D. Downey and A. Fenster, "Three-dimensional ultrasound: A maturing technology," *Ultrasound Quarterly*, vol. 14, no. 1, pp. 25–39, 1998.
15. S. Stergiopoulos, "Optimum bearing resolution for a moving towed array and extension of its physical aperture," *J. Acoust. Soc. Am.*, vol. 87, no. 5, pp. 2128–2140, 1990.
16. S. Stergiopoulos, "Implementation of adaptive and synthetic aperture beamformers in sonar systems," *Proc. IEEE*, vol. 86, pp. 358–396, Feb. 1998.
17. S. Stergiopoulos and A. Dhanantwari, "High resolution 3D ultrasound imaging system deploying a multi-dimensional array of sensors and method for multi-dimensional

- beamforming sensor signals," Assignee: Defence R&D Canada, U.S. Patent: 6,482,160, issued Nov. 19, 2002.
18. S. Stergiopoulos and A. Dhanantwari, "High resolution 3D ultrasound imaging system deploying a multi-dimensional array of sensors and method for multi-dimensional beam-forming sensor signals," Assignee: Defence R&D Canada, U.S. Patent: 6,719,696, issued Apr. 13, 2004.
  19. A. Dhanantwari, S. Stergiopoulos, C. Parodi, F. Bertora, A. Questa, and P. Pellegretti, "Adaptive 3D beamforming for ultrasound systems deploying linear and planar array phased array probes," in *IEEE Conference Proceedings, IEEE International Ultrasonics Symposium*, Honolulu, Hawaii, Oct. 2003, pp. 5–8.
  20. F. Zhang, A. Bilas, A. Dhanantwari, K. N. Plataniotis, R. Abiprojo, and S. Stergiopoulos, "Parallelization and performance of 3D ultrasound imaging beamforming algorithms on modern clusters, in *Proceedings of the 16th International Conference on Supercomputing (ICS'02)*, New York, June 2002.



---

**PART III**

---

## **FUNDAMENTAL ISSUES IN DISTRIBUTED SENSOR NETWORKS**



## CHAPTER 13

---

# Self-Localization of Sensor Networks

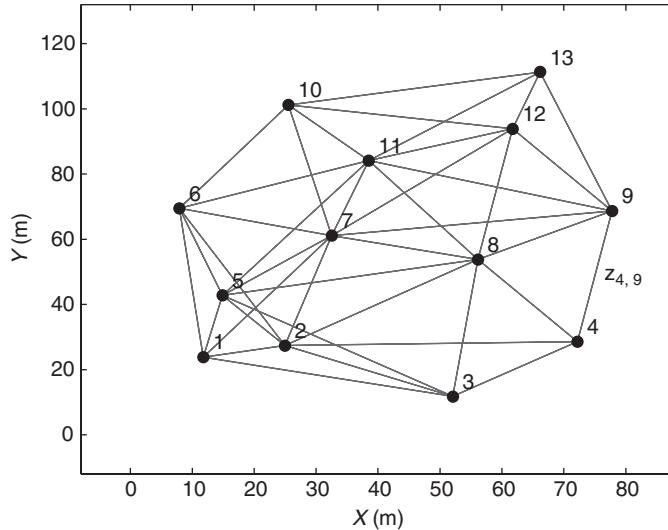
Joshua N. Ash and Randolph L. Moses  
Ohio State University, Columbus, Ohio

### 13.1 INTRODUCTION

Measurements from geographically distributed sensors enable signal processing algorithms to make inference in the spatial domain about the environment in which they are placed. Applications of inference and control in spatially distributed arrays range from precision agriculture, where sensors monitor the spatial variation in soil and crop conditions [1, 2], to noninvasive habitat monitoring [3, 4], to applications of object detection, classification, and tracking [5, 6]. Many other military and commercial applications, such as forest fire monitoring, inventory control, and structural monitoring, are given in the survey studies [7, 8].

In order to perform inference from spatially distributed sensors, knowledge of the sensor positions is typically required. With advances in micro-electro-mechanical systems (MEMS) and wireless communications, the size of sensor networks—as measured by both the number of nodes and size of deployment area—is rapidly increasing, with some current networks exceeding 1000 nodes [9]. Due to the large-scale and ad hoc deployment methodologies of such networks, an automated self-localization mechanism is a key enabling technology for modern sensor networks. This chapter will present popular localization algorithms while describing the foundational components of sensor network localization. Recent advances in localization technology are also considered along with performance bounds.

An outline of this chapter is as follows. In the remainder of this section we give a formal statement of the localization problem and provide a taxonomy of existing localization algorithms. Section 13.2 describes the most common types of measurements used for sensor localization and presents localization Cramér–Rao bounds (CRBs) for each measurement type. Specific localization algorithms are presented and demonstrated in Section 13.3. In Section 13.4 we describe a method of localization error analysis based on a decomposition of the location parameters into a relative shape component and a global placement component. Conclusions are given in Section 13.5.



**Figure 13.1** Illustration of the node localization problem. Each sensor node is represented by a vertex in the graph positioned at the sensor’s  $(x_i, y_i)$  location; edges  $\{z_{i,j}\}$  in the graph indicate the availability of an internode measurement, such as a distance or angle of arrival. The measurement set  $z = \{z_{i,j}\}$ , which need not contain all possible pairs, is combined with prior information in order to obtain coordinate estimates  $(\hat{x}_i, \hat{y}_i)$  of each node.

### 13.1.1 Self-Localization Problem

Sensor network self-localization (also called self-calibration and sensor localization) typically utilizes a set of internode measurements based on distance, time-of-arrival (TOA), time-difference-of-arrival (TDOA), received signal strength (RSS), or angle-of-arrival (AOA) observations of transmitted calibration signals. As illustrated in Figure 13.1, cooperative localization systems combine internode measurements, collected in a measurement vector  $z$ , with any available prior information in order to obtain coordinate estimates  $\{(\hat{x}_i, \hat{y}_i) : i \in (1, \dots, S)\}$  of the  $S$  constituent nodes of the network. We refer to this problem as absolute localization because the absolute locations of the nodes are sought. The prior information may be a priori knowledge of the locations of a subset of the sensors in the network, or it could be a more general constraint on the sensor positions, such as knowledge of the scene centroid. Probabilistic priors on node locations are also possible.

One common application of sensor networks is *source localization* (also called target localization), where the position of a foreign target is to be estimated. By considering the target as an additional unknown-location sensor and treating the positions of sensors making measurements of the target as known priors, source localization may be interpreted as a specific instance of the more general self-localization problem. As such, the theoretical results obtained for self-localization apply equally to source localization.

Source localization may be used to perform the most rudimentary form of sensor network localization. In this setting, a set of known-location sensors, referred to as anchor nodes (or beacon nodes), make direct measurements of each unknown-location node. Each unknown-location node is individually localized using a source localization

algorithm. These methods are typically called one-hop localization techniques [10] because in a measurement graph, where vertices denote sensors and edges denote measurements, the sensors are one-hop separated from the known-location anchors. In this chapter we consider the more general case consisting of both anchor-unknown and unknown-unknown measurements. This later scenario is often referred to as “network” or “cooperative” localization [11].

### 13.1.2 Algorithm Classifications

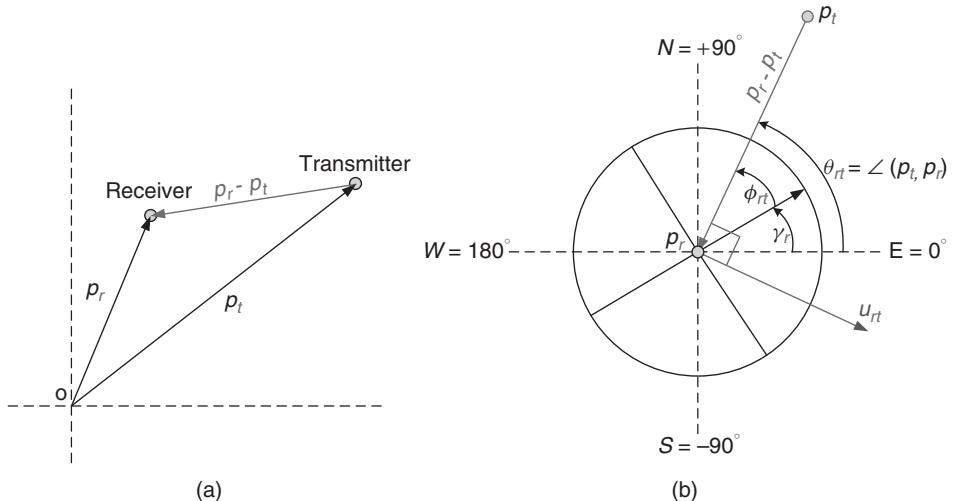
Sensor network localization has been well-studied in the research literature. Existing localization algorithms may be classified in a number of ways, including:

- *Centralized versus Distributed* In centralized processing, all of the local measurements are transmitted to a single node, called the fusion center, which computes the sensor locations for the entire network. While this simplifies processing, it introduces a single point of failure in the system and does not scale well with network size. In distributed algorithms, the estimation task is distributed over the network.
- *Relative versus Absolute* Relative localization algorithms only provide an estimate of a sensor network’s *shape*; that is, the  $(x, y)$  locations of the sensor nodes relative to one another but not anchored in an absolute reference frame. Absolute localization algorithms provide sensor position estimates with an absolute reference, latitude and longitude, for example.
- *Statistical Basis* Many localization algorithms have no statistical basis but produce correct estimates in the noiseless case. Statistically based algorithms consider the type of measurement noise and provide a tailored estimate. Classic techniques such as maximum-likelihood (ML) estimation and maximum a posteriori (MAP) estimation fall into this category.
- *Iterative versus Closed Form* Computational complexity plays an important part in localization algorithms because the number of sensors may be very large and the algorithms may be implemented on resource-constrained sensors. Iterative techniques are typically employed in algorithms requiring optimization of a complex nonlinear cost function. However, the high dimensionality of sensor localization can make these algorithms difficult to initialize and prone to local convergence problems. Closed-form algorithms do not suffer from these problems but rarely match the performance of iterative routines.
- *Measurement Type* Finally, most localization algorithms are specific to a particular type of measurements, such as intersensor distances or angles, and may be classified on that basis as well.

In Section 13.3 we will consider specific localization algorithms. Beforehand, however, we discuss the various measurement types and the fundamental performance limits that they impose.

## 13.2 MEASUREMENT TYPES AND PERFORMANCE BOUNDS

The set of measurements  $z$  used for localization may be any quantity that is position dependent such that inversion will yield an estimate  $\hat{\theta}$  of the sensor positions. Common



**Figure 13.2** Illustrations of basic measurement systems. (a) TOA and RSS measurements depend on the distance between the transmitter and receiver, which are described by position vectors  $\mathbf{p}_t$  and  $\mathbf{p}_r$ , respectively. (b) The AOA measurement  $\phi_{rt}$  between transmitter  $t$  and receiver  $r$  is made in node  $r$ 's local coordinate system, which is offset by an angle  $\gamma_r$  from the global coordinate system.

measurement types include TOA, AOA, and RSS. In these methods, a calibration signal is emitted, in turn, from each sensor in the network and received by a subset of its neighbors. The calibration signal may be any modality that is appropriate for the sensing application, hardware, and environmental conditions. Acoustic and radio-frequency (RF) calibration signals are common modalities.

### 13.2.1 Measurements

Time-of-arrival measurements depend on the transmitter–receiver distance (Fig. 13.2a) and the emission time of the calibration signal. For a receiving sensor node  $r$  at location  $\mathbf{p}_r = [x_r \ y_r]^T$  and a transmitting sensor  $t$  at location  $\mathbf{p}_t = [x_t \ y_t]^T$ , the measured quantity is

$$z_{t,r} = \tau_t + \frac{||\mathbf{p}_t - \mathbf{p}_r||_2}{c} + \eta_{t,r} \quad (\text{TOA}), \quad (13.1)$$

where  $c$  is the propagation velocity of the calibration signal,  $\tau_t$  is the emission time of the signal from node  $t$ , and  $\eta_{t,r}$  is a random variable modeling measurement noise. In this model,  $\tau_t$  and  $c$  are assumed known, and we estimate  $\{\mathbf{p}_i\}_{i=1}^S$  from the collection of available TOA measurement pairs  $z = \{z_{t,r}\}$ . TOA measurements are effectively measurements of range.

In AOA, measurements are of the form

$$z_{t,r} = \angle(p_t, p_r) - \gamma_r + \eta_{t,r} \quad (\text{AOA}), \quad (13.2)$$

where, as illustrated in Figure 13.2b,  $\angle(\mathbf{p}_t, \mathbf{p}_r)$  denotes the global frame angle between the receiver and transmitter of the calibration signal,  $\gamma_t$  is the orientation of the

receiver's local measurement coordinate system with respect to the global coordinate system, and  $\eta_{t,r}$  again represents measurement noise. The orientations  $\{\gamma_r\}$  are assumed known in the AOA model.

Received signal strength measurements are typically acquired on a logarithmic scale [e.g., power ratio in decibels referenced to 1 mW (dBm)] and follow a log-distance path loss model [12]:

$$z_{t,r} = P_t - L_{d_0} - 10\alpha \log_{10} \frac{\|\mathbf{p}_t - \mathbf{p}_r\|_2}{d_0} + \eta_{t,r} \quad (\text{RSS}), \quad (13.3)$$

where  $P_t$  is the transmit power of node  $t$ ,  $\alpha$  is the path loss exponent indicating the rate at which the signal is attenuated with distance,  $d_0$  is a short reference distance from the transmitter,  $L_{d_0}$  represents the loss (in decibels) of the signal to the reference distance  $d_0$ , and  $\eta_{t,r}$  is measurement noise. The variables  $\alpha$  and  $L_{d_0}$  are environmental parameters that must be measured. In the standard RSS model, the transmit power  $P_t$  is assumed known.

**13.2.1.1 Nuisance Parameters** Each of the three measurement models described above (TOA, AOA, and RSS) depend on knowledge of parameters that are often difficult to obtain in practice. When we relax the need for these quantities, they become unknown nuisance parameters that naturally degrade estimation performance but simplify the measurement acquisition process. For TOA, the emission times  $\{\tau_t\}$  may be difficult to estimate due to random delays through command and communication queues in the sensor node. When the emission times are unknown, the time of flight between the transmitter and receiver cannot be determined. In this case, the information bearing quantity is the difference between arrival times measured at distinct receivers corresponding to a common transmitter. As such, this measurement type is known as time difference of arrival (TDOA).

Correspondingly, if sensor orientations  $\{\gamma_r\}$  are not known in an AOA setting, a receiving node cannot determine the global bearing to a transmitter. It can, however, determine the bearing difference between two transmitters. This measurement type is known as angle difference of arrival (ADOA). ADOA-based localization is used, for example, when it is not desirable to equip each sensor with a digital compass.

In the context of signal strength measurements, the exact transmitter powers  $\{P_t\}$  may be unknown or not communicated. Without knowledge of the the transmit power, solving for the transmitter–receiver separation distance in (13.3) is ill-posed. However, signal strength differences between distinct receivers corresponding to a common transmitter (e.g.,  $z_{t_1,r_1} - z_{t_1,r_2}$ ) clearly eliminate the unknown transmit power and functionally depend on the two transmitter–receiver distances. Therefore, as in the previous two cases, it is the signal difference that bears the salient position information. This case is known as received signal strength difference (RSSD). The reference loss  $L_{d_0}$  need not be known in this case.

We thus have a unifying view of several common measurement types. TOA gives rise to TDOA when transmitter emission times  $\{\tau_t\}$  are unknown; AOA becomes ADOA when receiver orientations  $\{\gamma_r\}$  are unknown; and RSS becomes RSSD when the transmit powers  $\{P_t\}$  are unknown. Among other things, the Cramér–Rao bound analysis in the next section allows us to quantify the localization performance differences of these measurement approaches.

### 13.2.2 Cramér–Rao Bounds for Localization

The quality of a self-localization solution depends on a number of elements, including the type of measurements used (AOA, TDOA, etc.), the measurement noise distribution, the geometry of the true sensor positions, the connectivity of the measurement graph, the prior information on sensor locations, and the location estimation algorithm employed. By interpreting sensor localization as a parameter estimation problem, we may employ Cramér–Rao bounds (CRBs) to evaluate localization performance bounds in an estimator-independent way. This provides a benchmark for localization algorithms and allows us to explore the sensitivity of localization solutions to various network characteristics, such as the noise level and measurement connectivity.

The CRB formalism also allows us to evaluate the utility of the measurement types themselves with respect to one another. Clearly, for a given measurement type, lower noise results in improved location estimates. However, the comparison is less straightforward across measurement types. For example, given the alternatives of an acoustic TOA system that can measure arrival times with a standard deviation  $\sigma_t = 1 \text{ ms}$ , or an RF-based AOA system with angular measurement errors of  $\sigma_\theta = 3^\circ$ , it is not obvious which system will provide better performance for self-localization.

For a parameter vector  $\boldsymbol{\theta}$ , the CRB establishes a lower bound on the error covariance matrix for any unbiased estimator  $\hat{\boldsymbol{\theta}}$

$$E[(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T] \geq \Sigma_{\text{CRB}} \triangleq J^{-1}, \quad (13.4)$$

where the  $J$  denotes the Fisher information matrix (FIM), and for matrices  $A, B$ ,  $A \geq B$  means that  $A - B$  is positive semidefinite. The FIM is defined as [13]

$$J = E \left[ \frac{\partial}{\partial \boldsymbol{\theta}} \ln f_Z(\mathbf{z}; \boldsymbol{\theta}) \right] \left[ \frac{\partial}{\partial \boldsymbol{\theta}} \ln f_Z(\mathbf{z}; \boldsymbol{\theta}) \right]^T, \quad (13.5)$$

with  $f_Z(\mathbf{z}; \boldsymbol{\theta})$  denoting the probability density function of the observation vector  $\mathbf{z}$  as it depends on the parameter vector  $\boldsymbol{\theta}$ . The general form of the previous measurement models is

$$\mathbf{z} = \boldsymbol{\mu}(\boldsymbol{\theta}) + \boldsymbol{\eta} \in \mathbb{R}^M, \quad (13.6)$$

where  $\mathbf{z} = \{z_{t,r}\}$  is a vector of  $M$  measurements,  $\boldsymbol{\mu}$  is the mean of the observation, which is structured by the true parameter vector  $\boldsymbol{\theta} \in \mathbb{R}^N$ , and  $\boldsymbol{\eta} = \{\eta_{t,r}\}$  is zero-mean measurement noise.

Below we derive Fisher's information and the corresponding CRB for the six measurement types previously mentioned under the assumption of Gaussian noise,  $\boldsymbol{\eta} \sim \mathcal{N}(0, \Sigma_\eta)$ . For Gaussian errors, the FIM is shown to be [13]

$$J = \frac{\partial \boldsymbol{\mu}^T(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \Sigma_\eta^{-1} \frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}, \quad (13.7)$$

$$= G^T \Sigma_\eta^{-1} G \quad (13.8)$$

where  $G = \partial \boldsymbol{\mu}(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}^T$  is the  $M \times N$  Jacobian matrix of the measurement vector mean  $\boldsymbol{\mu}(\boldsymbol{\theta}) \in \mathbb{R}^M$ . In the following sections we evaluate the Jacobian  $G$  for the various measurement types. Afterwards,  $G$  may be substituted into (13.8) to obtain the FIM for the corresponding measurement type.

**13.2.2.1 Time of Arrival, Time Difference of Arrival, and Distances** Let  $\mathcal{M}$  denote the set of  $M$  ordered measurement pairs; that is,  $(r, t) \in \mathcal{M}$  if node  $r$  makes a measurement from a transmission originating at node  $t$ , and let  $\mathcal{M}(m) = (r_m, t_m) \in (1, \dots, S)^2$  denote the  $m$ th such pair.

For TOA, the  $m$ th measurement is of the form  $\mu(\theta)_m = \tau_{t_m} + \|\mathbf{p}_{t_m} - \mathbf{p}_{r_m}\|/c$  and the parameter vector includes all of the  $x$  and  $y$  coordinates of the  $S$  sensors:

$$\boldsymbol{\theta} = [x_1 \dots x_S \ y_1 \dots y_S]^T \in \mathbb{R}^N, \quad N = 2S, \quad (\text{TOA}). \quad (13.9)$$

The  $(m, n)$  element of  $G$  corresponds to the partial derivative of the  $m$ th measurement with respect to the  $n$ th parameter and is evaluated as

$$G_{m,n}^{\text{TOA}} = \frac{1}{c\|\mathbf{p}_{t_m} - \mathbf{p}_{r_m}\|} \times \begin{cases} -(x_{t_m} - x_{r_m}), & n = r_m, \\ (x_{t_m} - x_{r_m}), & n = t_m, \\ -(y_{t_m} - y_{r_m}), & n = S + r_m, \\ (y_{t_m} - y_{r_m}), & n = S + t_m, \\ 0, & \text{otherwise.} \end{cases} \quad (13.10)$$

The Fisher information matrix for TOA is then  $J^{\text{TOA}} = (G^{\text{TOA}})^T \Sigma_\eta^{-1} (G^{\text{TOA}})$ , and the corresponding CRB is equal to  $(J^{\text{TOA}})^{-1}$ . The FIM for distance measurements is the same as for TOA with the propagation velocity set to unity,  $c = 1$ .

In the case of TDOA, the measurements remain the same as in TOA, but the emission times are now unknown and the parameter vector becomes

$$\boldsymbol{\theta} = [x_1 \dots x_S \ y_1 \dots y_S \ \tau_1 \dots \tau_S]^T \in \mathbb{R}^N, \quad N = 3S, \quad (\text{TDOA}). \quad (13.11)$$

The Jacobian is

$$G^{\text{TDOA}} = [G^{\text{TOA}}, T], \quad (13.12)$$

where the  $m$ th row of  $T \in \{0, 1\}^{(M,S)}$  contains a 1 in the  $t_m$  position and is otherwise all zeros.

**13.2.2.2 Angle of Arrival and Angle Difference of Arrival** In the case of AOA, the  $m$ th measurement is of the form  $\mu(\theta)_m = \angle(\mathbf{p}_{t_m}, \mathbf{p}_{r_m}) - \gamma_{r_m}$  and the parameter vector is

$$\boldsymbol{\theta} = [x_1 \dots x_S \ y_1 \dots y_S]^T \in \mathbb{R}^N, \quad N = 2S, \quad (\text{AOA}). \quad (13.13)$$

The  $(m, n)$  element of the Jacobian evaluates to

$$G_{m,n}^{\text{AOA}} = \frac{1}{\|\mathbf{p}_{t_m} - \mathbf{p}_{r_m}\|^2} \times \begin{cases} (y_{t_m} - y_{r_m}), & n = r_m \\ -(y_{t_m} - y_{r_m}), & n = t_m \\ -(x_{t_m} - x_{r_m}), & n = S + r_m \\ (x_{t_m} - x_{r_m}), & n = S + t_m \\ 0, & \text{otherwise} \end{cases} \quad (13.14)$$

For ADOA the measurements remain the same as AOA, and the parameter vector is augmented with the sensor orientations, which are unknown under this model:

$$\boldsymbol{\theta} = [x_1 \dots x_S \ y_1 \dots y_S \ \gamma_1 \dots \gamma_S]^T \in \mathbb{R}^N, \quad N = 3S, \quad (\text{ADOA}). \quad (13.15)$$

The Jacobian becomes

$$G^{\text{ADOA}} = [G^{\text{AOA}}, R], \quad (13.16)$$

where the  $m$ th row of  $R \in \{0, -1\}^{(M,S)}$  contains a  $-1$  in the  $r_m$  position and is otherwise all zeros.

### 13.2.2.3 Received Signal Strength and Received Signal Strength Difference

Under RSS measurements, the  $m$ th measurement is of the form  $\mu(\theta)_m = P_{t_m} - L_{d_0} - 10\alpha \log_{10}(||\mathbf{p}_{t_m} - \mathbf{p}_{r_m}||/d_0)$  and the parameter vector is

$$\boldsymbol{\theta} = [x_1 \dots x_S \ y_1 \dots y_S]^T \in \mathbb{R}^N, \quad N = 2S, \quad (\text{RSS}). \quad (13.17)$$

The  $(m, n)$  element of the Jacobian evaluates to

$$G_{m,n}^{\text{RSS}} = \frac{10\alpha}{\ln(10) ||\mathbf{p}_{t_m} - \mathbf{p}_{r_m}||^2} \times \begin{cases} (x_{t_m} - x_{r_m}), & n = r_m, \\ -(x_{t_m} - x_{r_m}), & n = t_m, \\ (y_{t_m} - y_{r_m}), & n = S + r_m, \\ -(x_{t_m} - x_{r_m}), & n = S + t_m, \\ 0, & \text{otherwise.} \end{cases} \quad (13.18)$$

For RSSD, the transmit powers  $\{P_t\}$  are unknown and the parameter vector becomes

$$\boldsymbol{\theta} = [x_1 \dots x_S \ y_1 \dots y_S \ P_1 \dots P_S]^T \in \mathbb{R}^N, \quad N = 3S, \quad (\text{RSSD}). \quad (13.19)$$

The Jacobian for RSSD is an augmented version of that for RSS:

$$G^{\text{RSSD}} = [G^{\text{RSS}}, T], \quad (13.20)$$

where the matrix  $T$  is the same as in the TDOA case. While large path loss exponents  $\alpha$  are undesirable in communications because they reduce transmission range, we observe in (13.18) that  $G^{\text{RSS}}$ , and therefore the total Fisher information, actually improves with  $\alpha$ . This is because the sensitivity of received signal strength to changes in transmitter–receiver distance is increased for larger  $\alpha$ . In localization, the disadvantage of a large  $\alpha$  is that fewer sensors will receive the calibration signal due to the larger signal attenuation and that the signal-to-noise ratio (SNR) of the calibration signal will be reduced.

**13.2.2.4 Anchor Nodes as Location Constraints** Intersensor measurements, including all of those considered above, are invariant to the absolute position of the network as a whole. For example, time difference measurements would not change if the entire network was rigidly shifted and rotated. As such, these intersensor measurements provide information only about the relative shape of the network—not its absolute position or orientation. This causes the Fisher information matrix to be singular because the absolute position parameters in  $\boldsymbol{\theta}$  cannot be estimated uniquely. To remedy this situation, additional information must be supplied to the problem. The most common method is to specify the location of a small number of sensors beforehand. These a priori known-location sensors are referred to as anchor nodes and, in effect, “nail down” the relative scene to an absolute position and orientation. A minimum of

three anchor nodes is sufficient to resolve localization ambiguity for the measurement types considered in this chapter.

Because anchor nodes have known locations, their  $x, y$  coordinates are not random variables and should not be included in the parameter vector  $\theta$  to be estimated. Therefore, the columns of  $G$  corresponding to these known parameters should be removed before computing  $J = G^T \Sigma_{\eta}^{-1} G$ . The same effect is achieved by leaving  $G$  unaltered, computing the singular  $J$ , and then removing the rows and columns of  $J$  corresponding to the  $x$  and  $y$  coordinates of the anchor nodes. For a suitable number of measurements, the resulting FIM will be nonsingular and can be inverted to give the CRB.

Anchor nodes are a form of parametric constraints: Certain parameters in the estimation problem are constrained to have specific values. It is possible to use other, more general, constraints to resolve absolute localization ambiguity. For example, the location of the scene centroid could be specified along with the bearing angle to the first sensor. The impact of general constraints on the CRB is less straightforward than parameter elimination. A general formulation for constrained CRBs is given in [14] and considered in the localization context in [15].

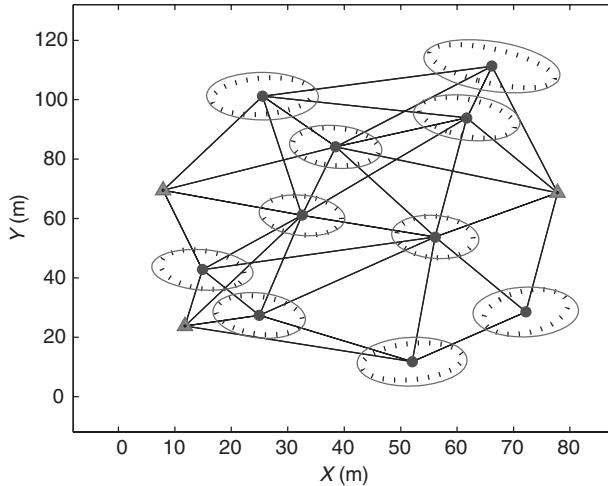
### 13.2.3 Numerical Examples of CRB Analysis

**13.2.3.1 Time of Arrival versus Time Difference of Arrival** In our first example, we use a CRB analysis to examine the performance difference between TOA and TDOA for localization of the sample network in Figure 13.1. Here we assume that nodes 1, 6, and 9 serve as anchor nodes, and we localize the remaining 9 sensors assuming that timing measurements from acoustic calibration signals ( $c = 343$  m/s) are measured. Zero-mean Gaussian errors with standard error  $\sigma_t = 20$  ms are assumed. As in Figure 13.1, only partial measurements are available—between sensors separated by less than 50 m. Using the results from the previous section, we calculate the CRB for TOA measurements  $\Sigma^{\text{TOA}}$  and for TDOA measurements  $\Sigma^{\text{TDOA}}$ . From a given CRB we may extract a  $2 \times 2$  covariance matrix representing the minimum estimation error for the  $x$  and  $y$  coordinate of each unknown-location sensor in the network. The  $2 \times 2$  covariance matrices for each sensor may be graphically represented as an uncertainty ellipse. In Figure 13.3 we plot  $3\sigma$  uncertainty ellipses for each sensor and for each measurement type. The TOA ellipses are strictly inside of the TDOA ellipses, indicating uniformly lower error as expected.

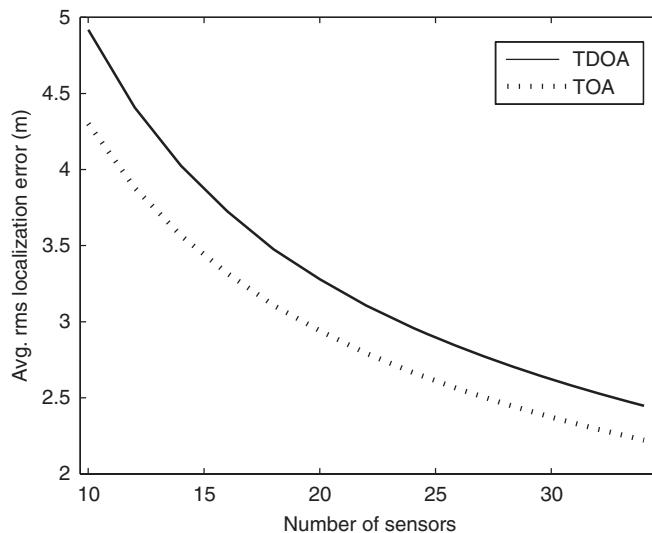
A common scalar error metric for localization performance is the scene root-mean-square (rms) distance between the true sensor positions and their estimates:

$$e_{\text{rms}} = \left[ \frac{1}{S} \sum_{i=1}^S E[\hat{d}_i^2] \right]^{1/2}, \quad (13.21)$$

where  $\hat{d}_i^2 = (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2$  is the squared distance between sensor  $i$  and its estimated position. For an unbiased estimator, the expected value  $E[\hat{d}_i^2] = \text{var}(x_i) + \text{var}(y_i)$  may be obtained by adding appropriate elements from the estimator error covariance matrix. If  $\Sigma$  represents the error covariance matrix of the position parameters (only), then  $e_{\text{rms}} = [1/S \text{tr } \Sigma]^{1/2}$ . When a CRB is substituted for the error covariance matrix, lower bounds on the rms error are obtained.



**Figure 13.3** CRB-derived  $3\sigma$  uncertainty ellipses for TDOA (—) and TOA (- -). TDOA error is strictly greater than TOA error, although only moderately so.



**Figure 13.4** Average rms localization error for TDOA and TOA for randomly distributed sensors in a 100-m  $\times$  100-m area.

The CRB can be an effective tool for studying trade-offs in localization system design, such as rms error versus number of anchors or versus measurement modality. Figure 13.4 is an example of such a study. Using the CRBs for TOA and TDOA, we plot in Figure 13.4 the average scene rms error as a function of the number of sensors in the network. For each given number of sensors, the average value of  $e_{\text{rms}}$  is taken over 200 random sensor deployments, consisting of uniformly distributed sensors in a 100-m  $\times$  100-m area. Four sensors were selected at random to serve as anchor nodes, and the measurement error and propagation velocity remain  $\sigma_t = 20$  ms and

$c = 343$  m/s, respectively. The positioning error of TDOA is seen to be approximately 12% greater than TOA for this scenario.

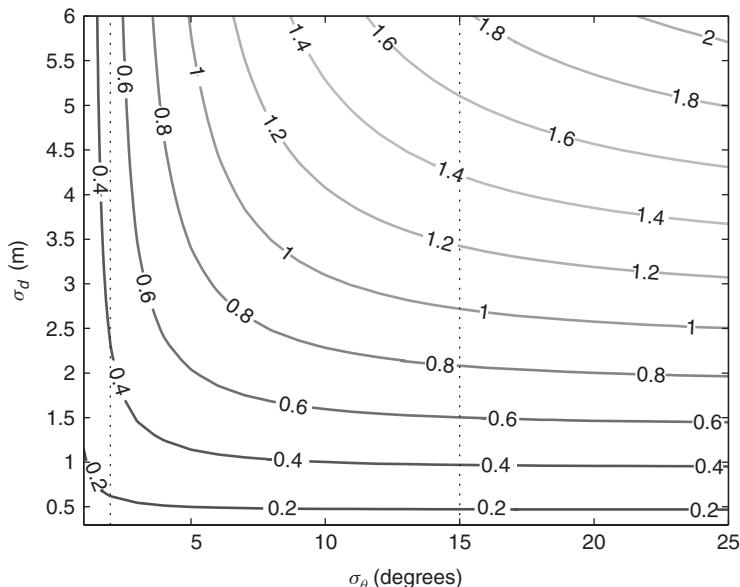
**13.2.3.2 Hybrid Measurement Systems** Cramér–Rao bound analysis also allows us to evaluate the utility of simultaneously using multiple types of measurements. Clearly the information present in two types of measurements is at least as great as one of them alone, and the estimation performance bound must be improved; however, it is not clear that the performance improvement justifies the additional hardware necessary for an additional measurement type and the necessarily greater communication and estimation complexity. By considering the Fisher information present in the joint measurement set, we may assess the utility of combining different forms of measurements.

When the measurement sets are statistically independent, the total Fisher information is given as the sum of that from each measurement type. For Fisher information matrices  $J_1$  and  $J_2$  from two different measurement types, we have

$$J_{\text{total}} = J_1 + J_2, \quad (13.22)$$

and the total CRB is evaluated as  $J_{\text{total}}^{-1}$ . The FIMs for each measurement type are evaluated exactly as in Section 13.2.2, although care must be taken to ensure that the parameter orderings are consistent if the nuisance parameters for each measurement type are different.

An example is provided in Figure 13.5 where we consider the fusion of distance and angle measurements for localization. For 16 nodes randomly deployed in



**Figure 13.5** Cramér–Rao bound analysis illustrating the utility of simultaneous distance and AOA measurements. Contours of equal rms localization error (in meters) are plotted as a function of the measurement error standard deviation of distance measurements ( $\sigma_d$ ) and the error standard deviation of angle measurements ( $\sigma_\theta$ ).

a 100-m  $\times$  100-m area, equal error contours are plotted as a function of the independent variables of the figure: the standard deviation of angle measurements  $\sigma_\theta$  and the standard deviation of available distance measurements  $\sigma_d$ . By considering large errors in one type of measurement, the performance of the other measurement modality alone can be inferred from the asymptotic nature of the contours. For example, without angle measurements, distance measurements with  $\sigma_d = 2$  m result in rms localization error of approximately 0.8 m. Following this contour into the region of large distance errors and low angular errors, we see that AOA measurements with  $\sigma_\theta \approx 2.8^\circ$  are required to achieve an equivalent level of localization performance.

The type of CRB analysis leading to Figure 13.5 is also useful in evaluating the utility of one type of measurement in the presence of another. For example, consider the two vertical cuts in Figure 13.5 at  $\sigma_\theta = 2^\circ$  and  $\sigma_\theta = 15^\circ$ . When  $\sigma_\theta = 15^\circ$ , any reduction in  $\sigma_d$  improves the localization estimates substantially. However, when  $\sigma_\theta = 2^\circ$ , the quality angle measurements dominate, and improved distance measurements have little improvement on overall scene estimation error until  $\sigma_d$  is less than 2 m, approximately.

### 13.3 LOCALIZATION ALGORITHMS

#### 13.3.1 Closed-Form Estimators

In this section we present two closed-form estimation routines that can be used for distance measurements and angle measurements. These algorithms produce relative estimates of the scene; that is, they do not use anchor nodes or other constraints to determine the actual scene translation or rotation. Rather, these absolute components are arbitrarily specified by the algorithm. However, as we will see, these relative scene estimates can be easily translated, rotated, and scaled for maximal agreement with a set of anchor nodes in a secondary step. The localization performance of these methods is not expected to be as good as iterative statistical methods, such as maximum likelihood; however, they are appealing because of their relatively low computational cost and lack of multimodal cost functions.

**13.3.1.1 Classical Multidimensional Scaling** Multidimensional scaling is a technique from statistics to construct point configurations of items from general item–item dissimilarity measures. The interpoint distances between the constructed point configuration forms a Euclidean distance matrix that approximates the original dissimilarity matrix. Here we describe one of the first practical methods to perform multidimensional scaling due to Torgerson [16] and Gower [17], known as classical scaling (CS), and consider its application to sensor localization.

Consider a set of points  $\mathbf{p}_i \in \mathbb{R}^p$ ,  $i = 1, \dots, S$  and the matrix  $D^{(2)}$ , which represents the squared Euclidean distance between all of the points,  $D_{i,j}^{(2)} = \|\mathbf{p}_i - \mathbf{p}_j\|^2$ . Let the  $S \times p$  coordinate matrix  $P$  be

$$P = \begin{bmatrix} \mathbf{p}_1^T \\ \mathbf{p}_2^T \\ \vdots \\ \mathbf{p}_S^T \end{bmatrix}, \quad (S \times p), \quad (13.23)$$

and define the  $S \times S$  inner product matrix  $B = PP^T$ . From the fact that  $B_{i,j} = \mathbf{p}_i^T \mathbf{p}_j$  we will establish a relation between  $D^{(2)}$  and  $B$ . Note that

$$D_{i,j}^{(2)} = \mathbf{p}_i^T \mathbf{p}_i + \mathbf{p}_j^T \mathbf{p}_j - 2\mathbf{p}_i^T \mathbf{p}_j \quad (13.24)$$

$$= B_{i,i} + B_{j,j} - 2B_{i,j}. \quad (13.25)$$

Therefore, the entire matrix  $D^{(2)}$  may be written

$$D^{(2)} = \mathbf{b}\mathbf{1}^T + \mathbf{1}\mathbf{b}^T - 2B, \quad (13.26)$$

where  $\mathbf{b} = \text{diag}(B)$  is an  $S \times 1$  vector consisting of the diagonal elements of  $B$ , and  $\mathbf{1}$  is an  $S \times 1$  vectors of ones.

Now, define the symmetric ‘‘centering matrix’’ as  $J = I - (1/S)\mathbf{1}\mathbf{1}^T$  and observe that  $JA$  is equal to the matrix  $A$  with the column means subtracted. Thus,  $JA$  is said to be column centered, and similarly  $AJ$  is row centered.

Finally, observe that

$$\begin{aligned} -\frac{1}{2}JD^{(2)}J &= -\frac{1}{2}J(\mathbf{b}\mathbf{1}^T + \mathbf{1}\mathbf{b}^T - 2B)J \\ &= -\frac{1}{2}J\mathbf{b}\mathbf{1}^T J - \frac{1}{2}J\mathbf{1}\mathbf{b}^T J + (JP)(JP)^T \\ &= -\frac{1}{2}J\mathbf{b}\mathbf{0}^T - \frac{1}{2}\mathbf{0}\mathbf{b}^T J + P_cP_c^T \\ &= B_c, \end{aligned} \quad (13.27)$$

where we have used the fact that centering a row (column) of all ones yields a row (column) of zeros. Here,  $P_c$  represents the centered coordinate matrix such that the average coordinate along any coordinate direction is zero, and  $B_c$  is the associated inner product matrix,  $B_c = P_cP_c^T$ .

Therefore, given an observation of squared distances  $D^{(2)}$ , we can form  $B_c = -\left(\frac{1}{2}\right)JD^{(2)}J$ , which can be factored by eigendecomposition

$$B_c = V\Lambda V^T = (V\Lambda^{1/2})(V\Lambda^{1/2})^T = \hat{P}_c\hat{P}_c^T \quad (13.28)$$

to give centered point estimates  $\hat{P}_c$  of the generating points  $P$ .

The scaling operation of multidimensional scaling comes by limiting the number of columns in  $\hat{P}_c$ , which controls the dimension of the estimated point configurations. This is accomplished by limiting the number of eigenvalue/eigenvector pairs that make up  $\hat{P}_c$ . Assuming ordered eigenvalues  $\lambda_1 \geq \dots \geq \lambda_S$  of  $B_c$ , with associated eigenvectors  $\{\mathbf{v}_1, \dots, \mathbf{v}_S\}$  the  $k$ -dimensional approximation is

$$\hat{P}_c(k) = [\mathbf{v}_1, \dots, \mathbf{v}_k] \text{ diag}(\lambda_1^{1/2}, \dots, \lambda_k^{1/2}). \quad (13.29)$$

In the sensor localization problem, we know, a priori that  $k = 2$  for localization in a plane, or  $k = 3$  for localization in space. The intersensor distances required for classical scaling may be obtained by inverting TOA or RSS measurements. When not all sensors make measurements of one another,  $D^{(2)}$  will have missing entries. The ISOMAP algorithm [18] extends CS to this scenario by replacing missing distances with the shortest path distance between associated nodes.

**13.3.1.2 Subspace-Based Multiangulation** This section describes a closed-form technique—dubbed robust angulation using subspace techniques (RAST)—for performing localization from AOA or ADOA measurements [19]. Each sensor’s location is described by a position vector  $\mathbf{p}_i \in \mathbb{R}^2$  in a global coordinate system. Each sensor  $r$  also maintains a local coordinate system centered at  $\mathbf{p}_r$  and rotated by an amount  $\gamma_r$  from the global system (Fig. 13.2b). In the global coordinate system, the AOA at receiving node  $r$ , of a transmission from node  $t$ , is

$$\theta_{rt} = \phi_{rt} + \gamma_r, \quad (13.30)$$

where  $\phi_{rt}$  is the measurement in  $r$ ’s local coordinate system.

We consider the AOA measurement model, where the orientation angles  $\{\gamma_r\}$  are known. In this case, the local angle measurements  $\{\phi_{rt}\}$  may be converted to global frame arrival angles  $\{\theta_{rt}\}$  via (13.30). From each angle  $\theta_{rt}$ , a unit vector

$$\mathbf{u}_{rt} = \begin{bmatrix} \sin \theta_{rt} \\ -\cos \theta_{rt} \end{bmatrix} \quad (13.31)$$

is formed that, as illustrated in Figure 13.2b, is orthogonal to the difference of the position vectors

$$\mathbf{u}_{rt}^\top (\mathbf{p}_t - \mathbf{p}_r) = 0. \quad (13.32)$$

Equation (13.32) forms the basis for the construction of a system of homogeneous equations that can be simultaneously solved in order to obtain position estimates of all sensors in the network.

As in Section 13.2.2.1, we let  $\mathcal{M}$  denote the set of  $M$  ordered measurement pairs. In expanding (13.32) into matrix form for all measurements, we first form the matrix  $U = \{U_{ij}\}$ , which is an  $M \times M$  block diagonal matrix, where each block is an element of  $\mathbb{R}^{2 \times 1}$ . The  $U_{ii}$  diagonal block of  $U$  is populated with the previously defined unit vectors  $U_{ii} = \mathbf{u}_{r',t'}$ , where  $(r', t') = \mathcal{M}(i)$ —the receiver-transmitter pair of the  $i$ th measurement. Additionally, we define a block matrix  $K = \{K_{ij}\}$  with  $M \times S$  blocks of size  $2 \times 2$ . The matrix  $K$  functions as a difference operator forming the position vector differences as in the second factor of (13.32). The nonzero elements of the  $i$ th row of  $K$  are populated as  $K_{it'} = I_2$  and  $K_{ir'} = -I_2$ , where  $I_2$  is the  $2 \times 2$  identity matrix and  $(r', t') = \mathcal{M}(i)$  as before. By stacking the position vectors of all sensors in the system  $\mathbf{p} = [\mathbf{p}_1^\top \mathbf{p}_2^\top \dots \mathbf{p}_S^\top]^\top$ , we obtain the following homogeneous linear system:

$$U^\top K \mathbf{p} = 0. \quad (13.33)$$

For example, a system with  $S = 3$  sensors and  $M = 4$  measurements may look like

$$\underbrace{\begin{bmatrix} \mathbf{u}_{21} & 0 & 0 & 0 \\ 0 & \mathbf{u}_{31} & 0 & 0 \\ 0 & 0 & \mathbf{u}_{32} & 0 \\ 0 & 0 & 0 & \mathbf{u}_{13} \end{bmatrix}}_{U^\top, M \times 2M} \underbrace{\begin{bmatrix} I_2 & -I_2 & 0_2 \\ I_2 & 0_2 & -I_2 \\ 0_2 & I_2 & -I_2 \\ -I_2 & 0_2 & I_2 \end{bmatrix}}_{K, 2M \times 2S} \underbrace{\begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}}_{p, 2S \times 1} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{M \times 1}. \quad (13.34)$$

The  $2S \times 1$  position vector  $\mathbf{p}$  we wish to find is clearly an element of the null space of  $U^\top K$ ,  $\mathcal{N}(U^\top K)$ ; however, the null space is not single dimensional. In general, the

null space is three dimensional, with every point in  $\mathcal{N}(U^T K)$  representing a particular scaling,  $x$  translation, and  $y$  translation of the desired generating point configuration  $\mathbf{p}$ . That is,  $\mathcal{N}(U^T K) = \text{span}(\mathbf{p}, \mathbf{v}_x, \mathbf{v}_y)$ , where the orthogonal vectors  $\mathbf{v}_x = [1 \ 0 \ 1 \ 0 \ \dots]^T$  and  $\mathbf{v}_y = [0 \ 1 \ 0 \ 1 \ \dots]^T$  represent  $x$  and  $y$  translations, respectively. The vectors  $\mathbf{v}_x, \mathbf{v}_y$  come directly from the difference operator  $K$ , ( $K\mathbf{v}_x = K\mathbf{v}_y = 0$ ) and reflect the fact that position vector differences, and therefore the measurements, do not depend on the particular  $x, y$  translation of the sensor positions.

The row-augmented form

$$A = \begin{bmatrix} U^T K \\ \mathbf{v}_x^T \\ \mathbf{v}_y^T \end{bmatrix} \in \mathbb{R}^{(M+2) \times (2S)} \quad (13.35)$$

will eliminate the translation vectors from the null space. If there are a sufficient number of measurements, meaning that degenerate cases such as only one or no measurements to a given sensor are excluded, then  $\mathcal{N}(A) = \text{span}(\mathbf{p})$ , and we can solve

$$A\mathbf{p} = 0 \quad (13.36)$$

using singular value decomposition (SVD). Let  $A = U_A \Sigma_A V_A^T$  be the SVD of  $A$ , then the (unit-norm) minimizing solution  $\hat{\mathbf{p}}$  of  $\|A\mathbf{p}\|$  is

$$\hat{\mathbf{p}} = V_A^{(2S)}, \quad (13.37)$$

where  $V_A^{(2S)}$  is the rightmost singular vector of  $A$ .

This algorithm reduces the position estimation problem to one of subspace identification and incurs a computational complexity equal to that of the single SVD required. The algorithm arbitrarily positions the centroid of the estimated scene at the origin and prescribes a scaling that results in  $\|\hat{\mathbf{p}}\| = 1$ . The extension to unknown orientations (ADOA) is presented in [19].

**13.3.1.3 Procrustes Alignment** It is often desirable to transform a set of relative node location estimates, such as those provided by MDS or RAST, into absolute coordinate estimates using anchor nodes. This process involves prescribing a rigid translation and rotation (and potentially a scaling) to the relative points such that the transformed estimates of the anchor positions closely align with the known anchor positions. Assuming, without loss of generality, that there are  $K$  anchor nodes that occupy the first  $K$  indices, we let  $\hat{P} = [\hat{x}_1 \hat{y}_1; \dots; \hat{x}_K \hat{y}_K] \in \mathbb{R}^{K \times 2}$  denote the estimated anchor positions from any relative estimation algorithm, and let  $P = [x_1 y_1; \dots; x_K y_K] \in \mathbb{R}^{K \times 2}$  denote the a priori known true positions. We seek a scalar scale factor  $s$ , a  $2 \times 2$  orthogonal rotation matrix  $R$ , and a  $2 \times 1$  translation vector  $\mathbf{t} = [t_x \ t_y]^T$  that minimizes

$$\|P - (s\hat{P}R + \mathbf{1}\mathbf{t}^T)\|_F, \quad (13.38)$$

where the norm is in the Frobenius sense.

The minimizing transformation parameters of (13.38) may be determined by SVD and are given as the solution to the extended orthogonal Procrustes problem, first solved by Schönemann and Carroll [20]. Let  $J$  be a centering matrix as defined in

Section 13.3.1.1, and consider the SVD  $U_p \Sigma_p V_p^T = P^T J \hat{P}$ . The desired transformation parameters are

$$R = V_p U_p^T, \quad (13.39)$$

$$s = \frac{\text{tr } P^T J \hat{P} R}{\text{tr } \hat{P}^T J \hat{P}}, \quad (13.40)$$

$$t = K^{-1}(P - s \hat{P} R)^T \mathbf{1}, \quad (13.41)$$

where  $\text{tr}$  is the matrix trace operator. If modifications to scale are not required, (13.40) may be ignored and  $s = 1$  substituted into (13.41) for optimal translation and rotation under this condition.

Once the optimal transformation parameters  $\{R, s, t\}$  are determined from the anchor nodes, the transformation may be applied to all nodes of the system.

### 13.3.2 Statistically Based Estimators

When some knowledge about the distribution of measurement noise is available, the localization routine may be designed to exploit this information in order to provide better position estimates. Maximum-likelihood (ML) estimation and maximum a posteriori (MAP) estimation, in the Bayesian case, are canonical examples of distribution-aware estimators. These are described in the localization context in this section.

**13.3.2.1 Maximum-Likelihood Estimation** For an observation vector  $z$  governed by a probability density function (pdf)  $f_Z(z; \theta)$  and parameter vector  $\theta$ , the ML estimate is  $\hat{\theta} = \arg \max_{\theta} f_Z(z; \theta)$ . If we follow the measurement model (13.6) and assume that the noise is Gaussian  $\eta \sim \mathcal{N}(0, \Sigma_{\eta})$ , the ML estimate takes the form

$$\begin{aligned} \hat{\theta} &= \arg \max_{\theta} \frac{1}{(2\pi)^{M/2} |\Sigma_{\eta}|^{1/2}} \exp\left(-\frac{1}{2}(z - \mu(\theta))^T \Sigma_{\eta}^{-1} (z - \mu(\theta))\right) \\ &= \arg \min_{\theta} (z - \mu(\theta))^T \Sigma_{\eta}^{-1} (z - \mu(\theta)). \end{aligned} \quad (13.42)$$

In (13.42), the parameter vector and the mean function  $\mu(\theta)$  depend on the particular measurement type. For example, elements of  $\mu(\theta)$  would take the form of (13.1) for TOA measurements. When the  $M$  measurements are jointly independent with individual noise variances  $\{\sigma_m^2\}$ , (13.42) reduces to

$$\hat{\theta} = \arg \min_{\theta} \sum_{m=1}^M \frac{1}{\sigma_m^2} (z_m - \mu_m(\theta))^2. \quad (13.43)$$

The ML technique also naturally accommodates hybrid measurement scenarios consisting of more than one measurement type, such as TOA and AOA. Taking these as examples, we may partition the observation vector as  $z = \{z^{\text{TOA}}, z^{\text{AOA}}\}$ , the mean vector as  $\mu = \{\mu^{\text{TOA}}, \mu^{\text{AOA}}\}$ , and the measurement counts as  $M = M_{\text{TOA}} + M_{\text{AOA}}$ . If all of the measurements are independent Gaussian, with  $\sigma_{\text{TOA}}^2$  and  $\sigma_{\text{AOA}}^2$  describing the

common variance of TOA and AOA measurements, respectively, then the total hybrid ML estimator becomes

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \frac{1}{\sigma_{\text{TOA}}^2} \sum_{m=1}^{M_{\text{TOA}}} (z_m^{\text{TOA}} - \mu_m^{\text{TOA}}(\boldsymbol{\theta}))^2 + \frac{1}{\sigma_{\text{AOA}}^2} \sum_{m=1}^{M_{\text{AOA}}} (z_m^{\text{AOA}} - \mu_m^{\text{AOA}}(\boldsymbol{\theta}))^2. \quad (13.44)$$

In general, (13.42)–(13.44) are nonlinear, nonconvex functions of  $\boldsymbol{\theta}$  that must be optimized numerically and are prone to local convergence problems. Low-complexity closed-form algorithms, such as those of Section 13.3.1 may be used as initialization routines for these iterative estimators.

**13.3.2.2 Maximum a Posteriori Estimation** We may consider the parameter vector  $\boldsymbol{\theta}$  as a random variable with a given prior pdf  $p_0(\boldsymbol{\theta})$ . For example, a Global Positioning System (GPS) receiver with known error statistics, on a subset of the nodes, could provide an informative prior on this subset of the sensor positions. This prior will be refined by updates from the intersensor calibration measurements.

From Bayes' rule, the posterior distribution is

$$f(\boldsymbol{\theta}|z) = \frac{f_Z(z|\boldsymbol{\theta})f_0(\boldsymbol{\theta})}{f_Z(z)}, \quad (13.45)$$

and we seek the MAP estimate that maximizes (13.45)

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} f_Z(z|\boldsymbol{\theta})f_0(\boldsymbol{\theta}). \quad (13.46)$$

In the deterministic setting, anchor nodes were used to disambiguate translations and rotations of the estimated scene. The prior probability fulfills this role in the Bayesian setting by providing the side information needed in order to obtain a unique solution to (13.46). The additional information from the prior also increases the Fisher information [21]:

$$J_T = J_M + J_P, \quad (13.47)$$

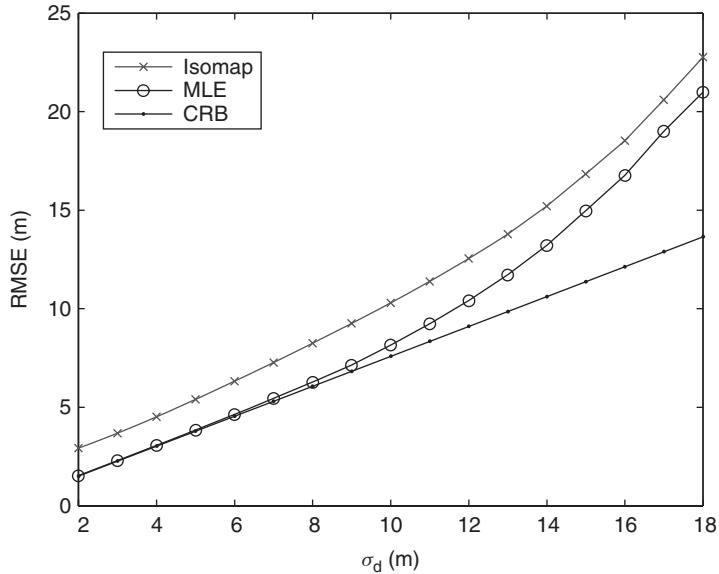
where  $J_T$  represents the total Fisher information,  $J_M = E_{\theta}[J_{\theta}]$  is the information from the measurements, and

$$J_P = E \left[ \frac{\partial}{\partial \boldsymbol{\theta}} \ln f_0(\boldsymbol{\theta}) \right] \left[ \frac{\partial}{\partial \boldsymbol{\theta}} \ln f_0(\boldsymbol{\theta}) \right]^T \quad (13.48)$$

is the contribution from the prior. In  $J_M = E_{\theta}[J_{\theta}]$ ,  $J_{\theta}$  is the appropriate measurement-dependent FIM from Section 13.2.2 as a function of  $\boldsymbol{\theta}$ , and  $E_{\theta}$  denotes expectation over the prior  $f_0(\boldsymbol{\theta})$ .

Similar to the deterministic case, the error covariance of any estimator  $\hat{\boldsymbol{\theta}}$ , including the MAP estimate (13.46), is bounded by the Fisher information matrix inverse,

$$E[(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T] \geq J_T^{-1}. \quad (13.49)$$



**Figure 13.6** Performance evaluation of distance-based estimators for the network in Figure 13.1. The maximum-likelihood estimate achieves the CRB for a portion of the noise range considered.

In this Bayesian setting, the expectation in (13.49) is over the measurements as well as the parameters, according to the prior distribution  $f_0(\boldsymbol{\theta})$ .

### 13.3.3 Example

As an example, we consider the localization of the sample network in Figure 13.1 using a sparse set of internode distance measurements contaminated with zero-mean Gaussian error  $\mathcal{N}(0, \sigma_d^2)$ . Localization is to be performed with an absolute reference frame using nodes 1, 6, and 9 as anchors. The first estimator we consider is the closed-form Isomap algorithm combined with the Procrustes algorithm for global alignment. The second estimator is the maximum-likelihood estimator (MLE). We initialize the iterative MLE with the output of Isomap+Procrustes. The scene rms errors [Eq. (13.21)] are shown in Figure 13.6 as a function of the noise standard deviation  $\sigma_d$ . For reference, the CRB is also shown.

As can be seen in Figure 13.6, the MLE achieves the CRB when the noise standard deviation is less than approximately 8 m. For larger values of  $\sigma_d$ , the initial value provided by the Isomap algorithm is frequently outside the attraction region of the global maximum of the likelihood function. In these cases, the minimization in (13.42) converges to a local minimum, not the global minimum, and the MLE begins to diverge from the CRB.

### 13.3.4 Related Localization Literature

This chapter has described several common source and sensor localization algorithms distilled from a large body of localization literature. Iterative methods based on MLEs

were derived in [22, 23], while other time- and distance-based localization algorithms are considered in [24–26]. Smith and Abel [27] provide closed-form estimates for source localization using range differences and time differences.

Because time synchronization can be difficult in distributed wireless sensor networks, many research studies for sensor localization consider RSS- and AOA-based methods because they do not require time synchronization. When the modality is RF, measurements can be made in the course of normal communication activities, which decreases the energy requirements for localization. ML estimation of sensor positions from RSS was considered in [23, 28] and the effect of random unknown transmit power in [29]. Estimation by spherical intersection, based on energy ratios, was considered in [30], while weighted least-squares methods for nonidentically distributed noise sources was considered in [31].

Because RSS measurements are prone to large noise, robust estimation techniques are important for reliable localization performance. Robust estimation based on signal strength orderings was considered in [32]. Similarly, some researchers have considered localization based on communication connectivity [33, 34]—essentially exploiting the idea that sensors within communication range must be in the same geographic proximity. Connectivity is effectively a one-bit quantization of RSS; this idea has been generalized to  $n$ -bit quantization in [35].

Simultaneous maximum-likelihood estimation of all sensor positions from AOA measurements was considered in [36]. Another centralized approach was taken in [33] utilizing semidefinite programming to estimate sensor locations from the intersection of AOA-derived constraint sets. A distributed AOA-based localization algorithm is presented in [37] where sensors first estimate their bearing to known-location beacons and then triangulate themselves.

Other distributed localization algorithms include [25, 38] where position estimates undergo successive refinements as neighboring nodes exchange and update their own position estimate. A distributed variant of MDS is derived and applied to sensor localization in [39]. Non-Gaussian posterior beliefs of sensor positions are computed in a distributed fashion in [40] using nonparametric belief propagation.

Finally, the survey articles [10, 11] provide a convenient entry point into additional localization literature.

### 13.4 RELATIVE AND TRANSFORMATION ERROR DECOMPOSITION

The typical metric for localization performance is the scene rms error (13.21), consisting of the sum of the expected squared  $x$  and  $y$  positioning errors for each sensor. The uncertainty ellipses described in Section 13.2.3.1 provide a finer means of error analysis in that they represent the per-node localization error and that they consider the correlation between the estimated  $x$  and  $y$  coordinates of each node—as seen by the angled error ellipses in Figure 13.3. However, any correlation between the estimated coordinates of different nodes is ignored. When this correlation is high, the error values can provide a misleading characterization of the localization performance. In this section we describe a refined error analysis that considers internode correlations by partitioning the total localization error into two components: relative shape error and global transformation error.

We refer to the relative arrangement of sensors—without regard to how the scene is translated, rotated, or scaled—as the *relative* configuration. Thus, two networks with

the same “shape” but with differing locations and orientations would be considered to have the same relative configuration. In absolute positioning, where the sensors are to be localized on a global coordinate system (e.g., latitude and longitude), the relative scene must be prescribed a particular location, orientation, and scaling. We refer to these elements as *transformation* parameters.

We are motivated to decompose the position parameters into relative and transformation components because these domains are informed upon by different information sources. Internode measurements provide information about the relative configuration but provide no information about transformation components. For example, internode distances are invariant to—and thus, do not inform upon—overall scene translation and rotation. This is the reason that the unconstrained Fisher information matrices in Section 13.2.2 are singular. Side information, such as anchor constraints or priors, inform upon the transformation domain and the relative domain. As such, we do not expect estimation performance to be the same in each domain. The error decomposition ideas of this section quantify these differences.

### 13.4.1 Definitions

Let  $\theta = [x_1 \ y_1 \ \dots \ x_S \ y_S]^T$  denote the absolute-location parameter vector of  $S$  sensors, and let  $\hat{\theta} = [\hat{x}_1 \ \hat{y}_1 \ \dots \ \hat{x}_S \ \hat{y}_S]^T$  be an estimate of  $\theta$ . If the estimator, taking as inputs internode measurements and side information, did not produce the optimal transformation parameters, then the total error

$$\epsilon = ||\theta - \hat{\theta}||^2 \quad (13.50)$$

may be further reduced by applying a rigid transformation to the previous estimate. Let

$$\alpha = [x_0 \ y_0 \ \phi_0 \ s_0] \quad (13.51)$$

denote transformation parameters corresponding to  $x$  translation,  $y$  translation, counterclockwise rotation, and scale, respectively. Initially, we assume that all four of the transformation components are not informed upon by the internode measurements, however, we will see later that a subset may be chosen for some applications. Let  $T_\alpha(\hat{\theta})$  denote the associated rigid transformation operation about the centroid:

$$T_\alpha(\hat{\theta}) = s_0 R_{\phi_0}(\hat{\theta} - \bar{\theta}) + \bar{\theta} + x_0 \mathbf{v}_x + y_0 \mathbf{v}_y, \quad (13.52)$$

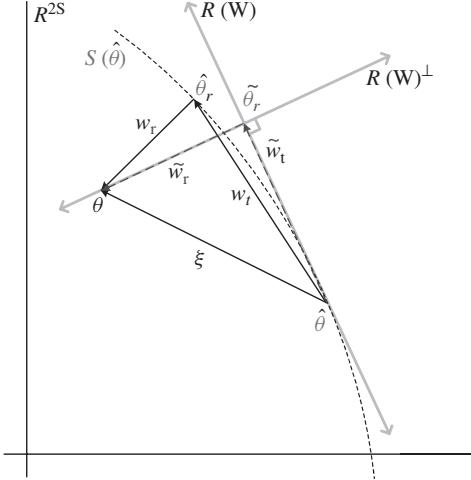
where  $\mathbf{v}_x = [1 \ 0 \ 1 \ 0 \ \dots]^T \in \mathbb{R}^{2S \times 1}$ ,  $\mathbf{v}_y = [0 \ 1 \ 0 \ 1 \ \dots]^T \in \mathbb{R}^{2S \times 1}$ , and the total rotation matrix  $R_{\phi_0}$  is composed of  $S$   $2 \times 2$  identical rotation matrices,  $R_{\phi_0} = \text{diag}([\Gamma_{\phi_0} \ \Gamma_{\phi_0} \ \dots \ \Gamma_{\phi_0}]) \in \mathbb{R}^{2S \times 2S}$ , where

$$\Gamma_{\phi_0} = \begin{bmatrix} \cos \phi_0 & -\sin \phi_0 \\ \sin \phi_0 & \cos \phi_0 \end{bmatrix}. \quad (13.53)$$

The vector  $\bar{\theta} = [\bar{x} \ \bar{y} \ \bar{x} \ \bar{y} \ \dots]^T \in \mathbb{R}^{2S \times 1}$  is composed of the centroid elements  $\bar{x} = S^{-1} \sum_i x_i$  and  $\bar{y} = S^{-1} \sum_i y_i$ .

Denote by  $\alpha_0 = \arg \min_\alpha ||\theta - T_\alpha(\hat{\theta})||^2$  the optimal transformation parameters, and let

$$\hat{\theta}_r = T_{\alpha_0}(\hat{\theta}) \quad (13.54)$$



**Figure 13.7** Geometric illustration of relative and transformation errors in the location parameter vector  $\theta$ . The manifold  $S(\hat{\theta})$  represents rigid translations and rotations of the coordinate estimates  $\hat{\theta}$ , and the point on  $S(\hat{\theta})$  closest to  $\theta$  represents the optimally transformed estimate,  $\hat{\theta}_r$ . The error vector  $\xi = \theta - \hat{\theta}$  may be decomposed into  $\xi = \mathbf{w}_r + \mathbf{w}_t$ , where  $\mathbf{w}_r = \theta - \hat{\theta}_r$  is the relative error vector and  $\mathbf{w}_t = \hat{\theta}_r - \hat{\theta}$ .  $\mathbf{w}_t$  and  $\mathbf{w}_r$  may be approximated, respectively, by  $\tilde{\mathbf{w}}_t$  and  $\tilde{\mathbf{w}}_r$ , the projections of the error vector  $\xi$  onto the transformation subspace  $\mathcal{R}(W)$  and the relative subspace  $\mathcal{R}(W)^\perp$ .

denote the transformed scene estimate. The optimal  $\alpha_0$  may be found using the Procrustes method of Section 13.3.1.3. As the translation and rotation components of  $\hat{\theta}$  have been optimally corrected in  $\hat{\theta}_r$ , the error

$$\epsilon_r \triangleq \|\theta - \hat{\theta}_r\|^2 \quad (13.55)$$

represents the relative error, or the error in the “shape” of the estimate  $\hat{\theta}$ . We define the transformation error  $\epsilon_t$  as the portion of the total error due to miss estimation of the transformation parameters

$$\epsilon_t \triangleq \epsilon - \epsilon_r. \quad (13.56)$$

These errors are illustrated graphically in Figure 13.7. Also depicted in this figure is the four-dimensional nonlinear manifold  $S(\hat{\theta})$  of equivalent shapes given by all possible translations, rotations, and scalings of  $\hat{\theta}$ . The point on  $S(\hat{\theta})$  closest to  $\theta$  is  $\hat{\theta}_r$ .

A linear subspace interpretation of the relative-absolute decomposition is obtained by linearizing  $S(\hat{\theta})$  through the transformation operator  $T_\alpha$ . This linearization is appropriate for small-deviation analysis such as the CRB and for analyzing algorithm performance in high SNR settings. The subspace interpretation allows us to simplify the expressions for  $\epsilon_r$ ,  $\epsilon_t$  and their expected values. A first-order Taylor series approximation of  $T_\alpha$  yields

$$T_\alpha(\theta) \approx \theta + x_0 \mathbf{v}_x + y_0 \mathbf{v}_y + \phi_0 \mathbf{v}_\phi + (1 - s_0) \mathbf{v}_s, \quad (13.57)$$

where

$$\mathbf{v}_x = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}, \quad \mathbf{v}_y = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ \vdots \end{bmatrix}, \quad \mathbf{v}_\phi = \begin{bmatrix} -(y_1 - \bar{y}) \\ (x_1 - \bar{x}) \\ -(y_2 - \bar{y}) \\ (x_2 - \bar{x}) \\ \vdots \end{bmatrix}, \quad \mathbf{v}_s = \begin{bmatrix} (x_1 - \bar{x}) \\ (y_1 - \bar{y}) \\ (x_2 - \bar{x}) \\ (y_2 - \bar{y}) \\ \vdots \end{bmatrix}. \quad (13.58)$$

We define the approximation  $\tilde{\boldsymbol{\theta}}_r$  of  $\hat{\boldsymbol{\theta}}_r$  as

$$\tilde{\boldsymbol{\theta}}_r = \hat{\boldsymbol{\theta}} + W\hat{\boldsymbol{\beta}}, \quad (13.59)$$

where  $\hat{\boldsymbol{\beta}} = [\hat{\beta}_x \hat{\beta}_y \hat{\beta}_\phi \hat{\beta}_s]^T$  are the minimizing transformation coefficients

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} \|\boldsymbol{\theta} - (\hat{\boldsymbol{\theta}} + W\boldsymbol{\beta})\|^2 \quad (13.60)$$

$$= W^T(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}), \quad (13.61)$$

and  $W$  is the orthonormal matrix

$$W = \left[ \frac{\mathbf{v}_x}{\|\mathbf{v}_x\|}, \frac{\mathbf{v}_y}{\|\mathbf{v}_y\|}, \frac{\mathbf{v}_\phi}{\|\mathbf{v}_\phi\|}, \frac{\mathbf{v}_s}{\|\mathbf{v}_s\|} \right]. \quad (13.62)$$

The range of  $W$ ,  $\mathcal{R}(W)$ , is a linear subspace approximation of the transformation space  $S(\hat{\boldsymbol{\theta}})$  while the orthogonal complement  $\mathcal{R}(W)^\perp$  represents the subspace of relative configurations. These subspaces facilitate calculating approximations of the transformation and relative errors, which may now be interpreted as projections of the total error vector  $\xi$  onto  $\mathcal{R}(W)$  and  $\mathcal{R}(W)^\perp$ . Let  $P_W = WW^T$  and  $P_W^\perp = I - WW^T$  denote the projection operators onto  $\mathcal{R}(W)$  and  $\mathcal{R}(W)^\perp$ , respectively.

Thus, as depicted in Figure 13.7, the linear approximation  $\tilde{\epsilon}_r$  of the relative error  $\epsilon_r$  is given by

$$\begin{aligned} \tilde{\epsilon}_r &\triangleq \|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}_r\|^2 \\ &= \|P_W^\perp \xi\|^2, \end{aligned} \quad (13.63)$$

and the corresponding linear approximation  $\tilde{\epsilon}_t$  of the transformation error is given as

$$\begin{aligned} \tilde{\epsilon}_t &\triangleq \epsilon - \tilde{\epsilon}_r \\ &= \|\xi\|^2 - \|P_W^\perp \xi\|^2 \\ &= \|P_W \xi\|^2. \end{aligned} \quad (13.64)$$

### 13.4.2 Expected Error

For an unbiased estimator  $\hat{\boldsymbol{\theta}}$ , we may express the expected values of the three estimation errors  $\epsilon$ ,  $\tilde{\epsilon}_r$ , and  $\tilde{\epsilon}_t$  in terms of the estimator covariance matrix  $\Sigma_{\hat{\boldsymbol{\theta}}} = E[\xi \xi^T]$ . Let

$$\Sigma_t = E[\hat{\boldsymbol{\beta}} \hat{\boldsymbol{\beta}}^T] = W^T \Sigma_{\hat{\boldsymbol{\theta}}} W \quad (13.65)$$

denote the covariance matrix of the transformation coefficients, and let

$$\Sigma_r = E[(P_W^\perp \xi)(P_W^\perp \xi)^T] = P_W^\perp \Sigma_{\hat{\theta}} P_W^\perp \quad (13.66)$$

denote the covariance matrix of the error in the relative subspace  $\mathcal{R}(W)^\perp$ . Then, the expected errors are

$$e \triangleq E[\epsilon] = E[\xi^T \xi] = \text{tr} \Sigma_{\hat{\theta}}, \quad (13.67)$$

$$e_r \triangleq E[\tilde{\epsilon}_r] = E[(P_W^\perp \xi)^T (P_W^\perp \xi)] = \text{tr} \Sigma_r, \quad (13.68)$$

$$e_t \triangleq E[\tilde{\epsilon}_t] = E[(P_W \xi)^T (P_W \xi)] = \text{tr} \Sigma_t, \quad (13.69)$$

and, as desired, the sum of the expected component errors equals the total:

$$e = \text{tr}[P_W^\perp, P_W]^T \Sigma_{\hat{\theta}} [P_W^\perp, P_W] \quad (13.70)$$

$$= e_r + e_t, \quad (13.71)$$

where we have made use of the fact that  $[P_W^\perp, P_W][P_W^\perp, P_W]^T = I$ . Lower bounds on the expected total, relative, and transformation errors may be obtained by substituting a localization CRB (deterministic or Bayesian, as appropriate) for  $\Sigma_{\hat{\theta}}$  in (13.65)–(13.69).

In the above, we assumed that translation, rotation, and scale were not informed upon by the internode measurements. However, as summarized in Table 13.1, only ADOA measurements are completely noninformative for all four of these transformation components. The other measurement types are only noninformative about three of the four parameters. As such, it may be desirable to restrict the space that we attribute to transformation error. This is easily controlled by limiting the columns in the matrix  $W$  (13.62). For example, distance measurements *are* informative about the network size (scale), and we would therefore remove the column associated with  $v_s$  from  $W$ .

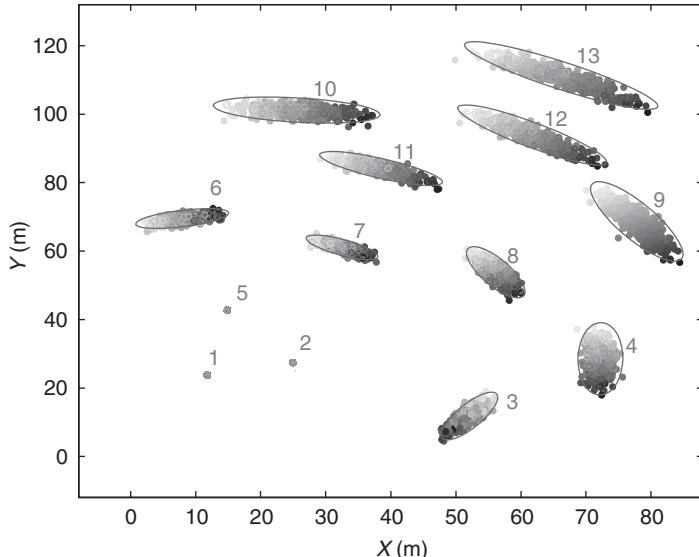
### 13.4.3 Examples

**13.4.3.1 Error Decomposition** We demonstrate the relative transformation error decomposition ideas applied to the localization of the sensor array depicted in Figure 13.1. We assume that the internode measurement range is limited to 50 m and that only a partial set of measurements, as depicted by the edges of the measurement graph in Figure 13.1, are available. We assume that the measurements are internode

**TABLE 13.1 Measurement Types and Their Associated Invariant Transformation Quantities**

Measurement Type	Noninformed Parameters
Distances	$x_0, y_0, \phi_0$
TOA & TDOA	$x_0, y_0, \phi_0$
RSS & RSSD	$x_0, y_0, \phi_0$
AOA	$x_0, y_0, s_0$
ADOA	$x_0, y_0, s_0, \phi_0$

*Note:*  $x$  translation ( $x_0$ ),  $y$  translation ( $y_0$ ), rotation ( $\phi_0$ ), and scale ( $s_0$ ).



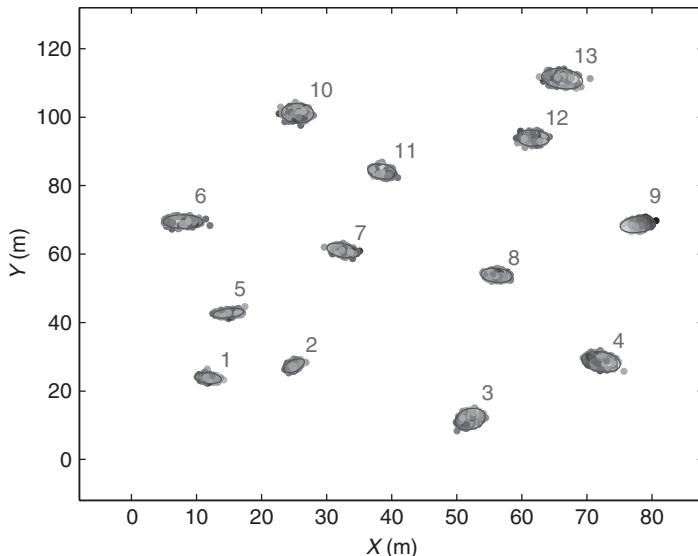
**Figure 13.8** Total error: Scatter plots of ML estimates of absolute positions exhibit large rotational uncertainty, as predicted by the  $3\sigma$  ellipses of the constrained CRB (—). Color coding of estimates illustrates high correlation between sensors.

distances corrupted by independent additive Gaussian noise with zero mean and standard deviation  $\sigma_d = 2.0$  m. Sensors 1, 2, and 5 serve as anchor nodes.

Figure 13.8 illustrates a scatter plot of 500 MLEs of the absolute positions of the sensors (corresponding to 500 simulated realizations of the measurement set). The elliptical shape of the point clusters indicates obvious correlation between the  $x$ - and  $y$ -coordinate estimates for a given node; however, there is also significant correlation across nodes. To demonstrate this, each of the 500 thirteen-node estimates is plotted with a different gray scale intensity. For a given scene estimate, the shading of all 13 nodes is the same and was determined by the position of the estimate of node 9 (chosen arbitrarily) relative to the principle axis of cluster 9. This results in the smooth shading seen for cluster 9 in Figure 13.8. If the estimates of the other node locations were uncorrelated, their cluster estimates would appear randomly colored. However, the general trend of the shading is seen in the other clusters as well. This suggests that the variability in the *shape* of the estimated scene is actually lower than what is implied by the size of the absolute scatter plots. The error decomposition provides a means of quantifying this qualitative observation.

The average empirical total error  $\epsilon$ , calculated as the average of  $\epsilon$  over the 500 estimates, was equal to  $168.0 \text{ m}^2$ . Using the anchor constraints and knowledge of the measurement type, we calculate the localization CRB,  $\Sigma_{\text{CRB}}$ , and determined the bound on total error to be  $\text{tr}\Sigma_{\text{CRB}} = 153.0$  m. The CRB-derived  $3\sigma$  uncertainty ellipse for each sensor is also plotted on Figure 13.8. While these ellipses are good predictors of the total error distribution, they do not give a complete picture of estimation error. In particular, they fail to capture the relative error and correlation structure observed above.

For each of the 500 scene estimates, we determine the optimally transformed relative estimate  $\hat{\theta}_r$  as in (13.54) and show the relative scatter plots of  $\{\hat{\theta}_r\}$  in Figure 13.9.



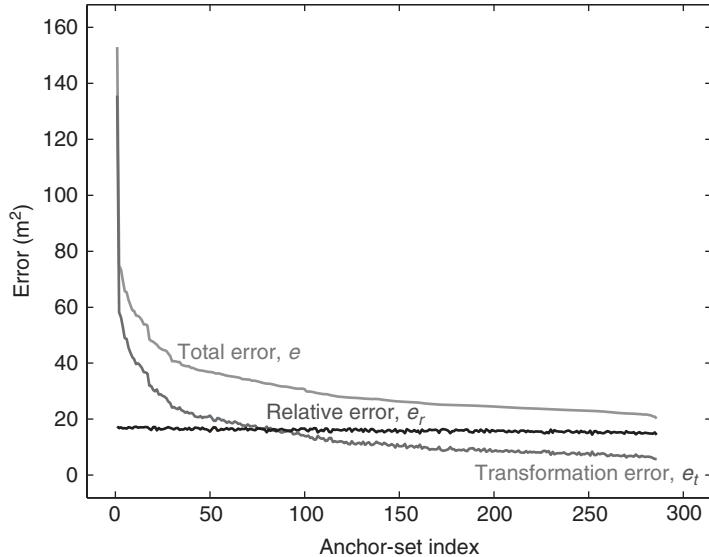
**Figure 13.9** Relative error: Large rotational uncertainty of Figure 13.8 is not seen in the optimally transformed relative estimates,  $\{\hat{\theta}_r\}$ . The  $3\sigma$  uncertainty ellipses (—) of the relative bound  $\Sigma_r$  accurately describe the empirical relative error.

The average empirical relative error  $\epsilon_r$  was calculated to be  $18.0 \text{ m}^2$ ; this compares favorably to the relative portion of the constrained CRB,  $\text{tr} P_W^\perp \Sigma_{\text{CRB}} P_W^\perp = 17.4 \text{ m}^2$ . We also see in Figure 13.9 that the shape of the relative estimates is well described by the relative portion of the CRB,  $\Sigma_r$ , and that the relative error is significantly less than the total error—as expected from the shading arguments above. In addition, there is much less correlation of localization error across nodes, as seen by the lack of shading structure in the relative estimates of Figure 13.9.

**13.4.3.2 Anchor Evaluation** In the previous example of localizing the sensors of Figure 13.1, we observed that the majority of the total error was in the transformation subspace due to the large rotational uncertainty. This large rotational uncertainty arose because the anchor nodes used (1, 5, and 9) were clustered in the corner of the network and provided little information about the transformation subspace. The total error may be reduced by selecting better anchor nodes. Here we consider all possible  $\binom{13}{3} = 286$  anchor triples and evaluate the resulting total error, relative error, and transformation error.

In Figure 13.10 we plot these three errors as a function of all possible anchor node triples, sorted by decreasing total error. From the figure we see that different anchor sets have little effect on the relative error but have a significant effect on the transformation error. This supports the earlier claim that errors in the relative and transformation domains may differ greatly because they are informed upon by different sources. In this example, the optimal anchor set with the minimum total error is  $\{3, 7, 13\}$ .

Clearly, anchor node selection has a large influence on total localization performance and should be optimized if possible. If all sensor locations were known, an analysis similar to Figure 13.10 would yield the optimal anchor set. Obviously, in practice, sensor locations are not known prior to anchor node locations being measured. This problem



**Figure 13.10** Total estimation error  $e = e_t + e_r$  for the localization of the sensors in Figure 13.1 as a function of selected anchor triples. The decomposition demonstrates that the variability in total error is primarily due to the transformation component,  $e_t$ . The relative error,  $e_r$ , is relatively invariant to which nodes serve as anchors.

is considered in [41] where general anchor placement heuristics are established such that the anchors are maximally informative about transformation parameters—which are not informed upon by internode measurements.

### 13.5 CONCLUSIONS

We have formulated sensor localization as a parameter estimation problem in which measurement errors are quantified as random variables, and prior information is provided as either constraints or probabilistic prior probabilities. This statistical framework provides a structure with which we can derive optimal and suboptimal localization algorithms, as well as algorithm-independent bounds such as the Cramér–Rao bound. The statistical framework also readily accommodates hybrid scenarios consisting of multiple types of measurements.

We derived a unified statistical estimation framework that encompasses all commonly used localization measurement types, including TOA, TDOA, AOA, ADOA, RSS, and RSSD solutions. We showed, for example, that TOA and TDOA can be formulated as the same measurement process, where in one case a parameter is assumed known, whereas in the other case that parameter is assumed to be deterministic but unknown; similarly, AOA and ADOA and RSS and RSSD localization can be modeled jointly.

A significant benefit of a statistical formulation of the localization problem is the ability to develop performance bounds from Fisher information and the corresponding Cramér–Rao inequality. These algorithm-independent bounds provide a computationally attractive means of comparing and making design trade-offs in localization

approaches; for example, we showed that TDOA-based localization resulted in an average of 12% higher localization error than a corresponding TOA-based approach for the example considered. In another example, the utility of combining distances and AOA measurements was considered as a function of the quality of each measurement type. Numerous other analyses—such as the impact of measurement density, the value of increased anchor nodes, AOA versus ADOA, and the like—may be considered using the Fisher information matrix derivations of this chapter.

Finally, we employed the statistical formulation to derive optimal estimation algorithms and also to provide a quantitative benchmark for suboptimal algorithms. In many cases the optimal algorithms require nonlinear minimization approaches that must be initialized. We derived a set of closed-form position estimation algorithms that were not directly founded on the statistical model; however, the model was used to compare performance of the closed-form algorithm against the best possible performance. In some cases the closed-form algorithm performance may be sufficient for direct use; in other cases, the algorithm can be used as an effective initial estimate to iterative refinement via an optimal algorithm. The CRB provides a performance benchmark to assist in making this decision.

## REFERENCES

1. D. Goense, J. Thelen, and K. Langendoen, “Wireless sensor networks for precise phytophthora decision support,” paper presented at the *5th European Conference on Precision Agriculture (5ECPA)*, June 2005.
2. P. C. Robert (Ed.), *International Journal on Advances in the Science of Precision Agriculture*, 1998–2006, available: <http://www.kluweronline.com/issn/1385-2256>.
3. A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, “Wireless sensor networks for habitat monitoring,” in *WSNA ’02: Proceedings of the 1st ACM international workShop on Wireless Sensor Networks and Applications*, ACM Press, New York, 2002, pp. 88–97.
4. A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, “Habitat monitoring: Application driver for wireless communications technology,” *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 2, Suppl., pp. 20–41, 2001.
5. D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, “Detection, classification, and tracking of targets,” *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 17–29, 2002.
6. A. Arora et al., “A line in the sand: A wireless sensor network for target detection, classification, and tracking,” *Computer Networks*, vol. 46, pp. 605–634, 2004.
7. I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” *J. Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
8. M. Perkins, N. Correal, and B. O’Dea, “Emergent wireless sensor network limitations: A plea for advancement in core technologies,” in *Sensors, 2002, Proc. IEEE*, vol. 2, nos. 12/14, pp. 1505–1509, June 2002.
9. A. Arora et al., “Exscal: Elements of an extreme scale wireless sensor network,” in *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, 2005*, Aug. 17–19, 2005, pp. 102–108.
10. G. Mao, B. Fidan, and B. D. Anderson, “Wireless sensor network localization techniques,” *Computer Networks*, vol. 51, pp. 2529–2553, 2007.
11. N. Patwari, J. N. Ash, S. Kyerountas, A. O. Hero III, R. L. Moses, and N. S. Correal, “Locating the nodes: Cooperative localization in wireless sensor networks,” *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, July 2005.

12. T. Rappaport, *Wireless Communications Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, 1996.
13. S. M. Kay, *Fundamentals of Statistical Signal Processing*, Vol. I: *Estimation Theory*, Vol. 1, Prentice Hall, Englewood Cliffs, NJ, 1993.
14. P. Stoica and B. Ng, "On the Cramér-Rao bound under parametric constraints," *IEEE Signal Process. Lett.*, vol. 5, no. 7, pp. 177–179, 1998.
15. J. N. Ash and R. L. Moses, "On the relative and absolute positioning errors in self-localization systems," *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5668–5679, 2008.
16. W. S. Torgerson, "Multidimensional scaling: I. Theory and method," *Psychometrika*, vol. 17, pp. 401–419, 1952.
17. J. C. Gower, "Some distance properties of latent root and vector methods used in multivariate analysis," *Biometrika*, vol. 53, pp. 325–338, 1966.
18. J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
19. J. N. Ash and L. C. Potter, "Robust system multiangulation using subspace methods," *Proc. Inform. Process. Sensor Networks*, pp. 61–68, Apr. 2007.
20. P. H. Schönemann and R. M. Carroll, "Fitting one matrix to another under choice of a central dilation and a rigid motion," *Psychometrika*, vol. 35, pp. 245–255, 1970.
21. H. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*, Wiley, New York, 1968.
22. R. L. Moses, D. Krishnamurthy, and R. Patterson, "A self-localization method for wireless sensor networks," *Eurasip J. Appl. Signal Process., Special Issue on Sensor Networks*, vol. 2003, no. 4, pp. 348–358, Mar. 2003.
23. N. Patwari, A. Hero III, M. Perkins, N. Correal, and R. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.
24. P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *IPSN '04: Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, ACM Press, New York, 2004, pp. 46–54.
25. A. Savvides, C.-C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *MobiCom '01: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ACM Press, New York, 2001, pp. 166–179.
26. D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, ACM Press, New York, 2004, pp. 50–61.
27. J. Smith and J. Abel, "Closed-form least-squares source location estimation from range-difference measurements," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-35, no. 12, 1987.
28. X. Sheng and Y.-H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 53, no. 1, pp. 44–53, 2005.
29. N. Patwari and A. Hero, "Signal strength localization bounds in ad hoc and sensor networks when transmit powers are random," paper presented at the IEEE Workshop on Sensor Array and Multichannel Processing, 2006, pp. 299–303.
30. D. Li and Y. H. Hu, "Energy-based collaborative source localization using acoustic microsensor array," *EURASIP J. Appl. Signal Process.*, vol. 2003, no. 4, pp. 321–337, 2003.
31. C. Meesookho, U. Mitra, and S. Narayanan, "On energy-based acoustic source localization for sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 365–377, 2008.

32. K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan, "Ecolocation: A sequence based technique for RF localization in wireless sensor networks," in *IPSN 2005, Fourth International Symposium on Information Processing in Sensor Networks, 2005*, Apr. 15, 2005, pp. 285–292.
33. L. Doherty, L. El Ghaoui, and K. Pister, "Convex position estimation in wireless sensor networks," *Proc. INFOCOM*, vol. 3, pp. 1655–1663, 2001.
34. T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *MobiCom '03: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, ACM Press, New York, 2003, pp. 81–95.
35. N. Patwari and A. Hero III, "Using proximity and quantized RSS for sensor localization in wireless networks," in *Proc. 2nd International ACM Workshop on Wireless Sensor Networks and App.*, San Diego, CA, Sept. 19, 2003.
36. R. L. Moses, D. Krishnamurthy, and R. Patterson, "An auto-calibration method for unattended ground sensors," *IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 3, no. 4, pp. 2941–2944, May 13–17, 2002.
37. D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," *Proc. INFOCOM*, vol. 3, pp. 1734–1743, 2003.
38. J. Albowicz, A. Chen, and L. Zhang, "Recursive position estimation in sensor networks," *IEEE Int. Conf. Network Protocols*, pp. 35–41, Nov. 2001.
39. J. Costa, N. Patwari, and A. Hero, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sensor Networks*, vol. 2, no. 1, pp. 39–64, 2006.
40. A. T. Ihler, J. W. Fisher III, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for sensor network self-calibration," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 809–819, 2005.
41. J. N. Ash and R. L. Moses, "On optimal anchor node placement in sensor localization by optimization of subspace principal angles," *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 2289–2292, Mar. 30–Apr. 4 2008.



---

## CHAPTER 14

---

# Multitarget Tracking and Classification in Collaborative Sensor Networks via Sequential Monte Carlo Methods

Tom Vercauteren<sup>1</sup> and Xiaodong Wang<sup>2</sup>

<sup>1</sup>Asclepios Research Project, INRIA Sophia Antipolis, Sophia Antipolis Cedex, France

<sup>2</sup>Electrical Engineering Department, Columbia University, New York, New York

### 14.1 INTRODUCTION

The convergence of recent developments in microelectromechanical systems (MEMS), microprocessors, and ad hoc networking protocols have enabled low-power and low-cost sensor nodes to collaborate and achieve large tasks [1]. Individually, each node owns limited sensing, communicating and computing capabilities, but, when a large number of them are used in conjunction, it is possible to achieve a reliable and robust network. These devices collect measurements from the physical environment, communicate with each other, and carry out computations in order to transmit only the required data to the end user. The network is then able to perform event detection, event identification, and location sensing in a field under observation [2]. Typical applications of such sensor networks are environmental monitoring, military surveillance, and space exploration, to name a few.

This chapter focuses on the problem of jointly tracking and classifying several targets evolving within densely scattered sensor nodes. On the one hand, multiple target tracking tackles the issue of sequentially estimating the state of a possible varying number of objects; and on the other hand, classification deals with the identification of those objects down to a given class. Due to the fact that the number of targets can vary, we are handling three closely coupled subjects: target detection, tracking, and classification. Considering the strong interrelations existing between those, it is natural to address them jointly.

As the amount of observation data is limited, making use of motion models for the targets is essential to obtain a good tracking performance. If the motion model describes the target's movement accurately, the model-based tracking algorithm will outperform a model-free algorithm [3]. In this chapter, we assume that the motion model of each target belongs to one of several given classes. For each target, the classification task consists of the estimation of which motion model better describes the target's

movement. Information on the class of a target provides very useful knowledge about its motion characteristics and thus allow the tracking to be more accurate.

Sensor nodes are usually prone to errors and, therefore, the measurements available to perform the aforementioned operations can either arise from the targets of interest when they are detected or be false detections. Those spurious measurements will be denoted as clutter noise. They could, for instance, be generated by returns from nearby objects or electromagnetic interference. They are generally random in number and intensity, which makes the clutter noise statistically separable from the useful information. One of the major problems in such a system arises from the generally unknown association between the available measurements and the targets of interest. These associations should thus be estimated at each time step. Traditionally, data association is handled by methods such as the nearest-neighbor algorithm or the joint probabilistic data association algorithm (JPDA) [4]. When dealing with nonlinear models and unknown number of targets, none of these methods can be applied directly. In this chapter, we use a statistical data association scheme.

Typical algorithms dealing with multitarget tracking are very computationally complex for the scenario considered here. Moreover, they generally require a centralized computation based on the measurements available from all sensors [4–6]. An important characteristic of sensor networks is their ability to cooperate among densely and randomly deployed sensor nodes [7]. Another significant feature is the low-power consumption requirement [1]. Sensor nodes carry limited, generally irreplaceable, power sources. Therefore, a sensor node can typically only sense objects that are in its neighborhood. It is thus of great importance to develop localized algorithms, where only a subset of the nodes are activated and are responsible for data fusion, instead of sending their raw measurements to an information fusion node.

The complex problem of multitarget tracking can then be tackled through sensor collaboration by dividing it into several easier, localized tasks. We can therefore consider the problem as that of tracking and classifying a single target with a possibly appearing (and then possibly vanishing) second target. In this chapter, the problem of information fusion and sensor selection is solved within the Bayesian framework. No closed-form solution for the posterior distribution of the target states is available, and therefore sequential Monte Carlo (SMC) methods are employed to approximate the filtering density. In order to model the varying number of targets, we will make use of a jump Markov system (JMS) [8, 9].

This chapter is an expanded version of [10] and is organized as follows. In Section 14.2 we provide a framework for multiple target tracking and classification in a collaborative sensor network. The general principle of sequential Monte Carlo inference is briefly reviewed in Section 14.3. The issue of classification is discussed in Section 14.4 where we present the joint tracking and classification SMC algorithm for a single target. The multitarget tracking and classification algorithm is developed in Section 14.5. In Section 14.6 we discuss the sensor selection scheme. Simulation results are presented in Section 14.7. Section 14.8 concludes the chapter.

## 14.2 SYSTEM DESCRIPTION AND PROBLEM FORMULATION

We assume that in a sensor network, the randomly deployed sensor nodes are able to collect data, process it, and route information back to the *sink* through a multihop

infrastructureless network. The sink then dispatches the data to the end user. One of the main concerns of any signal processing algorithm for sensor networks is to make efficient use of the limited available power resources. Each node should remain idle unless queried to perform a specific task [1]. We assume that the network has been initialized so that each node has the knowledge of its own position, its neighbors' identities, and their positions.

### **14.2.1 Leader-Based Tracking in Sensor Networks**

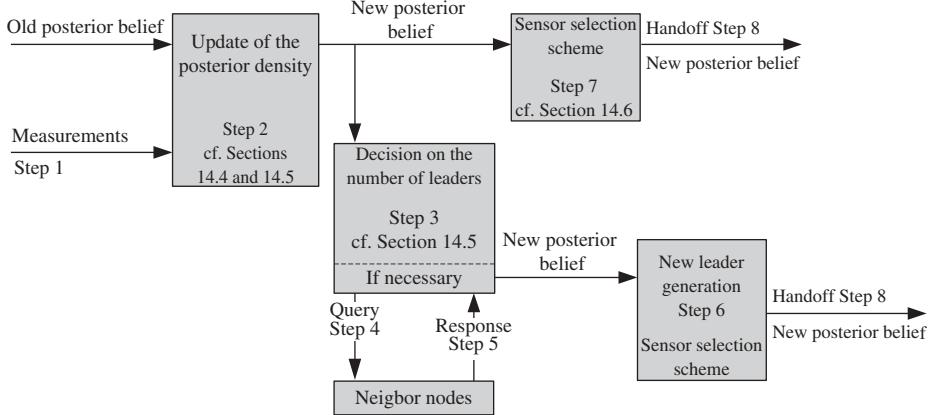
We consider a leader-based tracking scheme [also denoted as information-driven sensor querying (IDSQ) [2, 7]] in which, for each target and at each time step, only one sensor—the leader node—is active. Each leader node is thus focused on the tracking and classification of a single target. However, unless the other targets are far away, the leader cannot consider them as noise [4] because the statistical properties of the measurements arising from those other targets are, as opposed to the clutter noise, identical to those of the tracked target. We thus need to take those targets into account as soon as they appear in the field covered by the leader node. Moreover, if a previously untracked target appears, we need to be able to generate a new leader node dedicated to the tracking and classification of this appeared object. Hence a communication step between the nodes is necessary. In order not to generate a new leader if the appeared target is already tracked, a query should be made. At a given time step the current leader node performs the following operations:

1. The measurements are retrieved.
2. The posterior density (state, number, and class of the targets) is updated (cf. Sections 14.4 and 14.5).
3. Based on this information, a decision on whether a new target should be associated to a leader node is made (cf. Section 14.5).
4. If the decision is positive, a query is broadcast to the neighbor sensors in order to know if any of them is tracking a target.
5. If a response is received, nothing is done.
6. If no response comes back, a new leader node for the new target is generated.
7. A next leader node (for the first target) is chosen (cf. Section 14.6).
8. The belief state is handed over to this chosen node, which becomes the current leader node.
9. The node gets back to an idle state.

The above operations are systematically illustrated in Figure 14.1. The detailed signal processing algorithms involved in these operations are treated in the subsequent sections as indicated in Figure 14.1. Such a leader-based scheme has several advantages that are particularly attractive for collaborative sensor network applications, for example, only local computations are involved, no global knowledge is assumed, and there is no need for centralized control.

### **14.2.2 Multiple-Target Tracking and Classification**

When performing joint tracking and classification, we aim at giving a good estimate of the state of a target (position, velocity, and class). We consider a model-based target



**Figure 14.1** Operations performed by leader node in multiple-target tracking and classification.

tracking method [3] where the target motions and the observations can be represented by state-space models. When the number of targets is known and greater than one, the state of the system is the concatenation of the states of all targets, and we have to tackle the data association problem. Several approaches have been proposed such as JPDA [4].

Let  $r_t$  denote the number of targets at time  $t$ . Let  $T_t$  be the set of active targets at time  $t$  (its cardinality is  $r_t$ ). We denote by  $X_t = \{x_{t,i}, \gamma_i\}_{i \in T_t}$  the state of the targets at time  $t$ , where  $x_{t,i}$  stands for the position and velocity and  $\gamma_i$  stands for the class of the  $i$ th target at time step  $t$ . The system at time  $t$  is thus characterized by the vector  $(X_t, r_t)$ . Conditioned on the number of targets at time  $t$  and  $t + 1$ , the system dynamic model can be written as

$$f_{r_t, r_{t-1}}(X_t | X_{t-1}) = p(X_t | X_{t-1}, r_t, r_{t-1}). \quad (14.1)$$

We make the common assumption that each target moves independently from the other according to a Markovian transition dynamic. This dynamic depends on the class  $\gamma_i$  of the  $i$ th target. The dynamic of the system (conditioned on the number of targets at time  $t$  and  $t + 1$ ) can thus be decomposed into several equations:

$$x_{t,i} = F_{\gamma_{t,i}}(x_{t-1,i}, u_{t,i}), \quad \forall i \in T_t \cap T_{t-1}, \quad (14.2)$$

$$\gamma_{t,i} = \gamma_{t-1,i}, \quad \forall i \in T_t \cap T_{t-1}, \quad (14.3)$$

where the function  $F_{\gamma_{t,i}}$  is the  $\gamma_{t,i}$ th underlying motion model for the  $i$ th target. If  $i \notin T_{t-1}$ , the terms referring to  $t - 1$  in (14.2) and (14.3) correspond to a given prior information. The noise terms  $u_{t,i}$  are assumed to be white and pairwise independent.

Let  $m_t$  be the number and  $Z_t = (z_t^1, \dots, z_t^{m_t})$  be the vector of available measurements at time  $t$ . We assume that at most one measurement can arise from each target, and that several measurements can arise from the clutter. The data association vector is denoted by  $a_t$ , which is thus a vector of length  $m_t$  whose components take values in  $T_t \cup \{0\}$ . Note that  $a_t(m) = i$  means that the  $m$ th measurement has been generated by the  $i$ th target, whereas  $a_t(m) = 0$  means that the  $m$ th measurement is a spurious one, arising from the clutter. Conditioned upon the data association and the state of the

system, the measurements are assumed to be independent. The general model for the measurements is thus as follows:

$$z_t^m = H_t(x_{t,a_t(m)}, v_{t,m}), \quad \forall m \in \{m' | a_t(m') \neq 0\}, \quad (14.4)$$

$$z_t^m \sim p_c(z), \quad \forall m \in \{m' | a_t(m') = 0\}, \quad (14.5)$$

where  $H_t$  is the measurement function if the measurement arises from a target and is specified by some probability distribution  $p_c$  if it arises from the clutter. The noise terms  $v_{t,m}$  are assumed to be white and pairwise independent.

### 14.2.3 Target Dynamics

As mentioned before, we consider class-dependent motion models. Each target is considered as a point object moving according to its dynamic in a two-dimensional plane. Those motion models are essential to any model-based tracking algorithm and thus need to be well fit to the tracked targets. In this work we consider two different motion models explained below: the constant velocity model and the coordinated turn rate model. We refer the readers to [3] for an up-to-date survey of the available motion models. We present here the different target dynamics as discretized physical motion models. We use  $\tau$  to denote the length of a time step.

**14.2.3.1 Constant-Velocity Model** This model is the most commonly used one. It is assumed that the target moves with a constant velocity. We will here consider, for notational simplicity,  $x_t$  as the state of a single target from this class and  $u_t$  as the corresponding motion noise;  $x_t$  represents the coordinates and the velocities. We will denote with  $\alpha$  and  $\beta$  those coordinates:  $x_t \triangleq \{\alpha_t, \dot{\alpha}_t, \beta_t, \dot{\beta}_t\}$ . We have

$$x_{t+1} = \begin{pmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{pmatrix} x_t + \begin{pmatrix} \frac{\tau^2}{2} & 0 \\ \tau & 0 \\ 0 & \frac{\tau^2}{2} \\ 0 & \tau \end{pmatrix} u_t, \quad (14.6)$$

where

$$u_t \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}\right). \quad (14.7)$$

**14.2.3.2 Coordinated Turn Rate Model** This model assumes that the target moves with a constant speed (norm of the velocity vector) and a constant known turn rate  $\omega$ . Again we denote  $x_t$  as the state of a single target from this class and  $u_t$  as the corresponding motion noise. We have

$$x_{t+1} = \begin{pmatrix} 1 & \frac{\sin \omega \tau}{\omega} & 0 & -\frac{1 - \cos \omega \tau}{\omega} \\ 0 & \cos \omega \tau & 0 & -\sin \omega \tau \\ 0 & -\frac{1 - \cos \omega \tau}{\omega} & 1 & \frac{\sin \omega \tau}{\omega} \\ 0 & \sin \omega \tau & 0 & \cos \omega \tau \end{pmatrix} x_t + \begin{pmatrix} \frac{\tau^2}{2} & 0 \\ \tau & 0 \\ 0 & \frac{\tau^2}{2} \\ 0 & \tau \end{pmatrix} u_t, \quad (14.8)$$

where  $u_t$  has the same Gaussian distribution as in (14.7).

#### 14.2.4 Sensing Model

As mentioned previously, each sensor provides a set of measurements that can be divided into two distinct sets: the first one consists of the measurements generated by the detected targets and the second one is composed of the false detections generated by the clutter.

**14.2.4.1 Target-Originated Measurements** Many types of sensors provide measurements that are a function of the relative distance between the sensor and the sensed object (e.g., radar, ultrasound, sonar, etc.). Again we denote by  $x_t = \{\alpha_t, \dot{\alpha}_t, \beta_t, \dot{\beta}_t\}$  the state of a single target. The index  $s$  will refer to the sensor of interest whose position is  $\{\alpha^s, \beta^s\}$ . The distance between the sensor and the target is then

$$d_s(x_t) = ((\alpha_t - \alpha^s)^2 + (\beta_t - \beta^s)^2)^{1/2}. \quad (14.9)$$

A common example is given by sensors measuring the power of a radio signal emitted by the object. The received power typically exponentially decays with the relative distance. In a logarithmic scale, the measurements are modeled by

$$h(d) = K - 10\eta \log_{10}(d), \quad (14.10)$$

$$H_s(x_t) = h(d_s(x_t)) + v_t, \quad (14.11)$$

where the measurement noise  $v_t$  is assumed to be a zero-mean independent and identically distributed (i.i.d.) Gaussian, that is,  $v_t \sim \mathcal{N}(0, R)$ ;  $K$  is the transmission power and  $\eta \in [2, 5]$  is the path loss exponent. These parameters depend on the radio environment, antenna characteristics, terrain, and the like. Note that  $\eta = 2$  corresponds to the free space transmission and serves as a lower limit. Furthermore, a sensor can provide measurements of a target only within a certain range. Therefore, a target could be detected only if  $d_s(x_t) \in [d_{\min}, d_{\max}]$ . In that case we will denote by  $P_D$  the probability of detection, which is assumed known. Finally, it is assumed that one target can only provide a single measurement.

**14.2.4.2 Clutter Noise Model** The false detections are spurious measurements  $z$  assumed to be uniformly distributed  $\mathcal{U}_{A_{\text{meas}}}$  in the measurement area  $A_{\text{meas}} = [h(d_{\max}), h(d_{\min})]$  whose volume is denoted as  $V_{\text{meas}} = |h(d_{\min}) - h(d_{\max})|$ . The number of false detections  $m_t^0$  is typically generated by a Poisson distribution with parameter  $\lambda V_{\text{meas}}$ , where  $\lambda$  is the number of clutter measurements per unit volume. Hence we have

$$P(m_t^0 = k) = e^{-\lambda V_{\text{meas}}} \frac{(\lambda V_{\text{meas}})^k}{k!}, \quad k = 0, 1, 2, \dots \quad (14.12)$$

$$p(z) \sim \mathcal{U}_{A_{\text{meas}}}. \quad (14.13)$$

#### 14.2.5 Conditional Distribution of Measurements

The main issue when computing the conditional distribution of a set of measurements resides in the uncertainty in the origin of the measurements. We consider a statistical

data association conditioned on the number of targets in the field, which is related to the joint probabilistic data association (JPDA) method [4, 5]. Because of the arbitrariness in the ordering of the measurements, we assume that (without any knowledge of the value of the measurements) the associations are independent of the current state. The conditional distribution of the measurements can be expanded as

$$p(Z_t|X_t, r_t) = \sum_{a_t} p(Z_t|X_t, r_t, a_t) p(a_t|r_t, m_t). \quad (14.14)$$

It is seen in (14.14) that all possible data associations are enumerated. This is often a major problem when dealing with multiple targets. When a centralized (nonlocal) sensor is used, the measurements can arise from the entire field of interest, and ideally all possible data associations should be considered. In order to reduce the complexity of such a procedure, it is common to use only a subset of all possible data associations. This subset is constructed by allowing an association  $a_t(m) = i$  only if the  $m$ th measurement  $z_t^m$  is close to the estimate of the expected value  $E[H_s(x_{t,i})]$ . This idea is referred as the gating procedure [4]. In our problem the sensors can only provide local information, and thus a natural gating is made. The total number of measurements and targets will be small and thus the number of possible data associations remains small.

The prior probability of a data association (given the number of targets, the number of measurements, and the probability of detection  $P_D$ ) only depends on the set of detected targets and not on the order within the data association vector. The number of data associations in which the same set of targets is detected is given by

$$\binom{m_t}{m_t^0(a_t)} (m_t - m_t^0(a_t))! = \frac{m_t!}{m_t^0(a_t)!}.$$

The prior probability of a data association is then given by

$$\begin{aligned} p(a_t|r_t, m_t) &= \frac{m_t^0(a_t)!}{m_t!} P_D^{m_t - m_t^0(a_t)} (1 - P_D)^{r_t - (m_t - m_t^0(a_t))} e^{-\lambda V_{\text{meas}}} \frac{(\lambda V_{\text{meas}})^{m_t^0(a_t)}}{m_t^0(a_t)!}, \\ &\propto P_D^{-m_t^0(a_t)} (1 - P_D)^{r_t + m_t^0(a_t)} (\lambda V_{\text{meas}})^{m_t^0(a_t)}. \end{aligned} \quad (14.15)$$

The other term  $p(Z_t|X_t, r_t, a_t)$  in (14.14) is computed by assuming that the measurements are conditionally independent.

$$p(Z_t|X_t, r_t, a_t) = \left( \frac{1}{V_{\text{meas}}} \right)^{m_t^0(a_t)} \prod_{\{m | a_t(m) \neq 0\}} p(z_t^m | x_{t,a_t(m)}). \quad (14.16)$$

Within the framework described above, we aim at performing an online estimation of the a posteriori distributions of the target positions, number, and class affiliations  $p(X_t|Z_{1:t})$  at time  $t$  based on the measurements  $Z_{1:t}$  at densely deployed sensor nodes. The exact solution to this problem involves a very high dimensional integration that is infeasible in practice. We will employ the sequential Monte Carlo (SMC) techniques to solve this problem. The basic principle of SMC is discussed next.

### 14.3 SEQUENTIAL MONTE CARLO METHODS

We consider a generic dynamic model described by the following:

Initial state model:

$$p(X_0), \quad (14.17)$$

State transitions model:

$$p(X_t|X_{t-1}) \quad \forall t \geq 1, \quad (14.18)$$

Measurement model:

$$p(Z_t|X_t) \quad \forall t \geq 1. \quad (14.19)$$

The cumulative sets of states and measurements are denoted by  $X_{0:t} \triangleq (X_0, X_1, \dots, X_t)$  and  $Z_{1:t} \triangleq (Z_1, \dots, Z_t)$ . Suppose an online inference of  $X_{0:t}$  is of interest. That is, at current time  $t$  we wish to make an estimate of a function of the state variable  $X_{0:t}$ , say  $\psi(X_{0:t})$ , based on the currently available observations,  $Z_{1:t}$ . The optimal solution to this problem in the sense of minimum mean-square error is

$$E\{\psi(X_{0:t})|Z_{1:t}\} = \int \psi(X_{0:t}) p(X_{0:t}|Z_{1:t}) dX_{0:t}. \quad (14.20)$$

In most cases, an exact evaluation of this expectation is analytically intractable. Sequential Monte Carlo methods [11–15] are simulation-based techniques, making use of *sequential importance sampling* (SIS), which provide a reliable approximation of this solution.

Let us introduce an *arbitrary* proposal distribution  $q(X_{0:t}|Z_{1:t})$  from which we can easily draw samples. Provided that the support of  $q(\cdot)$  includes the support of  $p(X_{0:t}|Z_{0:t})$ , we have the following identity:

$$E_p\{\psi(X_{0:t})|Z_{1:t}\} = \int \psi(X_{0:t}) w(X_{0:t}) q(X_{0:t}|Z_{1:t}) dX_{0:t} = E_q\{\psi(X_{0:t}) w(X_{0:t})|Z_{1:t}\}, \quad (14.21)$$

where  $w(X_{0:t}) = p(X_{0:t}|Z_{1:t})/q(X_{0:t}|Z_{1:t})$  is denoted as the importance weight. Thus by drawing  $N$  random samples  $\{X_{0:t}^{(j)}\}_{j=1}^N$  from the proposal distribution  $q(X_{0:t}|Z_{1:t})$ , it is possible to obtain an estimate of (14.20) as

$$E_p\{\psi(X_{0:t})|Z_{1:t}\} \simeq \frac{1}{W_t} \sum_{j=1}^N w_t^{(j)} \psi(X_{0:t}^{(j)}), \quad (14.22)$$

where  $w_t^{(j)} \triangleq w(X_{0:t}^{(j)})$  and  $W_t \triangleq \sum_{j=1}^N w_t^{(j)}$ . The set,  $\{X_{0:t}^{(j)}, w_t^{(j)}\}_{j=1}^N$ , of random draws and weights is said to be *properly weighted* with respect to the target distribution  $p(X_{0:t}|Z_{1:t})$ . One such sample together with its weight is commonly denoted as a *particle*.

### 14.3.1 Sequential Importance Sampling

The posterior distribution can be expressed using the Bayes rule as

$$p(X_{0:t}|Z_{1:t}) = \frac{p(Z_{1:t}|X_{0:t})p(X_{0:t})}{p(Z_{1:t})}. \quad (14.23)$$

Therefore we get the following recursive formula:

$$p(X_{0:t}|Z_{1:t}) = p(X_{0:t-1}|Z_{1:t-1}) \frac{p(Z_t|X_t)p(X_t|X_{t-1})}{p(Z_t|Z_{1:t-1})}. \quad (14.24)$$

This motivates us to adopt a recursive importance sampling strategy by choosing a proposal density that can be factorized as  $q(X_{0:t}|Z_{1:t}) = q(X_{0:t-1}|Z_{1:t-1})q(X_t|X_{0:t-1}, Z_{1:t})$ . It is thus possible to sequentially draw from  $q(X_{0:t}|Z_{1:t})$  by keeping the past simulated streams  $\{X_{0:t-1}^{(j)}, w_{t-1}^{(j)}\}$  unmodified, and then drawing  $X_t^{(j)}$  from  $q(X_t|X_{0:t-1}^{(j)}, Z_{1:t})$ . The weights in (14.22) are also recursively updated and become

$$w_t^{(j)} \propto w_{t-1}^{(j)} \frac{p(Z_t|X_t^{(j)})p(X_t^{(j)}|X_{t-1}^{(j)})}{q(X_t^{(j)}|X_{0:t-1}^{(j)}, Z_{1:t})}. \quad (14.25)$$

### 14.3.2 Resampling Procedure

A common problem with the SIS algorithm is known as the degeneracy phenomenon. In [11, 13] it has been shown that the variance of the importance weights can only increase over time, which makes the degeneracy problem ineluctable. After a few iterations, some particles will have very small weights. Such samples are said to be ineffective. If there are too many ineffective samples, the Monte Carlo procedure becomes inefficient. Computational effort is lost updating particles with weights close to zero and the smaller amount of effective samples makes the approximation to the posterior distribution less accurate.

Two options are possible to tackle this problem. The first one is to choose the proposal density so that the variance of the importance weights is minimized, but this generally leads to a proposal density difficult to implement. The second option, called *resampling*, is a useful method to leave particles with small weights behind and to concentrate on particles with large weights whenever a significant degeneracy is observed. One simple resampling scheme can be described as follows (several are available and produce similar results [12]):

- Draw a sample stream  $\{\bar{X}_{0:t}^{(J)}\}_{J=1}^N$  from  $\{X_{0:t}^{(j)}\}_{j=1}^N$  with probabilities proportional to the weights  $\{w_t^{(j)}\}_{j=1}^N$ .
- Assign equal weight to each new sample  $\bar{X}_{0:t}^{(J)}$ , that is,  $\bar{w}_t^{(J)} = 1/N$ .

It is shown in [16] that samples drawn by the above resampling procedure are indeed properly weighted with respect to  $p(X_{0:t}|Z_{1:t})$ , provided that  $N$  is sufficiently large.

The degeneracy of the particles can be measured by the effective sample size  $N_{\text{eff}}$  defined as

$$N_{\text{eff}} \triangleq \frac{N}{1 + \text{Var} \left( \frac{p(X_t^{(j)} | Z_{1:t})}{q(X_t^{(j)} | X_{0:t-1}^{(j)}, Z_{1:t})} \right)}. \quad (14.26)$$

It can be approximated by [15] that

$$\widehat{N}_{\text{eff}} = \frac{1}{\sum_{j=1}^N (w_t^{(j)})^2}. \quad (14.27)$$

Heuristically,  $\widehat{N}_{\text{eff}}$  reflects the equivalent size of a set of i.i.d. samples for the set of  $N$  weighted ones. It is suggested in [12, 16] that resampling should be performed whenever the effective sample size becomes small, for example,  $\widehat{N}_{\text{eff}} \leq N/10$ .

## 14.4 JOINT SINGLE-TARGET TRACKING AND CLASSIFICATION

When tracking a target whose maneuvering capabilities are unknown, the use of a very general motion model can lead to very poor estimates. However, this uncertainty is often due to the lack of knowledge about the *type* of the tracked object and only a finite set of types is of interest. Another approach when dealing with such an uncertainty is to compare several classes of dynamic models such as those presented in Section 14.2. The more information we have about the possible trajectory envelope, the more efficient is the tracking. Besides, knowledge about the identity of a target is of major importance for the analysis of an event. In this section we propose an algorithm for jointly tracking and classifying a single target evolving within a sensor network.

### 14.4.1 Related Work

In [17], the authors introduced a Bayesian target classification method based on the estimate of kinematics only. Their major contribution was to point out the dependence between the target state and the target class, and then to integrate this dependence into a joint tracking and classification algorithm. Within this framework, the estimations are provided by a grid-based algorithm that is known to be very difficult to implement, especially in high dimensional spaces.

In [18] the problem of multitarget tracking (for a fixed number of targets) is also addressed. In their implementation, noisy measurements could force particles corresponding to the correct class to have negligible likelihoods and little chance of being selected during resampling. This would eliminate all particles corresponding to the correct class, and the class estimate would settle on a fixed and incorrect value. To solve this problem, it is proposed in [19] to use a separate particle filter for each possible class and a method for comparing different filters is given.

The problem of joint tracking and classification can be seen as that of simultaneously dealing with both a fixed model parameter (class) and state variables (position and velocity). Several works have proposed algorithms for dealing with static parameters within an SMC framework [20, 21]. However, the parameters considered there are continuous, which is fundamentally different from our joint tracking and classification problem.

Our approach combines the advantages of the previous works. Within an SMC framework, we first make sure to always be able to recover information related to a

specific class. And in order to devote more (but not all) computational load to the more likely classes, our approach will compound the different class information into a single filter.

#### 14.4.2 Class-Based Resampling Scheme

Our algorithm will rely on the framework presented in Section 14.2, but for the sake of clarity here we focus on the case of a single target. Extension to the multiple-target case will be presented in Section 14.5. In order to simplify the notations, we will here consider  $X_t = (x_t, \gamma)$  as the joint state and class of the target of interest.

The simplest way of dealing with the class is to include it in the state vector and then to use a simple particle filter for this augmented state. This makes use of the static evolution model (14.3) of the class parameter. Because of this absence of dynamic, the number of particles from each class will not change during the updating step. A change will only occur during the resampling stage. Sometimes, and especially during the initial steps, the particle filter may lock on the wrong class. This situation can sometimes last for quite a while. Then, all particles might eventually settle in a wrong class. To avoid this situation, one idea is to assume an artificial evolution of the class parameter by adding a small noise on the static model (14.3). This results in a model mismatch and, because only the recent observations still have an influence on the class estimation, this also leads to a *loss of information* as argued in [21].

Because of the finite number of available classes, it is possible to keep particles for each of those. If we assume that a sufficient number of particles from each class remains available at each time, then it would always be possible to recover from a misclassification. Subsequently, our aim is to keep a sufficient number of particles per class. Since a change in the number of particles arise only during the resampling, this suggests modifying the resampling algorithm so as to meet our needs.

**14.4.2.1 The Algorithm** We mentioned that the resampling scheme was essentially a way of eliminating trajectories with small weights and replicating those with large weights. This can be applied within the set of particles belonging to the same class. In order to concentrate on the class the target is most likely to belong to, we will first set the number of particles associated with each class and then perform the resampling within a class. The number of particles for each class will be drawn according to the class probabilities and then thresholded to keep a sufficient number of particles for each class while keeping a constant total number of particles. Because resampling is made per class, the total weight corresponding to a given class should not change. And because within a class we draw the indexes according to the weights, all particles within a class should have the same weight.

We denote  $\Lambda$  as the set of all possible classes;  $q(x_t | \gamma_t^{(j)}, X_{t-1}^{(j)}, Z_t)$  refers to a generic proposal distribution whose support includes the support of  $p(x_t | \gamma_t^{(j)}, X_{t-1}^{(j)}, Z_t)$ . Finally, we summarize the SMC algorithm for joint single tracking and classification as follows:

##### Sequential Importance Sampling

- For  $j = 1, \dots, N$ , set  $\gamma_t^{(j)} = \gamma_{t-1}^{(j)}$ , sample  $x_t^{(j)} \sim q(x_t | \gamma_t^{(j)}, X_{t-1}^{(j)}, Z_t)$ , and then set  $X_{0:t}^{(j)} = (X_{0:t-1}^{(j)}, (x_t^{(j)}, \gamma_t^{(j)}))$ .

- Compute the importance weight using (14.25) and (14.14) (with  $r_t = 1$ ):

$$w_t^{(j)} \propto w_{t-1}^{(j)} \frac{p(Z_t | x_t^{(j)}) p(x_t^{(j)} | x_{t-1}^{(j)}, \gamma_t^{(j)})}{q(x_t^{(j)} | \gamma_t^{(j)}, x_{t-1}^{(j)}, Z_t)}. \quad (14.28)$$

- Normalize the weights so that they sum up to 1.

### Estimation

- Compute the estimate of the probability of each class  $\hat{P}(\gamma | Z_{0:t}) = \sum_{j \in \{\gamma\}} w_t^{(j)}$ .
- Compute the estimate of the target state  $\hat{x}_t = \sum_{j=1}^N w_t^{(j)} x_t^{(j)}$ .

### Class-Based Resampling

- Draw the number of particles for each class  $N_\gamma$  according to a multinomial distribution with parameters  $\{\hat{P}(\gamma | Z_{0:t})\}_{\gamma \in \Lambda}$ .
- $\forall \gamma \in \Lambda$ , if  $N_\gamma < N_{\text{Threshold}}$ , set  $N_\gamma = N_{\text{Threshold}}$ .
- Reduce the number of particles from the class with the most particles until  $\sum_{\gamma \in \Lambda} N_\gamma = N$ .
- $\forall \gamma \in \Lambda$ , draw  $N_\gamma$  sample streams  $\{\bar{X}_{0:t}^{(j)}\}$  from  $\{X_{0:t}^{(j)}\}_{j \in \{j' | \gamma_t^{(j')} = \gamma\}}$  with probability proportional to the weights  $\{w_t^{(j')}\}_{j \in \{j' | \gamma_t^{(j')} = \gamma\}}$ .
- Assign equal weight to each new sample within a class, that is,

$$\bar{w}_t^{(j)} = \frac{\hat{P}(\gamma^{(j)} | Z_{0:t})}{N_{\gamma^{(j)}}}.$$

For a sufficient large  $N_{\text{Threshold}}$ , it is possible to show [22] that if the set of samples representing  $P(x_t | \gamma, Z_{0:t})$  is properly weighted, then it is also properly weighted after resampling within a class. Therefore, by using the expansion  $P(x_t | Z_{0:t}) = \sum_\gamma P(x_t | \gamma, Z_{0:t}) P(\gamma | Z_{0:t})$  our complete set of samples will remain properly weighted with respect to  $P(x_t | Z_{0:t})$ .  $N_{\text{Threshold}}$  is a tuning parameter of the algorithm.

**14.4.2.2 Kernel Smoothing of Belief State** For each time step  $t$ , we have current posterior samples  $X_t^{(j)}$  and weights  $w_t^{(j)}$ ,  $j = 1, 2, \dots, N$  providing a discrete Monte Carlo approximation to  $p(X_t | Z_t)$ . Using these samples, we can also approximate the target distribution by a kernel density estimation [23–25]:

$$p(X_t | Z_t) \cong \sum_{j=1}^N w_t^{(j)} \mathcal{N}(X_t | X_t^{(j)}, h^2 V_t), \quad (14.29)$$

where  $\mathcal{N}(\cdot | X_t^{(j)}, h^2 V_t)$  is a multivariate Gaussian kernel with the mean  $X_t^{(j)}$  and covariance matrix  $h^2 V_t$ , such that the target density is a mixture of Gaussian distribution weighted by the normalized sample weight  $w_t^{(j)}$  computed by Eq. (14.25) and where  $V_t$  is the Monte Carlo variance.

Standard density estimation methods suggest that the overall scaling parameter  $h$  be chosen as a slowly decreasing function of  $N$ , so that kernel components are naturally more concentrated about their location  $X_t^{(j)}$  for large  $N$ . A traditional specification of  $h$  with normal kernel is given by

$$h = \left[ \frac{4}{(1+2d)N} \right]^{1/1+4d},$$

where  $d$  is the dimension of the state  $X$  [23]. It is shown that the choice of the kernel with mean  $X_t^{(j)}$  makes the kernel located about the existing sample values. However, as shown in [21], this results in a kernel density function that is overdispersed relative to the posterior sample. To correct this, the following shrinkage of kernel location is given by

$$p(X_t | Z_t) \cong \sum_{j=1}^N w_t^{(j)} \mathcal{N}\left(X_t | m_t^{(j)}, h^2 V_t\right), \quad (14.30)$$

with  $m_t^{(j)} = aX_t^{(j)} + (1-a)\bar{X}_t$ , where  $a = \sqrt{1-h^2}$ . With these kernel locations, the resulting normal mixture retains the mean  $\bar{X}_t$  and now has the correct variance  $V_t$ , hence the overdispersion is trivially corrected.

With the kernel representation of samples, a novel resampling scheme can be obtained as follows [26]. For  $i = 1, \dots, N$ :

### Kernel-Based Resampling

- Generate  $i^{(j)} \in \{1, \dots, N\}$  with a probability proportional to the weights  $\{w_t^{(j)}\}_{j=1}^N$ .
- Draw a sample  $\{\tilde{X}_t^i\}$  from the kernel  $\mathcal{N}\left(X_t | m_t^{i^{(j)}}, h^2 V_t\right)$ .
- Assign equal weight to each new sample  $\tilde{X}_t^i$ , that is,  $\tilde{w}_t^{(i)} = 1$ .

Recall that resampling was suggested as a method to reduce the degeneracy problem, which is prevalent in SMC methods. However, it was pointed out that resampling in turn introduced other problems and, in particular, the problem of loss of diversity among the Markov streams. This arises due to the fact that in the resampling stage, samples are drawn from a discrete distribution rather than a continuous one. If this problem is not addressed properly, it may lead to a severe degeneracy problem where all streams occupy the same point in the state space, giving a poor representation of the posterior density. On the contrary, instead of sampling from discrete representation of the posterior density, the new resampling scheme samples from a continuous approximation, which will mitigate the degeneracy problem occurring in the general sampling scheme. Another important advantage of the kernel representation is it makes it possible to extrapolate the estimated distribution to the entire population.

Next we consider “collapsing” the kernel-smoothed belief state discussed above via a mixture of far few number of component distributions [26] in each time step  $t$ . Typically, there is redundancy approximating  $p(X_t | Z_t)$  with a mixture of possibly several hundred Gaussian components; even very irregular densities can be adequately matched by using mixtures having far few components. One way of reducing this

redundancy is to collapse or cluster mixture components by simply replacing nearest neighboring components with a single averaged one.

With reference to the mixture equation (14.30), suppose that  $N$  is large, so that  $X_t^{(j)}$  are dense and there is high redundancy, and that for some  $i$  and  $k$ ,  $X_t^{(i)}$  and  $X_t^{(k)}$  are the closest component locations in terms of Euclidean distance. Then the mixture will be essentially similar to one reduced to  $N - 1$  components by combining components  $i$  and  $k$  into one with location  $(w_i X_t^{(i)} + w_k X_t^{(k)})/(w_i + w_k)$  and associated weight  $w_i + w_k$ . Let  $K$  be the number of components in the final collapsed mixture, a simple agglomerate routine proceeds as follows [23]:

### **Collapsing Mixture**

1. Find the index, say  $i$ , of samples with the smallest weight  $w_i$ .
2. Find the the nearest neighbor, say  $k$ , of  $i$ th sample  $X_t^{(i)}$ .
3. Remove components  $k$  and  $i$  from the mixture.
4. Combine components  $k$  and  $i$ , insert the averaged value  $\bar{X}_t = (w_i X_t^{(i)} + w_k X_t^{(k)})/(w_i + w_k)$  with the weight  $\bar{w} = w_i + w_k$  into the mixture.
5. Repeat the above procedure and stop only when the number of mixture components is  $K$ .

The resulting mixture has a form with  $N$  reduced to  $K$ , with locations based on the final  $K$ -averaged values, associated combined weights, and the same scale matrix  $V$  but new and larger window width  $h$  based on the current, reduced sample size  $K$ .

Typically, there is a trade-off between the number of components and the approximation accuracy. For example, if  $N$  is several hundred, then adequate approximations are unlikely to be achieved by using very small  $K$  values. Furthermore, a very small number of components will lead to a loss of diversity among samples, namely the “sample impoverishment” problem, which is severe in the case of small process noise. A large number of components lead to extra communication burden and energy consumption in the sensor network.

## **14.5 MULTIPLE-TARGET TRACKING AND CLASSIFICATION**

As stated in Section 14.2, we make use of the sequential Monte Carlo techniques within a leader-based tracking scheme. In this framework it is possible to consider each target *independently* as long as the other ones are far away. It is subsequently compulsory to deal with a varying number of targets. Indeed as we mentioned earlier, a target perturbing the measurements available to a sensor cannot be considered as noise. In this section we will describe the main body of the general algorithm depicted in Figure 14.1 (steps 2–6).

### **14.5.1 Related Work**

In order to deal with an unknown or varying number of targets  $r_t$ , several alternatives are available. A classical approach is to estimate  $r_t$  separately from the rest of the state space by using a hypothesis test, for instance, and then to treat the estimated  $r_t$  as the true number of targets for the estimation of the other variables [6]. Another

possibility is to compare several tracking hypotheses with a different number of targets. In [27], random sets and finite set statistics are employed to achieve this objective. In order to estimate the state of the system, it is then necessary to find the peaks in the probability hypothesis density (the equivalent of the probability density function for random sets) using, for example, the EM algorithm [28]. In [8, 9], it is proposed to cast the multiple-target tracking problem into that of filtering a jump Markov system (JMS), where the number of targets, and thus the dimensionality of the state, follows a Markov chain.

Our approach also makes use of the jump Markov system in modeling the varying number of targets. However, in order to meet the requirements of the sensor networks, we will focus on maintaining the computational complexity as low as possible. We will also incorporate the class-based resampling scheme described in Section 14.4 so as to tackle the issue of jointly tracking and classifying the targets.

### 14.5.2 The SMC Solution

**14.5.2.1 The JMS Formulation** We assume that the evolution of the number of targets  $r_t$  is independent of the previous state of the targets  $X_{t-1}$ . The model dynamic is thus given by (14.2), (14.3), (14.14), and

$$\pi_{r_t, r_{t-1}} = p(r_t | r_{t-1}). \quad (14.31)$$

Our state becomes  $(X_t, r_t)$  with  $X_t = \{x_{t,i}, \gamma_i\}_{i \in T_t}$  and where we recall that  $T_t$  is the set of active targets. Our goal remains to sequentially estimate  $p(X_t | Z_{1:t})$ . Because we are given a dynamic model, the SMC methods presented in Section 14.3 are well fit to solve this problem.

**14.5.2.2 Optimal Sampling Density** The first issue encountered in this approach resides in the choice of the sampling density. In fact, the optimal choice for the sampling density is  $q(X_t, r_t | X_{0:t-1}^{(j)}, r_{0:t-1}^{(j)}, Z_{1:t}) = p(X_t, r_t | X_{t-1}^{(j)}, r_{t-1}^{(j)}, Z_t)$  [13]. Clearly, it is impossible to sample directly from this distribution, and, even if we could, the weight update would also require the evaluation of  $p(Z_t | X_{t-1}^{(j)})$ , which does not admit a closed-form expression.

For the above reasons, using the prior distribution as the sampling density is often a reasonable choice (e.g., in the single-target scenario). When dealing with  $r_t$ , the problem is more subtle. If the probability of a new target appearance is small, only a few particles will increase their dimension and thus become accurate, when a new target enters the field. After the resampling stage, only a few number of particles will be kept, which will result in a loss of diversity within the particles and could lead to loosing track of the target of interest. On the other hand, assuming a large probability of appearance would lead to drawing many inaccurate samples and falsely generating new leader nodes. It is therefore imperative to include the current observation in the proposal distribution of  $r_t$ . The optimal sampling density can be written as

$$q(X_t, r_t | X_{0:t-1}^{(j)}, r_{0:t-1}^{(j)}, Z_{1:t}) = p(X_t | r_t, X_{t-1}^{(j)}, r_{t-1}^{(j)}, Z_t) p(r_t | X_{t-1}^{(j)}, r_{t-1}^{(j)}, Z_t). \quad (14.32)$$

By Bayes' rule we get

$$p(r_t | X_{t-1}^{(j)}, r_{t-1}^{(j)}, Z_t) \propto p(Z_t | X_{t-1}^{(j)}, r_{t-1}^{(j)}, r_t) p(r_t | r_{t-1}^{(j)}), \quad (14.33)$$

where

$$\begin{aligned} p(Z_t | X_{t-1}^{(j)}, r_{t-1}^{(j)}, r_t) &= \sum_{a_t} p(Z_t | X_{t-1}^{(j)}, a_t, r_{t-1}^{(j)}, r_t) p(a_t | r_t, m_t) \\ &= \sum_{a_t} p(a_t | r_t, m_t) \left( \frac{1}{V_{\text{meas}}} \right)^{m_t^0(a_t)} \prod_{\{m | a_t(m) \neq 0\}} p(z_t^m | x_{t-1, a_t(m)}^{(j)}). \end{aligned} \quad (14.34)$$

Finally the computation of (14.34) requires

$$p(z_t^m | x_{t-1, a_t(m)}^{(j)}) = \int p(z_t^m | x_{t, a_t(m)}) p(x_{t, a_t(m)} | x_{t-1, a_t(m)}^{(j)}) dx_{t, a_t(m)}. \quad (14.35)$$

**14.5.2.3 Choice of Sampling Density** The quantity in (14.35) can be approximated by an unscented transform [29], as proposed in [8, 9], or simpler by using the mean or mode of this distribution. In our simulations, this approximation appeared sufficient for the previously existing targets. Our approximation of (14.35) will then become

$$p(z_t^m | x_{t-1, i}^{(j)}, a_t(m) = i) \simeq p(z_t^m | \mu_{t, i}^{(j)}), \quad \forall i \in T_{t-1} \cap T_t, \quad (14.36)$$

$$\mu_{t, i}^{(j)} = E[x_{t, i} | x_{t-1, i}^{(j)}, \gamma_i]. \quad (14.37)$$

For the newly appeared targets, there is no available  $x_{t-1, i}^{(j)}$ , and in that case, (14.36) is not accurate. Therefore, we need to compute  $p(z_t^m | a_t(m) \in T_t \cap \overline{T_{t-1}})$ . If no prior information on the position of the newly appeared target is available, we can use a uniform distribution of the position in the sensed volume. When the sensing model is given by (14.4), this can be done analytically and we get (cf. Appendix)

$$\begin{aligned} p(z_t^m | a_t(m) \in T_t \cap \overline{T_{t-1}}) &= \frac{\ln(10)}{10\eta d_{\max}^2} 10^{-\frac{z}{10\eta} + \frac{R \ln(10)}{200\eta^2} + \frac{K}{10\eta}} \\ &\times \left( \Phi \left( z - K + \frac{10\eta \ln(d_{\max}^2)}{\ln(10)}; \frac{R \ln(10)}{10\eta}, R \right) \right. \\ &\quad \left. - \Phi \left( z - K + \frac{10\eta \ln(d_{\min}^2)}{\ln(10)}; \frac{R \ln(10)}{10\eta}, R \right) \right), \end{aligned} \quad (14.38)$$

where  $\Phi(\cdot; \mu, \sigma^2)$  denotes the CDF of a Gaussian distribution  $\mathcal{N}(\mu, \sigma^2)$ .

Alternatively, if a specific prior on the position is used, it is possible to sample  $N$  particles for the newly appeared target, and approximate  $p(z | a_t(m) \in T_t \cap \overline{T_{t-1}})$  by Monte Carlo integration. The same samples can afterwards be used when drawing  $X_{t+1}^{(j)}$ .

In order to approximate  $p(X_t | r_t, X_{t-1}^{(j)}, Z_t)$  in (14.32), it is possible to perform a local linearization of the dynamic model similar to the extended Kalman filter [13, 15], but this would result in a heavier computational load, and thus consumes more power, whereas sampling from the prior is sufficient once the number of targets

is given. Subsequently, we propose to use the following proposal density:

$$q(X_t, r_t | X_{t-1}^{(j)}, r_{t-1}^{(j)}, Z_t) \propto p(X_t | r_t, X_{t-1}^{(j)}, r_{t-1}^{(j)}) \left( p(r_t | r_{t-1}^{(j)}) p(Z_t | \mu_t^{(j)}(r_t)) \right). \quad (14.39)$$

Nevertheless, for the newly appeared targets, sampling from the prior distribution can be inefficient if we assume a uniform distribution of the position within the entire area covered by the leader node. As in (14.38), in such a case an analytical formula for the distribution of the distance to the node is available (cf. Appendix). Let  $\zeta = d_s(x_{t,i})^2$ ,  $i \in T_t \cap \overline{T_{t-1}}$ , we get

$$p(\zeta | z_t^m, a_t(m) = \text{new target}) = \text{lognormal} \left( \frac{\ln(10)}{10\eta} (K - z), \frac{R \ln(10)^2}{100\eta^2} \right). \quad (14.40)$$

**14.5.2.4 Extension of Class-Based Resampling Scheme** In Section 14.4, we proposed a particle filter approach for dealing with the classification. As the number of targets is now greater than one, a direct generalization of the proposed algorithm would be to keep a sufficient number of particles per class association vector. However this would result in a substantial increase in the minimum number of particles we should keep. Since we use a leader-based tracking scheme, our attention is mainly focused on the first target. Hence, for the other targets we chose to depart from the real static evolution of the class and allow a small probability of switching between classes so as not to settle in the wrong class. The robust classification for those targets is actually made by the leader node responsible for them.

**14.5.2.5 Adjustment of JMS Formulation for Sensor Networks** When an estimate of  $E\{\psi(X_{0:t})|Z_{1:t}\}$  is needed, only a simple weighted average as shown in (14.22) is usually required. In the special case of JMS, there is an ambiguity in the order of the targets within a particle. Indeed when only the number of targets  $r_t$  is used to describe the set of targets  $T_t$ , it is possible that within a particle stream  $X_{0:t}^{(j)}$ , a given target dies and reappears. Because we have no information about the identity of a target, the order in the state vector is then shuffled. This issue is addressed in [27] and solved by using the EM algorithm in order to find the peaks of the probability hypothesis density. In [8, 9], rather curiously, this problem is not mentioned.

Because one sensor node can only sense in a small range, we make the assumption that no more than two targets can be at the same time in the field covered by the current leader node. We also assume that a target cannot die but can only leave the covered field. If the tracking is accurate, the first target will then always be considered. Therefore, we will assume that the set of active targets can only take two values,  $T_t = \{1\}$  or  $T_t = \{1; 2\}$ . We denote by  $P_b$  the probability of switching from  $\{1\}$  to  $\{1, 2\}$  (birth), and by  $P_v$  the probability of switching from  $\{1, 2\}$  to  $\{1\}$  (vanishing). With these assumptions, we eliminate the problem of the ordering ambiguity and the estimation becomes a trivial application of (14.22).

**14.5.2.6 Leader Node Generation** Given the posterior probability that a target has appeared, it is possible to decide whether the second target should be associated to a leader node or not. We next describe the proposed scheme for making this decision

(cf. Fig. 14.1, step 3). We based this part on our simulations and use an hysteresis thresholding on  $\hat{P}(r_t = 2|Z_{1:t})$  to make a decision on the number of targets in the field and avoid as much as possible false launching of new leader nodes. Let  $r_t^D$  denote this decision.

- Set  $r_0^D = 1$ .
- If  $\hat{P}(r_t = 2|Z_{1:t}) > P_{UP} \text{ Threshold}$  for a sufficiently large period of time, set  $r_t^D = 2$ .
- If  $\hat{P}(r_t = 1|Z_{1:t}) > P_{Down} \text{ Threshold}$  for a sufficiently large period of time, set  $r_t^D = 1$ .
- Else, let  $r_t^D = r_{t-1}^D$ .
- If  $r_t^D = r_{t-1}^D + 1$ , send a query to the neighbor sensors.

When  $r_t^D = r_{t-1}^D + 1$ , the current leader node sends a query to its neighbor sensors so as to know if any of them is tracking a target (cf. Fig. 14.1, step 4). The rest of this scheme is described in Section 14.2 (cf. Fig. 14.1, steps 5 and 6).

## 14.6 SENSOR SELECTION

Because our tracking scheme relies on a leader-based algorithm, the sensor selection step is essential. In this section, we consider an information-driven sensor selection algorithm (cf. Fig. 14.1, step 7). The choice of the sensor will determine the efficiency of the tracking (and thus of the classification) and the resource consumption of the nodes (e.g., power use). Depending on the cost of a handoff to a next leader node, it could be necessary to penalize such an operation. A trade-off is typically to be made between the information gain and the total cost [7]. A simple formulation of the sensor selection scheme can be given as that of maximizing the expectation of an objective function. We denote by  $s_t$  the sensor selected at time  $t$  and denote  $p_s(Z_{t+1}|Z_{1:t})$  to emphasize the dependence of  $p(Z_{t+1}|Z_{1:t})$  on  $s$ . This selection can be written as

$$s_{t+1} = \arg \max_s \left( E_{p_s(Z_{t+1}|Z_{1:t})} [\alpha \Upsilon_{\text{utility}}(s) + (1 - \alpha) \Upsilon_{\text{cost}}(s)] \right), \quad (14.41)$$

where  $\Upsilon_{\text{utility}}$  is the information gain by fusion of a set of measurements from the sensor  $s$ ,  $\Upsilon_{\text{cost}}$  is the cost of choosing the sensor, and  $\alpha$  is the relative weight between those quantities. Both costs are selected to be nonnegative functions.

The function  $\Upsilon_{\text{cost}}$  is characterized by link bandwidth, transmission latency, node battery power reserve, and the like. In our case this is the cost of handing the current belief state off to sensor  $s$ , acquiring data at sensor  $s$ , and combining the data with the current belief. It is thus sound to consider it as being a function of the distance (as a crude measure of the amount of energy required to transmit the data from sensor  $s_t$  to sensor  $s$ ) between the leader node  $s_t$  and the sensor  $s$ . Then this quantity is deterministic and the selection criterion would be an immediate extension of the most informative selection scheme. In this section we will therefore focus on the expected information gain  $E_{p_s(Z_{t+1}|Z_{1:t})} [\Upsilon_{\text{utility}}(s)]$  and propose a Bayesian sensor selection criterion. As evidenced by the presence of the expectation, such a selection criterion is based on the current belief only and does not use any new measurements.

In order to measure the information gain in choosing sensor  $s$ , we will make use of the notion of mutual information, which is a common criterion to measure the reduction

of uncertainty in a random variable due to the knowledge of another one [30]. This criterion is also advocated in [2] where its approximation relies on a grid-based method.

Let  $U \in \mathcal{U}$ ,  $V \in \mathcal{V}$ ,  $W \in \mathcal{W}$  be random variables having a conditional density  $p(u, v|w)$ . Conditioned on a single realization  $w$  of  $W$  (and not conditioned on the random variable  $W$ ), the mutual information between  $U$  and  $V$  is given by

$$I(U; V|W = w) \triangleq E_{p(u,v|w)} \left[ \log \frac{p(u, v|w)}{p(u|w)p(v|w)} \right]. \quad (14.42)$$

Let  $p_1$  and  $p_2$  be two probability densities, the Kullback–Leibler (KL) divergence between  $p_1$  and  $p_2$  is defined by

$$D(p_1||p_2) \triangleq E_{p_1} \left[ \log \frac{p_1}{p_2} \right]. \quad (14.43)$$

#### 14.6.1 Expected Information Gain

As mentioned earlier, within our tracking framework, we are mainly concerned about the *first* target of each leader node. We will consequently consider the information conveyed about this target only. We have

$$s_{t+1} = \arg \max_s I_s(x_{t+1,1}; \tilde{z}_{t+1}|Z_{1:t}) \quad (14.44)$$

where the conditioning is on the observed realization of  $Z_{1:t}$  and the sensors  $s$  of interest are in a specified neighborhood of  $s_t$  (e.g., the three closest sensors to the predicted position of the first target). Also,  $\tilde{z}_{t+1}$  is a random variable denoting the measurement that would arise from the first target. There is indeed no need to consider a data association problem here.

From the definition (14.42) of the mutual information we have

$$\begin{aligned} E[\Upsilon_{\text{utility}}(s)] &= I_s(x_{t+1,1}; \tilde{z}_{t+1}|Z_{1:t}) \\ &= E_{p_s(x_{t+1,1}, \tilde{z}_{t+1}|Z_{1:t})} \left[ \log \frac{p_s(x_{t+1,1}, \tilde{z}_{t+1}|Z_{1:t})}{p(x_{t+1,1}|Z_{1:t})p_s(\tilde{z}_{t+1}|Z_{1:t})} \right]. \end{aligned} \quad (14.45)$$

After simple calculations we get

$$E[\Upsilon_{\text{utility}}(s)] = E_{p_s(\tilde{z}_{t+1}|Z_{1:t})} \left[ D(p_s(x_{t+1,1}|Z_{1:t}, \tilde{z}_{t+1})||p(x_{t+1,1}|Z_{1:t})) \right]. \quad (14.46)$$

Our selection criterion can thus also be seen as that of maximizing the average KL distance between the one-step-ahead filtering density  $p_s(x_{t+1,1}|Z_{1:t}, \tilde{z}_{t+1})$  and the predictive density  $p(x_{t+1,1}|Z_{1:t})$ . The latter being performed with respect to the predictive density of the measurements  $p_s(\tilde{z}_{t+1}|Z_{1:t})$ .

This quantity will be computed as proposed in [8] by a Monte Carlo method. From (14.42) we get

$$E[\Upsilon_{\text{utility}}(s)] = E_{p_s(\tilde{z}_{t+1}|x_{t+1,1})p_s(x_{t+1,1}|Z_{1:t})} \left[ \log \frac{p_s(\tilde{z}_{t+1}|x_{t+1,1})}{p_s(\tilde{z}_{t+1}|Z_{1:t})} \right]. \quad (14.47)$$

In order to estimate (14.47), it is possible to use  $N$  samples  $\{(x_{t+1,1}^{(j)}, \tilde{z}_{t+1}^{(j)})\}_{j=1}^N$  drawn from the joint distribution  $p_s(\tilde{z}_{t+1}|x_{t+1,1})p(x_{t+1,1}|Z_{1:t})$  and get

$$E[\Upsilon_{\text{utility}}(s)] \simeq \frac{1}{N} \sum_{j=1}^N \log \frac{p_s(\tilde{z}_{t+1}^{(j)}|x_{t+1,1}^{(j)})}{p_s(\tilde{z}_{t+1}^{(j)}|Z_{1:t})}. \quad (14.48)$$

We will first approximate the posterior distribution  $p(x_{t+1,1}|Z_{1:t})$ . Using the trial distribution  $q(x_{t+1,1}) = p(x_{t+1,1}|x_{t,1}^{(j)})$ , we can draw samples  $\{x_{t+1,1}^{(j)}, w_{t+1}^{(j)}\}_{j=1}^N$  with the importance weight given by

$$w_{t+1}^{(j)} = \frac{p(x_{t+1,1}^{(j)}|Z_{1:t})}{q(x_{t+1,1}^{(j)}|Z_{1:t})} = w_t^{(j)}. \quad (14.49)$$

Based on the importance sampling principle,  $\{x_{t+1,1}^{(j)}, w_{t+1}^{(j)}\}_{j=1}^N$  is easily shown to be properly weighted with respect to the distribution  $p(x_{t+1,1}|Z_{1:t})$ . We can now sample  $\tilde{z}_{t+1}^{(j)} \sim p_s(\tilde{z}_{t+1}|x_{t+1,1}^{(j)})$ . By composition it can also be shown that the set of samples and weights,  $\{(x_{t+1,1}^{(j)}, \tilde{z}_{t+1}^{(j)}, w_{t+1}^{(j)})\}_{j=1}^N$ , is properly weighted with respect to  $p_s(x_{t+1,1}, \tilde{z}_{t+1}|Z_{1:t})$ . We now need to approximate  $p_s(\tilde{z}_{t+1}^{(j)}|Z_{1:t})$ , which is not directly available with those samples. However, by expanding this term as

$$p_s(\tilde{z}_{t+1}|Z_{1:t}) = \int p_s(\tilde{z}_{t+1}|x_{t+1,1})p(x_{t+1,1}|Z_{1:t})dx_{t+1,1}, \quad (14.50)$$

and using the set of samples and weights  $\{x_{t+1,1}^{(j)}, w_{t+1}^{(j)}\}_{j=1}^N$ , we get the following approximation:

$$\hat{p}_s(\tilde{z}_{t+1}^{(j)}|Z_{1:t}) = \sum_{k=1}^N w_{t+1}^{(k)} p_s(\tilde{z}_{t+1}^{(j)}|x_{t+1,1}^{(k)}). \quad (14.51)$$

Finally, the expected information gain can be approximated by

$$E[\Upsilon_{\text{utility}}(s)] \simeq \sum_{j=1}^N w_{t+1}^{(j)} \log \frac{p_s(\tilde{z}_{t+1}^{(j)}|x_{t+1,1}^{(j)})}{\hat{p}_s(\tilde{z}_{t+1}^{(j)}|Z_{1:t}^{(j)})}. \quad (14.52)$$

This scheme can appear computationally intensive, and, thus, solutions might be needed to reduce its complexity. Our goal here is not to approximate accurately the aforementioned expectation but only to find the index  $s$  maximizing it. Hence, a rough approximation should be sufficient. One option would be to select  $P$  (with  $P \ll N$ ) initial samples by a usual resampling stage and then perform this scheme with equal weights. We could also think of inserting this step every  $k$  steps or dynamically performing this scheme when, for instance, the Monte Carlo variance of the particles gets above a given threshold.

Another issue in the sensor selection is that a given sensor can already be assigned to the tracking of another target. In such a case, the belief would be handed off to the second best node and so on.

## 14.7 SIMULATION RESULTS

To illustrate the performance of the proposed algorithm, simulations are performed for two different scenarios. The first one presents a crossing of two targets from different classes. Both targets are given a leader node at time  $t = 0$ . The second scenario uses the same trajectories but only one of the targets is given a leader node at time  $t = 0$ . In both scenarios, only two possible motion models are considered as shown in Section (14.2.3).

For all simulations we take the following parameters. The probability of detection is  $P_D = 0.95$  when the target is in the range of the sensor. The transmission power for each sensor is  $K = 9 \text{ dBm}$  and the path loss index is  $\eta = 3$ . We assume that each sensor can only sense objects within a range of [4 m, 50 m], which approximatively corresponds to a measurement range of [-41 dBm, -9 dBm]. The number of clutter measurements is given by a Poisson distribution with parameter  $\lambda V_{\text{meas}} = 1$  where  $V_{\text{meas}} = 33 \text{ dBm}$ . The variance of the measurement noise is taken as  $R = \frac{1}{2}$ . For the first target 500 particles per leader node are used with a minimum of 80 per class. For the second target of each leader node, the probability of changing the class is chosen as 0.01.

We use equal probability of appearing and vanishing  $P_b = P_v = 0.1$ . Both up and down thresholds for the decision on the number of targets are set as 0.4, and the length of the window is 20 time steps. The time step is chosen to be  $\tau = 1 \text{ s}$ . For the constant velocity motion model, we take  $\sigma_x = \sigma_y = 0.005$  and for the constant turn rate model we use  $\sigma_x = \sigma_y = 0.006$ . The field is  $700 \times 500 \text{ m}^2$  covered by 300 randomly scattered sensor nodes.

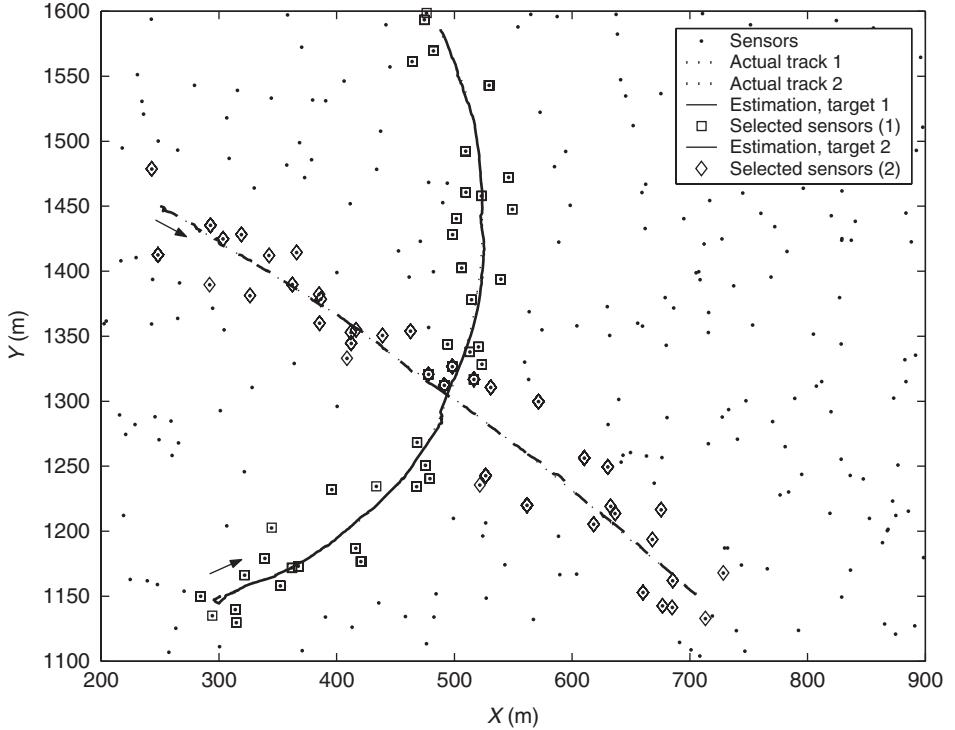
The prior for the target class is assumed to be uniform. The prior for the initial state is Gaussian with the true mean and the covariance matrix

$$\Sigma = \begin{pmatrix} 10^2 & 0 & 0 & 0 \\ 0 & 10^2 & 0 & 0 \\ 0 & 0 & 0.005^2 & 0 \\ 0 & 0 & 0 & 0.005^2 \end{pmatrix}. \quad (14.53)$$

For the second target (from the point of view of a leader node) the issue of the prior is more subtle. Indeed, we need to be able to sample from the prior at each time. Therefore, we assume that such a target could appear in a given Gaussian state region with the mean being the true state of the other target at time  $t = 269$  (the real time of appearance varies between 265 and 275 depending on the chosen sensor) and variance  $\Sigma$ . The exact time of appearance is not known and thus the prior remains constant during time. However, such a prior does not account for the motion of this target. Therefore, we update the prior by taking the mean equal to the estimate (but keeping the initial variance) when the posterior probability of having two targets is larger than 0.9.

### 14.7.1 Crossing of Two Tracked Targets

The first scenario corresponds to the crossing of two initially tracked targets. This situation should be the most typical one. In order to illustrate the performance of the classification, the first target (starting at the lower left corner of Fig. 14.2) belongs to the second class, that is, the coordinated turn rate model (14.8), and the second target



**Figure 14.2** Actual and estimated trajectories for two known targets, one sample run.

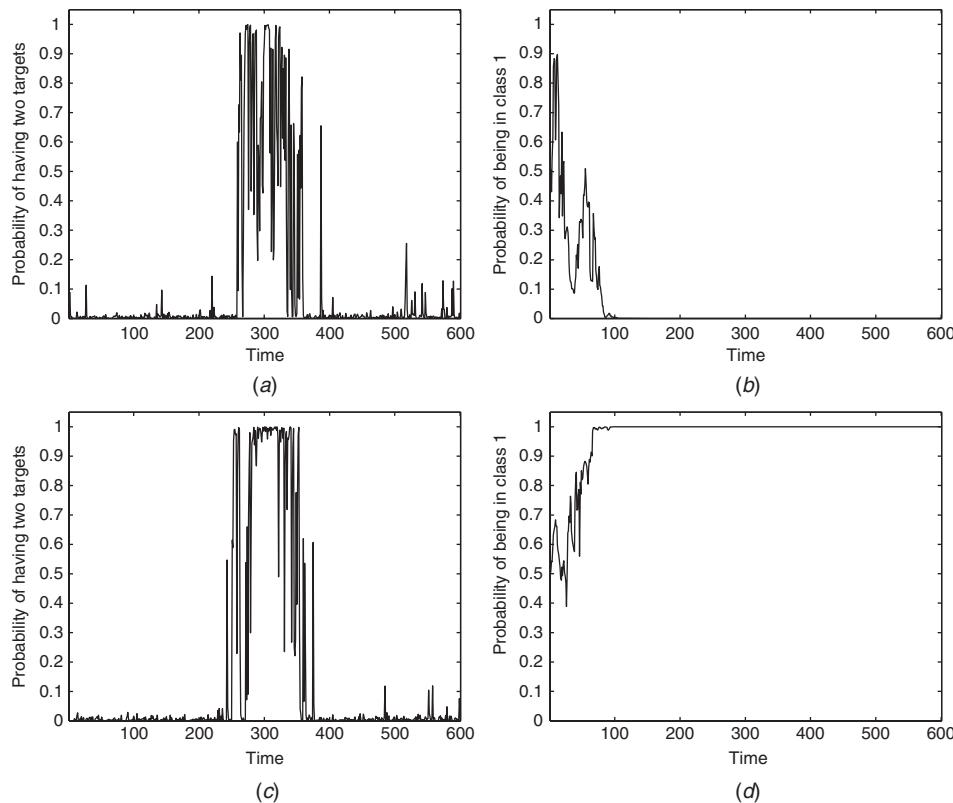
(starting at the upper left corner of Fig. 14.2) belongs to the first class, that is, the constant velocity model (14.6). The true trajectories of both targets are represented in Figure 14.2 with dotted lines. The arrows indicate the direction of the motion.

We show in Figure 14.3, on a single run and for each leader node, the classification of its first target and the detection of an entering target. We can see that for both leader nodes, the probability of having two targets in the field jumps almost as soon as the target enters the field. For the classification, we can see that during the initial steps, the first leader node misclassifies its target, but is able to recover and correctly classifies the target after some time.

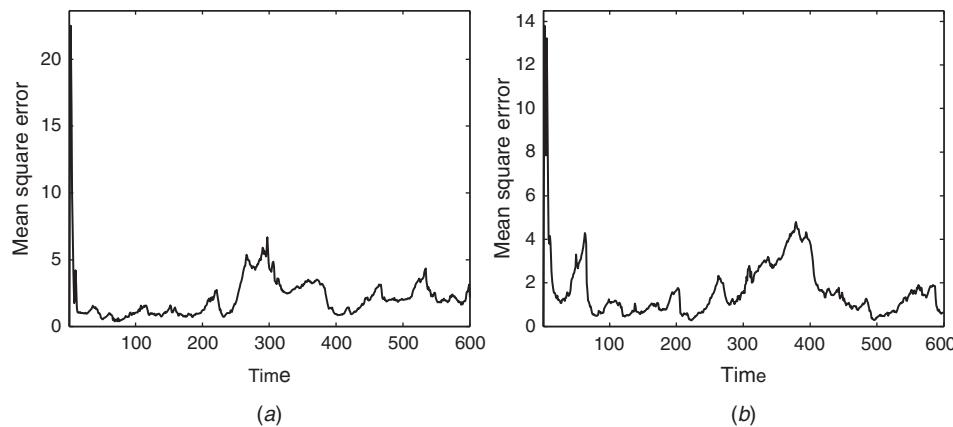
To evaluate the performance of the proposed algorithm in this case, we have performed 20 consecutive runs of the scenario. The true trajectories are kept identical to those shown in Figure 14.2 but independent measurements are simulated for each run. For each time step  $t$ , we use the square distance between the true position of the target and the estimated one as a measure of performance. We show in Figure 14.4 the average square error for each time step over those 20 runs. Our algorithm is able to accurately track each of the targets. The performance of the classification and target detection are shown in the following more complex scenario.

#### 14.7.2 Crossing of Tracked Target with Unknown Target

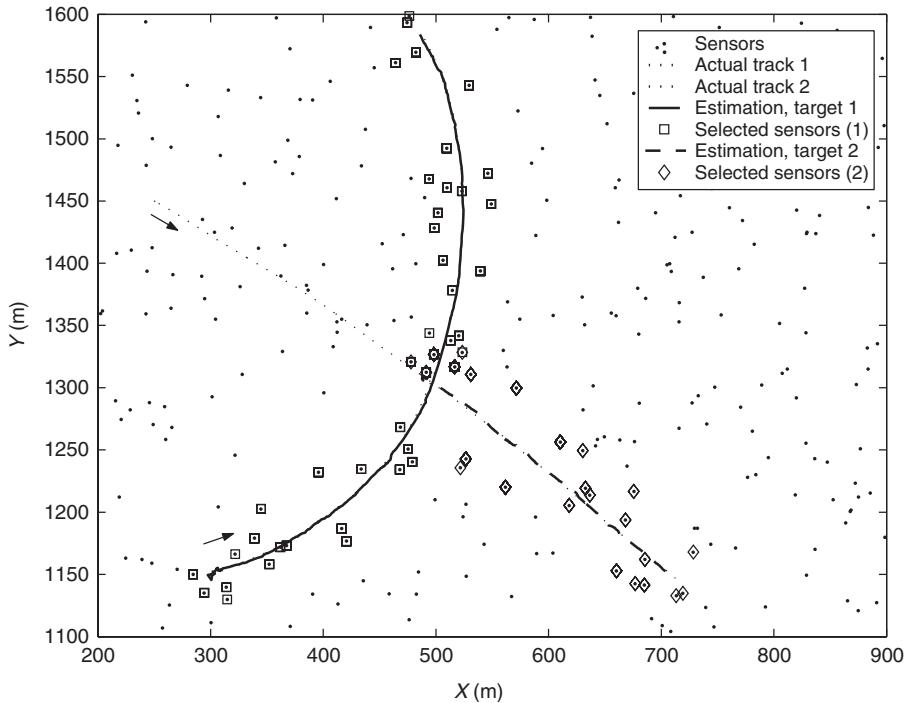
The second scenario corresponds to the crossing of two targets when only one is given a leader node at time  $t = 0$ . This situation illustrates the ability of our algorithm to



**Figure 14.3** Probabilities of having two targets in the field: (a) first leader node and (c) second leader node. Probabilities that the first target of the leader node is from the first class: (b) first leader node and (d) second leader node. One sample run.



**Figure 14.4** Mean-square error for the position on 20 runs: (a) first leader node and (b) second leader node.

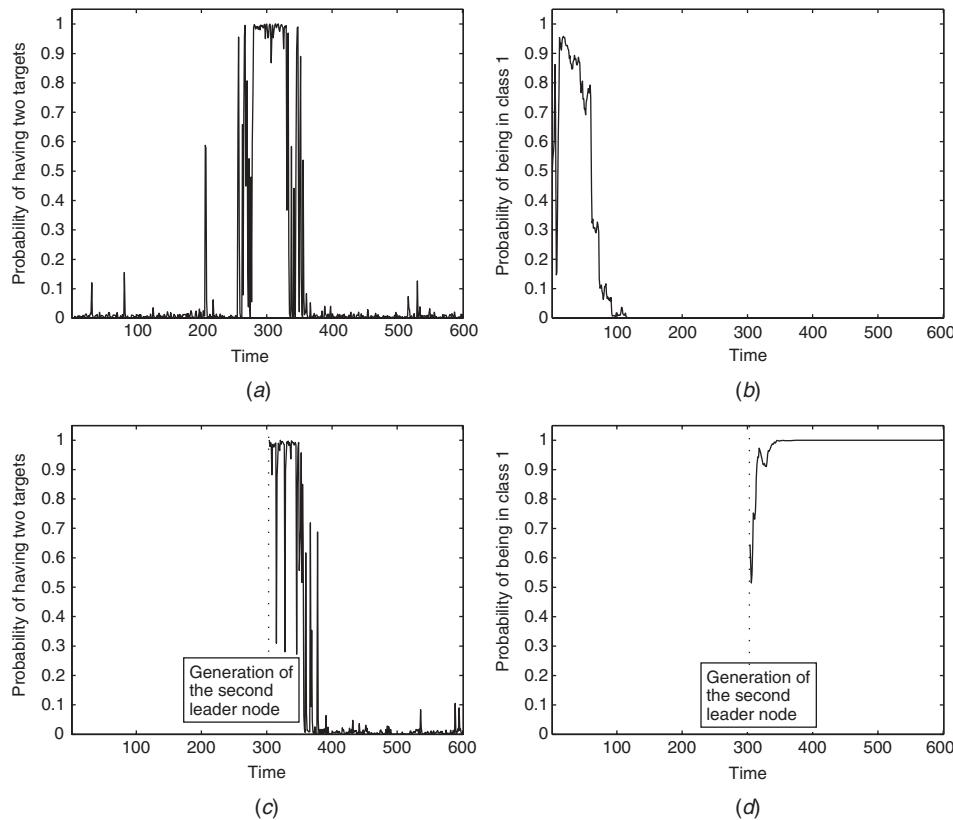


**Figure 14.5** Actual and estimated trajectories with an unknown target, one sample run.

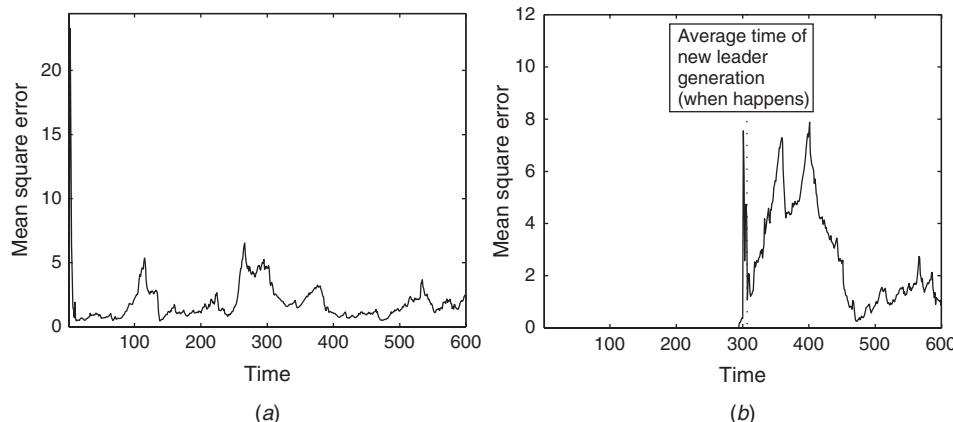
detect newly appeared targets and to track and classify them accurately. In order to show the performance of the classification, the first target (starting at the lower left corner of Fig. 14.5) belongs to the second class, that is, the coordinated turn rate model (14.8), and the second target (starting at the upper left corner of Fig. 14.5) belongs to the first class, that is, the constant velocity model (14.6). The true trajectories of both targets are identical to those in the first scenario. They are represented in Figure 14.5 with dotted lines. The arrows indicate the direction of the motion.

We show in Figure 14.6, on a single run and for each leader node, the classification of its first target and the detection of an entering target. We can see that for the first leader node, the probability of having two targets in the field jumps almost as soon as the second target enters the field. The second leader node is then generated and accurately tracks its target. For this leader node, we can see that the probability of having two targets in the field remains very large as long as the other target remains in the field. For the classification, we can see that during the initial steps, the first leader node almost locks onto the wrong class but is still able to recover and correctly classifies the target after some time.

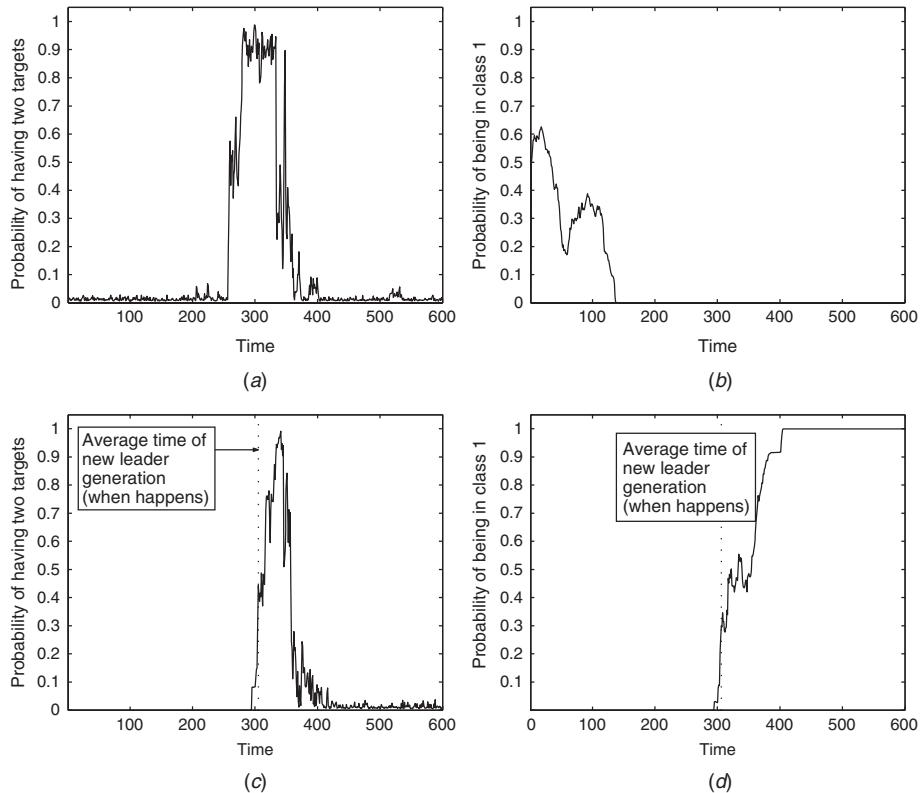
To evaluate the performance of the proposed algorithm, we have also performed 20 consecutive runs of this scenario (same trajectories, independent measurements simulated for each run). We can see that our algorithm is, in most cases, able to accurately generate a new leader node for the appearing target. However, in 5 runs out of 20, no leader node has been generated. We have indeed put strict constraints for a new leader node to be launched (cf. Section 14.5) in order to get a good initialization for this leader node. Figure 14.7 shows the mean-square error (MSE) on the position for



**Figure 14.6** Probabilities of having two targets in the field: (a) first leader node and (c) newly generated second leader node. Probabilities that the first target of the leader node is from the first class: (b) first leader node and (d) newly generated second leader node. One sample run.



**Figure 14.7** (a) Mean square error on 20 consecutive runs for the position of the first target of the first leader node. (b) MSE for the position of the first target of the second leader node when generated (15 out of 20).



**Figure 14.8** Probabilities of having two targets in the field averaged over all runs: (a) first leader node (20 consecutive runs) and (c) newly generated second leader node (15 runs out of 20). Probabilities that the first target of the leader node is from the first class averaged over all runs: (b) first leader node (20 consecutive runs) and (d) newly generated second leader node (15 runs out of 20).

both targets (for the second target it corresponds to the MSE on the 15 runs for which a leader node is generated).

The performance of the classification and detection of an entering target is shown in Figure 14.8. The average classification converges to the true value after some time steps. The average probability of having two targets in the field appears smoother than in the single-run case because the time of appearance and disappearance of the other target depends on the order of the selection of the leader nodes. Moreover for the second leader node, this also depends on the time when it is initiated.

## 14.8 CONCLUSION

In this chapter we have considered the application of the sequential Monte Carlo methodology to the problem of jointly tracking a possibly varying number of targets and identifying them down to a specific class (leading to accurate motion model of the target). The scenario under consideration is a collaborative sensor network where a nonlinear sensing model is assumed. The requirements of such networks (e.g., low power consumption,

distributed processing, etc.) have led us to the use of a leader-based scheme so as to solve this complex problem through collaboration among the sensors and by dividing the task into several easier, localized ones. The classical sequential Monte Carlo methods deal with the recursive estimation of a single-state process. Three extensions were presented in order to deal with the static evolution model of the class parameter, the varying dimension of the considered space, and the specificity of the leader-based tracking scheme. The first contribution resided in the design of a class-based resampling scheme leading to a robust classification of the targets while allowing a more computational load where needed. Furthermore we have used the filtering of jump Markov systems to deal with the varying number of targets and provided a scheme with low computational complexity. Finally, we have presented an SMC method to solve the problem of information-driven sensor selection. Our algorithm is able to detect a newly appeared target, and simulations have shown that, in most cases, the algorithm generates a new leader node that accurately tracks and classifies such targets.

## APPENDIX: DERIVATIONS OF (14.38) AND (14.40)

### Derivation of (14.38)

The position is assumed uniformly distributed in a disk centered on the sensor with inner radius  $d_{\min}$  and outer radius  $d_{\max}$ . We denote by  $\mathcal{C}_d^D$  a disk with radii  $d$  and  $D$ . Let  $\zeta$  denote the squared distance to the sensor:

$$P(\zeta \leq S) = P\left(x \in \mathcal{C}_{d_{\min}}^{\sqrt{S}}\right) = \frac{\text{Vol}(\mathcal{C}_{d_{\min}}^{\sqrt{S}})}{\text{Vol}(\mathcal{C}_{d_{\min}}^{d_{\max}})} = \frac{S - d_{\min}^2}{d_{\max}^2 - d_{\min}^2}, \quad (14.54)$$

$$p_\zeta(\zeta) = \frac{1}{d_{\max}^2} \mathbf{1}_{\zeta \in [d_{\min}^2, d_{\max}^2]}. \quad (14.55)$$

Let  $u = \tilde{h}(\zeta) = a - b \ln(\zeta)$  where  $a$  and  $b$  are constants;  $\tilde{h}$  is bijective and we have  $\zeta = e^{a-u/b}$ . Let  $v \sim \mathcal{N}(0, R)$ . Because of the bijectivity of  $\tilde{h}$  we have [31]

$$\begin{aligned} p_u(u) &= \frac{p_\zeta(e^{a-u/b})}{|\tilde{h}'(e^{a-u/b})|} \\ &= \frac{1}{bd_{\max}^2} e^{a-u/b} \mathbf{1}_{e^{a-u/b} \in [d_{\min}^2, d_{\max}^2]} \\ &= \frac{1}{bd_{\max}^2} e^{\frac{a-u}{b}} \mathbf{1}_{u \in [\tilde{h}(d_{\max}^2), \tilde{h}(d_{\min}^2)]}. \end{aligned} \quad (14.56)$$

Let  $z = u + v$ , we then have [31]

$$\begin{aligned} p_z(z) &= p_u(u)p_v(v) = \int p_u(z-\xi)p_v(\xi)d\xi \\ &= \int \frac{1}{bd_{\max}^2} e^{a-(z-\xi)/b} \mathbf{1}_{z-\xi \in [\tilde{h}(d_{\max}^2), \tilde{h}(d_{\min}^2)]} \frac{1}{\sqrt{2\pi R}} e^{-\xi^2/2R} d\xi \\ &= \int_{\xi \in [z-\tilde{h}(d_{\min}^2), z-\tilde{h}(d_{\max}^2)]} \frac{1}{bd_{\max}^2} \frac{1}{\sqrt{2\pi R}} \exp\left(-\frac{\xi^2}{2R} + \frac{\xi}{b} - \frac{z}{b} + \frac{a}{b}\right) d\xi \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{bd_{\max}^2} \exp\left(-\frac{z}{b} + \frac{a}{b} + \frac{R}{2b^2}\right) \int_{\xi \in [z - \tilde{h}(d_{\min}^2), z - \tilde{h}(d_{\max}^2)]} \exp\left[\frac{(\xi - R/b)^2}{2R}\right] d\xi \\
&= \frac{1}{bd_{\max}^2} \exp\left(-\frac{z}{b} + \frac{a}{b} + \frac{R}{2b^2}\right) \left( \Phi\left(z - \tilde{h}(d_{\max}^2); \frac{R}{b}, R\right) \right. \\
&\quad \left. - \Phi\left(z - \tilde{h}(d_{\min}^2); \frac{R}{b}, R\right) \right)
\end{aligned} \tag{14.57}$$

where  $\Phi(\cdot; \mu, \sigma^2)$  denotes the CDF of a Gaussian distribution  $\mathcal{N}(\mu, \sigma^2)$ .

### Derivation of (14.40)

We use the same notations as above. We then have  $u = z - v$ ,  $p(u|z) \sim \mathcal{N}(z, R)$ , and

$$\begin{aligned}
p(\zeta|z) &= \frac{p_{u|z}(a - b \ln(\zeta))}{|\frac{\partial h^{-1}}{\partial u}(a - b \ln(\zeta))|} \\
&= \frac{b}{\zeta \sqrt{2\pi R}} \exp\left[-\frac{1}{2\frac{R}{b^2}}(\ln(\zeta) + \frac{z-a}{b})\right] \\
&= \text{lognormal}\left(\frac{z-a}{b}, \frac{R}{b^2}\right).
\end{aligned} \tag{14.58}$$

## REFERENCES

1. I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
2. J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *EURASIP J. Appl. Signal Proc.*, no. 4, pp. 378–391, Mar. 2003.
3. X. Li and V. Jilkov, "A survey of maneuvering target tracking: Dynamic models," in *SPIE: Signal and Data Processing of Small Targets 2000*, vol. 4048-22, Apr. 2000.
4. Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic, New York, 1988.
5. T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. Oceanic Eng.*, vol. 8, no. 3, pp. 173–184, July 1983.
6. C. Hue, J.-P. L. Cadre, and P. Perez, "Sequential Monte Carlo methods for multiple target tracking and data fusion," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 309–325, Feb. 2002.
7. M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor network," *Int J. High Perf. Comput. Appl.*, vol. 16, no. 3, pp. 293–314, Mar. 2002.
8. A. Doucet, B. Vo, C. Andrieu, and M. Davy, "Particle filtering for multi-target tracking and sensor management," in *Int'l Conf. Information Fusion*, Vol. 1, 2002, pp. 474–481.
9. C. Andrieu, M. Davy, and A. Doucet, "Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1762–1770, July 2003.

10. T. Vercauteren, D. Guo, and X. Wang, "Joint multiple target tracking and classification in collaborative sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 714–723, 2005.
11. A. Kong, J. Liu, and W. Wong, "Sequential imputations and bayesian missing data problems," *J. Am. Statist. Assoc.*, vol. 89, no. 425, pp. 278–288, Mar. 1994.
12. J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *J. Am. Statist. Assoc.*, vol. 93, no. 443, pp. 1032–1044, 1998.
13. A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.
14. A. Doucet, N. de Freitas, and N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, 2001.
15. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
16. R. Chen and J. Liu., "Mixture Kalman filters," *J. R. Statist. Soc. B*, vol. 62, pp. 493–509, 2000.
17. S. Challa and G. W. Pulford, "Joint target tracking and classification using radar and ESM sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 3, pp. 1039–1055, 2001.
18. S. Herman and P. Moulin, "A particle filtering approach to FM-band passive radar tracking and automatic target recognition," in *Proc. IEEE Aerospace Conf.'02*, Vol. 4, 2002, pp. 1789–1808.
19. N. Gordon, S. Maskell, and T. Kirubarajan, "Efficient particle filters for joint tracking and classification," in *SPIE: Signal and Data Processing of Small Targets 2002*, Vol. 4728, 2002, pp. 439–449.
20. G. Storvik, "Particle filters for state-space models with the presence of unknown static parameters," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 281–289, Feb. 2002.
21. J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon (Eds.), Springer-Verlag, 2001.
22. D. Crisan and A. Doucet, "Convergence of sequential Monte Carlo methods," CUED-F-INFENG, Technical Report 381, 2000.
23. M. West, "Approximating posterior distributions by mixtures," *J. R. Statist. Soc. Ser. B*, vol. 55, pp. 409–409, 1993.
24. C. Musso, N. Oudjane, and F. LeGland, "Improving regularised particle filters," *Sequential Monte Carlo Methods in Practice*, pp. 247–271, 2001.
25. D. Guo, X. Wang, and R. Chen, "New sequential Monte Carlo methods for nonlinear dynamic systems," *Statist. Comput.*, vol. 15, no. 2, pp. 135–147, 2005.
26. D. Guo and X. Wang, "Dynamic sensor collaboration via sequential Monte Carlo," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, pp. 1037–1047, 2004.
27. H. Sidenbladh and S. Wirkander, "Particle filtering for random sets," *IEEE Trans. Aerosp. Electron. Syst.*, submitted for publication.
28. A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.
29. S. J. Julier and J. K. Uhlmann, "The scaled unscented transformation," in *Proc. IEEE American Control Conf.'02*, May 2002, pp. 4555–4559.
30. T. Cover and J. Thomas, *Elements of Information Theory*, Wiley, New York, 1990.
31. A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, New York, 2002.



---

## CHAPTER 15

---

# Energy-Efficient Decentralized Estimation

Jin-Jun Xiao<sup>1</sup>, Shuguang Cui<sup>2</sup>, and Zhi-Quan Luo<sup>1</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, Minnesota

<sup>2</sup>Department of Electrical and Computer Engineering, Texas A&M University, College Station, Texas

### 15.1 INTRODUCTION

A major design challenge in wireless sensor networks (WSNs) is the hard energy constraint. Since sensors have only small-size batteries whose replacement is costly if not impossible, sensor network operations must be energy efficient in order to prolong network lifespan [1]. In fact, a main objective of current sensor network research is to design energy-efficient devices and algorithms to support various aspects of network operations. The  $\mu$ AMPs project [2] at MIT and the PicoRadio project [3] at Berkeley focus on energy-constrained radios and their impact on ultra-low-power sensing and networking. Various energy-efficient algorithms have been proposed for network coverage [4], data gathering [5], and protocols of medium access control [6] and routing [7] (see also the survey work [8] and the references therein). These research activities focus on collaborative strategies and cross-layer designs for distributed data collection, processing, and communication in an energy-efficient manner.

In this chapter we focus on energy-efficient distributed estimation in WSNs. The distributed estimation scheme usually consists of three elements: local data compression, wireless communication, and final data fusion. Since data collected by sensors are distributed at different geographical locations, estimation in a WSN requires not only local information processing but also networkwide intelligent fusion. The latter adds a wireless communication and networking aspect to the problem that is absent from the traditional centralized estimation framework. In fact, a major challenge in WSN research is the integrated design of local signal processing operations and strategies for intersensor communication and networking, so as to strike a desirable trade-off among energy efficiency, simplicity, and overall system performance.

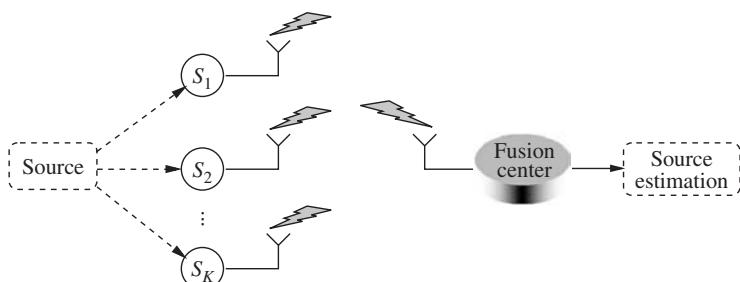
There has been a long history of research activities in distributed estimation under various network models. This problem has first been studied in the context of distributed control [9, 10] and tracking [11], later in data fusion [12, 13], and most recently in wireless sensor networks [14–17]. In the case that sensors only transmit

digitized discrete signals, the work of [16, 18, 19] addressed various design and implementation issues using the knowledge of joint distribution of sensor data. Recently, the analog data fusion schemes through a multiple-access channel have been well studied in the literature. From an information-theoretic perspective [14, 20–23] investigate the mean-squared error (MSE) performance versus transmit power for the quadratic CEO problem. Where the sensor measurements are not continuous but in a finite alphabet, type-based transmission schemes have been proposed in [14, 15, 24, 25].

Most of the existing work in distributed estimation assumes that the channels between the sensors and the fusion center are perfect (i.e., all messages are received by the fusion center without any distortion). The quantized bits from local sensors are assumed to be reliably transmitted to the fusion center for final data fusion. However, due to power limitations, fading, and channel noise, the signal sent by each individual sensor to the fusion center may be corrupted. Therefore, practical decentralized estimation schemes should take wireless channel distortion into consideration. Since communication is costly, as is the case in WSNs, there can be a significant power-saving advantage if less information needs to be transmitted and, on top of that, smart power scheduling strategies are desired to make efficient use of the network energy resource. Since wireless channel capacities are limited by two factors—bandwidth and power—we pose the following question: Given a fixed bandwidth and power budget, how should we encode the local messages, transmit the signal, and define the final fusion rule in order to maximize the overall system performance? A major part of this chapter is devoted to seeking the answer to this question.

Specifically, we study the joint estimation of one or multiple sources based on distributed observations from local sensors (c.f. Fig. 15.1). We incorporate channel distortions and communication errors in the problem formulation and investigate joint estimation strategies under energy constraints. One of the main goals is to minimize the sensor network power consumption while ensuring a desired estimation performance. In particular, we analyze the effect of both sensor coding strategies and wireless multiple-access channels (between sensors and the fusion center). It turns out that they both have a significant impact on the system design and performance:

- *Analog versus Digital* There are two main options for transmitting observations from local sensors to the fusion center: analog or digital. For the analog approach, the observed signal is transmitted via analog modulation to the fusion center. The most common analog approach is the so-called amplify-and-forward model [21, 26]. In the digital approach, the observed signal is digitized into bits, possibly



**Figure 15.1** Abstracted sensor network model.

compressed and/or encoded, then digitally modulated and transmitted. The digital approach is natural from a digital communication point of view and was adopted widely in previous studies [17, 27–31]. We will study both digital and analog approaches and compare their fundamental performance limits from an information-theoretic point of view.

- *Multiple-Access (MAC) Protocols* Between sensors and the fusion center, different MAC protocols may greatly affect the estimation performance. We consider two access schemes: orthogonal MAC and coherent MAC. Examples of the former include frequency division multiple access (FDMA) or time division multiple access (TDMA). In coherent MAC, we assume that signals transmitted from all sensors reach the receiver with phase coherency and are naturally summed at the fusion center. We will study the design of distributed estimation algorithms with both of the above two mentioned MAC protocols and analyze how a particular MAC affects the performance and computational complexity.

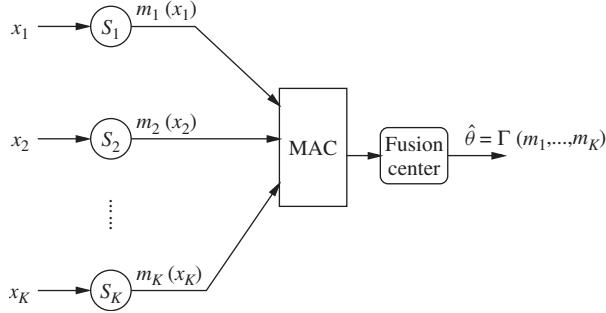
The remaining part of this chapter is organized as follows. Section 15.2 gives the problem formulation of the joint estimation of unknown sources through a sensor network with a fusion center. Section 15.3 discusses the *digital approach*. In this approach, sensor observations are uniformly quantized, sensors adopt orthogonal MAC channels to the fusion center, and each sensor applies an uncoded multilevel quadrature amplitude modulation (MQAM) transmission strategy for the quantized bits. We optimize the quantization levels and transmit power levels at local sensors to minimize the total transmit power while ensuring a given source reconstruction performance. In Section 15.4, we consider an *analog approach* in which sensor observations are amplified and forwarded to the fusion center. Both orthogonal and coherent MAC protocols are considered. For the case of orthogonal MAC, we introduce the concept of estimation diversity to analyze the system outage performance and explore the diversity gain over fading channels. The impact of MAC on the overall power and distortion performance is also discussed. In Section 15.5, we compare the performance between analog and digital approaches from an information-theoretic perspective. In Section 15.6, we study the analog approach of estimating multiple sources where we design optimal linear encoding and decoding matrices subject to energy and communication bandwidth constraints. Our main focus is the complexity of designing such optimal coding matrices. Section 15.7 gives concluding remarks.

## 15.2 SYSTEM MODEL

Consider the sensor network model illustrated in Figure 15.1, where a set of  $K$  distributed sensors make observations on an unknown source which is modeled by  $\theta$ . The observations are corrupted by additive noises and are described by

$$\mathbf{x}_i = \theta + n_i, \quad i = 1, 2, \dots, K. \quad (15.1)$$

We assume that source  $\theta$  has zero mean and variance  $\sigma_\theta^2$ , noise  $n_i$  is zero mean, spatially uncorrelated, and with variance  $\sigma_i^2$ . The data model (15.1) bears a number of variations in different applications. For example, by a suitable linear scaling, data model (15.1) is equivalent to the one where sensors observe  $\theta$  with different attenuations,



**Figure 15.2** Decentralized estimation scheme.

namely,  $x_i = h_i\theta + n_i$ . Indeed, if we let  $x'_i = x_i/h_i$  and  $n'_i = n_i/h_i$ , then  $x'_i = \theta + n'_i$ , which is identical to (15.1). We also consider the case of multiple sources and vector observations, for which the model is given by

$$\mathbf{x}_i = \mathbf{H}_i \mathbf{s} + \mathbf{n}_i, \quad (15.2)$$

where  $\mathbf{s} = [\theta_1, \theta_2, \dots, \theta_p]^T$  with observation matrices  $\mathbf{H}_i \in \mathbb{R}^{\ell_i \times p} : 1 \leq i \leq K$ .

Let us first focus on the single-source case. Suppose the sensors and the fusion center wish to jointly estimate  $\theta$  based on the spatially distributed observations  $\{x_i : i = 1, 2, \dots, K\}$ . This can be accomplished as follows (see Fig. 15.2). First, each sensor computes a local message  $m_i(x_i)$  based on its observation  $x_i$ . Second, all the sensor messages are transmitted to the fusion center, where they are combined to produce a final estimate of  $\theta$  via a real-valued fusion function  $\Gamma$ :

$$\hat{\theta} = \Gamma(\hat{m}_1(x_1), \hat{m}_2(x_2), \dots, \hat{m}(x)),$$

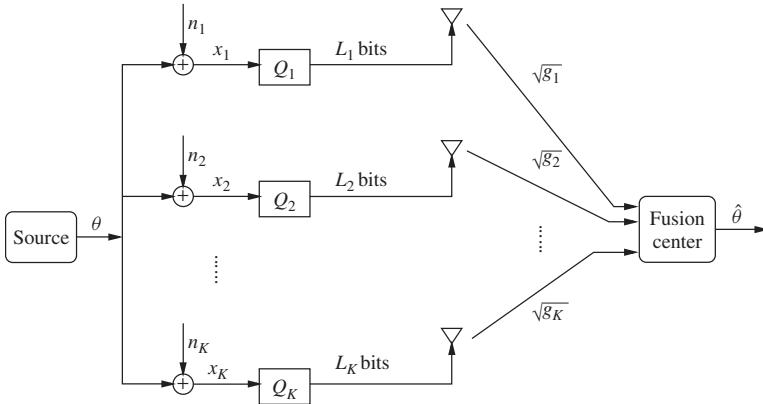
where  $\hat{m}_i$  is the received message at the fusion center when  $m_i$  is transmitted. We refer  $\{\Gamma, m_i : i = 1, 2, \dots, K\}$  as a *decentralized estimation scheme*. The problem of decentralized estimation is then to design the local message functions  $\{m_i : i = 1, 2, \dots, K\}$  and the fusion function  $\Gamma$  so that  $\hat{\theta}$  is as close to  $\theta$  as possible in a statistical sense.

In the remaining sections, we consider different schemes for the design of decentralized estimation schemes. We explore both digital and analog communication strategies with orthogonal and coherent MAC channels between sensors and the fusion center.

### 15.3 DIGITAL APPROACHES

In this section, we consider the decentralized estimation (illustrated in Fig. 15.1) within the framework of digital approaches coupled with orthogonal multiple-access channels. Recall the estimation problem given in (15.1). A digital decentralized estimation scheme can be described as follows (see Fig. 15.3):

- First, each sensor performs a local quantization of  $x_i$  and generates a discrete message  $m_i(x_i, L_i)$  of  $L_i$  bits, where the quantization rule  $Q_i : x_i \mapsto m_i(x_i, L_i)$  is to be designed.



**Figure 15.3** Decentralized quantization–communication–estimation.

- Each discrete message is then transmitted to the fusion center through orthogonal MACs which are modeled as additive white Gaussian noise (AWGN) channels with some known path gain, and the fusion center generates the final estimate  $\hat{\theta}$  based on the received signals.
- The fusion center finally combines the received signals to estimate  $\theta$  with a linear fusion rule with weights decided by the second-order statistics of the sensor observation noises and channel distortions.

The main purpose of this section is to investigate the adaptive quantization of sensor observations and its impact on energy saving. To minimize the total energy consumption under a given distortion constraint, we optimally choose the quantization levels and transmit power for each sensor by taking into account both their local signal-to-noise ratios (SNRs) and individual channel path gains. This approach is based on the universal decentralized estimation scheme [32, 33] with the energy models for the coded and uncoded MQAM transmissions [34–36] with the aim of minimizing the total sensor transmission power. The estimator at the fusion center is a generalized version of the linear minimum mean-squared estimator (LMMSE) which weighs the message functions linearly with weights determined by the observation and quantization noise.

### 15.3.1 Randomized Quantization

Suppose  $[-W, W]$  is the signal magnitude range that sensors can observe, that is,  $x = \theta + n \in [-W, W]$ , where  $W$  is a known parameter decided by the sensor dynamic range, and noise  $n$  has zero mean and variance  $\sigma^2$ . Suppose we want to quantize  $x$  into  $L$  bits regardless of the probability distribution of  $x$ . This can be achieved by uniformly dividing  $[-W, W]$  into intervals of length  $\Delta = 2W/(2^L - 1)$ , and rounding  $x$  to the neighboring endpoints of these small intervals in a probabilistic manner (see Fig. 15.4). More specifically, suppose  $l\Delta \leq x < (l + 1)\Delta$ , where  $-2^{L-1} \leq l \leq 2^{L-1}$ ; then  $x$  is quantized to  $m(x, L)$  according to

$$\mathbb{P}\{m(x, L) = l\Delta\} = 1 - r,$$

$$\mathbb{P}\{m(x, L) = (l + 1)\Delta\} = r,$$



**Figure 15.4** Probabilistic uniform quantization scheme.

with  $r = \text{def}(x - l\Delta)/\Delta \in [0, 1]$ . Notice that  $r$  is chosen in a particular way such that the quantized  $m(x, L)$  is unbiased. It is easy to see that  $m$  assumes  $2^L$  discrete values  $\{l\Delta : l = -(2^{L-1} - 1), \dots, -1, 0, 1, \dots, 2^{L-1} - 1\}$ , which can be represented in  $L$  bits. The quantization noise  $v(x, L) = \text{def } m(x, L) - x$  can be viewed as a Bernoulli random variable. The next lemma, whose proof is given in [37], shows that this message function is an unbiased estimator of  $\theta$  with a variance approaching  $\sigma^2$  at an exponential rate as  $L$  increases.

**Lemma 15.1** *Let  $m(x, L)$  be an  $L$ -bit quantization of  $x \in [-W, W]$  as defined above. Then  $m$  is an unbiased estimator of  $\theta$  and*

$$E(|m(x, L) - \theta|^2) \leq \frac{W^2}{(2^L - 1)^2} + \sigma^2$$

for all  $L \geq 1$ .

The upper bound given in the above lemma includes both quantization noise power  $W^2/(2^L - 1)^2$  and boise power  $\sigma^2$ . Given the communication bandwidth constraints, let us assume that the bit budget for sensor  $i$  is  $L_i$ . Using the randomized quantization strategy described above, we obtain the quantized version of  $x_i$ , which is denoted by  $m_i(x_i, L_i)$ , with the property of  $E(m_i) = \theta$  and  $\text{Var}(m_i) \leq \delta_i^2 + \sigma_i^2$ , in which  $\delta_i^2 = \text{def } W^2/(2^{L_i} - 1)^2$ . Without considering bit errors in the transmission, to estimate  $\theta$  at the fusion center based on  $m_i$ , we can set an optimal weight for  $m_i$  as  $1/(\sigma_i^2 + \delta_i^2)$ , giving rise to the following estimator:

$$\hat{\theta} = \left( \frac{1}{\sigma_\theta^2} + \sum_{i=1} \frac{1}{\sigma_i^2 + \delta_i^2} \right)^{-1} \sum_{i=1} \frac{m_i}{\sigma_i^2 + \delta_i^2}. \quad (15.3)$$

Notice that  $\hat{\theta}$  has an MSE upper bounded by  $[1/\sigma_\theta^2 + \sum_{i=1} 1/(\sigma_i^2 + \delta_i^2)]^{-1}$ .

### 15.3.2 Power Scheduling

To model the energy that is needed for sensor  $i$  to transmit  $L_i$  bits to the fusion center, we assume that the channel between each sensor and the fusion center is corrupted with AWGN whose double-sided power spectrum density is given by  $N_0/2$ . In addition, the channel between sensor  $i$  and the fusion center experiences a power gain proportional to  $g_i = d_i^{-\alpha}$ , where  $d_i$  is the transmission distance and  $\alpha$  is the path loss exponent. If sensor  $i$  sends  $L_i$  bits with quadrature amplitude modulation (QAM) with constellation size  $2^{L_i}$  at a bit error probability  $p_b^i$ , the total amount of required transmission energy [34, 35] is given by

$$E_i = \frac{c_i}{g_i} \left( \ln \frac{2}{p_b^i} \right) (2^{L_i} - 1), \quad (15.4)$$

where  $c_i$  is a constant determined by  $N_0$  and other system parameters [34, 35]. In addition, we assume that sensor  $i$  samples the observed signal at rate  $B_s$  and quantizes each sample to  $L_i$  bits. The transmission symbol rate is equal to the sampling rate  $B_s$ . We take the QAM constellation size to be  $2^{L_i}$  in order to minimize the delay. As such, the average transmit power consumption of node  $i$  is  $P_i = B_s E_i$ .

We next consider the optimal bit and power allocation among sensors. This can be naturally formulated as an energy minimization problem by choosing optimal  $L_i$ 's for sensors subject to certain performance constraints. For technical convenience, we propose to minimize the  $L^2$ -norm of the sensor power vector  $\mathbf{P} = \text{def}(P_1, P_2, \dots, P_K)$  and obtain the following formulation of the power-scheduling problem [37]:

$$\min \quad \|\mathbf{P}\|_2 \quad \text{s. t.} \quad D \leq D_0, \quad (15.5)$$

where  $D$  and  $D_0$  are the achieved and target MSE, respectively. To ensure that  $D \leq D_0$ , we analyze the effect of bit error rate (BER)  $p_b^i$  on the performance of data fusion in (15.3). This is given in the following remark, whose proof can be found in [37].

**Remark 15.1 (MSE due to BER)** Suppose the probability of bit error for the transmission of sensor  $i$  is  $p_b^i$  and  $\hat{m}_i$  is the decoded version of  $m_i$  at the receiver. Let  $D$  denote the MSE achieved by the estimator (15.3) based on the received messages  $\{\hat{m}_1, \hat{m}_2, \dots, \hat{m}_K\}$ . For any

$$p_0 \geq \frac{8W}{\sigma_i} \sqrt{\frac{K p_b^i}{3}} \quad \forall i,$$

it holds that

$$D \leq (1 + p_0)^2 \left( \frac{1}{\sigma_\theta^2} + \sum_{i=1}^K \frac{1}{\sigma_i^2 + \delta_i^2} \right)^{-1}.$$

Remark 15.1 shows that the actual achieved MSE is at most a constant factor away from what is achievable with perfect sensor channels provided that each sensor's BER is bounded above. We next write the optimization problem (15.5) in terms of  $L_i$ 's, which are the design variables. Assume that the constants  $c_i$  and  $B_s$  are the same for all nodes and the same target BER is chosen for all sensors. We can recast (15.5) into the following problem:

$$\begin{aligned} \min_{L_i \in \mathbb{Z}} \quad & \sum_{i=1}^K \frac{(2^{L_i} - 1)^2}{g_i^2} \\ \text{s. t.} \quad & (1 + p_0)^2 \left( \frac{1}{\sigma_\theta^2} + \sum_{i=1}^K \frac{1}{\sigma_i^2 + \delta_i^2} \right)^{-1} \leq D_0, \\ & \delta_i^2 = \frac{W^2}{(2^{L_i} - 1)^2}, \quad i = 1, \dots, K, \\ & L_i \geq 0, \quad i = 1, \dots, K. \end{aligned} \quad (15.6)$$

The detailed solution of the above problem is given in [37]. If we relax the integer constraint  $L_i \in \mathbb{Z}$  and allow  $L_i$  to take real values, the solution to the optimization problem in (15.6) has a water-filling phenomena:

$$L_i^{\text{opt}} = \begin{cases} 0 & \text{for } g_i \leq \lambda^{-1}, \\ \log\left(1 + \frac{W}{\sigma_i} \sqrt{g_i \lambda - 1}\right) & \text{for } g_i > \lambda^{-1}, \end{cases} \quad (15.7)$$

where  $\lambda$  is a universal constant decided jointly by the target MSE, sensor noise levels, and channel gains. The transmit power for node  $i$  is thus

$$P_i = c_i B_s \left( \ln \frac{2}{p_b} \right) \frac{W}{g_i \sigma_i} \sqrt{(g_i \lambda - 1)^+}.$$

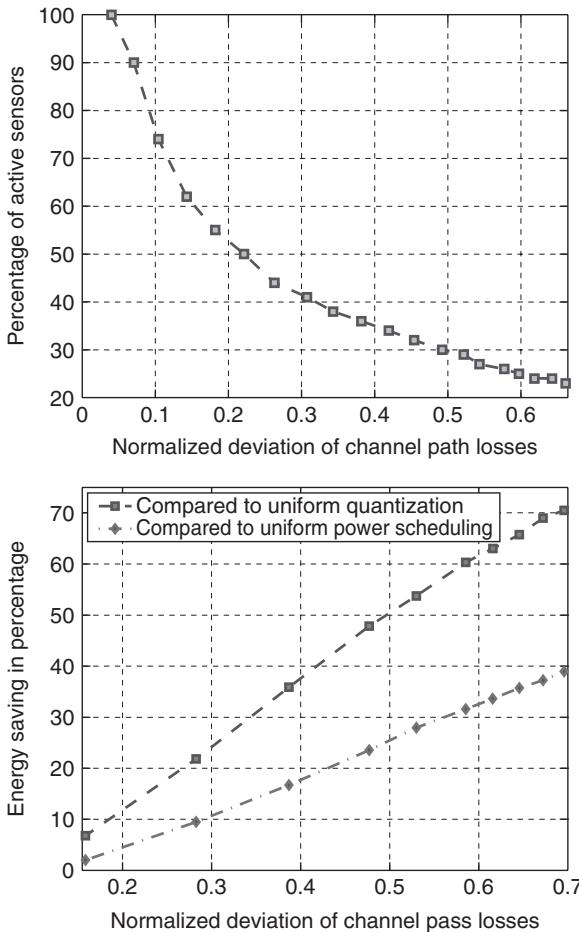
The message length formula in (15.7) is intuitively appealing as it indicates that the message length is roughly proportional to the logarithm of local SNR (i.e.,  $W^2/\sigma_i^2$ ). This is in the same spirit as the message length formula when the channel is distortionless (see [38]). Notice that when  $g_i \lambda \leq 1$ , or  $g_i \leq \lambda^{-1}$ , we have  $L_i = 0$  and therefore  $P_i = 0$ . Since  $g_i$  is the channel gain, this implies that when the channel quality for sensor  $i$  is worse than the threshold  $\lambda$  we should discard its observation in order to save energy. To implement the described power-scheduling scheme, the fusion center needs to broadcast the threshold  $\lambda$  whose value is based on the collected network information. Local sensors then decide the quantization message length  $L_i$  according to its own local information ( $\sigma_i$  and  $a_i$ ) and  $\lambda$  [c.f. (15.7)]. Two simulation results are shown in Figure 15.5, where the left plot shows that when the WSN becomes more heterogeneous, a large number of sensors with bad channel qualities or poor observations shut off, and the right plot shows a large amount of energy savings when compared to the uniform quantization strategy or the uniform power-scheduling strategy.

One of the extensions of the work above is to consider a general MAC protocol from local sensors to the fusion center. In fact, in Section 15.5 we are going to compare the performance limits of the digital approaches with another analog approach under both orthogonal and coherent MAC protocols. Our result therein shows that in the case of orthogonal MAC the digital approach obtains the most optimal trade-off between sensor power consumption and overall source estimation performance, while for the coherent MAC the digital approach is outperformed by an analog forward-and-amplify approach.

## 15.4 ANALOG APPROACHES

In this section, we consider an analog approach where sensors adopt an amplify-and-forward scheme, mainly motivated by the results derived in [21]. At each sensor  $i$ , instead of performing quantization, sensor observation  $x_i$  is directly scaled by an amplifying factor  $\sqrt{\alpha_i}$ . As such, the transmitted message is  $m_i = \sqrt{\alpha_i} x_i$ , which implies an average transmit power:

$$P_i = \alpha_i (\sigma_\theta^2 + \sigma_i^2). \quad (15.8)$$



**Figure 15.5** Number of active sensors versus channel homogeneity (left); power savings compared to uniform power scheduling or uniform quantization (right).

Depending on the MAC protocol from local sensors to the fusion center, we have the following two cases:

- *Orthogonal MAC* We assume that  $K$  sensors transmit their observations to the fusion center via  $K$  orthogonal channels, where different channels experience independent fading factors. Specifically, for channel  $i$ , we assume i.i.d. (over  $t$ ) block fading with the channel power gain denoted as  $g_i$  and i.i.d. (over time) noise denoted as  $n_{ci}$  of variance  $\sigma_{ci}^2$ . Without loss of generality, we can assume  $\sigma_{ci}^2 = \sigma_c^2$  for all  $i = 1, 2, \dots, K$ . We also assume pair-wise synchronization between each sensor and the fusion center, where synchronization among sensor nodes is not required. Under these assumptions, the received signal at the fusion center can be expressed as (see Fig. 15.6)

$$y_i = \sqrt{g_i} m_i + n_{ci}. \quad (15.9)$$

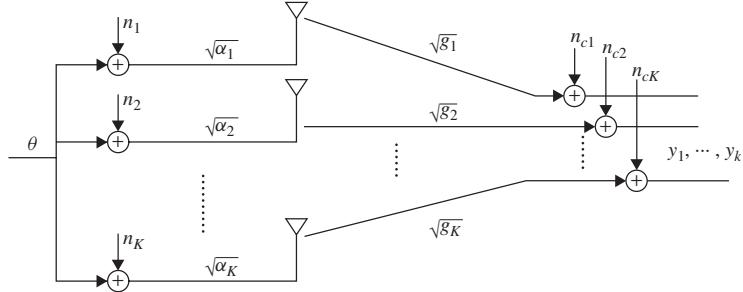


Figure 15.6 Amplify and forward with orthogonal MAC.

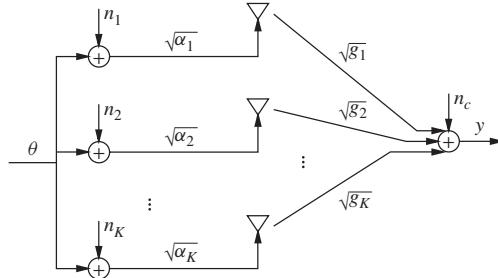


Figure 15.7 Amplify and forward with coherent MAC.

- *Coherent MAC* Another case is that all  $K$  sensors transmit simultaneously. Assuming perfect synchronization among all the sensors and the fusion center, we ensure that the transmitted signals from all sensors reach the fusion center as a coherent sum (without phase mismatch).<sup>1</sup> Of course, this leads to a challenging synchronization issue, especially when the network size is large. In this case, the received signal  $y$  at the fusion center can be expressed as (see Fig. 15.7)

$$y = \sum_{i=1}^K \sqrt{g_i} m_i + n_c, \quad (15.10)$$

in which we have assumed that the power gain of the channel from sensor  $i$  to the fusion center is  $g_i$  and channel noise  $n_c$  has zero mean and variance  $\sigma_c^2$ .

#### 15.4.1 Optimal Power Allocation

For each channel realization, we seek the optimal power allocation scheme to minimize the total power consumption while satisfying a certain distortion requirement. Specifically, for a target distortion  $D_0$ , we have the following optimization problem:

$$\begin{aligned} \min & \quad \sum_{i=1}^K P_i \\ \text{s. t.} & \quad \text{Var}[\hat{\theta}] \leq D_0. \end{aligned} \quad (15.11)$$

<sup>1</sup>Note that for orthogonal MAC we only need to assume pairwise synchronization between each sensor and the fusion center and synchronization among sensor nodes is not required.

We will study the above power allocation problem for both orthogonal and coherent MAC and also compare their performance:

- *Orthogonal MAC* It is easy to calculate that the MSE of the LMMSE of  $\theta$  based on  $y_i$  in (15.9) is given as [39]

$$\begin{aligned} \frac{1}{\text{Var}[\hat{\theta}]} &= \frac{1}{\sigma_\theta^2} + \sum_{i=1} \frac{\alpha_i g_i}{\sigma_i^2 \alpha_i g_i + \sigma_c^2} \\ &= \frac{1}{\sigma_\theta^2} \left( 1 + \sum_{i=1} \frac{\gamma_i}{1 + [(1 + \gamma_i) \sigma_c^2] / (g_i P_i)} \right), \end{aligned} \quad (15.12)$$

where  $\gamma_i = \sigma_\theta^2 / \sigma_i^2$  is the local sensor observation SNR. Plugging it and (15.8) into (15.11), we obtain the optimal scaling factor

$$\alpha_i^{\text{opt}} = \begin{cases} 0 & \text{for } g_i \leq \lambda^{-1}, \\ \frac{\gamma_i \sigma_c^2}{g_i} \left( \sqrt{\eta_i^{-1} \lambda} - 1 \right) & \text{for } g_i > \lambda^{-1}, \end{cases} \quad (15.13)$$

where  $\eta_i = g_i / (1 + \gamma_i^{-1})$  and  $\lambda$  is a universal constant decided jointly by the target MSE, sensor noise levels, and channel gains (see [40] for more details).

Similar to the result in Section 3.2, we see that the optimal strategy for minimum power transmission is to allocate more transmit power to sensors with higher channel gain and better observation quality, where the figure of merit is  $\eta_i = g_i / (1 + \gamma_i^{-1})$ . If a sensor has low  $\eta_i$ , it should be completely turned off to save power. As for numerical results, Figure 15.8a shows the average sum power consumption over different distortion target values. We see that the more strict distortion requirement we have (smaller  $D_0$ ), the more power we can save by deploying the optimal power allocation strategies, which is critical in energy-constrained sensor networks. In of Figure 15.8b, we plot the percentage of active sensors versus the total transmission power. We note that the number of active sensors is less when the total power budget is tight.

- *Coherent MAC* For the case of coherent MAC, we obtain the MSE of estimating  $\theta$  from  $y$  in (15.10) satisfying

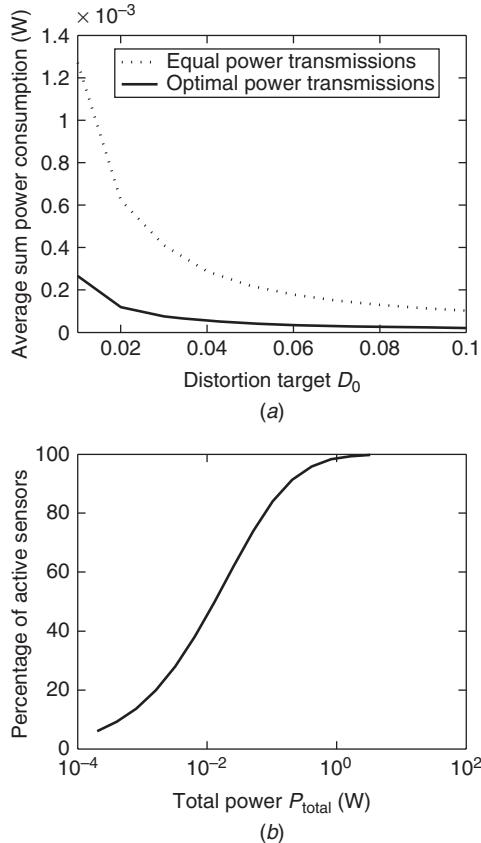
$$\begin{aligned} \text{Var}[\hat{\theta}]^{-1} &= \sigma_\theta^{-2} \\ &+ \left( \sum_{i=1} \alpha_i g_i \sigma_i^2 + \sigma_c^2 \right)^{-1} \left( \sum_{i=1} \sqrt{\alpha_i g_i} \right)^2. \end{aligned}$$

Similarly, by plugging this formula and (15.8) into (15.11), we can solve the optimal power-scheduling as follows:

$$P_i^{\text{opt}} = c_i P_{\text{tot}}, \quad i = 1, 2, \dots, K, \quad (15.14)$$

where  $P_{\text{tot}}$  is the total power budget for all sensors and

$$c_i = c \frac{g_i \gamma_i (\gamma_i + 1)}{(\gamma_i + 1 + g_i P)^2} \quad \text{with} \quad c = \left( \sum_{i=1} \frac{g_i \gamma_i (\gamma_i + 1)}{(\gamma_i + 1 + g_i P)^2} \right)^{-1}.$$



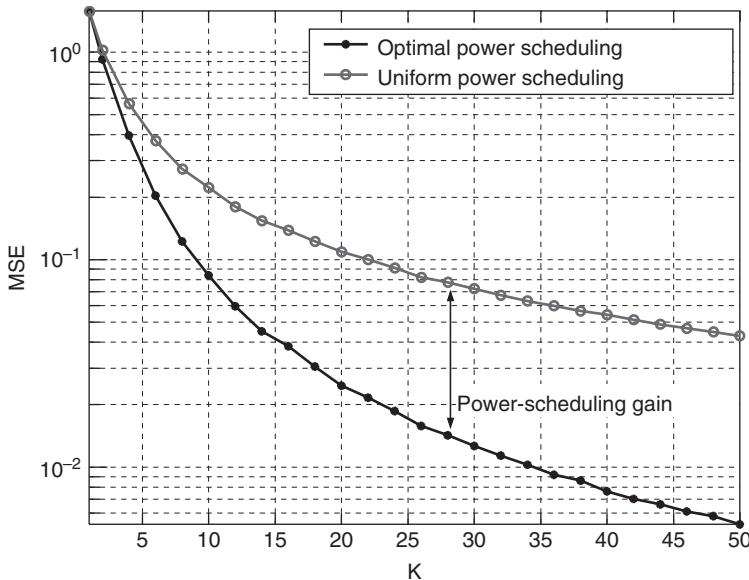
**Figure 15.8** (a) Average sum power versus distortion requirements; (b) percentage of active sensors versus total power.

With the above optimal power scheduling, the achieved MSE  $D$  in particular is given as

$$\frac{1}{D} = \frac{1}{\sigma_\theta^2} \left( 1 + \sum_{i=1} \frac{\gamma_i}{1 + [(1 + \gamma_i)/g_i](\sigma_c^2/P_{\text{tot}})} \right). \quad (15.15)$$

As a numerical example, Figure 15.9 plots the curves comparing the achieved MSE over  $K$  while the total power is kept as a constant such that  $P_{\text{tot}}/\sigma_c^2 = 20$  dB. As can be seen, the MSE performance gain of the optimal power scheduling becomes more significant as the total number of sensors  $K$  increases.

- *Performance Comparison* We next compare the MSE performance between orthogonal MAC and coherent MAC. It is interesting to see that (15.15) and (15.12) are almost identical except that, in each term of the right-hand-side sum, one is  $P_{\text{tot}}$  and the other is  $P_i$ . This reveals that the coherent MAC with the optimal power scheduling has the same MSE performance as that of the orthogonal MAC in which each sensor transmits power  $P_{\text{tot}}$ . Such a fact leads to a significant difference for the asymptotic performance (in terms of  $K$ ) between these two access schemes, which is discussed as follows.



**Figure 15.9** Performance comparison of achieved MSE for coherent MAC.

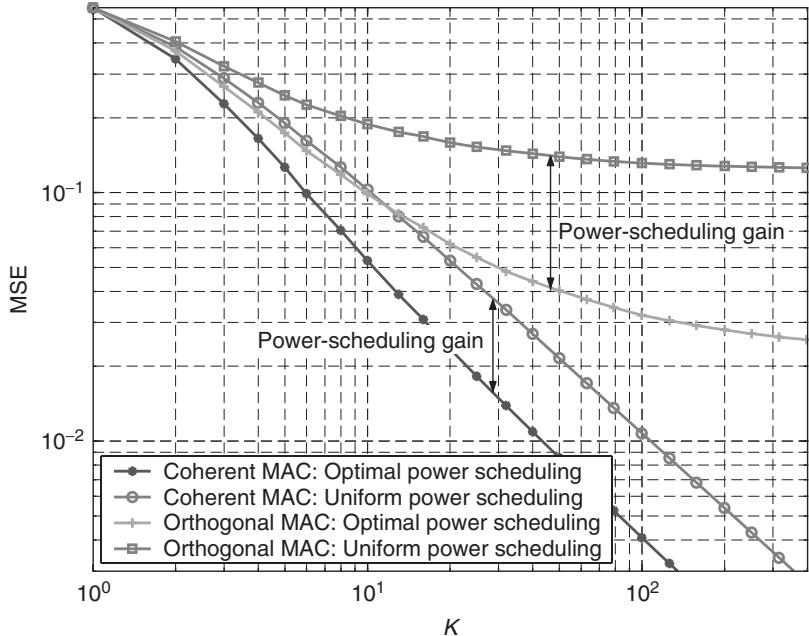
Let  $P_{\text{tot}}$  be fixed but  $K \rightarrow \infty$ . For the orthogonal MAC with uniform power allocation  $P_i = P_{\text{tot}}/K$ , under the assumption that  $\{g_i, \sigma_i^2 : 1 \leq i \leq K\}$  are i.i.d. over  $i$ , it follows from (15.12) that

$$\frac{1}{D} \rightarrow 1 + E \left[ \frac{g_i \gamma_i P}{1 + \gamma_i} \right].$$

This implies that for orthogonal MAC the achievable MSE is bounded from below even though  $K$  goes to infinity. This is in contrast to the asymptotic performance with the coherent MAC given in (15.15), which improves as  $1/K$  [c.f. (15.15)]. The performance of the orthogonal MAC is a consequence of using orthogonal links from the sensors to the fusion center, which leads to  $K$  different channel noises (i.e.,  $n_{ci} : 1 \leq i \leq K$ ) such that the corruption of channel noise cannot be eliminated even when  $K$  goes to infinity. However, in the coherent MAC, only one channel noise (i.e.,  $n_c$ ) is incurred per transmission. Thus, as a result of the coherent combination, the SNR at the received message scales up with  $K$  due to the correlation among transmitted messages, even when the total transmit power is finite. A numerical example illustrating the above results is given in Figure 15.10.

### 15.4.2 Estimation Diversity

Given the proposed joint estimation system, we are interested in investigating how the overall distortion performance is affected by the fact that we have multiple sensors with independent fading channels. One natural question is: How will the overall distortion scale with the number of sensors in the network? In previous sections, we have focused on minimizing the MSE of source estimation. In particular, for the case of orthogonal MAC, our analysis suggests that when the total amount of transmit power  $P_{\text{tot}}$  is fixed,



**Figure 15.10** MSE performance comparison between orthogonal and coherent MACs.

even if the total number of sensors  $K$  increases to infinity, the achieved distortion at the fusion center does not decrease below a certain level (see the analysis in the last paragraph of Section 15.4.1). However, are there any benefits of having more sensors in the network if we limit the amount of total power? To answer this question, let us define the outage probability  $\mathcal{P}_{D_0}$  to model the system reliability as follows:

$$\mathcal{P}_{D_0} = \text{Prob}\{\text{Var}[\hat{\theta}] > D_0\}, \quad (15.16)$$

where  $D_0$  is a predefined threshold. For a well-designed joint estimation scheme, it is desirable that the outage performance is enhanced as we increase the number of sensors due to the diversity given by independent fading channels. The following theorem quantifies this effect.

**Theorem 15.1** Consider the analog amplify-and-forward coding schemes when sensors have orthogonal MAC to the fusion center. Suppose the sensor observation SNR  $\{\gamma_i : i = 1, 2, \dots, K\}$  and channel gain  $\{g_i : i = 1, 2, \dots, K\}$  are both i.i.d. across  $i$ . Define  $\eta_i = \text{def } g_i/(1 + \gamma_i^{-1})$ . In addition, we assume that  $E[\eta_i]$  and  $E[\gamma_i^{-1} g_i^2]$  are finite. When the total number of sensor  $K$  is large, with the total power  $P_{\text{tot}}$  and equal-power allocation among sensors, we have the outage probability  $\mathcal{P}_{D_0} \sim \exp[-K I_\eta(a)]$ , where  $\sim$  means asymptotically converging to (as  $K$  becomes large),  $\eta$  is the common distribution of  $\eta_i$ , and  $I_\eta(a)$  is the rate function of  $\eta$ :

$$I_\eta(a) = \sup_{\theta \in \mathbb{R}} (\theta a - \log M_\eta(\theta)),$$

with  $a = [(\sigma_\theta^2 - D_0)\sigma_c^2]/(D_0 P_{\text{tot}})$  and  $M_\eta(\theta)$  the moment-generating function of  $\eta$ .

The more detailed explanation of the rate function and the proof of Theorem 15.1 are given [40]. From the theorem we see that  $K$  plays the role of estimation diversity order here in that the outage probability decreases exponentially with  $K$ . We remark that the fact that the outage probability decays exponentially with the number of sensors is due to the effect of independent measurements and fading coefficients, which bears similar properties as the probability of detection error in distributed detection [41–44]. Note that even though Theorem 15.1 is an asymptotic result for large  $K$ , we later show by simulation results that the outage probability curve illustrates diversity order of  $K$  (approximately) even for small values of  $K$  in practical scenarios.

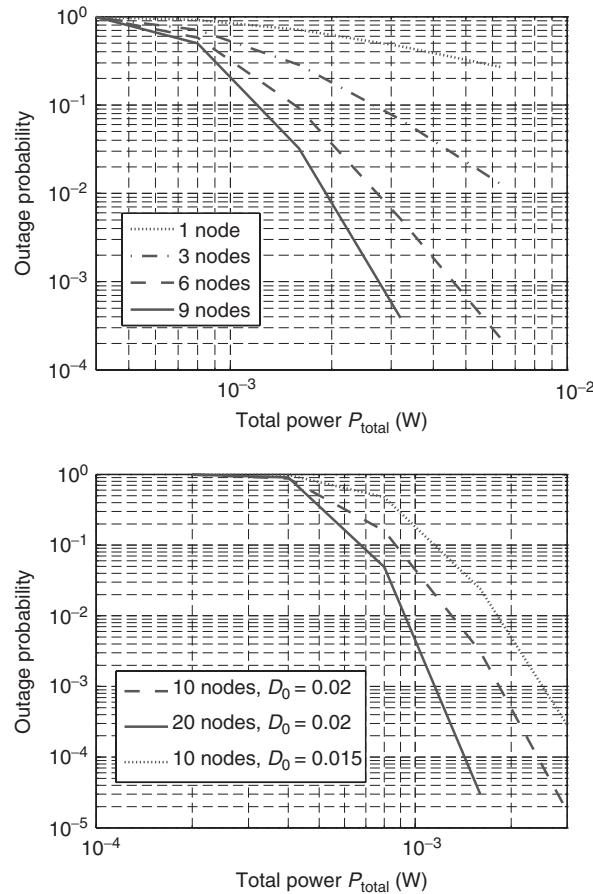
**Example 15.1** As an example, let us consider the case in which  $\gamma_i = 1$  for all  $i$ 's and  $\sqrt{g_i}$  is i.i.d. Rayleigh with probability density function (pdf)  $f(x) = (x/\xi_i^2) \exp[-x^2/(2\xi_i^2)]$ . For large  $K$  and  $P_{tot}$ , the rate function is given as

$$I_s(a) = \frac{\sigma_\theta^2 - D_0}{\xi_i^2 D_0} \frac{\sigma_c^2}{P_{tot}} - \log \frac{\sigma_\theta^2 - D_0}{\xi_i^2 D_0} \frac{\sigma_c^2}{P_{tot}} - 1 \\ \sim \log P_{tot},$$

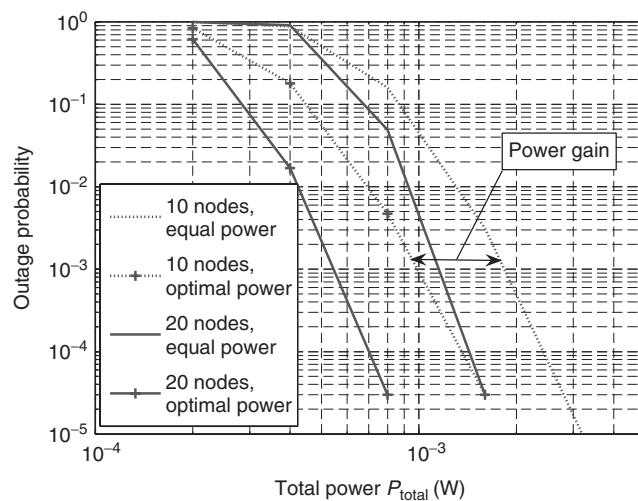
which means that the estimation convergence rate is approximately  $\log P_{tot}$ . In other words, for Rayleigh fading channels we have  $-\log P_{D_0} \sim K \log P_{tot}$ , which shows that the diversity order  $K$  is the slope of the outage probability versus power curve if things are plotted in the log-log fashion.

We now provide some numerical examples to verify the analytical results. We assume that the channels from sensors to fusion center have Rayleigh fading (see [40] for the details of simulation setup). The outage probability versus the total transmission power is plotted in Figure 15.11 (left) for different numbers of sensors, where we see that the 3-node case performs much better than the 1-node case and the 9-node case performs much better than the 3-node case. Approximately, when the logarithm of outage probability is plotted versus the logarithm of the total transmission power, the slope of the curve at the high-power region is proportional to the number of sensors, which is defined as the diversity order. Note that this definition of diversity order is based on the distortion outage performance, which is different from the traditional definition of diversity order in multiple-input multiple-output (MIMO) multiantenna systems [45], which is usually the slope of symbol error curves. However, the two definitions imply similar performance benefits from diversity. This type of diversity gain is also shown for large numbers of sensors in Figure 15.10 (right), where we see that the slope of the 20-node curve is twice that of the 10-node curve in the high-power regime. Not surprisingly, when we decrease  $D_0$  the outage probability will be increased.

In Figure 15.12, we compare the outage performance of the optimal power scheme with the case where all the sensors transmit with equal power. From the outage probability curves, we see that for the same number of sensor nodes the curve slopes are almost the same for both the equal-power and the optimal-power cases, which means that the optimal-power transmission strategy achieves the same diversity order of  $K$  as the uniform power strategy. In addition, the curve for the optimal power case is a left-shifted version of that for the equal-power case. This shift is the result of an adaptive power gain that is due to the fact that we allocate more power to nodes with better channels and observation qualities. This gain is similar to array or coding gains in traditional MIMO systems [45].



**Figure 15.11** Outage probability versus total power.



**Figure 15.12** Diversity gain and power allocation gain.

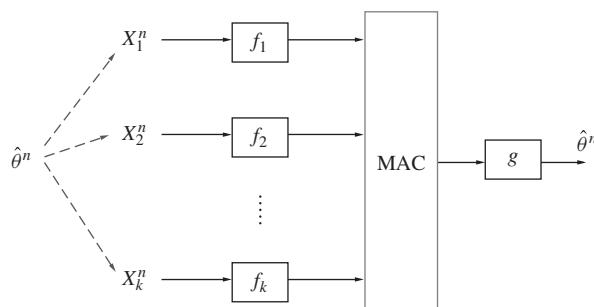
## 15.5 ANALOG VERSUS DIGITAL

For the decentralized estimation via sensor networks with a fusion center (as illustrated in Fig. 15.1), we have studied both digital and analog coding schemes in the previous two sections. In this section we compare the performance of these two coding schemes. We study this problem from an information-theoretic point of view and model the source acquisition, processing, communication, and reconstruction in sensor networks by a multiterminal joint source–channel communication system. We show the following results:

- When the MAC between sensors and the fusion center is orthogonal, the optimal coding strategy is digital, that is, the conventional separate source and channel coding can achieve the optimum [46].
- When the MAC between sensors and the fusion center is coherent, separation theorem breaks down, and it turns out that the simple amplify-and-forward strategy can easily outperform the conventional separate source and channel coding [21].

As depicted in Figure 15.13, the source parameter of interest is modeled by a discrete-time memoryless random process  $\{\theta(t) : 1 \leq t < \infty\}$ . Sensor observations are denoted by  $\{X_i(t) : i = 1, 2, \dots, K\}$ , and their joint conditional distribution (given the source  $\theta(t)$ ) is known. A general coding scheme with block length  $n$  can be described as follows. First, sensor observations  $X_i^n = \text{def}\{X_i(t) : 1 \leq t \leq n\}$  are encoded in a distributed fashion ( $f_i$  denotes the encoder of the  $i$ th sensor). Then a single decoder  $g$  decodes  $\theta^n = \text{def}\{\theta(t) : 1 \leq t \leq n\}$  based on the received information from the distributed encoders. We introduce cost constraints on the transmitted symbols from each individual sensor. Such constraints may include power constraints as a special case. The fundamental objective of the source–channel communication problem is to determine the optimal trade-off between cost and distortion in an information-theoretic sense regardless of complexity and delay.

When rate constraints are imposed on encoded messages, we obtain a source-coding problem in which the goal is to characterize the rate distortion region  $\mathcal{R}(D)$ . The latter consists of all rate-tuples  $R = (R_1, R_2, \dots, R)$  that allow for the reconstruction of the source  $\theta$  within certain distortion level  $D$  when the sensor observations are encoded at a rate not exceeding  $R_i$  at each sensor  $i$ . This is the so-called CEO problem that was first introduced in [47–49]. Except for inner and outer bounds on  $\mathcal{R}(D)$  derived



**Figure 15.13** Coding scheme in WSN with fusion center.

in [47, 49] and the quadratic Gaussian case addressed in [50], the CEO problem remains open to this date.

In several important cases, source coding and channel coding can be separated without performance loss. For example, in a point-to-point link, source coding and channel coding can be performed separately without performance degradation if the source and channel are both discrete and memoryless [(51, Theorem 21)]. This source–channel separation theorem is quite appealing from a practical standpoint since it implies that source coding can be performed without channel knowledge, and similarly channel encoding can be performed without the knowledge of the source. Unfortunately, the separation theorem does not extend to general links [(52, Chapter 14)]. An interesting counterexample can be found in [53] for lossless transmission of correlated sources through an interfering (nonorthogonal) multiple-access channel. In this case, separating source coding from channel coding is suboptimal.

However, for the sensor network in Figure 15.1, if the intersensor interference is resolved by reservation-based orthogonal protocols (e.g., TDMA or FDMA), that is, local sensors have noninterfering channels to the fusion center, it turns out that the optimal trade-off between cost and distortion can be achieved by separate source and channel coding [46]. Proving the separation theorem in this case entails a *multiple-letter characterization* of the rate distortion region  $\mathcal{R}(D)$ . By combining this multiletter representation of  $\mathcal{R}(D)$  with the MAC channel capacity  $\mathcal{C}(\Gamma)$ , we obtain the following theorem [46], which extends the results of [54, 55] for the lossless transmission of correlated sources from finite alphabets to the lossy transmission of continuously valued sources.

**Theorem 15.2** *For the multiterminal source channel communication depicted in Figure 15.13, if sensors have orthogonal MAC to the fusion center, the transmission cost and distortion pair  $(\Gamma, D)$  is achievable if and only if  $\mathcal{C}(\Gamma) \cap \mathcal{R}(D) \neq \emptyset$ .*

When the MAC between sensors and the fusion center is more general, the separation theorem in Theorem 1.5 breaks down. For the special case of estimating a Gaussian source using the MSE as distortion measure, it is shown in [21] that when each sensor has a fixed power budget and sensors have a coherent MAC to the fusion center, the distortion achieved by optimal separate source and channel coding decreases at a rate of  $1/\log K$ , while for a simple “analog” uncoded transmission, the MSE decreases as  $1/K$ .

When the MAC between sensors and the fusion center is orthogonal, since the separation theorem holds, the optimal trade-off between the total sensor power and the overall distortion can be achieved by the digital approach of separate source and channel coding. As a result, we claim that in this case the “digital” approach outperforms the “analog” uncoded transmission strategy.<sup>2</sup> On the contrary, when sensors have a coherent MAC to the fusion center, the “analog” uncoded transmission strategy significantly outperforms the digital approach with separate source and channel coding. This implies that the optimal sensor transmission strategy is related to the MAC protocol of

<sup>2</sup>In practise, we need to in addition take the coding complexity into consideration. The linear analog approach proposed for the case of orthogonal MAC in the previous section enjoys the advantages of being easy to implement and with zero delay. Although the optimal approach for the orthogonal MAC case is the digital one with separate source and channel coding, its realization may have very high complexity due to the fact that high-performance multiterminal source codes and channel codes have long block length.

the sensor network, which suggests that to obtain the optimal performance of a sensor network, cross-layer design and optimization are desired [56].

## 15.6 EXTENSION TO VECTOR MODEL

In previous sections we have been focusing on the scalar source estimation. In this section, we discuss the linear decentralized estimation of vector sources. We study the analog approaches of linearly encoding sensor observations by multiplying a matrix, which naturally performs dimensionality reduction of sensor observations. A main motivation for using an analog communication model is that it opens up the possibility of applying tools from analysis and convex optimization to the design of optimal decentralized signal processing schemes. Denote the unknown vector random signal by  $\mathbf{s} = [\theta_1, \theta_2, \dots, \theta_p]^T$ . We assume that  $\mathbf{s}$  has zero mean and covariance matrix  $\mathbf{C}_s$ . Sensor observation vectors  $\mathbf{x}_i \in \mathbb{R}^{\ell_i \times 1}$  are the linear combination of  $\mathbf{s}$  corrupted by additive noises and can be described as

$$\mathbf{x}_i = \mathbf{H}_i \mathbf{s} + \mathbf{n}_i, \quad (15.17)$$

where  $\mathbf{H}_i \in \mathbb{R}^{\ell_i \times p} : 1 \leq i \leq K$  are observation matrices;  $\mathbf{n}_i \in \mathbb{R}^{\ell_i \times 1} : 1 \leq i \leq K$  are spatially uncorrelated among different sensors and each  $\mathbf{n}_i$  has zero mean and covariance matrix  $\mathbf{C}_{\mathbf{n}_i}$  but is otherwise unknown. Without loss of generality, we can assume  $\mathbf{C}_s = \mathbf{I}_p$  and  $\mathbf{C}_{\mathbf{n}_i} = \mathbf{I}_{\ell_i}$ .<sup>3</sup>

In the linear encoding of vector observations, one immediate question is about how many real messages to which each observation  $\mathbf{x}_i$  shall be compressed. This is band limited by the degrees of freedom of the channel from sensor  $i$  to the fusion center. Assume that for each observation period sensor  $i$  can transmit  $q_i$  real messages to the fusion center, which is potentially decided by the channel bandwidth. With such an assumption, the message functions can be presented as

$$\mathbf{m}_i(\mathbf{x}_i) = \mathbf{A}_i \mathbf{x}_i, \quad \text{where } \mathbf{A}_i \in \mathbb{R}^{q_i \times \ell_i},$$

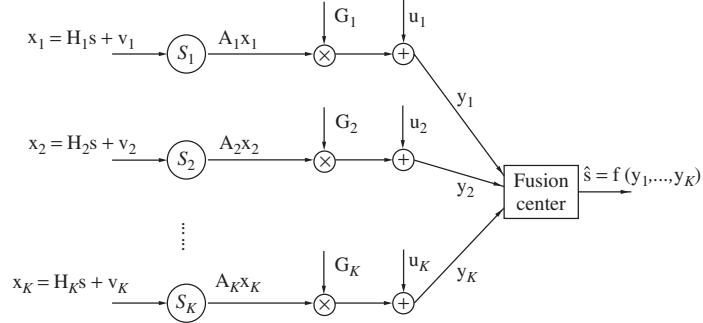
which implies that the  $\ell_i$ -dimensional  $\mathbf{x}_i$  is reduced to a  $q_i$ -dimensional message  $\mathbf{m}_i$  after encoding. Based on the received messages, the fusion center then generates an estimate  $\hat{\mathbf{s}}$  to minimize the overall MSE  $D = \text{def} \text{tr}(\mathbf{D})$ , where  $\mathbf{D} = \text{def} E[(\mathbf{s} - \hat{\mathbf{s}})(\mathbf{s} - \hat{\mathbf{s}})^T]$ . We describe the remaining part of the linear decentralized estimation for two MAC cases as follows:

- *Orthogonal MAC* In this case the  $K$  sensors transmit their observations to the fusion center via  $K$  orthogonal channels (see Fig. 15.14). For channel  $i$ , the received signal can be written as

$$\mathbf{y}_i = \mathbf{G}_i \mathbf{m}_i + \mathbf{u}_i,$$

where  $\mathbf{G}_i \in \mathbb{R}^{q_i \times q_i}$  is the channel matrix from sensor  $i$  to the fusion center and  $\mathbf{u}_i \in \mathbb{R}^{q_i \times 1}$  is the additive channel noise with covariance matrix  $\mathbf{C}_{\mathbf{u}_i}$ . Without loss of generality, we assume  $\mathbf{C}_{\mathbf{u}_i} = \mathbf{I}_{q_i}$ . It is easy to see that the linear MMSE

<sup>3</sup>Otherwise, we can introduce  $\mathbf{s}^{(1)} = \mathbf{C}_s^{-1/2} \mathbf{s}$ ,  $\mathbf{n}_i^{(1)} = \mathbf{C}_{\mathbf{n}_i}^{-1/2} \mathbf{n}_i$ ,  $\mathbf{x}_i^{(1)} = \mathbf{C}_{\mathbf{n}_i}^{-1/2} \mathbf{x}_i$  and  $\mathbf{H}_i^{(1)} = \mathbf{C}_{\mathbf{n}_i}^{-1/2} \mathbf{H}_i \mathbf{C}_s^{1/2}$ . Then we obtain an equivalent model  $\mathbf{x}_i^{(1)} = \mathbf{H}_i^{(1)} \mathbf{s}^{(1)} + \mathbf{n}_i^{(1)}$  in which  $\mathbf{C}_{\mathbf{s}^{(1)}} = \mathbf{I}_p$ ,  $\mathbf{C}_{\mathbf{v}_i^{(1)}} = \mathbf{I}_{\ell_i}$ .



**Figure 15.14** Linear decentralized estimation with orthogonal MAC.

estimator of  $s$  based on  $\{\mathbf{y}_i : 1 \leq i \leq K\}$  has an MSE matrix  $\mathbf{D}$  satisfying (see, e.g., [39, Theorem 12.1])

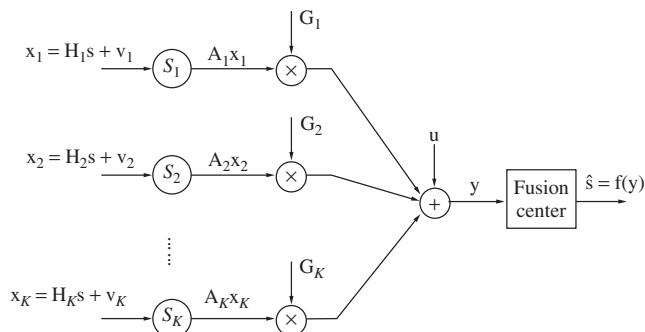
$$\begin{aligned} \mathbf{D}^{-1} &= \mathbf{I}_p \\ &+ \sum_{i=1}^K \mathbf{H}_i^T \mathbf{B}_i^T (\mathbf{B}_i \mathbf{B}_i^T + \mathbf{I}_q)^{-1} \mathbf{B}_i \mathbf{H}_i. \end{aligned} \quad (15.18)$$

where  $\mathbf{B}_i = \text{def } \mathbf{G}_i \mathbf{A}_i$ . We note that for the orthogonal MAC case the bandwidth constraint  $q_i$ 's can be different over sensors.

- *Coherent MAC* Another case is that all  $K$  sensors transmit simultaneously. The transmitted signals from all sensors reach the fusion center as a coherent sum under the assumption of perfect synchronization between sensors and the fusion center. In this case, we assume that each sensor transmits the same number of real messages, that is,  $q_i = q$  for all  $i$ 's. The received signal  $\mathbf{y}$  at the fusion center can be expressed as (see Fig. 15.15):

$$\mathbf{y} = \sum_{i=1}^K \mathbf{G}_i \mathbf{m}_i + \mathbf{u}, \quad (15.19)$$

where  $\mathbf{G}_i \in \mathbb{R}^{q \times q}$  are the channel matrix from sensor  $i$  to the fusion center and  $\mathbf{u} \in \mathbb{R}^{q \times 1}$  is the additive channel noise with covariance matrix  $\mathbf{C}_{\mathbf{u}}$ . Again, without loss



**Figure 15.15** Linear decentralized estimation with coherent MAC.

of generality, we can assume  $\mathbf{C}_u = \mathbf{I}_q$ . Given  $\mathbf{y}$ , the fusion center then generates an estimate  $\hat{\mathbf{s}}$ . Specifically, the linear MMSE estimator achieves an MSE satisfying

$$\mathbf{D}^{-1} = \mathbf{I}_p + \mathbf{H}^T \mathbf{B}^T (\mathbf{B} \mathbf{B}^T + \mathbf{I}_q)^{-1} \mathbf{B} \mathbf{H}, \quad (15.20)$$

where

$$\begin{aligned}\mathbf{H} &= [\mathbf{H}_1^T, \mathbf{H}_2^T, \dots, \mathbf{H}_K^T]^T, \\ \mathbf{B} &= [\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_K], \\ \mathbf{v} &= [\mathbf{v}_1^T, \mathbf{v}_2^T, \dots, \mathbf{v}_K^T]^T, \\ \ell &\stackrel{\text{def}}{=} \ell_1 + \ell_2 + \dots + \ell_K.\end{aligned}$$

The bandwidth constraints lead to a dimensionality condition on  $\mathbf{A}_i$ , that is,  $\mathbf{A}_i \in \mathbb{R}^{q_i \times \ell_i}$ . Suppose, in addition, that the transmit power constraint at sensor  $i$  is  $P_i$ . We thus have the following constraint on  $\mathbf{A}_i$ :

$$\begin{aligned}\text{tr} \{ E(\mathbf{m}_i \mathbf{m}_i^T) \} &= \text{tr} [\mathbf{A}_i (\mathbf{H}_i \mathbf{H}_i^T + \mathbf{I}_{\ell_i}) \mathbf{A}_i^T] \\ &\leq P_i.\end{aligned} \quad (15.21)$$

Therefore, to design the optimal linear decentralized estimation scheme, we shall solve the optimal encoding matrices  $\mathbf{A}_i : 1 \leq i \leq K$  subject to power and bandwidth constraints such that  $D = \text{tr}(\mathbf{D})$  is minimized. This leads to the following optimization problem:

$$\begin{aligned}\min_{\mathbf{A}_i: 1 \leq i \leq K} \quad & \text{tr}(\mathbf{D}) \\ \text{s.t.} \quad & \mathbf{D} \text{ satisfies (15.18) or (15.20)} \\ & \mathbf{A}_i \in \mathbb{R}^{q_i \times \ell_i}, \quad 1 \leq i \leq K \\ & \text{tr} [\mathbf{A}_i (\mathbf{H}_i \mathbf{H}_i^T + \mathbf{I}_{\ell_i}) \mathbf{A}_i^T] \leq P_i, \\ & 1 \leq i \leq K\end{aligned} \quad (15.22)$$

### 15.6.1 Orthogonal MAC

When the MAC is orthogonal, we obtain from (15.18), (15.21), and (15.22) the following optimization problem:

$$\begin{aligned}\min \quad & \text{tr} \left( \mathbf{I}_p + \sum_{i=1} \mathbf{H}_i^T \mathbf{B}_i^T (\mathbf{B}_i \mathbf{B}_i^T + \mathbf{I}_q)^{-1} \mathbf{B}_i \mathbf{H}_i \right)^{-1} \\ \text{s.t.} \quad & \text{tr} [\mathbf{A}_i (\mathbf{H}_i \mathbf{H}_i^T + \mathbf{I}_{\ell_i}) \mathbf{A}_i^T] \leq P_i, \\ & \mathbf{B}_i = \mathbf{G}_i \mathbf{A}_i, \quad 1 \leq i \leq K, \\ & \mathbf{A}_i \in \mathbb{R}^{\ell_i \times v_i}, \quad 1 \leq i \leq K.\end{aligned} \quad (15.23)$$

It is shown in [57] that (15.23) is nonconvex and NP-hard in general. More specifically, the computational complexity of solving (15.23) is NP-hard even in the case where  $\mathbf{u}_i = 0$  and  $\ell_i = 1$  for all  $i$ 's. In other words, even for the special case in which each sensor sends exactly one real-valued message to the fusion center, the problem of designing the optimal linear encoding functions is NP-hard.

### 15.6.2 Coherent MAC

In contrast to the orthogonal MAC, for the case of coherent MAC, we show in [58] that when the MAC between sensors and the fusion center is noiseless, the resulting problem has a closed-form solution, while in the noisy MAC case, the problem can be efficiently solved by semidefinite programming (SDP). More details are given as follows.

**15.6.2.1 Noiseless Channel Case** We consider the design of linear decentralized estimation by idealizing the communication link from sensors to the fusion center, or equivalently, setting  $\mathbf{C}_\mathbf{u} = \mathbf{0}$ . The main motivation here is to treat the decentralized estimation from a linear compression perspective and observe how the bandwidth alone (which limits the number of linearly encoded messages) affects the performance of the linear decentralized estimation. Accordingly we have the following problem:

$$\begin{aligned} \min_{\mathbf{B}_i} \quad & \text{tr}(\mathbf{D}) \\ \text{s.t.} \quad & \mathbf{D}^{-1} = \mathbf{I}_p + \mathbf{H}^T \mathbf{B}^T (\mathbf{B} \mathbf{B}^T)^\dagger \mathbf{B} \mathbf{H}, \\ & \mathbf{B} \in \mathbb{R}^{q \times \ell}. \end{aligned} \quad (15.24)$$

The solution has a closed form and the optimal encoding matrices  $\mathbf{A}_i^* \in \mathbb{R}^{q \times \ell_i}$  can be solved as follows [58]:

- First we solve for  $\mathbf{B}^* = \mathbf{V}_B \Lambda_B \mathbf{U}_H^T = \mathbf{V}_B \Lambda (\mathbf{U}_H^T)_{q \times \ell}$ , where  $\mathbf{V}_B$  is any unitary matrix of size  $q \times q$ ,  $\Lambda$  is any positive-definite diagonal matrix of size  $q \times q$ , and  $\mathbf{U}_H^T$  is the left eigenspace of  $\mathbf{H}$  where  $(\mathbf{U}_H^T)_{q \times \ell}$  denotes the first  $q$  rows of  $\mathbf{U}_H^T$ .
- After obtaining  $\mathbf{B}^*$ , we break  $\mathbf{B}^*$  to get each  $\mathbf{B}_i^*$  as follows:  $\mathbf{B}^* = (\mathbf{B}_1^*, \mathbf{B}_2^*, \dots, \mathbf{B}_K^*)$ , with  $\mathbf{B}_i \in \mathbb{R}^{q \times \ell_i}$ .
- The optimal encoding matrix is then  $\mathbf{A}_i^* = \mathbf{G}_i^{-1} \mathbf{B}_i^*$ :  $1 \leq i \leq K$ .

Moreover, the achieved MSE  $D = \text{tr}(\mathbf{D})$  as a function of  $q$  can be represented as

$$D(q) = \begin{cases} p - q + \sum_{i=1}^q \frac{1}{1 + \lambda_i(\mathbf{H}^T \mathbf{H})}, & q < p \\ \sum_{i=1}^p \frac{1}{1 + \lambda_i(\mathbf{H}^T \mathbf{H})}, & q \geq p \end{cases}$$

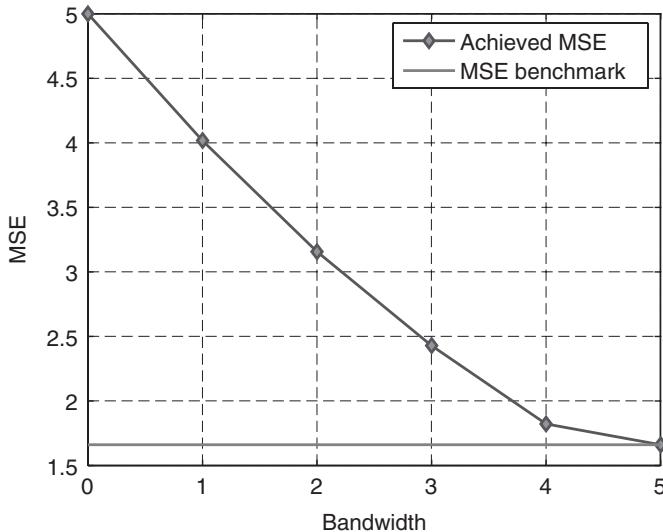
where  $\lambda_i(\mathbf{H}^T \mathbf{H}) : 1 \leq i \leq p$  are of decreasing order.

An observation is that when the bandwidth  $q$  reaches  $p$ , which is the number of components in the unknown signal  $\mathbf{s}$ , the achieved MSE with optimally designed encoding matrices obtains the centralized benchmark  $D_{\text{cen}} = \text{def} \text{tr}[(\mathbf{I}_p + \mathbf{H}^T \mathbf{H})^{-1}]$ , that is, the estimation MSE based on complete  $\mathbf{x}_i$ 's. Increasing the bandwidth  $q$  further does not improve the achievable MSE distortion performance. A numerical plot of the achieved MSE with different numbers of transmissions ( $q$ ) is given in Figure 15.16, in which we take  $p = 5$ ,  $K = 8$ , and assume that each sensor makes only one observation (i.e.,  $\ell_i = 1$ ) and each entry of  $\mathbf{H}_i$  has a complex Gaussian distribution. From the figure we see that once the bandwidth  $q$  reaches  $p = 5$ , the MSE reaches the benchmark.

**15.6.2.2 Noisy Channel Case** In this section we study the case where the channel noise is taken into consideration. Plugging (15.20) and (15.21) into the generic problem in (15.22), we can solve the optimal  $\mathbf{A}_i$  from the following problem:

$$\begin{aligned} & \min_{\mathbf{A}_i, \mathbf{B}_i : 1 \leq i \leq K} && \text{tr}(\mathbf{D}) \\ \text{s.t.} & && \mathbf{D}^{-1} = \mathbf{I}_p + \mathbf{H}^T \mathbf{B}^T (\mathbf{I}_q + \mathbf{B} \mathbf{B}^T)^{-1} \mathbf{B} \mathbf{H}, \\ & && \text{tr}[\mathbf{A}_i (\mathbf{H}_i \mathbf{H}_i^T + \mathbf{I}_{\ell_i}) \mathbf{A}_i^T] \leq P_i, \\ & && \mathbf{B}_i = \mathbf{G}_i \mathbf{A}_i, \quad 1 \leq i \leq K, \\ & && \mathbf{B} = [\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_K] \in \mathbb{R}^{q \times \ell}. \end{aligned} \quad (15.25)$$

Problem (15.25) is not convex over  $\{\mathbf{A}_i, \mathbf{B}_i : 1 \leq i \leq K\}$  due to the nonconvex property of the first constraint. To obtain an efficient solution, we can transform (15.25)



**Figure 15.16** MSE versus bandwidth  $q$  (noiseless channel case).

into the problem below by introducing auxiliary variables and applying Schur's complement [59]:

$$\begin{aligned} \min \quad & \text{tr}(\mathbf{D}) \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{I}_p + \mathbf{H}^T \mathbf{H} - \mathbf{R} & \mathbf{I}_p \\ \mathbf{I}_p & \mathbf{D} \end{bmatrix} \succeq 0, \\ & \begin{bmatrix} \mathbf{R} & \mathbf{H}^T \\ \mathbf{H} & \mathbf{I}_\ell + \mathbf{Q} \end{bmatrix} \succeq 0, \\ & \text{tr} [\mathbf{Q}_i (\mathbf{H}_i \mathbf{H}_i^T + \mathbf{I}_p)] \leq g_i P_i, \\ & 1 \leq i \leq K, \\ & \mathbf{Q} \succeq 0, \quad \text{rank}(\mathbf{Q}) = q. \end{aligned} \tag{15.26}$$

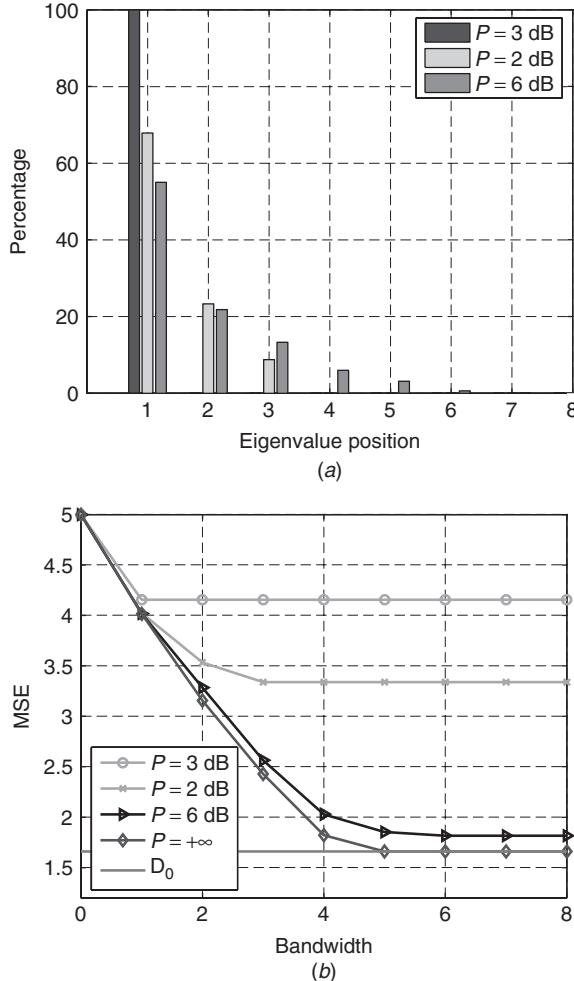
The above problem becomes an SDP [59] if we remove the last rank constraint. By solving the relaxed SDP problem, we obtain a  $\mathbf{Q}$  that may have a rank larger than  $q$ . To obtain the solution to the original problem, we perform eigendecomposition of  $\mathbf{Q}$ :  $\mathbf{Q} = \sum_{i=1}^{\ell} \lambda_i(\mathbf{Q}) \mathbf{v}_{\mathbf{Q},i} \mathbf{v}_{\mathbf{Q},i}^T$ . By taking the largest  $q$  principal eigencomponents of  $\mathbf{Q}$ , we obtain a solution for  $\mathbf{B}$  as follows:  $\mathbf{B} = [\sqrt{\lambda_1(\mathbf{Q})} \mathbf{v}_{\mathbf{Q},1}, \sqrt{\lambda_2(\mathbf{Q})} \mathbf{v}_{\mathbf{Q},2}, \dots, \sqrt{\lambda_q(\mathbf{Q})} \mathbf{v}_{\mathbf{Q},q}]$ , which is a  $q \times \ell$  matrix.

In the numerical example, we adopt the optimization toolbox: SeDuMi [60] to solve the SDP in (15.26) by relaxing the last rank constraint. We still take  $p = 5$ ,  $K = 8$ , and assume that each sensor makes only one observation (i.e.,  $\ell_i = 1$ ). We further assume that each entry of  $\mathbf{H}_i$  is complex Gaussian with unit variance and  $g_i = 1 \forall i$ . In addition, the channel noise is also assumed to have unit power. For the transmit power constraints, we take  $P_i = -3, 2, 6$  dB higher than the channel noise power, respectively. Note that the noiseless channel case we considered in Section 15.6.2.1 corresponds to an infinite transmit power in this general setup.

A numerical plot of the eigenvalue distribution of  $\mathbf{Q}^*$  solved from the SDP is given in the left plot of Figure 15.17a, in which the vertical axis represents the percentage of each eigenvalue against the total spectrum (sum of all eigenvalues) of  $\mathbf{Q}^*$ . We see that although the rank constraint of  $\mathbf{Q}$  has been relaxed, the optimal solution  $\mathbf{Q}^*$  only has the first few eigenvalues of significant contribution. The achieved MSE is given in Figure 15.17b, which implies that in the noisy-channel case increasing the bandwidth ( $q$  value) above a certain threshold does not improve  $D$ , where the threshold is jointly decided by the power constraint and dimension of the vector source.

## 15.7 CONCLUDING REMARKS

In this chapter, we have investigated the energy-efficient decentralized estimation in a wireless sensor network with a fusion center. Specifically, we discussed both the analog and digital approaches and studied the optimal power scheduling. For the digital approach, the proposed power-scheduling scheme suggests that the sensors with bad channels or poor observation qualities should decrease their quantization resolutions or simply become inactive in order to save power. For the remaining active sensors, their optimal quantization and transmit power levels are jointly determined by individual



**Figure 15.17** The eigenvalue distribution of  $\mathbf{Q}^*$  (left); MSE vs. bandwidth  $q$  (right); parameters:  $K = 8$ ,  $\ell_i = 1$ .

channel gains and local observation SNRs. It has been shown that such an optimal design strategy can lead to significant power savings when compared to the uniform scheduling of quantization bits and transmit powers across sensors.

For the analog approach, we considered the analog amplify-and-forward scheme for the estimation problem. Both orthogonal and coherent MACs were discussed. For the case of orthogonal MAC, we analyzed the system outage performance by exploring the estimation diversity brought by independent fading channels from different sensors. It is shown that the achievable diversity gain in a Rayleigh fading environment equals the total number of sensors in the network. In addition, selectivity gains can be achieved by turning off bad-quality sensors (e.g., sensors with poor observation qualities or small channel gains) without sacrificing the diversity gain. We have also investigated the impact of multiple access on the overall power and distortion performance. When the total network power consumption is fixed, for the orthogonal MAC, there is a floor

level for the achievable distortion even when the total number of sensors increases to infinity, while for the coherent MAC case, the end-to-end distortion vanishes over the network size and is asymptotically optimal.

We then extended the analog approach to the decentralized estimation of an unknown vector signal. We have shown that when the MAC channels between sensors and the fusion center are orthogonal, the complexity of designing the optimal encoding matrices is NP-hard in general. However, for the case of coherent MAC, when the channel is noiseless, the resulting problem has a closed-form solution, while in the noisy-channel case, the problem can be efficiently solved by SDP.

We also analyzed the optimality of digital and analog approaches by discussing the sensor network source–channel communication problem. We established the result that for the multiple-access channel with correlated sources, if the MAC interference between sensors and the fusion center is resolved by reservation-based protocols (e.g., TDMA/FDMA), the optimal trade-off between cost and distortion can be achieved by separate source and channel coding, while for the nonorthogonal case, the analog amplify-and-forward approach is asymptotically optimal and significantly outperforms the digital approaches.

## ACKNOWLEDGMENTS

The work of Xiao and Luo is supported in part by the National Science Foundation, grant number DMS-0610037, and by the USDOD ARMY, grant number W911NF-05-1-0567. The work of Cui is supported in part by the National Science Foundation, grant number CNS-0721935 and CCF-0726740, and USDOD, grant number HDTRA1-07-1-0037.

## REFERENCES

1. M. Bhardwaj, T. Garnett, and A. P. Chandrakasan, “Upper bounds on the lifetime of sensor networks,” *Proc. IEEE Int. Conf. Commun.*, vol. 3, pp. 785–790, June 2001.
2. A. P. Chandrakasan, R. Min, M. Bhardwaj, S. Cho, and A. Wang, “Power aware wireless microsensor systems,” Keynote paper at *ESSCIRC*, Florence, Italy, Sept. 2002.
3. J. M. Rabaey, M. J. Ammer, J. L. da Silva, D. Patel, and S. Roundy, “PicoRadio supports ad hoc ultralow power wireless networking,” *IEEE Computer*, vol. 33, no. 7, pp. 42–48, July 2000.
4. M. Cardei and J. Wu, “Energy-efficient coverage problems in wireless ad hoc sensor networks,” *J. Computer Commun. Sensor Networks*, to be published.
5. S. Lindsey, C. Raghavendra, and K. M. Sivalingam, “Data gathering algorithms in sensor networks using energy metrics,” *IEEE Trans. Parallel Distributed Syst.*, vol. 13, pp. 924–935, Sept. 2002.
6. W. Ye, J. Heidemann and D. Estrin, “An energy-efficient mac protocol for wireless sensor networks,” in *Proc. 21st Annual Joint Conf. IEEE Computer and Commun. Societies*, Vol. 3, New York, June 2002, pp. 1567–1576.
7. J. N. Al-Karaki and A. E. Kamal, “Routing techniques in wireless sensor networks: a survey,” *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 38–45, Mar.–Apr. 2004.
8. I. F. Akyildiz, W. Su, Y. Sankarsubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” *Computer Networks*, vol. 38, pp. 393–422, Mar. 2002.

9. D. A. Castanon and D. Teneketzis, "Distributed estimation algorithms for nonlinear systems," *IEEE Trans. Automatic Control*, vol. 30, no. 5, pp. 418–425, May 1985.
10. J. L. Speyer, "Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem," *IEEE Trans. Automatic Control*, vol. 24, no. 2, pp. 266–269, Apr. 1979.
11. A. S. Willsky, M. Bello, D. A. Castanon, B. C. Levy, and G. Verghese, "Combining and updating of local estimates and regional maps along sets of one-dimensional tracks," *IEEE Trans. Automatic Control*, vol. 27, no. 4, pp. 799–813, Aug. 1982.
12. Z. Chair and P. K. Varshney, "Distributed bayesian hypothesis testing with distributed data fusion," *IEEE Trans. Syst. Man Cybernet.*, vol. 18, no. 5, pp. 695–699, Sept.–Oct. 1988.
13. Z.-Q. Luo and J. N. Tsitsiklis, "On the communication complexity of distributed algebraic computation," *J. Assoc. Comput. Machinery*, vol. 40, pp. 1019–1047, Nov. 1993.
14. K. Liu, H. El-Gamal, and A. Sayeed, "On optimal parametric field estimation in sensor networks," in *Proc. IEEE/SP 13th Workshop on Statistical Signal Processing*, July 2005, pp. 1170–1175.
15. G. Mergen and L. Tong, "Type based estimation over multiaccess channels," *IEEE Trans. Signal Process.*, vol. 54, no. 2, pp. 613–626, Feb. 2006.
16. H. C. Papadopoulos, G. W. Wornell, and A. V. Oppenheim, "Sequential signal encoding from noisy measurements using quantizers with dynamic bias control," *IEEE Trans. Inform. Theory*, vol. 47, no. 3, pp. 978–1002, Mar. 2001.
17. A. Ribeiro and G. B. Giannakis, "Bandwidth-constrained distributed estimation for wireless sensor networks, Part I: Gaussian case," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1131–1143, Mar. 2006.
18. J. A. Gubner, "Distributed estimation and quantization," *IEEE Trans. Inform. Theory*, vol. 39, no. 4, pp. 1456–1459, July 1993.
19. W. M. Lam and A. R. Reibman, "Design of quantizers for decentralized estimation systems," *IEEE Trans. Commun.*, vol. 41, no. 11, pp. 1602–1605, Nov. 1993.
20. K. Eswaran and M. Gastpar, "On the quadratic AWGN CEO problem and non-Gaussian sources," in *Proc. IEEE International Symposium on Information Theory*, Adelaide, Australia, Sept. 2005, pp. 219–223.
21. M. Gastpar and M. Vetterli, "Source-channel communication in sensor networks," *Lecture Notes in Computer Science*, vol. 2634, pp. 162–177, Springer, New York, Apr. 2003.
22. P. Ishwar, R. Puri, K. Ramchandran, and S. S. Pradhan, "On rate-constrained distributed estimation in unreliable sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 765–775, Apr. 2005.
23. A. D. Sarwate and M. Gastpar, "Fading observation alignment via feedback," in *Proc. Fourth International Symposium on Information Processing in Sensor Networks*, 2005.
24. D. Marco, E. Duarte-Melo, M. Liu, and D. Neuhoff, "On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data," in *Proc. Second International Symposium on Information Processing in Sensor Networks*, 2003.
25. G. Mergen, V. Naware, and L. Tong, "Asymptotic detection performance of type-based multiple access in sensor networks," paper presented at the IEEE Workshop on Signal Processing Advances in Wireless Communications, New York, June 2005.
26. T. J. Goblick, "Theoretical limitations on the transmission of data from analog sources," *IEEE Trans. Inform. Theory*, vol. 11, pp. 558–567, Oct. 1965.
27. Z.-Q. Luo, "Communication complexity of some problems in distributed computation," PhD dissertation, technical report LIDS-TH-1909, Lab. for Information and Decision Systems, MIT, Cambridge, MA, 1989.

28. V. Megalooikonomou and Y. Yesha, "Quantizer design for distributed estimation with communication constraints and unknown observation statistics," *IEEE Trans. Commun.*, vol. 48, no. 2, pp. 181–184, Feb. 2000.
29. J. N. Tsitsiklis, "Decentralized detection," *Adv. Statist. Signal Process.*, vol. 2, pp. 297–344, 1993.
30. A. C. Yao, "Some complexity questions related to distributed computing," in *Proc. 11th Symp. on Theory of Computing*, pp. 209–213, 1979.
31. K. Zhang, X. R. Li, P. Zhang, and H. Li, "Optimal linear estimation fusion—Part VI: Sensor data compression," in *Proc. International Conf. on Information Fusion*, Queensland, Australia, 2003, pp. 221–228.
32. Z.-Q. Luo, "Universal decentralized estimation in a bandwidth constrained sensor network," *IEEE Trans. Inform. Theory*, vol. 51, no. 6, pp. 2210–2219, June 2005.
33. Z.-Q. Luo, "An isotropic universal decentralized estimation scheme for a bandwidth constrained ad hoc sensor network," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 735–744, Apr. 2005.
34. S. Cui, A. J. Goldsmith, and A. Bahai, "Joint modulation and multiple access optimization under energy constraints," in *Proc. IEEE Global Conf. on Communications*, Dallas, TX, Dec. 2004, pp. 151–155.
35. S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained modulation optimizatoin," *IEEE Trans. Wireless Commun.*, vol. 4, No. 5, pp. 2349–2360, Sept. 2005.
36. A. J. Goldsmith and S. B. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 4, pp. 8–27, Aug. 2002.
37. J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. J. Goldsmith, "Power scheduling of universal decentralized estimation in sensor networks," *IEEE Trans. Signal Process.*, vol. 54, no. 2, pp. 413–422, Feb. 2006.
38. J.-J. Xiao and Z.-Q. Luo, "Universal decentralized estimation in an inhomogeneous sensing environment," *IEEE Trans. Inform. Theory*, vol. 51, no. 10, pp. 3564–3575, Oct. 2005.
39. S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, Englewood Cliffs, NJ, 1993.
40. S. Cui, J. Xiao, A. J. Goldsmith, Z. Q. Luo, and H. V. Poor, "Estimation diversity and energy efficiency in distributed sensing," *IEEE Trans. Signal Process.*, vol. 55, no. 9, pp. 4683–4695, Sept. 2007.
41. J. F. Chamberland and V. Veeravalli, "Asymptotic results for decentralized detection in power-constrained wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, pp. 1007–1015, Aug. 2004.
42. A. D'Costa, V. Ramchandran, and A. Sayeed, "Distributed classification of gaussian space-time sources in wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, pp. 1026–1036, Aug. 2004.
43. K. Liu and A. Sayeed, "Optimal distributed detection strategies for wireless sensor networks," paper presented at The 43rd Allerton Conf. on Communications, Control, and Computing, Monticello, IL, Oct. 2004.
44. J.-J. Xiao and Z.-Q. Luo, "Universal decentralized detection in a bandwidth constrained sensor network," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2617–2624, Aug. 2005.
45. A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, Cambridge, 2003.
46. J.-J. Xiao and Z.-Q. Luo, "Multiterminal source-channel communication over an orthogonal multiple access channel," *IEEE Trans. Inform. Theory*, vol. 53, no. 9, pp. 3255–3264, Sept. 2007.

47. T. Berger, "Multiterminal source coding," in *The Information Theory Approach to Communications*, G. Longo (Ed.), vol. 229 of CISM Courses and Lectures, Springer-Verlag, 1978, pp. 171–231.
48. T. Berger, Z. Zhang, and H. Viswanathan, "The CEO problem," *IEEE Trans. Inform. Theory*, vol. 42, pp. 887–902, May 1996.
49. S. Y. Tung, "Multiterminal source coding," PhD Thesis, School of Electrical Engineering, Cornell University, Ithaca, NY, May 1978.
50. Y. Oohama, "The rate-distortion function for the quadratic Gaussian CEO problem," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1057–1070, May 1998.
51. C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.
52. T. M. Cover and J. Thomas, *Elements of Information Theory*, Wiley, New York, 1991.
53. T. M. Cover, A. A. El Gamal, and M. Salehi, "Multiple access channels with arbitrarily correlated sources," *IEEE Trans. Inform. Theory*, vol. 26, pp. 648–657, Nov. 1980.
54. J. Barros and S. D. Servetto, "Network information flow with correlated sources," *IEEE Trans. Inform. Theory*, to be published.
55. T. S. Han, "Cover-Slepian-Wolf theorem for a network of channels," *Inform. Control*, vol. 47, pp. 67–83, 1980.
56. S. Cui, R. Madan, A. J. Goldsmith, and S. Lall, "Cross-layer energy and delay optimization in small-scale sensor networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 10, pp. 3688–3699, Oct. 2007.
57. Z.-Q. Luo, G. B. Giannakis, and S. Zhang, "Optimal linear decentralized estimation in a bandwidth constrained sensor network," in *Proc. IEEE International Symposium on Information Theory*, Adelaide, Australia, Sept. 2005, pp. 1441–1445.
58. J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. J. Goldsmith, "Linear coherent decentralized estimation," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 757–770, Feb. 2008.
59. S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, 2003.
60. J. F. Sturm, "Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones," available: [http://www.optimization-online.org/DB\\_HTML/2001/10/395.html](http://www.optimization-online.org/DB_HTML/2001/10/395.html).
61. M. Gastpar, P. L. Dragotti, and M. Vetterli, "The distributed Karhunen-Loéve transform," *IEEE Trans. Inform. Theory*, to be published.
62. I. D. Schizas, G. B. Giannakis, and N. Jindal, "Distortion-rate analysis for distributed estimation with wireless sensor networks," *Proc. 43rd Allerton Conf. on Communications, Control, and Computing*, Sept. 2005.



---

## CHAPTER 16

---

# Sensor Data Fusion with Application to Multitarget Tracking

R.Tharmarasa<sup>1</sup>, K. Punithakumar<sup>1</sup>, T. Kirubarajan<sup>1</sup>, and Y. Bar-Shalom<sup>2</sup>

<sup>1</sup>ECE Department, McMaster University, Hamilton Ontario, Canada

<sup>2</sup>ECE Department, University of Connecticut, Storrs Connecticut

### 16.1 INTRODUCTION

Multisensor data fusion is an emerging technology in which data from several sensing devices are combined such that the resulting information is significantly better than that obtained when these devices operate individually. Recent advances in sensor technologies, signal processing techniques, and improved processor capabilities make it possible for large amounts of data to be fused in real time. This allows the use of many sophisticated algorithms and robust mathematical techniques in data fusion. Moreover, data fusion has received significant attention for military applications. Such applications involve a wide range of expertise, including filtering, data association, out-of-sequence measurements, and sensor registration.

Filtering plays a vital role in data fusion by obtaining the state estimate from the data received from one or more sensors. Tracking filters [1, 2] can be broadly categorized as linear and nonlinear. The Kalman filter [1, 3] is a widely known recursive filter that is most suited for linear Gaussian systems. However, most systems are inherently nonlinear. The extensions of Kalman filter, such as extended Kalman filter (EKF) [1] and unscented Kalman filter (UKF) [4–6], are applicable to nonlinear systems. They are restricted in that the resulting probability densities are approximated as Gaussian. When the system is nonlinear and non-Gaussian, particle filters (or sequential Monte Carlo methods) [7–10] provide better estimates than many other filtering algorithms. All the above filters are directly applicable to single-target tracking systems. However, they require special techniques to correlate measurements to tracks if they are to be applied to multitarget tracking. The recent probability hypothesis density (PHD) [11–17] filter, which avoids explicitly correlating measurements to tracks, can be applied directly to multitarget tracking problems. In general, a tracking filter requires a model for target dynamics, and a model mismatch would diminish the performance of the filter. Especially in the case of maneuvering targets, whose kinematics may evolve in a time-varying manner, different models may be required to describe the target dynamics

accurately at different times. Multiple model tracking algorithms such as the interacting multiple-model (IMM) [1, 18–22] estimator, which contains a bank of models matched to different modes of possible target dynamics, would perform better in such situations.

Data association [23–25] is an essential component in sensor fusion when there is uncertainty in the source of data origin. Data association refers to the methodology of correctly associating the measurements to tracks, measurements to measurements [26, 27], or tracks to tracks [23, 28–33] depending on the fusion architecture. To address data association, a number of techniques have been developed. Few such techniques include probability data association (PDA) [23, 34], joint probability data association (JPDA) [23], multiple hypothesis tracking (MHT) [23, 29, 35], single-frame assignment algorithm [36, 37], and the multiframe assignment algorithm [36–42].

In practical network systems, the measurement may not arrive in sequence due to network delays. This problem is identified as out-of-sequence measurement (OOSM) [43–46] processing and can be categorized as one-lag OOSM or multiple-lag OOSM, depending on the delays. A number of algorithms have been developed to address OOSM issues in multisensor networks [43–45]. These algorithms involve specifically accounting for the out-of-sequence reception of measurements and then adjusting the estimates as well as the corresponding error covariances so that the estimate at the latest time remains optimal.

The benefits afforded by the integration of data from multiple sensors are greatly diminished if sensor biases are present. In practical systems, measurements may be subjected to pointing errors, calibration errors, or computational errors, which in turn introduce the bias. As a result, knowledge about sensor location or attitude may not be accurate. This may result in severe degradation in the performance of data association, filtering, and fusion, leading to eventual loss of track quality. In sensor registration [47–49], the bias is estimated and the resulting values are used to debias measurements prior to fusion.

In this chapter, various choices for algorithms to handle data association, state estimation, fusion, fusion architecture, measurement processing, and debiasing are discussed in detail. In addition, their quantitative and qualitative merits are discussed. Various combinations of these algorithms will provide a complete tracking and fusion framework for multisensor networks with application to civilian as well as military problems. For example, the tracking and fusion techniques discussed here are applicable to fields like air traffic control, air/ground/maritime surveillance, mobile communication, transportation, video monitoring, and biomedical imaging/signal processing. The application of some of these algorithms in a sensor network environment is demonstrated on a representative real scenario where up to 1000 aircraft are tracked using multiple Federal Aviation Administration (FAA) radar systems. It is also shown that the algorithms are capable of processing large amounts of data and extracting all information available therein while remaining real time feasible. The benefits of multisensor fusion compared with single-sensor processing are quantified as well.

## 16.2 TRACKING FILTERS

Filtering is the estimation of the state of a dynamic system from noisy data. In order to estimate the state of a moving object (target), at least two models are required [1]:

1. *System Model* A model describing the evolution of the state with time, given by

$$x(k+1) = f(k, x(k)) + v(k). \quad (16.1)$$

2. *Measurement Model* A model relating the noisy measurements to the state, given by

$$z(k) = h(k, x(k)) + \omega(k), \quad (16.2)$$

where  $f$  and  $h$  are, in general, nonlinear functions,  $x(k)$  is the state of the target,  $z(k)$  is the measurement vector,  $v(k)$  is the process noise, and  $\omega(k)$  is the measurement noise at measurement time  $k$ .

In the Bayesian approach to dynamic state estimation, the aim is to construct the posterior probability density function (pdf) of the state given all the received measurements so far. Since this pdf contains all available statistical information, it is the complete solution to the estimation problem *if all the noises are white*. In principle, an optimal estimate of the state may be obtained from this pdf. In recursive filtering, the received measurements are processed sequentially rather than as a batch so that it is neither necessary to store the complete measurement set nor to reprocess existing measurement if a new measurement becomes available. Such a filter consists of two stages: prediction and update.

The prediction stage uses the system model to predict the state pdf forward from one measurement time to the next. Suppose that the required pdf  $p(x(k)|Z^k)$  at measurement time  $k$  is available, where  $Z^k = [z(1), z(2), \dots, z(k)]$ . The prediction stage involves using the system model (16.1) to obtain the prior pdf of the state at measurement time  $k+1$  and given by

$$p(x(k+1)|Z^k) = \int p(x(k+1)|x(k)) p(x(k)|Z^k) dx(k). \quad (16.3)$$

The update stage uses the latest measurement  $z(k+1)$  to update the prior via Bayes' formula

$$p(x(k+1)|Z^{k+1}) = \frac{p(z(k+1)|x(k+1)) p(x(k+1)|Z^k)}{p(z(k+1)|Z^k)}. \quad (16.4)$$

The above recursive propagation of the posterior density is only a conceptual solution and in general cannot be determined analytically. Analytical solutions exist only in a restrictive set of cases.

### 16.2.1 Kalman Filter

The Kalman filter assumes that the state and measurement models are linear, that is,  $f(k, x(k)) = F(k)x(k)$ ;  $h(k, x(k)) = H(k)x(k)$ , and the initial state error and all the noises entering into the system are Gaussian, that is,  $v(k)$  is white and Gaussian with zero mean and covariance  $Q(k)$ , and  $\omega(k)$  is white and Gaussian with zero mean and covariance  $R(k)$ . Under the above assumptions, if  $p(x(k)|Z^k)$  is Gaussian, it can be proved that  $p(x(k+1)|Z^{k+1})$  is also Gaussian and can be parameterized by a mean and covariance [1].

The Kalman filter algorithm consists of the following recursive relationship [1]:

$$\hat{x}(k+1|k) = F(k)\hat{x}(k|k), \quad (16.5)$$

$$P(k+1|k) = F(k)P(k|k)F(k)' + Q(k), \quad (16.6)$$

$$\hat{z}(k+1|k) = H(k+1)\hat{x}(k+1|k), \quad (16.7)$$

$$S(k+1) = H(k+1)P(k+1|k)H(k+1)' + R(k+1), \quad (16.8)$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)(z(k+1) - \hat{z}(k+1|k)), \quad (16.9)$$

$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W(k+1)', \quad (16.10)$$

where

$$W(k+1) = P(k+1|k)H(k+1)'S(k+1)^{-1}. \quad (16.11)$$

This is the optimal solution to the tracking problem if the above assumptions hold [1]. The implication is that no algorithm can do better than a Kalman filter in this linear Gaussian environment.

### 16.2.2 Information Filter

The information filter is an alternative form of the Kalman filter [1]. The information state vector  $\hat{y}(k|k-i)$  and information matrix  $Y(k|k-i)$  are defined as

$$\hat{y}(k|k-i) = P(k|k-i)^{-1}\hat{x}(k|k-i), \quad (16.12)$$

$$Y(k|k-i) = P(k|k-i)^{-1}. \quad (16.13)$$

The measurement information vector and corresponding matrix are defined as

$$i(k) = H(k)'R(k)^{-1}z(k), \quad (16.14)$$

$$I(k) = H(k)'R(k)^{-1}H(k). \quad (16.15)$$

The advantage of the information filter is that measurements from multiple sensors can be filtered easily by summing their information matrices and vectors:

$$\hat{y}(k|k) = \hat{y}(k|k-1) + \sum_{j=1}^{N_S} i(k, j), \quad (16.16)$$

$$Y(k|k) = Y(k|k-1) + \sum_{j=1}^{N_S} I(k, j), \quad (16.17)$$

where  $N_S$  is the number of sensors.

### 16.2.3 Extended Kalman Filter

While the Kalman filter assumes linearity, most of the real-world problems are nonlinear. The extended Kalman filter is a suboptimal state estimation algorithm for nonlinear systems. In EKF, local linearizations of the equations are used to describe the nonlinearity:

$$\hat{F}(k) = \frac{df(k)}{dx} \Big|_{x=\hat{x}(k|k)}, \quad (16.18)$$

$$\hat{H}(k) = \frac{dh(k+1)}{dx} \Big|_{x=\hat{x}(k+1|k)}. \quad (16.19)$$

The EKF is based on  $p(x(k)|Z^k)$  approximated by a Gaussian. Then the equations of the Kalman filter can be used with this approximation and the linearized functions [1], except the state and measurement prediction, are performed using the original nonlinear functions:

$$\hat{x}(k+1|k) = f(k, \hat{x}(k|k)), \quad (16.20)$$

$$\hat{z}(k+1|k) = h(k+1, \hat{x}(k+1|k)). \quad (16.21)$$

The above is a first-order EKF based on first-order series expansion of the nonlinearities. Higher order EKFs also exist, but the additional complexity and little or no benefit has prohibited its widespread use.

### 16.2.4 Unscented Kalman Filter

When the state transition and observation models are highly nonlinear, the EKF may perform poorly. The unscented Kalman filter does not approximate the nonlinear functions of state and measurement models as required by the EKF. Instead, the UKF uses a deterministic sampling technique known as the unscented transform to pick a minimal set of sample points called sigma points around the mean. Here, the propagated mean and covariance are calculated from the transformed samples [4]. The steps of UKF are described below.

**16.2.4.1 Sigma Point Generation** The state vector  $\hat{x}(k)$  with mean  $\hat{x}(k|k)$  and covariance  $P(k|k)$  is approximated by  $2n + 1$  weighted sigma points, where  $n$  is the dimension of the state vector, as

$$\chi^0(k|k) = \hat{x}(k|k), \quad w_0 = \frac{\kappa}{n + \kappa}, \quad (16.22)$$

$$\chi^i(k|k) = \hat{x}(k|k) + \left( \sqrt{(n + \kappa)P(k|k)} \right)_i, \quad w_i = \frac{1}{2(n + \kappa)}, \quad (16.23)$$

$$\chi^{i+n}(k|k) = \hat{x}(k|k) - \left( \sqrt{(n + \kappa)P(k|k)} \right)_i, \quad w_{i+n} = \frac{1}{2(n + \kappa)}, \quad (16.24)$$

where  $w_i$  is the weight associated with the  $i$ th point,  $\kappa$  is a scaling parameter,  $i = 1, 2, \dots, n$ , and  $\left( \sqrt{(n + \kappa)P(k|k)} \right)_i$  is the  $i$ th row or column of the matrix square root of  $(n + \kappa)P(k|k)$ .

#### 16.2.4.2 Recursion

1. Find the predicted target state  $\hat{x}(k+1|k)$  and corresponding covariance  $P(k+1|k)$ :
  - a. Transform the sigma points using the process model

$$\chi^i(k+1|k) = f(k, \chi^i(k|k)). \quad (16.25)$$

b. Find the predicted mean

$$\hat{x}(k+1|k) = \sum_{i=0}^{2n} w_i \chi^i(k+1|k). \quad (16.26)$$

c. Find the predicted covariance

$$\begin{aligned} P(k+1|k) = Q(k) + \sum_{i=0}^{2n} w_i [\chi_i(k+1|k) - \hat{x}(k+1|k)] & [\chi_i(k+1|k) \\ & - \hat{x}(k+1|k)]'. \end{aligned} \quad (16.27)$$

2. Find the predicted measurement  $\hat{z}(k+1|k)$  and the corresponding covariance  $S(k+1)$ :

a. Regenerate the sigma points  $\chi_i(k+1|k)$  using the mean  $\hat{x}(k+1|k)$  and covariance  $P(k+1|k)$  in order to incorporate the effect of  $Q(k)$ . If  $Q(k)$  is zero, the resulting  $\chi_i(k+1|k)$  will be the same as in (16.25). If the process noise is correlated with the state, then the noise vector must be stacked with the state vector  $\hat{x}(k|k)$  before generating the sigma points [4].

b. Find the predicted measurement mean  $\hat{z}(k+1|k)$ :

$$\hat{z}(k+1|k) = \sum_{i=0}^{2n} w_i \varphi^i(k+1|k), \quad (16.28)$$

where

$$\varphi^i(k+1|k) = h(k, \chi^i(k+1|k)). \quad (16.29)$$

c. Find the innovation covariance  $S(k+1)$  and gain  $W(k+1)$ :

$$\begin{aligned} S(k+1) = R(k+1) + \sum_{i=0}^{2n} w_i [\varphi^i(k+1|k) - \hat{z}(k+1|k)] \\ \times [\varphi^i(k+1|k) - \hat{z}(k+1|k)]', \end{aligned} \quad (16.30)$$

$$\begin{aligned} W(k+1) = \sum_{i=0}^{2n} w_i [\chi^i(k+1|k) - \hat{x}(k+1|k)] \\ \times [\chi^i(k+1|k) - \hat{x}(k+1|k)]' S(k+1)^{-1}. \end{aligned} \quad (16.31)$$

3. Update the state  $\hat{x}(k+1|k+1)$  and corresponding covariance  $P(k+1|k+1)$  using (16.9) and (16.10), respectively.

### 16.2.5 Particle Filter

If the true density is substantially non-Gaussian, then a Gaussian model as in the case of the Kalman filter will not yield accurate estimates. In such cases, particle

filters will yield an improvement in performance in comparison to the EKF or UKF. The particle filter provides a mechanism for representing the density,  $p(x(k)|Z^k)$  of the state vector  $x(k)$  at time epoch  $k$  as a set of random samples  $\{x^p(k) : p = 1, \dots, m\}$ , with associated weights  $\{w^p(k) : p = 1, \dots, m\}$ . That is, the particle filter attempts to represent an arbitrary density function using a finite number of points, instead of a single point that is sufficient for Gaussian distributions. Several variations of particle filters are available, and the reader is referred to [7] for detailed description. The sampling importance resampling (SIR) type of particle filter, which is arguably the most common technique to implement particle filters, is discussed below. In general, the particles are sampled either from the prior density or likelihood function. Taking the prior as the importance density, the method of SIR is used to produce a set of equally weighted particles that approximates  $p(x(k)|Z^k)$ , that is,

$$p(x(k)|Z^k) \approx \frac{1}{m} \sum_{p=1}^m \delta(x(k) - x^p(k)), \quad (16.32)$$

where  $\delta(\cdot)$  is the Dirac delta function. The prediction and update steps of the particle filter recursion are given below.

**Prediction** Take each existing sample,  $x^p(k)$ , and generate a sample  $x^{*p}(k+1) \sim p(x(k+1)|x^p(k))$ , using the system model. The set  $\{x^{*p}(k+1) : p = 1, \dots, m\}$  provides an approximation of the prior,  $p(x(k+1)|Z^k)$ , at time  $k+1$ .

**Update** At each measurement epoch, to account for the fact that the samples,  $x^{*p}(k+1)$  are not drawn from  $p(x(k+1)|Z^{k+1})$ , the weights are modified using the principle of importance sampling. When using the prior as the importance density, it can be shown that the weights are given by

$$w^p(k+1) \propto p(z(k+1)|x(k+1) = x^{*p}(k+1), Z^k). \quad (16.33)$$

A common problem with the above recursion is the degeneracy phenomenon, whereby the particle set quickly collapses to just a single particle. To overcome this problem a regularization can be imposed via reselection as follows.

**Reselection** Resample (with replacement) from  $\{x^{*p}(k+1) : p = 1, \dots, m\}$ , using the weights,  $\{w^{*p}(k+1) : p = 1, \dots, m\}$ , to generate a new sample,  $\{x^p(k+1) : p = 1, \dots, m\}$ , then set  $w^p(k+1) = 1/m$  for  $p = 1, \dots, m$ .

The mean of the posterior distribution is used to estimate,  $x(k+1|k+1)$  of the target state,  $x(k+1)$ , that is,

$$\hat{x}_{k+1} \approx \frac{1}{m} \sum_{p=1}^m x^p(k+1). \quad (16.34)$$

The accuracy of the particle filter-based estimate (16.32) depends on the number of particles employed. A more accurate state estimate can be obtained at the expense of

extra computation. The extension of particle filters allows them to be applicable to multitarget tracking problems [50].

### 16.2.6 Probability Hypothesis Density Method

In tracking multiple targets, if the number of targets is unknown and varying with time, it is not possible to compare states with different dimensions using ordinary Bayesian statistics. However, the problem can be addressed using finite set statistics (FISST) [11] to incorporate comparisons of states of different dimensions. FISST facilitates the construction of “multitarget densities” from multiple-target transition functions into computing set derivatives of belief mass functions [11], which makes it possible to combine states of different dimensions. The main practical difficulty with this approach is that the dimension of state space becomes large when many targets are present, which increases the computational load exponentially with the number of targets. Since the PHD is defined over the state space of one target in contrast to the full posterior distribution, which is defined over the state space of all the targets, the computation cost of propagating the PHD over time is much lower than propagating the full posterior density. A comparison in terms of computation and estimation accuracy of multitarget filtering using FISST particle filter and PHD particle filter is given in [16].

By definition, the PHD  $D_{k|k}(\mathbf{x}_k|Z^k)$ , with argument single-target state vector  $\mathbf{x}_k$  and given all the measurements  $Z^k$  up to time step  $k$ , is the density whose integral on any region  $S$  of the state space is the expected number of targets  $N_{k|k}$  contained in  $S$ . That is,

$$N_{k|k} = \int_S D_{k|k}(\mathbf{x}_k|Z^k) d\mathbf{x}_k. \quad (16.35)$$

Since this property uniquely characterizes the PHD and since the first-order statistical moment of the full target posterior distribution possesses this property, the first-order statistical moment of the full target posterior is indeed the PHD. The first moment of the full target posterior or the PHD, given all the measurement  $Z^k$  up to time step  $k$ , is given by [12]

$$D_{k|k}(\mathbf{x}_k|Z^k) = \int_{X_k \ni \mathbf{x}_k} f_{k|k}(X_k|Z^k) \delta X_k, \quad (16.36)$$

where  $X_k$  is the multitarget state. The approximate expected target states are given by the local maxima of the PHD. The following section gives the prediction and update steps of one cycle of the PHD filter.

**Prediction** In a general scenario of interest, there are target disappearances, target spawning, and entry of new targets. We denote the probability that a target with state  $\mathbf{x}_{k-1}$  at time step  $(k-1)$  will survive at time step  $k$  by  $e_{k|k-1}(\mathbf{x}_{k-1})$ , the PHD of spawned targets at time step  $k$  from a target with state  $\mathbf{x}_{k-1}$  by  $b_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$ , and the PHD of newborn spontaneous targets at time step  $k$  by  $\gamma_k(\mathbf{x}_k)$ . Then, the predicted PHD is given by

$$\begin{aligned} D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}) &= \gamma_k(\mathbf{x}_k) + \int [e_{k|k-1}(\mathbf{x}_{k-1}) f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})] \\ &\quad + b_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})] D_{k-1|k-1}(\mathbf{x}_{k-1}|Z_{1:k-1}) d\mathbf{x}_{k-1}, \end{aligned} \quad (16.37)$$

where  $f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$  denotes the single-target Markov transition density. The prediction equation (16.24) is lossless since there are no approximations.

**Update** The predicted PHD can be corrected with the available measurements  $Z_k$  at time step  $k$  to get the updated PHD. We assume that the number of false alarms is Poisson distributed with the average spatial density of  $\lambda_k$  and that the probability density of the spatial distribution of false alarms is  $c_k(\mathbf{z}_k)$ . Let the detection probability of a target with state  $\mathbf{x}_k$  at time step  $k$  be  $p_D(\mathbf{x}_k)$ . Then, the updated PHD at time step  $k$  is given by

$$D_{k|k}(\mathbf{x}_k|Z^k) \cong \left[ \sum_{\mathbf{z}_k \in Z_k} \frac{p_D(\mathbf{x}_k) f_{k|k}(\mathbf{z}_k|\mathbf{x}_k)}{\lambda_k c_k(\mathbf{z}_k) + \psi_k(\mathbf{z}_k|Z_{1:k-1})} + (1 - p_D(\mathbf{x}_k)) \right] D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}), \quad (16.38)$$

where the likelihood function  $\psi(\cdot)$  is given by

$$\psi_k(\mathbf{z}_k|Z_{1:k-1}) = \int p_D(\mathbf{x}_k) f_{k|k}(\mathbf{z}_k|\mathbf{x}_k) D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}) d\mathbf{x}_k, \quad (16.39)$$

and  $f_{k|k}(\mathbf{z}_k|\mathbf{x}_k)$  denotes the single-sensor/single-target likelihood. The update equation (16.38) is not lossless since approximations are made on predicted multitarget posterior to obtain the closed-form solution (16.38).

### 16.2.7 Interacting Multiple-Model Estimator

The multiple-model approach to tracking maneuvering targets by detecting maneuvers and identifying the appropriate model has been shown to be highly effective. In this approach, a finite number of filters operate in parallel, and the target motion is assumed to be in one of the models in the mode set of the tracker. In many target-tracking problems with linear, Gaussian systems, the interacting multiple-model (IMM) estimator [1, 20, 21] in which a bank of different hypothetical target motion models is used has been proven to have better performance over the (single-model) Kalman filter.

The IMM approach has been shown to be the most effective among the other multiple-model approaches such as the generalized pseudo-Bayesian (GPB) [18, 22] algorithms considering the compromise between complexity and performance. A GPB algorithm of order  $n$  ( $GPB_n$ ) requires  $N_r^n$  filters in its bank, where  $N_r$  is the number of models. The IMM estimator performs nearly as well as  $GPB_2$  but requires only  $N_r$  number of filters to operate in parallel. Thus it has significantly less computational complexity, which is almost same as that of  $GPB_1$ . Further, the IMM estimator does not require maneuver detection decision as in the case of variable state dimension (VSD) filter [1] algorithms and undergoes a soft switching between models based on the updated mode probabilities.

The special feature of the IMM estimator that distinguishes it from other suboptimal multiple-model (MM) estimators is the “mixing/interaction” between its “mode-matched” base state filtering modules at the beginning of each cycle. As shown in [51], the same feature is exactly what the IMM has in common with the *optimal* estimator for hybrid (MM) systems, and this can be seen as the main reason for its success.

### 16.2.7.1 Modeling Assumptions

#### Base State Model

$$x(k) = F[M(k)]x(k-1) + v[k-1, M(k)], \quad (16.40)$$

$$z(k) = H[M(k)]x(k) + w[k, M(k)]s, \quad (16.41)$$

where  $M(k)$  denotes the mode “at time  $k$ ” —in effect *during the sampling period ending at  $k$* .

*Mode (“Modal State”)* Among the possible  $r$  modes:

$$M(k) \in \{M_j\}_{j=1}^r. \quad (16.42)$$

The structure of the system and/or the statistics of the noises can differ from mode to mode:

$$F[M_j] = F_j, \quad (16.43)$$

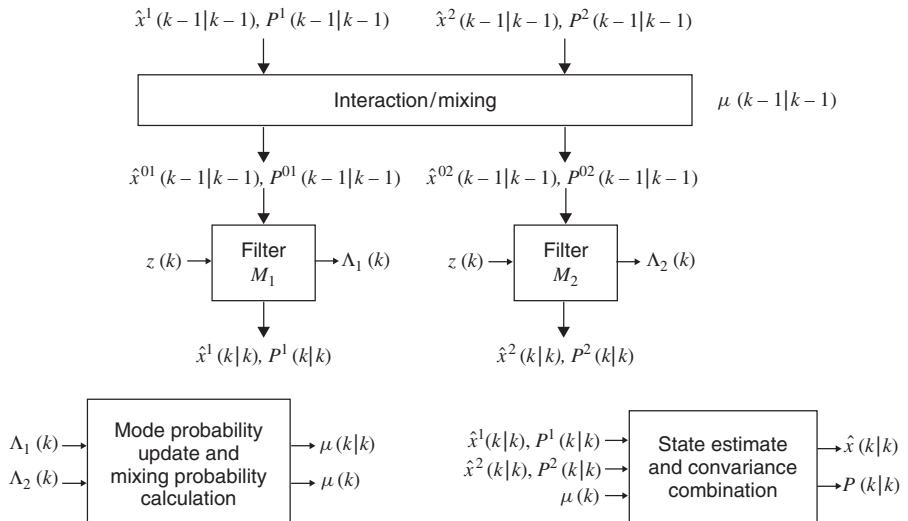
$$v(k-1, M_j) \sim \mathcal{N}(u_j, Q_j). \quad (16.44)$$

*Mode Jump Process* Markov chain with known transition probabilities:

$$P\{M(k) = M_j | M(k-1) = M_i\} = p_{ij}. \quad (16.45)$$

### 16.2.7.2 Interacting Multiple-Model Estimation Algorithm (Fig. 16.1)

- *Interaction* Mixing of the previous cycle mode-conditioned state estimates and covariance, using the mixing probabilities, to initialize the current cycle of each mode-conditioned filter



**Figure 16.1** The IMM estimation algorithm: one cycle ( $M_j(k) \triangleq \{M(k) = M_j\}$ ).

- *Mode-Conditioned Filtering* Calculation of the state estimates and covariances conditioned on a mode being in effect, as well as the mode-likelihood function ( $r$  parallel filters)
- *Probability Evaluation* Computation of the mixing and the updated mode probabilities
- *Overall State Estimate and Covariance* (for Output Only) Combination of the latest mode-conditioned state estimates and covariances

The IMM estimation algorithm has a *modular structure*.

### *Steps of IMM Estimation Algorithm*

1. *Interaction* ( $j = 1, \dots, r$ ) Initial estimate and covariance for filter  $j$ :

$$\hat{x}^{0j}(k-1|k-1) = \sum_{i=1}^r \hat{x}^i(k-1|k-1) \mu_{i|j}(k-1|k-1), \quad (16.46)$$

$$\begin{aligned} P^{0j}(k-1|k-1) &= \sum_{i=1}^r \mu_{i|j}(k-1|k-1) \\ &\times \{P^i(k-1|k-1) + [\hat{x}^i(k-1|k-1) - \hat{x}^{0j}(k-1|k-1)] \\ &\cdot [\hat{x}^i(k-1|k-1) - \hat{x}^{0j}(k-1|k-1)]'\}. \end{aligned} \quad (16.47)$$

2. *Mode-Conditioned Filtering* ( $j = 1, \dots, r$ ) The KF matched to  $M_j(k)$  (filter  $j$ ) uses  $z(k)$  to yield  $\hat{x}^j(k|k)$  and  $P^j(k|k)$ . Likelihood function corresponding to filter  $j$ :

$$\Lambda_j(k) = \mathcal{N}[z(k); \hat{x}^j(k|k-1), S_j(k)]. \quad (16.48)$$

3. *Probability Evaluations* Mixing probabilities ( $i, j = 1, \dots, r$ ):

$$\mu_{i|j}(k-1|k-1) = \frac{1}{\bar{c}_j} p_{ij} \mu_i(k-1), \quad (16.49)$$

$$\bar{c}_j \triangleq \sum_{i=1}^r p_{ij} \mu_i(k-1). \quad (16.50)$$

Update of the mode probabilities ( $j = 1, \dots, r$ ):

$$\mu_j(k) = \frac{1}{c} \Lambda_j(k) \bar{c}_j, \quad (16.51)$$

$$c \triangleq \sum_{j=1}^r \Lambda_j(k) \bar{c}_j. \quad (16.52)$$

4. Combination of Model-Conditioned Estimates and Covariances (for Output Purpose)

$$\hat{x}(k|k) = \sum_{j=1}^r \hat{x}^j(k|k) \mu_j(k), \quad (16.53)$$

$$P(k|k) = \sum_{j=1}^r \mu_j(k) \{ P^j(k|k) + [\hat{x}^j(k|k) - \hat{x}(k|k)][\hat{x}^j(k|k) - \hat{x}(k|k)]' \}. \quad (16.54)$$

Using multiple-model algorithms for benign nonmaneuvering targets might diminish the performance level of the tracker and increase the computational load. However, with higher target maneuverability, a multiple-model approach is needed. The decision to use a multiple-model estimator is made typically based on the maneuvering index [1], which quantifies the maneuverability of the target in terms of the process noise, sensor measurement noise, and sensor revisit interval. A study that compares the IMM estimator with the Kalman filter based on the maneuvering index could be found in [52].

### 16.2.8 Tracking with Multiple Sensors

It is assumed that there are  $N_S$  synchronized sensors. The measurement from sensor  $j$  at time  $k$  is

$$z(k, j) = H(k, j)x(k) + w(k, j), \quad j = 1, \dots, N_S. \quad (16.55)$$

The measurement noise sequences are zero mean, white, independent of the process noise, and independent from sensor to sensor with covariances  $R(k, j)$ .

Two widely used techniques to incorporate multiple sensors are:

- *Sequential Updating* The updating is carried out with the measurement of one sensor at a time. Start the recursion from the predicted state and covariance denoted by

$$\hat{x}(k|k, 0) = \hat{x}(k|k-1), \quad (16.56)$$

$$P(k|k, 0) = P(k|k-1). \quad (16.57)$$

The updates with the measurements at time  $k$  are

$$\begin{aligned} \hat{x}(k|k, j) &= \hat{x}(k|k, j-1) + W(k, j)(z(k, j) - H(k, j)\hat{x}(k|k, j-1)), \\ j &= 1, \dots, N_S, \end{aligned} \quad (16.58)$$

$$P(k|k, j) = P(k|k, j-1) - W(k, j)S(k, j)W(k, j)', \quad j = 1, \dots, N_S, \quad (16.59)$$

where

$$S(k, j) = H(k, j)P(k|k, j-1)H(k, j)' + R(k, j), \quad (16.60)$$

$$W(k, j) = P(k|k, j-1)H(k, j)'S(k, j)^{-1}. \quad (16.61)$$

For linear measurements, the order of updating in the sequential procedure is immaterial. For nonlinear measurements, however, measurement from the most accurate sensor should be updated first so as to reduce subsequent linearization errors.

- *Parallel Updating* Measurements from each sensor are simultaneously stacked and simultaneously updated.

### 16.3 DATA ASSOCIATION

In the previous section, it is assumed that there is no measurement origin uncertainty. However, the crux of the multitarget problem is to carry out the association process for measurements whose origins are uncertain due to:

- Random false alarms in the detection process
- Clutter due to spurious reflectors or radiators near the target of interest
- Interfering targets
- Decoys and countermeasures

Furthermore, the probability of obtaining a measurement from a target—the target detection probability—is usually less than unity.

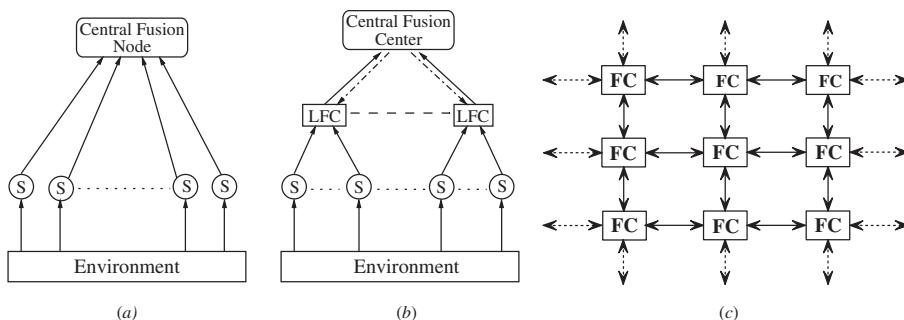
Data association problems may be categorized according to the pairs of information that are associated together. The possibilities are

- Measurement-to-measurement association—track formation
- Measurement-to-track association—track maintenance or updating
- Track-to-track association—track fusion (for distributed or decentralized tracking)

#### 16.3.1 Fusion Architectures

Three major types of architecture, namely, centralized, distributed, and decentralized, are commonly used in multisensor–multitarget tracking applications [23, 53, 54].

- *Centralized Tracking* In the centralized architecture (Fig. 16.2a), several sensors are monitoring the region of interest to detect and track the targets therein. All



**Figure 16.2** Common fusion architectures: (a) centralized; (b) distributed; (c) decentralized.

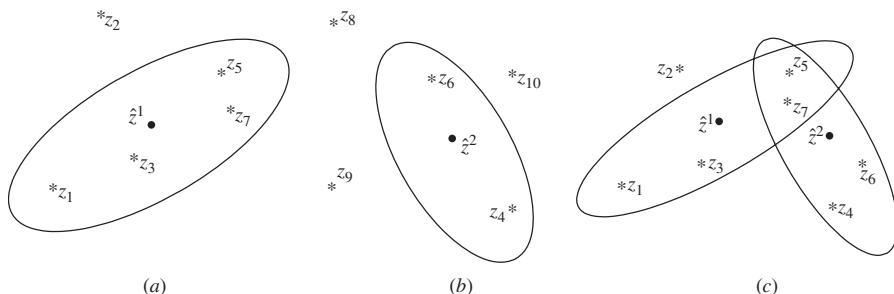
sensors generate measurements at each revisit time and report those measurements to a central fusion center (CFC). It in turn fuses all the acquired measurements and updates the tracks. This is the optimal architecture in terms of tracking performance. However, in a very large surveillance region with many sensors, this architecture may not be feasible because of limited resources, for example, communication bandwidth and computation power.

- *Distributed Tracking* In order to avoid the heavy communication and computational requirement of centralized fusion, distributed or hierarchical architecture, shown in Figure 16.2b, is used alternatively [54]. In this architecture, sensors are connected to local fusion centers (LFCs) and LFCs are in turn connected to a CFC. Each LFC updates its local tracks based on the measurements obtained from the local sensors and sends its tracks to CFC. Then, the CFC performs the track-to-track fusion and may send back the updated tracks to the LFCs, if feedback path is available.
- *Decentralized Tracking* When there is no CFC that can communicate with all the sensors or LFCs in a large surveillance region, neither centralized nor distributed tracking is possible. In such cases, an alternative called decentralized architecture, shown in Figure 16.2c, is used. Decentralized architecture is composed of multiple fusion centers and no CFC [54]. Here, each FC gets the measurements from one or more sensors that are connected to it and uses those measurements to update its tracks. In addition, tracks are also updated whenever an FC gets additional information from its neighbors. Note that even though many FCs are available, each FC can communicate only with its neighbors; the FCs within the communication distance every few measurement time steps.

### 16.3.2 Measurement-to-Track Association

A multidimensional gate is set up in the measurement space around the predicted measurement in order to avoid searching for the measurement from the target of interest in the entire measurement space. A measurement in the gate, while not guaranteed to have originated from the target the gate pertains to, is a valid association candidate, thus the name, validation region, or association region. If there is more than one measurement in the gate, this leads to an association uncertainty.

Figures 16.3a and 16.3b illustrate the gating for two well-separated and closely spaced targets, respectively. In the figures, • indicates the expected measurement and the \* indicates the received measurement.



**Figure 16.3** Validation regions: (a) well-separated targets; (b) closely spaced targets.

If the true measurement conditioned on the past is normally (Gaussian) distributed with its probability density function given by

$$p(z(k+1)|Z^k) = \mathcal{N}[z(k+1); \hat{z}(k+1|k), S(k+1)], \quad (16.62)$$

then the true measurement will be in the following region:

$$\mathcal{V}(k+1, \gamma) = \{z : [z - \hat{z}(k+1|k)]' S(k+1)^{-1} [z - \hat{z}(k+1|k)] < \gamma\} \quad (16.63)$$

with probability determined by the gate threshold  $\gamma$ . The region defined by (16.63) is called gate or validation region.

Some well-known approaches for data association in the presence of *well-separated targets*, where no measurement origin uncertainties exist, are discussed below.

**Nearest Neighbor (NN)** This is the simplest possible approach and uses the measurement nearest to the predicted measurement assuming it to be the correct one. The nearest measurement to the predicted measurement is determined according to the distance measure (norm of the innovation squared),

$$D(z) = [z - \hat{z}(k+1|k)]' S(k+1)^{-1} [z - \hat{z}(k+1|k)]. \quad (16.64)$$

**Strongest Neighbor (SN)** Select the strongest measurement (in terms of signal intensity) among the validated ones—this assumes that signal intensity information is available.

**Probabilistic Data Association (PDA)** This is a Bayesian approach that probabilistically associates all the validated measurements to the target of interest. The state update equation of the PDA filter is

$$\hat{x}(k|k) = \hat{x}(k|k-1) + W(k)v(k), \quad (16.65)$$

where

$$v(k) = \sum_{i=1}^{m(k)} \beta_i(k) v_i(k), \quad (16.66)$$

$$v_i(k) = (z_i(k) - \hat{z}(k|k-1)), \quad (16.67)$$

and  $m(k)$  is the number of validated measurements and

$$\beta_i(k) \triangleq \{\theta_i(k)|Z^k\} \quad (16.68)$$

is the conditional probability of the event that the  $i$ th validated measurement is correct.

The covariance associated with the updated state is

$$P(k|k) = \beta_0(k)P(k|k-1) + [1 - \beta_0(k)]P^c(k|k) + \tilde{P}(k), \quad (16.69)$$

where  $\beta_0(k)$  is the conditional probability of the event that none of the measurements is correct and the covariance of the state updated with the correct measurement is

$$P^c(k|k) = P(k|k-1) - W(k)S(k)W(k)', \quad (16.70)$$

and the spread of the innovations term is

$$\tilde{P}(k) \triangleq W(k) \left[ \sum_{i=1}^{m(k)} \beta_i(k) v_i(k) v_i(k)' - v(k) v(k)' \right] W(k)'. \quad (16.71)$$

Probabilistic data association is a common technique used for data association in conjunction with the Kalman filter or the extended Kalman filter. In most particle filtering algorithms, the nearest neighbor is used since it requires less computation.

The association of measurements in a multitarget environment with *closely spaced targets* must be done while simultaneously considering all the targets. The well-known approaches for closely spaced targets are discussed next.

**Joint Probabilistic Data Association (JPDA)** Extension of the PDA method: For a known number of targets it evaluates the measurement-to-target association probabilities (for the latest set of measurements) and combines them into the corresponding state estimates.

The steps of JPDA algorithms are:

A validation matrix that indicates all the possible sources of each measurement is set up.

From this validation matrix all the feasible joint association events are obtained according to the rules:

- One source for each measurement
- One measurement (or none) from each target

The probabilities of these joint events are evaluated according to the following assumptions:

- Target-originated measurements are Gaussian distributed around the predicted location of the corresponding target's measurement.
- False measurements are uniformly distributed in the surveillance region.
- The number of false measurements is distributed according to the Poisson prior (parametric JPDA) and the diffuse prior (nonparametric JPDA).

Marginal (individual measurement-to-target) association probabilities are obtained from the joint association probabilities.

The target states are estimated by separate (uncoupled) PDA filters using these marginal probabilities.

**Multiple Hypothesis Tracker (MHT)** This approach considers the association of sequences of measurements and evaluates the probabilities of all the association hypotheses.

This leads to a complexity that increases exponentially with time, and appropriate techniques have to be used to limit the number of hypotheses under consideration:

- Clustering to reduce the combinatorial complexity
- Pruning of low-probability hypotheses
- Merging of similar hypotheses

For each sequence of measurements—which is a hypothesized track—a standard KF yields the corresponding state estimate and covariance.

**Two-Dimensional Assignment** The fundamental idea behind two-dimensional (2D) assignment is that the measurements from the scan list  $\mathcal{M}(k)$  are matched (or deemed to have come from) the tracks in list  $\mathcal{T}(k-1)$  by formulating the matching as a constrained global optimization problem. The optimization is carried out to minimize the “cost” of associating (or not associating) the measurements to tracks.

To present the 2D assignment, define a binary assignment variable  $a(k, m, n)$  such that

$$a(k, m, n) = \begin{cases} 1, & \text{measurement } \mathbf{z}_m(t_{m_k}) \text{ is assigned to track } \mathcal{T}^n(k-1), \\ 0, & \text{otherwise,} \end{cases} \quad (16.72)$$

where  $t_{m_k}$  is the time stamp of the  $m$ th measurement from scan or frame  $k$ .

A set of complete assignments, which consists of the associations of all the measurements in  $\mathcal{M}(k)$  and the tracks in  $\mathcal{T}(k-1)$ , is denoted by  $\mathbf{a}(k)$ , that is,

$$\mathbf{a}(k) = \{a(k, m, n); m = 0, 1, \dots, M(k); n = 0, 1, \dots, N(k-1)\}, \quad (16.73)$$

where  $M(k)$  and  $N(k-1)$  are the cardinalities of the measurement and track sets, respectively. The indices  $m = 0$  and  $n = 0$  correspond to the nonexistent (or “dummy”) measurement and track. The dummy notation is used to formulate the assignment problem in a uniform manner, where the nonassociation possibilities are also considered, making it computer solvable.

The objective of the assignment is to find the optimal assignment  $\mathbf{a}^*(k)$ , which minimizes the global cost of association:

$$C(k|\mathbf{a}(k)) = \sum_{m=0}^{M(k)} \sum_{n=0}^{N(k-1)} a(k, m, n) c(k, m, n), \quad (16.74)$$

where  $c(k, m, n)$  is the cost of the assignment  $a(k, m, n)$ . That is,

$$\mathbf{a}^*(k) = \arg \min_{\mathbf{a}(k)} C(k|\mathbf{a}(k)). \quad (16.75)$$

The costs  $c(k, m, n)$  are the negative of the logarithm of the dimensionless likelihood ratio of the measurement-to-track associations, namely,

$$c(k, m, n) = -\ln \Lambda(k, m, n), \quad (16.76)$$

where

$$\Lambda(k, m, n) = \begin{cases} P_D p[v_m^n(t_{m_k})] / \lambda_e, & m > 0, n > 0, \\ 1, & m > 0, n = 0, \\ (1 - P_D), & m = 0, n > 0 \end{cases} \quad (16.77)$$

are the following likelihood ratios:

1. Measurement  $m > 0$  came from track  $n$  for  $n > 0$  (with the association-likelihood function being the probability density function of the corresponding innovation,  $p[v_m^n(k)]$  versus from an extraneous source whose spatial density is  $\lambda_e$ ).
2. Measurement  $m > 0$  came from none of the tracks (i.e., from the dummy track  $n = 0$ ) versus from an extraneous source (which is the same thing and, thus, the ratio is unity).
3. The measurement from track  $n$  is not in  $\mathcal{M}(k)$ , that is, track  $n$  is associated with the dummy measurement—the cost of not associating any measurement to a track amounts to the miss probability  $1 - P_D$ , where the nominal target detection probability is denoted by  $P_D$ .

The 2D assignment is subject to the following constraints:

*Validation* A measurement is assigned only to one of the tracks that validated it.

*One-to-One Constraint* Each track is assigned at most one measurement. The only exception is the dummy track ( $n = 0$ ), which can be associated with any number of measurements. Similarly, a measurement is assigned to at most one track. The dummy measurement ( $m = 0$ ) can be assigned to multiple tracks.

*Nonempty Association* The association cannot be empty, that is, the dummy measurement cannot be assigned to the dummy track. The candidate associations, subject to the above constraints, are given to the modified auction algorithm [37, 55] along with the corresponding association costs.

*Multidimensional (S-D) Assignments* In 2D assignment only the latest scan is used, and information about target evolution through multiple scans is lost. Also it is not possible to change an association later in light of subsequent measurements. A data association algorithm may perform better when a number of past scans are utilized. This corresponds to multidimensional assignment for data association. In S-D assignment (which is the optimization-based MHT with a sliding window) the latest  $S - 1$  scans of measurements are associated with the established track list (from time  $k - S + 1$ , where  $k$  is the current time, i.e., with a sliding window of time depth  $S - 1$ ) in order to update the tracks.

Similarly, to the 2D assignment, define a binary assignment variable  $a(k, \{m_s\}_{s=k-S+2}^k, n)$  such that

$$a(k, \{m_s\}_{s=k-S+2}^k, n) = \begin{cases} 1, & \text{measurements } \mathbf{z}_{m_{k-S+2}}(t_{m_{k-S+2}}), \dots, \mathbf{z}_{m_k}(t_{m_k}) \text{ are} \\ & \text{assigned to track } \mathcal{T}^n(k - S + 1), \\ 0, & \text{otherwise,} \end{cases} \quad (16.78)$$

which is the general version of (16.72). The cost associated with (16.78) is denoted as

$$c(k, \{m_s\}_{s=k-S+2}^k, n) = -\ln \Lambda(k, \{m_s\}_{s=k-S+2}^k, n), \quad (16.79)$$

and  $\Lambda(k, \{m_s\}_{s=k-S+2}^k, n)$  is the likelihood ratio that the  $S - 1$ -tuple of measurements given by  $\mathbf{z}_{m_{k-S+2}}(t_{m_{k-S+2}}), \dots, \mathbf{z}_{m_k}(t_{m_k})$  originated from the target represented by track  $\mathcal{T}^n(k - S + 1)$  versus being extraneous.

The objective of the  $S$ -D assignment is to find the  $S$ -tuples of measurement-to-track associations  $a(k, \{m_s\}_{s=1}^{S-1}, n)$ , which minimize the global cost of association given by

$$C(k|\mathbf{a}) = \sum_{n=0}^{N(k-S+1)} \sum_{m_{k-S+2}=0}^{M(k-S+2)} \sum_{m_{k-S+3}=0}^{M(k-S+3)} \cdots \sum_{m_k=0}^{M(k)} a(k, \{m_s\}_{s=k-S+2}^k, n) \\ c(k, \{m_s\}_{s=k-S+2}^k, n), \quad (16.80)$$

where  $M(k)$  is the number of measurements in scan  $k$  and  $\mathbf{a}$  is the complete set of associations analogous to that defined in (16.73) for the 2D assignment. The association likelihoods are given by

$$\Lambda(k, \{m_s\}_{s=k-S+2}^k, n) = \begin{cases} \prod_{s=k-S+2}^k [1 - P_D]^{1-u(m_s)} [P_D p[v_{m_s}^n(s)] \lambda_e]^{u(m_s)}, & n > 0, \\ 1, & n = 0, \end{cases} \quad (16.81)$$

where  $u(m)$  is a binary function such that

$$u(m) = \begin{cases} 1, & m > 0, \\ 0, & m = 0, \end{cases} \quad (16.82)$$

and  $p[v_{m_s}^n(s)]$  is the filter-calculated innovation pdf if the (kinematic) measurement  $\mathbf{z}_{m_s(t_{m_s})}$  is associated with track  $\mathcal{T}^n(k-S+1)$  continued with the (kinematic) measurements  $\mathbf{z}_{m_{k-S+2}}(t_{m_{k-S+2}}), \dots, \mathbf{z}_{m_{s-1}}(t_{m_{s-1}})$ .

The association costs are given to the generalized  $S$ -D assignment algorithm, which uses Lagrangian relaxation, as described in [40] and [37] to solve the assignment problem in quasi-polynomial time. The feasibility constraints are similar to those from the 2D assignment.

### 16.3.3 Measurement-to-Measurement Association

All the unassociated measurements in the measurement-to-track associations are used to form new tracks. Track formation in the presence of measurement uncertainty requires measurement-to-measurement association.

**16.3.3.1 Track Formation with Single Sensor** One commonly used approach is a logic-based one that uses gates and requires a certain sequence of detections in these gates. If the requirement is satisfied, then the measurement sequence is accepted as a valid track.

The following two-stage cascaded logic that assumes target position measurements is considered:

1. Every unassociated detection (measurement) is an “initiator”—it yields a tentative track.
2. At the sampling time (scan or frame) following the detection of an initiator, a gate is set up based on the assumed maximum and minimum target motion parameters as well as the measurement noise intensities.

This assumes that if the initiator is from a target, the measurement from it in the second scan (if detected) will fall inside the gate with nearly unity probability. Following a detection, this track becomes a preliminary track. If there is no detection, this track is dropped.

3. Since a preliminary track has two measurements, a linear or nonlinear filter can be initialized and used to set up a gate for the next (third) sampling time.
4. Starting from the third scan, a logic of  $m$  detections out of  $n$  scans (frames) is used for the subsequent gates.
5. If at the end (scan  $n + 2$  at the latest) the logic requirement is satisfied, the track becomes a confirmed track or an accepted track. Otherwise, it is dropped.

**16.3.3.2 Track Formation with Multiple Sensors** If  $S$  lists of measurements are obtained from  $S$  synchronous sensors, then the goal is to group the measurements that could have originated from the same (unknown) target. In one commonly used approach, each feasible  $S$ -tuple of measurement  $Z_{i_1 i_2 \dots i_S}$ , consisting of one measurement from each sensor, is assigned a cost [typically, a likelihood ratio similar to (16.79)] and then the set of  $S$ -tuples that minimizes the global cost is found. This optimization can be formulated as a multidimensional ( $S$ -D) assignment as described in Section 16.3.2.

The unknown target state, which is necessary to find the assignment cost, is replaced by its maximum-likelihood (ML) estimate:

$$X_u = \arg \max_X p(Z_{i_1 i_2 \dots i_S} | X). \quad (16.83)$$

Note that an  $S$ -tuple in the association needs to have a certain minimum number of measurements from a target in order for the state of the target to be observable.

#### 16.3.4 Track-to-Track Association

In a distributed or decentralized configuration, each fusion center has a number of tracks. The crucial question here is how to decide whether tracks from different fusion centers represent the same target—the problem of track-to-track association.

**16.3.4.1 Association for Tracks with Dependent Errors** In this algorithm, the problem of associating tracks represented by their local estimates and covariances from  $S$  fusion centers is considered [30]. While different sensors have, typically, independent measurement errors, the local state estimation errors for the same target will be dependent due to the common process noise (or common prior). This dependence is characterized by the cross-covariances of the local estimation errors [23].

The cross-covariance recursion is given by

$$P^{ij}(k|k) = [I - W^i(k)H^i(k)] \left[ F(k-1)P^{ij}(k-1|k-1)F(k-1)' \right] \quad (16.84)$$

$$+ Q(k-1) \left[ I - W^j(k)H^j(k)' \right]. \quad (16.85)$$

This is a linear recursion and its initial condition is assuming the initial errors to be uncorrelated, that is,  $P^{ij}(0|0) = 0$  (no common prior). This is a reasonable

assumption in view of the fact that the initial estimates are usually based on the initial measurements, which were assumed to have independent errors.

Let us consider the assignment formulation for track-to-track association from  $S$  fusion centers. Assume fusion center  $S_i$  has a list of  $N_i$  tracks. Define the binary assignment variable  $\chi_{i_1 i_2 \dots i_S}$  as

$$\chi_{i_1 i_2 \dots i_S} = \begin{cases} 1, & \text{tracks } i_1, i_2, \dots, i_S \text{ are from the same target,} \\ 0, & \text{otherwise.} \end{cases} \quad (16.86)$$

A subset of indices  $\{i_1, i_2, \dots, i_S\}$  could be zero in the assignment variable, meaning that no track will be from the target in the corresponding list of the fusion centers.

The  $S$ -D assignment formulation finds the most likely hypothesis by solving the following constrained optimization:

$$\min_{\chi_{i_1 i_2 \dots i_S}} \sum_{i_1=0}^{N_1} \sum_{i_2=0}^{N_2} \dots \sum_{i_S=0}^{N_S} c_{i_1 i_2 \dots i_S} \chi_{i_1 i_2 \dots i_S} \quad (16.87)$$

subject to the constraints

$$\sum_{i_2=0}^{N_2} \dots \sum_{i_S=0}^{N_S} \chi_{j i_1 i_2 \dots i_S} = 1, \quad j = 1, 2, \dots, N_1, \quad (16.88)$$

$$\sum_{i_1=0}^{N_1} \sum_{i_3=0}^{N_3} \dots \sum_{i_S=0}^{N_S} \chi_{i_1 j i_3 \dots i_S} = 1, \quad j = 1, 2, \dots, N_2, \quad (16.89)$$

$$\vdots \quad (16.90)$$

$$\sum_{i_1=0}^{N_1} \dots \sum_{i_{S-1}=0}^{N_{S-1}} \chi_{i_1 i_2 \dots i_{S-1} j} = 1, \quad j = 1, 2, \dots, N_S, \quad (16.91)$$

and

$$\chi_{i_1 i_2 \dots i_S} \in \{0, 1\}, \quad i_1 = 1, \dots, N_1, \quad i_2 = 1, \dots, N_2, \quad i_S = 1, \dots, N_S. \quad (16.92)$$

In (16.87) the assignment cost is

$$c_{i_1 i_2 \dots i_S} = -\log \lambda_{i_1 i_2 \dots i_S}, \quad (16.93)$$

where  $\lambda_{i_1 i_2 \dots i_S}$  is the likelihood ratio of the track association hypothesis versus all tracks from different targets, and given by

$$\lambda_{i_1 i_2 \dots i_S} = V^{M-1} \mathcal{N}[\hat{x}_{S_i}; 0, P_{S_i}] \left[ \prod_{s \in S_i} P_{D_s} \right] \left[ \prod_{s \in S_i} 1 - P_{D_s} \right], \quad (16.94)$$

where  $1/V$  is the diffuse pdf of track density,  $\mathcal{S}_i = \{j | i_j > 0, j = 1 : \dots, S\} = \{s_1, \dots, s_M\}$ ,  $M$  is the number of elements in  $\mathcal{S}_i$ , and

$$\hat{x}_{\mathcal{S}_i} = \left[ \hat{x}_{s_2}^{i_{s_2}} - \hat{x}_{s_1}^{i_{s_1}}, \dots, \hat{x}_{s_M}^{i_{s_M}} - \hat{x}_{s_1}^{i_{s_1}} \right], \quad (16.95)$$

and  $P_{\mathcal{S}_i}$  is its covariance matrix with diagonal blocks:

$$(P_{\mathcal{S}_i})_{j-1, j-1} = P_{s_1}^{i_{s_1}} + P_{s_j}^{i_{s_j}} - P_{s_1, s_j}^{i_{s_1} i_{s_j}} - \left( P_{s_1, s_j}^{i_{s_1} i_{s_j}} \right)', \quad j = 2, \dots, M, \quad (16.96)$$

and off-diagonal blocks

$$(P_{\mathcal{S}_i})_{j-1, g-1} = P_{s_1}^{i_{s_1}} - P_{s_1, s_j}^{i_{s_1} i_{s_j}} - \left( P_{s_1, s_j}^{i_{s_1} i_{s_j}} \right)' + P_{s_j, s_g}^{i_{s_j} i_{s_g}} \quad j, g = 2, \dots, M. \quad (16.97)$$

The ML estimate of the track states obtained by fusing the set of tracks  $\{i_{s_1}, \dots, i_{s_M}\}$  is given by

$$x_{\mathcal{S}_i}^{\text{fused}} = \left( E' \overline{P}_{\mathcal{S}_i}^{-1} E \right)^{-1} E' \overline{P}_{\mathcal{S}_i} \bar{x}_{\mathcal{S}_i}, \quad (16.98)$$

where  $E = [I_{n_x} \ I_{n_x} \ \dots, \ I_{n_x}]$  is  $(M \times n_x) \times n_x$  matrix and  $n_x$  is the dimension of the state vector. Also,

$$\bar{x}_{\mathcal{S}_i} = \left[ \hat{x}_{s_1}^{i_{s_1}}, \dots, \hat{x}_{s_M}^{i_{s_M}} \right], \quad (16.99)$$

$$\overline{P}_{\mathcal{S}_i} = \begin{bmatrix} P_{s_1}^{i_{s_1}} & P_{s_1, s_2}^{i_{s_1} i_{s_2}} & \dots & P_{s_1, s_M}^{i_{s_1} i_{s_M}} \\ P_{s_2, s_1}^{i_{s_2} i_{s_1}} & P_{s_2}^{i_{s_2}} & \dots & P_{s_2, s_M}^{i_{s_2} i_{s_M}} \\ \vdots & \vdots & \ddots & \vdots \\ P_{s_M, s_1}^{i_{s_2} i_{s_1}} & P_{s_M, s_2}^{i_{s_M} i_{s_2}} & \dots & P_{s_M}^{i_{s_M}} \end{bmatrix}. \quad (16.100)$$

The covariance matrix of the fused track is given by

$$P_{\mathcal{S}_i}^{\text{fused}} = \left( E' \overline{P}_{\mathcal{S}_i}^{-1} E \right)^{-1}. \quad (16.101)$$

**16.3.4.2 Tracklet Fusion** In this algorithm it is assumed that tracklets are transmitted between the fusion centers. A tracklet is a track computed such that its errors are not cross-correlated with the errors of any other data in the system for the same target [56].<sup>1</sup> It is equivalent to a track for a target based only on the most recent measurements since the data from the tracker was last transmitted for that target. The calculation of tracklet is explained below.

Suppose the last transmission was performed at time  $k-l$  and the next transmission is at time  $k$ . The state estimates and covariances of a target just after the last transmission and just before the next transmission are  $\hat{x}^j(k-l|k-l)$ ,  $P^j(k-l|k-l)$ , and  $\hat{x}^j(k|k)$ ,  $P^j(k|k)$ . Then the information-filter-based tracklet is given by [57]

$$y^j(k) = P^j(k|k)^{-1} \hat{x}^j(k|k) - P^j(k|k-l)^{-1} \hat{x}^j(k|k-l), \quad (16.102)$$

$$Y^j(k) = P^j(k|k)^{-1} - P^j(k|k-l)^{-1}, \quad (16.103)$$

<sup>1</sup>The technique in [56] is only an approximation.

where

$$\hat{x}^j(k|k-l) = F(k, k-l) \hat{x}^j(k-l|k-l), \quad (16.104)$$

$$P^j(k|k-l) = F(k, k-l) P^j(k-l|k-l) F(k, k-l)' + Q(k, k-l). \quad (16.105)$$

Since the received tracklets are assumed to be independent of each other and independent of the local tracks, tracklets can be associated with the list of local tracks using an *S-D* association technique by considering tracklets as the measurements—measurement  $Y^j(k)^{-1} y^j(k)$  with covariance  $Y^j(k)^{-1}$  and  $H(k) = I$ .

The fused local tracks can be obtained from a set of tracklets and the track they are associated with using:

$$(P^{\text{fused}}(k|k))^{-1} = P(k|k)^{-1} + \sum_{j=1}^M Y^j(k), \quad (16.106)$$

$$(P^{\text{fused}}(k|k))^{-1} \hat{x}^{\text{fused}}(k|k) = P(k|k)^{-1} \hat{x}(k|k) + \sum_{j=1}^M y^j(k). \quad (16.107)$$

## 16.4 OUT-OF-SEQUENCE MEASUREMENTS

In multisensor tracking systems that operate in a centralized manner, there are usually different time delays in transmitting the scans or frames from the various sensors to the center. This can lead to situations where measurements from the same target arrive out of sequence [43–46]. Such “out-of-sequence” measurement (OOSM) arrivals can occur even in the absence of scan/frame communication time delays.

The problem is as follows: At time  $t = t_k$ , one has

$$\hat{x}(k|k) \triangleq E[x(k)|Z^k], \quad P(k|k) \triangleq \text{cov}[x(k)|Z^k]. \quad (16.108)$$

Subsequently, the earlier measurement from time  $\tau$ , denoted from now on with discrete-time notation as  $\kappa$ ,

$$z(\kappa) \triangleq z(\tau) = H(\kappa)x(\kappa) + w(\kappa) \quad (16.109)$$

arrives after the state estimate (16.108) has been calculated. The earlier measurement in (16.109) is then used to update this estimate as

$$\hat{x}(k|\kappa) = E[x(k)|Z^\kappa], \quad P(k|\kappa) = \text{cov}[x(k)|Z^\kappa], \quad (16.110)$$

where

$$Z^\kappa \triangleq \{Z^\kappa, z(\kappa)\}. \quad (16.111)$$

### 16.4.1 One-Step-Lag OOSM

The OOSM is assumed to be within the last sampling interval. For the purpose of simplicity, this is called the one-step-lag problem, even though the lag is really a fraction of a time step.

An optimal approach consists of the following steps [43]:

- Retrodict the state from the latest time  $k$  to the earlier time  $\kappa$  accounting in full for the process noise  $v(k, \kappa)$ , which has a nonzero conditional mean given  $Z^k$ :

$$\hat{x}(\kappa|k) = F(\kappa, k) [\hat{x}(k|k) - Q(k, \kappa)H(k)'S(k)^{-1}v(k)], \quad (16.112)$$

where  $F(\kappa, k) = F(k, \kappa)^{-1}$  is the backward transition matrix.

- Evaluate the corresponding covariance;

$$P(\kappa|k) = F(\kappa, k)[P(k|k) + P_{vv}(k, \kappa|k) - P_{xv}(k, \kappa|k) - P_{xv}(k, \kappa|k)']F(\kappa, k)' \quad (16.113)$$

with

$$P_{vv}(k, \kappa|k) = Q(k, \kappa) - Q(k, \kappa)H(k)'S(k)^{-1}H(k)Q(k, \kappa), \quad (16.114)$$

$$P_{xv}(k, \kappa|k) = Q(k, \kappa) - P(k|k-1)H(k)'S(k)^{-1}H(k)Q(k, \kappa). \quad (16.115)$$

- Calculate the filter gain for updating the state  $x(k)$  with the earlier measurement  $z(\kappa)$ :

$$W(k, \kappa) = P_{xz}(k, \kappa|k)S(\kappa)^{-1} \quad (16.116)$$

with

$$P_{xz}(k, \kappa|k) = [P(k|k) - P_{xv}(k, \kappa|k)]F(\kappa, k)'H(k)', \quad (16.117)$$

$$S(\kappa) = H(\kappa)P(\kappa|k)H(\kappa)' + R(\kappa). \quad (16.118)$$

- Update the state estimate  $\hat{x}(k|k)$  to  $\hat{x}(k|\kappa)$  and calculate the corresponding covariance:

$$\hat{x}(k|\kappa) = \hat{x}(k|k) + W(k, \kappa)[z(\kappa) - H(\kappa)\hat{x}(\kappa|k)], \quad (16.119)$$

$$P(k|\kappa) = P(k|k) - P_{xz}(k, \kappa|k)S(\kappa)^{-1}P_{xz}(k, \kappa|k)'. \quad (16.120)$$

### 16.4.2 Multistep-Lag OOSM

The time  $\tau$ , at which the OOSM was made, is assumed to be such that  $t_{k-l} < \tau < t_{k-l+1}$  [44, 45].

**16.4.2.1 Approach 1: Equivalent Measurement-Based Algorithm** The approach that will allow to solve the  $l$ -step-lag problem as a one-step-lag problem is to define an equivalent measurement at time  $k$  that replaces all the measurements:

$$Z_{k-l+1}^k = \{z(k-l+1), \dots, z(k)\} \quad (16.121)$$

in the sense to be defined below. Then, the OOSM falls in the interval  $T_l = [t_{k-l}, t_k]$  during which the only measurement is the latest one, at  $t_k$ . In this manner, the OOSM with  $l$ -step lag becomes an OOSM with one-step lag.

The equivalent measurement at  $k$  is defined as

$$z^*(k) = H^*(k)x(k) + w^*(k) \quad (16.122)$$

with the standard assumptions about the noise as zero mean with covariance  $R^*(k)$ .

The procedure to determine  $R^*(k)$  is by ensuring that the equivalent update of the prediction from the “last” time, which is now  $t_{k-l}$ , yields the covariance as  $P(k|k)$ . The covariance update for the equivalent measurement is (in the information matrix form)

$$P^*(k|k)^{-1} = P(k|k)^{-1} = p(k|k-1)^{-1} + H^*(k)' R^*(k)^{-1} H^*(k). \quad (16.123)$$

One can conveniently choose  $H^*(k) = I$ , which yields

$$R^*(k)^{-1} = P(k|k)^{-1} - p(k|k-1)^{-1}. \quad (16.124)$$

The filter gain for the equivalent measurement is then

$$W^*(k) = P(k|k) R^*(k)^{-1}. \quad (16.125)$$

The equivalent innovation at  $k$  is

$$v^*(k) = W^*(k)^{-1} [\hat{x}(k|k) - \hat{x}(k|k-l)]. \quad (16.126)$$

The covariance of the equivalent innovation at  $k$  is

$$S^*(k) = p(k|k-1) + R^*(k). \quad (16.127)$$

Then, (16.112)–(16.120) is used to find the updated state and corresponding covariance, by replacing  $H(k)$  with  $I$ ,  $v(k)$  with  $v^*(k)$  and  $S(k)$  with  $S^*(k)$ .

**16.4.2.2 Approach 2: Smoothing-Based Algorithm** This approach generalize the one-step-lag OOSM algorithm to  $l$ -step lag using backward recursive smoother [45]. The steps of this approach are:

- Find the smoothed-state estimate  $\hat{x}(\kappa|k)$  and associated covariance  $P(\kappa|k)$ :

$$\hat{x}(\kappa|k) = \hat{x}(\kappa|k-l) + A(\kappa) (\hat{x}(k-l+1|k) - \hat{x}(k-l+1|k-l)), \quad (16.128)$$

$$P(\kappa|k) = P(\kappa|k-l) + A(\kappa) (P(k-l+1|k) - P(k-l+1|k-l)) A(\kappa)', \quad (16.129)$$

where

$$\hat{x}(\kappa|k-l) = F(\kappa, k-l) \hat{x}(k-l|k-l), \quad (16.130)$$

$$\hat{x}(k-l+1|k-l) = F(k-l+1, k-l) \hat{x}(k-l|k-l), \quad (16.131)$$

$$P(\kappa|k-l) = F(\kappa, k-l) P(k-l, k-l) F(\kappa, k-l)' + Q(\kappa, k-l), \quad (16.132)$$

$$P(k-l+1|k-l) = F(k-l+1, k-l) P(k-l, k-l) F(k-l+1, k-l)' + Q(k-l+1, k-l), \quad (16.133)$$

$$A(\kappa) = P(\kappa|k-l) F(k-l+1, \kappa) P(k-l+1|k-l)^{-1}, \quad (16.134)$$

and the smoother estimate  $\hat{x}(k-l+1|k)$  and covariance  $P(k-l+1|k)$  are calculated using the Rauch–Tung–Streibel (RTS) backward recursion:

$$\begin{aligned}\hat{x}(n, k) &= \hat{x}(n|n) + A(n)(\hat{x}(n+1|k) - \hat{x}(n+1|n)), \\ n &= k-1, \dots, k-l+1,\end{aligned}\quad (16.135)$$

$$\begin{aligned}P(n|k) &= P(n|n) + A(n)(P(n+1|k) - P(n+1|n))A(n)', \\ n &= k-1, \dots, k-l+1,\end{aligned}\quad (16.136)$$

where

$$A(n) = P(n|n)F(n+1, n)'P(n+1|n)^{-1}, \quad n = k-1, \dots, k-l+1. \quad (16.137)$$

- Calculate the cross-covariance  $P_{xz}(k, \kappa|k)$  between the state  $x(k)$  and the measurement  $z(\kappa)$ :

$$P_{xz}(k, \kappa|k) = P_{xx}(k, k-l+1|k)A(\kappa)'H(\kappa)'. \quad (16.138)$$

$P_{xx}(k, k-l+1|k)$  is calculated using the following backward recursion with the starting and ending value of  $n$  being  $k-1$  and  $k-l+1$ , respectively:

$$P_{xx}(k, n|k) = P_{xx}(k, n+1|k)A(n)', \quad n = k-1, \dots, k-l+1. \quad (16.139)$$

- Calculate the filter gain for updating the state  $x(k)$  with the earlier measurement  $z(\kappa)$  using (16.116).
- Updated the state estimate  $\hat{x}(k|\kappa)$  and the corresponding covariance  $P(k|\kappa)$  using (16.119) and (16.120), respectively.

## 16.5 RESULTS WITH REAL DATA [58]

### 16.5.1 Comparison of IMM Estimator with KF

In this section the performance of the IMM estimator with two linear models and the (single-model-based) KF are compared on a set of five radar data that span over a time interval of 7 min. The actual prediction errors and number of associations obtained using these two algorithms are tabulated. These results not only demonstrate the error reduction obtained with the IMM estimator but also indicate the magnitude of the actual errors in a typical ATC scenario. The root-mean-square (rms) prediction errors in the horizontal plane and vertical (altitude) are given in Tables 16.1 and 16.2, respectively. Both beacon and skin returns are included in comparing the performance of these two algorithms in the horizontal plane. However, in the altitude comparisons only beacon returns are included.

In Tables 16.1 and 16.2 the numbers of associated measurements in each sampling interval bin are listed for both schemes. Since the number of tracks formed (and the tracks themselves) is different for different estimators, the numbers of measurements in each bin are not the same for either estimators. The rows marked with the symbol  $\star$  do not contain a sufficient number; hence, one cannot make any statistically significant

**TABLE 16.1 rms Prediction Errors in Horizontal Plane per Sampling Interval Bin in Nautical Miles (nmi)**

Sampling Interval Bins (s)	Detections per Bin		rms Error (nmi)		% Reduction
	IMM	KF	IMM	KF	
[0,5)	3720	3692	0.36	0.38	5.6
[15,20)	222	213	0.43	0.61	28.6
[30,35)	22	14	1.45	1.13	★
[45,50)	57	11	0.83	3.80	★
[60,65)	13	1	0.87	0.87	★
[75,∞)	12	0	1.52	—	—

**TABLE 16.2 rms Prediction Errors in Altitude per Sampling Interval Bin in Feet (ft)**

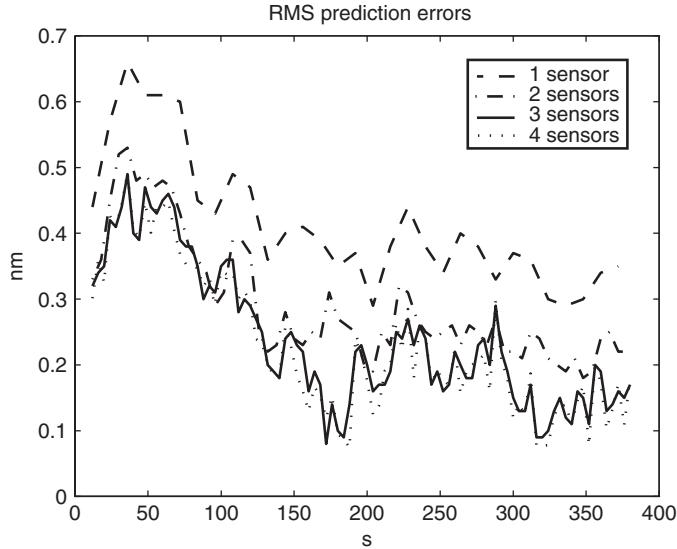
Sampling Bins (s)	Detections per Bin		RMS Error (ft)		% Reduction
	IMM	KF	IMM	KF	
[0,5)	3592	3586	34	34	0.0
[15,20)	201	193	106	115	7.8
[30,35)	12	12	147	516	★
[45,50)	13	11	376	420	★
[60,65)	0	1	—	1	—
[75,∞)	0	0	—	—	—

inference about the errors in these bins. Some uncommon situations of the measurements will be discussed later in this section. The bins with sufficient measurements indicate approximately 20–50% rms prediction error (innovation) reduction for the IMM design over the KF. Also, the IMM estimator associates more measurements than the KF.

### 16.5.2 Advantages of Fusing Multisensor Data

This section illustrates the advantages of multisensor tracking over single-sensor tracking in the sense of rms prediction error. Second, the computational complexity of the multisensor algorithm is also discussed.

In Figure 16.4, the comparison of the rms position prediction errors obtained using multisensor data and single-sensor data is presented. It is clear that the rms errors in multisensor tracking decrease as the number of sensors increases. The reason is straightforward: There are only a few measurements detected in the multisensor case that fill in the “gap” of two consecutive measurements detected by a single sensor. Therefore, the prediction time interval is shortened and the information based on which prediction made is more recent. Note that the improvement in the rms prediction error becomes smaller as the number of sensors increases. This is because the state update error decreases as the number of sensors increases. Then the measurement noise, which is approximately the same for single-sensor tracking or multisensor tracking, accounts for more of the rms prediction error.



**Figure 16.4** Comparison of rms errors in horizontal plane using different number of sensors.

The second advantage of fusing multisensor data is that multisensor tracking has *less* computation per scan than single-sensor tracking. Since the calculation of velocity gate (coarse gating) is computationally negligible, the computational complexity is primarily determined by the time spent fine-gating (validation [23]), which is significantly costlier than velocity gating. Table 16.3 presents the comparison of computation for fine gating (validation) between multisensor tracking and single-sensor tracking. As discussed in [59], the validation takes up about 95% of the total computations.

In single-sensor tracking, the time interval  $T$  and state estimation error  $P$  are larger than in multisensor tracking. Consequently, the velocity gate size is larger than in the multisensor case resulting in more candidate measurements to be passed into fine gating, especially in dense air traffic regions. This accounts for reduction in computation using multisensor tracking. Thus, the computations for a multisensor tracking are less than the total computations for single-sensor tracking.

**TABLE 16.3 Comparison of Computational Complexity in Multisensor and Single-Sensor Tracking for Measurements from Sensor  $R$**

Scan Number	Number of Fine-Gating Computation		Time of Fine-Gating Computation (s)	
	5 sensors	$R$	5 sensors	$R$
$R_2$	1779	1819	5.60	5.95
$R_5$	1601	1988	5.05	6.26
$R_8$	1799	2045	5.67	6.46
$R_{11}$	1672	1826	5.27	5.75
$R_{14}$	1883	2364	5.93	8.27

Note:  $R_i$  represents the  $i$ th scan from sensor  $R$ .

**TABLE 16.4 rms Prediction Errors in Horizontal Plane during Maneuvering Periods, per Sampling Interval Bin, in Nautical Miles (nmi)**

Sampling Interval Bins (s)	Detections per Bin		rms Error (nmi)		
	IMM-CT	IMM-L	IMM-CT	IMM-L	% Reduction
[0,5)	420	412	0.42	0.46	8.7
[15,20)	83	79	0.55	1.06	48.0
[30,35)	9	9	1.42	2.06	★
[45,50)	14	13	1.80	2.25	★
[60,65)	3	3	1.70	2.18	★
[75,∞)	1	0	1.81	—	—

### 16.5.3 Comparison of IMM-L and IMM-CT Estimators

In this section the performances of IMM-L (with linear models) and IMM-CT (with coordinated turn model) are compared on the five radar databases over a time interval of 15 min. All three approaches of IMM-CT considered (first-order EKF, second-order EKF, and Kastella's method) were found to have minor differences in rms position prediction errors and number of associations, although the first-order EKF is much simpler than the other two. Table 16.4 shows the comparison of IMM-L and IMM-CT in the horizontal plane based on the measurements of the maneuvering periods. The selected periods corresponds to the intervals for which the CT model probability exceeded 0.5. The bins with sufficient data points indicate approximately 10–50% rms prediction error reduction in the critical maneuvering periods for the IMM-CT over the IMM-L. Also, IMM-CT associates, to a slight degree, more maneuvering measurements during these periods than IMM-L. During the nonmaneuvering periods, these two designs have identical performance.

## 16.6 SUMMARY

In this chapter, various choices for algorithms to handle data association, state estimation, fusion, fusion architecture, measurement processing, and debiasing were discussed in detail. In addition, their quantitative and qualitative merits were discussed. Various combinations of these algorithms provide a complete tracking and fusion framework for multisensor networks with application to civilian as well as military problems. For example, the tracking and fusion techniques discussed here are applicable to fields like air traffic control, air/ground/maritime surveillance, mobile communication, transportation, video monitoring, and biomedical imaging/signal processing. Using a representative multisensor–multitarget tracking problem with real data, it was also shown that the algorithms are capable of processing large amounts of data and extracting all information available therein while remaining real time feasible. The benefits of multisensor fusion compared with single-sensor processing were quantified as well.

## REFERENCES

1. Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, New York: Wiley, 2001.

2. F. Daum, "Nonlinear filters: Beyond the Kalman filter," *IEEE Aerospace Electron. Syst. Mag.*, vol. 20, no. 8, part 2, pp. 57–69, Aug. 2005.
3. R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, vol. 82, pp. 34–45, Mar. 1960.
4. S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. SPIE Conf. on Signal Processing, Sensor Fusion and Target Recognition VI*, Vol. 3068, Orlando, FL, Apr. 1997, pp. 182–193.
5. S. Sarkka, "On unscented Kalman filtering for state estimation of continuous-time nonlinear systems," *IEEE Trans. Automatic Control*, vol. 52, no. 9, pp. 1631–1641, Sept. 2007.
6. R. Zhan and J. Wan, "Iterated unscented Kalman filter for passive target tracking," *IEEE Trans. Aerospace Electron. Syst.*, vol. 43, no. 3, pp. 1155–1163, July 2007.
7. M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
8. A. Doucet, N. de Freitas and N. Gordon, *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag, 2001.
9. N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. Radar Signal Process.*, vol. 140, no. 2, pp. 107–113, Apr. 1993.
10. B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House Publishers, 2004.
11. R. Mahler, *An Introduction to Multisensor-Multitarget Statistics and Its Application*, Technical Monograph, Lockheed Martin, 2000.
12. R. Mahler, "Multi-target moments and their application to multi-target tracking," in *Proc. of the Workshop on Estimation, Tracking and Fusion: A Tribute to Yaakov Bar-Shalom*, Monterey, CA, 2001, pp. 134–166.
13. R. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerospace Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.
14. R. Mahler, "Random set theory for target tracking and identification," In *Handbook of Multisensor Data Fusion*, D. L. Hall and J. Lindas (Eds.), CRC Press: Boca Raton, FL, 2002, chapter 14.
15. K. Punithakumar, T. Kirubarajan, and A. Sinha, "Multiple-model probability hypothesis density filter for tracking maneuvering targets," *IEEE Trans. Aerospace Electron. Syst.*, vol. 44, no. 1, pp. 87–98, Jan. 2008.
16. H. Sidenbladh, "Multi-target particle filtering for the probability hypothesis density," in *Proc. 6th International Conf. of Information Fusion*, Vol. 2, July 2003, pp. 800–806.
17. B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *Proc. 6th International Conf. of Information Fusion*, Vol. 2, July 2003, pp. 792–799.
18. G. A. Ackerson and K. S. Fu, "On state estimation in switching environments," *IEEE Trans. Automatic Control*, vol. 15, no. 1, pp. 10–17, Feb. 1970.
19. W. D. Blair, G. A. Watson, T. Kirubarajan, and Y. Bar-Shalom, "Benchmark for radar resource allocation and tracking in the presence of ECM," *IEEE Trans. Aerospace Electron. Syst.*, vol. 34, no. 4, pp. 1097–1114, Oct. 1998.
20. H. A. P. Blom, "A sophisticated tracking algorithm for ATC surveillance data," in *Proc. International Radar Conf.*, Paris, May 1984, pp. 393–398.
21. H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Automatic Control*, vol. 33, no. 8, pp. 780–783, Aug. 1988.

22. C. B. Chang and M. Athans, "State estimation for discrete system with switching parameters," *IEEE Trans. Aerospace Electron. Syst.*, vol. 14, no. 3, pp. 418–425, May 1978.
23. Y. Bar-Shalom and X. R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS Publishing, 1995.
24. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.
25. D. Smith and S. Singh, "Approaches to multisensor data fusion in target tracking: A survey," *IEEE Trans. Knowledge Data Eng.*, vol. 18, no. 12, pp. 1696–1710, Dec. 2006.
26. T. Kirubarajan, Y. Bar-Shalom, and D. Lerro, "Bearings-only tracking of maneuvering targets using a batch-recursive estimator," *IEEE Trans. Aerospace Electron. Syst.*, vol. 37, no. 3, pp. 770–780, July 2001.
27. T. Sathyam, A. Sinha, and T. Kirubarajan, "Computationally efficient assignment-based algorithms for data association for tracking with angle-only sensors," in *Proc. SPIE Conf. on Signal and Data Processing of Small Targets*, Vol. 6699, San Diego, CA, Aug. 2007.
28. A. T. Alouani, J. E. Gray, and D. H. McCabe, "Theory of distributed estimation using multiple asynchronous sensors," *IEEE Trans. Aerospace Electron. Syst.*, vol. 41, no. 2, pp. 717–722, Apr. 2005.
29. Y. Bar-Shalom, "Dimensionless score function for multiple hypothesis tracking," *IEEE Trans. Aerospace Electron. Syst.*, vol. 43, no. 1, pp. 392–400, Jan. 2007.
30. Y. Bar-Shalom and H. Chen, "Multisensor track-to-track association for tracks with dependent errors," *J. Adv. Inform. Fusion*, vol. 1, no. 1, pp. 3–14, July 2006.
31. H. Chen, T. Kirubarajan, and Y. Bar-Shalom, "Performance limits of track-to-track fusion versus centralized estimation: Theory and application," *IEEE Trans. Aerospace Electron. Syst.*, vol. 39, no. 2, pp. 386–400, Apr. 2003.
32. K. C. Chang, R. K. Saha, and Y. Bar-Shalom, "On optimal track-to-track fusion," *IEEE Trans. Aerospace Electron. Syst.*, vol. 33, no. 4, pp. 1271–1276, Oct. 1997.
33. H. You and Z. Jingwei, "New track correlation algorithms in a multisensor data fusion system," *IEEE Trans. Aerospace Electron. Syst.*, vol. 42, no. 4, pp. 1359–1371, Oct. 2006.
34. T. Kirubarajan and Y. Bar-Shalom, "Probabilistic data association techniques for target tracking in clutter," *Proc. IEEE*, vol. 92, no. 3, pp. 536–557, Mar. 2004.
35. S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace Electron. Syst. Mag.*, vol. 19, no. 1, part 2, pp. 5–18, Jan. 2004.
36. Y. Bar-Shalom, T. Kirubarajan, and C. Gokberk, "Tracking with classification-aided multi-frame data association," *IEEE Trans. Aerospace Electron. Syst.*, vol. 41, no. 3, pp. 868–878, July 2005.
37. K. R. Pattipati, T. Kirubarajan, and R. L. Popp, "Survey of assignment techniques for multitarget tracking," in *Proc. of the Workshop on Estimation, Tracking, and Fusion: A Tribute to Yaakov Bar-Shalom*, Monterey, CA, May 2001.
38. A. Capponi and H. W. De Waard, "A mean track approach applied to the multidimensional assignment problem," *IEEE Trans. Aerospace Electron. Syst.*, vol. 43, no. 2, pp. 450–471, Apr. 2007.
39. M. R. Chummun, T. Kirubarajan, K. R. Pattipati, and Y. Bar-Shalom, "Fast data association using multidimensional assignment with clustering," *IEEE Trans. Aerospace Electron. Syst.*, vol. 37, no. 3, pp. 898–913, July 2001.
40. S. Deb, M. Yeddanapudi, K. R. Pattipati, and Y. Bar-Shalom, "A generalized  $S$ -dimensional assignment for multisensor-multitarget state estimation," *IEEE Trans. Aerospace Electron. Syst.*, vol. 33, no. 2, pp. 523–538, Apr. 1997.
41. T. Kirubarajan, H. Wang, Y. Bar-Shalom, and K. R. Pattipati, "Efficient multisensor fusion using multidimensional data association," *IEEE Trans. Aerospace Electron. Syst.*, vol. 37, no. 2, pp. 386–400, Apr. 2001.

42. L. Lin, Y. Bar-Shalom, and T. Kirubarajan, "New assignment-based data association for tracking move-stop-move targets," *IEEE Trans. Aerospace Electron. Syst.*, vol. 40, no. 2, pp. 714–725, Apr. 2004.
43. Y. Bar-Shalom, "Update with out-of-sequence measurements in tracking: Exact solution," *IEEE Trans. Aerospace Electron. Syst.*, vol. 38, no. 3, pp. 769–777, July 2002.
44. Y. Bar-Shalom, H. Chen, and M. Mallick, "One-step solution for the multistep out-of-sequence-measurement problem in tracking," *IEEE Trans. Aerospace Electron. Syst.*, vol. 40, no. 1, pp. 27–37, Jan. 2004.
45. M. Mallick and K. Zhang, "Optimal multiple-lag out-of-sequence measurement algorithm based on generalized smoothing framework," in *Proc. SPIE Conf. on Signal and Data Processing of Small Targets*, San Diego, CA, Aug. 2005.
46. K. Zhang, X. R. Li, and Y. Zhu, "Optimal update with out-of-sequence measurements," *IEEE Trans. Signal Process.*, vol. 53, no. 6, pp. 1992–2004, June 2005.
47. Y. Bar-Shalom, "Airborne GMTI radar position bias estimation using static-rotator targets of opportunity," *IEEE Trans. Aerospace Electron. Syst.*, vol. 37, no. 2, pp. 695–699, Apr. 2001.
48. K. Kastella, B. Yeary, T. Zadra, R. Brouillard, and E. Frangione, "Bias modeling and estimation for GMTI applications," in *Proc. 3rd International Conf. on Information Fusion*, Paris, France, July 2000.
49. X. Lin, Y. Bar-Shalom, and T. Kirubarajan, "Multisensor multitarget bias estimation for general asynchronous sensors," *IEEE Trans. Aerospace Electron. Syst.*, vol. 41, no. 3, pp. 899–921, July 2005.
50. C. Kreucher, K. Kastella, and A. O. Hero III, "Multitarget tracking using the joint multitarget probability density," *IEEE Trans. Aerospace Electron. Syst.*, vol. 41, no. 4, pp. 1396–1414, Oct. 2005.
51. Y. Bar-Shalom, S. Challa, and H. A. P. Blom, "IMM estimator versus optimal estimator for hybrid systems," *IEEE Trans. Aerospace Electron. Syst.*, vol. 41, no. 3, pp. 986–991, July 2005.
52. T. Kirubarajan and Y. Bar-Shalom, "Kalman filter versus IMM estimator: When do we need the latter?" *IEEE Trans. Aerospace Electron. Syst.*, vol. 39, no. 4, pp. 1452–1457, Oct. 2003.
53. M. E. Liggins, C. Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," *Proc. IEEE*, vol. 85, no. 1, pp. 95–107, Jan. 1997.
54. N. Xiong and P. Svensson, "Multi-sensor management for information fusion: Issues and approaches," *Information Fusion*, vol. 3, no. 1, pp. 163–186, June 2002.
55. K. R. Pattipati, S. Deb, Y. Bar-Shalom, and R. B. Washburn, "A new relaxation algorithm and passive sensor data association," *IEEE Trans. Automatic Control*, vol. 37, no. 2, pp. 198–213, Feb. 1992.
56. O. E. Drummond, "A hybrid sensor fusion algorithm architecture and tracklets," in *Proc. SPIE Conf. on Signal and Data Processing of Small Targets*, Vol. 3163, San Diego, CA, July 1997, pp. 485–502.
57. O. E. Drummond, "Track and tracklet fusion filtering using data from distributed sensors," in *Proc. Estimation, Tracking and Fusion: A Tribute to Yaakov Bar-Shalom*, Monterey, CA, May 2001, pp. 167–186.
58. H. Wang, T. Kirubarajan, and Y. Bar-Shalom, "Precision large scale air traffic surveillance using an IMM estimator with assignment," *IEEE Trans. Aerospace Electron. Syst.*, vol. 35, no. 1, pp. 255–266, Jan. 1999.
59. R. L. Popp, K. R. Pattipati, and Y. Bar-Shalom, "Dynamically adaptable  $m$ -Best 2D assignment and multi-level parallelization," *IEEE Trans. Aerospace Electron. Syst.*, vol. 35, no. 4, pp. 1145–1160, Oct. 1999.

60. K. Panta, V. Ba-Ngu, and S. Singh, "Novel data association schemes for the probability hypothesis density filter," *IEEE Trans. Aerospace Electron. Syst.*, vol. 43, no. 2, pp. 556–570, Apr. 2007.
61. P. J. Shea, T. Zadra, D. Klamer, E. Frangione, R. Brouillard, and K. Kastella, "Precision tracking of ground targets," in *Proc. IEEE Aerospace Conference*, Big Sky, MT, Mar. 2000.
62. B. A. van Doorn and H. A. P. Blom, "Systematic error estimation in multisensor fusion systems," in *Proc. SPIE Conf. on Signal and Data Processing of Small Targets*, Vol. 1954, Orlando, FL, Apr. 1993.



---

## CHAPTER 17

---

# Distributed Algorithms in Sensor Networks

Usman A. Khan, Soummya Kar, and José M. F. Moura

Carnegie Mellon University, Department of Electrical and Computer Engineering,  
Pittsburgh, Pennsylvania

### 17.1 INTRODUCTION

Advances in integrated electronics, radio-frequency (RF) technologies and sensing devices have made it possible to deploy large numbers of cheap sensors for the purposes of monitoring, tracking, estimation, and control of complex large-scale dynamical systems through collaborative signal processing [1–3]. For example, consider a detection problem where the state of the environment is monitored “locally” by sensors; each sensor makes a measurement, based on which it may make a local decision—the current state of the sensor. A problem of interest is how to fuse these local decisions. An approach is to send these states to a fusion center where the optimal detector is formulated; this has been considered extensively in the literature since the early work in [4–6], the book by Varshney [7], and, more recently [8, 9]. This centralized or parallel architecture, which may have several advantages, is neither robust nor scalable when the size of the sensor network grows because of resource (bandwidth, power, etc.) constraints at the sensors and because it has a single point of failure. An alternative architecture for resource-constrained networks is a weblike topology with decentralized and distributed inference algorithms where each sensor updates its own local detector based on the state information of its neighboring sensors, iteratively, such that its state converges to the state of the optimal centralized or parallel detector. The distributed algorithms are, in general, iterative because the information flow is limited due to the sparse connectivity of these networks; the information is fused by the successive iterations. In this chapter, we consider such distributed *linear* algorithms in broad generality and provide a systematic study of classes of these algorithms.

Distributed algorithms have been studied widely in the literature. Early references include [5, 10–12], which provide a generic theory for developing and analyzing distributed and parallel algorithms. Recently, there has been renewed interest in the sensor network community on the so-called consensus problems and its various generalizations; see the numerous recent studies on the subject. Consensus can be broadly

interpreted as a distributed protocol in which the sensors collaborate to reach an agreeable state. Much research effort has been directed to the average consensus problem [13–16]. Agreement and consensus have been important problems in distributed computing [17, 18]. The problem of dynamic load balancing for distributed multiprocessors leads to an algorithm that is essentially consensus. Reference [19] gives spectral conditions on the weight matrix of the network graph for its convergence. In the multiagent and control literature, reference [13] develops a model for emergent behavior, schooling, and flocking described in [20, 21]. It presents conditions for alignment, that is, for all agents to agree to a value that lies in the convex hull of the initial conditions. Reference [14] identifies the algebraic connectivity of the underlying graph as controlling the convergence rate of the continuous-time average-consensus algorithm. For additional comments and a survey of consensus for multiagent coordination see [22, 23] and the references therein. Consensus and its generalizations [24–37] form the building block of a large number of distributed protocols, including flocking [38, 39], multivehicle coordination [23], and distributed estimation [40–44].

In this chapter, we explore consensus algorithms in a larger context where the sensors may converge to an arbitrary linear weighted combination of their initial states (not just their average). To capture in the same framework a broader class of algorithms, we consider the notion of anchors, which are nodes in the network that maintain a fixed state throughout the algorithm. The anchors may play the role of leaders of the system that drive the algorithm in a certain direction; see also [45]. The sensors iteratively update their states as a linear weighted combination of the states of the neighboring sensors. It is essential that the weights of the linear combination be computed using only local information that is available at the sensor itself and at the neighboring sensors. This requirement assures that the algorithms remain decentralized and distributed.

Consensus “averaging” algorithms do not require anchors as the goal is to converge to the average of the sensors’ initial state. Hence, we term the consensus-averaging algorithm as consensus in “zero dimensions.” On the other hand, when the goal is to converge to some function of the anchors’ states, forgetting the initial condition, we term such consensus algorithms as consensus in “ $n$  dimensions,” where  $n$  is the number of anchors.<sup>1</sup> These types of algorithms include sensor localization [46], leader–follower (type) algorithms [45], and Jacobi or Gauss–Seidel (type) algorithms for sensor networks [47]. In all these problems, the anchors are essential to provide a frame of reference to the algorithm.

The methodology we take in this chapter relies on basic principles. Using these principles, we establish a framework that unifies several classes of distributed iterative linear algorithms for sensor networks. We then specialize our framework to different scenarios, namely, averaging (in zero-dimension consensus), localization, leader–follower, and Jacobi (in multidimension consensus) and show that these are naturally extracted from our framework.

We summarize the rest of the chapter. In Section 17.2, we provide necessary preliminaries that are fundamental to the remaining of the chapter. Section 17.3 sketches the distributed detection problem that motivates distributed consensus algorithms. Section 17.4 describes the general form of consensus algorithms. Section 17.5 discusses consensus in zero dimensions (averaging algorithms), and Section 17.6 provides consensus in higher dimensions. Sections 17.7–17.9 give examples of

<sup>1</sup>The notion of *dimension* is elaborated in more detail later in the chapter.

consensus algorithms in higher dimensions. In particular, Section 17.7 describes the leader–follower algorithm, Section 17.8 describes a distributed localization algorithm, and Section 17.9 describes the Jacobi algorithm. Finally, we conclude in Section 17.10 with a brief description of a few advanced topics.

## 17.2 PRELIMINARIES

### 17.2.1 Spectral Graph Theory

In this section, we explore relevant concepts from graphs and spectral graph theory (see [48–50] for details.) Consider a network of  $N$  sensors where the sensors communicate according to a given network topology. The communication topology of this sensor network can be captured by a *directed* graph,  $\mathcal{G} = (V, E)$ , where the set of  $N$  sensors,  $V$ , denotes the vertices of the graph  $\mathcal{G}$ , and the edge set,  $E \subseteq V \times V$ , associated to  $\mathcal{G}$  is defined as

$$E = \{(l, j) | j \rightarrow l\}, \quad (17.1)$$

that is, the set  $E$  contains all pairs  $(l, j)$  such that the sensor  $j$  is connected to sensor  $l$ ; in other words, we can say that sensor  $j$  can send information to sensor  $l$ . Notice that since the graph  $\mathcal{G}$  is directed,  $(l, j) \in E \Rightarrow (j, l) \notin E$ .

The  $N \times N$  *adjacency* matrix,  $\mathbf{A} = \{a_{lj}\}$ , of a graph is defined as

$$a_{lj} = \begin{cases} 1, & \text{if } (l, j) \in E, \\ 0, & \text{otherwise.} \end{cases} \quad (17.2)$$

Because the graph is directed, the adjacency matrix may not be symmetric.

The set of *neighbors*,  $\mathcal{K}(l)$ , of sensor  $l$  collects all the sensors that are connected to  $l$  and sensor  $l$  itself, that is,

$$\mathcal{K}(l) = \{l\} \cup \{j | (l, j) \in E\}. \quad (17.3)$$

In other words, the neighbors of sensor  $l$  are all those sensors that can send information to sensor  $l$ .

The number of edges entering a vertex is called the *in-degree*,  $d_{\text{in}}$ , of the vertex.

The number of edges exiting a vertex is called the *out-degree*,  $d_{\text{out}}$ , of the vertex.

The  $N \times N$  *Laplacian* matrix,  $\mathbf{L} = \{L_{lj}\}$ , of a graph is defined as

$$L_{lj} = \begin{cases} -1, & j \in \mathcal{K}(l), j \neq l, \\ d_{\text{in}}, & j = l, \end{cases} \quad (17.4)$$

where  $|\mathcal{K}(l)|$  is the numbers of neighbors of sensor  $l$ .

A *path* of length  $K$  is a sequence of vertices,  $l_0, l_1, \dots, l_K$ , such that from each of these vertices there is an edge to the next vertex in the sequence, that is,  $(l_{i+1}, l_i) \in E, i = 0, \dots, K - 1$ .

A graph is *strongly connected* if there is a path between any pair of its vertices.

A directed graph is *balanced* if, for each of its vertices, the in-degree,  $d_{\text{in}}$ , equals the out-degree,  $d_{\text{out}}$ .

### 17.2.2 Matrix Theory

In this section, we review relevant definitions from the theory of nonnegative matrices (see [51] for details).

For a matrix  $\mathbf{P}$ ,  $\lambda_i(\mathbf{P})$  is its  $i$ th eigenvalue and  $\rho(\mathbf{P})$  is its spectral radius, that is,

$$\rho(\mathbf{P}) = \max_i |\lambda_i(\mathbf{P}^H \mathbf{P})|^{1/2}. \quad (17.5)$$

A matrix,  $\Upsilon = \{v_{lj}\}$ , is *nonnegative* if all of its elements are nonnegative, that is,  $v_{lj} \geq 0, \forall l, j$ . The matrix is positive if the inequality is strict.

A matrix  $\Upsilon = \{v_{lj}\}$  is *row(column)-stochastic* if it is nonnegative and its row (column) sums are equal to 1, that is,

$$\sum_j v_{lj} = 1, \quad \forall l.$$

A matrix  $\Upsilon$  is *doubly stochastic* if it is both row-stochastic and column-stochastic.

A matrix  $\mathbf{P} = \{p_{lj}\}$  is *row substochastic* if it is nonnegative, and at least one of its row has a sum strictly less than 1, that is,  $\sum_j p_{lj} < 1$ , for some  $l$ .

The graph  $\mathcal{G}(\Upsilon)$  associated to a matrix  $\Upsilon$  is the graph with the adjacency matrix,  $\mathbf{A} = \{a_{lj}\}$ , such that

$$a_{lj} = \begin{cases} 1, & v_{lj} \neq 0, j \neq l, \\ 0, & \text{otherwise.} \end{cases} \quad (17.6)$$

A matrix  $\Upsilon$  is *irreducible* if its associated graph  $\mathcal{G}(\Upsilon)$  is strongly connected.

A nonnegative matrix  $\Upsilon$  is *primitive* if there exists a natural number  $q$  such that  $\Upsilon^q$  is positive. A sufficient condition for a matrix  $\Upsilon$  to be primitive is that it is nonnegative, irreducible, and its trace is positive.

An  $M \times M$  matrix  $\mathbf{D} = \{d_{lj}\}$  is *strict diagonally dominant* if

$$|d_{ll}| > \sum_{j=1, j \neq l}^M |d_{lj}|, \quad \forall l. \quad (17.7)$$

A strictly diagonal dominant matrix is nonsingular; this result, also known as the Levy–Desplanques–Hadamard theorem, follows from the Gershgorin circle theorem and has been rediscovered by many authors [52, 53]. If in addition the matrix is Hermitian with nonnegative (positive) diagonal entries, then it is positive semidefinite (positive definite).

### 17.2.3 Markov Chains

In this section, we review basic definitions of Markov chains. Let an  $N \times N$  matrix,  $\Upsilon = \{v_{ij}\}$ , denote the transition probability matrix of a Markov chain with  $N$  states,  $s_i, i = 1, \dots, N$ .

A state  $s_i$  is called an *absorbing state* if the probability of leaving that state is 0 (i.e.,  $v_{ij} = 0, i \neq j$ , in other words  $v_{ii} = 1$ ).

A Markov chain is said to be an *absorbing Markov chain* if it has at least one absorbing state, and if from every state it is possible to go with nonzero probability to an absorbing state (not necessarily in one step).

In an absorbing Markov chain, a state that is not absorbing is called a *transient state*. For additional background, see, for example, [54].

### 17.2.4 Distributed Algorithms

A distributed, iterative linear algorithm to perform a certain task can now be written as

$$\mathbf{c}_l(t+1) = \sum_{j \in \mathcal{K}(l)} v_{lj} \mathbf{c}_j(t), \quad (17.8)$$

where  $\mathbf{c}_l(t) \in \mathbb{R}^{1 \times m}$  is a row vector that denotes the  $m$ -dimensional state of sensor  $l$  at time  $t$  and  $v_{lj}$  are the weights of the linear combination that are pertinent to the underlying task. For example, if the goal is to locate the sensors in two dimensions,  $m = 2$ , while if it is to locate in three dimensions, it is  $m = 3$ . For a given application,  $m$  takes an appropriate value. Note that the algorithm in (17.8) is distributed, that is, (i) the state update at any sensor requires the state from the neighboring sensors only, that is, the states from those sensors that can communicate with the sensor in question; and (ii) the weights,  $v_{lj}$ , are to be computed using local information only. By local, we mean that the information required to compute the weights,  $v_{lj}$ , is available only at the  $l$ th sensor and its close-by neighbors. Hence, the above algorithm captures all the requirements introduced in Section 17.1 to carry out a distributed, iterative procedure in a large-scale network.

Algorithm (17.8) can be written in matrix notation for compaction and analysis purposes. Let the  $N \times m$  matrix

$$\mathbf{C}(t) = [\mathbf{c}_1^T(t), \dots, \mathbf{c}_N^T(t)]^T$$

collect the states of all ( $N$ ) the sensors in the network. Then, the matrix form of (17.8) is given by

$$\mathbf{C}(t+1) = \boldsymbol{\Upsilon} \mathbf{C}(t), \quad (17.9)$$

$$= \boldsymbol{\Upsilon}^{t+1} \mathbf{C}(0), \quad (17.10)$$

where the  $N \times N$  iteration matrix, the matrix of the weights,  $v_{lj}$ ,  $\boldsymbol{\Upsilon} = \{v_{lj}\}$ , can be designed to perform different tasks.

The design of the iteration matrix  $\boldsymbol{\Upsilon}$  in the context of specific applications will be the main focus of the rest of the chapter. There are two scenarios that can arise in the context of specific applications: (i) In the first case, the sensor communication graph  $\mathcal{G}$  is given and remains fixed, that is, we cannot add any other edge to  $\mathcal{G}$ . The nonzero pattern of the iteration matrix  $\boldsymbol{\Upsilon}$  follows directly from the nonzero pattern in the adjacency matrix of the sensor communication graph  $\mathcal{G}$ . It is straightforward to note that in this case,  $\mathcal{G}(\boldsymbol{\Upsilon}) \subseteq \mathcal{G}$ . (ii) In the second case, the iteration matrix  $\boldsymbol{\Upsilon}$  is given and remains fixed. The sparsity (nonzero) pattern of the iteration matrix  $\boldsymbol{\Upsilon}$  provides the communication network over which the sensors communicate. In other words, if we have a nonzero in the iteration matrix  $\boldsymbol{\Upsilon}$  at  $(l, j)$ th location, then the sensor  $j$

is required to send information to sensor  $l$ . This translates to an edge,  $(l, j) \in E$ , in the sensor communication graph  $\mathcal{G}$ . Hence, for a given iteration matrix  $\Upsilon$  the sensor communication graph  $\mathcal{G}$  is the communication graph  $\mathcal{G}(\Upsilon)$  associated to the iteration matrix  $\Upsilon$ .

### 17.3 DISTRIBUTED DETECTION

In this section, we study the simple binary hypothesis test where the state of the environment takes one of two possible alternatives,  $H_0$  (target absent) or  $H_1$  (target present). The true state,  $H$ , is monitored by a network,  $\mathcal{G}$ , of  $N$  sensors. These sensors collect measurements,  $\mathbf{y} = (y_1, \dots, y_N)$ , that are independent and identically distributed (i.i.d.) conditioned on the true state  $H$ ; their known conditional probability density is  $p_i(y) = p(y|H_i)$ ,  $i = 0, 1$ . Each sensor,  $l$ , starts by computing the (local) log-likelihood ratio,

$$c_l = \ln \frac{p(y_l|H_1)}{p(y_l|H_0)}, \quad (17.11)$$

of its measurement  $y_l$ . The optimal decision is

$$L = \frac{1}{N} \sum_{l=1}^N c_l \begin{cases} \hat{H}=1 & \geqslant \\ \hat{H}=0 & \end{cases} v, \quad (17.12)$$

where  $v$  denotes an appropriate threshold derived, for example, from a Bayes' criteria that minimizes the average probability of error,  $P_e$ .

To be specific, we consider the simple binary hypothesis problem

$$H_i: y_l = \mu_i + \xi_l, \quad \xi_l \sim \mathcal{N}(0, \sigma^2), \quad i = 0, 1, \quad (17.13)$$

where, without loss of generality, we let  $\mu_1 = -\mu_0 = \mu$ . Under this model, the local likelihoods,  $c_l$ , are also Gaussian, that is,

$$H_i: c_l \sim \mathcal{N}\left(\frac{2\mu\mu_i}{\sigma^2}, \frac{4\mu^2}{\sigma^2}\right). \quad (17.14)$$

From (17.12), the test statistic for the optimal decision is also Gaussian, that is,

$$H_i: L \sim \mathcal{N}\left(\frac{2\mu\mu_i}{\sigma^2}, \frac{4\mu^2}{N\sigma^2}\right). \quad (17.15)$$

The error performance of the minimum probability of error,  $P_e$  [Bayes' detector with threshold  $v = 0$  in (17.12)] is given by

$$P_e = \text{erfc}^* \left( \frac{d}{2} \right) = \int_{d/2}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx, \quad (17.16)$$

where the equivalent signal-to-noise ratio (SNR)  $d^2$  that characterizes the performance is given by [55]

$$d = \frac{2\mu\sqrt{N}}{\sigma}. \quad (17.17)$$

This is a standard problem studied often in the context of decentralized detection and sensor networks with parallel architectures since the early work in [56]; see, for example [7, 57, 58], and some of our own work [9, 59, 60].

We now consider the algorithm given in (17.8) to carry out this detection problem in a decentralized and distributed manner. The state of each sensor in this case is one dimensional, that is,  $m = 1$ . Let the initial condition at sensor  $l$  be given by (17.11), that is,

$$c_l(0) = \ln \frac{p(y_l|H_1)}{p(y_l|H_0)}. \quad (17.18)$$

Our goal is to design the weights,  $v_{lj}$ , in (17.8) such that, asymptotically, each sensor reaches the optimal decision (17.12), that is,

$$\begin{aligned} \lim_{t \rightarrow \infty} c_l(t+1) &= L, \\ &= \frac{1}{N} \sum_{l=1}^N c_l(0), \end{aligned} \quad (17.19)$$

which is the numerical average of the initial condition. Hence, the distributed detection problem reduces to computing the average of the initial conditions at each sensor in the sensor network in a distributed, iterative way. The specific algorithm used for this purpose is the “average-consensus” algorithm [61]. From (17.10), we note that, when we have the following property on the weight matrix,  $\Upsilon = \{v_{lj}\}$ ,

$$\lim_{t \rightarrow \infty} \Upsilon^{t+1} = \frac{\mathbf{1}\mathbf{1}^T}{N}, \quad (17.20)$$

the average-consensus algorithm converges asymptotically to the optimal detector. In (17.20),  $\mathbf{1}$  is a vector of 1’s. We discuss the properties under which (17.20) can be realized in Section 17.5. This distributed average-consensus detector achieves asymptotically (in the number of iterations) the optimal error performance,  $P_e$ , of the optimal (centralized) detector given by (17.16); see [62, 63].

## 17.4 CONSENSUS ALGORITHMS

We view the sensor network as a graph  $\mathcal{G}$ . We consider that the sensors in the network belong to two different groups: (i) *sensors* in the strict sense that are collected in the set  $\Omega$ ; and (ii) *anchors* that are grouped in the set  $\kappa$ . The cardinalities of each set are  $|\Omega| = M$  and  $|\kappa| = n$ . The sensors in  $\Omega$  update their state in a distributed way to accomplish a desired task. The state of the anchors in  $\kappa$  remains constant throughout the algorithm [45]. In other words, the anchors serve as inputs or leaders to the system; they help guide the algorithm in the direction of completing the desired particular task. The set of nodes in the network is denoted as  $\Theta = \Omega \cup \kappa$ , with  $|\Theta| = N = M + n$ . It may be helpful to consider the sensors as followers in a leader–follower (type of) task. From this point onward, we use the term “sensor” to imply a nonanchor node. We refer to distributed algorithms for sensor networks as consensus algorithms if the sensors cooperate to achieve an agreeable state. Before we proceed, we introduce the definition of *dimension*.

**Definition 17.1 (Dimension)** *The number of anchors in  $\kappa$  is the dimension of the consensus algorithm.*

With each sensor,  $l$ , we associate the  $m$ -dimensional row vector,  $\mathbf{x}_l(t)$ , that denotes its state at time  $t$ . Similarly, the row vector,  $\mathbf{u}_k(t)$ , denotes the  $m$ -dimensional state of the  $k$ th anchor at time  $t$ . Since we consider that the state of the anchors stays constant, we have

$$\mathbf{u}_k(t) \equiv \mathbf{u}_k$$

in the sequel. Let

$$\begin{aligned}\mathbf{U} &= [\mathbf{u}_1^T, \dots, \mathbf{u}_n^T]^T, \\ \mathbf{X}(t) &= [\mathbf{x}_{n+1}^T(t), \dots, \mathbf{x}_N^T(t)]^T.\end{aligned}$$

We can partition the state,  $\mathbf{C}(t)$  in (17.9) of the entire sensor network as

$$\mathbf{C}(t) = \begin{bmatrix} \mathbf{U} \\ \mathbf{X}(t) \end{bmatrix}. \quad (17.21)$$

Similarly, we can partition the  $N \times N$  iteration matrix,  $\mathbf{\Upsilon}$  in (17.9), as

$$\mathbf{\Upsilon} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{B} & \mathbf{P} \end{bmatrix}, \quad (17.22)$$

where  $\mathbf{I}_n$  denotes the  $n \times n$  identity matrix,  $\mathbf{B} \in \mathbb{R}^{M \times n}$ , and  $\mathbf{P} \in \mathbb{R}^{M \times M}$ . With the above partition, we can write the general form of consensus algorithms as

$$\mathbf{X}(t + 1) = \mathbf{P}\mathbf{X}(t) + \mathbf{B}\mathbf{U}. \quad (17.23)$$

The matrix  $\mathbf{P}$  collects the updating weights for each of the sensors in terms of its neighboring sensors. Similarly, the matrix  $\mathbf{B}$  collects the updating weights for each of those sensors that are connected to the anchors.

It is straightforward to note that (17.23) can be expanded in terms of the initial condition,  $\mathbf{X}(0)$ , as follows:

$$\mathbf{X}(t + 1) = \mathbf{P}^{t+1}\mathbf{X}(0) + \sum_{k=0}^t \mathbf{P}^k \mathbf{B}\mathbf{U}. \quad (17.24)$$

*Distributed Representation* Let  $\kappa_{\mathcal{K}(l)} = \mathcal{K}(l) \cap \kappa$ , that is, the neighbors of sensor  $l$  that are anchors, and let  $\Omega_{\mathcal{K}(l)} = \mathcal{K}(l) \cap \Omega$ , that is, the neighbors of sensor  $l$  that are sensors. The distributed representation of the consensus algorithms in (17.23) can be written as

$$\mathbf{x}_l(t + 1) = \sum_{j \in \Omega_{\mathcal{K}(l)}} p_{lj} \mathbf{x}_j(t) + \sum_{k \in \kappa_{\mathcal{K}(l)}} b_{lk} \mathbf{u}_k. \quad (17.25)$$

### 17.4.1 Classification of Consensus Algorithms

We now characterize the general distributed algorithms for sensor networks into the following two categories.

**17.4.1.1 Zero-Dimension Consensus** In this case, the spectral radius of  $\mathbf{P}$  is unity, that is,

$$\rho(\mathbf{P}) = 1. \quad (17.26)$$

With this assumption, the sum in (17.24) diverges unless  $n = 0$  (no anchors); in this case, (17.23) reduces to the form

$$\mathbf{X}(t + 1) = \mathbf{P}\mathbf{X}(t), \quad (17.27)$$

$$= \mathbf{P}^{t+1}\mathbf{X}(0). \quad (17.28)$$

Since we have  $n = 0$ , we term this as a consensus algorithm in zero dimensions. It is worth mentioning that with this algorithm the convergent values,  $\lim_{t \rightarrow \infty} \mathbf{X}(t + 1)$ , are a function of the initial condition,  $\mathbf{X}(0)$ . The average-consensus algorithm mentioned in Section 17.3 falls into this category.

**17.4.1.2 Consensus in Higher Dimensions** In this case, the spectral radius of  $\mathbf{P}$  is strictly less than unity, that is,

$$\rho(\mathbf{P}) < 1. \quad (17.29)$$

With this assumption in (17.24), if there are no anchors,  $n = 0$ , then  $\mathbf{X}(t + 1)$  converges to a  $\mathbf{0}$  matrix of appropriate dimensions, that is,

$$\lim_{t \rightarrow \infty} \mathbf{X}(t + 1) = \mathbf{0}.$$

The number of anchors should be at least 1 for the algorithm to converge to anything other than  $\mathbf{0}$ . Here, we assume  $n \geq 1$  and term this as a consensus in “higher dimensions.” We explore the following examples of the consensus problem in higher dimensions.

1. *Leader–follower algorithm*—The sensors converge to a linear combination of the anchors. Within certain conditions, the weights of the linear combination can be designed by appropriately choosing the elements of the matrices  $\mathbf{B}$  and  $\mathbf{P}$ .
2. *Localization in sensor networks*—The sensors determine their exact locations in  $m$ -dimensional Euclidean space,  $\mathbb{R}^m (m \geq 1)$ , in the presence of a minimal number,  $m + 1$ , of anchors that know their locations exactly.
3. Solving linear systems of equations  $\mathbf{DX} = \mathbf{U}$ .

**17.4.1.3 Remarks** The notion of “dimension” is associated to the convergent state,  $\lim_{t \rightarrow \infty} \mathbf{C}(t + 1)$ , of the algorithm in (17.9). This convergent state resides in an  $n$ -dimensional space that is spanned by the  $n$  eigenvectors of the iteration matrix,  $\mathbf{Y}$ , corresponding to eigenvalue 1, where  $n$  is the number of anchors. This justifies the use of the word “dimensions” in Definition 17.1 to characterize the consensus algorithm. Mathematically, we note that the iteration matrix,  $\mathbf{Y}$ , can be expressed as

$$\mathbf{Y} = \mathbf{Q}\Lambda\mathbf{R}, \quad (17.30)$$

where the matrix

$$\mathbf{Q} = [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \dots \quad \mathbf{q}_N]$$

is the matrix of right-normalized eigenvectors,  $\mathbf{q}_i$ , of  $\Upsilon$ ; the matrix

$$\mathbf{R}^H = [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \dots \quad \mathbf{r}_N]$$

is the matrix of left-normalized eigenvectors,  $\mathbf{r}_i^H$ , or  $\Upsilon$ ; and the matrix

$$\Lambda = \text{diag}[\lambda_1(\Upsilon) \cdots \lambda_m(\Upsilon)]$$

is the diagonal matrix of the eigenvalues,  $\lambda_i(\Upsilon)$ , of the iteration matrix  $\Upsilon$ .

Since the iteration matrix  $\Upsilon$  is block lower triangular, its eigenvalues are the eigenvalues of the diagonal blocks,  $\mathbf{I}_n$  and  $\mathbf{P}$ . From (17.29), we have

$$\lambda_i(\mathbf{P}) < 1, \quad i = 1, \dots, M,$$

$$\lim_{t \rightarrow \infty} \lambda_i^{t+1} = 0, \quad i = 1, \dots, M.$$

We assume that  $\mathbf{Q} = \mathbf{R}^{-1}$ , which is true for normal matrices  $\Upsilon$ ; see [64]. We can write the convergent state,  $\mathbf{C}(t + 1)$ , in the limit as  $t \rightarrow \infty$ , as

$$\begin{aligned} \lim_{t \rightarrow \infty} \mathbf{C}(t + 1) &= \lim_{t \rightarrow \infty} \Upsilon^{t+1} \mathbf{C}(0) \\ &= \lim_{t \rightarrow \infty} \mathbf{Q} \Lambda^{t+1} \mathbf{R} \mathbf{C}(0) \\ &= \lim_{t \rightarrow \infty} [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \dots \quad \mathbf{q}_N] \begin{bmatrix} \mathbf{I}_n^{t+1} & & & \\ & \lambda_1^{t+1} & & \\ & & \ddots & \\ & & & \lambda_M^{t+1} \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^H \\ \mathbf{r}_2^H \\ \vdots \\ \mathbf{r}_N^H \end{bmatrix} \mathbf{C}(0) \\ &= \sum_{i=1}^n \mathbf{q}_i \mathbf{r}_i^H \mathbf{C}(0). \end{aligned} \tag{17.31}$$

## 17.5 ZERO-DIMENSION (AVERAGE) CONSENSUS

As we mentioned in Section 17.4.1, if  $\rho(\mathbf{P}) = 1$  and  $n = 0$  (the dimension of the consensus algorithm) then (17.23) reduces to (17.27), which can be written in terms of the initial condition as given in (17.28). In this case, the  $N \times N$  iteration matrix  $\Upsilon$  reduces to an  $M \times M$  iteration matrix  $\mathbf{P}$  since  $n = 0$  and  $N = M + n$ . Without loss of generality, we assume that  $m = 1$ , that is, each sensor has a scalar state,  $x_l(0)$  at sensor  $l$ , and the objective is to design the iteration matrix  $\mathbf{P}$  such that

$$\lim_{t \rightarrow \infty} x_l(t + 1) = \frac{1}{M} \sum_{k=1}^M x_k(0). \tag{17.32}$$

Recall that  $\mathbf{1}$  denotes an  $M \times 1$  column vector of 1's. In matrix form, (17.32) translates to

$$\lim_{t \rightarrow \infty} \mathbf{X}(t + 1) = \frac{\mathbf{1}\mathbf{1}^T}{M} \mathbf{X}(0). \tag{17.33}$$

The design criterion of the iteration matrix  $\mathbf{P}$  can now be written as

$$\lim_{t \rightarrow \infty} \mathbf{P} = \frac{\mathbf{1}\mathbf{1}^T}{M}.$$

### 17.5.1 Design of the Iteration Matrix $\mathbf{P}$

We choose the elements of the iteration matrix  $\mathbf{P} = \{p_{ij}\}$  such that  $\mathbf{P}$  is doubly stochastic. Since  $\mathbf{P}$  is row-stochastic, we note that  $\mathbf{1}/\sqrt{M}$  is the right (normalized) eigenvector of the iteration matrix  $\mathbf{P}$ , with eigenvalue 1, that is,

$$\frac{1}{\sqrt{M}} \mathbf{P}\mathbf{1} = \frac{1}{\sqrt{M}} \mathbf{1}. \quad (17.34)$$

Since  $\mathbf{P}$  is column-stochastic, we note that  $\mathbf{1}^T/\sqrt{M}$  is the left (normalized) eigenvector of the iteration matrix  $\mathbf{P}$ , also with eigenvalue 1, that is,

$$\frac{1}{\sqrt{M}} \mathbf{1}^T \mathbf{P} = \frac{1}{\sqrt{M}} \mathbf{1}^T. \quad (17.35)$$

Along with (17.35), we also assume that the associated graph,  $\mathcal{G}(\mathbf{P})$ , is balanced.

### 17.5.2 Convergence

We assume the eigenvalues,  $\lambda_{i(\mathbf{P})}$ , of the iteration matrix,  $\mathbf{P}$ , are ordered in terms of their magnitude, that is,

$$|\lambda_{1(\mathbf{P})}| \geq |\lambda_{2(\mathbf{P})}| \geq \cdots \geq |\lambda_{M(\mathbf{P})}|.$$

Since  $\mathbf{P}$  is doubly stochastic,

$$\rho(\mathbf{P}) = |\lambda_{1(\mathbf{P})}| = 1;$$

see [51]. With the additional properties that the iteration matrix  $\mathbf{P}$  is primitive and irreducible, it can be shown [51] that the iteration matrix  $\mathbf{P}$  has exactly one simple (of multiplicity 1) eigenvalue of 1, and,

$$|\lambda_{i(\mathbf{P})}| < |\lambda_{1(\mathbf{P})}| = 1, \quad i = 2, \dots, M. \quad (17.36)$$

Since the iteration matrix  $\mathbf{Y}$  reduces to  $\mathbf{P}$  when  $n = 0$ , we associate the same eigenexpansion to  $\mathbf{P}$  as in (17.30). The only difference is that the matrices in the expansion are now  $M \times M$ . We have

$$\begin{aligned} \mathbf{P}^{t+1} &= \mathbf{Q}\Lambda^{t+1}\mathbf{R}, \\ &= \sum_{i=1}^M \lambda_{i(\mathbf{P})}^{t+1} \mathbf{q}_i \mathbf{r}_i^H, \\ &= \lambda_{1(\mathbf{P})}^{t+1} \mathbf{q}_1 \mathbf{r}_1^H + \sum_{i=2}^M \lambda_{i(\mathbf{P})}^{t+1} \mathbf{q}_i \mathbf{r}_i^H. \end{aligned} \quad (17.37)$$

From (17.34)–(17.36), in the limit as  $t \rightarrow \infty$ , (17.37) is given by

$$\lim_{t \rightarrow \infty} \mathbf{P}^{t+1} = \frac{1}{M} \mathbf{1} \mathbf{1}^T. \quad (17.38)$$

Hence, using such a  $\mathbf{P}$  gives us the required average-consensus algorithm.<sup>2</sup>

### 17.5.3 Example

For any arbitrary strongly connected and balanced graph with Laplacian matrix  $\mathbf{L}$  and adjacency matrix  $\mathbf{A} = \{a_{lj}\}$ , it can be shown [65] that

$$\mathbf{P} = \mathbf{I}_M - \epsilon \mathbf{L}$$

with

$$0 < \epsilon < \max_{l,l \neq j} \sum_l a_{lj}$$

gives the distributed iterative average-consensus algorithm.

## 17.6 CONSENSUS IN HIGHER DIMENSIONS

In this section, we consider the general distributed consensus algorithms with  $n \geq 1$ , that is, we have at least one anchor, as given in (17.25). The matrix form of (17.25) is given in (17.23), which can be written in terms of the initial condition as (17.24). For the sake of convenience, we copy (17.24) here again:

$$\mathbf{X}(t+1) = \mathbf{P}^{t+1} \mathbf{X}(0) + \sum_{k=0}^t \mathbf{P}^k \mathbf{B} \mathbf{U}. \quad (17.39)$$

To proceed with the convergence analysis of (17.39), we provide the following lemma:

**Lemma 17.1** *If a matrix  $\mathbf{P}$  is such that*

$$\rho(\mathbf{P}) < 1, \quad (17.40)$$

*then*

$$\lim_{t \rightarrow \infty} \mathbf{P}^{t+1} = \mathbf{0}, \quad (17.41)$$

$$\lim_{t \rightarrow \infty} \sum_{k=0}^t \mathbf{P}^k = (\mathbf{I} - \mathbf{P})^{-1}. \quad (17.42)$$

*Proof* The proof is trivial.

<sup>2</sup>In this section, the zero-dimensional consensus algorithm is provided in the context of average consensus. If we remove the requirement that the graph  $\mathcal{G}(\mathbf{P})$  associated to the iteration matrix  $\mathbf{P}$  is balanced, then the zero-dimension consensus algorithm converges to a convex combination [65],  $\sum_l w_l x_l(0)$  ( $\sum_i w_i = 1, w_i \geq 0$ ), of the initial condition.

We note here that a sufficient condition for the design of  $\mathbf{P}$  guaranteeing (17.40) is that  $\mathbf{P}$  be irreducible. This condition translates into a connected graph,  $\mathcal{G}$  for the sensor network. With a connected graph assumption, we can design the elements of  $\mathbf{P}$  such that (17.40) holds. From Lemma 17.1, in the limit as  $t \rightarrow \infty$ , we can write (17.39) as

$$\lim_{t \rightarrow \infty} \mathbf{X}(t + 1) = (\mathbf{I} - \mathbf{P})^{-1} \mathbf{B} \mathbf{U}. \quad (17.43)$$

The asymptotic state,  $\lim_{t \rightarrow \infty} \mathbf{X}(t + 1)$ , of the sensors is independent of the sensor initial condition,  $\mathbf{X}(0)$ . We further note that (17.40) implies that no eigenvalue ( $\lambda_i(\mathbf{I} - \mathbf{P}) = 1 - \lambda_i(\mathbf{P})$ ) of the matrix,  $\mathbf{I} - \mathbf{P}$ , can be 0; hence,  $\mathbf{I} - \mathbf{P}$  is always invertible if (17.40) holds. With (17.43), we can design the matrices,  $\mathbf{P} = \{p_{lj}\}$  and  $\mathbf{B} = \{b_{lj}\}$ , to accomplish different tasks. The next sections describe important applications of such algorithms with emphasis on the design of the matrices  $\mathbf{P}$  and  $\mathbf{B}$ .

## 17.7 LEADER–FOLLOWER (TYPE) ALGORITHMS

We characterize leader–follower algorithms into the following two categories.

### 17.7.1 One Anchor, $n = 1$

In this case, we have only one anchor,  $n = 1$ ; the goal is for the entire sensor network to converge to the state of the anchor. Let  $\mathbf{1}_M$  be the  $M \times 1$  column vector of 1's; we require the state of the entire sensor network, asymptotically, to be

$$\lim_{t \rightarrow \infty} \mathbf{X}(t + 1) = \mathbf{1}_M \mathbf{u}_1, \quad (17.44)$$

where  $\mathbf{U} = \mathbf{u}_1$  is the state of the only anchor.

Since  $n = 1$ , the iteration matrix  $\mathbf{\Upsilon}$  can be partitioned as

$$\mathbf{\Upsilon} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{b} & \mathbf{P} \end{bmatrix}, \quad (17.45)$$

where  $\mathbf{b}$  is an  $M \times 1$  column vector.

*Design of Iteration matrix  $\mathbf{\Upsilon}$*  With the iteration matrix  $\mathbf{\Upsilon}$ , as given in (17.45), and relating (17.44) to (17.43), we require

$$(\mathbf{I} - \mathbf{P})^{-1} \mathbf{b} = \mathbf{1}_M. \quad (17.46)$$

Let

$$\mathbf{b} = [b_{11} \cdots b_{M1}].$$

We can write (17.46) as

$$\mathbf{b} = (\mathbf{I} - \mathbf{P}) \mathbf{1}_M, \quad (17.47)$$

$$= \mathbf{1}_M - \mathbf{P} \mathbf{1}_M. \quad (17.48)$$

Elementwise, for  $l = 1, \dots, M$ , we have the following condition on  $\mathbf{b}$  and  $\mathbf{P}$ :

$$b_{l1} + \sum_{j=1}^M p_{lj} = 1, \quad l = 1, \dots, M, \quad (17.49)$$

in addition to (17.40).

**Example 17.1** We may choose the following design strategy. If sensor  $l$  is connected to anchor 1, we choose

$$p_{lj} = \frac{1 - \epsilon_l}{|\mathcal{K}(l)|},$$

$$b_{l1} = \epsilon_l,$$

where  $\epsilon_l \in (0, 1)$ . If sensor  $l$  is not connected to anchor 1, we choose

$$p_{lj} = \frac{1}{|\mathcal{K}(l)|}, \quad (17.50)$$

$$b_{l1} = 0. \quad (17.51)$$

It can be noted that with the above choice the matrix  $\mathbf{P}$  is irreducible and further substochastic guaranteeing (17.40).

### 17.7.2 Multiple Anchors, $n > 1$

In this case, we have multiple anchors ( $n > 1$ ) and we examine the subspace to which the sensors can converge. From (17.43), the state of all sensors, asymptotically, is

$$\lim_{t \rightarrow \infty} \mathbf{X}(t+1) = \mathbf{WU}, \quad (17.52)$$

where

$$\mathbf{W} = (\mathbf{I} - \mathbf{P})^{-1} \mathbf{B}. \quad (17.53)$$

We study what are the possible matrices  $\mathbf{W}$  that we can achieve keeping the interconnected structure of the network intact, that is, the sparsity pattern of the matrices  $\mathbf{B}$  and  $\mathbf{P}$ . To this end, we give the following results.

**Lemma 17.2** Let  $r_A$  be the rank of the  $M \times M$  matrix  $(\mathbf{I} - \mathbf{P})^{-1}$ , and let  $r_B$  be the rank of the  $M \times n$  matrix  $\mathbf{B}$ , then

$$\text{rank}(\mathbf{I} - \mathbf{P})^{-1} \mathbf{B} \leq \min(r_A, r_B), \quad (17.54)$$

$$\text{rank}(\mathbf{I} - \mathbf{P})^{-1} \mathbf{B} \geq r_A + r_B - M. \quad (17.55)$$

*Proof* The proof is available on pages 95 and 96 in [66].

Since  $\mathbf{I} - \mathbf{P}$  is invertible,  $r_A = M$ . In general, we will assume the number of anchors,  $n$ , to be much smaller than the number of sensors,  $M$ , that is,  $M \gg n$ . From Lemma 17.2, we note that

$$\text{rank}(\mathbf{I} - \mathbf{P})^{-1}\mathbf{B} \leq r_B, \quad (17.56)$$

$$\text{rank}(\mathbf{I} - \mathbf{P})^{-1}\mathbf{B} \geq M + r_B - M = r_B. \quad (17.57)$$

Since  $\mathbf{W} = (\mathbf{I} - \mathbf{P})^{-1}\mathbf{B}$ , we have

$$\text{rank}(\mathbf{W}) = r_B. \quad (17.58)$$

The above result suggests that the rank of  $\mathbf{W}$  equals the rank of  $\mathbf{B}$ .

**Example 17.2** Let

$$w_i \in [0, 1], \quad \sum_i w_i = 1, \quad i = 1, \dots, n. \quad (17.59)$$

For the sensors to converge to a convex combination of the anchors, we take the weight matrix  $\mathbf{W}$  to be of the form

$$\mathbf{W} = \begin{bmatrix} w_1 & w_2 & \dots & w_n \\ \vdots & & & \\ w_1 & w_2 & \dots & w_n \end{bmatrix}. \quad (17.60)$$

It follows that  $\text{rank}(\mathbf{B}) = 1$ , since  $\text{rank}(\mathbf{W}) = 1$ , that is, all columns of  $\mathbf{B}$  should be linearly dependent. Thus, if sensor  $l$  has a communication link to any anchor  $k \in \kappa$ , it has to have a communication link to all the anchors. Note that a sensor,  $l$ , has a communication link to an anchor,  $k \in \kappa$ , when  $b_{lk} \neq 0$ , that is, the  $k$ th column of  $\mathbf{B}$  has a nonzero at the  $l$ th location. For the matrix  $\mathbf{B}$  to have rank 1, no column of  $\mathbf{B}$  can have a zero at the  $l$ th location.<sup>3</sup> Mathematically,

$$b_{lk} \neq 0, \quad \text{for any } k \in \kappa \Rightarrow b_{lk} \neq 0, \forall k \in \kappa. \quad (17.61)$$

Similarly,

$$b_{lk} = 0, \quad \text{for any } k \in \kappa \Rightarrow b_{lk} = 0, \forall k \in \kappa. \quad (17.62)$$

Furthermore, let  $\mathbf{b}_k$  denote the  $k$ th column of the matrix  $\mathbf{B}$ ; then for every  $k, j \in \kappa$  there exists some  $\beta_{kj} \in \mathbb{R}$ , such that

$$\mathbf{b}_k = \beta_{kj} \mathbf{b}_j. \quad (17.63)$$

A distributed algorithm designed with the elements,  $\mathbf{B} = \{b_{lj}\}$ , chosen in the way described above will result into all the sensors converging to a convex combination

<sup>3</sup>We assume that  $\mathbf{b}_k \neq \mathbf{0}, \forall k$ . If this is not true, then the  $k$ th anchor does not send information to any sensor and, hence, does not take part in the algorithm. So it can be removed from the analysis.

of the anchors. An important case is when  $\text{rank}(\mathbf{W}) = r_B$ . This arises in distributed sensor localization, as we discuss in the next section.

## 17.8 LOCALIZATION IN SENSOR NETWORKS

In [46], we presented a distributed sensor localization algorithm (finding the unknown locations of  $M$  sensors) in  $m$ -dimensional Euclidean spaces,  $\mathbb{R}^m$  ( $m \geq 1$ ), that we now cast in the framework of (17.23). The algorithm uses only mutual distances among a sensor and its neighbors and requires a minimal number,  $m + 1$ , of anchors. The anchors are the sensors that know their locations exactly, for example, they can be instrumented with a Global Positioning System (GPS) or have a fixed geographic location (lighthouses or beacons). The iteration matrix is derived using barycentric coordinates, Cayley–Menger determinants, and some concepts from Euclidean geometry that we describe briefly in the following.

### 17.8.1 Background

For the following, we assume that  $\kappa$  is the set of  $m + 1$  points in  $\mathbb{R}^m$ .

*Convex Hull* The convex hull,  $\mathcal{C}(\kappa)$ , of the set  $\kappa$  is defined as the minimal convex set that contains all the points in  $\kappa$ . When any point,  $l$ , is contained in the convex hull,  $\mathcal{C}(\kappa)$ , of  $\kappa$ , we write it as  $l \subseteq \mathcal{C}(\kappa)$ .

*Generalized Volume* The generalized volume,  $A_\kappa$ , of the set  $\kappa$  is the volume of the  $m$ -dimensional hypercube,  $\mathcal{C}(\kappa)$ . For simplicity, consider the area of a triangle when  $m = 2$  and the volume of a tetrahedron when  $m = 3$ .

*Cayley–Menger Determinants* The Cayley–Menger determinant provides the generalized volume,  $A_\kappa$ , of  $\mathcal{C}(\kappa)$  [67]. Let  $\mathbf{1}_{m+1}$  denote a column vector of  $m + 1$  1's; the Cayley–Menger determinant is given by

$$A_\kappa^2 = \frac{1}{s_{m+1}} \begin{vmatrix} 0 & \mathbf{1}_{m+1}^\top \\ \mathbf{1}_{m+1} & \boldsymbol{\Gamma} \end{vmatrix}, \quad (17.64)$$

where  $\boldsymbol{\Gamma} = \{d_{lj}^2\}$ ,  $l, j \in \kappa$ , is the matrix of squared distances,  $d_{lj}$ , among the  $m + 1$  points in  $\kappa$  and

$$s_m = \frac{2^m (m!)^2}{(-1)^{m+1}}, \quad m = \{0, 1, 2, \dots\}. \quad (17.65)$$

Its first few terms are  $-1, 2, -16, 288, -9216, 460800, \dots$

*Barycentric Coordinates* Let  $l \in \mathbb{R}^m$  and  $\Theta_l$  be a set of  $m + 1$  points in  $\mathbb{R}^m$  such that

$$l \subseteq \mathcal{C}(\Theta_l) \quad \text{and} \quad A_{\Theta_l} \neq 0. \quad (17.66)$$

Let  $\mathbf{c}^*$  denote the  $m$ -dimensional row vector of the coordinates of the point  $l$ . The barycentric coordinates [68, 69] provide a linear representation of the coordinates of a

point,  $l$ , in terms of the coordinates of  $m + 1$  points,  $k \in \Theta_l$ , given that (17.66) holds. Mathematically,

$$\mathbf{c}_l^* = \sum_{j \in \Theta_l} v_{lj} \mathbf{c}_j^*, \quad l \in \mathcal{C}(\Theta_l), \quad (17.67)$$

where  $v_{lk}$  are the unique barycentric coordinates given by

$$v_{lk} = \frac{A_{\{l\} \cup \Theta_l \setminus \{k\}}}{A_{\Theta_l}}, \quad (17.68)$$

where \ denotes the set difference,  $A_{\{l\} \cup \Theta_l \setminus \{k\}}$  is the generalized volume of the set  $\{l\} \cup \Theta_l \setminus \{k\}$ , that is, the set  $\Theta_l$  with sensor  $l$  added and node  $k$  removed. Since the barycentric coordinates are a function of the generalized volumes, Cayley–Menger determinants can be employed to calculate them. It is shown in [46] that, using Cayley–Menger determinants, the barycentric coordinates can be computed from the distance information among the nodes in the set  $\{l \cup \Theta_l\}$ . Hence the computation of the barycentric coordinates is local.

### 17.8.2 Assumption

We provide the necessary assumptions we require for the development of the localization algorithm.

*(A0) Nondegeneracy* The generalized volume of  $\kappa$ ,  $A_\kappa \neq 0$ . Nondegeneracy simply states that the anchors do not lie on a hyperplane.

*(A1) Anchor Nodes* The anchors' locations are known exactly, that is, their state remains constant

$$\mathbf{u}_k(t) = \mathbf{u}_k^*, \quad k \in \kappa, t \geq 0, \quad (17.69)$$

where  $*$  denotes the exact coordinates.

*(A2) Convexity* All the sensors lie inside the convex hull of the anchors

$$\mathcal{C}(\Omega) \subset \mathcal{C}(\kappa), \quad (17.70)$$

where  $\mathcal{C}(\cdot)$  denotes the convex hull of the points in its argument.

*(A3) Triangulation* We assume that at each sensor,  $l \in \Omega$ , there exists a triangulation set,  $\Theta_l$ , such that

$$|\Theta_l| = m + 1, \quad (17.71)$$

$$A_{\Theta_l} \neq 0, \quad (17.72)$$

$$\Theta_l \subseteq \mathcal{K}(l), \quad l \notin \Theta_l, \quad l \in \mathcal{C}(\Theta_l), \quad (17.73)$$

where  $A_{\Theta_l}$  is the generalized volume of  $\mathcal{C}(\Theta_l)$ .

The triangulation set,  $\Theta_l$ , is a subset of  $m + 1$  sensors, possibly anchors, of the neighborhood set,  $\mathcal{K}(l)$ , of sensor  $l$ ,  $l \notin \Theta_l$ ; it is such that sensor  $l$  lies in the convex

hull of these  $m + 1$  sensors. In [46], a convex hull inclusion test is provided, that is, a test that determines if  $l$  lies in the convex hull of  $m + 1$  nodes arbitrarily chosen sensors. We assume the  $\mathcal{K}(l), \forall l$ , is large enough such that the triangulation procedure results in a triangulation set,  $\Theta_l, \forall l$ , satisfying the properties in (17.71)–(17.73). The probability of success of finding such a triangulation set,  $\Theta_l$ , in a small neighborhood is characterized in [46] in terms of the communication radius at each sensor and the density of deployment of the sensors. It is further shown in this reference that this probability can be made arbitrarily close to 1 by choosing appropriate network parameters.

### 17.8.3 Distributed Localization Algorithm

With the distributed localization algorithm that we consider now, the state of sensor  $l$ ,  $\mathbf{x}_l \in \mathbb{R}^m$ , is an  $m$ -dimensional row vector representing its location coordinates. Following the notation and dimensions in (17.23), we have  $M$  sensors with unknown locations in the set  $\Omega$  and  $m + 1$  anchors with known location in the set  $\kappa$ , that is,  $n = m + 1$ . The localization algorithm has the following phases:

*Setup Phase* Each sensor,  $l \in \Omega$ , identifies its triangulation set,  $\Theta_l$ , from its neighbors,  $\mathcal{K}(l)$ .

*Iterate Phase* Once a  $\Theta_l$  is identified at each sensor  $l$ , the localization algorithm is given by

$$\mathbf{x}_l(t+1) = \sum_{j \in \Omega_{\Theta_l}} p_{lj} \mathbf{x}_j(t) + \sum_{k \in \kappa_{\Theta_l}} b_{lk} \mathbf{u}_k, \quad (17.74)$$

where  $\Omega_{\Theta_l} = \Omega \cap \Theta_l$  and  $\kappa_{\Theta_l} = \kappa \cap \Theta_l$ .

### 17.8.4 Iteration Matrix $\Upsilon$ for Localization

As can be seen from (17.74), each sensor  $l$  expresses its state in terms of at most  $m + 1$  neighboring sensors' (or anchors') states. Note that this algorithm is similar to (17.25) apart from the choice of the neighborhoods. Hence, the  $l$ th row of the iteration matrix  $\Upsilon$  has at most  $m + 1$  nonzeros at appropriate locations. From (17.69), the iteration matrix, further, has a 1 as its  $(k, k)$ ,  $k \in \kappa$  element.

For each sensor  $l \in \Omega$ , the weights,

$$v_{lj} = \begin{cases} p_{lj}, & j \in \Omega_{\Theta_l}, \\ b_{lj}, & j \in \kappa_{\Theta_l}, \end{cases} \quad (17.75)$$

in (17.74) are the barycentric coordinates of sensor,  $l \in \mathbb{R}^m$ , with respect to  $m + 1$  nodes in the triangulation set,  $\Theta_l$ . From (17.68), and the fact that the generalized volumes are nonnegative and

$$\sum_{k \in \Theta_l} A_{\Theta_l \cup \{l\} \setminus \{k\}} = A_{\Theta_l}, \quad (17.76)$$

since  $l \in \mathcal{C}(\Theta_l)$ , it follows that, for each  $l \in \Omega$ ,  $k \in \Theta_l$ ,

$$v_{lk} \in [0, 1], \quad \text{and} \quad \sum_{k \in \Theta_l} v_{lk} = 1. \quad (17.77)$$

### 17.8.5 Convergence

With (17.77), we note that the iteration matrix  $\Upsilon$  is row-stochastic and can be considered as a transition probability matrix of a Markov chain. We further note that the Markov chain associated to  $\Upsilon$  has at least one absorbing states because  $n \geq 1$ . From our assumption (A3), it can be shown that the underlying Markov chain is absorbing; see [46]. With the absorbing Markov chain interpretation and the partition of  $\Upsilon$  as in (17.22), we note that the  $M \times (m + 1)$  block  $\mathbf{B} = \{b_{lj}\}$  is the subblock that contains the probabilities of the transient states to reach the absorbing states in onestep; and the  $M \times M$  block  $\mathbf{P} = \{p_{lj}\}$  is the subblock that contains the probabilities of the transient states to reach other transient states. We now give two important lemmas on absorbing Markov chains. For (A3) to hold, it is trivial to note that  $\mathbf{B} \neq \mathbf{0}$  and, thus,  $\mathbf{P}$  is a substochastic matrix.

**Lemma 17.3** *If  $\mathbf{P}$  is a substochastic matrix, then*

$$\rho(\mathbf{P}) < 1. \quad (17.78)$$

*Proof* For a proof, see [51].

From Lemma 17.3, the convergence of (17.74) is already established in (17.43).

All we need to show is that  $(\mathbf{I} - \mathbf{P})^{-1} \mathbf{B} \mathbf{U}$  corresponds to the exact coordinates of the sensors written in terms of the coordinates of the anchors. To this end, let

$$\mathbf{C}^* = \begin{bmatrix} \mathbf{U} \\ \mathbf{X}^* \end{bmatrix} \quad (17.79)$$

denote the matrix of the exact coordinates. Then from (17.67), it is trivial to show that (17.79) is the fixed point of the matrix form (17.23) of the localization algorithm (17.74), that is,

$$\begin{bmatrix} \mathbf{U} \\ \mathbf{X}^* \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ \mathbf{B} & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{X}^* \end{bmatrix}, \quad (17.80)$$

and we can write the exact coordinates of the sensors,  $\mathbf{X}^*$ , as

$$\mathbf{X}^* = (\mathbf{I}_{m+1} - \mathbf{P})^{-1} \mathbf{B} \mathbf{U}. \quad (17.81)$$

Hence, the convergent state,  $\lim_{t \rightarrow \infty} \mathbf{X}(t + 1)$ , is indeed the exact solution of the sensors' positions in terms of the anchors.

## 17.9 LINEAR SYSTEM OF EQUATIONS: DISTRIBUTED ALGORITHM

In this section, we solve a linear system of equations in a distributed, iterative fashion, using the general consensus algorithms. In particular, we are interested in solving

$$\mathbf{D}\mathbf{X} = \mathbf{U}, \quad (17.82)$$

where  $\mathbf{X} \in \mathbb{R}^{M \times m}$  denotes the unknown state of a sensor network that follows (17.82). The system matrix,  $\mathbf{D} \in \mathbb{R}^{M \times M}$ , is strict diagonally dominant and sparse, and the

anchors have the fixed state,  $\mathbf{U} \in \mathbb{R}^{M \times m}$ . In this case, we note that the number of anchors,  $n$ , is equal to the number of sensors,  $M$ , that is,  $n = M$ . Hence the total number of nodes in the network is  $N = 2M$ . Linear systems of equations appear naturally in sensor networks, for example, power flow equations in power systems monitored by sensors [70] or time synchronization algorithms in sensor networks [71].

### 17.9.1 Design of Iteration Matrix $\Upsilon$

Let  $\mathbf{M} = \text{diag}(\mathbf{D})$ . We make the following design choice:

$$\mathbf{B} = \mathbf{M}^{-1}, \quad (17.83)$$

$$\mathbf{P} = \mathbf{M}^{-1}(\mathbf{M} - \mathbf{D}). \quad (17.84)$$

With the above design choice the iteration matrix  $\Upsilon$  is given by

$$\Upsilon = \begin{bmatrix} \mathbf{I}_M & \mathbf{0} \\ \mathbf{M}^{-1} & \mathbf{M}^{-1}(\mathbf{M} - \mathbf{D}) \end{bmatrix}. \quad (17.85)$$

At each sensor,  $l$ , we note that the  $l$ th row,  $\mathbf{p}_l$ , of the matrix  $\mathbf{P}$  in (17.84) is a function of the  $l$ th row,  $\mathbf{d}_l$ , of the system matrix,  $\mathbf{D}$ , and the  $l$ th diagonal element,  $m_{ll}$ , of the diagonal matrix,  $\mathbf{M}^{-1}$ . With (17.84), the sparsity pattern of  $\mathbf{P}$  is the same as the sparsity pattern of the system matrix  $\mathbf{D}$ , since  $\mathbf{M}$  is a diagonal matrix. Hence, the underlying sensor communication graph  $\mathcal{G}$  comes directly from the graph  $\mathcal{G}(\mathbf{D})$  associated to the system matrix  $\mathbf{D}$ . The nonzero elements of the system matrix  $\mathbf{D}$  thus establish the interconnections among the sensors. The reason for assuming a sparse system matrix  $\mathbf{D}$  is apparent here since a full system matrix  $\mathbf{D}$  will result in an all-to-all communication graph among the sensors. Each sensor is, further, directly connected to exactly one anchor. The anchors in this case can be considered as dummy anchors, with their states being available at each sensor they are connected to in the graph  $\mathcal{G}(\Upsilon)$  associated to  $\Upsilon$ .

### 17.9.2 Convergence

To establish the convergence, we give the following lemma.

**Lemma 17.4** *Let  $\mathbf{D}$  be a strict diagonally dominant matrix. Let the matrix  $\mathbf{P}$  be given by (17.84). Then*

$$\rho(\mathbf{P}) < 1. \quad (17.86)$$

*Proof* The proof is straightforward and relies on Gershgorin's circle theorem [72].

With Lemma 17.4, we note that a distributed iterative algorithm with  $\mathbf{B}$  and  $\mathbf{P}$  as given in (17.83) and (17.84), respectively, converges to

$$\begin{aligned} \lim_{t \rightarrow \infty} \mathbf{X}(t+1) &= (\mathbf{I}_M - \mathbf{P})^{-1} \mathbf{B} \mathbf{U}, \\ &= (\mathbf{I}_M - \mathbf{M}^{-1}(\mathbf{M} - \mathbf{D}))^{-1} \mathbf{M}^{-1} \mathbf{U}, \\ &= (\mathbf{I}_M - \mathbf{I}_M + \mathbf{M}^{-1} \mathbf{D})^{-1} \mathbf{M}^{-1} \mathbf{U}, \\ &= \mathbf{D}^{-1} \mathbf{M} \mathbf{M}^{-1} \mathbf{U}, \\ &= \mathbf{D}^{-1} \mathbf{U}. \end{aligned} \quad (17.87)$$

Hence, the sensors asymptotically reach the solution of (17.82) in a distributed fashion.

### 17.9.3 Remarks

The distributed algorithm given in (17.23) with the matrices  $\mathbf{B}$  and  $\mathbf{P}$  given in (17.83) and (17.84) is the matrix extension of the well-known Jacobi algorithm [47]. The distributed algorithm (17.23), when implemented with the given matrices  $\mathbf{B}$  and  $\mathbf{P}$ , thus, gives a sensor network framework for solving the linear system of equations (17.82). The Jacobi algorithm can be further specialized to sparse symmetric, positive definite matrices  $\mathbf{D}$ ; see [47]. When these matrices have the special structure of being banded, an iterative distributed algorithm with a collapse operator is provided in [74]. The collapse operator adds a collapse step to the general model in (17.23) that exploits structural properties of banded matrices to make the algorithm computationally more efficient.

## 17.10 CONCLUSIONS

In this chapter, we presented a unifying view of commonly used linear distributed iterative algorithms in sensor networks. The notion of “consensus” has been described in a broader framework in higher dimensions that allows us to use the same set of tools for the treatment of average-consensus, multiagent coordination, sensor localization, and distributed algorithms to solve linear systems of algebraic equations. Throughout the development, we assumed the environment to be deterministic, in the sense that intersensor communication is noiseless and there is no uncertainty in the algorithm parameters. In a practical wireless sensor network application, the sensing environment is, in general, random, leading to communication link failures among the sensors, randomness in the system parameters, and quantized data exchanges due to bandwidth restrictions. Under broad assumptions of environment uncertainty, the framework in this chapter can be extended to account for these random phenomena; see, for example [31, 34], which treat the average-consensus problem under imperfect communication; and, for sensor localization, see [46], which treats the scenario when noisy intersensor distance measurements are available and intersensor communication is imperfect.

Another aspect that deserves special attention in the design of distributed algorithms, but is not treated explicitly in this chapter, involves optimizing the system parameters to improve the convergence rate of these iterative algorithms. Since in a distributed setting, there is no central location where all the information is collected or available, this optimization should be carried out in a distributed manner. In the context of gossip (or average-consensus) algorithms, [26] presents a distributed algorithm for optimizing the gossip probabilities to maximize the convergence rate, using a distributed spectral analysis protocol developed in [74]. An interesting future research direction is to design such distributed optimization algorithms for improving the convergence rate of the generic “consensus” algorithms considered in this chapter.

## REFERENCES

1. M. D. Ilić, L. Xie, U. A. Khan, and J. M. F. Moura, “Modelling future cyberphysical energy systems,” paper presented at the IEEE Power Engineering Society General Meeting, Pittsburgh, PA, July 2008.

2. M. D. Ilić, L. Xie, U. A. Khan, and J. M. F. Moura, “Modeling, sensing and control of future cyber-physical energy systems,” *IEEE Trans. Syst. Man Cybernet. Spec. Issue Eng. Cyber-Phys. Ecosyst.*, Jul. 2008, submitted for publication.
3. U. A. Khan, D. Ilić, and J. M. F. Moura, “Cooperation for aggregating complex electric power networks to ensure system observability,” paper presented at the IEEE International Conference on Infrastructure Systems, Rotterdam, Netherlands, Nov. 2008, accepted for publication.
4. R. R. Tenney and N. R. Sandell, “Detection with distributed sensors,” *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-17, no. 4, pp. 501–510, July 1981.
5. J. N. Tsitsiklis, “Problems in decentralized decision making and computation,” PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1984.
6. J. N. Tsitsiklis, “Decentralized detection by a large number of sensors,” *Math. Control Signals Syst.*, vol. 1, no. 2, pp. 167–182, 1988.
7. P. K. Varshney, *Distributed Detection and Data Fusion*, Secaucus, NJ: Springer, 1996.
8. J.-F. Chamberland and V. V. Veeravalli, “Decentralized detection in sensor networks,” *IEEE Trans. Signal Processing*, vol. 51, pp. 407–416, Feb. 2003.
9. S. Aldosari and J. M. F. Moura, “Detection in sensor networks: The saddlepoint approximation,” *IEEE Trans. Signal Process.*, vol. 55, no. 1, pp. 327–340, Jan. 2007.
10. J. N. Tsitsiklis and M. Athans, “On the complexity of decentralized decision making and detection problems,” *IEEE Trans. Automatic Control*, vol. AC-30, no. 5, pp. 440–446, May 1985.
11. J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *IEEE Trans. Automatic Control*, vol. AC-31, no. 9, pp. 803–812, Sept. 1986.
12. H. J. Kushner and G. Yin, “Asymptotic properties of distributed and communicating stochastic approximation algorithms,” *Siam J. Control Optimization*, vol. 25, no. 5, pp. 1266–1290, Sept. 1987.
13. A. Jadbabaie, J. Lin, and A. S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” *IEEE Trans. Automatic Control*, vol. AC-48, no. 6, pp. 988–1001, June 2003.
14. R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *IEEE Trans. Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sept. 2004.
15. L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Syst. Control Lett.*, vol. 53, pp. 65–78, 2004.
16. S. Kar, S. A. Aldosari, and J. M. F. Moura, “Topology for distributed inference on graphs,” *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2609–2613, June 2008.
17. M. J. Fischer, N. A. Lynch, and M. S. Paterson, “Impossibility of distributed consensus with one faulty process,” *J. Assoc. Comput. Machin.*, vol. 32, no. 2, pp. 374–382, Apr. 1985.
18. N. A. Lynch, *Distributed Algorithms*, San Francisco, CA: Morgan Kaufmann, 1997.
19. G. V. Cybenko, “Dynamic load balancing for distributed memory multiprocessors,” *J. Parallel Distrib. Comput.*, vol. 7, pp. 279–301, 1989.
20. C. Reynolds, “Flocks, birds, and schools: A distributed behavioral model,” *Computer Graphics*, vol. 21, pp. 25–34, 1987.
21. T. Vicsek, A. Czirok, E. Ben Jacob, I. Cohen, and O. Schochet, “Novel type of phase transitions in a system of self-driven particles,” *Phys. Rev. Lett.*, vol. 75, pp. 1226–1229, 1995.
22. W. Ren, R. W. Beard, and E. M. Atkins, “A survey of consensus problems in multi-agent coordination,” paper presented at the American Control Conference, Portland, OR, June 2005, pp. 1859–1864.

23. R. Olfati-Saber, J. Alex Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *IEEE Proc.*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
24. Y. Hatano and M. Mesbahi, "Agreement over random networks," paper presented at the 43rd IEEE Conference on Decision and Control, Vol. 2, Dec. 2004, pp. 2010–2015.
25. Y. Hatano, A. K. Das, and M. Mesbahi, "Agreement in presence of noise: Pseudogradients on random geometric networks," paper presented at the 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference, CDC-ECC '05, Seville, Spain, Dec. 2005.
26. S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Trans. Networks*, vol. 14, no. SI, pp. 2508–2530, 2006.
27. S. Kar and J. M. F. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3315–3326, July 2008.
28. M. E. Yildiz and A. Scaglione, "Differential nested lattice encoding for consensus problems," in *ACM/IEEE Information Processing in Sensor Networks*, Cambridge, MA, Apr. 2007.
29. A. T. Salehi and A. Jadbabaie, "On consensus in random networks," paper presented at the The Allerton Conference on Communication, Control, and Computing, Allerton House, IL, Sept. 2006.
30. M. Porfiri and D. J. Stilwell, "Stochastic consensus over weighted directed networks," in *Proceedings of the 2007 American Control Conference*, New York, July 11–13, 2007.
31. S. Kar and José M. F. Moura, "Distributed consensus algorithms in sensor networks: Link failures and channel noise," *IEEE Trans. Signal Process.*, 2008, accepted for publication.
32. A. Kashyap, T. Basar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, pp. 1192–1203, July 2007.
33. T. C. Aysal, M. J. Coates, and M. G. Rabbat, "Distributed average consensus with dithered quantization," *IEEE Trans. Signal Process.*, 2008, to appear.
34. S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Quantized data," available: <http://arxiv.org/abs/0712.1609>, Nov. 2007.
35. A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," Technical Report 2778, LIDS-MIT, Nov. 2007.
36. P. Frasca, R. Carli, F. Fagnani, and S. Zampieri, "Average consensus on networks with quantized communication," *Int. J. Robust Nonlinear Control*, 2008, submitted for publication.
37. M. Huang and J. H. Manton, "Stochastic approximation for consensus seeking: Mean square and almost sure convergence," in *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, LA, Dec. 12–14, 2007.
38. R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.
39. H. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Trans. Automatic Control*, vol. 52, no. 5, pp. 863–868, 2007.
40. U. A. Khan and J. M. F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. Signal Process.*, vol. 56, part 1, no. 10, pp. 4919–4935, Oct. 2008.
41. I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus-based distributed parameter estimation in ad hoc wireless sensor networks with noisy links," in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, Honolulu, HI, 2007, pp. 849–852.
42. R. Olfati-Saber, "Distributed Kalman filters with embedded consensus filters," paper presented at the 44th IEEE Conference on Decision and Control, Seville, Spain, Dec. 2005, pp. 8179–8184.

43. M. Alanyali and V. Saligrama, "Distributed tracking in multi-hop networks with communication delays and packet losses," paper presented at the 13th IEEE Workshop on Statistical Sig. Proc., Bordeaux, France, July 2005, pp. 1190–1195.
44. S. Kar, J. M. F. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," available: <http://arxiv.org/abs/0809.0009>, Aug. 2008.
45. A. Rahmani and M. Mesbahi, "Pulling the strings on agreement: Anchoring, controllability, and graph automorphism," paper presented at the American Control Conference, New York City, July 11–13, 2007, pp. 2738–2743.
46. U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Trans. Signal Process.*, submitted for publication, see <http://arxiv.org/abs/0802.3563>, Aug. 2008.
47. D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computations*, Englewood Cliffs, NJ: Prentice-Hall, 1989.
48. F. R. K. Chung, *Spectral Graph Theory*, Providence, RI: American Mathematical Society, 1997.
49. B. Bollobás, *Modern Graph Theory*, New York: Springer Verlag, 1998.
50. B. Mohar, "The Laplacian spectrum of graphs," in *Graph Theory, Combinatorics, and Applications*, Vol. 2, Y. Alavi, G. Chartrand, O. R. Oellermann, and A. J. Schwenk (Eds.), New York: Wiley, 1991, pp. 871–898.
51. A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, New York: Academic, 1970.
52. R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge: Cambridge University Press, 1985.
53. P. Lancaster and M. Tismenetsky, *The Theory of Matrices, Second Edition with Applications, Computer Science and Applied Mathematics*, New York: Academic, 1985.
54. C. M. Grinstead and J. L. Snell, *Introduction to Probability*, American Mathematical Society, 1997.
55. H. L. Van Trees, *Detection, Estimation, and Modulation Theory: Part I*, New York: Wiley, 1968.
56. R. R. Tenney and N. R. Sandell, Jr., "Detection with distributed sensors," *IEEE Trans. Aerospace Electron. Syst.*, vol. 17, no. 4, pp. 501–510, 1981.
57. J. N. Tsitsiklis, "Decentralized detection," in *Advances in Statistical Signal Processing*, H. V. Poor and J. B. Thomas (Eds.) Greenwich, CT: JAI Press, 1993, pp. 297–344.
58. P. K. Willett, P. F. Swaszek, and R. S. Blum, "The good, bad, and ugly: Distributed detection of a known signal in dependent Gaussian noise," *IEEE Trans. Signal Process.*, vol. 48, no. 12, pp. 3266–3279, 2000.
59. S. Aldosari and J. M. F. Moura, "Fusion in sensor networks with communication constraints," in *IPSN04 Information Processing in Sensor Networks*, Berkeley CA, Apr. 2004, IEEE/ACM, pp. 108–115.
60. S. Aldosari and J. M. F. Moura, "Detection in decentralized sensor networks," in *ICASSP04, IEEE International Conference on Signal Processing*, Vol. II, Montreal, Québec, Canada, May 17–21 2004, New York: IEEE, pp. 277–280.
61. L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Controls Lett.*, vol. 53, no. 1, pp. 65–78, Apr. 2004.
62. S. A. Aldosari and J. M. F. Moura, "Distributed detection in sensor networks: connectivity graph and small-world networks," paper presented at the 39th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, Oct. 2005, pp. 230–234.

63. S. A. Aldosari and J. M. F. Moura, "Topology of sensor networks in distributed detection," paper presented at the ICASSP'06, IEEE International Conference on Signal Processing, Vol. 5, Toulouse, France, May 2006, pp. 1061–1064.
64. F. R. Gantmacher, *Matrix Theory*, Vol. I, Chelsea Publishing 1959.
65. R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
66. G. E. Shilov and R. A. Silverman, *Linear Algebra*, Courier Dover, 1977.
67. M. J. Sippl and H. A. Scheraga, "Cayley–Menger coordinates," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 83, no. 8, pp. 2283–2287, Apr. 1986.
68. G. Springer, *Introduction to Riemann Surfaces*, Reading, MA: Addison-Wesley, 1957.
69. J. G. Hocking and G. S. Young, *Topology*, Reading, MA: Addison-Wesley, 1961.
70. L. S. Zurlo, P. E. Mercado, and C. E. de la Vega, "Parallelization of the linear load flow equations," *Power Tech. Proc. 2001 IEEE Porto*, vol. 3, p. 5, 2001.
71. L. Schenato and G. Gamba, "A distributed consensus protocol for clock synchronization in wireless sensor network," paper presented at the Decision and Control, 2007 46th IEEE Conference on, Dec. 2007, pp. 2289–2294.
72. G. Golub and C. Van Loan, *Matrix Computations*, Baltimore, MD: Johns Hopkins University Press, 1996.
73. U. A. Khan and J. M. F. Moura, "Distributed Iterate-Collapse inversion (DICI) algorithm for  $L$ -banded matrices," paper presented at the 33rd IEEE International Conference on Acoustics, Speech, and Signal Processing, Las Vegas, NV, Mar. 30–Apr. 4, 2008.
74. D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," in *STOC '04: Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, New York: ACM, 2004, pp. 561–568.



## CHAPTER 18

---

# Cooperative Sensor Communications

Ahmed K. Sadek<sup>1</sup>, Weifeng Su<sup>2</sup>, and K. J. Ray Liu<sup>3</sup>

<sup>1</sup>Corporate Research and Development, Qualcomm Incorporated, San Diego, California

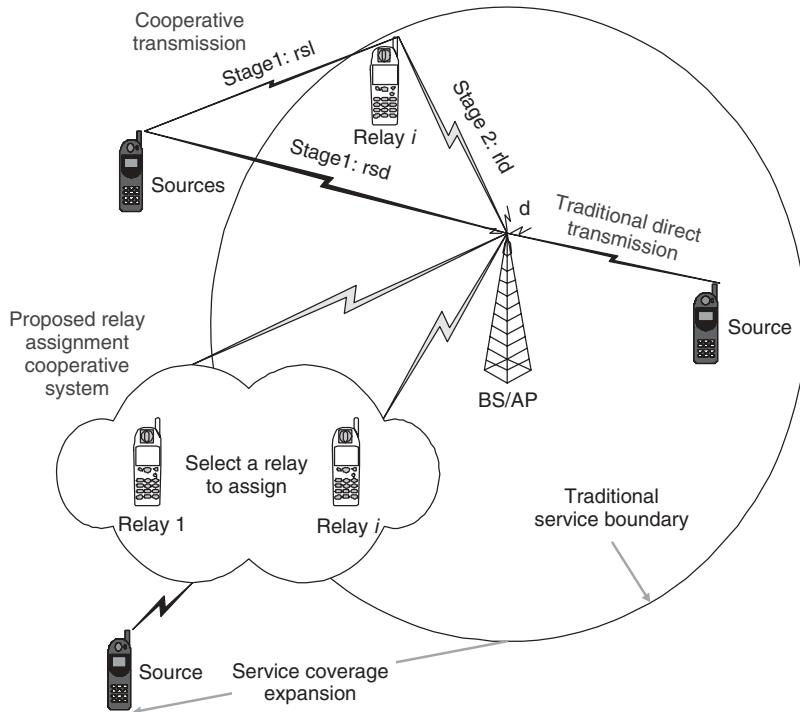
<sup>2</sup>State University of New York, Department of Electrical Engineering, Buffalo, New York

<sup>3</sup>University of Maryland, Department of Electrical and Computer Engineering, College Park, Maryland

### 18.1 INTRODUCTION

The unprecedented improvements shown by electronic devices in terms of computing power per unit area, communication capacity, and energy efficiency are fueling the increased pervasiveness of networks of a large number of small devices that collaborate in a distributed fashion to perform some functions. Among this type of networks, the dense sensor network presents unique characteristics and research problems. Of these, this chapter focuses on the inherent limited energy available to the component nodes and of the possibility of the nodes cooperating during both sensing and communication. One of the most severe impairments to wireless communications is channel fading. Fading causes a significant loss in the transmitted power compared to the benign additive white Gaussian noise (AWGN) channel because, when the signal experiences a deep fade, the receiver can not decode it. A widely used technique to combat the effects of channel fading is diversity. Spatial diversity achieved through multiple antennas installed at the transmitter and/or the receiver have gained tremendous interest and are included in the design of future wireless macronetworks. The gains of multiple input–multiple output (MIMO) systems in terms of increasing the channel capacity, higher throughput, improved error performance, and better energy efficiency are well established by now. In practice, however, installing multiple antennas on a device might not be feasible because of space or cost limitations. Besides, to achieve full diversity gains in MIMO, there must be sufficient separation between the antenna elements at the transmitter and receiver sides, which is difficult to achieve in practice. This will result in the fades of the channels between different antenna pairs to be correlated, which can reduce the diversity gains of the system. The above problems are more pronounced in sensor networks because of the compact sensor sizes.

To overcome the above limitations of achieving MIMO gains in future wireless networks, we must think of new techniques beyond traditional point-to-point communications. The traditional view of a wireless system is that it is a set of nodes trying to communicate with each other. However, from another point of view, because of the



**Figure 18.1** Illustrating the difference between the direct and cooperative transmission schemes and the coverage extension prospected by cooperative transmission.

broadcast nature of wireless channel, we can think of those nodes as a set of antennas distributed in the wireless system. Adopting this point of view, nodes in the network can cooperate together for distributed transmission and processing of information. The cooperating node acts as a relay node for the source node.

Cooperative communications is a new communication paradigm that generates independent paths between the user and the base station via introducing a relay channel as illustrated in Figure 18.1 [1–3]. The relay channel can be thought of as an auxiliary channel to the direct channel between the source and destination [4, 5]. Since the relay node is usually several wavelengths far from the source, the relay channel is guaranteed to fade independently from the direct channel, which introduces a full rank MIMO channel between the source and the destination. Hence, cooperative communications is a new paradigm shift for the fourth-generation wireless system that will guarantee high data rates to all users in the network, and we anticipate it will be the key technology aspect in the fifth-generation wireless networks [6].

In this chapter, we first lay the fundamental definitions of cooperative communications and discuss some of the most popular relaying techniques. We then consider the symbol error rate performance of some relaying protocols along with the optimal power allocation problem between the source and relay nodes. The rest of the chapter focuses on the energy efficiency of cooperative communications in sensor networks. In particular, an analytical framework for studying the energy efficiency of cooperation in wireless networks is presented. In this framework, we consider the overhead in the processing and receiving power introduced by cooperation. By taking into consideration

such overhead, we study the trade-off in the gains provided by cooperation in the form of a reduction in the transmit power, due to the spatial diversity gain, and the increase in the receiving and processing power that results from the operation of the relay. This trade-off is shown to depend on many parameters such as the values of the receive and processing powers, the application, the power amplifier loss, and several other factors. The results reveal an interesting threshold behavior; below a certain threshold distance between the source and destination direct transmission becomes more energy efficient than cooperation. The results also provide guidelines for the design of power allocation strategies, relay assignment algorithms, and the selection of the optimal number of relays to help the source.

## 18.2 COOPERATIVE RELAY PROTOCOLS

In this section, we explain what is the relay channel and in what aspects it is different from the direct point-to-point channel. We also describe several protocols that can be implemented at the relay channel, and the performance of these protocols is assessed based on their outage probability and diversity gains.

### 18.2.1 Cooperative Communications

Cooperative communications protocols can be generally categorized into fixed relaying schemes and adaptive relaying schemes [1]. In fixed relaying, the channel resources are divided between the source and the relay in a fixed (deterministic) manner. The processing at the relay differs according to the employed protocol. In the amplify-and-forward (AF) protocol, the relay simply scales the received version and transmits an amplified version of it to the destination. Another possibility of processing at the relay node is for the relay to decode the received signal, reencode it, and then retransmit it to the receiver. This kind of relaying is termed as decode-and-forward (DF) protocol.

Fixed relaying has the advantage of easy implementation, but the disadvantage of low bandwidth efficiency. This is because half of the channel resources are allocated to the relay for transmission, which reduces the overall rate. This is true especially when the source–destination channel is not very bad because under such scenario a high percentage from the packets transmitted by the source to the destination can be received correctly by the destination and the relay transmissions are wasted. Adaptive relaying techniques try to overcome this problem. Adaptive relaying techniques comprise selective and incremental relaying.

In selective relaying, the relay and the source are assumed to know the fade of the channel between them, and if the signal-to-noise ratio of the signal received at the relay exceeds a certain threshold, the relay performs DF operation on the message. On the other hand, if the channel between the source and the relay falls below the threshold, the relay idles. Furthermore, assuming reciprocity in the channel, the source also knows that the relay idles, and the source transmits a copy of its signal to the destination instead. For the second adaptive relaying protocol, namely incremental relaying, a feedback channel from the destination to the relay is utilized. The destination feeds back an acknowledgment to the relay if it was able to receive the source's message correctly in the first transmission phase, and the relay does not need to transmit then.

In this section, we will consider the performance of the above cooperation protocols in terms of outage capacity. Outage capacity given that the channel is required to

support a transmission rate  $R$  is defined as

$$\Pr [I(x, y) \leq R],$$

where  $I(x, y)$  is the mutual information of a channel with input  $x$  and  $y$  is the channel output. Note that the mutual information is a random variable because the channel varies in a random way due to fading.

Since, in practice, a device cannot listen and transmit simultaneously, a half-duplex constraint is assumed throughout the chapter, meaning that the relay cannot transmit and receive at the same time because, if full-duplex is permitted, then the transmitting signal will cause severe interference to the incoming weak received signal.

### 18.2.2 Cooperation Protocols

A simple cooperation strategy can be modeled with two phases:

- In phase 1, a source sends information to its destination, and the information is also received by the relay at the same time.
- In phase 2, the relay can help the source by forwarding or retransmitting the information that it receives in phase 1.

This scenario is depicted in Figure 18.2. Cooperation is usually done in two orthogonal phases, either in time division multiple access (TDMA) or frequency division multiple access (FDMA), to avoid interference between the two phases. Figure 18.2 depicts a general relay channel, where the source transmits with power  $P_1$  and the relay transmits with power  $P_2$ . We will consider the special case where the source and the relay transmits with equal power  $P$ . Optimal power allocation is studied in the following section.

### 18.2.3 Fixed Cooperation Strategies

In fixed relaying, the channel resources are divided between the source and the relay in a fixed (deterministic) manner. The processing at the relay differs according to the employed protocol. The most common techniques for fixed relaying are AF and DF protocols.

**18.2.3.1 Amplify-and-Forward Protocol** In AF protocol, the relay simply scales the received version and transmits an amplified version of it to the destination. The AF relay channel can be modeled as follows. The signal transmitted from the source

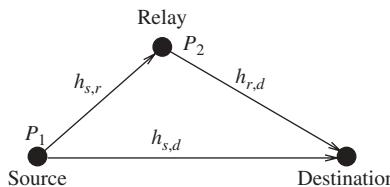


Figure 18.2 Simplified cooperation model.

$x$  is received at both the relay and destination as

$$y_{s,r} = \sqrt{P}h_{s,r}x + n_{s,r}, \quad \text{and} \quad y_{s,d} = \sqrt{P}h_{s,d}x + n_{s,d}, \quad (18.1)$$

where  $h_{s,r}$  and  $h_{s,d}$  are the channel fades between the source and the relay and destination, respectively, and are modeled as Rayleigh flat fading channels with variances  $\sigma_{s,r}^2$  and  $\sigma_{s,d}^2$ , respectively. The terms  $n_{s,r}$  and  $n_{s,d}$  denote the additive white Gaussian noise with zero mean and variance  $N_o$ . In this protocol, the relay amplifies the signal from the source and forwards it to the destination ideally to equalize the effect of the channel fade between the source and the relay. The relay does that by simply scaling the received signal by a factor that is inversely proportional to the received power, which is denoted by

$$\beta_r = \frac{\sqrt{P}}{\sqrt{P|h_{s,r}|^2 + N_o}}. \quad (18.2)$$

The signal transmitted from the relay is thus given by  $\beta_r y_{s,r}$  and has power  $P$  equal to the power of the signal transmitted from the source. To calculate the mutual information between the source and the destination, we need to calculate the total instantaneous signal-to-noise ratio (SNR) at the destination. The SNR received at the destination is the sum of the SNRs from the source and relay links. The SNR from the source link is given by

$$\text{SNR}_{s,d} = \Gamma|h_{s,d}|^2, \quad (18.3)$$

where  $\Gamma = P/N_o$ .

In the following we show the calculations for the received SNR from the relay link. In phase 2 the relay amplifies the received signal and forwards it to the destination with transmitted power  $P$ . The received signal at the destination in phase 2 according to (18.2) is given by

$$y_{r,d} = \frac{\sqrt{P}}{\sqrt{P|h_{s,r}|^2 + N_o}} h_{r,d} y_{s,r} + n_{r,d}, \quad (18.4)$$

where  $h_{r,d}$  is the channel coefficient from the relay to the destination and  $n_{r,d}$  is an additive noise. More specifically, the received signal  $y_{r,d}$  in this case is

$$y_{r,d} = \frac{\sqrt{P}}{\sqrt{P|h_{s,r}|^2 + N_o}} \sqrt{P} h_{r,d} h_{s,r} x + n'_{r,d}, \quad (18.5)$$

where

$$n'_{r,d} = \frac{\sqrt{P}}{\sqrt{P|h_{s,r}|^2 + N_o}} h_{r,d} n_{s,r} + n_{r,d}. \quad (18.6)$$

Assume that the noise terms  $n_{s,r}$  and  $n_{r,d}$  are independent; then the equivalent noise  $n'_{r,d}$  is a zero-mean complex Gaussian random variable with variance

$$N'_o = \left( \frac{P|h_{r,d}|^2}{P|h_{s,r}|^2 + N_o} + 1 \right) N_o. \quad (18.7)$$

The destination receives two copies from the signal  $x$  through the source link and relay link. There are different techniques to combine the two signals, and the optimal technique that maximizes the overall SNR is maximal-ratio combiner (MRC). MRC combining requires a coherent detector that has knowledge of all channel coefficients, and the SNR at the output of the MRC is equal to the sum of the received SNR from both branches.

With knowledge of the channel coefficients  $h_{s,d}$ ,  $h_{s,r}$ , and  $h_{r,d}$ , the output of the MRC detector at the destination can be written as

$$y = a_1 y_{s,d} + a_2 y_{r,d}. \quad (18.8)$$

The combining factors  $a_1$  and  $a_2$  should be designed to maximize the combined SNR. This can be solved by formulating an optimization problem and selecting these factors correspondingly. An easier way to design them is by resorting to signal space and detection theory principles. Since, the AWGN terms span the whole space, to minimize the noise effects the detector should project the received signals  $y_{s,d}$  and  $y_{s,r}$  to the desired signal spaces. Hence,  $y_{s,d}$  and  $y_{r,d}$  should be projected along the directions of  $h_{sd}$  and  $h_{rd}h_{sr}$ , respectively, after normalizing the noise variance terms in both received signals. Therefore  $a_1$  and  $a_2$  are given by

$$a_1 = \frac{\sqrt{P} h_{s,d}^*}{N_o} \quad \text{and} \quad a_2 = \frac{\sqrt{\frac{P}{P|h_{s,r}|^2+N_o}} \sqrt{P} h_{s,r}^* h_{r,d}^*}{\left( \frac{P|h_{r,d}|^2}{P|h_{s,r}|^2+N_o} + 1 \right) N_o}. \quad (18.9)$$

By assuming that the transmitted symbol  $x$  in (18.1) has average energy 1, the instantaneous SNR of the MRC output is

$$\gamma = \gamma_1 + \gamma_2, \quad (18.10)$$

where

$$\gamma_1 = \frac{|a_1 \sqrt{P} h_{s,d}|^2}{|a_1|^2 N_o} = P|h_{s,d}|^2 / N_o, \quad (18.11)$$

and

$$\begin{aligned} \gamma_2 &= \frac{|a_2 \frac{\sqrt{P}}{\sqrt{P|h_{s,r}|^2+N_o}} \sqrt{P} h_{r,d} h_{s,r}|^2}{N'_o |a_2|^2} = \frac{\frac{P^2}{P|h_{s,r}|^2+N_o} |h_{s,r}|^2 |h_{r,d}|^2}{\left( \frac{P|h_{r,d}|^2}{P_1|h_{s,r}|^2+N_o} + 1 \right) N_o} \\ &= \frac{1}{N_o} \frac{P^2 |h_{s,r}|^2 |h_{r,d}|^2}{P|h_{s,r}|^2 + P|h_{r,d}|^2 + N_o}. \end{aligned} \quad (18.12)$$

From the above, the instantaneous mutual information as a function of the fading coefficients for amplify-and-forward is given by

$$I_{AF} = \frac{1}{2} \log(1 + \gamma_1 + \gamma_2). \quad (18.13)$$

Substituting for the values of the SNR of both links, we can write the mutual information as

$$I_{AF} = \frac{1}{2} \log \left( 1 + \Gamma |h_{sd}|^2 + f(\Gamma |h_{sr}|^2, \Gamma |h_{rd}|^2) \right), \quad (18.14)$$

where

$$f(x, y) \triangleq \frac{xy}{x + y + 1}. \quad (18.15)$$

The outage probability can be obtained by averaging over the exponential channel gain distribution as follows:

$$\Pr[I_{\text{AF}} < R] = E_{h_{s,d}, h_{s,r}, h_{r,d}} \left[ \frac{1}{2} \log \left( 1 + \Gamma |h_{s,d}|^2 + f(\Gamma |h_{s,r}|^2, \Gamma |h_{r,d}|^2) \right) < R \right]. \quad (18.16)$$

Calculating the above integration, the outage probability at high SNR is given by

$$\Pr[I_{\text{AF}} < R] \simeq \left[ \frac{\sigma_{s,r}^2 + \sigma_{rd}^2}{2\sigma_{s,d}^2 (\sigma_{s,r}^2 \sigma_{r,d}^2)} \right] \left( \frac{2^{2R} - 1}{\Gamma} \right)^2, \quad (18.17)$$

where the multiplicative factor of 2 in  $2R$  is because half of the bandwidth is lost in cooperation by allocating them to the relay. The outage expression decays as  $\Gamma^{-2}$ , which means that the AF protocol achieves diversity 2.

**18.2.3.2 Decode-and-Forward Protocol** Another possibility of processing at the relay node is for the relay to decode the received signal, reencode it, and then retransmit it to the receiver. This kind of relaying is termed a decode-and-forward (DF) scheme. If the decoded signal at the relay is denoted by  $\hat{x}$ , the transmitted signal from the relay can be denoted by  $\sqrt{P}\hat{x}$ , given that  $\hat{x}$  has unit variance. The DF scheme requires the correct decoding at the relay node, otherwise the signal is considered decoded in error at the destination. It is clear that for such a scheme the diversity achieved is one because the performance of the system is limited by the worst link from the source–relay and source–destination. This will be illustrated through the numerical examples. In the sequel we focus on an adaptive form of DF known as selective DF, which avoids this problem.

#### 18.2.4 Adaptive Cooperation Strategies

Fixed relaying suffers from deterministic loss in the transmission rate. In other words, there is always 50% loss in the spectral efficiency because transmission is done in two phases. To overcome this problem, adaptive relaying protocols can be developed to improve this inefficiency. Adaptive relaying protocols comprise two strategies: selective and incremental relaying.

**18.2.4.1 Selective Relaying** In selective relaying, the relay and the source are assumed to know the channel state condition between them, and if the SNR of the signal received at the relay exceeds a certain threshold, the relay performs DF on the message. On the other hand, if the channel between the source and the relay falls below the threshold, the relay idles. Furthermore, assuming reciprocity in the channel, the source also knows that the relay idles, and the source transmits a copy of its signal to the destination instead. Selective relaying improves upon the performance of DF, as the SNR threshold at the relay can be designed to overcome the inherent problem in DF that the relay is required to decode correctly.

If the SNR between the source and the relay is less than the specified threshold, the source transmits in the two communication phases. The received SNR at the destination

will be twice for direct transmission, while the rate will be half of that of direct transmission, which constitutes a loss in the multiplexing gain. On the other hand, if the SNR in the source–relay link exceeds the threshold, then the relay is able to decode the source’s signal reliably and the received SNR at the destination is the sum of the received SNR from the source and the relay, that is, exactly as in the DF case. According to the above, the mutual information for selective relaying is given by

$$I_{\text{SDF}} = \begin{cases} \frac{1}{2} \log(1 + 2\Gamma|h_{s,d}|^2), & |h_{s,r}|^2 < g(\Gamma), \\ \frac{1}{2} \log(1 + \Gamma|h_{s,d}|^2 + \Gamma|h_{r,d}|^2), & |h_{s,r}|^2 \geq g(\Gamma), \end{cases} \quad (18.18)$$

where  $g(\Gamma) = 2^{2R} - 1/\Gamma$ .

The outage probability for selective relaying can be derived as follows. Using the law of total probability, conditioning on whether the relay forwards the source signal or not we have

$$\begin{aligned} P[I_{\text{SDF}} < R] &= P[I_{\text{SDF}} < R | |h_{s,r}|^2 < g(\Gamma)] \Pr[|h_{s,r}|^2 < g(\Gamma)] \\ &\quad + P[I_{\text{SDF}} < R | |h_{s,r}|^2 > g(\Gamma)] \Pr[|h_{s,r}|^2 > g(\Gamma)]. \end{aligned}$$

From (18.18), the outage probability for selective DF is given by

$$\begin{aligned} P[I_{\text{SDF}} < R] &= P\left[\frac{1}{2} \log(1 + 2\Gamma|h_{s,d}|^2) < R | |h_{s,r}|^2 < g(\Gamma)\right] \Pr[|h_{s,r}|^2 < g(\Gamma)] \\ &\quad + P\left[\frac{1}{2} \log(1 + \Gamma|h_{s,d}|^2 + \Gamma|h_{r,d}|^2) | |h_{s,r}|^2 > g(\Gamma)\right] \Pr[|h_{s,r}|^2 > g(\Gamma)]. \end{aligned}$$

From the above, the source can achieve diversity order 2 because the first term in the summation above is the product of two probabilities that each account for diversity order 1, and the second term in the summation has also a product of two terms, the first having diversity order 2. In other words, the selective relaying scheme can achieve diversity order 2 because in order for an outage event to happen, either both the source–destination and source–relay channels should be in outage or the combined source–destination and relay–destination channel should be in outage. All the random variables in the above expression are independent exponential random variables, which makes the calculation of the outage probability straightforward, and the outage expression at high SNR is given by

$$\Pr[I_{\text{SDF}} < R] \asymp \left[ \frac{\sigma_{s,r}^2 + \sigma_{r,d}^2}{2\sigma_{s,d}^2 (\sigma_{s,r}^2 \sigma_{r,d}^2)} \right] \left( \frac{2^{2R} - 1}{\Gamma} \right)^2, \quad (18.19)$$

which is identical to the AF case. This means that at high SNR, both selective relaying and AF have the same outage performance.

**18.2.4.2 Incremental Relaying** For incremental relaying, it is assumed that there is a feedback channel from the destination to the relay. The destination feeds back an acknowledgment to the relay if it was able to receive the source’s message correctly in the first transmission phase, and the relay does not need to transmit then. This protocol has the best spectral efficiency among the previously described protocols because the relay does not need to transmit always, and, hence, the second transmission phase

becomes opportunistic depending on the channel state condition of the direct channel between the source and the destination. Nevertheless, incremental relaying achieves diversity order 2 as shown below.

In incremental relaying, if the source transmission in the first phase was successful, then there is no second phase and the source transmits new information in the next time slot. On the other hand, if the source transmission was not successful in the first phase, the relay can use any of the fixed relaying protocols to transmit the source signal from the first phase. We will focus here on the relay utilizing AF protocol.

Note that the transmission rate is random in incremental relaying. If the first phase was successful, the transmission rate is  $R$ , while if the first transmission was in outage the transmission rate becomes  $R/2$  as in fixed relaying. The outage probability can be calculated as follows:

$$\Pr[I_{\text{IR}} < R] = \Pr[I_D < R] \Pr[I_{\text{AF}} < \frac{R}{2} | I_D < R]. \quad (18.20)$$

The outage expression for both direct transmission and AF relaying were computed before. As a function of the SNR and the rate  $R$ , the outage probability

$$\Pr[I_{\text{IR}} < R] = \Pr \left[ |h_{s,d}|^2 + \frac{1}{\Gamma} f(\Gamma |h_{s,r}|^2, \Gamma |h_{rd}|^2) \leq g(\Gamma) \right], \quad (18.21)$$

where  $g(\Gamma) = 2^R - 1/\Gamma$ . The spectral efficiency is  $R$  if the source–destination channel is not in outage, and  $R/2$  if the channel is not in outage. The average spectral efficiency is given by

$$\begin{aligned} \bar{R} &= R \Pr[|h_{s,r}|^2 \geq g(\Gamma)] + \frac{R}{2} \Pr[|h_{s,r}|^2 < g(\Gamma)] \\ &= \frac{R}{2} \left[ 1 + \exp \left( -\frac{2^R - 1}{\Gamma} \right) \right]. \end{aligned} \quad (18.22)$$

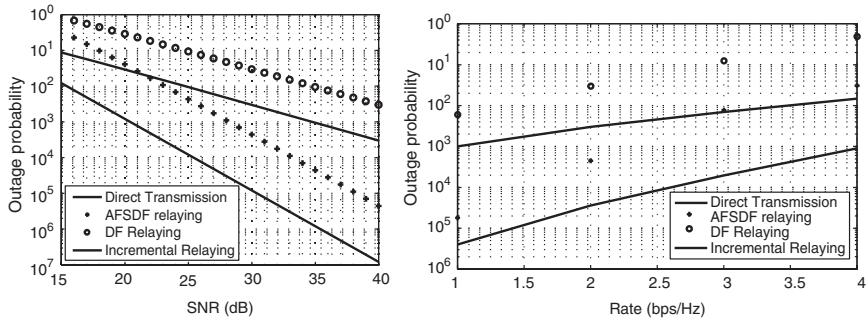
For large SNR, we have

$$\Pr[I_{\text{IAF}} < R] \asymp \left( \frac{1}{\sigma_{s,d}^2} \frac{\sigma_{s,r}^2 + \sigma_{r,d}^2}{\sigma_{s,r}^2 \sigma_{r,d}^2} \right) \left( \frac{2^{\bar{R}} - 1}{\text{SNR}} \right)^2. \quad (18.23)$$

**Example 18.1** Figure 18.3 compares the performance of the different relaying protocols discussed versus SNR and rate, respectively. Selective relaying has the same performance as AF relaying so it is not plotted. It is clear from both figures that incremental relaying has the best performance. This is because, incremental relaying operates at much higher spectral efficiency than the other relaying protocols and achieves full diversity gain of 2.

The simulation results also show that DF relaying offers only diversity gain of 1. In the following, we only consider selective relaying with DF operation at the relay node, and we refer to this scheme as DF relaying for convenience.

In the following section we study the symbol error rate performance of the single relay case. General  $M$ -PSK and  $M$ -QAM modulation are considered, and the symbol error rate formulas are used to derive optimal power allocation between the source and the relay. The Symbol Error Rate (SER) for the multiple relay scenario is addressed in [7, 8].



**Figure 18.3** Outage probability versus (a) SNR and (b) spectral efficiency for different relaying protocols.

### 18.3 SER ANALYSIS AND OPTIMAL POWER ALLOCATION

In the first part of this section, we analyze the SER performance and determine an asymptotic optimum power allocation for the DF cooperation systems in which we derive the closed-form SER formulations explicitly for the systems with  $M$ -PSK and  $M$ -QAM modulations, provide an approximation to reveal the asymptotic system performance, and determine an asymptotic optimum power allocation for the DF cooperation systems. In the second part of this section, we investigate the SER performance for the AF cooperation systems. First, we derive a simple closed-form Moment Generating Function (MGF) expression for the harmonic mean of two random variables. Then, based on the simple MGF expression, closed-form SER formulations are given for the AF cooperation systems. We also provide a tight SER approximation to show the asymptotic performance of the AF cooperation systems and determine an optimum power allocation. Finally, we provide performance comparison between the cooperation systems with the DF and AF protocols.

In the previous section equal power allocation at the source and the relay was assumed. In the following,  $P_1$  denotes the power allocated at the source node and  $P_2$  denotes the power allocated at the relay. The variable  $\tilde{P}_2$  equals  $P_2$  if the relay transmits and equals 0 otherwise.

#### 18.3.1 SER Analysis for DF Cooperative Communications

With knowledge of the channel coefficients  $h_{s,d}$  and  $h_{r,d}$ , the destination detects the transmitted symbols by jointly combining the received signal  $y_{s,d}$  from the source and  $y_{r,d}$  from the relay. The combined signal at the MRC detector can be written as

$$y = a_1 y_{s,d} + a_2 y_{r,d}, \quad (18.24)$$

in which the factors  $a_1$  and  $a_2$  are determined such that the SNR of the MRC output is maximized, and they can be specified as  $a_1 = \sqrt{P_1 h_{s,d}^* / N_0}$  and  $a_2 = \sqrt{\tilde{P}_2 h_{r,d}^* / N_0}$ . Assume that the transmitted symbol has average energy 1; then the SNR of the MRC output is

$$\gamma = \frac{P_1 |h_{s,d}|^2 + \tilde{P}_2 |h_{r,d}|^2}{N_0}. \quad (18.25)$$

If  $M$ -PSK modulation is used in the system, with the instantaneous SNR  $\gamma$  in (18.25), the conditional SER of the system with the channel coefficients  $h_{s,d}$ ,  $h_{s,r}$ , and  $h_{r,d}$  can be written as [9]

$$P_{\text{PSK}}^{h_{s,d}, h_{s,r}, h_{r,d}} = \Psi_{\text{PSK}}(\gamma) \frac{1}{\pi} \int_0^{(M-1)\pi/M} \exp\left(-\frac{b_{\text{PSK}}\gamma}{\sin^2 \theta}\right) d\theta, \quad (18.26)$$

where  $b_{\text{PSK}} = \sin^2(\pi/M)$ . If  $M$ -QAM ( $M = 2^k$  with  $k$  even) signals are used in the system, the conditional SER of the system can also be expressed as [9]

$$P_{\text{QAM}}^{h_{s,d}, h_{s,r}, h_{r,d}} = \Psi_{\text{QAM}}(\gamma), \quad (18.27)$$

where

$$\Psi_{\text{QAM}}(\gamma) \triangleq 4K Q(\sqrt{b_{\text{QAM}}\gamma}) - 4K^2 Q^2(\sqrt{b_{\text{QAM}}\gamma}) \quad (18.28)$$

in which  $K = 1 - 1/\sqrt{M}$ ,  $b_{\text{QAM}} = 3/(M-1)$ , and  $Q(u) = 1/\sqrt{2\pi} \int_u^\infty \exp(-t^2/2) dt$  is the Gaussian  $Q$  function [10]. It is easy to see that in case of  $Q$ -PSK or 4-QAM modulation, the conditional SER in (18.26) and (18.27) are the same.

Note that in phase 2, we assume that if the relay decodes the transmitted symbol  $x$  from the source correctly, then the relay forwards the decoded symbol with power  $P_2$  to the destination, that is,  $\tilde{P}_2 = P_2$ ; otherwise the relay does not send, that is,  $\tilde{P}_2 = 0$ . If an  $M$ -PSK symbol is sent from the source, then at the relay, the chance of incorrect decoding is  $\Psi_{\text{PSK}}(P_1|h_{s,r}|^2/\mathcal{N}_0)$ , and the chance of correct decoding is  $1 - \Psi_{\text{PSK}}(P_1|h_{s,r}|^2/\mathcal{N}_0)$ . Similarly, if an  $M$ -QAM symbol is sent out at the source, then the chance of incorrect decoding at the relay is  $\Psi_{\text{QAM}}(P_1|h_{s,r}|^2/\mathcal{N}_0)$ , and the chance of correct decoding is  $1 - \Psi_{\text{QAM}}(P_1|h_{s,r}|^2/\mathcal{N}_0)$ .

Let us first focus on the SER analysis in case of  $M$ -PSK modulation. Taking into account the two scenarios of  $\tilde{P}_2 = P_2$  and  $\tilde{P}_2 = 0$ , we can calculate the conditional SER in (18.26) as

$$\begin{aligned} P_{\text{PSK}}^{h_{s,d}, h_{s,r}, h_{r,d}} &= \Psi_{\text{PSK}}(\gamma)|_{\tilde{P}_2=0} \Psi_{\text{PSK}}\left(\frac{P_1|h_{s,r}|^2}{\mathcal{N}_0}\right) \\ &\quad + \Psi_{\text{PSK}}(\gamma)|_{\tilde{P}_2=P_2} \left[1 - \Psi_{\text{PSK}}\left(\frac{P_1|h_{s,r}|^2}{\mathcal{N}_0}\right)\right] \\ &= \frac{1}{\pi^2} \int_0^{(M-1)\pi/M} \exp\left(-\frac{b_{\text{PSK}}P_1|h_{s,d}|^2}{\mathcal{N}_0 \sin^2 \theta}\right) d\theta \\ &\quad \times \int_0^{(M-1)\pi/M} \exp\left(-\frac{b_{\text{PSK}}P_1|h_{s,r}|^2}{\mathcal{N}_0 \sin^2 \theta}\right) d\theta \\ &\quad + \frac{1}{\pi} \int_0^{(M-1)\pi/M} \exp\left[-\frac{b_{\text{PSK}}(P_1|h_{s,d}|^2 + P_2|h_{r,d}|^2)}{\mathcal{N}_0 \sin^2 \theta}\right] d\theta \\ &\quad \times \left[1 - \frac{1}{\pi} \int_0^{(M-1)\pi/M} \exp\left(-\frac{b_{\text{PSK}}P_1|h_{s,r}|^2}{\mathcal{N}_0 \sin^2 \theta}\right) d\theta\right]. \end{aligned} \quad (18.29)$$

Averaging the conditional SER (18.29) over the Rayleigh fading channels  $h_{s,d}$ ,  $h_{s,r}$ , and  $h_{r,d}$ , we obtain the SER of the DF cooperation system with  $M$ -PSK modulation as follows:

$$\begin{aligned} P_{\text{PSK}} &= F_1 \left( 1 + \frac{b_{\text{PSK}} P_1 \sigma_{s,d}^2}{\mathcal{N}_0 \sin^2 \theta} \right) F_1 \left( 1 + \frac{b_{\text{PSK}} P_1 \sigma_{s,r}^2}{\mathcal{N}_0 \sin^2 \theta} \right) \\ &\quad + F_1 \left[ \left( 1 + \frac{b_{\text{PSK}} P_1 \sigma_{s,d}^2}{\mathcal{N}_0 \sin^2 \theta} \right) \left( 1 + \frac{b_{\text{PSK}} P_2 \sigma_{r,d}^2}{\mathcal{N}_0 \sin^2 \theta} \right) \right] \left[ 1 - F_1 \left( 1 + \frac{b_{\text{PSK}} P_1 \sigma_{s,r}^2}{\mathcal{N}_0 \sin^2 \theta} \right) \right], \end{aligned} \quad (18.30)$$

where  $F_1(x(\theta)) = (1/\pi) \int_0^{(M-1)\pi/M} [1/x(\theta)] d\theta$ , in which  $x(\theta)$  denotes a function with variable  $\theta$ .

For DF cooperation systems with  $M$ -QAM modulation, the conditional SER in (18.27) with the channel coefficients  $h_{s,d}$ ,  $h_{s,r}$ , and  $h_{r,d}$  can be similarly determined as

$$\begin{aligned} P_{\text{QAM}}^{h_{s,d}, h_{s,r}, h_{r,d}} &= \Psi_{\text{QAM}}(\gamma)|_{\tilde{P}_2=0} \Psi_{\text{QAM}} \left( \frac{P_1 |h_{s,r}|^2}{\mathcal{N}_0} \right) \\ &\quad + \Psi_{\text{QAM}}(\gamma)|_{\tilde{P}_2=P_2} \left[ 1 - \Psi_{\text{QAM}} \left( \frac{P_1 |h_{s,r}|^2}{\mathcal{N}_0} \right) \right]. \end{aligned} \quad (18.31)$$

By substituting (18.28) into (18.31) and averaging it over the fading channels  $h_{s,d}$ ,  $h_{s,r}$ , and  $h_{r,d}$ , the SER of the DF cooperation system with  $M$ -QAM modulation can be given by

$$\begin{aligned} P_{\text{QAM}} &= F_2 \left( 1 + \frac{b_{\text{QAM}} P_1 \sigma_{s,d}^2}{2\mathcal{N}_0 \sin^2 \theta} \right) F_2 \left( 1 + \frac{b_{\text{QAM}} P_1 \sigma_{s,r}^2}{2\mathcal{N}_0 \sin^2 \theta} \right) \\ &\quad + F_2 \left[ \left( 1 + \frac{b_{\text{QAM}} P_1 \sigma_{s,d}^2}{2\mathcal{N}_0 \sin^2 \theta} \right) \left( 1 + \frac{b_{\text{QAM}} P_2 \sigma_{r,d}^2}{2\mathcal{N}_0 \sin^2 \theta} \right) \right] \\ &\quad \times \left[ 1 - F_2 \left( 1 + \frac{b_{\text{QAM}} P_1 \sigma_{s,r}^2}{2\mathcal{N}_0 \sin^2 \theta} \right) \right], \end{aligned} \quad (18.32)$$

where

$$F_2(x(\theta)) = \frac{4K}{\pi} \int_0^{\pi/2} \frac{1}{x(\theta)} d\theta - \frac{4K^2}{\pi} \int_0^{\pi/4} \frac{1}{x(\theta)} d\theta \quad (18.33)$$

in which  $x(\theta)$  denotes a function with variable  $\theta$ . In order to get the SER formulation in (18.32), we used two special properties of the Gaussian  $Q$  function as follows:

$$\begin{aligned} Q(u) &= \frac{1}{\pi} \int_0^{\pi/2} \exp \left( -\frac{u^2}{2 \sin^2 \theta} \right) d\theta \quad \text{and} \\ Q^2(u) &= \frac{1}{\pi} \int_0^{\pi/4} \exp \left( -\frac{u^2}{2 \sin^2 \theta} \right) d\theta \end{aligned}$$

for any  $u \geq 0$  [9, 11].

**18.3.1.1 SER Upper Bound and Asymptotically Tight Approximation** Even though the closed-form SER formulations in (18.30) and (18.32) can be efficiently calculated numerically, they are very complex, and it is hard to get insight into the system performance from these. In the following theorem, we provide an upper bound as well as an approximation, which are useful in demonstrating the asymptotic performance of the DF cooperation scheme. The SER approximation is asymptotically tight at high SNR.

**Theorem 18.1** [12] The SER of the DF cooperation systems with  $M$ -PSK or  $M$ -QAM modulation can be upper-bounded as

$$P_s \leq \frac{(M-1)\mathcal{N}_0^2}{M^2} \cdot \frac{MbP_1\sigma_{s,r}^2 + (M-1)bP_2\sigma_{r,d}^2 + (2M-1)\mathcal{N}_0}{(\mathcal{N}_0 + bP_1\sigma_{s,d}^2)(\mathcal{N}_0 + bP_1\sigma_{s,r}^2)(\mathcal{N}_0 + bP_2\sigma_{r,d}^2)}, \quad (18.34)$$

where  $b = b_{\text{PSK}}$  for  $M$ -PSK signals and  $b = b_{\text{QAM}}/2$  for  $M$ -QAM signals. Furthermore, if all of the channel links  $h_{s,d}$ ,  $h_{s,r}$ , and  $h_{r,d}$  are available, that is,  $\sigma_{s,d}^2 \neq 0$ ,  $\sigma_{s,r}^2 \neq 0$ , and  $\sigma_{r,d}^2 \neq 0$ , then for sufficiently high SNR, the SER of the systems with  $M$ -PSK or  $M$ -QAM modulation can be tightly approximated as

$$P_s \approx \frac{\mathcal{N}_0^2}{b^2} \cdot \frac{1}{P_1\sigma_{s,d}^2} \left( \frac{A^2}{P_1\sigma_{s,r}^2} + \frac{B}{P_2\sigma_{r,d}^2} \right), \quad (18.35)$$

where in case of  $M$ -PSK signals,  $b = b_{\text{PSK}}$  and

$$A = \frac{M-1}{2M} + \frac{\sin \frac{2\pi}{M}}{4\pi}, \quad B = \frac{3(M-1)}{8M} + \frac{\sin \frac{2\pi}{M}}{4\pi} - \frac{\sin \frac{4\pi}{M}}{32\pi}; \quad (18.36)$$

while in case of  $M$ -QAM signals,  $b = b_{\text{QAM}}/2$  and

$$A = \frac{M-1}{2M} + \frac{K^2}{\pi}, \quad B = \frac{3(M-1)}{8M} + \frac{K^2}{\pi}. \quad (18.37)$$

*Proof* First, let us show the upper bound in (18.34). In case of  $M$ -PSK modulation, the closed-form SER expression was given in (18.30). By removing the negative term in (18.30), we have

$$\begin{aligned} P_{\text{PSK}} &\leq F_1 \left( 1 + \frac{b_{\text{PSK}}P_1\sigma_{s,d}^2}{\mathcal{N}_0 \sin^2 \theta} \right) F_1 \left( 1 + \frac{b_{\text{PSK}}P_1\sigma_{s,r}^2}{\mathcal{N}_0 \sin^2 \theta} \right) \\ &\quad + F_1 \left[ \left( 1 + \frac{b_{\text{PSK}}P_1\sigma_{s,d}^2}{\mathcal{N}_0 \sin^2 \theta} \right) \left( 1 + \frac{b_{\text{PSK}}P_2\sigma_{r,d}^2}{\mathcal{N}_0 \sin^2 \theta} \right) \right]. \end{aligned} \quad (18.38)$$

We observe that in the right-hand side of the above inequality, all integrands have their maximum value when  $\sin^2 \theta = 1$ . Therefore, by substituting  $\sin^2 \theta = 1$  into (18.38),

we have

$$\begin{aligned}
P_{\text{PSK}} &\leq \frac{(M-1)^2}{M^2} \cdot \frac{\mathcal{N}_0^2}{(\mathcal{N}_0 + b_{\text{PSK}} P_1 \sigma_{s,d}^2)(\mathcal{N}_0 + b_{\text{PSK}} P_1 \sigma_{s,r}^2)} \\
&\quad + \frac{M-1}{M} \cdot \frac{\mathcal{N}_0^2}{(\mathcal{N}_0 + b_{\text{PSK}} P_1 \sigma_{s,d}^2)(\mathcal{N}_0 + b_{\text{PSK}} P_2 \sigma_{r,d}^2)} \\
&= \frac{(M-1)\mathcal{N}_0^2}{M^2} \cdot \frac{Mb_{\text{PSK}} P_1 \sigma_{s,r}^2 + (M-1)b_{\text{PSK}} P_2 \sigma_{r,d}^2 + (2M-1)\mathcal{N}_0}{(\mathcal{N}_0 + b_{\text{PSK}} P_1 \sigma_{s,d}^2)(\mathcal{N}_0 + b_{\text{PSK}} P_1 \sigma_{s,r}^2)(\mathcal{N}_0 + b_{\text{PSK}} P_2 \sigma_{r,d}^2)},
\end{aligned}$$

which validates the upper bound in (18.34) for  $M$ -PSK modulation. Similarly, in case of  $M$ -QAM modulation, the SER in (18.32) can be upper bounded as

$$\begin{aligned}
P_{\text{QAM}} &\leq F_2 \left( 1 + \frac{b_{\text{QAM}} P_1 \sigma_{s,d}^2}{2\mathcal{N}_0 \sin^2 \theta} \right) F_2 \left( 1 + \frac{b_{\text{QAM}} P_1 \sigma_{s,r}^2}{2\mathcal{N}_0 \sin^2 \theta} \right) \\
&\quad + F_2 \left[ \left( 1 + \frac{b_{\text{QAM}} P_1 \sigma_{s,d}^2}{2\mathcal{N}_0 \sin^2 \theta} \right) \left( 1 + \frac{b_{\text{QAM}} P_2 \sigma_{r,d}^2}{2\mathcal{N}_0 \sin^2 \theta} \right) \right]. \tag{18.39}
\end{aligned}$$

Note that, the function  $F_2(x(\theta))$  defined in (18.33) can be rewritten as

$$F_2(x(\theta)) = \frac{4K}{\pi \sqrt{M}} \int_0^{\pi/2} \frac{1}{x(\theta)} d\theta + \frac{4K^2}{\pi} \int_{\pi/4}^{\pi/2} \frac{1}{x(\theta)} d\theta, \tag{18.40}$$

which does not contain a negative term. Moreover, the integrands in (18.39) have their maximum value when  $\sin^2 \theta = 1$ . Thus, by substituting (18.40) and  $\sin^2 \theta = 1$  into (18.39), we have

$$\begin{aligned}
P_{\text{QAM}} &\leq \left( \frac{2K}{\sqrt{M}} + K^2 \right)^2 \frac{\mathcal{N}_0^2}{(\mathcal{N}_0 + \frac{b_{\text{QAM}}}{2} P_1 \sigma_{s,d}^2)(\mathcal{N}_0 + \frac{b_{\text{QAM}}}{2} P_1 \sigma_{s,r}^2)} \\
&\quad + \left( \frac{2K}{\sqrt{M}} + K^2 \right) \frac{\mathcal{N}_0^2}{(\mathcal{N}_0 + \frac{b_{\text{QAM}}}{2} P_1 \sigma_{s,d}^2)(\mathcal{N}_0 + \frac{b_{\text{QAM}}}{2} P_2 \sigma_{r,d}^2)} \\
&= \frac{(M-1)\mathcal{N}_0^2}{M^2} \cdot \frac{M \frac{b_{\text{QAM}}}{2} P_1 \sigma_{s,r}^2 + (M-1) \frac{b_{\text{QAM}}}{2} P_2 \sigma_{r,d}^2 + (2M-1)\mathcal{N}_0}{(\mathcal{N}_0 + \frac{b_{\text{QAM}}}{2} P_1 \sigma_{s,d}^2)(\mathcal{N}_0 + \frac{b_{\text{QAM}}}{2} P_1 \sigma_{s,r}^2)(\mathcal{N}_0 + \frac{b_{\text{QAM}}}{2} P_2 \sigma_{r,d}^2)},
\end{aligned}$$

in which  $K = 1 - 1/\sqrt{M}$ . Therefore, the upper bound in (18.34) also holds for  $M$ -QAM modulation.

In the following, we show the asymptotically tight approximation (18.35) with the assumption that all of the channel links  $h_{s,d}$ ,  $h_{s,r}$ , and  $h_{r,d}$  are available, that is,  $\sigma_{s,d}^2 \neq 0$ ,  $\sigma_{s,r}^2 \neq 0$ , and  $\sigma_{r,d}^2 \neq 0$ . First, let us consider the  $M$ -PSK modulation. In the SER

formulation (18.30), we observe that for sufficiently large power  $P_1$  and  $P_2$ ,

$$\begin{aligned} 1 + \frac{b_{\text{PSK}} P_1 \sigma_{s,d}^2}{\mathcal{N}_0 \sin^2 \theta} &\approx \frac{b_{\text{PSK}} P_1 \sigma_{s,d}^2}{\mathcal{N}_0 \sin^2 \theta} \\ 1 + \frac{b_{\text{PSK}} P_1 \sigma_{s,r}^2}{\mathcal{N}_0 \sin^2 \theta} &\approx \frac{b_{\text{PSK}} P_1 \sigma_{s,r}^2}{\mathcal{N}_0 \sin^2 \theta} \\ 1 + \frac{b_{\text{PSK}} P_2 \sigma_{r,d}^2}{\mathcal{N}_0 \sin^2 \theta} &\approx \frac{b_{\text{PSK}} P_2 \sigma_{r,d}^2}{\mathcal{N}_0 \sin^2 \theta} \end{aligned}$$

that is, the 1s are negligible with sufficiently large power. Thus, for sufficiently high SNR, the SER in (18.30) can be tightly approximated as

$$\begin{aligned} P_{\text{PSK}} &\approx F_1 \left( \frac{b_{\text{PSK}} P_1 \sigma_{s,d}^2}{\mathcal{N}_0 \sin^2 \theta} \right) F_1 \left( \frac{b_{\text{PSK}} P_1 \sigma_{s,r}^2}{\mathcal{N}_0 \sin^2 \theta} \right) \\ &\quad + F_1 \left( \frac{b_{\text{PSK}}^2 P_1 P_2 \sigma_{s,d}^2 \sigma_{s,r}^2}{\mathcal{N}_0^2 \sin^4 \theta} \right) \left[ 1 - F_1 \left( \frac{b_{\text{PSK}} P_1 \sigma_{s,r}^2}{\mathcal{N}_0 \sin^2 \theta} \right) \right] \\ &\approx F_1 \left( \frac{b_{\text{PSK}} P_1 \sigma_{s,d}^2}{\mathcal{N}_0 \sin^2 \theta} \right) F_1 \left( \frac{b_{\text{PSK}} P_1 \sigma_{s,r}^2}{\mathcal{N}_0 \sin^2 \theta} \right) + F_1 \left( \frac{b_{\text{PSK}}^2 P_1 P_2 \sigma_{s,d}^2 \sigma_{s,r}^2}{\mathcal{N}_0^2 \sin^4 \theta} \right), \\ &= \frac{A^2 \mathcal{N}_0^2}{b_{\text{PSK}}^2 P_1^2 \sigma_{s,d}^2 \sigma_{s,r}^2} + \frac{B \mathcal{N}_0^2}{b_{\text{PSK}}^2 P_1 P_2 \sigma_{s,d}^2 \sigma_{r,d}^2}, \end{aligned} \tag{18.41}$$

in which

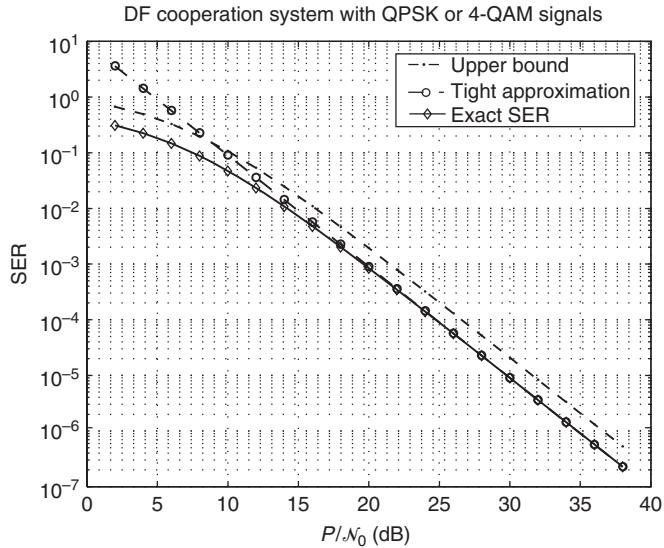
$$\begin{aligned} A &= \frac{1}{\pi} \int_0^{(M-1)\pi/M} \sin^2 \theta d\theta = \frac{M-1}{2M} + \frac{\sin \frac{2\pi}{M}}{4\pi} \\ B &= \frac{1}{\pi} \int_0^{(M-1)\pi/M} \sin^4 \theta d\theta = \frac{3(M-1)}{8M} + \frac{\sin \frac{2\pi}{M}}{4\pi} - \frac{\sin \frac{\pi}{M}}{32\pi}. \end{aligned}$$

Note that the second approximation is due to the fact that

$$1 - F_1 \left( \frac{b_{\text{PSK}} P_1 \sigma_{s,r}^2}{\mathcal{N}_0 \sin^2 \theta} \right) = 1 - \frac{\mathcal{N}_0}{\pi b_{\text{PSK}} P_1 \sigma_{s,r}^2} \int_0^{(M-1)\pi/M} \sin^2 \theta d\theta \approx 1$$

for sufficiently large  $P_1$ . Therefore, the asymptotically tight approximation in (18.35) holds for the  $M$ -PSK modulation. In case of  $M$ -QAM signals, similarly the SER formulation in (18.32) can be tightly approximated at high SNR as follows:

$$\begin{aligned} P_{\text{QAM}} &\approx F_2 \left( \frac{b_{\text{QAM}} P_1 \sigma_{s,d}^2}{2\mathcal{N}_0 \sin^2 \theta} \right) F_2 \left( \frac{b_{\text{QAM}} P_1 \sigma_{s,r}^2}{2\mathcal{N}_0 \sin^2 \theta} \right) + F_2 \left( \frac{b_{\text{QAM}}^2 P_1 P_2 \sigma_{s,d}^2 \sigma_{r,d}^2}{4\mathcal{N}_0^2 \sin^4 \theta} \right) \\ &= \frac{4A^2 \mathcal{N}_0^2}{b_{\text{QAM}}^2 P_1^2 \sigma_{s,d}^2 \sigma_{s,r}^2} + \frac{4B \mathcal{N}_0^2}{b_{\text{QAM}}^2 P_1 P_2 \sigma_{s,d}^2 \sigma_{r,d}^2}, \end{aligned} \tag{18.42}$$



**Figure 18.4** Comparison of the exact SER formulation, the upper bound and the asymptotically tight approximation for the DF cooperation system with  $Q$ -PSK or 4-QAM signals. We assumed that  $\sigma_{s,d}^2 = \sigma_{s,r}^2 = \sigma_{r,d}^2 = 1$ ,  $N_0 = 1$ , and  $P_1 = P_2 = P/2$ .

where

$$A = \frac{4K}{\pi\sqrt{M}} \int_0^{\pi/2} \sin^2 \theta d\theta + \frac{4K^2}{\pi} \int_{\pi/4}^{\pi/2} \sin^2 \theta d\theta = \frac{M-1}{2M} + \frac{K^2}{\pi},$$

$$B = \frac{4K}{\pi\sqrt{M}} \int_0^{\pi/2} \sin^4 \theta d\theta + \frac{4K^2}{\pi} \int_{\pi/4}^{\pi/2} \sin^4 \theta d\theta = \frac{3(M-1)}{8M} + \frac{K^2}{\pi}.$$

Thus, the asymptotically tight approximation in (18.35) also holds for the  $M$ -QAM signals.

In Figure 18.4, we compare the asymptotically tight approximation (18.35) and the SER upper bound (18.34) with the exact SER formulations (18.30) and (18.32) in case of QPSK (or 4-QAM) modulation. In this case, the parameters  $b$ ,  $A$ , and  $B$  in the upper bound (18.34) and the approximation (18.35) are specified as  $b = 1$ ,  $A = \frac{3}{8} + 1/4\pi$  and  $B = \frac{9}{32} + 1/4\pi$ . We can see that the upper bound (18.34) (dashed line with  $\cdot$ ) is asymptotically parallel with the exact SER curve (solid line with  $\diamond$ ), which means that they have the same diversity order. The approximation (18.35) (dashed line with  $\circ$ ) is loose at low SNR, but it is tight at reasonable high SNR. It merges with the exact SER curve at an SER of  $10^{-3}$ . Both the SER upper bound and the approximation show the asymptotic performance of the DF cooperation systems. Specifically, from the asymptotically tight approximation (18.35), we observe that the link between source and destination contributes diversity order 1 in the system performance. The term  $(A^2/P_1\sigma_{s,r}^2) + (B/P_2\sigma_{r,d}^2)$  also contributes diversity order 1 in the performance, but it depends on the balance of the two channel links from source to relay and from relay to destination. Therefore, the DF cooperation systems show an overall performance of diversity order 2.

**18.3.1.2 DF Optimum Power Allocation** Note that the SER approximation (18.35) is asymptotically tight at high SNR. In the following, we determine an asymptotic optimum power allocation for the DF cooperation protocol based on the asymptotically tight SER approximation.

Specifically, we try to determine an optimum transmitted power  $P_1$  that should be used at the source and  $P_2$  at the relay for a fixed total transmission power  $P_1 + P_2 = P$ . According to the asymptotically tight SER approximation (18.35), it is sufficient to minimize

$$G(P_1, P_2) = \frac{1}{P_1 \sigma_{s,d}^2} \left( \frac{A^2}{P_1 \sigma_{s,r}^2} + \frac{B}{P_2 \sigma_{r,d}^2} \right).$$

By taking derivative in terms of  $P_1$ , we have

$$\frac{\partial G(P_1, P_2)}{\partial P_1} = \frac{1}{P_1 \sigma_{s,d}^2} \left( -\frac{A^2}{P_1^2 \sigma_{s,r}^2} + \frac{B}{P_2^2 \sigma_{r,d}^2} \right) - \frac{1}{P_1^2 \sigma_{s,d}^2} \left( \frac{A^2}{P_1 \sigma_{s,r}^2} + \frac{B}{P_2 \sigma_{r,d}^2} \right).$$

By setting the above derivation as 0, we come up with an equation as follows:

$$B \sigma_{s,r}^2 (P_1^2 - P_1 P_2) - 2A^2 \sigma_{r,d}^2 P_2^2 = 0.$$

With the power constraint, we can solve the above equation and arrive at the following result.

**Theorem 18.2** [12] In the DF cooperation systems with  $M$ -PSK or  $M$ -QAM modulation, if all of the channel links  $h_{s,d}$ ,  $h_{s,r}$ , and  $h_{r,d}$  are available, that is,  $\sigma_{s,d}^2 \neq 0$ ,  $\sigma_{s,r}^2 \neq 0$ , and  $\sigma_{r,d}^2 \neq 0$ , then for sufficiently high SNR, the optimum power allocation is

$$P_1 = \frac{\sigma_{s,r} + \sqrt{\sigma_{s,r}^2 + 8(A^2/B)\sigma_{r,d}^2}}{3\sigma_{s,r} + \sqrt{\sigma_{s,r}^2 + 8(A^2/B)\sigma_{r,d}^2}} P, \quad (18.43)$$

$$P_2 = \frac{2\sigma_{s,r}}{3\sigma_{s,r} + \sqrt{\sigma_{s,r}^2 + (8A^2/B)\sigma_{r,d}^2}} P, \quad (18.44)$$

where  $A$  and  $B$  are specified in (18.36) and (18.37) for  $M$ -PSK and  $M$ -QAM signals, respectively.

The result in Theorem 18.2 is somewhat surprising since the asymptotic optimum power allocation does not depend on the channel link between source and destination, it depends only on the channel link between source and relay and the channel link between relay and destination. Moreover, we can see that the optimum ratio of the transmitted power  $P_1$  at the source over the total power  $P$  is less than 1 and larger than  $\frac{1}{2}$ , while the optimum ratio of the power  $P_2$  used at the relay over the total power  $P$  is larger than 0 and less than  $\frac{1}{2}$ , that is,

$$\frac{1}{2} < \frac{P_1}{P} < 1 \quad \text{and} \quad 0 < \frac{P_2}{P} < \frac{1}{2}.$$

It means that we should always put more power at the source and less power at the relay. If the link quality between source and relay is much less than that between relay and destination, that is,  $\sigma_{s,r}^2 \ll \sigma_{r,d}^2$ , then from (18.43) and (18.44),  $P_1$  goes to  $P$  and  $P_2$  goes to 0. It implies that we should use almost all of the power  $P$  at the source, and use little power at the relay. On the other hand, if the link quality between source and relay is much larger than that between relay and destination, that is,  $\sigma_{s,r}^2 > \sigma_{r,d}^2$ , then both  $P_1$  and  $P_2$  go to  $P/2$ . It means that we should put equal power at the source and the relay in this case.

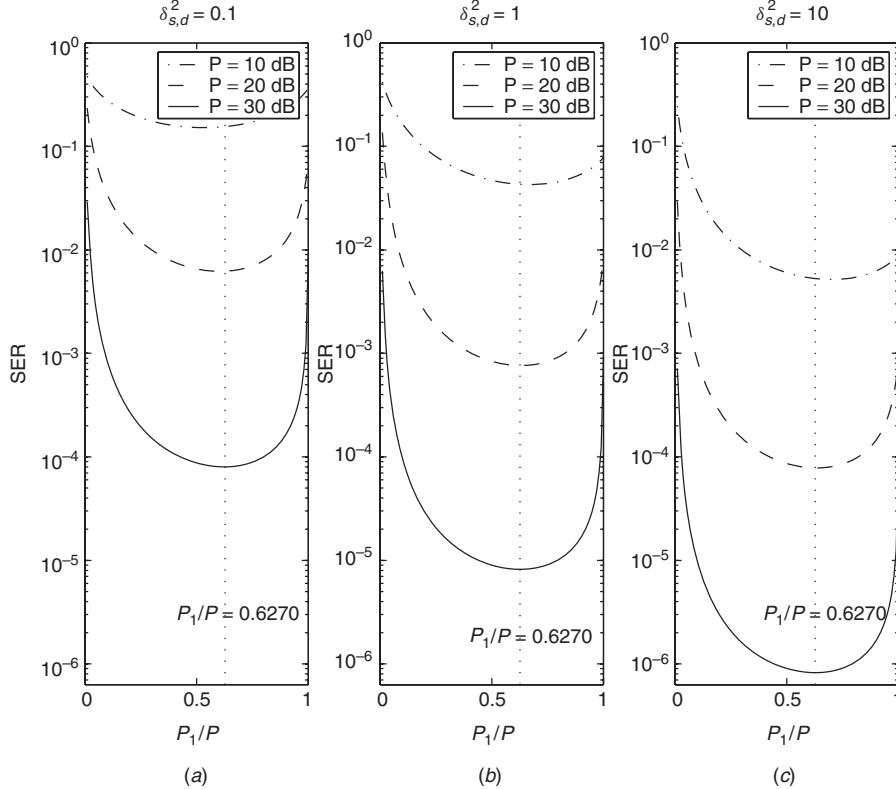
We interpret the result in Theorem 18.2 as follows. Since we assume that all of the channel links  $h_{s,d}$ ,  $h_{s,r}$ , and  $h_{r,d}$  are available in the system, the cooperation strategy is expected to achieve a performance diversity of order 2. The system is guaranteed to have a performance diversity of order 1 due to the channel link between source and destination. However, in order to achieve a diversity of order 2, the channel link between source and relay and the channel link between relay and destination should be appropriately balanced. If the link quality between source and relay is bad, then it is difficult for the relay to correctly decode the transmitted symbol. Thus, the forwarding role of the relay is less important and it makes sense to put more power at the source. On the other hand, if the link quality between source and relay is very good, the relay can always decode the transmitted symbol correctly, so the decoded symbol at the relay is almost the same as that at the source. We may consider the relay as a copy of the source and put almost equal power on them. We want to emphasize that this interpretation is good only for sufficiently high SNR scenario and under the assumption that all of the channel links  $h_{s,d}$ ,  $h_{s,r}$ , and  $h_{r,d}$  are available. Actually, this interpretation is not accurate in general. For example, in case the link quality between source and relay is the same as that between relay and destination, that is,  $\sigma_{s,r}^2 = \sigma_{r,d}^2$ , the asymptotic optimum power allocation is given by

$$P_1 = \frac{1 + \sqrt{1 + 8A^2/B}}{3 + \sqrt{1 + 8A^2/B}} P, \quad (18.45)$$

$$P_2 = \frac{2}{3 + \sqrt{1 + 8A^2/B}} P, \quad (18.46)$$

where  $A$  and  $B$  depend on specific modulation signals. For example, if BPSK modulation is used, then  $P_1 = 0.5931P$  and  $P_2 = 0.4069P$ ; while if QPSK modulation is used, then  $P_1 = 0.6270P$  and  $P_2 = 0.3730P$ . In case of 16-QAM,  $P_1 = 0.6495P$  and  $P_2 = 0.3505P$ . We can see that the larger the constellation size, the more power should be put at the source.

It is worth pointing out that even though the asymptotic optimum power allocation in (18.43) and (18.44) are determined for high SNR, they also provide a good solution to a realistic moderate SNR scenario as in Figure 18.5, in which we plotted exact SER as a function of the ratio  $P_1/P$  for a DF cooperation system with QPSK modulation. We considered the DF cooperation system with  $\sigma_{s,r}^2 = \sigma_{r,d}^2 = 1$  and three different qualities of the channel link between source and destination: (a)  $\sigma_{s,d}^2 = 0.1$ ; (b)  $\sigma_{s,d}^2 = 1$ ; and (c)  $\sigma_{s,d}^2 = 10$ . The asymptotic optimum power allocation in this case is  $P_1/P = 0.6270$  and  $P_2/P = 0.3730$ . From the figures, we can see that the ratio  $P_1/P = 0.6270$  almost provides the best performance for different total transmit power  $P = 10, 20, 30$  dB.



**Figure 18.5** SER of the DF cooperation systems with  $\sigma_{s,r}^2 = 1$  and  $\sigma_{r,d}^2 = 1$ : (a)  $\sigma_{s,d}^2 = 0.1$ , (b)  $\sigma_{s,d}^2 = 1$ , and (c)  $\sigma_{s,d}^2 = 10$ . The asymptotic optimum power allocation is  $P_1/P = 0.6270$  and  $P_2/P = 0.3730$ .

### 18.3.2 SER Analysis for AF Cooperative Communications

In this section, we investigate the SER performance for the AF cooperative communication systems. First, we derive a simple closed-form MGF expression for the harmonic mean of two independent exponential random variables. Second, based on the simple MGF expression, closed-form SER formulations are given for the AF cooperation systems with  $M$ -PSK and  $M$ -QAM modulations. Third, we provide an SER approximation, which is tight at high SNR, to show the asymptotic performance of the systems. Finally, based on the tight approximation, we are able to determine an optimum power allocation for the AF cooperation systems.

**18.3.2.1 SER Analysis by MGF Approach** In the AF cooperation systems, the relay amplifies not only the received signal but also the noise. The equivalent noise  $\eta'_{r,d}$  at the destination in phase 2 is a zero-mean complex Gaussian random variable with variance  $[(P_2|h_{r,d}|^2/P_1|h_{s,r}|^2 + \mathcal{N}_0) + 1]\mathcal{N}_0$ . Therefore, with knowledge of the channel coefficients  $h_{s,d}$ ,  $h_{s,r}$ , and  $h_{r,d}$ , the output of the MRC detector at the destination can be written as

$$y = a_1 y_{s,d} + a_2 y_{r,d}, \quad (18.47)$$

where  $a_1$  and  $a_2$  are specified as

$$a_1 = \frac{\sqrt{P_1} h_{s,d}^*}{\mathcal{N}_0} \quad \text{and} \quad a_2 = \frac{\sqrt{\frac{P_1 P_2}{P_1 |h_{s,r}|^2 + \mathcal{N}_0}} h_{s,r}^* h_{r,d}^*}{\left( \frac{P_2 |h_{r,d}|^2}{P_1 |h_{s,r}|^2 + \mathcal{N}_0} + 1 \right) \mathcal{N}_0}. \quad (18.48)$$

By assuming that the transmitted symbol  $x$  has average energy 1, we know that the instantaneous SNR of the MRC output is

$$\gamma = \gamma_1 + \gamma_2 \quad (18.49)$$

where  $\gamma_1 = P_1 |h_{s,d}|^2 / \mathcal{N}_0$ , and

$$\gamma_2 = \frac{1}{\mathcal{N}_0} \frac{P_1 P_2 |h_{s,r}|^2 |h_{r,d}|^2}{P_1 |h_{s,r}|^2 + P_2 |h_{r,d}|^2 + \mathcal{N}_0}. \quad (18.50)$$

It has been shown in [13] that the instantaneous SNR  $\gamma_2$  in (18.50) can be tightly upper bounded as

$$\tilde{\gamma}_2 = \frac{1}{\mathcal{N}_0} \frac{P_1 P_2 |h_{s,r}|^2 |h_{r,d}|^2}{P_1 |h_{s,r}|^2 + P_2 |h_{r,d}|^2}, \quad (18.51)$$

which is the harmonic mean of two exponential random variables  $P_1 |h_{s,r}|^2 / \mathcal{N}_0$  and  $P_2 |h_{r,d}|^2 / \mathcal{N}_0$ . According to (18.26) and (18.27), the conditional SER of the AF cooperation systems with  $M$ -PSK and  $M$ -QAM modulations can be given as follows:

$$P_{\text{PSK}}^{h_{s,d}, h_{s,r}, h_{r,d}} \approx \frac{1}{\pi} \int_0^{(M-1)\pi/M} \exp \left[ -\frac{b_{\text{PSK}}(\gamma_1 + \tilde{\gamma}_2)}{\sin^2 \theta} \right] d\theta, \quad (18.52)$$

$$P_{\text{QAM}}^{h_{s,d}, h_{s,r}, h_{r,d}} \approx 4K Q \left[ \sqrt{b_{\text{QAM}}(\gamma_1 + \tilde{\gamma}_2)} \right] - 4K^2 Q^2 \left[ \sqrt{b_{\text{QAM}}(\gamma_1 + \tilde{\gamma}_2)} \right], \quad (18.53)$$

where  $b_{\text{PSK}} = \sin^2(\pi/M)$ ,  $b_{\text{QAM}} = 3/(M-1)$ , and  $K = 1 - 1/\sqrt{M}$ . Note that we used the SNR approximation  $\gamma \approx \gamma_1 + \tilde{\gamma}_2$  in the above derivation.

Let us denote the MGF of a random variable  $Z$  as [9]

$$\mathcal{M}_Z(s) = \int_{-\infty}^{\infty} \exp(-sz) p_Z(z) dz \quad (18.54)$$

for any real number  $s$ . By averaging over the Rayleigh fading channels  $h_{s,d}$ ,  $h_{s,r}$ , and  $h_{r,d}$  in (18.52) and (18.53), we obtain the SER of the AF cooperation systems in terms of MGF  $\mathcal{M}_{\gamma_1}(s)$  and  $\mathcal{M}_{\tilde{\gamma}_2}(s)$  as follows:

$$P_{\text{PSK}} \approx \frac{1}{\pi} \int_0^{(M-1)\pi/M} \mathcal{M}_{\gamma_1} \left( \frac{b_{\text{PSK}}}{\sin^2 \theta} \right) \mathcal{M}_{\tilde{\gamma}_2} \left( \frac{b_{\text{PSK}}}{\sin^2 \theta} \right) d\theta, \quad (18.55)$$

$$P_{\text{QAM}} \approx \left( \frac{4K}{\pi} \int_0^{\pi/2} - \frac{4K^2}{\pi} \int_0^{\pi/4} \right) \mathcal{M}_{\gamma_1} \left( \frac{b_{\text{QAM}}}{2 \sin^2 \theta} \right) \mathcal{M}_{\tilde{\gamma}_2} \left( \frac{b_{\text{QAM}}}{2 \sin^2 \theta} \right) d\theta \quad (18.56)$$

in which, for simplicity, we use the following notation:

$$\left( \frac{4K}{\pi} \int_0^{\pi/2} - \frac{4K^2}{\pi} \int_0^{\pi/4} \right) x(\theta) d\theta \triangleq \frac{4K}{\pi} \int_0^{\pi/2} x(\theta) d\theta - \frac{4K^2}{\pi} \int_0^{\pi/4} x(\theta) d\theta,$$

where  $x(\theta)$  denotes a function with variable  $\theta$ .

From (18.55) and (18.56), we can see that the remaining problem is to obtain the MGF  $\mathcal{M}_{\gamma_1}(s)$  and  $\mathcal{M}_{\tilde{\gamma}_2}(s)$ . Since  $\gamma_1 = P_1|h_{s,d}|^2/\mathcal{N}_0$  has an exponential distribution with parameter  $\mathcal{N}_0/(P_1\sigma_{s,d}^2)$ , the MGF of  $\gamma_1$  can be simply given by [9]

$$\mathcal{M}_{\gamma_1}(s) = \frac{1}{1 + \frac{sP_1\sigma_{s,d}^2}{\mathcal{N}_0}}. \quad (18.57)$$

However, it is not easy to get the MGF of  $\tilde{\gamma}_2$ , which is the harmonic mean of two exponential random variables  $P_1|h_{s,r}|^2/\mathcal{N}_0$  and  $P_2|h_{r,d}|^2/\mathcal{N}_0$ . This has been investigated in [13] by applying Laplace transform, and a solution was presented in terms of hypergeometric function as follows:

$$\begin{aligned} \mathcal{M}_{\tilde{\gamma}_2}(s) &= \frac{16\beta_1\beta_2}{3(\beta_1 + \beta_2 + 2\sqrt{\beta_1\beta_2} + s)^2} \\ &\times \left[ \frac{4(\beta_1 + \beta_2)}{\beta_1 + \beta_2 + 2\sqrt{\beta_1\beta_2} + s} {}_2F_1\left(3, \frac{3}{2}; \frac{5}{2}; \frac{\beta_1 + \beta_2 - 2\sqrt{\beta_1\beta_2} + s}{\beta_1 + \beta_2 + 2\sqrt{\beta_1\beta_2} + s}\right) \right. \\ &\quad \left. + {}_2F_1\left(2, \frac{1}{2}; \frac{5}{2}; \frac{\beta_1 + \beta_2 - 2\sqrt{\beta_1\beta_2} + s}{\beta_1 + \beta_2 + 2\sqrt{\beta_1\beta_2} + s}\right) \right] \end{aligned} \quad (18.58)$$

in which  $\beta_1 = \mathcal{N}_0/(P_1\sigma_{s,r}^2)$ ,  $\beta_2 = \mathcal{N}_0/(P_2\sigma_{r,d}^2)$ , and  ${}_2F_1(\cdot, \cdot; \cdot; \cdot)$  is the hypergeometric function.<sup>1</sup> Because the hypergeometric function  ${}_2F_1(\cdot, \cdot; \cdot; \cdot)$  is defined as an integral, it is hard to use in an SER analysis aimed at revealing the asymptotic performance and optimizing the power allocation. Using an alternative approach, we found a simple closed-form solution for the MGF of  $\tilde{\gamma}_2$  as shown in the next section.

**18.3.2.2 Simple MGF Expression for Harmonic Mean** In the following, we obtain at first a general result on the probability density function (pdf) for the harmonic mean of two independent random variables. Then, we are able to determine a simple closed-form MGF expression for the harmonic mean of two independent exponential random variables.

**Theorem 18.3** [12] Suppose that  $X_1$  and  $X_2$  are two independent random variables with pdf  $p_{X_1}(x)$  and  $p_{X_2}(x)$  defined for all  $x \geq 0$ , and  $p_{X_1}(x) = 0$  and  $p_{X_2}(x) = 0$

<sup>1</sup>A hypergeometric function with variables  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $z$  is defined as [14]

$${}_2F_1(\alpha, \beta; \gamma; z) = \frac{\Gamma(\gamma)}{\Gamma(\beta)\Gamma(\gamma - \beta)} \int_0^1 t^{\beta-1} (1-t)^{\gamma-\beta-1} (1-tz)^{-\alpha} dt$$

where  $\Gamma(\cdot)$  is the gamma function.

for  $x < 0$ . Then the pdf of  $Z = X_1 X_2/x_1 + X_2$ , the harmonic mean of  $X_1$  and  $X_2$ , is

$$p_Z(z) = z \int_0^1 \frac{1}{t^2(1-t)^2} p_{X_1}\left(\frac{z}{1-t}\right) p_{X_2}\left(\frac{z}{t}\right) dt U(z) \quad (18.59)$$

in which  $U(z) = 1$  for  $z \geq 0$  and  $U(z) = 0$  for  $z < 0$ .

Note that we do not specify the distributions of the two independent random variables in Theorem 18.3. The proof is omitted due to the space limitation. Suppose that  $X_1$  and  $X_2$  are two independent exponential random variables with parameters  $\beta_1$  and  $\beta_2$ , respectively, that is,  $p_{X_1}(x) = \beta_1 e^{-\beta_1 x} U(x)$  and  $p_{X_2}(x) = \beta_2 e^{-\beta_2 x} U(x)$ . Then, according to Theorem 18.3, the pdf of the harmonic mean  $Z = X_1 X_2/x_1 + X_2$  can be simply given as

$$p_Z(z) = z \int_0^1 \frac{\beta_1 \beta_2}{t^2(1-t)^2} \exp\left[-\left(\frac{\beta_1}{1-t} + \frac{\beta_2}{t}\right)z\right] dt U(z). \quad (18.60)$$

The pdf of the harmonic mean  $Z$  has been presented in [13] in term of the zero-order and first-order modified Bessel functions [14]. The pdf expression in (18.60) is critical for us to obtain a simple closed-form MGF result for the harmonic mean  $Z$ .

Let us start calculating the MGF of the harmonic mean of two independent exponential random variables by substituting the pdf of  $Z$  (18.60) into the definition (18.54) as follows:

$$\begin{aligned} \mathcal{M}_Z(s) &= \int_0^\infty e^{-sz} z \int_0^1 \frac{\beta_1 \beta_2}{t^2(1-t)^2} \exp\left[-\left(\frac{\beta_1}{1-t} + \frac{\beta_2}{t}\right)z\right] dt dz \\ &= \int_0^1 \frac{\beta_1 \beta_2}{t^2(1-t)^2} \left\{ \int_0^\infty z \exp\left[-\left(\frac{\beta_1}{1-t} + \frac{\beta_2}{t} + s\right)z\right] dz \right\} dt \end{aligned} \quad (18.61)$$

in which we switch the integration order. Since

$$\int_0^\infty z \exp\left[-\left(\frac{\beta_1}{1-t} + \frac{\beta_2}{t} + s\right)z\right] dz = \left(\frac{\beta_1}{1-t} + \frac{\beta_2}{t} + s\right)^{-2},$$

the MGF in (18.61) can be determined as

$$\mathcal{M}_Z(s) = \int_0^1 \frac{\beta_1 \beta_2}{[\beta_2 + (\beta_1 - \beta_2 + s)t - st^2]^2} dt, \quad (18.62)$$

which is an integration of a quadratic trinomial and has a closed-form solution [14]. For notation simplicity, denote  $\alpha = (\beta_1 - \beta_2 + s)/2$ . According to the results on the integration over quadratic trinomial ([14], Eqs. 2.103.3 and 2.103.4), for any  $s > 0$ , we have

$$\begin{aligned} \int_0^1 \frac{1}{(\beta_2 + 2\alpha t - st^2)^2} dt &= \frac{st - \alpha}{2(\beta_2 s + \alpha^2)(\beta_2 + 2\alpha t - st^2)} \Big|_0^1 \\ &+ \frac{s}{4(\beta_2 s + \alpha^2)^{3/2}} \ln \left| \frac{-st + \alpha - \sqrt{\beta_2 s + \alpha^2}}{-st + \alpha + \sqrt{\beta_2 s + \alpha^2}} \right|_0^1 \end{aligned}$$

$$\begin{aligned}
&= \frac{\beta_2 s + \alpha(\beta_1 - \beta_2)}{2\beta_1\beta_2(\beta_2 s + \alpha^2)} \\
&\quad + \frac{s}{4(\beta_2 s + \alpha^2)^{3/2}} \ln \frac{(\beta_2 + \alpha + \sqrt{\beta_2 s + \alpha^2})^2}{\beta_1\beta_2}. \quad (18.63)
\end{aligned}$$

By substituting  $\alpha = (\beta_1 - \beta_2 + s)/2$  into (18.63) and denoting  $\sigma = 2\sqrt{\beta_2 s + \alpha^2}$ , we obtain a simple closed-form MGF for the harmonic mean  $Z$  as follows:

$$\mathcal{M}_Z(s) = \frac{(\beta_1 - \beta_2)^2 + (\beta_1 + \beta_2)s}{\sigma^2} + \frac{2\beta_1\beta_2 s}{\sigma^3} \ln \frac{(\beta_1 + \beta_2 + s + \sigma)^2}{4\beta_1\beta_2}, \quad s > 0, \quad (18.64)$$

where  $\sigma = \sqrt{(\beta_1 - \beta_2)^2 + 2(\beta_1 + \beta_2)s + s^2}$ . We can see that if  $\beta_1$  and  $\beta_2$  go to zero, then  $\sigma$  can be approximated as  $s$ . In this case, the MGF in (18.64) can be simplified as

$$\mathcal{M}_Z(s) \approx \frac{\beta_1 + \beta_2}{s} + \frac{2\beta_1\beta_2}{s^2} \ln \frac{s^2}{\beta_1\beta_2}. \quad (18.65)$$

Note that in (18.65), the second term goes to zero faster than the first term. As a result, the MGF in (18.65) can be further simplified as

$$\mathcal{M}_Z(s) \approx \frac{\beta_1 + \beta_2}{s}. \quad (18.66)$$

We summarize the above discussion in the following theorem.

**Theorem 18.4** [12] Let  $X_1$  and  $X_2$  be two independent exponential random variables with parameters  $\beta_1$  and  $\beta_2$ , respectively. Then, the MGF of  $Z = X_1 X_2 / x_1 + X_2$  is

$$\mathcal{M}_Z(s) = \frac{(\beta_1 - \beta_2)^2 + (\beta_1 + \beta_2)s}{\sigma^2} + \frac{2\beta_1\beta_2 s}{\sigma^3} \ln \frac{(\beta_1 + \beta_2 + s + \sigma)^2}{4\beta_1\beta_2} \quad (18.67)$$

for any  $s > 0$ , in which

$$\sigma = \sqrt{(\beta_1 - \beta_2)^2 + 2(\beta_1 + \beta_2)s + s^2}. \quad (18.68)$$

Furthermore, if  $\beta_1$  and  $\beta_2$  go to zero, then the MGF of  $Z$  can be approximated as

$$\mathcal{M}_Z(s) \approx \frac{\beta_1 + \beta_2}{s}. \quad (18.69)$$

We can see that the closed-form solution in (18.67) does not involve any integration. If  $X_1$  and  $X_2$  are independent and identically distributed (i.i.d.) exponential random variables with parameter  $\beta$ , then according to the result in Theorem 18.4, the MGF of  $Z = X_1 X_2 / x_1 + X_2$  can be simply given as

$$\mathcal{M}_Z(s) = \frac{2\beta}{4\beta + s} + \frac{4\beta^2 s}{\sigma_0^3} \ln \frac{2\beta + s + \sigma_0}{2\beta}, \quad (18.70)$$

where  $s > 0$  and  $\sigma_0 = \sqrt{4\beta s + s^2}$ . Note that we still do not see how the MGF expression in (18.58) in terms of hypergeometric function can be directly reduced to the simple closed-form solution (18.67) in Theorem 18.4. The approximation in (18.69) will provide a very simple solution for the SER calculations in (18.55) and (18.56).

**18.3.2.3 Closed-Form SER Expressions and Asymptotically Tight Approximation** Now let us apply the result of Theorem 18.4 to the harmonic mean of two random variables  $X_1 = P_1|h_{s,r}|^2/\mathcal{N}_0$  and  $X_2 = P_2|h_{r,d}|^2/\mathcal{N}_0$ . They are two independent exponential random variables with parameters  $\beta_1 = \mathcal{N}_0/(P_1\sigma_{s,r}^2)$  and  $\beta_2 = \mathcal{N}_0/(P_2\sigma_{r,d}^2)$ , respectively.

With the closed-form MGF expression in Theorem 18.4, the SER formulations in (18.55) and (18.56) for AF systems with  $M$ -PSK and  $M$ -QAM modulations can be determined, respectively, as

$$P_{\text{PSK}} \approx \frac{1}{\pi} \int_0^{(M-1)\pi/M} \frac{1}{1 + \frac{b_{\text{PSK}}}{\beta_0 \sin^2 \theta}} \left\{ \frac{(\beta_1 - \beta_2)^2 + (\beta_1 + \beta_2) \frac{b_{\text{PSK}}}{\sin^2 \theta}}{\sigma^2} + \frac{2\beta_1\beta_2 b_{\text{PSK}}}{\sigma^3 \sin^2 \theta} \ln \frac{(\beta_1 + \beta_2 + \frac{b_{\text{PSK}}}{\sin^2 \theta} + \sigma)^2}{4\beta_1\beta_2} \right\} d\theta, \quad (18.71)$$

$$P_{\text{QAM}} \approx \left[ \frac{4K}{\pi} \int_0^{\pi/2} - \frac{4K^2}{\pi} \int_0^{\pi/4} \right] \frac{1}{1 + \frac{b_{\text{QAM}}}{2\beta_0 \sin^2 \theta}} \left\{ \frac{(\beta_1 - \beta_2)^2 + (\beta_1 + \beta_2) \frac{b_{\text{QAM}}}{2 \sin^2 \theta}}{\sigma^2} + \frac{\beta_1\beta_2 b_{\text{QAM}}}{\sigma^3 \sin^2 \theta} \ln \frac{(\beta_1 + \beta_2 + \frac{b_{\text{QAM}}}{2 \sin^2 \theta} + \sigma)^2}{4\beta_1\beta_2} \right\} d\theta \quad (18.72)$$

in which  $\beta_0 = \mathcal{N}_0/(P_1\sigma_{s,d}^2)$ ,  $\beta_1 = \mathcal{N}_0/(P_1\sigma_{s,r}^2)$ ,  $\beta_2 = \mathcal{N}_0/(P_2\sigma_{r,d}^2)$ , and  $\sigma^2 = (\beta_1 - \beta_2)^2 + 2(\beta_1 + \beta_2)s + s^2$  with  $s = b_{\text{PSK}}/\sin^2 \theta$  for  $M$ -PSK modulation and  $s = b_{\text{QAM}}/(2 \sin^2 \theta)$  for  $M$ -QAM modulation. We observe that it is hard to understand the AF system performance based on the SER formulations in (18.71) and (18.72), even though they can be numerically calculated. In the following, we try to simplify the SER formulations by taking advantage of the MGF approximation in Theorem 18.4 to reveal the asymptotic performance of the AF cooperation systems.

We focus on the AF system with  $M$ -PSK modulation at first. Note that both  $\beta_1 = \mathcal{N}_0/(P_1\sigma_{s,r}^2)$  and  $\beta_2 = \mathcal{N}_0/(P_2\sigma_{r,d}^2)$  go to zero when the SNR goes to infinity. According to the MGF approximation (18.69) in Theorem 18.4, the SER formulation in (18.71) can be approximated as

$$\begin{aligned} P_{\text{PSK}} &\approx \frac{1}{\pi} \int_0^{(M-1)\pi/M} \frac{1}{1 + \frac{b_{\text{PSK}}}{\beta_0 \sin^2 \theta}} \cdot \frac{\beta_1 + \beta_2}{\frac{b_{\text{PSK}}}{\sin^2 \theta}} d\theta \\ &= \frac{1}{\pi} \int_0^{(M-1)\pi/M} \frac{(\beta_1 + \beta_2) \sin^4 \theta}{b_{\text{PSK}}(\sin^2 \theta + \frac{b_{\text{PSK}}}{\beta_0})} d\theta \end{aligned} \quad (18.73)$$

$$\approx \frac{B}{b_{\text{PSK}}^2} \beta_0(\beta_1 + \beta_2), \quad (18.74)$$

where

$$B = \frac{1}{\pi} \int_0^{(M-1)\pi/M} \sin^4 \theta d\theta = \frac{3(M-1)}{8M} + \frac{\sin \frac{2\pi}{M}}{4\pi} - \frac{\sin \frac{4\pi}{M}}{32\pi}.$$

To obtain the approximation in (18.74), we ignore the term  $\sin^2 \theta$  in the denominator in (18.73), which is negligible for sufficiently high SNR. Similarly, for the AF system with  $M$ -QAM modulation, the SER formulation in (18.72) can be approximated as

$$\begin{aligned} P_{\text{QAM}} &\approx \left[ \frac{4K}{\pi} \int_0^{\pi/2} - \frac{4K^2}{\pi} \int_0^{\pi/4} \right] \frac{1}{1 + \frac{b_{\text{QAM}}}{2\beta_0 \sin^2 \theta}} \cdot \frac{\beta_1 + \beta_2}{\frac{b_{\text{QAM}}}{2 \sin^2 \theta}} d\theta \\ &= \left[ \frac{4K}{\pi} \int_0^{\pi/2} - \frac{4K^2}{\pi} \int_0^{\pi/4} \right] \frac{4(\beta_1 + \beta_2) \sin^4 \theta}{b_{\text{QAM}}(2 \sin^2 \theta + \frac{b_{\text{QAM}}}{\beta_0})} d\theta \end{aligned} \quad (18.75)$$

$$\approx \frac{4B}{b_{\text{QAM}}^2} \beta_0 (\beta_1 + \beta_2), \quad (18.76)$$

where

$$B = \left[ \frac{4K}{\pi} \int_0^{\pi/2} - \frac{4K^2}{\pi} \int_0^{\pi/4} \right] \sin^4 \theta d\theta = \frac{3(M-1)}{8M} + \frac{K^2}{\pi}.$$

Since for sufficiently high SNR, the term  $2 \sin^2 \theta$  in the denominator in (18.75) is negligible, we ignore it to have the approximation in (18.76). We summarize the above discussion in the following theorem.

**Theorem 18.5** At sufficiently high SNR, the SER of the AF cooperation systems with  $M$ -PSK or  $M$ -QAM modulation can be approximated as

$$P_s \approx \frac{B \mathcal{N}_0^2}{b^2} \cdot \frac{1}{P_1 \sigma_{s,d}^2} \left( \frac{1}{P_1 \sigma_{s,r}^2} + \frac{1}{P_2 \sigma_{r,d}^2} \right), \quad (18.77)$$

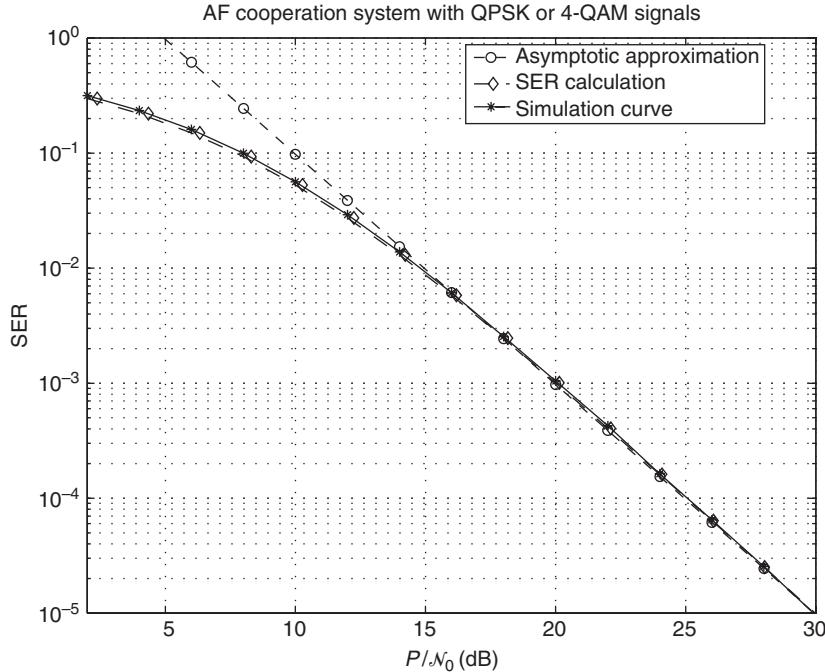
where in case of  $M$ -PSK signals,  $b = b_{\text{PSK}}$  and

$$B = \frac{3(M-1)}{8M} + \frac{\sin \frac{2\pi}{M}}{4\pi} - \frac{\sin \frac{4\pi}{M}}{32\pi}; \quad (18.78)$$

while in case of  $M$ -QAM signals,  $b = b_{\text{QAM}}/2$  and

$$B = \frac{3(M-1)}{8M} + \frac{K^2}{\pi}. \quad (18.79)$$

We compare the SER approximations (18.71), (18.72) and (18.77) with SER simulation result in Figure 18.6 in case of AF cooperation system with QPSK (or 4-QAM) modulation. It is easy to check that for both QPSK and 4-QAM modulations, the parameters  $B$  in (18.78) and (18.79) are the same, in which  $B = \frac{9}{32} + 1/4\pi$ . We can see that the theoretical calculation (18.71) or (18.72) matches with the simulation curve, except



**Figure 18.6** Comparison of the SER approximations and the simulation result for the AF cooperation system with  $Q$ -PSK or 4-QAM signals. We assumed that  $\sigma_{s,d}^2 = \sigma_{s,r}^2 = \sigma_{r,d}^2 = 1$ ,  $N_0 = 1$ , and  $P_1/P = 2/3$  and  $P_2/P = \frac{1}{3}$ .

for a little bit difference between them at low SNR, which is due to the approximation of the SNR  $\tilde{\gamma}_2$  in (18.51). Furthermore, the simple SER approximation in (18.77) is tight at high SNR, which is good enough to show the asymptotic performance of the AF cooperation system. From Theorem 18.5, we can conclude that the AF cooperation systems also provide an overall performance of diversity order 2, which is similar to that of DF cooperation systems.

**18.3.2.4 AF Optimum Power Allocation** In the following, we determine an asymptotic optimum power allocation for the AF cooperation systems based on the tight SER approximation in (18.77) for sufficiently high SNR.

For a fixed total transmitted power  $P_1 + P_2 = P$ , we are going to optimize  $P_1$  and  $P_2$  such that the asymptotically tight SER approximation in (18.77) is minimized. Equivalently, we try to minimize

$$G(P_1, P_2) = \frac{1}{P_1 \sigma_{s,d}^2} \left( \frac{1}{P_1 \sigma_{s,r}^2} + \frac{1}{P_2 \sigma_{r,d}^2} \right).$$

By taking derivative in terms of  $P_1$ , we have

$$\frac{\partial G(P_1, P_2)}{\partial P_1} = \frac{1}{P_1 \sigma_{s,d}^2} \left( -\frac{1}{P_1^2 \sigma_{s,r}^2} + \frac{1}{P_2^2 \sigma_{r,d}^2} \right) - \frac{1}{P_1^2 \sigma_{s,d}^2} \left( \frac{1}{P_1 \sigma_{s,r}^2} + \frac{1}{P_2 \sigma_{r,d}^2} \right).$$

By setting the above derivation as 0, we have  $\sigma_{s,r}^2(P_1^2 - P_1P_2) - 2\sigma_{r,d}^2P_2^2 = 0$ . Together with the power constraint  $P_1 + P_2 = P$ , we can solve the above equation and arrive at the following result.

**Theorem 18.6** For sufficiently high SNR, the optimum power allocation for the AF cooperation systems with either  $M$ -PSK or  $M$ -QAM modulation is

$$P_1 = \frac{\sigma_{s,r} + \sqrt{\sigma_{s,r}^2 + 8\sigma_{r,d}^2}}{3\sigma_{s,r} + \sqrt{\sigma_{s,r}^2 + 8\sigma_{r,d}^2}} P, \quad (18.80)$$

$$P_2 = \frac{2\sigma_{s,r}}{3\sigma_{s,r} + \sqrt{\sigma_{s,r}^2 + 8\sigma_{r,d}^2}} P. \quad (18.81)$$

From Theorem 18.6, we observe that the optimum power allocation for the AF cooperation systems is not modulation dependent, which is different from that for the DF cooperation systems in which the optimum power allocation depends on specific  $M$ -PSK or  $M$ -QAM modulation as stated in Theorem 18.2. This is due to the fact that in the AF cooperation systems, the relay amplifies the received signal and forwards it to the destination regardless what kind of received signal is. While in the DF cooperation systems, the relay forwards information to the destination only if the relay correctly decodes the received signal, and the decoding at the relay requires specific modulation information, which results in the modulation-dependent optimum power allocation scheme.

On the other hand, the asymptotic optimum power allocation scheme in Theorem 18.6 for the AF cooperation systems is similar to that in Theorem 18.2 for the DF cooperation systems, in the sense that both of them do not depend on the channel link between source and destination, and depend only on the channel link between source and relay and the channel link between relay and destination. Similarly, we can see from Theorem 18.6 that the optimum ratio of the transmitted power  $P_1$  at the source over the total power  $P$  is less than 1 and larger than  $\frac{1}{2}$ , while the optimum ratio of the power  $P_2$  used at the relay over the total power  $P$  is larger than 0 and less than  $\frac{1}{2}$ . In general, the equal power strategy is not optimum. For example, if  $\sigma_{s,r}^2 = \sigma_{r,d}^2$ , then the optimum power allocation is  $P_1 = \frac{2}{3}P$  and  $P_2 = \frac{1}{3}P$ .

### 18.3.3 Comparison of DF and AF Cooperation Gains

Based on the asymptotically tight SER approximations and the optimum power allocation solutions we established in the previous two sections, we determine in this section the overall cooperation gain and diversity order for the DF and AF cooperation systems, respectively. Then, we are able to compare the cooperation gain between the DF and AF cooperation protocols.

Let us first focus on the DF cooperation protocol. According to the asymptotically tight SER approximation (18.35) in Theorem 18.1, we know that for sufficiently high SNR, the SER performance of the DF cooperation systems can be approximated as

$$P_s \approx \frac{\mathcal{N}_0^2}{b^2} \cdot \frac{1}{P_1\sigma_{s,d}^2} \left( \frac{A^2}{P_1\sigma_{s,r}^2} + \frac{B}{P_2\sigma_{r,d}^2} \right), \quad (18.82)$$

where  $A$  and  $B$  are specified in (18.36) and (18.37) for  $M$ -PSK and  $M$ -QAM signals, respectively. By substituting the asymptotic optimum power allocation (18.43) and (18.44) into (18.82), we have

$$P_s \approx \sigma_{\text{DF}}^{-2} \left( \frac{P}{N_0} \right)^{-2}, \quad (18.83)$$

where

$$\sigma_{\text{DF}} = \frac{2\sqrt{2}b\sigma_{s,d}\sigma_{s,r}\sigma_{r,d}}{\sqrt{B}} \frac{\left(\sigma_{s,r} + \sqrt{\sigma_{s,r}^2 + 8(A^2/B)\sigma_{r,d}^2}\right)^{1/2}}{\left(3\sigma_{s,r} + \sqrt{\sigma_{s,r}^2 + 8(A^2/B)\sigma_{r,d}^2}\right)^{3/2}}, \quad (18.84)$$

in which  $b = b_{\text{PSK}}$  for  $M$ -PSK signals and  $b = b_{\text{QAM}}/2$  for  $M$ -QAM signals. From (18.83), we can see that the DF cooperation systems can guarantee a performance diversity of order 2. Note that the term  $\sigma_{\text{DF}}$  in (18.84) depends only on the statistics of the channel links. We call it the *cooperation gain* of the DF cooperation systems, which indicates the best performance gain that we are able to achieve through the DF cooperation protocol with any kind of power allocation. If the link quality between source and relay is much less than that between relay and destination, that is,  $\sigma_{s,r}^2 \ll \sigma_{r,d}^2$ , then the cooperation gain is approximated as

$$\sigma_{\text{DF}} = \frac{b\sigma_{s,d}\sigma_{s,r}}{A}$$

in which

$$A = \frac{M-1}{2M} + \frac{\sin \frac{2\pi}{M}}{4\pi} \rightarrow \frac{1}{2}$$

( $M$  large) for  $M$ -PSK modulation, or

$$A = \frac{M-1}{2M} + \frac{K^2}{\pi} \rightarrow \frac{1}{2} + \frac{1}{\pi}$$

( $M$  large) for  $M$ -QAM modulation. For example, in case of QPSK modulation,

$$A = \frac{3}{8} + \frac{1}{4\pi} = 0.4546.$$

On the other hand, if the link quality between source and relay is much larger than that between relay and destination, that is,  $\sigma_{s,r}^2 > > \sigma_{r,d}^2$ , then the cooperation gain can be approximated as

$$\sigma_{\text{DF}} = \frac{b\sigma_{s,d}\sigma_{r,d}}{2\sqrt{B}}$$

in which

$$B = \frac{3(M-1)}{8M} + \frac{\sin \frac{2\pi}{M}}{4\pi} - \frac{\sin \frac{4\pi}{M}}{32\pi} \rightarrow \frac{3}{8}$$

( $M$  large) for  $M$ -PSK modulation, or

$$B = \frac{3(M-1)}{8M} + \frac{K^2}{\pi} \rightarrow \frac{3}{8} + \frac{1}{\pi}$$

( $M$  large) for  $M$ -QAM modulation. For example, in case of QPSK modulation,

$$B = \frac{9}{32} + \frac{1}{4\pi} = 0.3608.$$

Similarly, for the AF cooperation protocol, from the asymptotically tight SER approximation (18.77) in Theorem 18.5, we can see that for sufficiently high SNR, the SER performance of the AF cooperation systems can be approximated as

$$P_s \approx \frac{B\mathcal{N}_0^2}{b^2} \cdot \frac{1}{P_1\sigma_{s,d}^2} \left( \frac{1}{P_1\sigma_{s,r}^2} + \frac{1}{P_2\sigma_{r,d}^2} \right), \quad (18.85)$$

where  $b = b_{\text{PSK}}$  for  $M$ -PSK signals and  $b = b_{\text{QAM}}/2$  for  $M$ -QAM signals, and  $B$  is specified in (18.78) and (18.79) for  $M$ -PSK and  $M$ -QAM signals, respectively. By substituting the asymptotic optimum power allocation (18.80) and (18.81) into (18.85), we have

$$P_s \approx \sigma \text{AF}^{-2} \left( \frac{P}{\mathcal{N}_0} \right)^{-2}, \quad (18.86)$$

$$\sigma_{\text{AF}} = \frac{2\sqrt{2}b\sigma_{s,d}\sigma_{s,r}\sigma_{r,d}}{\sqrt{B}} \frac{\left( \sigma_{s,r} + \sqrt{\sigma_{s,r}^2 + 8\sigma_{r,d}^2} \right)^{1/2}}{\left( 3\sigma_{s,r} + \sqrt{\sigma_{s,r}^2 + 8\sigma_{r,d}^2} \right)^{3/2}}, \quad (18.87)$$

which is termed the *cooperation gain* of the AF cooperation systems that indicates the best asymptotic performance gain of the AF cooperation protocol with the optimum power allocation scheme. From (18.86), we can see that the AF cooperation systems can also guarantee a performance diversity of order 2, which is similar to that of the DF cooperation systems.

Since both the AF and DF cooperation systems are able to achieve a performance diversity of order 2, it is interesting to compare their cooperation gain. Let us define a ratio  $\lambda = \sigma_{\text{DF}}/\sigma \text{AF}$  to indicate the performance gain of the DF cooperation protocol compared with the AF protocol. According to (18.84) and (18.87), we have

$$\lambda = \left( \frac{\sigma_{s,r} + \sqrt{\sigma_{s,r}^2 + 8(A^2/B)\sigma_{r,d}^2}}{\sigma_{s,r} + \sqrt{\sigma_{s,r}^2 + 8\sigma_{r,d}^2}} \right)^{1/2} \left( \frac{3\sigma_{s,r} + \sqrt{\sigma_{s,r}^2 + 8\sigma_{r,d}^2}}{\sigma_{s,r} + \sqrt{3\sigma_{s,r}^2 + 8(A^2/B)\sigma_{r,d}^2}} \right)^{3/2} \quad (18.88)$$

where  $A$  and  $B$  are specified in (18.36) and (18.37) for  $M$ -PSK and  $M$ -QAM signals, respectively. We further discuss the ratio  $\lambda$  for the following three cases.

**Case 1** If the channel link quality between source and relay is much less than that between relay and destination, that is,  $\sigma_{s,r}^2 \ll \sigma_{r,d}^2$ , then

$$\lambda = \frac{\sigma_{\text{DF}}}{\sigma \text{AF}} \rightarrow \frac{\sqrt{B}}{A}. \quad (18.89)$$

In case of BPSK modulation,  $A = \frac{1}{4}$  and  $B = \frac{3}{16}$ , so  $\lambda = \sqrt{3} > 1$ . In case of QPSK modulation,  $A = \frac{3}{8} + 1/4\pi$  and  $B = \frac{9}{32} + 1/4\pi$ , so  $\lambda = 1.3214 > 1$ . In general, for  $M$ -PSK modulation ( $M$  large),

$$A = \frac{M-1}{2M} + \frac{\sin \frac{2\pi}{M}}{4\pi} \rightarrow \frac{1}{2} \quad \text{and} \quad B = \frac{3(M-1)}{8M} + \frac{\sin \frac{2\pi}{M}}{4\pi} - \frac{\sin \frac{4\pi}{M}}{32\pi} \rightarrow \frac{3}{8},$$

so

$$\lambda \rightarrow \frac{\sqrt{6}}{2} \approx 1.2247 > 1.$$

For  $M$ -QAM modulation ( $M$  large),

$$A = \frac{M-1}{2M} + \frac{K^2}{\pi} \rightarrow \frac{1}{2} + \frac{1}{\pi} \quad \text{and} \quad B = \frac{3(M-1)}{8M} + \frac{K^2}{\pi} \rightarrow \frac{3}{8} + \frac{1}{\pi},$$

$$\lambda \rightarrow \frac{\sqrt{\frac{3}{8} + \frac{1}{\pi}}}{\frac{1}{2} + \frac{1}{\pi}} \approx 1.0175 > 1.$$

We can see that if  $\sigma_{s,r}^2 \ll \sigma_{r,d}^2$ , the cooperation gain of the DF systems is always larger than that of the AF systems for both  $M$ -PSK and  $M$ -QAM modulations. The advantage of the DF cooperation systems is more significant if  $M$ -PSK modulation is used.

**Case 2** If the channel link quality between source and relay is much better than that between relay and destination, that is,  $\sigma_{s,r}^2 > \sigma_{r,d}^2$ , from (18.88) we have  $\lambda = \sigma_{\text{DF}}/\sigma_{\text{AF}} \rightarrow 1$ . This implies that if  $\sigma_{s,r}^2 > \sigma_{r,d}^2$ , the performance of the DF cooperation systems is almost the same as that of the AF cooperation systems for both  $M$ -PSK and  $M$ -QAM modulations. Since the DF cooperation protocol requires decoding process at the relay, we may suggest the use of the AF cooperation protocol in this case to reduce the system complexity.

**Case 3** If the channel link quality between source and relay is the same as that between relay and destination, that is,  $\sigma_{s,r}^2 = \sigma_{r,d}^2$ , we have

$$\lambda = \left( \frac{1 + \sqrt{1 + 8(A^2/B)}}{4} \right)^{1/2} \left( \frac{6}{3 + \sqrt{1 + 8(A^2/B)}} \right)^{3/2}.$$

In case of BPSK modulation,  $A = \frac{1}{4}$  and  $B = \frac{3}{16}$ , so  $\lambda \approx 1.1514 > 1$ . In case of QPSK modulation,  $A = \frac{3}{8} + 1/4\pi$  and  $B = \frac{9}{32} + 1/4\pi$ , so  $\lambda \approx 1.0851 > 1$ . In general, for  $M$ -PSK modulation ( $M$  large),

$$A = \frac{M-1}{2M} + \frac{\sin \frac{2\pi}{M}}{4\pi} \rightarrow \frac{1}{2} \quad \text{and} \quad B = \frac{3(M-1)}{8M} + \frac{\sin \frac{2\pi}{M}}{4\pi} - \frac{\sin \frac{4\pi}{M}}{32\pi} \rightarrow \frac{3}{8},$$

so

$$\lambda \rightarrow \left( \frac{1 + \sqrt{1 + 16/3}}{4} \right)^{1/2} \left( \frac{6}{3 + \sqrt{1 + 16/3}} \right)^{3/2} \approx 1.0635 > 1.$$

For  $M$ -QAM modulation ( $M$  large),

$$A = \frac{M-1}{2M} + \frac{K^2}{\pi} \rightarrow \frac{1}{2} + \frac{1}{\pi} \quad \text{and} \quad B = \frac{3(M-1)}{8M} + \frac{K^2}{\pi} \rightarrow \frac{3}{8} + \frac{1}{\pi},$$

$$\lambda \rightarrow \left( \frac{1 + \sqrt{1 + 8(\frac{1}{2} + \frac{1}{\pi})^2 / (\frac{3}{8} + \frac{1}{\pi})}}{4} \right)^{1/2} \left( \frac{6}{3 + \sqrt{1 + 8(\frac{1}{2} + \frac{1}{\pi})^2 / (\frac{3}{8} + \frac{1}{\pi})}} \right)^{3/2} \approx 1.0058.$$

We can see that if the modulation size is large, the performance advantage of the DF cooperation protocol is negligible compared with the AF cooperation protocol. Actually, with QPSK modulation, the ratio of the cooperation gain is  $\lambda \approx 1.0851$ , which is already small.

From the above discussion, we can see that the performance of the DF cooperation protocol is always not less than that of the AF cooperation protocol. However, the performance advantage of the DF cooperation protocol is not significant unless (i) the channel link quality between the relay and the destination is much stronger than that between the source and the relay, and (ii) the constellation size of the signaling is small. There are trade-offs between these two cooperation protocols. The complexity of the AF cooperation protocol is less than that of the DF cooperation protocol in which decoding process at the relay is required. For high data rate cooperative communications (with large modulation size), we may use the AF cooperation protocol to reduce the system complexity while the performance is comparable.

## 18.4 ENERGY EFFICIENCY IN COOPERATIVE SENSOR NETWORKS

Up to this point the overhead of cooperative communications and introducing a relay channel has not been considered. This overhead will reduce the gains demonstrated by cooperative communications. In this section, we model this overhead and study its impact on cooperative communications. In particular we try to answer the question when is cooperative communications more energy efficient than direct transmission. To make the analysis more tractable, we consider an outage probability analysis framework [15].

The gains of cooperative diversity were established under the ideal model of negligible listening and computing power. In sensor networks, and depending on the type of motes used, the power consumed in receiving and processing may constitute a significant portion of the total consumed power. Cooperative diversity can provide gains in terms of savings in the required transmit power in order to achieve a certain performance requirement because of the spatial diversity it adds to the system. However, if one takes into account the extra processing and receiving power consumption at the relay and destination nodes required for cooperation, then there is obviously a trade-off between the gains in the transmit power and the losses due to the receive and processing powers when applying cooperation. Hence such a trade-off between the gains promised by cooperation and this extra overhead in terms of the energy efficiency of the system should be taken into consideration in the network design.

In this section the gains of cooperation under such extra overhead are studied. Moreover, some practical system parameters as the power amplifier loss, the quality of service (QoS) required, the relay location, and the optimal number of relays are considered.

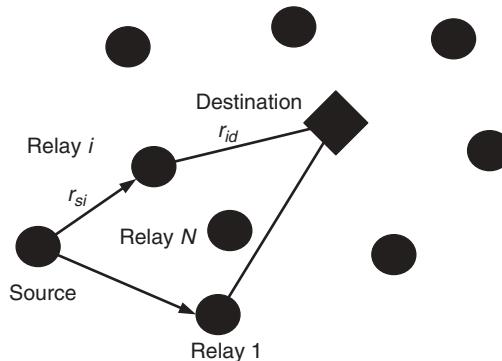
Two communications architectures are considered, direct transmission and cooperative transmission. The performance metric for comparison between the two architectures is the energy efficiency of the communication scheme. More specifically, for both architectures the optimal total power consumption to achieve certain QoS requirements and is computed, and the cooperation gain defined as the ratio between the power required for direct transmission and cooperation. When this ratio is smaller than one, this indicates that direct transmission is more energy efficient, and that the extra overhead induced by cooperation outweighs its gains in the transmit power. Comparisons between optimal power allocation at the source and relay nodes and equal power allocation are demonstrated. The results reveal that under some scenarios, equal power allocation is almost equivalent to optimal power allocation. The effect of relay location on the performance is investigated to provide guidelines for relay assignment algorithms.

#### 18.4.1 System Model

Consider a single source–destination pair separated by distance  $r_{s,d}$ . The number of potential relays available to help the source is  $N$ . This is illustrated in Figure 18.7, where the distances between source and relay  $i$ , and relay  $i$  and destination are  $r_{s,i}$  and  $r_{i,d}$ , respectively, and  $i \in \{1, 2, \dots, N\}$ . First we analyze the performance of the single relay scenario, and later we extend the results for arbitrary finite  $N$ .

We compare the performance of two communication scenarios. In the first scenario only direct transmission between the source and destination nodes is allowed, and this accounts for conventional direct transmission. In direct transmission, if the channel link between the source and destination encounters a deep fade or strong shadowing, for example, then the communication between these two nodes fails. Moreover, if the channel is slowly varying, which is the case in sensor networks due to the stationarity or limited mobility of the nodes, then the channel might remain in the deep fade state for a long time (strong time correlation), hence conventional automatic repeat request (ARQ) might not help in this case.

In the second communication scenario, we consider a two-phase cooperation protocol. In the first phase, the source transmits a signal to the destination, and due to the broadcast nature of the wireless medium the relay can overhear this signal. If the destination receives the packet from this phase correctly, then it sends back an acknowledgment (ACK) and the relay just idles. On the other hand, if the destination



**Figure 18.7** System model of the multirelay scenario.

cannot decode the received packet correctly, then it sends back a negative acknowledgement (NACK). In this case, if the relay was able to receive the packet correctly in the first phase, then it forwards it to the destination. So the idea behind this cooperation protocol is to introduce a new ARQ in another domain, which is the spatial domain, as the links between different pairs of nodes in the network fade independently. The assumptions of high temporal correlation and independence in the spatial domain will be verified through experiments as discussed in Section 18.5.

Next the wireless channel and system models are described. We consider a sensor network in which the link between any two nodes in the network is subject to narrowband Rayleigh fading, propagation path loss, and additive white Gaussian noise (AWGN). The channel fades for different links are assumed to be statistically mutually independent. This is a reasonable assumption as the nodes are usually spatially well separated. For medium access, the nodes are assumed to transmit over orthogonal channels, thus no mutual interference is considered in the signal model. All nodes in the network are assumed to be equipped with single-element antennas, and transmission at all nodes is constrained to the half-duplex mode, that is, any terminal cannot transmit and receive simultaneously.

The power consumed in a transmitting or receiving stage is described as follows. If a node transmits with power  $P$ , only  $P(1 - \alpha)$  is actually utilized for radio-frequency (RF) transmission, where  $(1 - \alpha)$  accounts for the efficiency of the RF power amplifier, which generally has a nonlinear gain function. The processing power consumed by a transmitting node is denoted by  $P_c$ . Any receiving node consumes  $P_r$  power units to receive the data. The values of the parameters  $\alpha$ ,  $P_r$ ,  $P_c$  are assumed the same for all nodes in the network and are specified by the manufacturer. Following, we describe the received signal model for both direct and cooperative transmissions.

First, we describe the received signal model for the direct transmission mode. In the direct transmission scheme, which is employed in current wireless networks, each user transmits his signal directly to the next node in the route, which we denote as the destination  $d$  here. The signal received at the destination  $d$  from source user  $s$  can be modeled as

$$y_{s,d} = \sqrt{P_s^D(1 - \alpha)r_{s,d}^{-\gamma}}h_{s,d}x + n_{s,d}, \quad (18.90)$$

where  $P_s^D$  is the transmission power from the source in the direct communication scenario,  $x$  is the transmitted data with unit power, and  $h_{s,d}$  is the channel fading gain between the two terminals  $s$  and  $d$ . The channel fade of any link is modeled as a zero-mean circularly symmetric complex Gaussian random variable with unit variance. In (18.90),  $\gamma$  is the path loss exponent, and  $r_{s,d}$  is the distance between the two terminals. The term  $n_{s,d}$  in (18.90) denotes additive noise; the noise components throughout the paper are modeled as AWGN with variance  $N_o$ .

Second, we describe the signal model for cooperative transmission. The cooperative transmission scenario comprises two phases as illustrated before. The signals received from the source at the destination  $d$  and relay 1 in the first stage can be modeled, respectively, as

$$y_{s,d} = \sqrt{P_s^c(1 - \alpha)r_{s,d}^{-\gamma}}h_{s,d}x + n_{s,d}, \quad (18.91)$$

$$y_{s,1} = \sqrt{P_s^c(1 - \alpha)r_{s,1}^{-\gamma}}h_{s,1}x + n_{s,1}, \quad (18.92)$$

where  $P_s^c$  is the transmission power from the source in the cooperative scenario. The channel gains  $h_{s,d}$  and  $h_{s,1}$  between the source–destination and source–relay are modeled as zero-mean circular symmetric complex Gaussian random variables with zero mean. If the SNR of the signal received at the destination from the source falls below the threshold  $\beta$ , the destination broadcasts a NACK. In this case, if the relay was able to receive the packet from the source correctly in the first phase, it forwards the packet to the destination with power  $P_1$

$$y_{1,d} = \sqrt{P_1(1-\alpha)r_{1,d}^{-\gamma}}h_{1,d}x + n_{1,d}. \quad (18.93)$$

Cooperation results in additional spatial diversity by introducing this artificial multipath through the relay link. This can enhance the transmission reliability against wireless channel impairments as fading but will also result in extra receiving and processing power. In the next section, we discuss this in more detail.

#### 18.4.2 Performance Analysis and Optimum Power Allocation

We characterize the system performance in terms of outage probability. Outage is defined as the event that the received SNR falls below a certain threshold  $\beta$ , hence, the probability of outage  $P_O$  is defined as,

$$P_O = \mathcal{P}(\text{SNR} \leq \beta). \quad (18.94)$$

If the received SNR is higher than the threshold  $\beta$ , the receiver is assumed to be able to decode the received message with negligible probability of error. If an outage occurs, the packet is considered lost. The SNR threshold  $\beta$  is determined according to the application and the transmitter/receiver structure. For example, larger values of  $\beta$  is required for applications with higher QoS requirements. Also increasing the complexity of transmitter and/or receiver structure, for example, applying strong error coding schemes, can reduce the value of  $\beta$  for the same QoS requirements.

Based on the derived outage probability expressions, we can formulate a constrained optimization problem to minimize the total consumed power subject to a given outage performance. We then compare the total consumed power for the direct and cooperative scenarios to quantify the energy savings, if any, gained by applying cooperative transmission.

**18.4.2.1 Direct Transmission** As discussed before, the outage is defined as the event that the received SNR falls below a predefined threshold, which we denoted by  $\beta$ . From the received signal model in (18.90), the received SNR from a user at a distance  $r_{s,d}$  from the destination is given by

$$\text{SNR}(r_{s,d}) = \frac{|h_{s,d}|^2 r_{s,d}^{-\gamma} P_s^D (1-\alpha)}{N_o}, \quad (18.95)$$

where  $|h_{s,d}|^2$  is the magnitude square of the channel fade and follows an exponential distribution with unit mean; this follows because of the Gaussian zero-mean distribution

of  $h_{s,d}$ . Hence, the outage probability for the direct transmission mode  $P_{OD}$  can be calculated as

$$\mathcal{P}_{OD} = \mathcal{P}(\text{SNR}(r_{s,d}) \leq \beta) = 1 - \exp\left(-\frac{N_o \gamma r_{s,d}^\gamma}{(1-\alpha) P_s^D}\right). \quad (18.96)$$

The total transmitted power  $P_{tot}^D$  for the direct transmission mode is given by

$$P_{tot}^D = P_s^D + P_c + P_r \quad (18.97)$$

where  $P_s^D$  is the power consumed at the RF stage of the source node,  $P_c$  is the processing power at the source node, and  $P_r$  is the receiving power at the destination. The requirement is to minimize this total transmitted power subject to the constraint that we meet a certain QoS requirement that the outage probability is less than a given outage requirement, which we denote by  $\mathcal{P}_{out}^*$ . Since both the processing and receiving powers are fixed, the only variable of interest is the transmitting power  $P_s^D$ .

The optimization problem can be formulated as follows:

$$\min_{P_s^D} P_{tot}^D, \quad (18.98)$$

$$\text{s.t. } \mathcal{P}_{OD} \leq \mathcal{P}_{out}^*, \quad (18.99)$$

(where s.t. is such that). The outage probability  $\mathcal{P}_{OD}$  is a decreasing function in the power  $P_s^D$ . Substituting  $\mathcal{P}_{out}^*$  in the outage expression in (18.96), we get after some simple arithmetic that the optimal transmitting power is given by

$$P_s^{D*} = -\frac{\beta N_o r_{s,d}^\gamma}{(1-\alpha) \ln(1 - \mathcal{P}_{out}^*)}. \quad (18.100)$$

The minimum total power required for direct transmission in order to achieve the required QoS requirement is therefore given by

$$P_{tot}^* = P_c + P_r - \frac{\beta N_o r_{s,d}^\gamma}{(1-\alpha) \ln(1 - \mathcal{P}_{out}^*)}. \quad (18.101)$$

In the next section we formulate the optimal power allocation problem for the cooperative communication scenario.

**18.4.2.2 Cooperative Transmission** For the optimal power allocation problem in cooperative transmission, we consider two possible scenarios.

- In the first scenario, the relay is allowed to transmit with different power than the source and hence the optimization space is two dimensional: source and relay power allocations. The solution for this setting provides the minimum possible total consumed power. However, the drawback of this setting is that the solution for the optimization problem is complex and might not be feasible to implement in sensor nodes.
- The second setting that we consider is constraining the source and relay nodes to transmit with equal powers. This is much easier to implement as the optimization

space is one dimensional in this case; moreover, a relaxed version of the optimization problem can render a closed-form solution.

Clearly the solution of the equal power allocation problem provides a suboptimal solution to the general case in which we allow different power allocations at the source and the relay. It is interesting then to investigate the conditions under which these two power allocation strategies have close performance.

First, we characterize the optimal power allocations at the source and relay nodes. Consider a source–destination pair that are  $r_{sd}$  units distance. Let us compute the conditional outage probability for given locations of the source and the helping relay. As discussed before, cooperative transmission encompasses two phases. Using (18.91), the SNR received at the destination  $d$  and relay 1 from the source  $s$  in the first phase are given by

$$\text{SNR}_{s,d} = \frac{|h_{s,d}|^2 r_{s,d}^{-\gamma} P_s^C (1 - \alpha)}{N_o}, \quad (18.102)$$

$$\text{SNR}_{s,1} = \frac{|h_{s,1}|^2 r_{s,1}^{-\gamma} P_s^C (1 - \alpha)}{N_o}. \quad (18.103)$$

While from (18.93), the SNR received at the destination from the relay in the second phase is given by

$$\text{SNR}_{1,d} = \frac{|h_{1,d}|^2 r_{1,d}^{-\gamma} P_1 (1 - \alpha)}{N_o}. \quad (18.104)$$

Note that the second phase of transmission is only initiated if the packet received at the destination from the first transmission phase is not correctly received. The terms  $|h_{s,d}|^2$ ,  $|h_{s,1}|^2$ , and  $|h_{1,d}|^2$  are mutually independent exponential random variables with unit mean.

The outage probability of the cooperative transmission  $\mathcal{P}_{OC}$  can be calculated as follows:

$$\mathcal{P}_{OC} = \mathcal{P}((\text{SNR}_{s,d} \leq \beta) \cap (\text{SNR}_{s,l} \leq \beta)) \quad (18.105)$$

$$+ \mathcal{P}((\text{SNR}_{s,d} \leq \beta) \cap (\text{SNR}_{l,d} \leq \beta) \cap (\text{SNR}_{s,l} > \beta)) \quad (18.106)$$

$$= (1 - f(r_{s,d}, P_s^C)) (1 - f(r_{s,l}, P_s^C)) \\ + (1 - f(r_{s,d}, P_s^C)) (1 - f(r_{l,d}, P_l)) f(r_{s,l}, P_s^C), \quad (18.107)$$

where

$$f(x, y) = \exp \left[ -\frac{N_o \beta x^\gamma}{y(1 - \alpha)} \right]. \quad (18.108)$$

The first term in the above expression corresponds to the event that both the source–destination and the source–relay channels are in outage, and the second term corresponds to the event that both the source–destination and the relay–destination channels are in outage while the source–relay channel is not. The above expression can be simplified as follows:

$$\mathcal{P}_{OC} = (1 - f(r_{s,d}, P_s^C)) (1 - f(r_{l,d}, P_l)) f(r_{s,l}, P_s^C). \quad (18.109)$$

The total average consumed power for cooperative transmission to transmit a packet is given by

$$\begin{aligned} E[P_{\text{tot}}^C] = & (P_s^C + P_c + 2P_r)\mathcal{P}(\text{SNR}_{s,d} \geq \beta) \\ & + (P_s^C + P_c + 2P_r)\mathcal{P}(\text{SNR}_{s,d} < \beta)\mathcal{P}(\text{SNR}_{s,1} < \beta) \\ & + (P_s^C + P_1 + 2P_c + 3P_r)\mathcal{P}(\text{SNR}_{s,d} < \beta)\mathcal{P}(\text{SNR}_{s,1} > \beta), \end{aligned} \quad (18.110)$$

where the first term in the right-hand side corresponds to the event that the direct link in the first phase is not in outage; therefore, the total consumed power is only given by that of the source node, and the 2 in front of the received power term  $P_r$  is to account for the relay receiving power. The second term in the summation corresponds to the event that both the direct and the source–relay links are in outage, hence the total consumed power is still given as in the first term. The last term in the total summation accounts for the event that the source–destination link is in outage while the source–relay link is not, and hence we need to account for the relay transmitting and processing powers and the extra receiving power at the destination. Using the Rayleigh fading channel model, the average total consumed power can be given as follows:

$$\begin{aligned} P_{\text{tot}}^C = & (P_s^C + P_c + 2P_r)f(r_{s,d}, P_s^C) \\ & + (P_s^C + P_c + 2P_r)(1 - f(r_{s,d}, P_s^C))(1 - f(r_{s,l}, P_s^C)) \\ & + (P_s^C + P_1 + 2P_c + 3P_r)(1 - f(r_{s,d}, P_s^C)) \times f(r_{s,l}, P_s^C). \end{aligned} \quad (18.111)$$

We can formulate the power minimization problem in a similar way to (18.98) with the difference that there are two optimization variables in the cooperative transmission mode, namely, the transmit powers  $P_s^C$  and  $P_1$  at the source and relay nodes, respectively. The optimization problem can be stated as follows:

$$\min_{P_s^C, P_1} P_{\text{tot}}^C(P_s^C, P_1), \quad (18.112)$$

$$\text{s.t. } \mathcal{P}_{\text{OC}}(P_s^C, P_1) \leq \mathcal{P}_{\text{out}}^*. \quad (18.113)$$

This optimization problem is nonlinear and does not admit a closed-form solution. Therefore, we resort to numerical optimization techniques in order to solve for this power allocation problem at the relay and source nodes, and the results are shown in the simulations section.

In the above formulation we considered optimal power allocation at the source and relay node in order to meet the outage probability requirement. The performance attained by such an optimization problem provides a benchmark for the cooperative transmission scheme. However, in a practical setting, it might be difficult to implement such a complex optimization problem at the sensor nodes. A more practical scenario would be that all the nodes in the network utilize the same power for transmission. Denote the equal transmission power in this case by  $P_{\text{CE}}$ ; the optimization problem in this case can be formulated as

$$\min_{P_{\text{CE}}} P_{\text{tot}}^C(P_{\text{CE}}), \quad (18.114)$$

$$\text{s.t. } \mathcal{P}_{\text{OC}}(P_{\text{CE}}) \leq \mathcal{P}_{\text{out}}^*. \quad (18.115)$$

Beside being a one-dimensional optimization problem that can be easily solved, the problem can be relaxed to render a closed-form solution. Note that at enough high SNR the following approximation holds  $\exp(-x) \simeq (1 - x)$ ; where  $x$  here is proportional to  $1/\text{SNR}$ .

Using the above approximation in (18.111), and after some mathematical manipulation, the total consumed power can be approximated as follows:

$$P_{\text{tot}}^C \simeq P_{\text{CE}} + P_c + 2P_r + (P_{\text{CE}} + P_c + P_r) \frac{k_1}{P_{\text{CE}}} - (P_{\text{CE}} + P_c + P_r) \frac{k_1 k_2}{P_{\text{CE}}^2}. \quad (18.116)$$

Similarly, the outage probability can be written as follows:

$$\mathcal{P}_{\text{OC}} \simeq \frac{k_1 k_2}{P_{\text{CE}}^2} + \frac{k_1 k_3}{P_{\text{CE}}^2} - \frac{k_1 k_2 k_3}{P_{\text{CE}}^3}, \quad (18.117)$$

where

$$k_1 = \frac{\beta N_o r_{s,d}^\gamma}{1 - \alpha}, \quad k_2 = \frac{\beta N_o r_{s,l}^\gamma}{1 - \alpha}, \quad k_3 = \frac{\beta N_o r_{l,d}^\gamma}{1 - \alpha}.$$

This is a constrained optimization problem in one variable and its Lagrangian is given by

$$\frac{\partial P_{\text{tot}}^C}{\partial P_{\text{CE}}} + \lambda \frac{\partial \mathcal{P}_{\text{OC}}}{\partial P_{\text{CE}}} = 0 \quad (18.118)$$

where the derivatives of the total power consumption  $P_{\text{tot}}^C$  and the outage probability  $\mathcal{P}_{\text{OC}}$  with respect to the transmit power  $P_{\text{CE}}$  are given by

$$\frac{\partial P_{\text{tot}}^C}{\partial P_{\text{CE}}} = 1 + \frac{k_1 k_2 - (P_c + P_r) k_1}{P_{\text{CE}}^2} + \frac{2k_1 k_2 (P_c + P_r)}{P_{\text{CE}}^3}, \quad (18.119)$$

$$\frac{\partial \mathcal{P}_{\text{OC}}}{\partial P_{\text{CE}}} = \frac{-2(k_1 k_2 + k_1 k_3)}{P_{\text{CE}}^3} + \frac{3k_1 k_2 k_3}{P_{\text{CE}}^4}, \quad (18.120)$$

respectively. Substituting the derivatives in (18.119) into the Lagrangian in (18.118), and performing simple change of variables  $1/P_{\text{CE}} = x$ , the Lagrangian can be written in the following simple polynomial form:

$$1 + [k_1 k_2 - (P_c + P_r) k_1] x^2 + 2[k_1 k_2 (P_c + P_r) - \lambda(k_1 k_2 + k_1 k_3)] x^3 + 3\lambda k_1 k_2 k_3 x^4 = 0, \quad (18.121)$$

under the outage constraint

$$(k_1 k_2 + k_1 k_3) x^2 - k_1 k_2 k_3 x^3 = \mathcal{P}_{\text{out}}^*. \quad (18.122)$$

The constraint equation above is only a polynomial of order 3, so it can be easily solved and we can find the root that minimizes the cost function.

### 18.4.3 Multirelay Scenario

In this section, we extend the study to the case when there is more than one potential relay. Let  $N$  be the number of relays assigned to help a given source. The cooperation protocol then works as an  $N$ -stage ARQ protocol as follows. The source node transmits its packets to the destination and the relays try to decode this packet. If the destination does not decode the packet correctly, it sends a NACK, that can be heard by the relays. If the first relay is able to decode the packet correctly, it forwards the packet with power  $P_1$  to the destination. If the destination does not receive correctly again, then it sends a NACK, and the second candidate relay, if it received the packet correctly, forwards the source's packet to the destination with power  $P_2$ . This is repeated until the destination gets the packet correctly or the  $N$  trials corresponding to the  $N$  relays are exhausted.

We model the status of any relay by 1 or 0, corresponding to whether the relay received the source's packet correctly or not, respectively. Writing the status of all the relays in a column vector results in a  $N \times 1$  vector whose entries are either 0 or 1. Hence, the decimal number representing this  $N \times 1$  vector can take any integer value between 0 and  $2^N - 1$ . Denote this vector by  $S_k$  where  $k \in \{0, 1, 2, \dots, 2^N - 1\}$ .

For a given status of the  $N$  relays, an outage occurs if and only if the links between the relays that decoded correctly and the destination are all in outage. Denote the set of the relays that received correctly by  $\chi(S_k) = \{i : S_k(i) = 1, 1 \leq i \leq N\}$ , and  $\chi^c(S_k)$  as the set of relays that have not received correctly, that is,  $\chi^c(S_k) = \{i : S_k(i) = 0, 1 \leq i \leq N\}$ . The conditional probability of outage given the relays status  $S_k$  is thus given by

$$\mathcal{P}_{\text{OC}|S_k} = \mathcal{P} \left( \text{SNR}_{s,d} \leq \beta \bigcap_{j \in \chi(S_k)} (\text{SNR}_{j,d} \leq \beta) \right). \quad (18.123)$$

The total outage probability is thus given by

$$\mathcal{P}_{\text{OC}} = \sum_{k=0}^{2^N-1} \mathcal{P}(S_k) \mathcal{P}_{\text{OC}|S_k}. \quad (18.124)$$

We then need to calculate the probability of the set  $S_k$ , which can then be written as

$$\mathcal{P}(S_k) = \mathcal{P} \left( \bigcap_{i \in \chi(S_k)} (\text{SNR}_{s,i} \geq \beta) \bigcap_{j \in \chi^c(S_k)} (\text{SNR}_{s,j} \leq \beta) \right). \quad (18.125)$$

The average outage probability expression can thus be given by

$$\mathcal{P}_{\text{OC}} = \sum_{k=0}^{2^N-1} [1 - f(r_{s,d}, P_s^c)] \prod_{j \in \chi(S_k)} [1 - f(r_{j,d}, P_j)] f(r_{s,j}, P_s^c) \quad (18.126)$$

$$\cdot \prod_{j \in \chi^c(S_k)} [1 - f(r_{s,j}, P_s^c)]. \quad (18.127)$$

where  $P_j$ ,  $j \in \{1, 2, \dots, N\}$  is the power allocated to the  $j$ th relay.

Next we compute the average total consumed power for the  $N$ -relays scenario. First we condition on some relays' status vector  $\chi(S_k)$ :

$$E[P_{\text{tot}}^c] = E[E[P_{\text{tot}}^c | \chi(S_k)]] = \sum_{k=0}^{2^N-1} P(\chi(S_k)) E[P_{\text{tot}}^c | \chi(S_k)]. \quad (18.128)$$

For a given  $\chi(S_k)$ , we can further condition on whether the source gets the packet through from the first trial or not. This event happens with probability  $f(r_{s,d}, P_s^c)$ , and the consumed power in this case is given by

$$P_{\text{tot}}^{c,1} = P_s^c + (N+1)P_r + P_c. \quad (18.129)$$

The complementary event that the source failed to transmit its packet from the direct transmission phase happens with probability  $1 - f(r_{s,d}, P_s^c)$ , and this event can be further divided into two mutually exclusive events. The first is when the first  $|\chi(S_k)|-1$  relays from the set  $\chi(S_k)$  fails to forward the packet, and this happens with probability  $\prod_{i=1}^{|\chi(S_k)|-1} [1 - f(r_{i,d}, P_i)]$ , and the corresponding consumed power is given by

$$P_{\text{tot}}^{c,2} = P_s^c + (N+1+|\chi(S_k)|)P_r + (|\chi(S_k)|+1)P_c + \sum_{n=1}^{|\chi(S_k)|} P_{\chi(S_k)(n)}. \quad (18.130)$$

And the second is when one of the intermediate relays in the set  $\chi(S_k)$  successfully forwards the packet, and this happens with probability  $\prod_{m=1}^{j-1} [1 - f(r_{m,d}, P_m)] f(r_{j,d}, P_j)$  if this intermediate relay was relay number  $j$ , and the corresponding power is given by

$$P_{\text{tot}}^{c,3,j} = P_s^c + (N+1+j)P_r + (1+j)P_c + \sum_{i=1}^j P_{\chi(S_k)(i)}. \quad (18.131)$$

From (18.128) to (18.131) the average total consumed power can be given by

$$E[P_{\text{tot}}^c] = \sum_{k=0}^{2^N-1} P(\chi(S_k)) \times \left\{ f(r_{s,d}, P_s^c) P_{\text{tot}}^{c,1} + (1 - f(r_{s,d}, P_s^c)) \right. \quad (18.132)$$

$$\times \left[ \prod_{i=1}^{|\chi(S_k)|-1} (1 - f(r_{i,d}, P_i)) P_{\text{tot}}^{c,2} \right. \quad (18.133)$$

$$\left. + \sum_{j=1}^{|\chi(S_k)|-1} \prod_{m=1}^{j-1} (1 - f(r_{m,d}, P_m)) f(r_{j,d}, P_j) P_{\text{tot}}^{c,3,j} \right\}. \quad (18.134)$$

The optimization problem can then be written as

$$\min_{\mathbf{P}} P_{\text{tot}}^C(\mathbf{P}), \quad (18.135)$$

$$\text{s.t. } \mathcal{P}_{\text{OC}}(\mathbf{P}) \leq \mathcal{P}_{\text{out}}^*, \quad (18.136)$$

where  $\mathbf{P} = [P_s^c, P_1, P_2, \dots, P_N]^T$ .

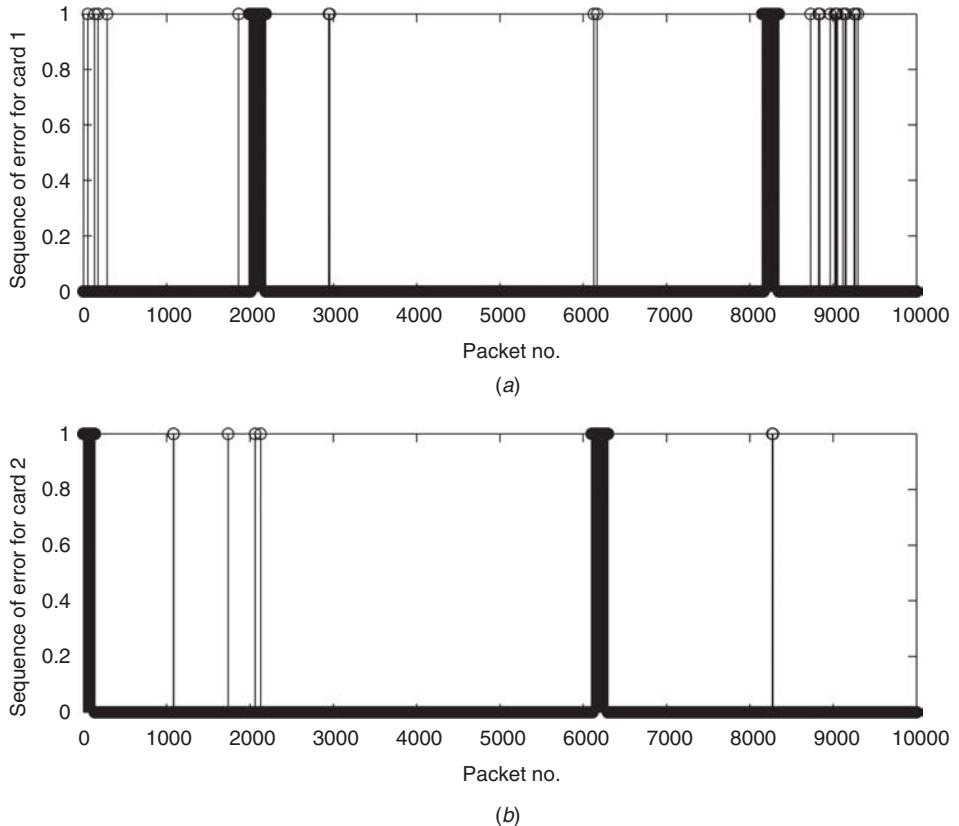
## 18.5 EXPERIMENTAL RESULTS

In the system model, it is assumed that the channel independence is between the following links: the source–relay link, the source–destination link, and the relay–destination link. Moreover, a strong motivation for applying cooperative transmission instead of ARQ in the time domain is the assumption of high temporal correlation, which results in delay and requires performing interleaving at the transmitter side. In this section, we show the results for a set of experiments to justify these two fundamental assumptions.

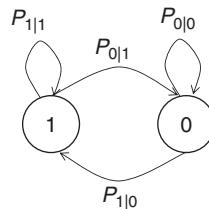
The experiments are set up as follows. We have three wireless nodes in the experiments, one of them acts as the sender and the other two act as receivers. Each wireless node is a computer equipped with a IEEE 802.11g wireless card; specifically, we utilized three LINKSYS wireless-G USB network adaptors. The sender’s role is to broadcast data packets with a constant rate, while the two receivers’ role is to decode the packets and record which packet is erroneous. The traffic rate is 100 packets per second, and the size of each packet is 554 bytes (including packet headers). The two receivers are placed together, with the distance between them being 20 cm. The distance between the transmitter and the receiver is around 5 m. The experiments have been mainly conducted in office environments. The experiments results, which are illustrated next, have revealed two important observations: the channels exhibit strong time correlation for each receiver, while there is negligible dependence between the two receivers. Figure 18.8 illustrates one instantiation of the experiments. Figure 18.8a illustrates the results obtained at the first receiver and Figure 18.8b is for the second receiver.

For each figure, the horizontal axis denotes the sequence number of the first 100,000 packets, and the vertical axis denotes whether a packet is erroneous or not. First, from these results we can see that packet errors exhibit strong correlation in time. For example, for the first receiver, most erroneous packets cluster at around the 22nd second and around the 83rd second. Similar observations also hold for the second receiver. If we take a further look at the results, we can see that in this set of experiments the duration for the cluster is around 2 s. To help better understand the time correlation of erroneous packets, we have also used a two-state Markov chain to model the channel, as illustrated in Figure 18.9. In this model “1” denotes that the packet is correct, and “0” denotes that the packet is erroneous.  $P_{i|j}$  denotes the transition probability from state  $i$  to state  $j$ , that is, the probability to reach state  $j$  given the previous state is  $i$ . The following transition probabilities have been obtained after using the experimental results to train the model:  $P_{1|0} = 0.03$ ,  $P_{1|1} = 0.999$ ,  $P_{0|0} = 0.97$ , and  $P_{0|1} = 0.001$ . These results also indicate strong time correlation. For example, given the current received packet is erroneous, the probability that the next packet is also erroneous is around  $P_{0|0} = 0.97$ .

Now we take a comparative look at the results obtained at the two receivers. From these results we can see that although there exists slight correlation in packet errors between the two receivers, it is almost negligible. To provide more concrete evidence of independence, we have estimated the correlation between the two receivers using the obtained experiment results. Specifically, we have measured the correlation coefficient between the received sequences at the two receivers, and we found that the correlation coefficient is almost 0, which indicates a strong spatial independence between the two receivers.



**Figure 18.8** Sequence of packet errors at the two utilized wireless cards.



**Figure 18.9** Modeling the channel by a two (on–off) state Markov chain to study the time correlation.

### 18.5.1 Numerical Examples

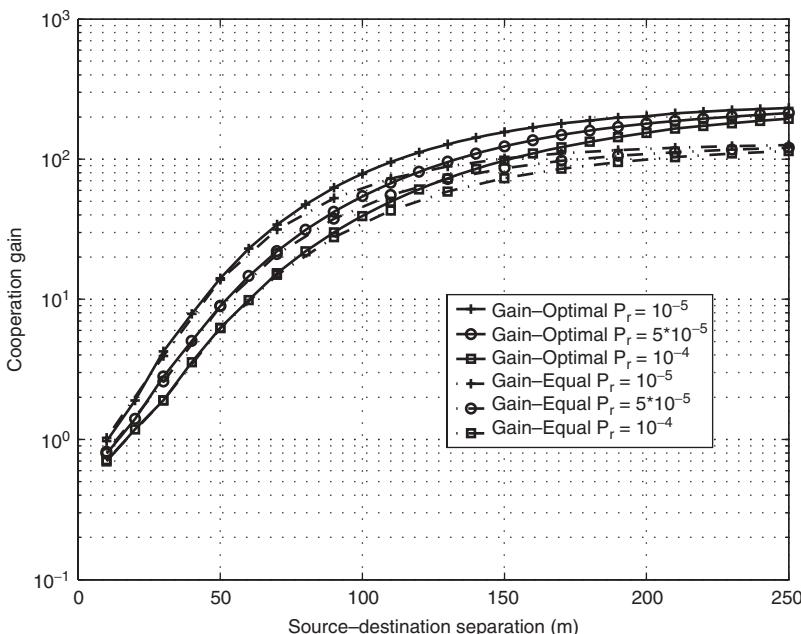
As discussed in the previous sections, there are different system parameters that can control whether we can gain from cooperation or not, among which are the received power consumption, the processing power, the SNR threshold, the power amplifier loss, and the relative distances between the source, relay, and destination.

In order to understand the effect of each of these parameters, we are going to study the performance of cooperative and direct transmission when varying one of these

parameters and fixing the rest. This is described in more detail in the following. In all of the numerical examples, the aforementioned parameters take the following values when considered fixed:  $\alpha = 0.3$ ,  $\beta = 10$ ,  $N_o = 10^{-3}$ ,  $P_c = 10^{-4}$  W,  $P_r = 5 \times 10^{-5}$ , QoS =  $\mathcal{P}_{\text{out}}^* = 10^{-4}$ . We define the cooperation gain as the ratio between the total power required for direct transmission to achieve a certain QoS, and the total power required by cooperation to achieve the same QoS.

**Example 18.2** We study the effect of varying the receive power  $P_r$  as depicted in Figure 18.10. We plot the cooperation gain versus the distance between the source and the destination for different values of receive power  $P_r = 10^{-4}, 5 \times 10^{-5}, 10^{-5}$  W. At source–destination distances below 20 m, the results reveal that direct transmission is more energy efficient than cooperation, that is, the overhead in receive and processing power due to cooperation outweighs its gains in saving the transmit power. For  $r_{s,d} > 20$  m, the cooperation gain starts increasing as the transmit power starts constituting a significant portion of the total consumed power. This ratio increases until the transmit power is the dominant part of the total consumed power and hence the cooperation gain starts to saturate.

In the plotted curves, the solid lines denote the cooperation gain when utilizing optimal power allocation at the source and the relay, while the dotted curves denote the gain for equal power allocation. For  $r_{s,d} \leq 100$  m, both optimal power allocation and equal power allocation almost yield the same cooperation gain. For larger distances, however, a gap starts to appear between optimal and equal power allocation. The rationale behind these observations is that at small distances the transmit power is a small percentage of the total consumed power and hence optimal and equal power



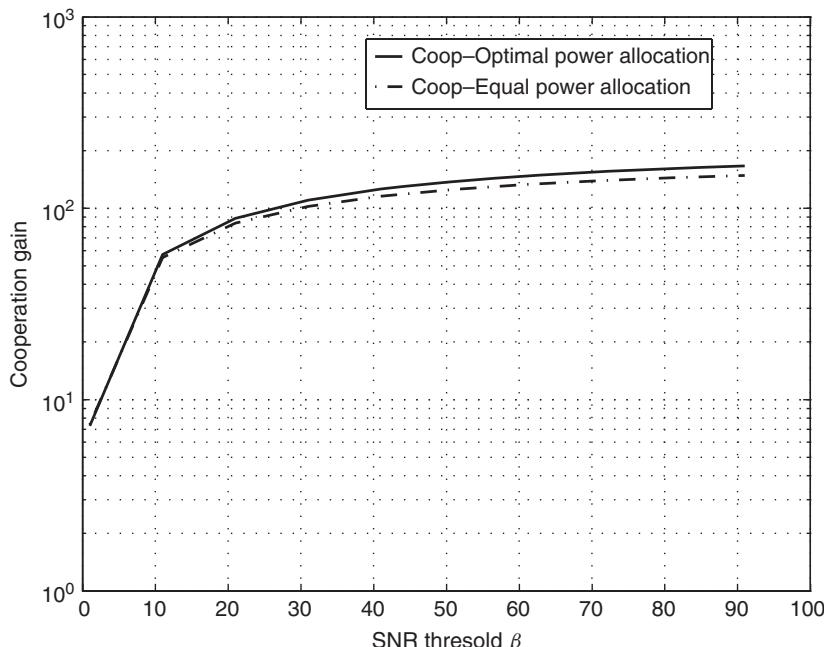
**Figure 18.10** Cooperation gain versus the source–destination distance for different values of received power consumption.

allocation almost have the same behavior, while at larger distances, transmit power plays a more important role and hence a gap starts to appear.

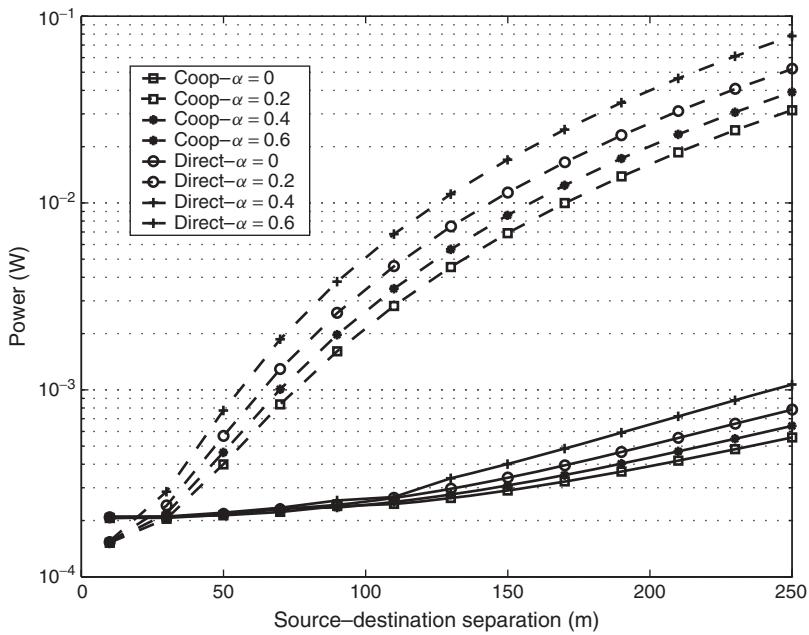
**Example 18.3** In this example, we study the effect of changing the SNR threshold  $\beta$  as depicted in Figure 18.11. The distance between source and destination  $r_{s,d}$  is fixed to 100 m. It is clear that the cooperation gain increases with increasing  $\beta$ , and that for the considered values of the system parameters, equal power allocation provides almost the same gains as optimal power allocation. In Figure 18.12 we study the effect of the power amplifier loss  $\alpha$ .

In this case, we plot the total consumed power for cooperation and direct transmission versus distance for different values of  $\alpha$ . Again below 20 m separation between the source and the destination, direct transmission provides better performance over cooperation. It can also be seen from the plotted curves that the required power for direct transmission is more sensitive to variations in  $\alpha$  than the power required for cooperation. The reason is that the transmit power constitutes a larger portion in the total consumed power in direct transmission than in cooperation, and hence the effect of  $\alpha$  is more significant. The QoS, measured by the required outage probability, has similar behavior and the results are depicted in Figure 18.13.

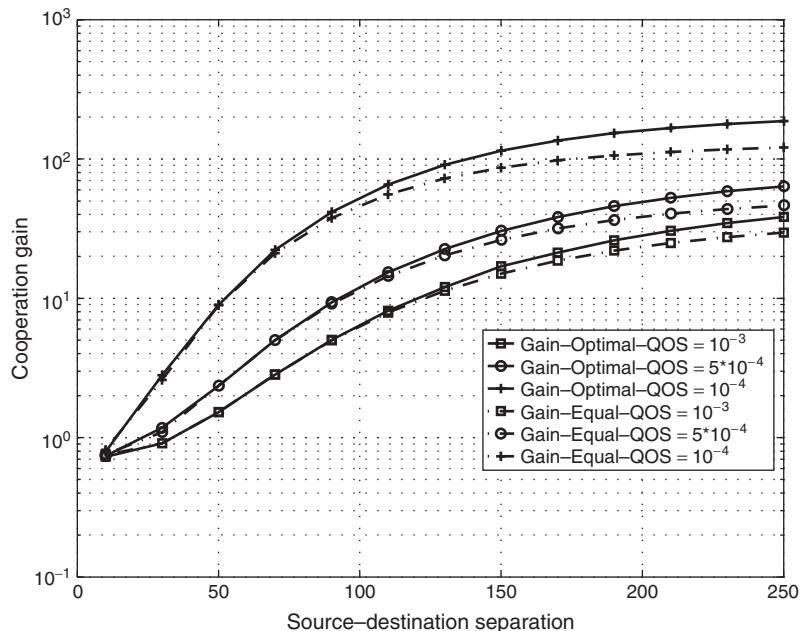
**Example 18.4** In this example, we study the effect of varying the relay location. We consider three different positions for the relay, close to the source, in the middle between the source and the destination, and close to the destination. In particular, the relay position is taken equal to  $(r_{s,l} = 0.2r_{s,d}, r_{l,d} = 0.8r_{s,d})$ ,  $(r_{s,l} = 0.5r_{s,d}, r_{l,d} = 0.5r_{s,d})$ , and  $(r_{s,l} = 0.8r_{s,d}, r_{l,d} = 0.2r_{s,d})$ .



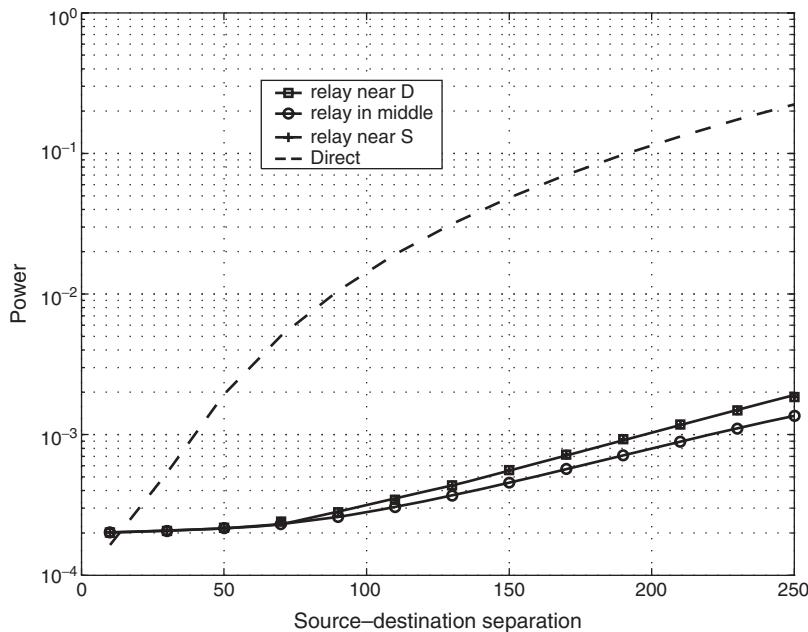
**Figure 18.11** Cooperation gain versus the SNR threshold  $\beta$ .



**Figure 18.12** Optimal power consumption for both cooperation and direct transmission scenarios for different values of power amplifier loss  $\alpha$ .



**Figure 18.13** Cooperation gain versus the source–destination distance for different values of QoS.



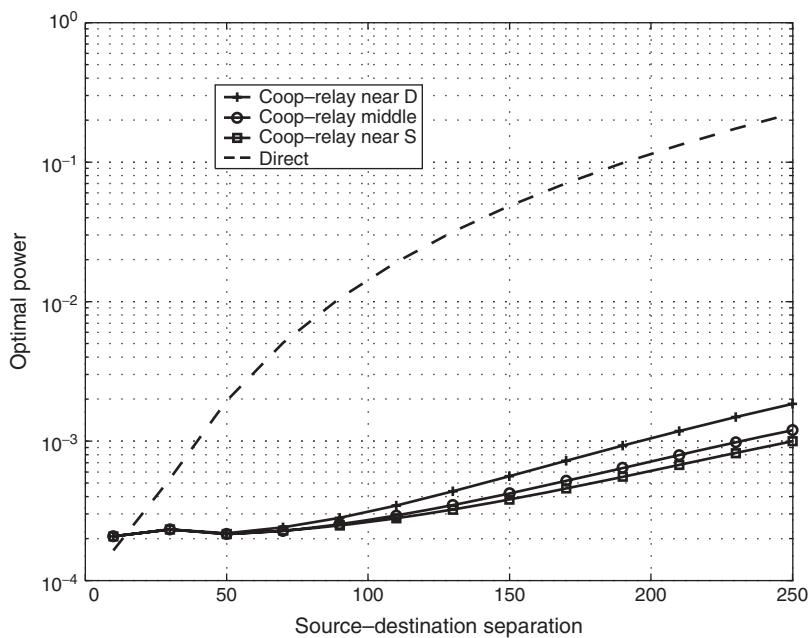
**Figure 18.14** Optimal consumed power versus distance for different relay locations for equal power allocation at source and relay.

Figures 18.14 and 18.15 depict the power required for cooperation and direct transmission versus  $r_{s,d}$  for equal power and optimal power allocation, respectively. In the equal power allocation scenario, the relay in the middle gives the best results, and the other two scenarios, relay close to source and relay close to destination, provide the same performance.

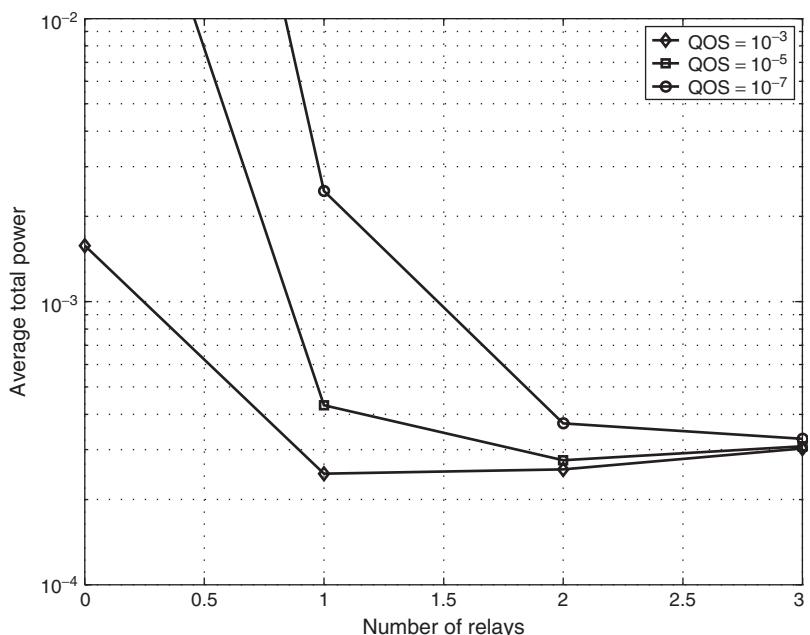
This can be expected because for the equal power allocation scenario the problem becomes symmetric in the source–relay and relay–destination distances. For the optimal power allocation scenario depicted in Figure 18.15, the problem is no more symmetric because different power allocation is allowed at the source and relay. In this case, numerical results show that the closer the relay is to the source the better the performance. The intuition behind this is that when the relay is closer to the source, the source–relay channel is very good and almost error-free.

From both figures, it is also clear that for small source–destination separation  $r_{s,d}$ , equal and optimal power allocation almost provide the same cooperation gain while for larger  $r_{s,d}$  optimal power allocation provides more gain. Another important observation is that at small distances below 100 m, the location of the relay does not affect the performance much. This makes the algorithms required to select a relay in cooperative communications simpler to implement for source–destination separations in this range. Finally, the threshold behavior below 20 m still appears where direct transmission becomes more energy efficient.

**Example 18.5** Figure 18.16 depicts the multiple relays scenario for different values of outage probability  $\mathcal{P}_{\text{out}}^*$ . The results are depicted for a source–destination distance of 100 m, and for  $N = 0, 1, 2, 3$  relays, where  $N = 0$  refers to direct transmission.



**Figure 18.15** Optimal consumed power versus distance for different relay locations for optimal power allocation at source and relay.



**Figure 18.16** Optimal consumed power versus number of relays for different values of required outage probability.

As shown in Figure 18.16, for small values of required outage probability, one relay is more energy efficient than two or three relays. As we increase the required QoS, reflected by  $\mathcal{P}_{\text{out}}^*$ , the optimal number of relays increases. Hence, our analytical framework can also provide guidelines to determining the optimal number of relays under any given scenario.

## 18.6 CONCLUSIONS

Cooperative communications is a new communication paradigm that generalizes MIMO communications to much broader applications. In this new paradigm, the terminals dispersed in a wireless network cell can be thought of as distributed antennas. Via cooperation among these nodes, MIMO-like gains can be achieved by increasing the diversity gains of the system. Different protocols were described to implement cooperation. We described the performance of these algorithms through calculating outage capacity and characterizing diversity gains. The performance of adaptive relaying techniques in general outperforms that of fixed relaying techniques because of the extra information utilized in implementing the protocols, for example, knowledge of the received SNR in selective relaying and the feedback from the destination in incremental relaying. On the other hand, fixed relaying techniques are simpler to implement compared to fixed relaying techniques due to the extra overhead needed to implement the former protocols. Therefore, it is ultimately a trade-off between performance and complexity that the system designer must decide on. The SER performance for the single-relay scenario is considered for both DF and AF relaying. Exact and approximate expressions for the SER for general  $M$ -PSK and  $M$ -QAM modulation are derived. Using the approximate expressions, optimal power allocation at the source and relay node is derived. It is shown that equal power allocation is in general suboptimal.

The energy efficiency of cooperation in wireless networks is studied under a practical setting where the extra overhead of cooperation is taken into account. The approach taken was to formulate a constrained optimization problem to minimize the total consumed power under a given QoS requirement. The numerical results reveal that for short distance separations between the source and the destination, for example, below a threshold of 20 m, the overhead of cooperation outweighs its gains and direct transmission is more efficient. Above that threshold, cooperation gains can be achieved. It was also shown that simple equal power allocation at the source and the relay achieves almost the same gains as optimal power allocation at these two nodes for distances below 100 m, for the specific parameters used.

Furthermore, choosing the optimal relay location for cooperation plays an important role when the source–destination separation exceeds 100 m, and the best relay location depends on the power allocation scheme, whether optimal or equal. The results can also be used to provide guidelines in determining the optimal number of relays for any given communication setup.

## REFERENCES

1. J. N. Laneman, D. N. C. Tse, and G. W. Wornell, “Cooperative diversity in wireless networks: Efficient protocols and outage behavior,” *IEEE Trans. Inform. Theory*, vol. 50, pp. 3062–3080, Dec. 2004.

2. A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity—Part I: System description," *IEEE Trans. Commun.*, vol. 51, pp. 1927–1938, Nov. 2003.
3. A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity—Part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, vol. 51, pp. 1939–1948, Nov. 2003.
4. T. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inform. Theory*, vol. 25, no. 5, pp. 572–584, Sept. 1979.
5. E. C. van der Meulen, "Three-terminal communication channels," *Adv. Appl. Prob.*, vol. 3, pp. 120–154, 1971.
6. K. J. R. Liu, A. K. Sadek, W. Su, and A. Kwasinski, *Cooperative Communications and Networking*, Cambridge University Press, 2008.
7. A. K. Sadek, W. Su, and K. J. R. Liu, "Multi-node cooperative communications in wireless networks," *IEEE Trans. Signal Process.*, vol. 55, no. 1, pp. 341–355, Jan. 2007.
8. K. Seddik, A. K. Sadek, W. Su, and K. J. R. Liu, "Outage analysis and optimal power allocation for multi-node amplify-and-forward relay networks," *IEEE Signal Process. Lett.*, vol. 14, pp. 377–380, June 2007.
9. M. K. Simon and M.-S. Alouini, "A unified approach to the performance analysis of digital communication over generalized fading channels," *Proc. IEEE*, vol. 86, no. 9, pp. 1860–1877, Sept. 1998.
10. J. G. Proakis, *Digital Communications*, 4th ed., New York: McGraw-Hill, 2001.
11. J. W. Craig, "A new, simple and exact result for calculating the probability of error for two-dimensional signal constellations," in *Proc. IEEE MILCOM*, Boston, MA, 1991, pp. 25.5.1–25.5.5.
12. W. Su, A. K. Sadek, and K. J. R. Liu, "Cooperative communication protocols in wireless networks: Performance analysis and optimum power allocation," *Wireless Personal Commun. (Springer)*, vol. 44, no. 2, pp. 181–217, Jan. 2008.
13. M. O. Hasna and M.-S. Alouini, "Performance analysis of two-hop relayed transmissions over Rayleigh fading channels," in *Proc. IEEE Vehicular Technology Conf. (VTC)*, Vol. 4, Sept. 2002, pp. 1992–1996.
14. I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*, New York: Academic, 1980.
15. A. K. Sadek, W. Yu, and K. J. R. Liu, "When does cooperation have better performance in sensor networks?" *Proc. IEEE SECON*, Reston, VA, 2006.



## CHAPTER 19

---

# Distributed Source Coding

Zixiang Xiong<sup>1</sup>, Angelos D. Liveris, and Yang Yang

<sup>1</sup>Department of ECE, Texas A&M University, College Station, Texas

### 19.1 INTRODUCTION

In many emerging applications (e.g., distributed sensor networks), multiple correlated sources need to be separately compressed at distributed terminals before being transmitted to a central unit. Due to complexity and/or power constraints, the transmitters are often not allowed to communicate with each other. This gives rise to the problem of distributed source coding (DSC).

The foundation of DSC was laid by Slepian and Wolf in their 1973 study [1], which considered separate lossless compression of two correlated sources and showed the surprising result that separate encoding (with joint decoding) suffers no rate loss when compared to joint encoding. This seminal work (on Slepian–Wolf coding [1]) was subsequently extended to other DSC scenarios.

In 1976, Wyner and Ziv [2] extended one special case of Slepian–Wolf (SW) coding, namely, lossless source coding with decoder side information, to *lossy* source coding with decoder side information. Unlike SW coding, there is in general a rate loss associated with Wyner–Ziv (WZ) coding [2] when compared to lossy source coding with side information also available at the encoder. An exception occurs with quadratic Gaussian WZ coding when the source and side information are jointly Gaussian and the distortion measure is the mean-squared error (MSE).

In 1977, Berger [3] introduced the general problem of multiterminal (MT) source coding by considering separate *lossy* source coding of two (or more) sources.<sup>1</sup> Two classes of MT source coding problems have been studied in the literature. Berger [3] and Tung [4] treated the case in which each encoder observes *directly* its source; later, Yamamoto and Itoh [5] and Flynn and Gray [6] focused on another scenario where each encoder cannot observe directly the source that is to be reconstructed at the decoder but is rather provided only with a noisy version. These two classes are distinguished as the *direct* and *indirect* (or *remote*) MT source coding problem, respectively. Note that in the latter case, often referred to as the CEO problem [7, 8], a single source is

<sup>1</sup>One can loosely think of MT source coding as the lossy version of SW coding.

to be reconstructed at the decoder. Unlike SW coding, there is in general a rate loss associated with direct MT source coding. Furthermore, unlike WZC coding, even in the quadratic Gaussian setup, direct MT source coding suffers rate loss (when compared with joint encoding). However, the supremum sum-rate loss with quadratic Gaussian two-terminal source coding is only  $\frac{1}{2} \log_2 \frac{5}{4} = 0.161$  b/s [9].

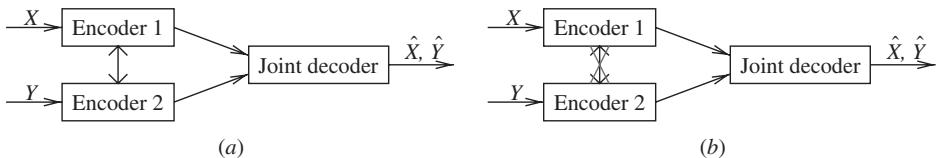
Although research on DSC has been ongoing for more than 30 years, the theory of DSC—as part of network information theory—is still incomplete. Driven by applications of DSC, especially in distributed sensor networks, renewed interests and heated research efforts in the past 9 years have led to progresses in both theory and code designs. For example, in 2005 Wagner et al. [10] proved tightness of the Berger–Tung sum-rate bound, leading to a complete characterization of the rate region of quadratic Gaussian MT source coding of two sources.<sup>2</sup> In addition, by exploiting recent advances in channel coding (e.g., turbo [11] and LDPC codes [12, 13]), practical code designs based on powerful channel codes [14–18] have been recently devised to approach the theoretical limit of DSC. These exciting new developments have made applications such as distributed video coding [19] viable. This chapter reviews the theory, code designs, and applications of DSC.

## 19.2 THEORETICAL BACKGROUND

### 19.2.1 Slepian–Wolf Coding

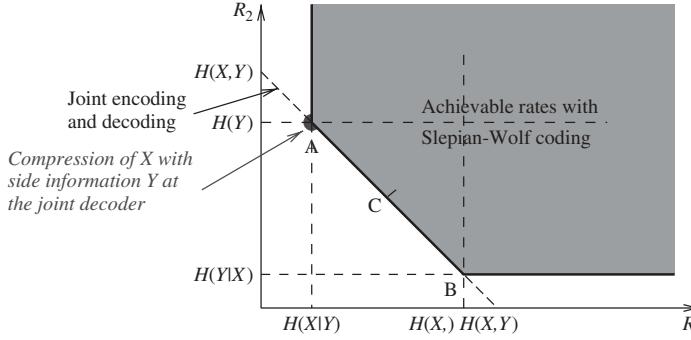
Let  $\{(X_i, Y_i)\}_{i=1}^\infty$  be a sequence of independent and identically distributed (i.i.d.) drawings of a pair of correlated discrete random variables  $X$  and  $Y$ . For lossless compression with  $\hat{X} = X$  and  $\hat{Y} = Y$  after decompression, we know from Shannon’s source coding theory [20] that a rate given by the joint entropy  $H(X, Y)$  of  $X$  and  $Y$  is sufficient if we are encoding them together (see Fig. 19.1a). For example, we can first compress  $Y$  into  $H(Y)$  bits per sample, and based on the complete knowledge of  $Y$  at both the encoder and the decoder, we then compress  $X$  into  $H(X|Y)$  bits per sample. But what if  $X$  and  $Y$  must be separately encoded for some user to reconstruct both of them?

One simple way is to do separate coding with rate  $R = H(X) + H(Y)$ , which is greater than  $H(X, Y)$  when  $X$  and  $Y$  are correlated. In a landmark study [1], Slepian and Wolf showed that  $R = H(X, Y)$  is sufficient even for separate encoding of correlated



**Figure 19.1** (a) Joint encoding of  $X$  and  $Y$ . Encoders collaborate and a rate  $H(X, Y)$  is sufficient. (b) Distributed/separate encoding of  $X$  and  $Y$ . Encoders do not collaborate. The SW theorem says that a rate  $H(X, Y)$  is also sufficient provided that decoding of  $X$  and  $Y$  is done jointly.

<sup>2</sup>However, the rate region for the quadratic Gaussian MT source coding problem with more than two terminals is still unknown.



**Figure 19.2** The SW rate region for two sources.

sources (see Fig. 19.1b)! Specifically, the SW theorem says that the achievable region of DSC for discrete sources  $X$  and  $Y$  is given by

$$R_1 \geq H(X|Y), \quad R_2 \geq H(Y|X), \quad R_1 + R_2 \geq H(X, Y), \quad (19.1)$$

which is shown in Figure 19.2. This result is quite surprising since it means that there is no loss of coding efficiency with separate encoding when compared to joint encoding. The proof of the SW theorem is based on random *binning*. Binning is a key concept in DSC and refers to partitioning the space of all possible outcomes of a random source into disjoint subsets or *bins*. The achievability of SW coding was generalized by Cover [21] to arbitrary ergodic processes, countably infinite alphabets, and arbitrary number of correlated sources.

### 19.2.2 Wyner–Ziv Coding

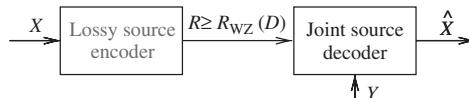
Wyner–Ziv coding concerns the problem of how many bits are needed to encode  $X$  under the constraint that the average distortion between  $X$  and the coded version  $\hat{X}$  is  $E\{d(X, \hat{X})\} \leq D$ , assuming the side information  $Y$  is available at the decoder but not at the encoder. This problem, schematically depicted in Figure 19.3, is one instance of DSC. It generalizes a special setup of SW coding in that coding of discrete  $X$  is with respect to a fidelity criterion rather than lossless.

For both discrete and continuous source  $X$  and general distortion measure  $d(\cdot, \cdot)$ , Wyner and Ziv [2, 22] gave the rate distortion function  $R_{WZ}(D)$  for this problem as

$$R_{WZ}(D) = \inf_{\{Y \rightarrow X \rightarrow Z; \hat{X} \in \mathcal{X}: E\{d(X, \hat{X}|Z, Y)\} \leq D\}} I(X; Z|Y) \quad (19.2)$$

where  $Y \rightarrow X \rightarrow Z$  means that  $X$ ,  $Y$  and the auxiliary random variable  $Z$  form a Markov chain (in this order). According to [2],

$$R_{WZ}(D) \geq R_{X|Y}(D) = \inf_{\{\hat{X} \in \mathcal{X}: E\{d(X, \hat{X}|Y)\} \leq D\}} I(X; \hat{X}|Y), \quad (19.3)$$



**Figure 19.3** WZ coding or lossy source coding with side information.

where  $R_{X|Y}(D)$  is the classic rate distortion function of coding  $X$  with side information  $Y$  available at both the encoder and the decoder.

Unlike SW coding, it is seen from (19.3) that generally there is a rate loss with WZ coding when compared to lossy coding of source  $X$  with the side information  $Y$  available at both the encoder and the decoder. Zamir quantified this loss in [23], showing a  $<0.22$  b/s loss for binary sources with Hamming distance and a  $<0.5$  b/s loss for continuous sources with MSE distortion. For example, in the binary symmetric case,  $X$  and  $Y$  are binary symmetric sources, the correlation between them is modeled as a binary symmetric channel (BSC) with crossover probability  $p$  ( $0 < p < 0.5$ ), and the distortion measure is the Hamming distance. The WZ rate distortion function for this case is [2]

$$R_{WZ}(D) = \text{l.c.e.}\{H(p * D) - H(D), (p, 0)\}, \quad 0 \leq D \leq p, \quad (19.4)$$

the lower convex envelope (l.c.e.) is  $H(p * D) - H(D)$  and the point  $(D = p, R = 0)$ , where  $p * D = (1 - p)D + (1 - D)p$ . In contrast, we know from [20] that

$$R_{X|Y}(D) = \begin{cases} H(p) - H(D), & 0 \leq D \leq \min\{p, 1 - p\}, \\ 0, & D > \min\{p, 1 - p\}. \end{cases} \quad (19.5)$$

In Figure 19.4, both  $R_{WZ}(D)$  and  $R_{X|Y}(D)$  are plotted for  $p = 0.27$ . WZ coding obviously suffers rate loss in this binary symmetric case. Note that, when  $D = 0$ , the WZ problem degenerates to the SW problem with  $R_{WZ}(0) = R_{X|Y}(0) = H(X|Y) = H(p)$ .

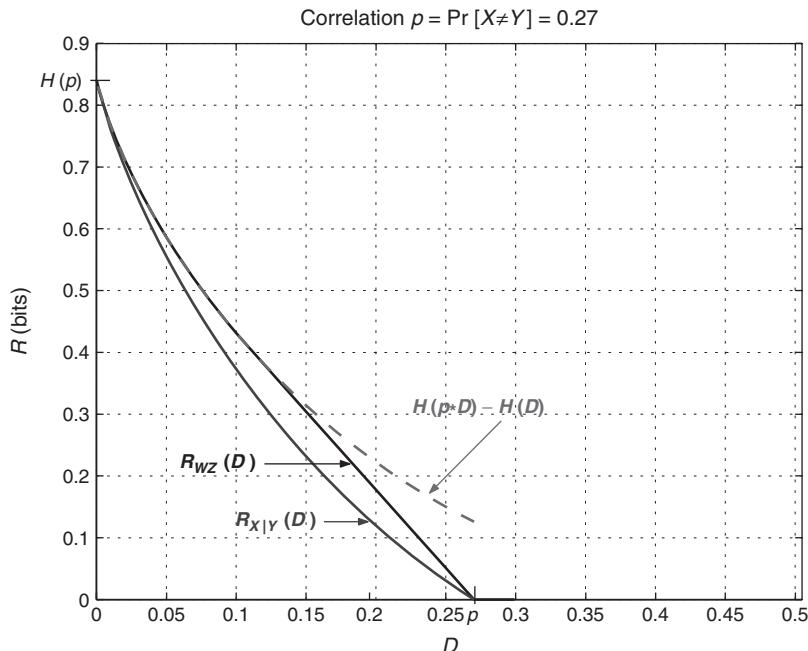


Figure 19.4  $R_{WZ}(D)$  and  $R_{X|Y}(D)$  for the binary symmetric case with  $p = 0.27$ .

However, in the quadratic Gaussian case,  $X$  and  $Y$  are stationary Gaussian memoryless sources and the distortion measure is MSE; let the covariance matrix of  $X$  and  $Y$  be

$$\Lambda = \begin{bmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{bmatrix}$$

with  $|\rho| < 1$ , then [2, 22]

$$R_{WZ}(D) = R_{X|Y}(D) = \frac{1}{2} \log^+ \left[ \frac{\sigma_X^2(1 - \rho^2)}{D} \right], \quad (19.6)$$

where  $\log^+ x = \max\{\log_2 x, 0\}$ . Thus we have an exception with no rate loss<sup>3</sup> in quadratic Gaussian WZ coding.

### 19.2.3 Multiterminal Source Coding

**19.2.3.1 Direct MT Source Coding** The direct MT source coding setup is depicted in Figure 19.5. The encoders observe sources  $Y_1$  and  $Y_2$ , which take values in  $\mathcal{Y}_1 \times \mathcal{Y}_2$ , and are drawn i.i.d. from the joint probability density function (pdf)  $f_{Y_1, Y_2}(y_1, y_2)$ . Each sequence of  $n$  source samples is grouped as a *source block*  $Y_1^n$  and  $Y_2^n$ , where  $Y_1^n = \{Y_{1,i}\}_1^n$ ,  $Y_2^n = \{Y_{2,i}\}_1^n$ . Two encoder functions

$$\phi_1: \mathcal{Y}_1^n \rightarrow \{1, 2, \dots, 2^{nR_1}\} \quad \text{and} \quad \phi_2: \mathcal{Y}_2^n \rightarrow \{1, 2, \dots, 2^{nR_2}\} \quad (19.7)$$

separately compress  $Y_1^n$  and  $Y_2^n$  to  $W_1$  and  $W_2$  at rates  $R_1$  and  $R_2$ , respectively. A decoder function

$$\varphi: \{1, 2, \dots, 2^{nR_1}\} \times \{1, 2, \dots, 2^{nR_2}\} \rightarrow \mathcal{Y}_1^n \times \mathcal{Y}_2^n \quad (19.8)$$

reconstructs the source block as  $\{\hat{Y}_1^n, \hat{Y}_2^n\}$  based on the received  $W_1$  and  $W_2$ .

For a distortion pair  $(D_1, D_2)$  and a given distortion measure  $d(\cdot, \cdot)$ , a rate pair  $(R_1, R_2)$  is *achievable* if for any  $\epsilon > 0$ , there exists a large enough  $n$  and a triple  $(\phi_1, \phi_2, \varphi)$  such that the distortion constraints

$$\frac{1}{n} \sum_{i=1}^n E[d(Y_{1,i}, \hat{Y}_{1,i})] \leq D_1 + \epsilon \quad \text{and} \quad \frac{1}{n} \sum_{i=1}^n E[d(Y_{2,i}, \hat{Y}_{2,i})] \leq D_2 + \epsilon \quad (19.9)$$

are satisfied. The *achievable rate region*  $\mathcal{R}^*(D_1, D_2)$  is the convex hull of the set of all achievable rate pairs  $(R_1, R_2)$ .

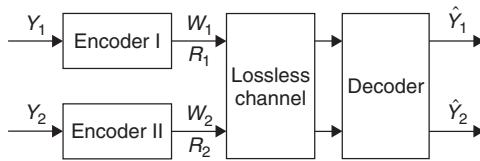


Figure 19.5 Two-terminal direct MT source coding.

<sup>3</sup>It was only shown in [22] that WZ coding of  $X$  suffers no rate loss when  $X$  and  $Y$  are zero mean and jointly Gaussian with MSE distortion. Pradhan et al. [24] recently extended the no rate loss condition for WZ coding to  $X = Y + Z$ , where  $Z$  is independently Gaussian but  $Y$  (hence  $X$ ) could follow more general distributions.

The exact achievable rate region for the direct MT source coding problem is still unknown. Only inner and outer rate regions are provided. For auxiliary random variables  $Z_1$  and  $Z_2$  let

$$\begin{aligned}\tilde{\mathcal{R}}(Z_1, Z_2) = \{(R_1, R_2) : R_i \geq I(Y_1 Y_2; Z_i | Z_j), i, j = 1, 2, i \neq j, \\ R_1 + R_2 \geq I(Y_1 Y_2; Z_1 Z_2)\},\end{aligned}\quad (19.10)$$

then the inner rate region is given by [3–5]

$$\begin{aligned}\hat{\mathcal{R}}(D_1, D_2) = \text{conv}\{\tilde{\mathcal{R}}(Z_1, Z_2) : Z_1 \rightarrow Y_1 \rightarrow Y_2 \rightarrow Z_2, \\ \exists \varphi(Z_1^n, Z_2^n) \text{ satisfying (19.9)}\},\end{aligned}\quad (19.11)$$

while the outer rate region is [3–5]

$$\check{\mathcal{R}}(D_1, D_2) = \text{conv}\{\tilde{\mathcal{R}}(Z_1, Z_2) : Z_1 \rightarrow Y_1 \rightarrow Y_2, Z_2 \rightarrow Y_2 \rightarrow Y_1, \\ \exists \varphi(Z_1^n, Z_2^n) \text{ satisfying (19.9)}\}. \quad (19.12)$$

Let  $\partial\hat{\mathcal{R}}(D_1, D_2)$  be the set of all boundary points of the rate region  $\hat{\mathcal{R}}(D_1, D_2)$ ; likewise, let  $\partial\check{\mathcal{R}}(D_1, D_2)$  be the set of all boundary points of the rate region  $\check{\mathcal{R}}(D_1, D_2)$ . We call  $\partial\hat{\mathcal{R}}(D_1, D_2)$  the *inner bound*, and  $\partial\check{\mathcal{R}}(D_1, D_2)$  the *outer bound*.

For the direct Gaussian MT source coding problem with MSE distortion measure  $d(\cdot, \cdot)$ , where the sources  $(Y_1, Y_2)$  are jointly Gaussian random variables with variances  $(\sigma_{y_1}^2, \sigma_{y_2}^2)$  and correlation coefficient  $\rho = E[Y_1 Y_2]/\sigma_{y_1} \sigma_{y_2}$ , the Berger–Tung (BT) inner rate region (19.11) becomes [25]

$$\hat{\mathcal{R}}^{\text{BT}}(D_1, D_2) = \hat{\mathcal{R}}_1^{\text{BT}}(D_1, D_2) \cap \hat{\mathcal{R}}_2^{\text{BT}}(D_1, D_2) \cap \hat{\mathcal{R}}_{12}^{\text{BT}}(D_1, D_2), \quad (19.13)$$

where

$$\begin{aligned}\hat{\mathcal{R}}_i^{\text{BT}}(D_1, D_2) = \left\{ (R_1, R_2) : R_i \geq \frac{1}{2} \log^+ \left[ (1 - \rho^2 + \rho^2 2^{-2R_j}) \frac{\sigma_{y_i}^2}{D_i} \right] \right\}, \\ i, j = 1, 2, i \neq j,\end{aligned}\quad (19.14)$$

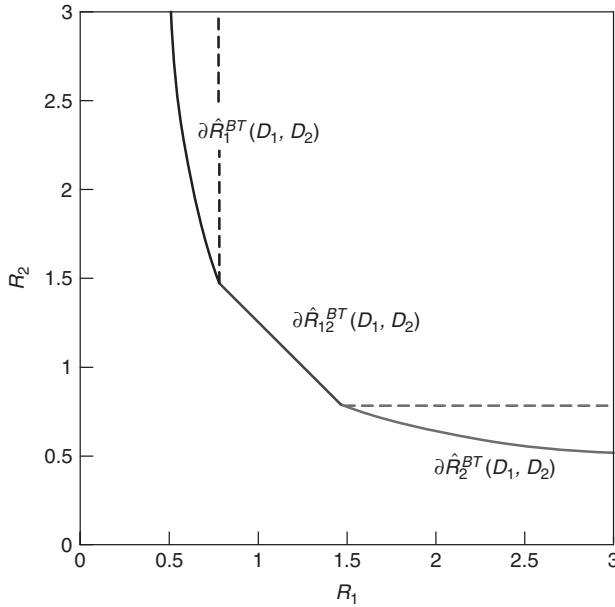
$$\hat{\mathcal{R}}_{12}^{\text{BT}}(D_1, D_2) = \left\{ (R_1, R_2) : R_1 + R_2 \geq \frac{1}{2} \log^+ \left[ (1 - \rho^2) \frac{\beta_{\max} \sigma_{y_1}^2 \sigma_{y_2}^2}{2D_1 D_2} \right] \right\}, \quad (19.15)$$

with

$$\beta_{\max} = 1 + \sqrt{1 + \frac{4\rho^2 D_1 D_2}{(1 - \rho^2)^2 \sigma_{y_1}^2 \sigma_{y_2}^2}},$$

and  $\log^+ x = \max\{\log x, 0\}$ .

Recently, the achievable BT rate region  $\hat{\mathcal{R}}^{\text{BT}}(D_1, D_2)$  is shown to be tight [10] for the two-terminal direct Gaussian MT source coding problem, that is,  $\hat{\mathcal{R}}^{\text{BT}}(D_1, D_2) = \mathcal{R}^*(D_1, D_2)$ . The main contribution of [10] is the formulation of an intermediate  $\mu$ -sum problem that connects the direct Gaussian MT source coding problem at hand with the quadratic Gaussian CEO problem [7, 8], whose solution is already known [26].



**Figure 19.6** The BT rate region for the direct Gaussian MT source coding problem with  $\sigma_{y_1}^2 = \sigma_{y_2}^2 = \sigma_y^2 = 1$ ,  $\rho = 0.9$ ,  $D_1 = D_2 = 0.1$ .

The boundary of the rate region  $\hat{\mathcal{R}}^{\text{BT}}(D_1, D_2)$  consists of a diagonal line segment and two curved portions (see Fig. 19.6 for an example) if and only if (iff) [10]

$$\rho^2 \frac{D_1}{\sigma_{y_1}^2} + 1 - \rho^2 > \frac{D_2}{\sigma_{y_2}^2} \quad \text{and} \quad \rho^2 \frac{D_2}{\sigma_{y_2}^2} + 1 - \rho^2 > \frac{D_1}{\sigma_{y_1}^2}. \quad (19.16)$$

Under this constraint, the set of all achievable rate pairs that minimize the sum-rate  $R = R_1 + R_2$  is called the *sum-rate bound* and will be denoted as  $\partial\hat{\mathcal{R}}_{12}^{\text{BT}}(D_1, D_2)$ .

In the special case when  $D_1 = D_2 = D$  and  $\sigma_{y_1}^2 = \sigma_{y_2}^2 = \sigma_y^2$ , the sum-rate bound  $\partial\hat{\mathcal{R}}_{12}^{\text{BT}}(D_1, D_2)$  becomes

$$\begin{aligned} \partial\hat{\mathcal{R}}_{12}^{\text{BT}}(D) = \left\{ (R_1, R_2) : R_1, R_2 \geq \frac{1}{2} \log^+ \left[ \frac{\sigma_y^2 \beta_{\max}^*}{2D} - \frac{\rho^2}{1 - \rho^2} \right]; \right. \\ \left. R_1 + R_2 = \frac{1}{2} \log^+ \left[ (1 - \rho^2) \frac{\beta_{\max}^* \sigma_y^4}{2D^2} \right] \right\}, \end{aligned} \quad (19.17)$$

where

$$\beta_{\max}^* = 1 + \sqrt{1 + \frac{4\rho^2 D^2}{(1 - \rho^2)^2 \sigma_y^4}}.$$

It is represented by the diagonal line segment in Figure 19.6.

Compared to joint encoding (and decoding), direct Gaussian MT source coding always suffers sum-rate loss, whose supremum was shown to be  $\frac{1}{2} \log_2 \frac{5}{4} = 0.161$  b/s [9] in the two-terminal setup.

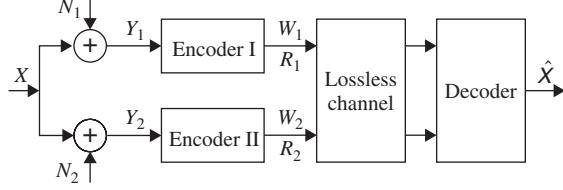


Figure 19.7 Two-terminal indirect MT source coding.

**19.2.3.2 Indirect MT Source Coding** The indirect MT source coding setup with two encoders is depicted in Figure 19.7. The *remote source*  $X$  and two noises  $N_1$  and  $N_2$  are mutually independent i.i.d. random variables drawn from the joint pdf  $f_{X, N_1, N_2}(x, n_1, n_2) = f_X(x)f_{N_1}(n_1)f_{N_2}(n_2)$ . The block  $\{Y_1^n, Y_2^n\}$  is a length- $n$  sequence of noisy observations:  $Y_1^n = X^n + N_1^n$ ,  $Y_2^n = X^n + N_2^n$  at the two encoders. The indirect system shares the form of encoder functions  $(\phi_1, \phi_2)$  with the direct system (19.7), while having a different decoder function

$$\psi : \{1, 2, \dots, 2^{nR_1}\} \times \{1, 2, \dots, 2^{nR_2}\} \rightarrow \mathcal{X}^n, \quad (19.18)$$

which reconstructs the remote source block as  $\hat{X}^n$ . Similar to the direct case, we define the achievable rate region  $\mathcal{R}^*(D)$  as the convex hull of the set of all achievable rate pairs  $(R_1, R_2)$  such that for any  $\epsilon > 0$ , there exists a large enough  $n$  and a triple  $(\phi_1, \phi_2, \psi)$  satisfying the distortion constraint

$$\frac{1}{n} \sum_{i=1}^n E[d(X_i, \hat{X}_i)] \leq D + \epsilon. \quad (19.19)$$

The exact achievable rate region for the indirect MT source coding problem is also unknown. For auxiliary random variables  $Z_1$  and  $Z_2$ , the inner rate region is given by [3–5]:

$$\begin{aligned} \hat{\mathcal{R}}(D) = \text{conv}\{\tilde{\mathcal{R}}(Z_1, Z_2) : Z_1 &\rightarrow Y_1 \rightarrow X \rightarrow Y_2 \rightarrow Z_2, \\ &\exists \psi(Z_1^n, Z_2^n) \text{ satisfying (19.19)}\}, \end{aligned} \quad (19.20)$$

while the outer rate region is [3–5]

$$\check{\mathcal{R}}(D) = \text{conv}\{\tilde{\mathcal{R}}(Z_1, Z_2) : Z_1 \rightarrow Y_1 \rightarrow X \rightarrow Y_2, Z_2 \rightarrow Y_2 \rightarrow X \rightarrow Y_1, \\ \exists \psi(Z_1^n, Z_2^n) \text{ satisfying (19.19)}\}. \quad (19.21)$$

In the indirect Gaussian MT source coding problem with MSE distortion measure,  $X$  is an i.i.d. Gaussian random variable  $\sim \mathcal{N}(0, \sigma_x^2)$ , and for  $i = 1, 2$  the noisy observations at the two encoders are given by  $Y_i = X + N_i$ , where  $N_1 \sim \mathcal{N}(0, \sigma_{n_1}^2)$  and  $N_2 \sim \mathcal{N}(0, \sigma_{n_2}^2)$  are i.i.d. Gaussian random variables independent of each other and  $X$ . For this special case, Yamamoto and Itoh [5] reported the Yamamoto–Itoh (YI) achievable rate region, which can be expressed in an equivalent form in terms of  $(\sigma_x^2, \sigma_{n_1}^2, \sigma_{n_2}^2, D)$  as

$$\hat{\mathcal{R}}^{\text{YI}}(D) = \text{conv}\left(\hat{\mathcal{R}}_1^{\text{YI}}(D) \cap \hat{\mathcal{R}}_2^{\text{YI}}(D) \cap \hat{\mathcal{R}}_{12}^{\text{YI}}(D)\right), \quad (19.22)$$

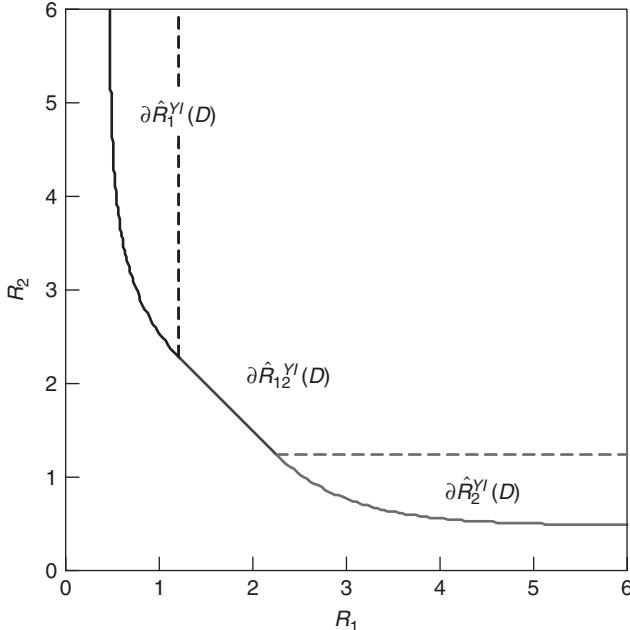
where

$$\begin{aligned} \hat{\mathcal{R}}_i^{\text{YI}}(D) = & \left\{ (R_1, R_2) : R_i \geq \frac{1}{2} \log^+ \right. \\ & \left[ \frac{\sigma_x^4 (2^{-2R_j} \sigma_x^2 + \sigma_{n_j}^2)^2 (\sigma_x^2 + \sigma_{n_j}^2)^{-1}}{2^{-2R_j} \sigma_x^4 (D - \sigma_{n_i}^2) + \sigma_x^2 D (\sigma_{n_1}^2 + \sigma_{n_2}^2) - \sigma_{n_1}^2 \sigma_{n_2}^2 (\sigma_x^2 - D)} \right] \left. \right\}, \\ & i, j = 1, 2, i \neq j, \quad (19.23) \end{aligned}$$

$$\hat{\mathcal{R}}_{12}^{\text{YI}}(D) = \left\{ (R_1, R_2) : R_1 + R_2 \geq \frac{1}{2} \log^+ \left[ \frac{4\sigma_x^2}{\sigma_{n_1}^2 \sigma_{n_2}^2 D (\frac{1}{\sigma_x^2} - \frac{1}{D} + \frac{1}{\sigma_{n_1}^2} + \frac{1}{\sigma_{n_2}^2})^2} \right] \right\}. \quad (19.24)$$

The YI achievable rate region (19.22) is shown to be tight [26],<sup>4</sup> that is,  $\hat{\mathcal{R}}^{\text{YI}}(D) = \mathcal{R}^*(D)$ . The boundary of  $\hat{\mathcal{R}}^{\text{YI}}(D)$  consists of a diagonal line segment and two curved portions (see Fig. 19.8 for an example) iff

$$\frac{1}{\sigma_x^2} + \frac{1}{\sigma_{n_1}^2} + \frac{1}{\sigma_{n_2}^2} > \frac{1}{D} > \max \left\{ \frac{1}{\sigma_x^2} - \frac{1}{\sigma_{n_1}^2} + \frac{1}{\sigma_{n_2}^2}, \frac{1}{\sigma_x^2} + \frac{1}{\sigma_{n_1}^2} - \frac{1}{\sigma_{n_2}^2} \right\}. \quad (19.25)$$



**Figure 19.8** The YI rate region for the indirect MT problem with  $\sigma_x^2 = 1$ ,  $\sigma_{n_1}^2 = \sigma_{n_2}^2 = 0.1$ ,  $D = 0.07$ .

<sup>4</sup>In fact, [26] showed that the achievable rate region of indirect Gaussian MT source coding is tight for any number of terminals.

Under this constraint, the *sum-rate bound*  $\partial\hat{\mathcal{R}}_{12}^{\text{YI}}(D)$  is defined as the set of all achievable rate pairs that minimize the sum-rate  $R = R_1 + R_2$ .

Note that in the symmetric case with  $\sigma_{n_1}^2 = \sigma_{n_2}^2 = \sigma_n^2$ , the sum-rate bound  $\partial\hat{\mathcal{R}}_{12}^{\text{YI}}(D)$  becomes

$$\partial\hat{\mathcal{R}}_{12}^{\text{YI}}(D) = \left\{ (R_1, R_2) : R_1 + R_2 = \frac{1}{2} \log^+ \left[ \frac{\sigma_x^2}{D\theta_2^2} \right], R_1, R_2 \geq \frac{1}{2} \log^+ \left[ \frac{2\sigma_x^2}{(\sigma_x^2 + D)\theta_2} \right] \right\}, \quad (19.26)$$

where

$$\theta_2 = 1 - \frac{\sigma_n^2}{2\sigma_x^2} \left[ \frac{\sigma_x^2}{D} - 1 \right]^+.$$

Compared to joint encoding of  $Y_1$  and  $Y_2$  (and joint decoding of the remote source  $X$ ), the sum-rate loss of indirect Gaussian two-terminal source coding is

$$\frac{1}{2} \log^+ \left[ \frac{4}{(\sigma_{n_1}^2 + \sigma_{n_2}^2)(\frac{1}{\sigma_x^2} - \frac{1}{D} + \frac{1}{\sigma_{n_1}^2} + \frac{1}{\sigma_{n_2}^2})} \right],$$

which goes to infinity when

$$D \rightarrow \left( \frac{1}{\sigma_x^2} + \frac{1}{\sigma_{n_1}^2} + \frac{1}{\sigma_{n_2}^2} \right)^{-1}.$$

In addition, compared to classic source coding of  $X$  (with one terminal), we see from (19.26) that the sum-rate loss with indirect Gaussian MT source coding is  $\frac{1}{2} \log^+[1/\theta_2^2]$  b/s if the number of terminals is  $L = 2$ . When  $L > 2$ , assuming  $\sigma_{n_1}^2 = \sigma_{n_2}^2 = \dots = \sigma_{n_L}^2 = \sigma_n^2$ , the minimum sum-rate of indirect Gaussian MT source coding, that is, for the Gaussian CEO problem, is [26]

$$R_L(D) = \frac{1}{2} \log^+ \left[ \frac{\sigma_x^2}{D\theta_L^L} \right] \quad (19.27)$$

where

$$\theta_L = 1 - \frac{\sigma_n^2}{L\sigma_x^2} \left[ \frac{\sigma_x^2}{D} - 1 \right]^+.$$

The rate loss (again over classic source coding of  $X$ ) in this case is  $\frac{1}{2} \log^+[1/\theta_L^L]$  b/s, with

$$\lim_{L \rightarrow \infty} \frac{1}{2} \log^+ \left[ \frac{1}{\theta_L^L} \right] = \frac{\sigma_n^2 \log(e)}{2\sigma_x^2} \left[ \frac{\sigma_x^2}{D} - 1 \right]^+.$$

Since

$$\frac{\sigma_x^2}{D} - 1 \geq \log \frac{\sigma_x^2}{D} \quad \text{and} \quad \lim_{D \rightarrow 0} \frac{\left[ \frac{\sigma_x^2}{D} - 1 \right]^+}{\log \frac{\sigma_x^2}{D}} = \infty,$$

thus the rate loss is relatively very large for small  $D$  [8].

## 19.3 CODE DESIGNS

### 19.3.1 Slepian–Wolf Code Design

Just like in Shannon's channel coding theorem [20], the random binning argument used in the proof of the SW theorem is asymptotic and nonconstructive. For practical SW coding, we can first try to design codes to approach the blue corner point  $A$  with  $R_1 + R_2 = H(X|Y) + H(Y) = H(X, Y)$  in the SW rate region of Figure 19.2. This is a problem of source coding of  $X$  with side information  $Y$  at the decoder (or *asymmetric* SW coding). If this can be done, then the other corner point  $B$  of the SW rate region can be approached by swapping the roles of  $X$  and  $Y$  and all points between these two corner points can be realized by time sharing—for example, using the two codes designed for the corner points 50% of the time each will result in the midpoint  $C$ . Another alternative is to design codes that directly approach the midpoint  $C$  or any point between  $A$  and  $B$  in Figure 19.2. These approaches are referred to as *symmetric* SW coding.

We start with *asymmetric* SW coding. The aim is to code  $X$  at a rate that approaches  $H(X|Y)$  based on the conditional statistics of (or the correlation model between)  $X$  and  $Y$  but not the specific  $y$  at the encoder. Slepian and Wolf first realized the close connection of DSC to channel coding and suggested the use of linear channel codes as a constructive approach for SW coding. The basic idea, published in a 1974 work by Wyner [27], is to partition the space of all possible source outcomes into bins indexed by *syndromes* of some “good” linear channel code for the specific correlation model. The set of all valid codewords (with zero syndrome) of the channel code forms only one bin, while other bins are shifts of this zero-syndrome bin. This syndrome-based approach is detailed below.

Let  $\mathcal{C}$  be an  $(n, k)$  binary linear block code with generator matrix  $\mathbf{G}$  of size  $k \times n$  and parity-check matrix  $\mathbf{H}$  of size  $(n - k) \times n$  such that  $\mathbf{G}\mathbf{H}^T = \mathbf{0}$ . The syndrome of any length- $n$  binary sequence  $\mathbf{x}$  with respect to code  $\mathcal{C}$  is defined as  $\mathbf{s}^{n-k} = \mathbf{x}\mathbf{H}^T$ , which is a length- $(n - k)$  binary sequence. Hence there are  $2^{n-k}$  distinct syndromes, each indexing  $2^k$  length- $n$  binary source sequences. A *coset*  $\mathcal{C}_{\mathbf{s}^{n-k}}$  of code  $\mathcal{C}$  is defined as the set of all length- $n$  sequences with syndrome  $\mathbf{s}^{n-k}$ , that is,  $\mathcal{C}_{\mathbf{s}^{n-k}} = \{\mathbf{x} \in \{0, 1\}^n : \mathbf{x}\mathbf{H}^T = \mathbf{s}^{n-k}\}$ .

Consider the problem of asymmetric SW coding of a binary source  $X$  with decoder side information  $Y$  (with discrete [1] or continuous [16] alphabet). Syndrome-based SW coding of  $\mathbf{x}$  proceeds as follows:

- *Encoding* The encoder computes the syndrome  $\mathbf{s}^{n-k} = \mathbf{x}\mathbf{H}^T$  and sends it to the decoder at rate  $R_X = n - k/n$  b/s. By the SW theorem [1],  $R_X = n - k/n \geq H(X|Y)$ .
- *Decoding* Based on the side information  $\mathbf{y}$  and received syndrome  $\mathbf{s}^{n-k}$ , the decoder finds the most probable source sequence  $\hat{\mathbf{x}}$  in the coset  $\mathcal{C}_{\mathbf{s}^{n-k}}$ , that is,  $\hat{\mathbf{x}} = \arg \max_{\mathbf{x} \in \mathcal{C}_{\mathbf{s}^{n-k}}} P(\mathbf{x}|\mathbf{y})$ .

**Example 19.1** Let us consider an example of the asymmetric SW coding process based on a systematic  $(7,4)$  Hamming code. Let  $X^n$  and  $Y^n$  be two uniformly distributed vectors of length  $n = 7$  bits and assume that the Hamming distance between  $\mathbf{x}$  and  $\mathbf{y}$  is always at most 1 bit. Under this assumption the vector  $\mathbf{x} \oplus \mathbf{y}$  can take eight different values and we assume that they are all equally likely. The source message  $X^n$  is

encoded and sent to the decoder, where the side information  $Y^n$  is available. The SW bound for this case is  $H(X^n|Y^n) = 3$  bits [1]. The systematic (7,4) Hamming code can correct at least one bit error in a 7-bit sequence and, thus, can be used to achieve the asymmetric SW limit in this example. The systematic (7,4) Hamming code is defined by the generator matrix

$$\mathbf{G} = [\mathbf{I}_4 \mathbf{P}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \quad (19.28)$$

Its parity matrix is

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Let the realizations of the sources of  $X^n$  and  $Y^n$  be  $\mathbf{x} = [0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0]$  and  $\mathbf{y} = [0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0]$ . Following the above encoding procedure, we form syndromes for  $\mathbf{x}$  as

$$\mathbf{s}^3 = \mathbf{x} \mathbf{H}^T = [0 \ 0 \ 1].$$

The decoder chooses the 7-bit sequence in the coset indexed by syndrome  $\mathbf{s}^3 = [0 \ 0 \ 1]$ , which is closest in Hamming distance sense to the side information  $\mathbf{y} = [0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0]$ . Since  $\mathbf{x} = [0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0]$  is a member of this coset and all members of a coset of the (7,4) Hamming code have Hamming distance of at least three between them, the decoder correctly decides for  $\hat{\mathbf{x}} = [0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0]$  and achieves the SW limit without error.

This syndrome-based approach was first implemented by Pradhan and Ramchandran [28] using block and trellis codes. More advanced channel codes such as turbo codes are later used for asymmetric SW coding [29–33] to achieve better performance.

The state of the art of SW code design [14] uses LDPC codes because of their capacity-approaching performance and their flexible code designs via density evolution [34]. Another reason lies in low-complexity LDPC decoding based on the message-passing algorithm, which can be applied in SW coding with only slight modification [14]. Specifically, as in the conventional message-passing algorithm, the input for the  $i$ th variable node is the log-likelihood ratio (LLR) of  $x_i$  defined as

$$L_{\text{ch}}(x_i) = \log \frac{P(X = 0|Y = y_i)}{P(X = 1|Y = y_i)}, \quad 0 \leq i \leq n - 1. \quad (19.29)$$

If  $X$  is uniform with  $P(X = 1) = P(X = 0) = \frac{1}{2}$ , we have

$$L_{\text{ch}}(x_i) \triangleq \log \frac{P(Y = y_i|X = 0)}{P(Y = y_i|X = 1)}, \quad 0 \leq i \leq n - 1. \quad (19.30)$$

The  $j$ th syndrome bit  $s_j$ ,  $0 \leq j \leq n - k - 1$ , is in fact the binary sum of the source bits corresponding to the ones in the  $j$ th row of the parity-check matrix  $\mathbf{H}$ . Hence the  $j$ th check node in the Tanner graph is related to  $s_j$ . The only difference from conventional LDPC decoding is that one needs to flip the sign of the check-to-bit LLR if the corresponding syndrome bit  $s_j$  is one [14]. Moreover, density evolution [34] can be employed to analyze the iterative decoding procedure without any modification [35].

Example 19.1 is very special since it is constructed such that the SW limit is achieved without any decoding error. In practice, there is a small probability of loss in general at the SW decoder due to channel coding. In addition, the code rate  $k/n$  of  $\mathcal{C}$  and syndrome-based code design depend on the source correlation model. In the binary symmetric model,  $\{(X_i, Y_i)\}_{i=1}^{\infty}$  is a sequence of i.i.d. drawings of a pair of correlated binary Bernoulli (0.5) random variables  $X$  and  $Y$ , and the correlation between  $X$  and  $Y$  is modeled by a “virtual” BSC with crossover probability  $p$ . In this case  $H(X|Y) = H(p) = -p \log_2 p - (1-p) \log_2(1-p)$ . The best result [14] reported in the literature shows a 0.03-bit gap to the SW limit  $H(p)$  when length  $10^5$ -bit LDPC codes are used with a target bit error rate of  $10^{-6}$ .

We now turn to *symmetric* SW code design and again focus on the binary symmetric case, that is,  $X$  and  $Y$  are binary and unbiased with  $P(X \oplus Y = 1) = p$ , where  $\oplus$  denotes binary addition. Our goal is to separately compress  $X$  and  $Y$ , and to jointly reconstruct them. Due to the SW theorem [1], any rate pair  $(R_X, R_Y)$  that satisfies

$$\begin{aligned} R_X &\geq H(X|Y) = H(p), & R_Y &\geq H(Y|X) = H(p), \\ R_X + R_Y &\geq H(X, Y) = H(p) + 1 \end{aligned} \quad (19.31)$$

is achievable.

In [15] an efficient algorithm to design good symmetric SW codes by partitioning a single linear parity-check code was proposed. Although this algorithm can be applied to compression of multiple correlated sources (see Section 19.4.1), we restrict ourselves to two sources here.

Suppose that we aim at approaching a point  $(R_X, R_Y)$  (i.e., to compress  $X$  at rate  $R_X$  and  $Y$  at  $R_Y$ ) that satisfies (19.31). Let  $\mathcal{C}$  be an  $(n, k)$  linear channel block code with  $k = (2 - R_X - R_Y)n$ . Although both systematic and nonsystematic codes can be used for  $\mathcal{C}$  [15, 36], for the sake of easy exposition, we assume that  $\mathcal{C}$  is a systematic channel code with generator matrix  $\mathbf{G} = [\mathbf{I}_k \quad \mathbf{P}_{k \times (n-k)}]$ . We partition  $\mathcal{C}$  into two subcodes,  $\mathcal{C}_1$  and  $\mathcal{C}_2$ , defined by generator matrices

$$\mathbf{G}_1 = [\mathbf{I}_{m_1} \quad \mathbf{O}_{m_1 \times m_2} \quad \mathbf{P}_1] \quad \text{and} \quad \mathbf{G}_2 = [\mathbf{O}_{m_2 \times m_1} \quad \mathbf{I}_{m_2} \quad \mathbf{P}_2],$$

which consist of the top  $m_1$  and bottom  $m_2$  rows of  $\mathbf{G}$ , respectively, where  $m_1 \triangleq (1 - R_X)n$ ,  $m_2 \triangleq (1 - R_Y)n$  (thus  $m_1 + m_2 = k$ ). Then the parity-check matrices for  $\mathcal{C}_1$  and  $\mathcal{C}_2$  can be written as

$$\mathbf{H}_1 = \begin{bmatrix} \mathbf{O}_{m_2 \times m_1} & \mathbf{I}_{m_2} & \mathbf{O}_{m_2 \times (n-k)} \\ \mathbf{P}_1^T & \mathbf{O}_{(n-k) \times m_2} & \mathbf{I}_{n-k} \end{bmatrix}, \quad (19.32)$$

and

$$\mathbf{H}_2 = \begin{bmatrix} \mathbf{I}_{m_1} & \mathbf{O}_{m_1 \times m_2} & \mathbf{O}_{m_1 \times (n-k)} \\ \mathbf{O}_{(n-k) \times m_1} & \mathbf{P}_2^T & \mathbf{I}_{n-k} \end{bmatrix}, \quad (19.33)$$

respectively.

*Encoding* It is done by multiplying  $\mathbf{x}$  and  $\mathbf{y}$ , the realization of  $X^n$  and  $Y^n$ , respectively, by the corresponding parity-check matrix  $\mathbf{H}_1$  and  $\mathbf{H}_2$ , respectively. We partition the length- $n$  vectors  $\mathbf{x}$  and  $\mathbf{y}$  into three parts (of lengths  $m_1$ ,  $m_2$ , and  $n - k$ ):

$$\mathbf{x} = [\mathbf{u}_1^{m_1} \quad \mathbf{u}_2^{m_2} \quad \mathbf{u}_3^{n-k}], \quad \mathbf{y} = [\mathbf{v}_1^{m_1} \quad \mathbf{v}_2^{m_2} \quad \mathbf{v}_3^{n-k}]. \quad (19.34)$$

Then, the resulting syndrome vectors are

$$\begin{aligned}\mathbf{s}_1^{n-m_1} &= \mathbf{xH}_1^T = [\mathbf{u}_2^{m_2} \quad \mathbf{u}_3^{n-k} \oplus \mathbf{u}_1^{m_1} \mathbf{P}_1] \quad \text{and} \\ \mathbf{s}_2^{n-m_2} &= \mathbf{xH}_2^T = [\mathbf{v}_1^{m_1} \quad \mathbf{v}_3^{n-k} \oplus \mathbf{v}_2^{m_2} \mathbf{P}_2],\end{aligned}\quad (19.35)$$

which are directly sent to the decoder. It is easy to see that the total number of transmitted bits for  $\mathbf{x}$  and  $\mathbf{y}$  is  $m_2 + (n - k) = nR_X$  and  $m_1 + (n - k) = nR_Y$ , respectively, with the desirable sum-rate of  $R_X + R_Y$  b/s.

*Decoding* Upon receiving the syndrome vectors  $\mathbf{s}_1^{n-m_1}$  and  $\mathbf{s}_2^{n-m_2}$ , the decoder forms an auxiliary length- $n$  row vector as

$$\begin{aligned}\mathbf{s}^n &= [\mathbf{v}_1^{m_1} \quad \mathbf{u}_2^{m_2} \quad (\mathbf{u}_3^{n-k} \oplus \mathbf{v}_3^{n-k}) \oplus \mathbf{u}_1^{m_1} \mathbf{P}_1 \oplus \mathbf{v}_2^{m_2} \mathbf{P}_2] \\ &= [\mathbf{v}_1^{m_1} \quad \mathbf{u}_2^{m_2} \quad (\mathbf{u}_3^{n-k} \oplus \mathbf{v}_3^{n-k}) \oplus [\mathbf{u}_1^{m_1} \quad \mathbf{v}_2^{m_2}] \mathbf{P}].\end{aligned}$$

Then it finds a codeword  $\mathbf{c}^n$  of the main code  $\mathcal{C}$  closest (in Hamming distance) to  $\mathbf{s}^n$ . If the decoding is successful, the decoder returns  $\mathbf{c}^n = [\mathbf{u}_1^{m_1} \quad \mathbf{v}_2^{m_2} \quad [\mathbf{u}_1^{m_1} \quad \mathbf{v}_2^{m_2}] \mathbf{P}]$  as the closest codeword. Let the vector  $[\hat{\mathbf{u}}_1^{m_1} \quad \hat{\mathbf{v}}_2^{m_2}]$  be the systematic part of  $\mathbf{c}^n$ , then  $\mathbf{x}$  and  $\mathbf{y}$  are recovered as

$$\begin{aligned}\hat{\mathbf{x}} &= \hat{\mathbf{u}}_1^{m_1} \mathbf{G}_1 \oplus [\mathbf{O}_{1 \times m_1} \quad \mathbf{u}_2^{m_2} \quad \mathbf{u}_3^{n-k} \oplus \mathbf{u}_1^{m_1} \mathbf{P}_1] \quad \text{and} \\ \hat{\mathbf{y}} &= \hat{\mathbf{v}}_2^{m_2} \mathbf{G}_2 \oplus [\mathbf{v}_1^{m_1} \quad \mathbf{O}_{1 \times m_2} \quad \mathbf{v}_3^{n-k} \oplus \mathbf{v}_2^{m_2} \mathbf{P}_2].\end{aligned}\quad (19.36)$$

**Example 19.2** We now extend the asymmetric code construction of Example 19.1 to the symmetric case using the same systematic (7,4) Hamming code. In the symmetric case both source messages are separately encoded and sent to the joint decoder, whose task is to losslessly reconstruct both of them. The SW bound for this case is 10 bits [1]. This was achieved in the asymmetric scenario by transmitting one source, the side information  $Y^n$  at rate  $R_Y = H(Y^n) = 7$  bits and by coding the second source  $X^n$  at  $R_X = H(X^n|Y^n) = 3$  bits. We show how the same total rate can be achieved by using  $R_X = R_Y = 5$  bits. We first form two subcodes of the (7,4) Hamming code,  $\mathcal{C}_1$  and  $\mathcal{C}_2$ , by partitioning its generator matrix  $\mathbf{G}$  in (19.28) into two generator matrices,  $\mathbf{G}_1$  that contains the first two rows of  $\mathbf{G}$ , and  $\mathbf{G}_2$  that contains the last two rows.  $X^n$  is coded using  $\mathcal{C}_1$  and  $Y^n$  using  $\mathcal{C}_2$ . Let  $\mathbf{P}^T = [\mathbf{P}_1^T \quad \mathbf{P}_2^T]$ . Then for the  $5 \times 7$  parity-check matrices  $\mathbf{H}_1$  and  $\mathbf{H}_2$  of  $\mathcal{C}_1$  and  $\mathcal{C}_2$ , respectively, we get from (19.32) and (19.33)

$$\mathbf{H}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{H}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Using the same realizations of the sources as in the asymmetric example, we start the above encoding process by forming syndromes for both  $\mathbf{x}$  and  $\mathbf{y}$ . To do so, we write  $\mathbf{x}$  and  $\mathbf{y}$  as

$$\mathbf{x} = [\mathbf{u}_1^2 \quad \mathbf{u}_2^2 \quad \mathbf{u}_3^3] = [00 \quad 10 \quad 110], \quad \mathbf{y} = [\mathbf{v}_1^2 \quad \mathbf{v}_2^2 \quad \mathbf{v}_3^3] = [01 \quad 10 \quad 110].$$

The 5-bit syndromes,  $s_1^5$  and  $s_2^5$ , formed by the two subcodes are

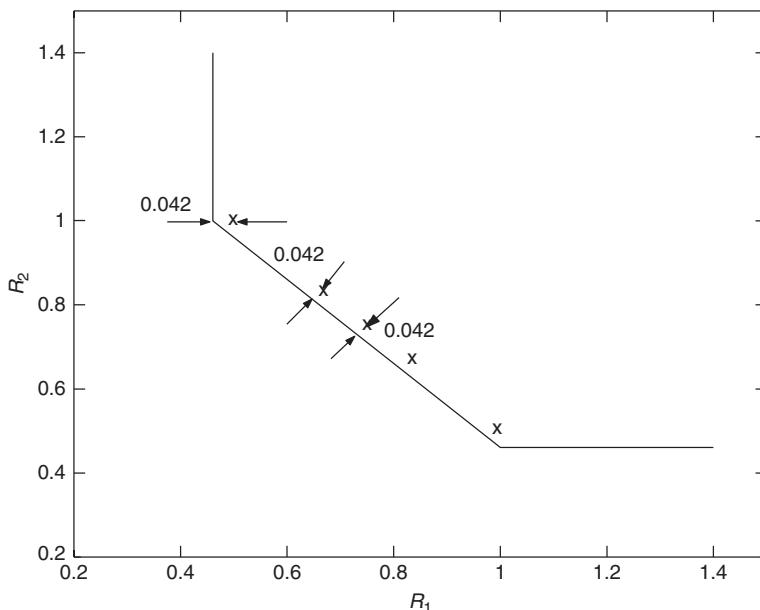
$$s_1^5 = \mathbf{x}H_1^T = [10 \quad 110]^T \quad \text{and} \quad s_2^5 = \mathbf{y}H_2^T = [01 \quad 001]^T.$$

The decoding process starts by first combining the two 5-bit syndromes,  $s_1^5$  and  $s_2^5$ , into the auxiliary 7-bit vector  $\mathbf{s}^7 = [01 \quad 10 \quad 111]$ . Because of the type of correlation between  $X^n$  and  $Y^n$ , the auxiliary vector  $\mathbf{s}^7$  is expected to be within Hamming distance of one from one of the codewords, so the decoder is always error free as the minimum Hamming distance of the code  $\mathcal{C}$  is three. For  $\mathbf{s}^7 = [01 \quad 10 \quad 111]$  the closest codeword is  $\mathbf{c}^7 = [0010111]$ . The corresponding reconstructions  $\hat{\mathbf{u}}_1^2 = 00$  and  $\hat{\mathbf{v}}_2^2 = 10$  are then obtained as the systematic part of the codeword  $\mathbf{c}^7$ . The sources are finally reconstructed as  $\hat{\mathbf{x}} = \hat{\mathbf{u}}_1^2 \mathbf{G}_1 \oplus [00 \quad s_1^5] = [0010110]$  and  $\hat{\mathbf{y}} = \hat{\mathbf{v}}_2^2 \mathbf{G}_2 \oplus [\mathbf{v}_1^2 \quad 00 \quad \mathbf{v}_3^3 \oplus \mathbf{v}_2^2 \mathbf{P}_2] = [0110110]$ .

In addition, to realize the same total rate of 10 bits with  $R_X = 4$  bits and  $R_Y = 6$  bits (or with  $R_X = 6$  bits and  $R_Y = 4$  bits), all we need to do is to form  $\mathcal{C}_1$  and  $\mathcal{C}_2$  by partitioning  $\mathbf{G}$  in (19.28) into  $\mathbf{G}_1$  with its first three rows (or one row) and  $\mathbf{G}_2$  that contains its last one row (or three rows) before proceeding with encoding and decoding. The curious reader will see that Example 19.1 with  $R_X = 3$  bits and  $R_Y = 7$  bits merely corresponds to assigning all four rows of  $\mathbf{G}$  to  $\mathbf{G}_1$ .

It is shown in [15] that if the  $(n, k)$  main code  $\mathcal{C}$  approaches the capacity of a BSC with crossover probability  $p$ , then the above designed symmetric SW code approaches the SW limit for the same binary symmetric correlation channel model. Using turbo/LDPC codes, the authors obtain the best results for binary symmetric sources that are 0.038–0.042 bit way from the SW sum-rate limit. See Figure 19.9.

In summary, if the correlation between the source output  $X$  and the side information  $Y$  can be modeled with a “virtual” correlation channel, then a good channel code



**Figure 19.9** Symmetric SW coding results [15] using length  $10^5$ -bit turbo codes with a bit error rate of  $10^{-5}$  for binary symmetric sources.

over this channel can provide us with a good SW code through the syndromes and the associated coset codes. Thus, when the source correlation is known a priori, the seemingly source coding problem of SW coding is actually a channel coding one, and near-capacity channel codes such as turbo and LDPC codes can be used to approach the SW limits.

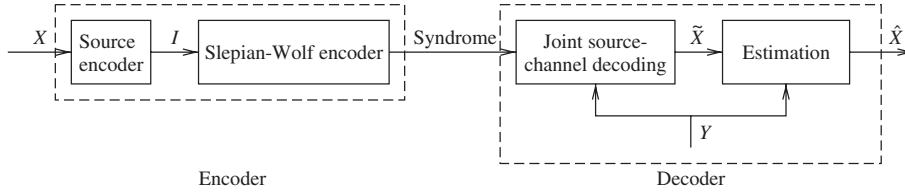
For the binary setup considered in this section there is significant work on an alternative approach in the literature using parity bits instead of syndrome bits as described above. Details about this approach can be found in [29, 31]. In general the syndrome bits approach allows the direct use of available channel codes while the parity bits approach requires a more intensive code design effort to achieve similar performance [33, 37]. The real advantage of the parity bits approach is in the case of noisy channel(s) and joint source-channel coding [31, 38, 39] where transmitting unprotected syndrome bits over a noisy channel is no longer an alternative.

Practical SW code designs for other correlation models [40, 41] and more than two sources have appeared recently in the literature. However, there is no systematic approach to general practical SW code design yet, in the sense of being able to account for an arbitrary number of sources [42, 43] with nonbinary alphabets [44] and possibly with memory [45] in the marginal source and/or correlation statistics. In addition, the more challenging problems of scalable [46–48] and/or universal SW code design remain open. Garcia-Frias and Zhao [29] include a step toward universal SW code design by allowing estimation of the unknown correlation parameter between the two correlated sources at the iterative SW decoder.

Nevertheless, there has been a significant amount of works, and the available SW code designs perform well for a number of different scenarios. These designs are very important, as asymmetric or symmetric SW coding plays the role of conditional or joint entropy coding, respectively. Not only can SW coding be considered an alternative [49] to entropy coding in classic lossless source coding but also the extension of entropy coding to problems with side information and/or distributed sources. In addition, apart from its importance as a separate problem, when combined with quantization, SW coding can provide a practical approach to lossy DSC problems, such as the WZ problem considered next, similarly to the way quantization and entropy coding are combined in classic lossy source coding.

### 19.3.2 Wyner–Ziv Code Design

In WZ coding, because we are introducing loss/distortion to the source, quantization of the source  $X$  is needed. Usually there is still correlation remaining in the quantized version of  $X$  and the side information  $Y$ , and SW coding should be employed to exploit this correlation to reduce the rate. Since SW coding is based on channel coding, WZ coding is a source–channel coding problem. There are quantization loss due to source coding and binning loss due to channel coding. In order to reach the WZ limit, one needs to employ both source codes (e.g., TCQ [50]) that can achieve the granular gain [51] and channel codes (e.g., turbo and LDPC codes) that can approach the SW limit. In addition, the side information  $Y$  can be used in jointly decoding and estimating  $\hat{X}$  at the decoder to help reduce the distortion  $d(X, \hat{X})$  for nonbinary sources, especially at low bit rate. Figure 19.10 depicts the block diagram of a generic WZ coder. Implemented code designs for the binary symmetric and the quadratic Gaussian cases are discussed next.



**Figure 19.10** Block diagram of generic WZ coder.

**19.3.2.1 Binary Symmetric Case** Recall that in the syndrome-based scheme [27] for lossless SW coding, a linear  $(n, k)$  binary block code is used. There are  $2^{n-k}$  distinct syndromes, each indexing a set (bin) of  $2^k$  binary words of length  $n$  that preserve the Hamming distance properties of the original code. In compressing, a sequence of  $n$  input bits is mapped into its corresponding  $(n - k)$  syndrome bits, achieving a compression ratio of  $n : (n - k)$ .

For WZ coding, Shamai, Verdu, and Zamir [52] generalized the syndrome-based scheme of [27] using nested linear binary block codes [53]. According to this nested scheme, a linear  $(n, k_2)$  binary block code is again used to partition the space of all binary words of length  $n$  into  $2^{n-k_2}$  bins of  $2^{k_2}$  elements, each indexed by a unique syndrome value. Out of these  $2^{n-k_2}$  bins only  $2^{k_1-k_2}$  ( $k_1 \geq k_2$ ) are used, and the elements of the remaining  $2^{n-k_2} - 2^{k_1-k_2}$  sets are “quantized” to the closest, in Hamming distance sense, binary word of the allowable  $2^{k_1-k_2} \times 2^{k_2} = 2^{k_1}$  ones. This “quantization” can be viewed as a  $(n, k_1)$  binary block source code. Then the linear  $(n, k_2)$  binary block code can be considered to be a coarse channel code nested inside the  $(n, k_1)$  fine source code.

Let  $\mathbf{G}_1$  be the  $k_1 \times n$  generator matrix for the  $(n, k_1)$  source code and  $\mathbf{G}_2$  be the  $k_2 \times n$  generator matrix for the  $(n, k_2)$  channel code. The nested code construction assumes that we can write

$$\mathbf{G}_2 = \mathbf{G}_3 \mathbf{G}_1,$$

where  $\mathbf{G}_3$  is an  $k_2 \times k_1$  generator matrix of a  $(k_1, k_2)$  linear block code. If  $\mathbf{H}_1$ ,  $\mathbf{H}_2$ , and  $\mathbf{H}_3$  are parity-check matrices corresponding to the generator matrices  $\mathbf{G}_1$ ,  $\mathbf{G}_2$ , and  $\mathbf{G}_3$ , respectively, then we form  $\mathbf{H}_2$  as

$$\mathbf{H}_2 = \begin{bmatrix} \mathbf{H}_3 & \mathbf{O}_{(k_1-k_2) \times (n-k_1)} \\ \mathbf{H}_1 & \end{bmatrix} \quad (19.37)$$

where for simplicity we assumed that  $\mathbf{G}_1$  is systematic, but the extension to a nonsystematic  $\mathbf{G}_1$  is straightforward.

Now note that for an  $n$ -bit sequence  $\mathbf{x} = [\mathbf{x}_1^{k_1} \quad \mathbf{x}_2^{(n-k_1)}]$ , its syndrome with respect to  $\mathbf{H}_2$  is

$$\mathbf{s}_2 = \mathbf{x} \mathbf{H}_2^T = \begin{bmatrix} \mathbf{x}_1^{k_1} \mathbf{H}_3^T & \mathbf{x} \mathbf{H}_1^T \end{bmatrix} = [\mathbf{s}_3 \quad \mathbf{s}_1].$$

**Encoding** The first step in the encoding process is to “quantize” the  $n$ -bit source output  $\mathbf{x}$  using the code  $\mathbf{G}_1$  to the closest  $\tilde{\mathbf{x}}$  out of the  $2^{k_1}$   $n$ -bit codewords of  $\mathbf{G}_1$ . In the second step of the encoding process the syndrome of the  $n$ -bit sequence  $\tilde{\mathbf{x}}$  is determined with respect to the code  $\mathbf{G}_2$ , which based on the above discussion is

$$\tilde{\mathbf{s}}_2 = \tilde{\mathbf{x}} \mathbf{H}_2^T = \begin{bmatrix} \tilde{\mathbf{x}}_1^{k_1} \mathbf{H}_3^T & \tilde{\mathbf{x}} \mathbf{H}_1^T \end{bmatrix} = [\tilde{\mathbf{s}}_3 \quad \mathbf{O}_{1 \times (n-k_1)}],$$

where in the last step we used the fact that  $\tilde{\mathbf{x}}$  is a codeword of  $\mathbf{G}_1$  and therefore its syndrome with respect to the code  $\mathbf{G}_1$  is an all-zeros sequence. So, the  $(k_1 - k_2)$ -bit sequence  $\tilde{\mathbf{s}}_3$  is the encoder output.

*Decoding* Receiving  $\tilde{\mathbf{s}}_3$ , the decoder forms the syndrome  $\tilde{\mathbf{s}}_2 = [\tilde{\mathbf{s}}_3 \quad \mathbf{O}_{n-k_1}]$  and then determines the member of the coset of the code  $\mathbf{G}_2$  indexed by  $\tilde{\mathbf{s}}_2$  that is closest to the side information  $\mathbf{y}$ . Just note that the coset indexing is done through  $\mathbf{H}_2$  as defined in (19.37) because other valid  $\mathbf{H}_2$  do not have the last  $(n - k_1)$  bits equal to zero when only codewords from code  $\mathbf{G}_1$  are considered. If the decoding is successful, the decoder reconstructs  $\tilde{\mathbf{x}}$  without error, that is, the only distortion remaining at the output of the decoder is the quantization error introduced by the encoder.

This nested approach is meaningful when the correlation between the source and side information is weaker than the correlation between the source and quantized source (distortion introduced by the source code). Otherwise just setting  $\hat{\mathbf{x}}$  at the decoder equal to the side information  $\mathbf{y}$  suffices and nothing needs to be transmitted by the encoder. In the following example the correlation is weaker than the distortion and the nested binary code construction is demonstrated as well as the corresponding encoding and decoding processes.

**Example 19.3** Let  $X^n$  and  $Y^n$  be two uniformly distributed vectors of length  $n = 7$  bits and assume that the Hamming distance between  $\mathbf{x}$  and  $\mathbf{y}$  is always at most two bits. Under this assumption the vector  $\mathbf{x} \oplus \mathbf{y}$  can take 29 different values, and we assume that they are all equally likely. The source message  $X^n$  is encoded and sent to the decoder, where the side information  $Y^n$  is available. The SW bound for this case is  $H(X^n|Y^n) = \log_2(29) = 4.858$  bits [1]. The systematic (7,4) Hamming code is used in this case as the  $(n, k_1)$  binary block source code quantizing each 7-bit realization of  $X^n$  to the closest codeword.  $\mathbf{G}_1$  is given in (19.28). As before let us denote the quantized version of  $X^n$  as  $\tilde{X}^n$ . The Hamming distance between  $\tilde{\mathbf{x}}$  and  $\mathbf{x}$  is at most 1 bit and between  $\tilde{\mathbf{x}}$  and  $\mathbf{y}$  is at most 3 bits. Using the repeat-by-7  $(n, k_2)$  channel code with  $\mathbf{G}_2 = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$  we can correct up to 3 bit errors and thus recover  $\tilde{\mathbf{x}}$  from  $\mathbf{y}$  without errors. In this example  $\mathbf{G}_3 = [1 \ 1 \ 1 \ 1]$  and thus

$$\mathbf{H}_3 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}. \quad (19.38)$$

Let the realizations of the source  $X^n$  and the side information  $Y^n$  be  $\mathbf{x} = [1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0]$  and  $\mathbf{y} = [1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1]$ , respectively. Then  $\tilde{\mathbf{x}} = [1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]$  and the encoder uses the first 4 bits of  $\tilde{\mathbf{x}}$  to form its output:

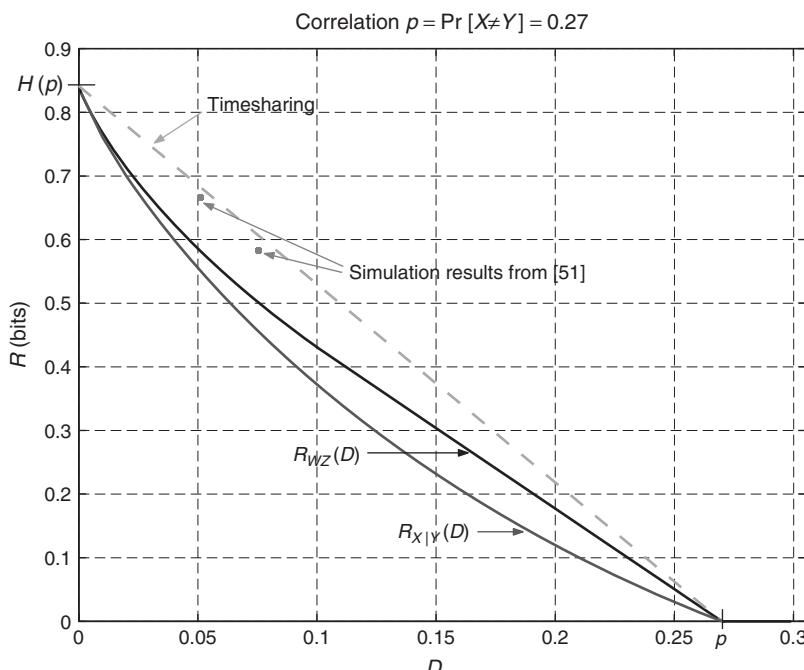
$$\tilde{\mathbf{s}}_3 = \tilde{\mathbf{x}}_1^4 \mathbf{H}_3^T = [0 \ 1 \ 0]. \quad (19.39)$$

The decoder upon receiving  $\tilde{\mathbf{s}}_3$  forms  $\tilde{\mathbf{s}}_2 = [0 \ 1 \ 0 \ 0 \ 0 \ 0]$ . This syndrome indexes the coset of  $\mathbf{G}_2$  with members  $\{[0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1], [1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]\}$ . Since  $\mathbf{y} = [1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1]$ , the decoder outputs  $\hat{\mathbf{x}} = [1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]$ , that is, it recovered  $\tilde{\mathbf{x}} = [1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]$  without error. In this setup there are some rate savings as 3 bits are sent from the encoder per 7-bit sequence instead of the 4.858 bits required for lossless transmission, at the cost of introducing 0.125 distortion per source bit.

In the above example time sharing between SW coding at a rate of 4.858 bits with zero distortion and zero-rate transmission with distortion 0.241 (just setting  $\hat{\mathbf{x}} = \mathbf{y}$ ), requires a bit rate of 2.342 bits per 7-bit sequence to achieve distortion of 0.125 per source bit, that is, 0.658 bits better than the proposed nested code approach. To come close to the WZ limit and to outperform time sharing, both codes in the above nested scheme should be good, that is, a good fine source code is needed with a good coarse channel subcode [52, 53]. In the above example just the systematic (7,4) Hamming source code suffers a rate loss of 0.115 bits per source bit or 0.805 bits per 7-bit sequence where the difference is taken from the  $1 - h(d)$  per bit binary rate distortion function [20]. So the use of a more powerful source code and probably also a more powerful channel code would have helped outperform time sharing and come close to the theoretical WZ limit in this case.

Liveris et al. proposed a more powerful binary WZ coding scheme in [54] that outperformed time sharing. The scheme in [54] is based on concatenated codes, where from the constructions in [32] the use of good channel codes is guaranteed. As for the source code, its operation resembles that of TCQ, and, hence, it is expected to be a good source code. The scheme in [54] can come within 0.09 bit from the theoretical limit (see Fig. 19.11). This is the only result reported so far coming so close to the binary WZ limit.

In this binary setup, no estimation is performed in the WZ decoder unlike the more general WZ approach shown in Figure 19.10. So, in Figure 19.11 the 0.09-bit gap in rate between the simulated points and the WZ limit can be separated into source



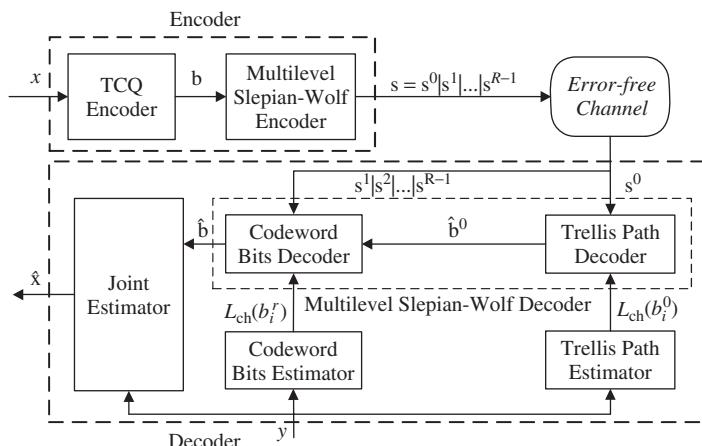
**Figure 19.11** Simulated performance of the nested scheme in [54] for binary WZ coding for correlation  $p = 0.27$ . The time-sharing line between the zero-rate point  $(p, 0)$  and the SW point  $[0, H(p)]$  is also shown.

coding rate loss and channel coding rate loss, showing the relative strengths of the two components of the nested code. For the curve portion of  $R_{WZ}(D)$  and for relatively low correlation, for example,  $p = 0.27$  as in Figure 19.11, the binary WZ code design problem provides very valuable insight of the interaction between source and channel coding in a nested code, as described in the nested construction in this section. When targeting the line portion of  $R_{WZ}(D)$  or higher correlation, that is, lower  $p$ , time sharing between a high-rate code [e.g., an SW code with rate  $H(p)$  or a nested code such as the ones simulated in Fig. 19.11] and the zero-rate point should be used.

**19.3.2.2 Quadratic Gaussian Case** Zamir et al. [53] outlined a constructive mechanism for quadratic Gaussian WZ coding using a pair of nested lattice codes. An SW-coded nested quantization paradigm was put forth in [16] for WZ coding. At high-resolution/rate, asymptotic performance bounds of SW-coded nested quantization similar to those in classic source coding are established in [16], showing that ideal SW-coded one-/two-dimensional (1D/2D) nested lattice quantization performs 1.53/1.36 dB worse than the WZ distortion rate function  $D_{WZ}(R)$  with probability of almost 1; performances close to the corresponding theoretical limit were obtained by using 1D and 2D nested lattice quantizers, together with irregular LDPC codes for SW coding. Since it is very difficult to implement high-dimensional lattice quantizers, research on trellis-based codes as a way of realizing high-dimensional nested lattice codes has been studied recently [28, 55–57].

Yang et al. [17] considered TCQ and LDPC codes for the quadratic Gaussian WZ coding problem. The block diagram of the resulting SWC-TCQ scheme is depicted in Figure 19.12.

After TCQ of the source  $X$ , LDPC codes are used to implement SW coding of the quantized source  $Q(X)$  with side information  $Y$  at the decoder. Assuming 256-state TCQ and ideal SW coding in the sense of achieving the theoretical limit  $H(Q(X)|Y)$ , they experimentally show that SW-coded TCQ performs 0.2 dB away from the WZ distortion rate function  $D_{WZ}(R)$  at high rate. This result mirrors that of entropy-constrained TCQ in classic source coding of Gaussian sources. Furthermore, using 8192-state TCQ and assuming ideal SW coding, their simulations show that



**Figure 19.12** Block diagram of the proposed SWC-TCQ scheme.

SW-coded TCQ performs only 0.1 dB away from  $D_{WZ}(R)$  at a high rate. These results establish the practical performance limit of SW-coded TCQ for quadratic Gaussian WZ coding. Practical designs give performance very close to the theoretical limit. For example, with 8192-state TCQ, irregular LDPC codes for SW coding and optimal non-linear estimation at the decoder, the performance gap to  $D_{WZ}(R)$  shown in Figure 19.13 is 0.20 and 0.93 dB at 3.83 and 1.05 b/s, respectively. At low rate, WZ code design becomes more challenging because the rate of TCQ has to be above certain nominal value (e.g., 1 b/s) and estimation plays an important role. The authors of [17] resort to trellis-coded VQ and nonlinear estimation at low rate. Using 256-state 4D TCVQ, they reported the performance gap to  $D_{WZ}(R)$  is 0.54 and 0.80 dB at 1.0 and 0.5 b/s, respectively. See also Figure 19.13.

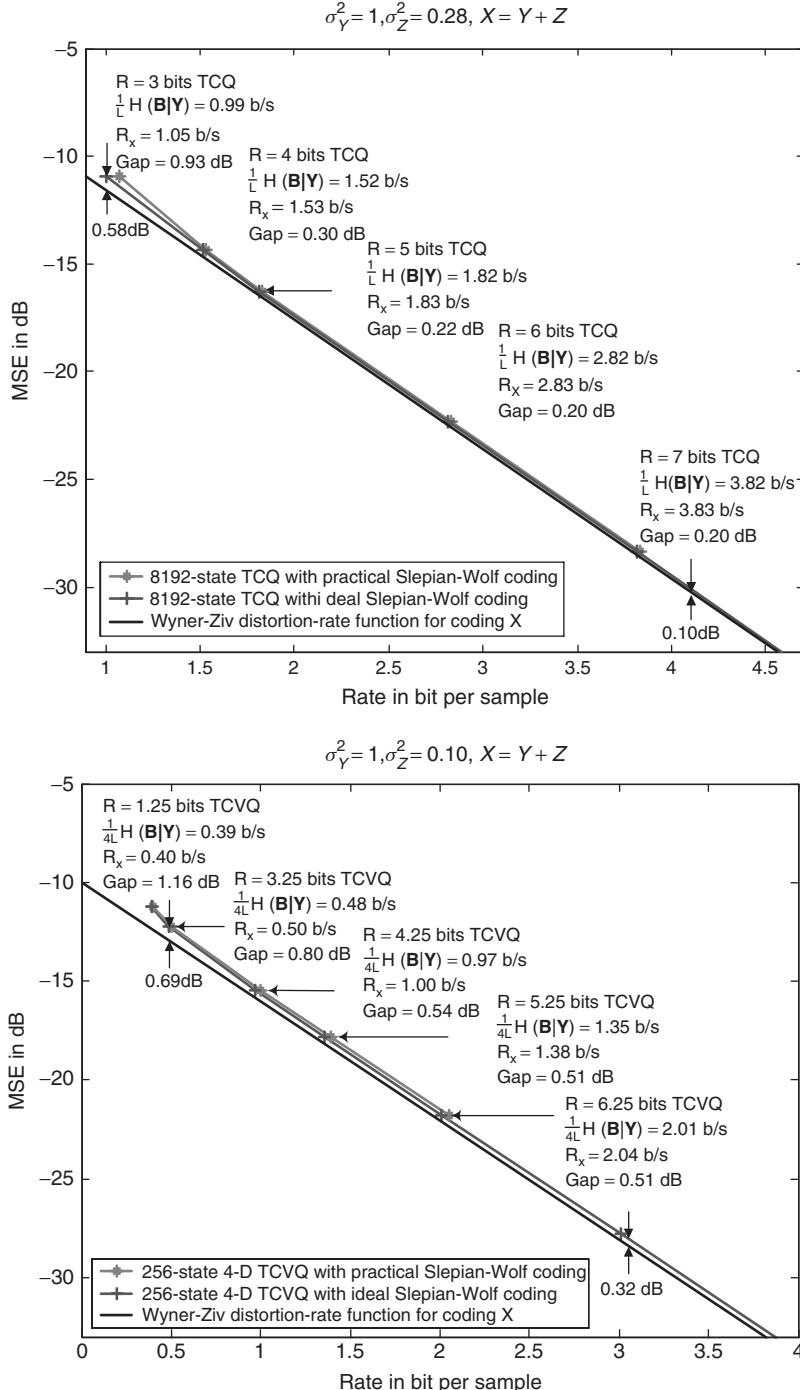
**19.3.2.3 Successive WZ Coding** Steinberg and Merhav [58] recently extended Equitz and Cover's work [59] on successive refinement of information to WZ coding, showing that both the binary symmetric source and the jointly Gaussian source (considered in Section 19.2.2) are successively refinable. Cheng and Xiong [35] further pointed out that the broader class of sources that satisfy the general condition of no rate loss [24] for WZ coding is also successively refinable. Practical layered WZ code design for Gaussian sources based on nested scalar quantization and multilevel LDPC code for SW coding was also presented in [35]. Layered Wyner–Ziv video coding for error-robust delivery was studied in [60] and details are given in Section 19.4.3.

### 19.3.3 Multiterminal Source Code Design

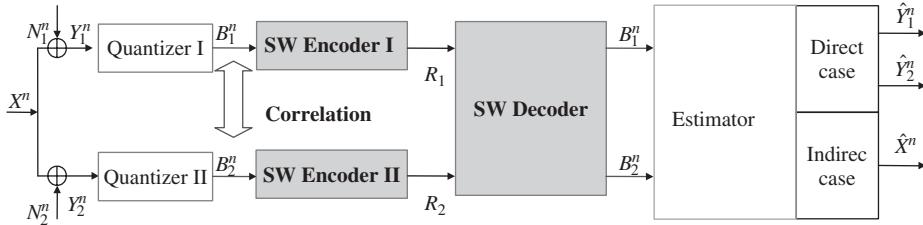
Like WZ coding, MT source coding is also a joint source–channel coding problem: first, its lossy nature necessitates quantization of the sources; second, the distributed nature of the encoders calls for compression (after quantization) by SW coding, which is commonly implemented by a channel code. More importantly, one of the conclusions of the theoretical works of [10, 26] is that VQ plus SW coding is indeed optimal for the quadratic Gaussian MT source coding with two terminals.<sup>5</sup> Following this guiding principle, Yang et al. [18] employed SW-coded quantization (SWCQ) for practical MT source coding. A generic block diagram of SWCQ for MT source coding is depicted in Figure 19.14. Unlike nested lattice codes suggested by Zamir et al. [53] and generalized coset codes used by Pradhan and Ramchandran [61], which are essentially nested source–channel codes, SWCQ explicitly separates the SW coding component from the vector quantizers at the encoder (while employing joint estimation/reconstruction at the decoder). SWCQ not only allows us to design a good source code and a good channel code individually, but also enables us to evaluate the practical performance loss due to source coding and channel coding separately. Moreover, SWCQ is very general as it applies to both direct and indirect MT source coding problems. It also generalizes similar approaches developed in [16, 17] for WZ coding.

More specifically, by combining TCQ with asymmetric and symmetric SW coding, respectively, the work of [18] presents two practical designs under the SWCQ framework for both direct and indirect quadratic Gaussian MT source coding with two encoders. The first *asymmetric SWCQ* scheme employs TCQ, asymmetric SW coding, and source splitting [62] to realize MT source coding with two encoders. It

<sup>5</sup>We point out that separate VQ and SW coding is in general not optimal for MT source coding.



**Figure 19.13** WZ coding results [17] based on TCQ and SW coding. At high rate, ideal SW-coded TCQ (with 8192 states) performs 0.1 dB away from the theoretical limit. Results with practical SW coding based on irregular LDPC codes are also included. (Left) 8192-state TCQ plus SW coding. (Right) 4D TCVQ plus SW coding.



**Figure 19.14** Block diagram of SWCQ for MT source code design.

is “split” into one classic source coding component and two WZ coding components. While classic source coding relies on entropy-coded VQ, WZ coding is implemented by combining TCQ and turbo/LDPC codes (for asymmetric SW coding). In the second *symmetric SWCQ* scheme of [18], the outputs of two TCQs are compressed using symmetric SW coding, which is based on the concept of channel code partitioning [15] for arbitrary rate allocation between the two encoders. A multilevel channel coding framework for symmetric SW coding is developed to exploit the joint statistics of the quantized sources. Furthermore, arithmetic coding is employed at each encoder to exploit the cross-bit-plane correlation in each of the quantized sources for further compression.

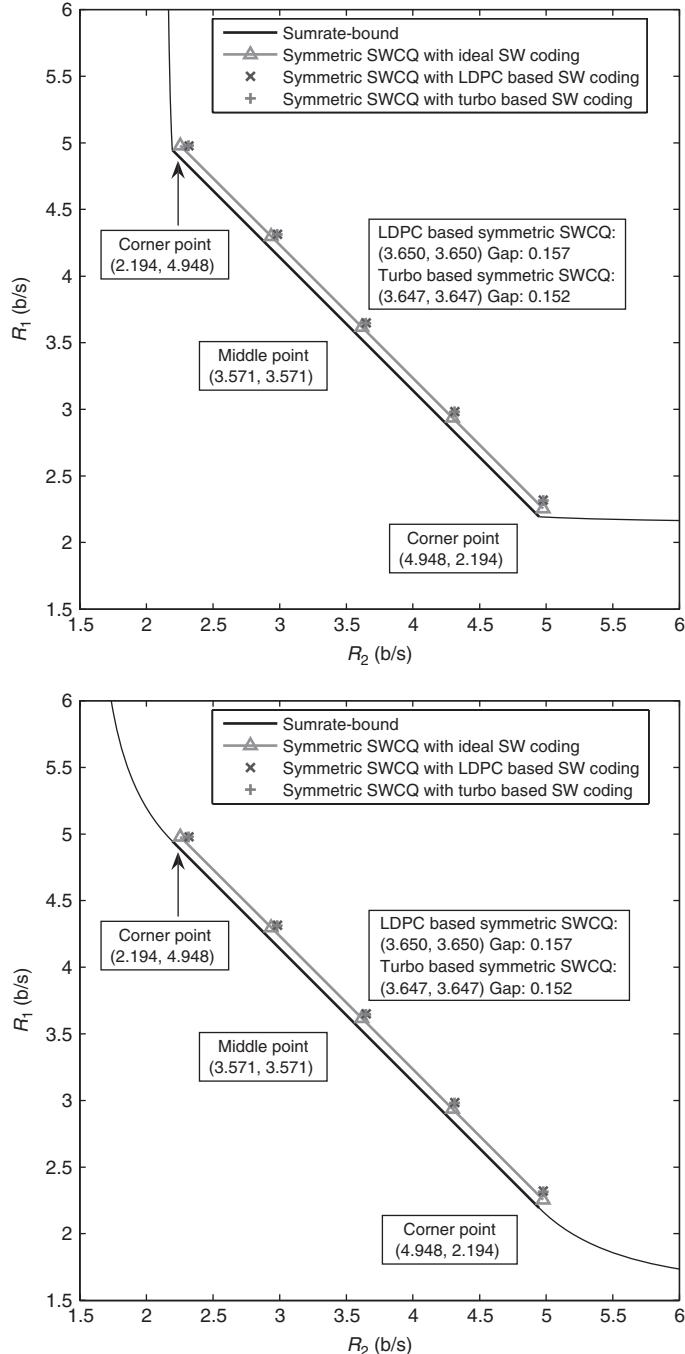
Compared to asymmetric SWCQ that involves source splitting, symmetric SWCQ is conceptually simpler because it only has one quantization step and one SW coding step and more elegant because all compression is done in one step that includes both classic entropy coding and syndrome-based channel coding for compression. However, in practice, it is easier to design longer turbo/LDPC codes for asymmetric SW coding than for symmetric SW coding.

To demonstrate the effectiveness of SWCQ, it is shown that, assuming ideal source coding and ideal SW coding (realized, e.g., via capacity-achieving channel coding), both asymmetric SWCQ and symmetric SWCQ can achieve *any* point on the sum-rate bound of the rate region for both direct and indirect MT source coding. High-rate performance analysis of SWCQ under practical TCQ and ideal SW coding is also given in [18]. Practical designs using TCQ and turbo/LDPC codes for asymmetric SW coding, and TCQ, arithmetic coding, and turbo/LDPC code for symmetric SW coding perform only 0.139–0.194 b/s away from the sum-rate bounds of quadratic Gaussian MT source coding. See Figure 19.15.

For MT source coding with more than two terminals, although the rate region for the general quadratic Gaussian setup is still unknown, it was also shown in [10] that the Berger–Tung sum-rate in the case with symmetric Gaussian sources is tight. This makes code design in this special case a very interesting research topic, for example, via classic source coding at the first encoder and WZ coding sequentially at other encoders.

## 19.4 APPLICATIONS

So far we have been focusing on the theory of DSC and code designs devised in recent years. The driving force behind recent theoretical progress [10] on DSC has been the applications of DSC in MT communication networks (e.g., distributed sensor



**Figure 19.15** Results of symmetric SWCQ [18] with TCQ and turbo/LDPC-based SW coding for the direct and indirect MT problems. The corner point with practical LDPC-based SW coding is (2.320, 4.979) b/s, with a total sum-rate loss of 0.157 b/s. The corner point with practical turbo-based SW coding is (2.315, 4.979) b/s, with a total sum-rate loss of 0.152 b/s. (Left) Direct MT:  $D_1^* = D_2^* = -30$  dB and  $\rho = 0.99$ . (Right) Indirect MT:  $D^* = -22.58$  dB and  $\sigma_{n_1}^2 = \sigma_{n_2}^2 = 1/99$ .

networks and ad hoc wireless networks). The fact that we are capable of designing limit-approaching SW, WZ, and MT source codes nowadays (at least for certain scenarios) has made these potential applications much closer to reality. It is expected that the recent flurry of research activities on DSC will only intensify in the future. In the sequel, we briefly highlight application areas of DSC, especially the new and exciting field of distributed video coding that subsumes both WZ video coding and MT video coding.

#### 19.4.1 Slepian–Wolf Coding for Lossless MT Networks

From Section 19.2.1, we see that SW coding captures the essence of the compression problem in lossless MT networks<sup>6</sup>—in fact, “MT source coding theory can be said to have been launched by the award-winning paper by Slepian and Wolf” [3]. By extending the approach of channel code partitioning to symmetric SW code design, Stanković et al. [15] additionally provided code designs for general lossless MT source coding networks [63, 64] with arbitrary number of encoders and decoders. The main idea is to split the MT network into subnetworks, each being an SW coding system with multiple sources [20], to design a code for such a subnetwork, and then to combine these codes into a single general MT code. The code designs obtained in this way are capable of approaching the theoretical limits in some special cases (e.g., the simple network example given in [27]).

Partitioning the network into SW subnetworks has some additional practical advantages, especially in large networks [65]. First, if a node fails, then only the traffic in the subnetwork is affected and not the decoding of all network nodes’ compressed transmission. Second, it is difficult to collect correlation statistics among all nodes in a network to determine the amount of SW compression each node should perform. This task becomes easier if only the correlation among nodes in a small network neighborhood is considered. Other network optimization issues, such as the network transmission structure [65], can also be considered jointly with SW coding and partitioning into SW subnetworks for overall optimized performance in a large network.

#### 19.4.2 Slepian–Wolf Coding for Secure Biometrics

Slepian–Wolf coding has also been proposed in [66] for use in secure biometrics. The idea is to protect biometric (e.g., iris) data by enrolling (or releasing as a public key) the syndrome bits with respect to an LDPC code of a private key sequence extracted from an image of a designated user’s eye. Authentication is successful only when the private key extracted from an image of a user’s eye gives the same syndrome bits (or public key). This work is underpinned by the information-theoretic studies on cryptography [67] and secrecy capacities [68] of multiple terminals.

#### 19.4.3 Wyner–Ziv Video Coding

Today’s standard techniques (e.g., MPEG-4 [69] and H.264 [70]) for video compression are developed for “downlink” broadcast applications with one heavy encoder

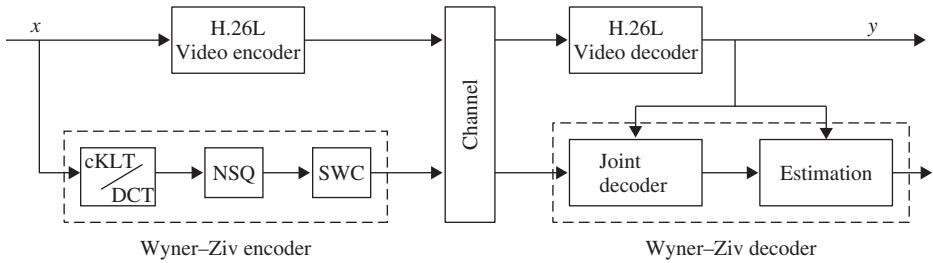
<sup>6</sup>From Sections 19.3.2 and 19.3.3, we observe that SW coding plays an important role in both WZ coding and MT source coding as well. This is analogous to classic source coding, where lossless entropy coding is an integral part of lossy source coding. Thus, the role of SW coding cannot be overemphasized in DSC.

and multiple light decoders. The growing popularity of video sensor networks, video cellular phones, and Web cameras has generated the need for low-complexity and power-efficient multimedia systems that can handle multiple video input and output streams. For such applications, we need a video coding system with multiple low-complexity encoders and one (or more) high-complexity decoders. In addition, the system must be robust to channel errors (e.g. wireless channels are error prone) so that the decoder at the base station can recover the scene with high fidelity using all received bit streams.

Distributed source coding provides a promising technique for “uplink” applications, and several groups have recently explored video compression based on DSC principles. One approach targets emerging applications (e.g., uplink video communications from handheld devices) that demand low encoding complexity—a scenario that is the opposite of video broadcast for which standard coders with heavy encoding are designed. Puri and Ramchandran proposed a coder in [71] that attempts to swap the encoder–decoder complexity of standard coders. Their encoder consists of the DCT, uniform quantization, and trellis coding for SW compression, while their decoder performs heavy-duty motion estimation. Girod et al. [19] also investigated distributed video coding using a relatively low-complexity turbo-code-based SW encoder. Whereas both coders perform better than independent intraframe (e.g., H.264 Intra) coding with the lowest encoding complexity, they suffer substantial R-D penalty when compared to H.264 Inter coding [70] with high encoding complexity (mainly due to motion estimation). The European Union’s DISCOVER project team [72] on distributed video coding reported results that are about 2/1 dB better than H.264 Intra with/without channel feedback. The latest distributed video coder from IBM Research [73] performs about 3.2 dB better than H.264 Intra and roughly 2 dB worse than H.264 Inter. Thus, there is still a relatively large gap between what WZC or DSC theory promises (to the extent of no performance loss in certain special cases when compared to joint encoding) and what practical low-complexity distributed video coders can achieve.

Another approach is to deemphasize low-complexity encoding while focusing on error robust WZ video coding. Video coding standards such as MPEG-4 [69] and H.264 [70] perform well under perfect network conditions but do not support rate scalability or provide error robustness over packet losses. This is due to the DPCM paradigm that underlies standard video coding, where packet losses often cause the encoder and the decoder to lose sync, resulting in error drifting/propagation with severe degradation of the video quality. For example, Sehgal et al. [74] discussed how coset-based WZ video coding can be used to alleviate prediction mismatch in DPCM-based standard video coders. Their coder is “state free” in the sense that the decoder does not have to maintain the same states as the encoder. Girod et al. [19] presented a robust video transmission system by using WZC to generate parity bits for protecting an MPEG-encoded bit stream of the same video.

Xu et al. worked on layered WZ video coding for robust video delivery. In [60], a novel *layered* video coder based on standard video coding and successive WZC [35, 58] was presented. Treating a standard coded video as the base layer (or side information), a layered WZ bit stream of the original video sequence is generated to enhance the base layer such that it is still decodable with commensurate qualities at rates corresponding to layer boundaries. The block diagram of the layered WZ coding scheme of [60] is shown in Figure 19.16.



**Figure 19.16** Block diagram of layered Wyner-Ziv video coding.

From Figure 19.16, we see that layered WZ coding is very much like MPEG-4/H.26L FGS coding [75, 76] in “spirit” in terms of having an embedded enhancement layer with good R-D performance. However, the key difference is that the enhancement layer is generated “blindly” without knowing the base layer in WZ coding, whereas FGS coding makes the assumption that the base layer can be delivered error free with strong forward error correction (FEC). This new approach thus avoids the problems (e.g., error drifting/propagation) associated with encoder–decoder mismatch in standard DPCM-based coders, leading to better error robustness. See Figure 19.17.

Xu et al.’s follow-up works [77, 78] consider layered WZ video coding and digital fountain codes [79] for receiver-driven layered multicast and distributed source–channel coding of video using Raptor codes [80].

#### 19.4.4 Wyner–Ziv Coding for Compress–Forward Relaying and Receiver Cooperation

Performance of wireless ad hoc networks is limited by the available resources (e.g., bandwidth and power). To save bandwidth and/or power, the key is to allow cooperation between network nodes or cooperative diversity [81]. The idea of

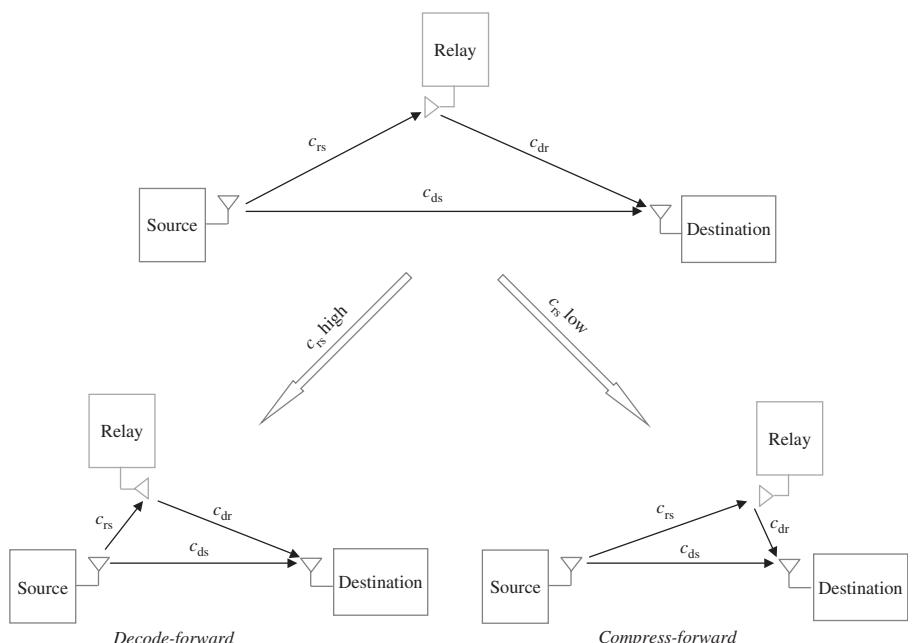


**Figure 19.17** Error robustness performance of layered WZ video coding [74] compared with H.26L FGS for football when both the base layer and enhancement layer bit streams are protected with 20% RS-based FEC and transmitted over a simulated CDMA2000 1X channel with 6% PDU loss rate. The 10th decoded frame by (a) H.26L FGS and (b) layered WZ video coding in the 7th simulated transmission.

cooperative diversity is for nodes, which might each have only one antenna, to join together to code and transmit data or to receive and decode data, thereby operating in some way as a multiple antenna system. Since cooperative diversity is largely based on relaying messages, its information-theoretic foundation is built upon the 1979 work of Cover and El Gamal [82] on capacity bounds for relay channels.

Although the capacity of the general relay channel is not known, several coding schemes have been proposed to obtain bounds on the achievable rate. These schemes can be classified into decode-forward and observe-forward. The simplest observe-forward scheme is amplify-forward, and a more sophisticated scheme is compress-forward, which is rooted in [82], where the relay employs WZ coding to compress the signal it has received from the source within certain distortion. Decode-forward works better when the relay is close to the destination. Compress-forward has higher computational complexity than decode-forward, but it gives many rate points that are not achievable with any other coding strategies, and it provides the best solution when the relay is close to the destination. See Figure 19.18.

Liu et al. [83] studied compress-forward coding with BPSK modulation for the half-duplex Gaussian relay channel. Lower and upper performance bounds and a practical code design based on WZ coding are given. Simulation results show that, by using LDPC codes for error protection at the source and nested scalar quantization and IRA codes for WZ coding (or more precisely, distributed joint source–channel coding) at the relay, the practical implementation comes within 1.6–3.03 dB away from the upper bound of compress-forward coding in terms of the source transmission power.



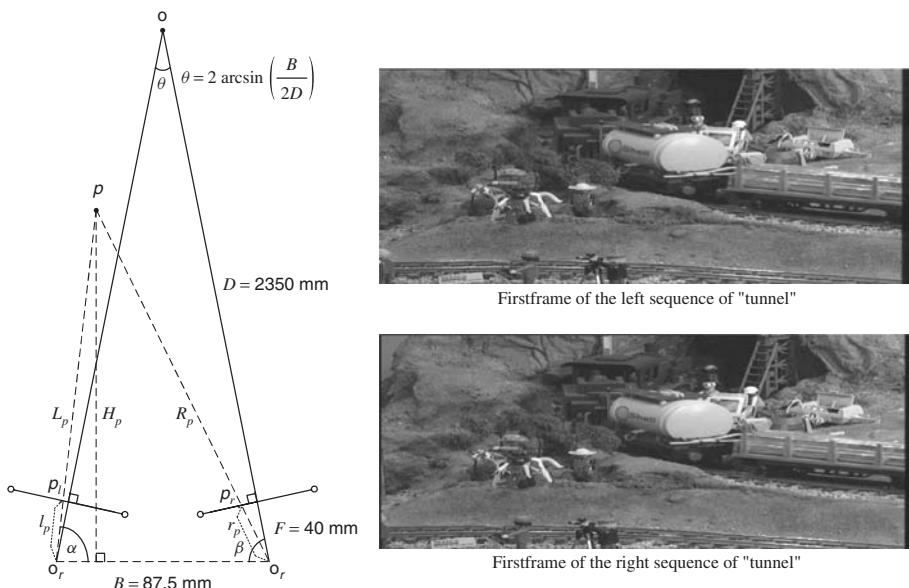
**Figure 19.18** For the relay channel, decode-forward works better when the relay is close to the source, but compress-forward is preferred when the relay is close to the destination since the relay can employ WZ coding to exploit the correlation between the signals received at the relay and the destination.

Cooperative diversity naturally arises in ad hoc networks as it enables great power savings with cheap, simple, and mobile nodes, while supporting decentralized routing and control algorithms.

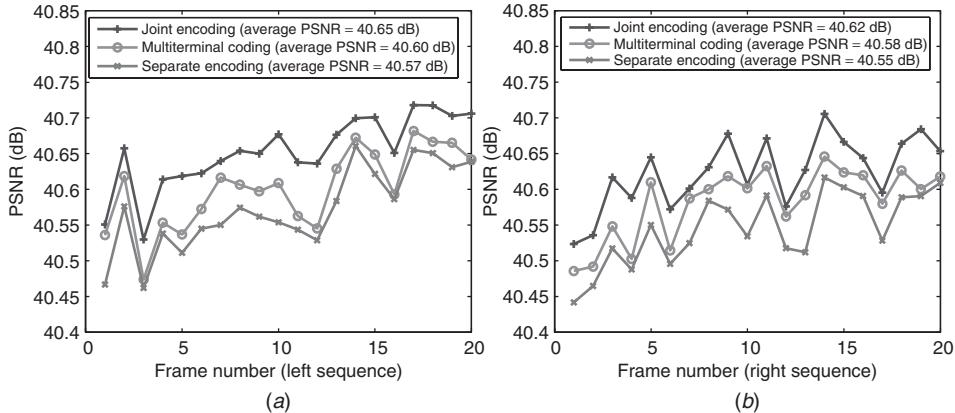
The simplest nontrivial setup is when the nodes form pairs, that is, clusters of two. In a two-transmitter two-receiver cooperative channel, the two single-antenna transmitters want to communicate messages to the two remote single-antenna receivers over the same wireless radio channel. In receiver cooperation, two (closely located) single-antenna receivers cooperate to facilitate decoding messages from two remote single-antenna transmitters. Since this cooperative channel can be viewed as a combination of the interference channel and the relay channel, its best achievable rate regions are obtained by combining decode-forward or compress-forward coding techniques for the relay channel with coding for the interference channel [84]. Because the distance between the two receivers is expected to be much smaller than that between a transmitter and a receiver, compress-forward with WZ coding provides the highest achievable rates and is shown in [85] to asymptotically achieve the capacity as the interference and signal-to-noise ratio (SNR) approach infinity. Preliminary work on code designs for receiver cooperation (and transmitter cooperation) appeared in [86].

#### 19.4.5 Multiterminal Video Coding

Following recent works [10, 18] on the rate region of the quadratic Gaussian two-terminal source coding problem and limit-approaching code designs, Yang et al. [87] examined MT source coding of two correlated video sequences captured by calibrated cameras as shown in Figure 19.19 to save the sum rate over independent coding. The key issue is correlation modeling when dealing with practical video sources.



**Figure 19.19** Three-dimensional camera settings (left) and first pair of frames (right) from tunnel sequences: top-right is the left first frame, and bottom-right is the right first frame.



**Figure 19.20** Comparison (in terms of PSNR vs. frame number) [87] between separate H.264 encoding, asymmetric MT video coding, and joint encoding of the stereo tunnel sequences (at the same sum rate of 6.58 Mbps): (a) left sequence and (b) right sequence.

Two MT video coding schemes are proposed in [87]. In the *asymmetric scheme*, the first video sequence is coded by H.264 and used at the joint decoder to facilitate WZ coding of the second video sequence. The first I-frame of the right sequence is successively coded by H.264 and SW coding. An efficient stereo-matching algorithm based on loopy belief propagation is then adopted at the decoder to produce pixel-level disparity maps between the corresponding frames of the two decoded video sequences on the fly. Based on the disparity maps, side information for both motion vectors and motion-compensated residual frames of the second sequence are generated at the decoder before WZ encoding. In the *symmetric scheme*, source splitting is employed for compression of both I-frames to allow flexible rate allocation between the two sequences. Experimental results of both schemes on stereo video sequences using H.264, LDPC codes for SW coding of the motion vectors and scalar quantization in conjunction with LDPC codes for WZ coding of the residual coefficients show better video quality when compared to separate H.264 coding at the same sum rate. Figure 19.20 compares separate H.264 encoding, asymmetric MT video coding, and joint encoding of the stereo tunnel sequences in terms of PSNR versus frame number (at the same sum rate of 6.58 Mbps).

## 19.5 CONCLUSIONS

In this chapter, we have reviewed more than 30 years of DSC theory, surveyed recently developed limit-approaching code designs, and highlighted applications of DSC in secure biometrics, lossless compression in MT communications networks, receiver cooperation in cooperative networks, and more importantly, distributed video coding. Along the way, we also pointed out some challenging theoretical and practical issues that need to be addressed before DSC can “take off” and have real impact in practice. For example, one of the most difficult problems is universal SW coding. Unless this problem is adequately addressed, the gap will still exist between the current DSC theory, which assumes full knowledge of the source correlation, and practice, where

the joint statistics is often not known a priori and usually time varying. In addition, we only focused on DSC here, to make its applications in sensor networks practical, many other issues, for example, cross-layer design and node synchronization, have to be addressed. We hope this chapter will serve as an introductory material in getting more researchers interested in DSC so that they can make their original contributions to the field.

## REFERENCES

1. D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, pp. 471–480, July 1973.
2. A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 22, pp. 1–10, Jan. 1976.
3. T. Berger, "Multiterminal source coding," in *The Information Theory Approach to Communications*, G. Longo (Ed.), New York: Springer-Verlag, 1977.
4. S. Tung, "Multiterminal rate-distortion theory," PhD dissertation, School of Electrical Engineering, Cornell University, Ithaca, NY, 1978.
5. H. Yamamoto and K. Itoh, "Source coding theory for multiterminal communication systems with a remote source," *Trans. IECE Jpn.*, vol. E63, pp. 700–706, Oct. 1980.
6. T. Flynn and R. Gray, "Encoding of correlated observations," *IEEE Trans. Inform. Theory*, vol. 33, pp. 773–787, Nov. 1987.
7. T. Berger, Z. Zhang, and H. Viswanathan, "The CEO problem," *IEEE Trans. Inform. Theory*, vol. 42, pp. 887–902, May 1996.
8. Y. Oohama, "The rate-distortion function for the quadratic Gaussian CEO problem," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1057–1070, May 1998.
9. Y. Yang and Z. Xiong, "The supremum sum-rate loss of quadratic Gaussian direct multiterminal source coding," in *Proc. UCSD Workshop on Information Theory and Its Applications*, San Diego, CA, Jan. 2008.
10. A. Wagner, S. Tavildar, and P. Viswanath, "The rate region of the quadratic Gaussian two-terminal source-coding problem," *IEEE Trans. Inform. Theory*, vol. 54, 2008.
11. C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.
12. R. Gallager, *Low Density Parity Check Codes*, Cambridge, MA: MIT Press, 1963.
13. D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999.
14. A. Liveris, Z. Xiong and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Lett.*, vol. 6, pp. 440–442, Oct. 2002.
15. V. Stanković, A. Liveris, Z. Xiong, and C. Georghiades, "On code design for the general Slepian-Wolf problem and for lossless multiterminal communication networks," *IEEE Trans. Inform. Theory*, vol. 52, pp. 1495–1507, Apr. 2006.
16. Z. Liu, S. Cheng, A. Liveris, and Z. Xiong, "Slepian-Wolf coded nested lattice quantization for Wyner-Ziv coding: High-rate performance analysis and code design," *IEEE Trans. Inform. Theory*, vol. 52, pp. 4358–4379, Oct. 2006.
17. Y. Yang, S. Cheng, Z. Xiong, and W. Zhao, "Wyner-Ziv coding based on TCQ and LDPC codes," *IEEE Trans. Communications*, vol. 57, pp. 376–387, February 2009.
18. Y. Yang, V. Stankovic, Z. Xiong, and W. Zhao, "On multiterminal source code design," *IEEE Trans. Inform. Theory*, vol. 54, pp. 2278–2302, May 2008.

19. B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, pp. 71–83, Jan. 2005.
20. T. Cover and J. Thomas, *Elements of Information Theory*, New York: Wiley, 1991.
21. T. Cover, "A proof of the data compression theorem of Slepian and Wolf for ergodic sources," *IEEE Trans. Inform. Theory*, vol. 22, pp. 226–228, Mar. 1975.
22. A. Wyner, "The rate-distortion function for source coding with side information at the decoder—II: General sources," *Inform. Control*, vol. 38, pp. 60–80, 1978.
23. R. Zamir, "The rate loss in the Wyner-Ziv problem," *IEEE Trans. Inform. Theory*, vol. 42, pp. 2073–2084, Nov. 1996.
24. S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1181–1203, May 2003.
25. Y. Oohama, "Gaussian multiterminal source coding," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1912–1923, Nov. 1997.
26. Y. Oohama, "Rate-distortion theory for Gaussian multiterminal source coding systems with several side informations at the decoder," *IEEE Trans. Inform. Theory*, vol. 51, pp. 2577–2593, July 2005.
27. A. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inform. Theory*, vol. 20, pp. 2–10, Jan. 1974.
28. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Trans. Inform. Theory*, vol. 49, pp. 626–643, Mar. 2003.
29. J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Commun. Lett.*, vol. 5, pp. 417–419, Oct. 2001.
30. J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," in *Proc. Globecom'01*, San Antonio, TX, Nov. 2001.
31. A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. DCC'02*, Snowbird, UT, Apr. 2002.
32. A. Liveris, Z. Xiong, and C. Georghiades, "Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes," in *Proc. DCC'03*, Snowbird, UT, Mar. 2003.
33. D. Van Renterghem, X. Jaspar, B. Macq, and L. Vandendorpe "Distributed source coding with optimized irregular turbo codes," in *Proc. ICC'07*, Glasgow, Scotland, June 2007.
34. T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.
35. S. Cheng and Z. Xiong, "Successive refinement for the Wyner-Ziv problem and layered code design," *IEEE Trans. Signal Process.*, vol. 53, pp. 3269–3281, Aug. 2005.
36. N. Gehrig and P. Dragotti, "Symmetric and asymmetric Slepian-Wolf codes with systematic and nonsystematic linear codes," *IEEE Commun. Lett.*, vol. 9, pp. 61–63, Jan. 2005.
37. M. Sartipi and F. Fekri, "Distributed source coding using short to moderate length rate-compatible LDPC codes: The entire Slepian-Wolf rate region," *IEEE Trans. Commun.*, vol. 56, pp. 400–411, Mar. 2008.
38. A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Joint source-channel coding of binary sources with side information at the decoder using IRA codes," in *Proc. MMSP'02*, St. Thomas, U.S. Virgin Islands, Dec. 2002.
39. J. Garcia-Frias, Y. Zhao, and W. Zhong, "Turbo-like codes for transmission of correlated sources over noisy channels," *IEEE Signal Process. Mag.*, vol. 24, pp. 58–66, Sept. 2007.
40. D. Schonberg, K. Ramchandran, and S. S. Pradhan, "Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources," in *Proc. DCC'04*, Snowbird, UT, Mar. 2004.

41. M. Fresia, L. Vandendorpe, and H. V. Poor “Distributed source coding using Raptor codes for hidden Markov sources,” in *Proc. DCC’08*, Snowbird, UT, Mar. 2008.
42. A. Liveris, C. Lan, K. Narayanan, Z. Xiong, and C. Georghiades, “Slepian-Wolf coding of three binary sources using LDPC codes,” in *Proc. Intl. Symp. Turbo Codes and Related Topics*, Brest, France, Sept. 2003.
43. C. Lan, A. Liveris, K. Narayanan, Z. Xiong, and C. Georghiades, “Slepian-Wolf coding of multiple  $M$ -ary sources using LDPC codes,” in *Proc. DCC’04*, Snowbird, UT, Mar. 2004.
44. Y. Zhao and J. Garcia-Frias, “Data compression of correlated non-binary sources using punctured turbo codes,” in *Proc. DCC’02*, Snowbird, UT, Apr. 2002.
45. J. Garcia-Frias and W. Zhong, “LDPC codes for compression of multiterminal sources with hidden Markov correlation,” *IEEE Commun. Lett.*, pp. 115–117, Mar. 2003.
46. A. Eckford and W. Yu, “Rateless Slepian-Wolf codes,” in *Proc. Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2005.
47. B. Ndzana, A. Shokrollahi, and J. Abel, “Fountain codes for the Slepian-Wolf problem,” in *Proc. Allerton’06*, Monticello, IL, Sept. 2006.
48. J. Jiang, D. He, and A. Jagmohan, “Rateless Slepian-Wolf coding based on rate adaptive LDPC codes,” in *Proc. ISIT’07*, Nice, France, June 2007.
49. G. Caire, S. Shamai, and S. Verdu, “Lossless data compression with low-density parity-check codes,” in *Multiantenna Channels: Capacity, Coding and Signal Processing*, G. Foschini and S. Verdu (Eds.), Providence, RI: American Mathematical Society, 2003.
50. M. Marcellin and T. Fischer, “Trellis coded quantization of memoryless and Gaussian-Markov sources,” *IEEE Trans. Commun.*, vol. 38, pp. 82–93, Jan. 1990.
51. M. Eyuboglu and D. Forney, Jr., “Lattice and trellis quantization with lattice- and trellis-bounded codebooks—high-rate theory for memoryless sources,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 46–59, Jan. 1993.
52. S. Shamai, S. Verdu, and R. Zamir, “Systematic lossy source/channel coding,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 564–579, Mar. 1998.
53. R. Zamir, S. Shamai, and U. Erez, “Nested linear/lattice codes for structured multiterminal binning,” *IEEE Trans. Inform. Theory*, vol. 48, pp. 1250–1276, June 2002.
54. A. Liveris, Z. Xiong, and C. Georghiades, “Nested turbo codes for the binary Wyner-Ziv problem,” in *Proc. ICIP’03*, Barcelona, Spain, Sept. 2003.
55. X. Wang and M. T. Orchard, “Design of trellis codes for source coding with side information at the decoder,” in *Proc. DCC’01*, Snowbird, UT, Mar. 2001.
56. J. Chou, S. Pradhan, and K. Ramchandran, “Turbo and trellis-based constructions for source coding with side information,” in *Proc. DCC’03*, Snowbird, UT, Mar. 2003.
57. D. Rebollo-Monedero, S. Rane, A. Aaron, and B. Girod, “High-rate quantization and transform coding with side information at the decoder,” *Signal Process.*, vol. 86, pp. 3123–3130, Nov. 2006.
58. Y. Steinberg and N. Merhav, “On successive refinement for the Wyner-Ziv problem,” *IEEE Trans. Inform. Theory*, vol. 50, pp. 1636–1654, Aug. 2004.
59. W. Equitz and T. Cover, “Successive refinement of information,” *IEEE Trans. Inform. Theory*, vol. 37, pp. 269–274, Mar. 1991.
60. Q. Xu and Z. Xiong, “Layered Wyner-Ziv video coding,” *IEEE Trans. Image Process.*, vol., pp. 3791–3803, Dec. 2006.
61. S. Pradhan and K. Ramchandran, “Generalized coset codes for distributed binning,” *IEEE Trans. Inform. Theory*, vol. 51, pp. 3457–3474, Oct. 2005.
62. B. Rimoldi and R. Urbanke, “Asynchronous Slepian-Wolf coding via source-splitting,” in *Proc. ISIT’97*, Ulm, Germany, June 1997, p. 271.

63. I. Csiszar and J. Korner, "Towards a general theory of source networks," *IEEE Trans. Inform. Theory*, vol. 26, pp. 155–165, Mar. 1980.
64. T. Han and K. Kobayashi, "A unified achievable rate region for a general class of multi-terminal source coding systems," *IEEE Trans. Inform. Theory*, vol. 26, pp. 277–288, May 1980.
65. R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "Networked Slepian-Wolf: Theory, algorithms, and scaling laws," *IEEE Trans. Inform. Theory*, vol. 51, pp. 4057–4073, Dec. 2005.
66. E. Martinian, S. Yekhanin, and J. Yedidia, "Secure biometrics via syndromes," in *Proc. Allerton'05*, Monticello, IL, Oct. 2005.
67. R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography II: CR capacity," *IEEE Trans. Inform. Theory*, vol. 44, pp. 225–240, Jan. 1998.
68. I. Csiszar and P. Narayan, "Secrecy capacities for multiple terminals," *IEEE Trans. Inform. Theory*, vol. 50, pp. 3047–3061, Dec. 2004.
69. MPEG-4 Video VM, ver. 13.0, ISO/IEC JTC 1/SC29/WG11 N2687, Mar. 1999.
70. T. Wiegand, G. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 13, pp. 560–576, July 2003.
71. R. Puri, A. Majumdar, and K. Ramchandran, "PRISM: A video coding paradigm with motion estimation at the decoder," *IEEE Trans. Image Process.*, vol. 16, pp. 2436–2448, Oct. 2007.
72. C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, and J. Ostermann, "Distributed monoview and multiview video coding," *IEEE Signal Process. Mag.*, vol. 24, pp. 67–76, Sept. 2007.
73. D. He, A. Jagmohan, L. Lu, and V. Sheinin, "Wyner-Ziv video compression using rateless LDPC codes," in *Proc. VCIP'08*, San Jose, CA, Jan. 2008.
74. A. Sehgal, A. Jagmohan, and N. Ahuja, "Wyner-Ziv coding of video: Applications to error resilience," *IEEE Trans. Multimedia*, vol. 6, pp. 249–258, Apr. 2004.
75. W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 11, pp. 301–317, Mar. 2001.
76. Y. He, R. Yan, F. Wu, and S. Li, "H.26L-based fine granularity scalable video coding," ISO/IEC MPEG 58th meeting, M7788, Pattaya, Thailand, Dec. 2001.
77. Q. Xu, V. Stanković, and Z. Xiong, "Wyner-Ziv video compression and fountain codes for receiver-driven layered multicast," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 17, pp. 901–906, July 2007.
78. Q. Xu, V. Stanković, and Z. Xiong, "Distributed source-channel coding of video using Raptor codes," *IEEE JSAC*, vol. 25, pp. 851–861, May 2007.
79. M. G. Luby, "LT codes," in *Proc. 43rd IEEE Symp. the Foundations of Computer Science*, Vancouver, BC, Canada, Nov. 2002, pp. 271–280.
80. A. Shokrollahi, "Raptor codes," *IEEE Trans. Inform. Theory*, vol. 52, pp. 2551–2567, June 2006.
81. *Special Issue on Models, Theory, and Codes for Relaying and Cooperation in Communication Networks*, *IEEE Trans. Inform. Theory*, vol. 53, Oct. 2007.
82. T. Cover and A. El Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inform. Theory*, vol. 25, pp. 572–584, Sept. 1979.
83. Z. Liu, V. Stankovic, and Z. Xiong, "Wyner-Ziv coding for the half-duplex relay channel," in *Proc. ICASSP'05*, Philadelphia, PA, Mar. 2005.
84. A. Carleial, "Interference channels," *IEEE Trans. Inform. Theory*, vol. 24, pp. 60–70, Jan. 1978.

85. A. Høst-Madsen, "Capacity bounds for cooperative diversity," *IEEE Trans. Inform. Theory*, vol. 52, pp. 1522–1544, Apr. 2006.
86. M. Uppal, Z. Liu, V. Stanković, A. Høst-Madsen, and Z. Xiong, "Capacity bounds and code designs for cooperative diversity," in *Proc. UCSD Workshop on Information Theory and Its Applications*, San Diego, CA, Feb. 2006.
87. Y. Yang, V. Stanković, Z. Xiong, and W. Zhao, "Two-terminal video coding," *IEEE Trans. Image Processing*, vol. 18, pp. 534–551, March 2009.



---

## CHAPTER 20

---

# Network Coding for Sensor Networks

Christina Fragouli

Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

## 20.1 INTRODUCTION

Network coding is a new area that promises to revolutionize the way we treat information in a network and have a deep impact on all network functionalities, such as routing, network storage, and network design. Four monographs and a book have recently been published [1–5] on this subject as well as a number of tutorial articles [6–9]. This chapter explores the application of network coding in sensor networks.

Network coding deals with a very fundamental principle, namely, how we treat information flow; it is thus no surprise that it has reached and offers benefits for a wide range of applications, such as peer-to-peer networks, router design, chip design, distributed storage, network security, and network tomography (for a review of such applications see, e.g., [4]). It is this author’s belief, however, that ad hoc wireless sensor networks is an area where network coding can have an immediate impact for the following reasons:

- This is an environment that is currently in the design stage and thus has the flexibility to accommodate new protocols.
- Network coding requires intermediate nodes to perform some sort of packet processing. In sensor networks, node processing has always been advocated, albeit for different reasons: for example, for information aggregation. Thus, unlike, for example, Internet routers, where packet processing would need the addition of new functionalities, and thus introducing network coding capabilities would be a more long-term investment, in sensor nodes these functionalities are already in place or very easily added.
- Sensor networks offer a challenging environment due to the inherent challenges of the wireless medium and the ad hoc structure of the network that needs to be maintained. Additionally, sensor nodes are simple devices with very limited resources and often deployed in hard-to-reach environments where maintenance is too costly. Thus, design properties such as energy efficiency, load balancing, and robustness to node failures become not only desirable but of critical importance.

These are problems network coding promises to help with, thus motivating the further study and development of network coding techniques for sensor networks.

### 20.1.1 What Is Network Coding

The novel paradigm in network operation that network coding brings is that, instead of having individual source packets traversing a network, we have combinations of packets, each bringing some type of “evidence” about the source packets. These packet combinations are created throughout the network: We allow intermediate network nodes to process their incoming information packets and, in particular, combine them to create new packets. A receiver collects a sufficient number of such combined packets and uses them to retrieve the original information sent by the sources. The area of network coding is centered around the application of this basic idea of dealing with evidence instead of individual packets.

In the following two “classical” examples, used to illustrate network coding over wired and wireless networks, respectively, “xor” corresponds to addition over the binary field.

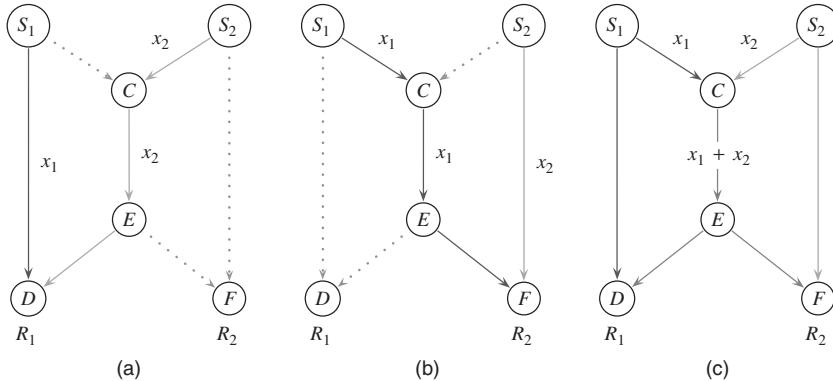
**Example 20.1** *The Butterfly Network* figure 20.1 depicts a communication network represented as a directed graph where vertices correspond to terminals and edges correspond to channels. This example is commonly known in the network coding literature as the butterfly network. Assume that we have slotted time, and that through each channel we can send one bit per time slot. We have two sources  $S_1$  and  $S_2$ , and two receivers  $R_1$  and  $R_2$ . Each source produces one bit per time slot, which we denote by  $x_1$  and  $x_2$ , respectively (unit rate sources).

If receiver  $R_1$  uses all the network resources by itself, it could receive both sources. Indeed, we could route the bit  $x_1$  from source  $S_1$  along the path  $\{AD\}$  and the bit  $x_2$  from source  $S_2$  along the path  $\{BC, CE, ED\}$ , as depicted in Figure 20.1a. Similarly, if the second receiver  $R_2$  uses all the network resources by itself, it could also receive both sources. We can route the bit  $x_1$  from source  $S_1$  along the path  $\{AC, CE, EF\}$ , and the bit  $x_2$  from source  $S_2$  along the path  $\{BF\}$  as depicted in Figure 20.1b.

Now assume that both receivers want to simultaneously receive the information from both sources. That is, we are interested in multicasting. We then have a “contention” for the use of edge  $CE$ , arising from the fact that through this edge we can only send one bit per time slot. However, we would like to simultaneously send bit  $x_1$  to reach receiver  $R_2$  and bit  $x_2$  to reach receiver  $R_1$ .

Traditionally, information flow was treated like fluid through pipes, and independent information flows were kept separate. Applying this approach, we would have to make a decision at edge  $CE$ : Either use it to send bit  $x_1$  or use it to send bit  $x_2$ . If, for example, we decide to send bit  $x_1$ , then receiver  $R_1$  will only receive  $x_1$ , while receiver  $R_2$  will receive both  $x_1$  and  $x_2$ .

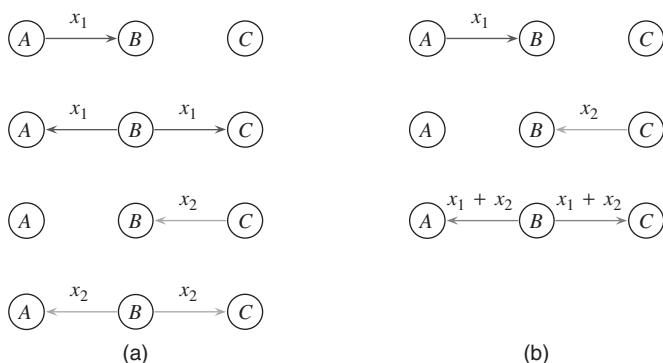
The simple but important observation made in the seminal work by Ahlswede, Cai, Li, and Yeung is that we can allow intermediate nodes in the network to process their incoming information streams and not just forward them. In particular, node  $C$  can take bits  $x_1$  and  $x_2$  and xor them to create a third bit  $x_3 = x_1 + x_2$ , which it can then send through edge  $CE$  (the xor operation corresponds to addition over the binary field).  $R_1$  receives  $\{x_1, x_1 + x_2\}$  and can solve this system of equations to retrieve  $x_1$  and  $x_2$ . Similarly,  $R_2$  receives  $\{x_2, x_1 + x_2\}$  and can solve this system of equations to retrieve  $x_1$  and  $x_2$ .



**Figure 20.1** Butterfly network. Sources  $S_1$  and  $S_2$  multicast their information to receivers  $R_1$  and  $R_2$ .  $S_1$  and  $S_2$  multicast to both  $R_1$  and  $R_2$ . All links have capacity 1. With network coding (by xor-ing the data on link  $CD$ ), the achievable rates are 2 for each source, the same as if every destination were using the network for its sole use. Without network coding, the achievable rates are less (e.g., if both rates are equal, the maximum rate is 1.5). (a) Routing to  $R_1$ , (b) routing to  $R_2$ , and (c) network coding.

The previous example shows that if we allow intermediate nodes in the network to combine information streams and extract the information at the receivers, we can increase the throughput when multicasting. It thus illustrates that network coding can offer throughput benefits when compared to routing. The next example shows that in a wireless environment, network coding can be used to offer benefits in terms of battery life, wireless bandwidth, and delay.

**Example 20.2** Consider a wireless ad hoc network, where devices  $A$  and  $C$  would like to exchange the binary files  $x_1$  and  $x_2$  using device  $B$  as a relay. We assume that time is slotted and that a device can either transmit or receive a file during a time slot (half-duplex communication). Figure 20.2 depicts on the left the standard approach: Nodes  $A$  and  $C$  send their files to relay  $B$ , which in turn forwards each file to the corresponding destination.



**Figure 20.2** Nodes  $A$  and  $B$  exchange information via relay  $B$ . The network coding approach uses one broadcast transmission less: (a) without and (b) with network coding.

*The network coding approach takes advantage of the natural capability of wireless channels for broadcasting to give benefits in terms of resource utilization, as illustrated in Figure 20.2.*

*In particular, node C receives both files  $x_1$  and  $x_2$ , and bitwise  $\text{xors}$  them to create the file  $x_1 + x_2$ , which it then broadcasts to both receivers using a common transmission. Node A has  $x_1$  and can thus decode  $x_2$ . Node C has  $x_2$  and can thus decode  $x_1$ .*

*This approach offers benefits in terms of energy efficiency (node B transmits once instead of twice), delay (the transmission is concluded after three instead of four time-slots), wireless bandwidth (the wireless channel is occupied for a smaller amount of time), and interference (if there are other wireless nodes attempting to communicate in the neighborhood). The benefits in the previous example arise from that broadcast transmissions are made maximally useful to all their receivers.*

*Note that  $x_1 + x_2$  is nothing but some type of binning or hashing for the pair  $(x_1, x_2)$  that the relay needs to transmit. Binning is not a new idea in wireless communications. The new element is that we can efficiently implement such ideas in practice, using simple algebraic operations.*

### 20.1.2 Network Coding for Sensor Networks: Chapter Overview

The application of network coding to sensor networks is very much in the research stage—some approaches have been proposed, but the theory and practice of the field are not mature.

If we wanted to extract a set of identifying features of sensor networks, we could perhaps highlight the following.

- *Dynamically Changing Network* We have a wireless network: As opposed to wireline, wireless implies a shared medium and time variability. A node by increasing its transmission power may reach every other node in the network, but it may at the same time create significant interference. Moreover, channels vary over time, due, for example, to fading or node mobility.
- *Decentralized Operation* The network organization is ad hoc, thus decentralized network operation and management is required.
- *Distributed Function Computation* In sensor networks, we often need to compute global functions of data distributed over the network with minimal coordination. For example, we may want to compute the average of all observed values.
- *Restricted Resources* Finally, sensor nodes are typically cheap simple devices, with restricted computational power and battery life and prone to failures.

How network coding interacts with each of these features will be the main axis around which we will attempt to spin this chapter. In this vein, we will sequentially examine a number of attributes network coding offers in network operation, translate them into sensor network functionalities and applications, and discuss how they interact with the previous features. We do not claim to exhaustively cover the literature, but we will attempt to give representative examples to illustrate the various network coding attributes. Throughout the chapter we will also attempt to identify open research problems and questions.

More specifically, we will discuss in Section 20.3 the coupon collector problem and how it is related to data gathering. Section 20.4 illustrates how network coding helps with distributed storage, using as an example a particular sensor network application. In Section 20.5 we will argue that network coding allows decentralized operation and again present a particular sensor network application that builds on this observation. Section 20.6 investigates how network coding allows to take advantage of the capabilities of broadcasting to increase the system reliability. Section 20.7 reviews joint networks, source, and channel coding, while Section 20.8 discusses how ideas from network coding can be useful in identity-aware sensor networks, where the bulk of the data to be transferred consists of the node identities as opposed to information data. Finally, Section 20.9 concludes with a brief discussion and open problems. Before proceeding in our main theme, we start by discussing, in Section 20.2, how we can implement network coding in a practical setting. The reader familiar with these concepts is encouraged to skip ahead.

## 20.2 HOW CAN WE IMPLEMENT NETWORK CODING IN A PRACTICAL SENSOR NETWORK?

In the two network coding examples we presented in Section 20.1.1, we implicitly assumed that there is synchronization between the network nodes, and each node performs fixed encoding operations. The receivers know these operations, and use this knowledge to decode. For example, in the butterfly network in Figure 20.1,  $x_1$  and  $x_2$  arrive simultaneously at node  $C$ . Node  $C$  always performs the same operation on these packets and forwards the resulting packet  $x_1 + x_2$  to node  $E$ . The receivers  $R_1$  and  $R_2$  know which linear combination their received packets correspond to. For example,  $R_1$  knows it receives  $x_1$  through edges  $AD$  and  $x_1 + x_2$  through edge  $ED$ .

In a practical sensor network, such assumptions are hard to implement. Synchronization is hard to maintain in a distributed setting. Moreover, the network structure changes quite often due to varying channel conditions, nodes moving, or nodes dying. Each network change implies that we need to redesign what linear combining operations network nodes do and, accordingly, inform the receivers. However, distributing information regarding the overall network structure and coding operations is costly. Thus clearly, network coding cannot be a viable solution unless it can be implemented in a decentralized manner.

Fortunately, three ideas, which appeared successively in time, give us an elegant and flexible way to perform network coding in a completely decentralized manner. These are:

1. Randomly chose the linear combinations at each network node [10].
2. Append “coding vectors” at the header of each packet to allow the receivers to decode without need of synchronization [11].
3. Use subspace coding to achieve the same goal as in idea 2 more efficiently [12].

The first idea determines what intermediate nodes in the network do. The second and third offer two alternative approaches for the encoding of the data at the sources and corresponding decoding at the receivers. We will discuss these ideas in subsequent sections, after first briefly discussing operations over finite fields.

### 20.2.1 Operation over Finite Field

To perform network coding, we need to employ linear operations—additions and multiplications—over finite fields. A finite field  $\mathbb{F}_q$  of size  $q$ , with  $q$  a prime number or a power of a prime number, contains  $q$  symbols. For example, the field  $\mathbb{F}_4$  has the symbols  $\{0, 1, 2, 3\}$ . We can do addition and multiplication over the finite field using tables of operations. There also exist algorithms that use shifts and additions to implement multiplication and addition more efficiently [13].

The network operates by exchanging and processing packets that contain symbols over a finite field  $\mathbb{F}_q$ . Using a field of size  $q = 2^m$ , and a packet of length  $L$  symbols over  $\mathbb{F}_q$ , simply means that our packet contains  $Lm$  bits. These bits are divided into  $L$  groups, each group consisting of  $m$  bits. Each group of  $m$  bits is treated as one symbol of  $\mathbb{F}_q$  and processed using operations over  $\mathbb{F}_q$  by the network nodes. For example, if each of the source packets  $x_1$  and  $x_2$  has length 5 symbols over  $\mathbb{F}_{2^8}$ , that is,

$$x_1 = [x_{11} \dots x_{15}], \quad x_2 = [x_{21} \dots x_{25}], \quad x_{ij} \in \mathbb{F}_{2^8},$$

then each of these packets has length 5 bytes, or 40 bits. Now assume a network node receives  $x_1$  and  $x_2$  and would like to create the linear combination  $2x_1 + 6x_2$ . It then performs the linear combining byte per byte. That is, it creates the packet

$$2x_1 + 6x_2 = [2x_{11} + 6x_{21} \dots 2x_{15} + 6x_{25}].$$

### 20.2.2 Randomized Network Coding

Assume we have  $n$  source packets  $\{x_1, \dots, x_n\}$  that contain symbols over a field  $\mathbb{F}_q$ , and we want to convey them to multiple destinations over a network using network coding. Throughout the network, intermediate nodes perform linear combining of the source packets. Thus, a destination receives combinations of the form

$$c_1x_1 + c_2x_2 + \dots + c_nx_n,$$

where  $c_i \in \mathbb{F}_q$ . In the network coding literature, the vector of coefficients

$$c = [c_1, c_2, \dots, c_n]$$

is called a *coding vector*. Each destination can retrieve the data, if it receives  $n$  linearly independent combinations of the source packets, or,  $n$  linearly independent coding vectors. For example, let  $\{\rho_i\}$  be the combined packets a destination collects. We can write in a matrix form:

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_n \end{bmatrix} = \underbrace{\begin{bmatrix} c_{11} & c_{21} & \dots & c_{n1} \\ c_{12} & c_{22} & \dots & c_{n2} \\ \dots & \dots & \dots & \dots \\ c_{1n} & c_{2n} & \dots & c_{nn} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \quad (20.1)$$

If the linear combinations are independent, and matrix  $\mathbf{A}$  is full rank, we can solve the above equations and retrieve the source packets. For example, in the butterfly network

in Figure 20.1, the receivers need to solve systems of equations as in (20.1) with matrices

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

The task of network code design amounts to deciding what linear combinations to form throughout the network so that each receiver gets a full-rank set of equations.

Randomized network coding is based on the simple idea that, for a field size  $q$  large enough, there exist so many valid solutions, that even random choices of the coefficients allow us to find a valid solution with high probability. Thus, we can simply ask each intermediate node in the network to create and send uniform at random linear combinations of the packets it has received. The associated probability of error can be made arbitrarily small by selecting a suitably large alphabet size [10]. For example, if we could choose the coefficients  $\{c_{ij}\}$  of matrix  $\mathbf{A}$  in (20.1) uniformly at random, the matrix  $\mathbf{A}$  would be full rank with probability at least  $(1 - 1/q)^n$ . In practice, simulation results indicate that even for small field sizes (e.g., using  $m = 8$  bits per symbol, i.e.,  $q = 2^8$ ) the probability of error becomes negligible [14].

Randomized network coding requires no centralized or local information, is scalable, and yields to a very simple implementation. Thus, it is very well suited to a number of practical applications, such as sensor networks and more generally dynamically changing networks.

### 20.2.3 Generations and Coding Vectors

The next question to answer is, even if we randomly select what linear combinations to perform, how do we convey to the destinations what are the linear combinations they have received so that they can decode. Moreover, in a network where information gets generated at a constant rate, we need to decide what packets to combine and how often do we decode. To achieve these, we cannot rely on synchronization since packets are subject to random delays, may get dropped, and follow different routes.

The approach in [11] first groups the packets into *generations*. Packets are combined only with other packets in the same generation. A generation number is appended to the packet headers to make this possible (one byte is sufficient for this purpose). The size of a generation can be thought of as the number of source packets  $n$  in synchronized networks: It determines the size of matrices the receivers need to invert to decode the information. Since inverting an  $n \times n$  matrix requires  $\mathcal{O}(n^3)$  operations, and also since waiting to collect  $n$  packets affects the delay, it is desirable to keep the generation size small. On the other hand, the size of the generation affects how well packets are “mixed,” and thus it is desirable to have a fairly large generation size. Indeed, if we use a large number of small-size generations, intermediate nodes may receive packets destined to the same receivers but belonging to different generations. Characterizing this trade-off is an open research problem.

As a second step, the approach in [11] appends within *each* packet header a vector of length  $n$  that describes which linear combination of the source packets  $\{x_1, \dots, x_n\}$  it contains. These vectors are what we called coding vectors. The encoded data is called the *information vector*. For example, the coding vector  $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$ , where the 1 is at the  $i$ th position, means that the information vector is equal to  $x_i$  (i.e., is not encoded). A packet that contains the linear combination  $\rho = c_1x_1 + c_2x_2 + \dots + c_nx_n$  has the coding vector  $(c_1, \dots, c_n)$  and the information vector  $\rho$ .

The coding vectors are updated locally at each node that performs linear combining, to reflect the new linear combination of the source packets that the new packet carries. For example, if a node receives two packets with coding vectors  $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$  and  $(c_1, \dots, c_n)$ , with corresponding information vectors  $x_i$  and  $\rho$ , it can create the new information vector  $\alpha x_i + \rho$  for some value  $\alpha \in \mathbb{F}_q$ . To send this new information vector, it will use the coding vector  $(c_1, \dots, c_{i-1}, c_i + \alpha, c_{i+1}, \dots, c_n)$ . Combining can occur recursively and several times inside the network.

Each receiver examines the coding vectors of the packets it receives, to learn what are the linear combinations it has received. In particular, the coding vectors it receives are nothing but the rows of the matrix  $\mathbf{A}$  in (20.1) that determine the linear equations it needs to solve.

Appending coding vectors to packets incurs an additional overhead. For example, for a packet that contains 1400 bytes, where every byte is treated as a symbol over  $\mathbb{F}_{2^8}$ , if we have  $h = 50$  sources, then the overhead is approximately  $50/1400 \approx 3.6\%$ .

#### 20.2.4 Subspace Coding

The approach based on appending coding vectors in order to be able to decode at the receiver is well suited for large packets where the overhead is small. In wireless sensor networks, and generally, wireless networks, the situation is quite opposite: It is quite often the case that packets consist of a few bits. In such cases, using coding vectors can add a significant overhead.

A new approach recently proposed in [12, 15] promises to be helpful in this situation. This approach is again designed to work with randomized network coding and is based on using subspaces as “codewords” to convey the information from the sources to the receivers. For simplicity, we will here consider a single source transmitting  $n$ -independent packets to receivers, but the same approach can easily be extended to multiple sources [16].

Consider a source that would like to convey  $n$ -independent source packets to receivers over a network that employs randomized network coding. Assume that each packet has length  $\lambda$  over  $\mathbb{F}_q$ . The  $n$  packets can take in a total  $M = q^{n\lambda}$  values. Thus the source for each set of packets has one of these values to convey.

The source can achieve this as follows. First, it selects to operate over an  $nL$ -dimensional vector space  $V$  over  $\mathbb{F}_q$ , that is, a vector space, where vectors have length  $nL$  and have elements in  $\mathbb{F}_q$ . A basis of this space consists of  $nL$  linearly independent vectors. For example, the space  $\mathbb{F}_2^3$  has the basis

$$\{\mathbf{e}_1 = [1 \ 0 \ 0], \mathbf{e}_2 = [0 \ 1 \ 0], \mathbf{e}_3 = [0 \ 0 \ 1]\}.$$

A subspace  $\pi$  is a subset of the vector space  $V$  that is a vector space itself. We can think of subspaces as “planes” that contain the origin. For example, the space  $\mathbb{F}_2^3$  contains seven two-dimensional subspaces. One such subspace is  $\pi_1 = \langle \mathbf{e}_1, \mathbf{e}_2 \rangle$ . Another is  $\pi_2 = \langle \mathbf{e}_2 + \mathbf{e}_3, \mathbf{e}_1 \rangle$ . It also contains seven one-dimensional subspaces, one corresponding to each nonzero vector. Moreover, the subspace (plane)  $\pi_1 = \langle \mathbf{e}_1, \mathbf{e}_2 \rangle$  contains the three “line” (one-dimensional) sub-subspaces  $\pi_3 = \langle \mathbf{e}_1 \rangle$ ,  $\pi_4 = \langle \mathbf{e}_2 \rangle$ , and  $\pi_5 = \langle \mathbf{e}_1 + \mathbf{e}_2 \rangle$ . Therefore, we can define subspaces of lower dimension as sub-subspaces of higher dimensional subspaces. In the above example, we can see that  $\pi_3 \subset \pi_1 \subset \mathbb{F}_2^3 = V$ . We say that two subspaces are *distinct* if they differ in at least one dimension. For example,  $\pi_1 = \langle \mathbf{e}_1, \mathbf{e}_2 \rangle$  and  $\pi_2 = \langle \mathbf{e}_2 + \mathbf{e}_3, \mathbf{e}_1 \rangle$  are distinct.

The source selects a codebook of  $M$  distinct subspaces, and each set of  $n$  packets is mapped to a different such subspace. The receivers learn this codebook. To convey the value of the source packets, the source needs to convey what is the particular subspace to which these packets are mapped. To do so, it inserts in the network a set of basis vectors (packets) that span the subspace. Assume, for example, it sends the vectors  $\{b_1, \dots, b_k\}$  that span a subspace  $\pi$ . The critical observation is that the mixing through randomized network coding that the intermediate nodes perform, preserves the subspaces. Indeed, linear operations, no matter what these operations are, can only create vectors that are in the span of the basis  $\{b_1, \dots, b_k\}$  and thus within  $\pi$ . As a result, every node that receives  $k$  linearly independent vectors will be able to identify which is the subspace  $\pi$  that the source has sent. The source has then transmitted information through the choice of the subspace that it sends. This property makes the use of subspaces for encoding robust to the topology of the network and to arbitrary linear operations performed at the intermediate nodes.

Using coding vectors is a special case of subspace coding [12]. We can see this through an example. Assume the source has  $n = 2$  packets of length  $\lambda = 2$  bits each. Assume that the first packet is  $x_1 = [x_{11} \ x_{12}]$  and the second packet is  $x_2 = [x_{21} \ x_{22}]$ . Using the coding vector approach, the source sends one packet consisting of the coding vector  $[1 \ 0]$  followed by the information vector  $[x_{11} \ x_{12}]$  and another packet consisting of the coding vector  $[0 \ 1]$  followed by the information vector  $[x_{21} \ x_{22}]$ . Let

$$b_1 = [1 \ 0 \ x_{11} \ x_{12}], \quad b_2 = [0 \ 1 \ x_{21} \ x_{22}].$$

We can think of these two packets that the source sends as spanning a two-dimensional subspace of the four-dimensional space  $\mathbb{F}_{2^4}$ . Each time the source has a new set of packets, it will send a different such subspace; in total the source will send 1 out of 16 distinct subspaces since it observes 16 different values.

This corresponds to a particular choice of subspaces in the subspace coding scheme. Observe that  $\mathbb{F}_{2^4}$  contains not only 16, but in fact 35 distinct two-dimensional subspaces. Thus, the sources, using the same packet length, could have conveyed a much higher rate to the receivers, by incorporating in the codebook all the two-dimensional subspaces available. Alternatively, using the subspace approach, we can convey the same information with smaller packet length, and dispense from the coding vector overhead. This promising approach has just started to be explored in the literature.

## 20.3 DATA COLLECTION AND COUPON COLLECTOR PROBLEM

Many of the benefits network coding offers are manifestations of the same simple underlying principle: network coding, by mixing independent packets, makes each packet maximally useful for all its potential receivers. This principle is very nicely captured in the coupon collector problem [17].

### 20.3.1 The Coupon Collector Problem

Assume that coupons of  $n$  different types are placed uniformly at random inside boxes of some commodity. For example, photographs of endangered animals are placed inside cereal boxes. The coupon collector problem asks how many boxes does a coupon

collector need to buy on the average in order to collect all  $n$  coupons. The well-known answer to this question is [18]:

**Theorem 20.1** *The collector needs to buy on the average*

$$n \log n + \Theta(1) \quad (20.2)$$

*boxes.*

*Proof* Let  $X_i$  denote the number of boxes the coupon collector needs to buy in order to increase her collection from  $i - 1$  to  $i$  coupons. When exactly  $i - 1$  coupons have been obtained, the probability that a box brings a new coupon equals

$$p_i = 1 - \frac{i-1}{n}.$$

Thus  $X_i$  follows a geometric distribution with average value

$$E(X_i) = \frac{1}{p_i} = \frac{n}{n-i+1}.$$

Clearly, the average time  $X$  to collect all  $n$  coupons equals

$$E(X) = E\left(\sum_{i=1}^n X_i\right) = \sum_{i=1}^n E(X_i) = n \sum_{i=1}^n \frac{1}{i} = nH_n,$$

where

$$H_n = \sum_{i=1}^n \frac{1}{i} = \log n + \Theta(1)$$

is the harmonic number.

As we see, initially with every box she gets the collector collects a new coupon. However, as her collection increases, so does the probability that when she buys a box she finds a coupon she already has. In fact, she spends the longest time collecting the last few coupons. This is known as the *rare coupons problem*. This is exactly the problem with which network coding can help.

Network coding places a linear combination of coupons, instead of placing a distinct coupon, in each packet [17]. These linear combinations might be chosen uniformly at random or could correspond to some specifically designed codes.

**Theorem 20.2** *Use of network coding allows to collect all the  $n$  coupons in  $\Theta(n)$  time.*

*Proof* Assume that inside each box we place linear combinations chosen uniformly at random over the field  $\mathbb{F}_q$ . We can follow the same proof technique as before, where now, we want each box to bring a combination linearly independent from the combinations previously observed. In a sense, we are collecting basis elements of the  $n$ -dimensional space  $\mathbb{F}_q^n$ . When exactly  $i - 1$  linear independent combinations have

been obtained, these span a subspace of  $\mathbb{F}_q^n$  of dimension  $i - 1$ . The probability that a box brings a randomly chosen vector that does not belong in this subspace equals

$$p_i = 1 - \frac{q^{i-1}}{q^n}.$$

Thus the average time  $X$  to collect  $n$  linear combinations can be calculated as

$$E(X) = E\left(\sum_{i=1}^n \frac{q^n}{q^n - q^{i-1}}\right) < \frac{q}{q-1}n.$$

### 20.3.2 Data Collection in Sensor Networks

Consider a sensor network with  $n$  nodes, where each node  $i$  has an independent observation  $x_i$ . There also exists a set of  $k$  collector nodes. We want the union of information that these  $k$  nodes collect to be sufficient to retrieve all observations  $x_i$ . We consider two models. In the first model, the sensor nodes themselves are mobile, while the collector nodes are static. We call this the *mobile nodes model*. In the second model, we have  $k$  collectors that move randomly among the nodes and collect the information. We call this the *moving collector model* [19].

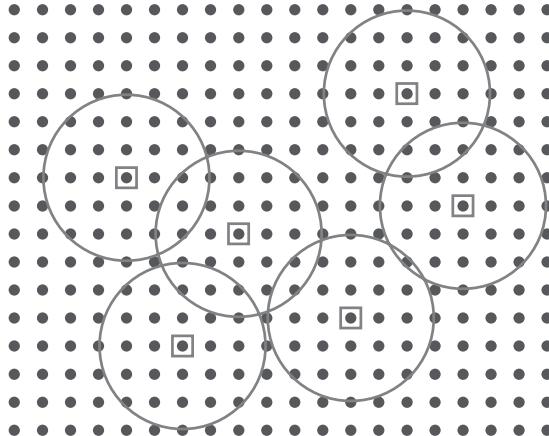
**20.3.2.1 Mobile Node Model** This model corresponds to applications where sensor nodes are placed on mobile objects such as cars or wildlife, that measure statistics to be communicated to base stations. Assume that sensor nodes transmit at a constant range to other sensor nodes as well as to the base stations.

In the case of forwarding, we have a variation of the coupon collector problem, where now we have  $k$  collectors, and we are asking how many boxes should the collectors buy so that the union of their coupons covers all  $n$  possibilities. For  $k$  constant with respect to  $n$ , which is the most realistic case, it is easy to see that the same order arguments apply. That is, we need  $\Theta(n \log n)$  transmissions, while use of network coding results to  $\Theta(n)$  transmissions.

**20.3.2.2 Mobile Collector Model** In this model nodes are static, while collectors are mobile. In particular, we will assume for simplicity that nodes are placed on a square grid, and that we have a total of  $n^2$  nodes. We consider operating this network into two phases, each phase consisting of multiple rounds.

In the first phase, the sensor nodes exchange information, using probabilistic forwarding or randomized network coding, during  $m < n^2$  rounds. At each round all nodes of the square grid transmit once. We assume that each node broadcast transmission is successfully received by its four closest neighbors. As a result, in the case of network coding, after  $m$  rounds each node will have  $4m + 1$  observations, which will depend on the information of the  $\Theta(m^2)$  nodes that are within a radius of  $m$ . Note that information collected by neighbor nodes may have a significant intersection.

In the case of forwarding, we will assume that at each round each node forwards with equal probability one of the four messages it has received in the previous round. Then after  $m$  rounds each node will have collected  $4m + 1$  data  $x_i$ , from nodes within a radius of  $m$ . In particular, given our transmission model, each bit  $x_i$  will perform a random walk with  $m$  steps, and thus on the average we expect it to have reached nodes



**Figure 20.3** Covering a square grid with randomly thrown disks.

within distance  $\Theta(\sqrt{m})$ . We will make the simplifying assumption that each node will receive all  $\Theta(m)$  information symbols  $x_i$  within a radius of  $\Theta(\sqrt{m})$ .

In the second phase a mobile collector samples  $k$  nodes uniformly at random from the square grid. We are asking what is the minimum number of nodes to sample to collect all information. Obviously,  $k \geq n^2/(4m + 1)$ .

We can think of our problem as randomly covering the square grid with disks of radius  $m$ , as depicted in Figure 20.3. Consider a particular node  $i$ . The probability that the node is not covered by a disk during a given round equals the probability that the center of the disk is not within distance  $m$  from node  $i$ . That is, if we uniformly at random choose the center to be any of the  $n^2$  points of the grid, then the center is not one of the  $\Theta(m^2)$  points that are in distance  $m$  from node  $i$ . So this probability equals

$$1 - \frac{m^2}{n^2}.$$

Repeating the experiment for  $k$  rounds, the probability that a node is not covered equals

$$\left(1 - \frac{m^2}{n^2}\right)^k.$$

Assume that we choose  $m = \sqrt{n}$  and that we require that the probability a node is not covered decays at least as  $1/n$ . Then we need at least  $k = n \log n$  rounds. In fact, since

$$k \geq \frac{n^2}{4m + 1} \geq n \log n$$

we need  $k = \Theta(n\sqrt{n})$  rounds, that is, the optimal number of rounds. (Each round corresponds to sampling a node as the collector moves).

In the case of forwarding, we now have that the probability a node is not covered (from disks of radius  $\sqrt{m} = n^{1/4}$ ) equals

$$\left(1 - \frac{1}{n\sqrt{n}}\right)^k$$

and thus we need

$$\Theta(n\sqrt{n} \log n)$$

rounds. As a conclusion, our approximate analysis again indicates a loss factor of  $\log n$ .

## 20.4 DISTRIBUTED STORAGE AND SENSOR NETWORK DATA PERSISTENCE

Distributed storage refers to using multiple nodes (e.g., computers over the Internet) connected through a network to store bulk data. Often, data is replicated in nodes dispersed inside the network to offer reliability and facilitate local access. An immediate problem arising is how to manage the data replication and access. It is well known that using an erasure code for the data replication, instead of simple repetition, leads to a more efficient data representation. Use of network coding for information exchange and update is also proving a promising technique in this context; see, for example, [20]. We will here discuss two sensor network applications in this framework, growth codes and regenerating codes.

### 20.4.1 Growth Codes

An application of the distributed storage problem to sensor networks is in designing sensor networks that can sustain information in the presence of disasters, such as earthquakes, floods, fires, where a significant percentage (or even the vast majority) of the sensor nodes are destroyed. This is a distributed storage problem where we want the random surviving nodes to rescue as much information as possible and eventually forward this information to the sink. To achieve that we can take advantage of the fact that, although there is limited bandwidth to send data to the sink, there still remains available bandwidth for nodes to exchange information so that, if a node fails, its information is not lost. Thus we can have each sensor node act as a storage point.

Clearly, there exist trade-offs between how much information nodes exchange and store and how many nodes need to survive to rescue a significant portion of the information. A naive solution at one extreme would be to replicate all the data on all sensor nodes. In this case, a single surviving node would be sufficient. However, replicating all the data on each node would necessitate a large amount both of information exchange and memory size. Moreover, if multiple nodes survive, they would have the same information. We want the surviving nodes to have saved a significant portion of the information, while at the same time minimizing the replicated data.

This problem can be viewed as one more application of the coupon collector problem, where the coupons are the data, the surviving nodes are the randomly chosen boxes, and the trade-off is how many coupons should we place in each box. Thus similar techniques and ideas as in Section 20.3 can be applied in this case, where, instead of replicating uncoded data, we store linear combinations of the observed data.

An approach that aims to save computational resources uses what are termed “growth codes” [21]. The intuition for these codes can also be traced back to a variation of the coupon collector problem. In this variation we want, instead of placing uniform at random linear combinations of coupons inside the boxes (data in the sensors), to use the minimum possible combining while still collecting all coupons by buying on

the average  $\Theta(n)$  boxes. To achieve this, we use an extra assumption, namely that we know at each point in time how many coupons the collector has acquired, and we can decide what to put in the remaining boxes based on this.

Let  $x_1, \dots, x_n$  denote the coupons. We will consider linear combinations (codewords) over the binary field, which can be implemented with simple xor operations. The degree of a codeword will refer to the number of data that take part in the linear combination. For example, the degree of the codeword  $x_1 + x_5$  equals 2. If we form uniform at random combinations, implemented over the binary field where half the coefficients would be zero and the remaining one, the degree of the codewords placed inside the boxes would equal on the average  $n/2$ . We want to leverage the knowledge of how many coupons the collector has in order to reduce this degree. The idea is to use initially degree 1 and gradually increase the degree as more and more coupons are collected.

Assume that initially the boxes contain one of the symbols  $x_i$ , selected uniformly at random. Recall that in this case, as we saw in the proof of the coupon collector problem (Theorem 20.1), when we randomly select boxes, the probability that box  $i$  brings a new coupon equals

$$p_i = 1 - \frac{i-1}{n}.$$

Thus the first boxes we select bring new information. It is only after we have collected almost half the coupons that the probability a new box will bring a new coupon will be smaller than the probability it will bring a replicate coupon. This is when coding can become useful. Let  $R_1$  denote this number of coupons, with

$$R_1 = \frac{n-1}{2}.$$

Once we reach  $R_1$  coupons, we want to start inserting in the boxes codewords of degree 2. These codewords will, with higher probability, bring useful rather than repetitive information, up to collecting  $R_2$  coupons, with  $R_2 = 2n - 1/3$ . Continuing along the same lines we increment the degree of the codewords placed inside the boxes as the collector acquires more and more coupons to ensure that the probability each box will reveal a new coupon is higher than the probability it will not. This implies that we want to have degree  $i+1$  only once we have collected

$$R_i = \frac{in-1}{i+1}$$

coupons.

The sequence  $\{R_i\}$  can be translated to a probability distribution on the degrees of encoded symbols we would like to have inside the network. In particular, when the sink receives  $k = \Theta(n)$  codewords, we would like  $R_1$  of them to have degree one,  $R_2 - R_1$  to have degree 2, and so on. In other words, we would like to implement the degree probability distribution

$$\frac{R_1}{k}, \frac{R_2 - R_1}{k}, \dots, \frac{k - R_{n-1}}{k}.$$

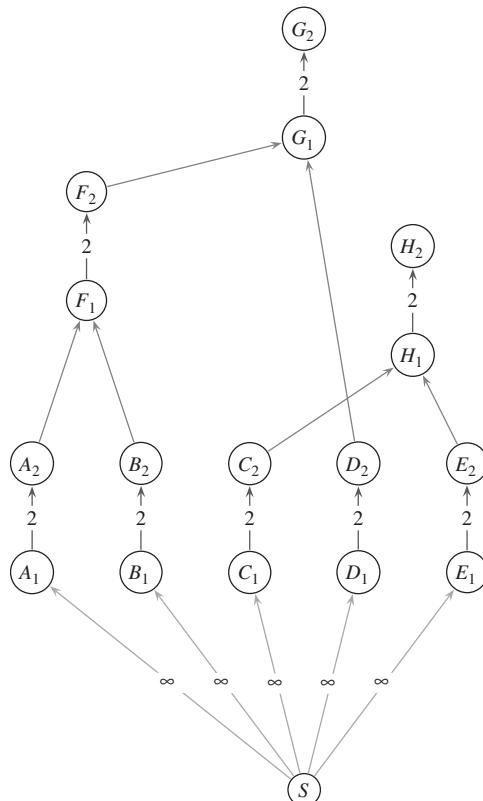
We can achieve this by operating the network in rounds and creating codewords of increasing degree as time progresses. Codewords start in the network with degree 1, and their degree increases over time as they travel through the network to the sink. Each

sensor node was hard-coded before deployment of the values  $R_1$ ,  $R_2 - R_1$ , and so on, and increases, if it can use stored information, the degree of codewords it propagates to  $i + 1$  only after round  $R_i - R_{i-1}$ . Simulation results show good performance of the proposed approach in TinyOs sensor networks [21].

#### 20.4.2 Regenerating Codes

A related problem to distributed storage is that of information update. Assume we use linear combining to store data, several network nodes fail, and we introduce new nodes to repair the system. The question is, how do we generate an encoded fragment for a new node, in a distributed way from the surviving nodes, while transferring as little data as possible across the network. In particular, what is the minimum amount of data we need to transfer, and what coding operations do we need to use at network nodes to achieve it?

The approach in [22] proposes to study this problem by transforming it to an equivalent network code design problem. In particular, let us assume that each sensor node has finite storage capability of 2 bits. We can create a graph, where every sensor is represented by two vertices connected through an edge of capacity 2 bits. For example, in Figure 20.4, sensor node A is represented by vertices  $A_1$  and  $A_2$  connected through



**Figure 20.4** Sensor network where each sensor is represented by two vertices, and the memory of each sensor is explicitly represented through an edge connecting these two vertices.

an edge. This edge captures the storage capability of the sensor. Assume that initially the sensor network consists of nodes  $A$ ,  $B$ ,  $C$ ,  $D$ , and  $E$ , all of them connected to a source through infinite capacity links. The source captures a common source of information or in general a setup phase where nodes can exchange their information without communication capacity constraints. Thus assume that the source has 5 bits to distribute  $x_1, \dots, x_5$ . The nodes, although they can potentially learn each others information, due to storage constraints, each can only store 2 (coded or uncoded) bits. To decide what to store at each node, we could, for example, use an MDS code with 10 codewords, and store 2 of the codewords in each sensor. MDS codes have the property that recovering any 5 of the codewords will allow us to retrieve the data.

Assume now that the new nodes  $F$ ,  $G$ , and  $H$  arrive and are connected to the network as depicted in Figure 20.4. We would like the information stored to all nodes,  $A$  to  $H$ , to now form an MDS code as well.

This is not always possible because, node  $F$  for example, cannot reconstruct the original information but, instead, can only use the symbols stored in nodes  $A$  and  $B$  to create the symbols to be stored in its memory. However, if we could anticipate the possible arrival and connection of network nodes, we could solve a network code design problem on the network in Figure 20.4 and find appropriate linear combinations to store at nodes  $A$  and  $B$  so that this is possible. In fact, using the network representation of the problem in Figure 20.4 we can ask questions such as, how many nodes do we need to connect to retrieve the information, what is the minimum storage required to retrieve the information from a given number of randomly chosen nodes, and what is the communication requirements between the network nodes to achieve this?

A similar graph representation of memory constraints also finds application in peer-to-peer networks; see, for example [23].

## 20.5 DECENTRALIZED OPERATION AND UNTUNED RADIOS

Let  $G = (V, E)$  be a graph (network) with the set of vertices  $V$  and the set of edges  $E \subset V \times V$ . Consider a unicast connection between a source  $S$  to a receiver  $R$ . A *cut* between  $S$  and  $R$  is a set of edges whose removal disconnects  $S$  from  $R$  in the graph. A *mincut* is a cut with the smallest (minimal) value. The *value* of the cut is the sum of the capacities of the edges in the cut.

Assume that the source  $S$  knows the mincut value between itself and the receiver  $R$ . Then, the source can transmit information at a rate equal to this mincut value, by routing information symbols along edge-disjoint paths. This is captured in the well-known mincut max-flow theorem, which was proved in 1956 [24, 25], and extended Menger's theorem proved in 1927 [26].

Assume now that instead of routing we use randomized network coding, where every node in the network sends a uniform at-random combination of the information symbols it has received. Then, we can again achieve rate arbitrarily close to the mincut value [10, 27, 28]. Thus, whether we use network coding or routing, we can achieve exactly the same rate.

The interesting point is that, if we use network coding, all nodes in the network can do *exactly the same operation*, randomly combine their incoming flows and transmit them to their outgoing flows, no matter what is their position in the network. In other words, *even if we have a random network* between the source and the destination, and we know nothing of its structure, provided the mincut to the receiver is maintained,

and allowing all nodes to operate in exactly the same fashion, allows to achieve a rate equal to the mincut. This is not possible in the case of routing, where nodes, depending on their position in the paths from the source to the destination, would need to know which information to forward toward each of their outgoing links. The following example shows a practical application of this result in sensor networks.

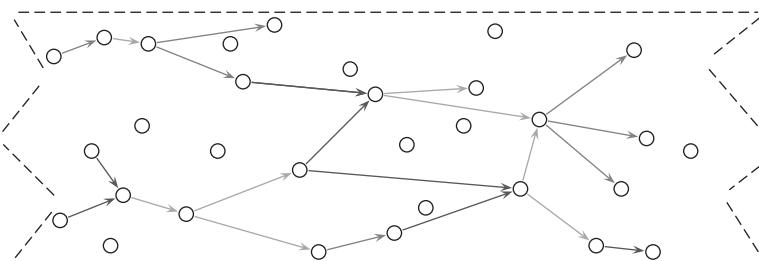
### 20.5.1 Untuned Radios in Sensor Networks

Massive deployment of sensor networks requires that the nodes are easy to manufacture and thus have very low cost, while at the same time the radio component of the node enables communication with as low energy as possible. Narrowband radios consume less power than spread-spectrum or other wideband techniques and thus achieve the later goal. However, the analog components needed for the narrowband radios are more costly as compared to wideband radios because an expensive and bulky off-chip quartz crystal needs to be used: This crystal provides the same low-frequency reference at both transmitter and receiver and ensures that the transmitters carrier frequency and the receivers detection frequency are well matched.

The approach proposed in [29] is to eliminate the quartz crystal and replace it with an on-chip resonator. This architecture allows the sensor node to be developed entirely of thin-film technologies. However, the variations in manufacturing process result in untuned nodes: Each sensor transmits at a randomly chosen (from a finite set) frequency, and receives from a randomly chosen set of the frequencies.

The observation in [29] is that, if we densely enough deploy these cheap sensors, for example, along a corridor to connect a source to a destination through multihops as depicted in Figure 20.5, then we can still create a connected network. In the traditional sensor network, a node would connect to all nodes within its transmission radius (neighbors). The only difference now is that a node will connect to all neighbors that additionally have a matching reception frequency. The randomness in the transmit and receive frequencies of the components of the network simply lead to a different random network configuration.

The challenging part in this new network is how, without centralized knowledge of the node frequencies and topology, we can route the information from the source to the destination. This is where network coding becomes useful. If we know the mincut, we can simply have all sensor nodes perform the same functionality, randomized network



**Figure 20.5** Untuned sensor nodes randomly positioned on a stripe. Each node has a transmitter frequency and a receiver frequency randomly selected from a finite set. A node can communicate with all nodes that are in the neighborhood and have a receiver frequency that matches its transmitter frequency.

coding, and deliver any rate below the mincut to the receiver. If we do not know the mincut, we can estimate it. Using bin through ball arguments, it can be shown that the connectivity pattern of this network, combined with randomized network coding, allows to achieve a fraction  $1/e$  of the max-flow as compared to a tuned network with perfect coordination [29].

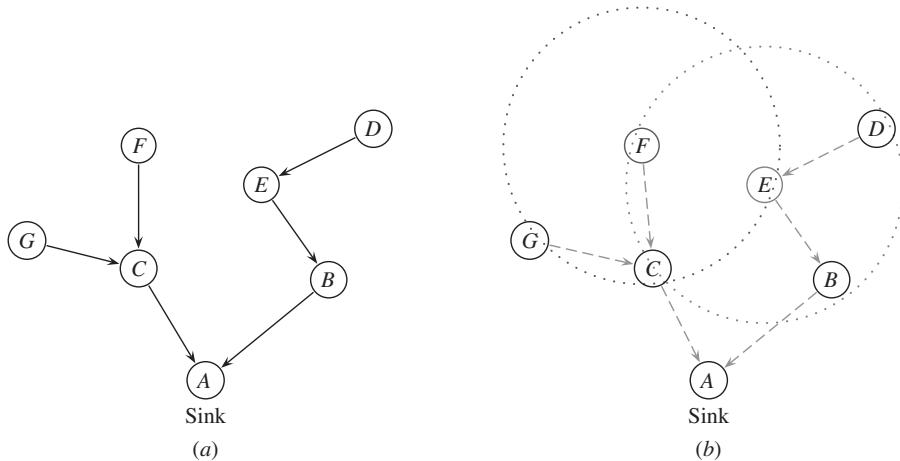
## 20.6 BROADCASTING AND MULTIPATH DIVERSITY

Use of network coding allows to leverage the broadcast capability of the wireless medium and thus achieve increased reliability. To illustrate this idea in a sensor network setting, let us consider a simple example.

Six sensor nodes measuring independent random variables need to convey their measurements to the sink node  $A$ . To do so, the traditional approach has the nodes first, in a setup phase, organize themselves in a “minimum distance” tree structure, such as the one depicted in Figure 20.6. During the setup phase, each node exchanges probe packets with its neighbors and gets connected to the neighbor with whom it can communicate most reliably (has the best channel). Once this communication structure is in place, the nodes simply use the tree to forward the packets with their information to the sink.

This approach is simple to implement and would not really need any improvement if we were operating in an idealized wired environment. Instead, we are using a wireless medium, where interference and time-varying channel quality cause packets to get corrupted and dropped. A natural way to increase the reliability of the network is to leverage the broadcast nature of the wireless medium.

Broadcasting implies that each time a node transmits all neighboring nodes that can successfully receive the packet do so. As a result, multiple copies of the same packet become available inside the network. This offers a form of multipath diversity since there are multiple potential paths connecting each node to the sink. In Figure 20.6, when node  $F$  broadcasts, apart from node  $C$ , potentially nodes  $E$  and  $G$  may also



**Figure 20.6** Sensor network where (left) nodes connect through a minimum tree structure to the sink node  $A$  and (right) where nodes employ broadcasting: (a) without and (b) with broadcasting.

receive the transmitted packet. Similarly, when node  $E$  transmits, both nodes  $B$  and  $C$  may receive the transmitted packet. Thus even if there are packet losses, the probability that a particular packet will survive significantly increases.

Unfortunately, the traditional approach cannot be easily extended to incorporate this feature. This is because, if channel conditions are favorable, the number of copies can increase very fast, causing congestion and unnecessary waste of the limited network resources to convey repetitive information. Restricting the number of copies would require elaborate centralized synchronization and control.

Network coding on the other hand provides an elegant way to deal with this situation. Each network node simply forms a number of linear combinations of all the packets it has successfully received—and broadcasts these packets toward the destination. The number of packets each node transmits is fixed, independent of how many packets the node successfully receives. If we were to consider the source packets as random variables, the number of transmitted packets is chosen to eventually provide sufficient equations to the sink to be able to decode and retrieve the variables. In this approach, if channel conditions are favorable, then a source packet will take part in multiple linear combinations—if not, it will take part in fewer. Thus, there are no longer repetitive packets in the network—and we can fully utilize the diversity broadcasting offers.

To leverage this promising first observation, a number of questions need to be addressed. In the traditional approach, the optimal, in terms of reliability and energy efficiency, structure is that of the “minimum distance” tree, where the distance is measured to be proportional to the channel quality. In the case of broadcasting, what is now the optimal network structure? Clearly, it is no longer a tree since there does not exist a unique path from each node to the sink. And how should this structure be constructed and maintained? A first algorithm toward this goal is proposed in [30].

## 20.7 NETWORK, CHANNEL, AND SOURCE CODING

We will here argue that random linear combinations can be used to facilitate information delivery (network coding), to offer redundancy (channel coding), as well as compress redundant information (source coding). Thus all these three functionalities can be potentially combined.

Assume we have a set of data  $\{x_1, \dots, x_n\}$  that may be correlated, such as temperature measured at geographically close sensor nodes. We want to convey these data to multiple sinks, that is, we want the union of the information the sinks receive to allow us to retrieve all data. Moreover, we assume that packets may get dropped due to link congestion or error.

Assume first that the data are uncorrelated and there are no losses. To deliver the data to multiple sinks using routing, we need to ensure that each source is allocated to a particular sink to avoid data replication. Which sink it is allocated to may depend on considerations such as channel quality or load balancing the information among the different sinks. In networks that change over time, such as mobile sensor networks, elaborate routing schemes might be needed. Network coding allows us to deliver the data to multiple destinations, in a very natural manner: It is sufficient that the union of the sinks collect  $n$  linearly independent combinations. We simply need to create and propagate “degrees of freedom,” that is, independent linear combinations of the data, inside the network, and collect a sufficient number of them. For example, a particular source  $x_i$  may potentially send information to all sinks, in the sense that all sinks may

receive linear combinations that contain  $x_i$ , without incurring unnecessary increase of the information rate delivered. Collecting degrees of freedom is the same idea that has found tremendous success in rateless codes, such as raptor codes [31]. The differences are that with network coding we perform linear combining also within the network, thus better adapting to the network configuration, at the cost of higher decoding complexity.

Assume now that packets get dropped, which we can model using erasure channels. To be able to still deliver the sensor data reliably, we need to add error protection. One method is to rely on MAC layer retransmissions to provide resilience to errors. An alternative approach is to use forward error correction (FEC), which does not require feedback. FEC is well matched for the cases where feedback cannot readily be used, for example, when we employ broadcasting or when sensors fail, and their lack of transmission is experienced as erasures. Perhaps the simplest way to encode the information  $\{x_1, \dots, x_n\}$  against erasures is to simply create and send uniform at random linear combinations of the data. As long as we collect  $n$  linearly independent equations, we can solve a system of equations and retrieve the data.

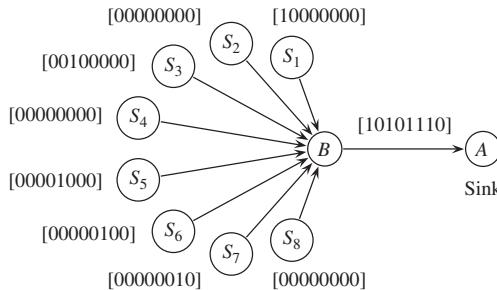
Finally, assume that the data are correlated. That is, if the random variables  $\{x_i\}$  are binary,  $H(\{x_i\}) = k < n$  where  $H(\cdot)$  stands for entropy. This means that we do not need to deliver all  $n$  random variables, but, instead, we can compress the information and deliver the compressed version to save communication resources. In this case as well, using random linear mappings from  $n$  to  $k$  variables can be used to remove the redundancy and deliver the data to the sink (with a probability of error in the reconstruction as described by error exponents [10, 32]).

Observe that for all three previous cases, network coding, channel coding, and source coding, we can simply use random combinations of the data. It thus seems very natural, at least in theory, to combine all these functionalities together using the common operation of linear combining. However, how to perform these tasks in a practical setting is an area still largely unexplored, particularly for the source coding part. A first approach for distributed source schemes for the case of two correlated sources is presented in [33].

## 20.8 IDENTITY-AWARE SENSOR NETWORKS

In many sensor network applications, we are interested in calculating a function of the sensor node measurements. These functions are commonly symmetric, which means that they are invariant under permutations of their inputs. The intuition is that we are interested in the nodes' measurements rather than their identity. Many statistical functions that might be of practical use in sensor networks, such as average, max, and threshold, are symmetric.

The situation is reversed in ad hoc wireless sensor networks that monitor the evolution of an environmental variable over time and space: Sensors are often used to track *whether* and *where* a certain condition occurs—temperature exceeds a safety threshold, a perimeter is violated, and soil or water is contaminated; in other cases, they are used to track (typically small) incremental changes at different locations, for example, the evolution of snow height at different mountain peaks for avalanche prediction or seismic activity for earthquake prediction. In such scenarios, it makes sense to associate each sensor with a fixed location and have it report, periodically, its identity and measurement to a collecting sink; assuming a network of tens or hundreds of nodes, the identities of the reporting nodes now become the bulk of the communicated data,



**Figure 20.7** The sources  $S_1, \dots, S_8$  send their ID and 1 bit of information to sink  $A$  through a relay node  $B$ .

whereas the message itself (i.e., each reported measurement) can be as small as a single bit. We describe such paradigms as *identity-aware* sensor networks [30].

Consider the simplified network of Figure 20.7, where the nodes communicate over an IEEE 802.15.4-compatible link layer. Suppose each node  $S_i, i = 1, \dots, 8$  needs to communicate 1 bit of information to the sink  $A$ ; it specifies this single bit in a packet and sends it through the intermediate node  $B$ . To relay this information to  $A$ ,  $B$  could naively forward it the 8 packets; this would result in 8 wireless frame transmissions, that is,  $8 \times 17$  bytes of MAC layer headers to transmit only 8 bits of information. To avoid this overhead,  $B$  could combine all information in a single packet: Package the 8 bits in a vector, with the understanding that position  $i$  corresponds to the message sent by node  $S_i$ ; this is the simplest example of using a “code” to represent the identity of a node along with its message. The problem with such in-network coding is that it requires the intermediate node  $B$  to understand and process the contents of incoming packets; it would be more practical to develop a coding scheme that operates on an end-to-end basis, that is, information is always encoded at its source and decoded at the sink, while each node is oblivious to the codes used by other nodes.

Now consider the following alternative: Each node  $S_i$  sends out an 8-bit packet with its message encoded in bit  $i$  and all other bits set to 0; node  $B$  just  $\text{xors}$  all incoming packets and sends the resulting 8-bit packet to  $A$ . This approach leads to efficient communication on link  $BA$ , while keeping node  $B$  functionality simple; the price we pay is a small decrease in efficiency on the  $S_iB$  links, which now have to carry 8- (rather than 1-) bit packets, which is insignificant considering the MAC-header overhead.<sup>1</sup>

In general, the idea is that each node  $S_i$  employs a different *codebook*, that is, a different mapping of messages to packets; the sink knows the codebook used by each source and, hence, can determine who sent what, that is the sender implicitly communicates its identity through its choice of codebook. This approach agrees with the insight we have from information theory: The scenario of Figure 20.7 is reminiscent of the classical multiple-access channel problem, where multiple users simultaneously transmit to a single receiver over a common channel; it is well known that the users do not have to explicitly specify their identities, as long as they choose distinct enough codebooks that can be disambiguated at the receiver ([34], Chapter 14).

This is an approach inspired from network coding, where intermediate nodes perform linear operations over the binary field to improve the network performance. This approach can be extended over arbitrary networks and can offer benefits in terms of energy efficiency benefits, load balancing, loss resilience, and scalability [30].

<sup>1</sup>For IEEE 802.15.4-compatible link layer, each MAC packet has a header of 17 bytes.

## 20.9 DISCUSSION

This chapter attempted to give a few examples from the literature to illustrate the potential use of network coding in a sensor network environment. We presented a list of ideas and their corresponding sensor network applications. We discussed how use of network coding offers a more efficient solution to the coupon collector problem and discussed applications in distributed storage and information collection. We pointed out how network coding can help leverage broadcasting and how it can lead to less costly sensor nodes through the use of untuned radios. We then briefly argued that network coding can be combined with source and channel coding. Finally, we considered identity-aware networks and presented a solution inspired from network coding. Many questions are still unexplored, several of them already pointed out during our discussion.

An additional direction of open problems would be the question of providing security over sensor networks that employ network coding techniques. It may be possible to directly translate the approaches designed for security over wireless networks to the case of sensor networks, while other approaches might be too computationally intensive.

## ACKNOWLEDGMENTS

The author would like to thank Katerina Argyraki for reading a draft version of this chapter and providing useful suggestions. This work was supported by the EU Project FP7 215252 N-CRAVE and by the Hasler Foundation ManCom, project no. 2072.

## REFERENCES

1. R. W. Yeung and N. Cai, “Network error correction, i: Basic concepts and upper bounds,” *Commun. Inf. Syst.*, vol. 6, pp. 19–35, 2006.
2. N. Cai and R. W. Yeung, “Network error correction, ii: Lower bounds,” *Commun. Inf. Syst.*, vol. 6, pp. 37–54, 2006.
3. C. Fragouli and E. Soljanin, “Network coding: Fundamentals,” *Found. Trends Networking*, vol. 2, pp. 1–133, 2007.
4. C. Fragouli and E. Soljanin, “Network coding: Applications,” *Found. Trends Networking*, vol. 2, pp. 135–269, 2008.
5. T. Ho and D. S. Lun, *Network Coding: An Introduction*, Cambridge: Cambridge University Press, 2008.
6. J. L. C. Fragouli and J. Widmer, “Network coding: An instant primer,” *ACM SIGCOMM Computer Commun. Rev.*, 2006.
7. M. Effros, R. Koetter, and M. Medard, “Breaking network lojams,” *Sci. Am.*, 2007.
8. P. Chow and Y. Wu, “Network coding for the internet and wireless networks,” Microsoft Research MSR-TR-2007-70, 2007.
9. “Network coding: Networking’s next revolution?” *Network Word*, 2007.
10. T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 4413–4430, Oct. 2006.
11. P. A. Chou, Y. Wu, and K. Jain, “Practical network coding,” in *Proc. Allerton*, Oct. 2003.
12. R. Koetter and F. Kschischang, “Coding for errors and rrasures in random network coding,” in *ISIT*, June 2007.

13. N. R. Wagner, *The Laws of Cryptography with Java Code*, available online at Neal Wagner's home page, 2003.
14. Y. Wu, P. A. Chou, and K. Jain, "A comparison of network coding and tree packing," in *ISIT 2004*, 2004.
15. D. Silva and F. R. Kschischang, "Using rank-metric codes for error correction in random network coding," in *ISIT*, June 2007.
16. M. JadariSiavoshani, C. Fragouli, and S. Diggavi, "Non-coherent network coding for multiple sources," in *ISIT*, 2008.
17. S. Deb, M. Médard, and C. Choute, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," *IEEE/ACM Trans. Networking*, vol. 14, pp. 2486–2507, June 2006.
18. M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press, 2005.
19. C. Fragouli, J. Widmer, and J.-Y. L. Boudec, "On the benefits of network coding for wireless applications," paper presented at the Network Coding Workshop, Boston, 2006.
20. A. G. Dimakis and K. Ramchandran, "Network coding for distributed storage in wireless networks," in *Networked Sensing Information and Control, Signals and Communication Series*, Springer Verlag, 2007.
21. A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, "Growth codes: Maximizing sensor network data persistence," in *SIGCOMM '06: Proceedings of the 2006 conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2006, pp. 255–266.
22. Y. Wu, A. Dimakis, and K. Ramchandran, "Deterministic regenerating codes for distributed storage," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2398–2409, June 2006.
23. R. Yeung, "Avalanche: A network coding analysis," *Commun. Inform. Systems*, 2005.
24. L. R. F. Jr. and D. R. Fulkerson, "Maximal flow through a network," *Can. J. Math.*, vol. 8, pp. 399–404, 1956.
25. P. Elias, A. Feinstein, and C. E. Shannon, "Note on maximum flow through a network," *IRE Trans. Inform. Theory*, vol. 2, pp. 117–119, 1956.
26. K. Menger, "Zur allgemeinen kurventheorie," *Fund. Math.*, vol. 10, pp. 95–115, 1927.
27. R. Ahlswede, N. Cai, S-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, pp. 1204–1216, July 2000.
28. S-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, pp. 371–381, Feb. 2003.
29. D. Petrović, K. Ramchandran, and J. Rabaey, "Overcoming untuned radios in wireless networks with network coding," *IEEE/ACM Trans. Networking*, vol. 14, pp. 2649–2657, June 2006.
30. M. J. Siavoshani, L. Keller, K. Argyraki, S. Diggavi, and C. Fragouli, "Identity aware sensor networks," EPFL Technical Report, 2008.
31. A. Shokrollahi, "Raptor codes," *IEEE Trans. Inform. Theory*, vol. 52, pp. 2551–2567, 2006.
32. I. Csiszar, "Linear codes for sources and source networks: Error exponents, universal coding," *IEEE Trans. Inform. Theory*, vol. 28, pp. 585–592, 1982.
33. Y. Wu, V. Stankovic, Z. Xiong, and S. yuan Kung, "On practical design for joint distributed source coding and network coding," paper presented at the First Workshop on Network Coding, Theory and Applications, 2005.
34. T. Cover and J. Thomas, *Elements of information theory*, Hoboken, NJ: Wiley, 2006.



## CHAPTER 21

---

# Information-Theoretic Studies of Wireless Sensor Networks

Liang-Liang Xie<sup>1</sup> and P. R. Kumar<sup>2</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada

<sup>2</sup>Department of Electrical and Computer Engineering, and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois

### 21.1 INTRODUCTION

Wireless sensor networking is an emerging technology that has a wide range of potential applications including environment and habitat monitoring, smart spaces, health monitoring, traffic control, and the like. Such a network normally consists of a large number of nodes distributed in space, each of which may be equipped with one or more sensors, a radio transceiver, a microcontroller, and an energy source, usually a battery. In many applications, it is often very difficult to change or recharge batteries for these nodes<sup>1</sup>. Thus, prolonging the network lifetime by efficiently using the battery energy is a critical issue in the operation of wireless sensor networks. Generally, in sensor networks, wireless communications consume much more energy compared to the other node activities. Developing much more energy-efficient communications schemes will certainly lead to significant energy savings [1].

In many applications of sensor networks, nodes remain largely inactive for a long time but become suddenly active when something is detected. It greatly saves energy to let nodes sleep, instead of listening to idle channels for most of the time [2]. Since nodes consume much less energy in sleep mode, a possible strategy is to wake up as few nodes as possible when an event is detected that needs to be communicated to the sink. This consideration leads to a single-hop operation where the source node directly transmits to the sink without the help of other nodes. However, if the hop is of long distance, the communication rate will be low due to signal attenuation, which leads in turn to long transmission time. On the other hand, although multihop communication involves multiple nodes, it greatly increases the communication rate, and thus shortens the total awake time. One obvious way to shorten the transmission time is to let each node transmit at its peak power. However, since the communication rate only increases

<sup>1</sup>This has adverse environmental implications.

logarithmically with power, after some point, increasing transmit power will actually consume more energy in total although the wake time is shortened. Hence, it may not be favorable to work at high signal-to-noise ratio (SNR) in terms of energy saving in sensor networks. Therefore, it is of great interest to clarify the optimal operation regime in terms of transmit power, power attenuation, number of hops, distance of each hop, and the like.

Compared to wireline communication, wireless is essentially different in its broadcast nature, which necessarily causes interference. In current protocols for wireless networks, interference is often regarded as undesirable, and a common tactic is to avoid it at least locally by silencing other transmitters in the neighborhood. An example is the RTS-CTS handshake in IEEE 802.11 [3]. However, viewed more fundamentally, “interference” is really not noise and is actually a signal that carries information, only unintentionally received. This motivates the challenge of exploiting interference rather than succumbing to it. In this chapter we develop wireless communication schemes that exploit such unintentionally received signals.

In addition, since it is typical to have multiple sensors activated by the same event, they may have correlated information to transmit. This typical phenomenon in sensor networks of high correlation among the information collected at different nodes makes them further different from other wireless networks. It is favorable to exploit the correlation and cooperation among these nodes in developing communication schemes. A simple way to avoid sending redundant information from different sensor nodes is by using the classical Slepian–Wolf source coding technique. However, from the viewpoint of robustness, redundant information can possibly improve the reliability, especially for failure-prone sensor networks. Actually, when source coding and channel coding are jointly considered, it is possible to send redundant information without consuming additional transmission energy. In this chapter we also develop such joint source–channel coding schemes.

## 21.2 INFORMATION-THEORETIC STUDIES

Before undertaking any special designs for the wireless communications in sensor networks, a good understanding is needed of the capacity of general wireless networks. The fundamental question is: How much information can a wireless network transport at most? In [4] it was shown that the capability of a wireless network manifests itself not only in the information transmission *rate* but also in the information transmission *distance*. To reflect this, the concept of *transport capacity* was introduced to account for the total rate–distance product (in the unit “bit-meters/time unit”) that a wireless network can support. One key result obtained was that the transport capacity of a wireless network grows at most like the square root of the product of the area of the network and the number of the nodes. Another was that if the node locations are random, and every node chooses a random destination for its originating traffic, then, as the number  $n$  of nodes increases, there is a sharp cutoff of  $\Theta(1/\sqrt{n \log n})$  for the uniform rate that can be supported for every such source–destination pair.

The scaling laws obtained in [4] are, however, not conclusive due to the restrictions deliberately imposed on the mode of operation in terms of the simplicity of the receivers employed. In particular, when using a simple receiver, interference gives rise to collisions. Thus, it is of interest to study the benefit achievable by more sophisticated multiuser radios that can employ techniques such as successive interference

cancellation where strong interference is actually easy to delete, code division multiple access (CDMA) where multiple transmitters can simultaneously send information to the same receiver, and other cooperative strategies such as amplifying and forwarding without decoding. To an information theorist, the ultimate goal is to find out what is possible or impossible, without making a priori technological presumptions.

In a subsequent study [5], general wireless networks were therefore studied in an information-theoretic setting. Since “distance” plays such a crucial role, as evidenced by the conservation laws for the transport capacity alluded to above, it was incorporated into the model not only through making explicit the distances between nodes but also through explicitly modeling the attenuation of signals with distance  $\rho$  by the factor  $e^{-\gamma\rho}/\rho^\delta$ . A fundamental result established in [5] is that the transport capacity is always upper bounded by a multiple of the total power used by the transmissions of all the nodes in the network, provided that the signals are attenuated sufficiently with distance. This multiple thus corresponds to the maximum bit-meters of transport that a network can deliver per unit energy consumed by transmissions. For planar networks, it was shown that either  $\gamma > 0$  or  $\gamma = 0$  but  $\delta > 3$  was sufficient for the existence of such an irreducible energy cost per unit transport, while  $\gamma > 0$  or  $\gamma = 0$  but  $\delta > 2$  was sufficient for linear networks where the nodes are arranged along a line. On the other hand, counterexamples were also provided of multiple relay networks to show that if  $\gamma = 0$  but  $\delta < \frac{3}{2}$  for two-dimensional networks, or  $\gamma = 0$  but  $\delta < 1$  for one-dimensional networks, then the transport capacity can indeed be unbounded even with bounded total transmission power.

For wireless networks where each node has the same constraint on its transmission power, the above result immediately establishes a linear scaling law for the transport capacity since the total transmission power itself grows linearly in the number of nodes. This is a slight sharpening of, but in essential conformity with, the  $O(\sqrt{An})$  scaling law established in [4] since the area of the domain grows at least like  $n$  with a minimum internode spacing. Since linear scaling is in fact achievable, as constructively shown in [4], and that too using only simple decode-and-forward multiple hopping where at each hop all concurrent interference is treated as noise, the optimality of the order of the best case transport capacity is thus established for the range of attenuations where this linear scaling is established. Note that this also proves that the above architecture for information transport is optimal to within a constant factor.

Thus, interest centers on determining precisely for what range of path loss exponents  $\delta$  (with  $\gamma = 0$ ) linear scaling of transport capacity can indeed be established. With [5] formulating and resolving this issue for  $\delta > 3$  for two-dimensional networks, and  $\delta > 2$  for one-dimensional networks, there remained a gap  $\frac{3}{2} \leq \delta \leq 3$  for two-dimensional networks, and  $1 \leq \delta \leq 2$  for one-dimensional networks, where the scaling law behavior was unknown. In subsequent works [6–8], improvements were made, and, at the present time, the best results known [8] are that  $\delta > 2$  (with phase fading) and  $\delta > \frac{5}{2}$  (without fading) for two-dimensional networks, and  $\delta > \frac{5}{4}$  (with phase fading) and  $\delta > \frac{3}{2}$  (without fading) for one-dimensional networks, are sufficient for linear scaling to hold.

Instead of transport capacity, it is relatively easier to upper bound the average rate per communication pair by assuming all the pairs are uniformly distributed in the network and they are communicating at the same rate. In this setting, it has been shown [9–11] that the average rate tends to zero as the number of nodes in the network grows to infinity when  $\delta > 1$  (with phase fading) and  $\delta > 0$  (without fading)

for two-dimensional networks. For such a result, the required attenuation exponent  $\delta$  is much smaller compared to that needed for linear scaling of the transport capacity.

In all these works, the information-theoretic tool used to prove the upper bounds is the cut-set bound, which is also known as the max-flow min-cut bound; see [12, Section 14.10] for a general formulation in terms of mutual informations. For the specific application to wireless networks, a formula in terms of powers was presented in [5].

Essentially, the cut-set bound is an application of Fano's inequality to the network scenario. It is known that Fano's inequality provides a tight upper bound on the rate achievable from one source to one destination. For a network with multiple nodes, the idea is to dissect it into two sets, with one regarded as the virtual "source" and the other as the virtual "destination." Then, by Fano's inequality, one can bound the total rate achievable from the nodes in the source set to the nodes in the destination set. However, this bound is no longer tight, unless all the nodes in the source set can cooperate in the encoding, and also all the nodes in the destination set can similarly cooperate in the decoding, both of which are generally not feasible.

### 21.2.1 Some Related Studies of Sensor Networks

This section surveys some recent related studies of sensor networks. The capability of large-scale sensor networks was investigated in [13]. The authors consider a data-gathering wireless sensor network in which densely deployed sensors take periodic samples of the sensed field and then scalar quantize, encode, and transmit them to a single receiver/central controller where snapshot images of the sensed field are reconstructed. Subject to a constraint on the quality of the reconstructed field, the main focus of the study is on determining how fast data can be collected. The question is: Can the encoder compress sufficiently to meet the limit imposed by the transport capacity? The conclusion is that for the given scenario, even though the correlation between sensor data increases as the density increases, any data compression scheme is insufficient to transport the amount of data required for the given quality.

Gupta et al. [14] focus on techniques that exploit correlations in sensor data to minimize communication energy costs incurred during data gathering in a sensor network. The proposed approach is to select a small subset of sensor nodes that are sufficient to reconstruct data of the entire sensor network. The selected set of sensors must also be connected to relay data to the data-gathering node. A set of energy-efficient distributed algorithms are designed to select a connected correlation-dominating set of small size. Simulation results are provided to demonstrate the efficiency of the designed algorithms.

The power efficiency of a communications channel, that is, the maximum bit rate that can be achieved per unit power (energy rate) is considered in [15]. It is shown that for a random wireless sensor network with users (nodes) placed in a domain of fixed area, the power efficiency scales at least by a factor of  $\sqrt{n}$ , with probability converging to 1 as  $n \rightarrow \infty$ . A random ad hoc network with  $n$  relay nodes and  $r$  simultaneous transmitter/receiver pairs located in a domain of fixed area is examined. It is shown that as long as  $r \leq \sqrt{n}$ , one can achieve a power efficiency that scales by a factor of  $\sqrt{n}$ .

In [16], dense wireless sensor networks deployed to observe arbitrary random fields are studied. First, the transport capacity of many-to-one dense wireless networks is shown to scale as  $\Theta(\log(N))$  when the number of sensors grows to infinity and the total

average power remains fixed. Then, this result is used along with information-theoretic tools to derive sufficient and necessary conditions that characterize the set of random fields observable by dense sensor networks. In particular, for random fields that can be modeled as discrete random sequences, a certain form of source–channel coding separation theorem is derived. It is shown that one can achieve any desired nonzero mean-square estimation error for continuous, Gaussian, and spatially band-limited fields, through a scheme composed of single-dimensional quantization, distributed Slepian–Wolf source coding, and a proposed antenna sharing strategy.

The study in [17] focuses on the joint source–channel communication perspective in sensor networks. It is well known that these two tasks may not be addressed separately without sacrificing optimality, and the optimal performance is generally unknown. This work presents a lower bound on the best achievable end-to-end distortion as a function of the number of sensors, their total transmit power, the number of degrees of freedom of the underlying source process, and the spatiotemporal communication bandwidth. It is shown that the standard practice of separating source from channel coding may incur an exponential penalty in terms of communication resources, as a function of the number of sensors. Hence, such code designs effectively prevent scalability.

In [18], the transport capacity of a data-gathering wireless sensor network under different organizations of the communication system is studied. In particular, a flat as well as a hierarchical/clustering architecture to realize many-to-one communications are considered. The capacity of the network under this many-to-one data-gathering scenario is reduced in comparison to random one-to-one communication due to the unavoidable creation of a point of traffic concentration at the data collector/receiver. The overall throughput bound of  $\lambda = W/n$  per node, where  $W$  is the transmission capacity, is exhibited. It is also shown how the introduction of clustering can improve the throughput.

The model utilized in [19] assumes that each sensor observes only a subset of the state of nature, that sensor observations are localized and dependent, and that a sensor network output across different states of nature is neither identical nor independently distributed. Using a random coding argument, a lower bound on the “sensing capacity” of a sensor network is established, which characterizes the ability of a sensor network to distinguish among all states of nature. This lower bound is computed for sensors of varying range, noise models, and sensing functions.

In [20], coding strategies for estimation under communication constraints in tree-structured sensor networks are developed. The strategies are based on a generalization of Wyner–Ziv source coding with decoder side information. Solutions for general trees are developed, and the results are illustrated in serial (pipeline) and parallel (hub-and-spoke) networks. Additionally, the strategies can be applied to other network information theory problems.

In [21], the use of channel state information (CSI) for random access in fading channels is studied. A reception model that takes into account the channel states of various users is introduced. Under the assumption that each user has access to its CSI, the authors propose a variant of the slotted ALOHA protocol for medium access control, where the transmission probability is allowed to be a function of the CSI. It is shown that the effect of transmission control is equivalent to changing the probability distribution of the channel state. The theory is then applied to CDMA networks with linear minimum mean-square error receivers and matched filters to illustrate the effectiveness of using channel state. It is shown that through the use of channel state, with

arbitrarily small power, it is possible to achieve an asymptotic stable throughput that is lower bounded by the spreading gain of the network.

It is widely recognized that joint optimization of the operations of sensing, processing, and communication can result in significant savings in the use of network resources. In [22], a distributed joint source–channel communication architecture is proposed for energy-efficient estimation of sensor field data at a distant destination, and the corresponding relationships between power, distortion, and latency are analyzed as a function of number of sensor nodes. The approach is based on distributed computation of appropriately chosen projections of sensor data at the destination. Phase-coherent transmissions from the sensor nodes enable exploitation of the distributed beamforming gain for energy efficiency.

## 21.3 RELAY SCHEMES

We now turn to a more detailed discussion of relay schemes. These kinds of channel coding schemes fully exploit all the signals received, maximizing the signal power that can be used while minimizing interference and thus achieving energy efficiency.

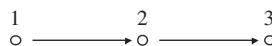
### 21.3.1 Relay Channel

The relay schemes can be easily motivated in a simple three-node network as depicted in Figure 21.1. Suppose node 1 is the source, which wants to send information to the destination node 3. In many situations, the destination node 3 may be at a great distance from node 1 so that any signal transmitted directly from node 1 to node 3 suffers such a considerable attenuation that it precludes any direct reliable communication at a high rate. In such a situation, one wants to exploit the presence of the intermediate node 2, so that node 1 can first transmit to node 2, with node 2 then transmitting to node 3. This results in two shorter range communications, both reliably feasible at a high enough rate. The fundamental question that arises at this point is whether the signal transmitted by node 1 necessarily causes “interference” to node 3? The fact is that although this signal is intended for node 2, it carries exactly the same information that node 3 wants to decode eventually.

Motivated by this problem, the “relay channel” first proposed in [23, 24] almost 40 years ago has become one of the basic topics of multiuser information theory, where the interest centers on developing coding schemes such that node 3 can effectively exploit both the signals transmitted by node 1 and node 2. Two fundamentally different coding strategies, called decode-and-forward and compress-and-forward, differing in whether the relay node 2 decodes the information or not, were developed in [25].

Denoting the signals transmitted by node 1 and node 2 as  $x_1(t)$  and  $x_2(t)$ , respectively, and the signals received by node 2 and node 3 as  $y_2(t)$  and  $y_3(t)$ , respectively, we assume that they are related by the probability transition function

$$p(y_2(t), y_3(t)|x_1(t), x_2(t)), \quad \text{for any time } t,$$



**Figure 21.1** One-relay network.

which describes the discrete memoryless channel involved. It has been proved in [25] that any rate  $R$  satisfying the the following inequality is achievable with a decode-and-forward strategy:

$$R < \max_{p(x_1, x_2)} \min\{I(X_1; Y_2|X_2), I(X_1, X_2; Y_3)\}. \quad (21.1)$$

Examining more closely the two constraints on  $R$ :

$$R < I(X_1; Y_2|X_2), \quad (21.2)$$

$$R < I(X_1, X_2; Y_3), \quad (21.3)$$

we observe that (21.2) is what is needed for the relay node 2 to indeed be able to decode the information based on the signal transmitted by node 1. The second constraint (21.3) applies irrespective of the particular scheme used since it represents the limitation that, at best, node 3 can only make use of the signals transmitted by nodes 1 and 2.

At first sight, both (21.2) and (21.3) look quite understandable and even straightforward. This is, however, misleading. If one looks closely at formula (21.1), it is actually very surprising that any rate satisfying only these two constraints is feasible since the maximization is over  $p(x_1, x_2)$  rather than over  $p(x_1)p(x_2)$ , which can only be achieved by node 1 and node 2 cooperating with each other when transmitting signals. That this is feasible is rather surprising since there is always a positive delay before node 2 can decode the information concerning the intention of node 1. But, by that time, node 1 would have moved on to transmit new information. Hence, node 2 can never catch up with node 1, which raises the issue of how they can cooperate together to transmit to node 3.

Indeed, the coding scheme developed in [25] to achieve (21.1) is nontrivial. The essential technique used is what is called block Markov encoding, which also has profound applications in other areas of multiuser information theory, including the multiple-access channel with feedback [26].

### 21.3.2 Multiple Relays

A natural extension of the one-relay network in Figure 21.1 is to the case of multiple relays, depicted in Figure 21.2, where it takes multiple hops to send information from source node 1 to destination node  $n$ . A natural question is whether formula (21.1) can be extended. Surprisingly, such an extension studied in [5, 27–32] turned out to be not trivial. It was realized in [5] that any rate  $R$  satisfying the following inequality is achievable:

$$R < \max_{p(x_1, \dots, x_{n-1})} \min_{2 \leq k \leq n} I(X_1, \dots, X_{k-1}; Y_k|X_k, \dots, X_{n-1}). \quad (21.4)$$

Although (21.4) is still achievable with a decode-and-forward strategy, it is not achievable with the specific “irregular” encoding/successive decoding scheme



**Figure 21.2** Multiple-relay network.

developed in [25]. Instead, in [5], a “regular” encoding/“sliding-window” decoding scheme was shown to achieve (21.4). The sliding-window decoding had been used in [33] in the context of the multiple-access channel with generalized feedback. Later on, it was discovered in [30] that (21.4) can also be achieved with the “backward” decoding scheme, which was invented in [34] and has since been used for the one-relay channel in [35]. Recently, a modified successive decoding scheme (having the flavor of sliding-window decoding) was developed in [36] to achieve (21.4). Among all these schemes, the sliding-window decoding scheme is the simplest, while the backward decoding gives rise to large decoding delay and is also the most involved.

Formula (21.4) has a similar interpretation as (21.1). For each node  $k \in \{2, \dots, n\}$ , the corresponding constraint is

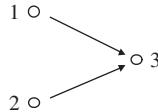
$$R < I(X_1, \dots, X_{k-1}; Y_k | X_k, \dots, X_{n-1}). \quad (21.5)$$

The conditional mutual information on the right-hand side above implicitly presumes that for the decoding at node  $k$ , the signals transmitted by nodes  $k, \dots, n-1$  are known a priori, and that the signals transmitted by nodes  $1, \dots, k-1$  are cooperating in providing the information. Let us examine the first issue of why node  $k$  should know what will be transmitted by nodes  $k+1, \dots, n-1$ . The reason is that, in this system, there is only one source–destination pair, for which information is passed along the route  $1 \rightarrow 2 \rightarrow \dots \rightarrow n$ . Thus, any information obtained by nodes  $k+1, \dots, n-1$  has already been obtained by node  $k$ . Hence node  $k$  does indeed already know completely what nodes  $k+1, \dots, n-1$  know, and therefore knows what they will transmit.

The preceding interpretation illustrates the remarkable feature of formula (21.4): There is no interference at all in the whole network! To any node, the signal transmitted by any other node is either a “real” signal that can be used for decoding or an a priori known signal that can be subtracted completely.

Compared to the simple practice of regarding other transmitters as interferers, the relay schemes therefore seem to be obsessed with exploiting interference rather than succumbing to it. But is it worth it? How much can be gained by using such “smarter” schemes? A study in [5] shows that in wireless networks with low signal attenuation, the multirelay scheme can achieve higher order (superlinear) scaling laws compared to those obtained in [4]. However, when signal attenuation is high enough, the scaling laws cannot be improved, as proved in [5], but it is still possible to achieve substantially higher rates, especially for small-scale networks.

So far, we have been focused on the decode-and-forward strategy, which fundamentally relies on the relay nodes being able to decode the information they are transmitting. However, the relay nodes can help even without decoding the information themselves. This is the strategy employed in schemes such as compress-and-forward, amplify-and-forward, and the like. Unfortunately, there is no single, known relay strategy that is superior to all others in all scenarios. Which relay strategy is better really depends on the network topology, power distribution, and the like (see [30, 37]) For our discussion here, we prefer to concentrate on the decode-and-forward strategy since it is the only one where interference completely disappears [as in (21.4)]. In either compress-and-forward or amplify-and-forward, interference is inevitably present since without decoding, noise cannot be filtered out and will always be forwarded along with the signal.



**Figure 21.3** Multiple-access network.

### 21.3.3 Multiple Sources

When there is only one source in the network and all transmitted signals are devoted to it, interference can be completely avoided, as (21.4) demonstrates. But what if there are multiple sources? Will signals devoted to different sources always interfere with each other?

Consider the simple two-source network in Figure 21.3, where nodes 1 and 2 are two separate sources. Let us consider the case where both source node 1 and source node 2 want to send information to the same destination node 3. Can they transmit at the same time and be both successful? The answer actually is yes and is now well known. From the characterization of the capacity region for the multiple access channel in [38, 39], we know that not only can both transmissions be simultaneously successful but also that simultaneous transmission is indeed a way to achieve the maximal rates and can be realized using the coding scheme of CDMA.

Denoting the signals transmitted by node 1 and node 2 by  $x_1(t)$  and  $x_2(t)$ , respectively, and the signal received by node 3 by  $y_3(t)$ , we consider the discrete memoryless channel where they are related by the probability transition function:

$$p(y_3(t)|x_1(t), x_2(t)), \quad \text{for any time } t.$$

Any rate pair  $(R_1, R_2)$  satisfying the following is achievable with CDMA:

$$R_1 < I(X_1; Y_3|X_2), \quad (21.6)$$

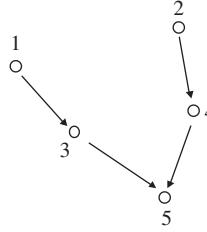
$$R_2 < I(X_2; Y_3|X_1), \quad (21.7)$$

$$R_1 + R_2 < I(X_1, X_2; Y_3). \quad (21.8)$$

This rate region (21.6)–(21.8) has been proved to be maximal in [38, 39]. If only (21.6) and (21.7) were present, it would seem that source 1 and source 2 can be decoded separately without interfering with each other. However, the sum-rate constraint (21.8) makes it impossible to always achieve both (21.6) and (21.7) at the same time. Therefore, the two sources are indeed affecting each other. Nevertheless, (21.6)–(21.8) is the best one can do.

### 21.3.4 Two-Source Relay Channel

Next, we develop a multiple-relay scheme for networks with *multiple sources*. We will try to preserve the spirit of (21.4), in the sense that all useful signals are used and all a priori known interferences are subtracted. We will also try to achieve the best rate region in a form similar to (21.6)–(21.8), for nodes that have multiple sources to decode. The final scheme will accordingly have the flavors of both the multiple-relay and the multiple-access channels.



**Figure 21.4** Two-source relay network.

As a starting point toward a general multisource multirelay coding scheme, consider the network depicted in Figure 21.4, where two source nodes 1 and 2 want to send independent information to the same destination node 5, with nodes 3 and 4 acting as the relays.

According to their relative locations, two relay routes are chosen in the network:  $1 \rightarrow 3 \rightarrow 5$  and  $2 \rightarrow 4 \rightarrow 5$ . That is, node 3 helps source node 1, and node 4 helps source node 2. For the reasons discussed in the introduction, we only consider the decode-and-forward strategy in this chapter. Therefore, we consider schemes where node 3 needs to decode the information sent by node 1 and node 4 needs to decode the information sent by node 2.

Before rigorously stating the achievable rates for the network, we need to introduce some information-theoretic notation for the network channel model. This network channel is modeled by

$$(\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \times \mathcal{X}_4, p(y_3, y_4, y_5 | x_1, x_2, x_3, x_4), \mathcal{Y}_3 \times \mathcal{Y}_4 \times \mathcal{Y}_5),$$

where  $\mathcal{X}_i$  and  $\mathcal{Y}_j$  are finite input and output alphabets, respectively, and  $p(y_3, y_4, y_5 | x_1, x_2, x_3, x_4)$  is a probability distribution on  $\mathcal{Y}_3 \times \mathcal{Y}_4 \times \mathcal{Y}_5$  for each  $(x_1, x_2, x_3, x_4) \in \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \times \mathcal{X}_4$ . At any time  $t = 1, 2, \dots$ , each node  $i \in \{1, 2, 3, 4\}$  sends  $x_i(t) \in \mathcal{X}_i$  into the channel, and each node  $j \in \{3, 4, 5\}$  receives  $y_j(t) \in \mathcal{Y}_j$  from the channel. The distribution of the outputs  $[y_3(t), y_4(t), y_5(t)]$  only depends on the inputs at time  $t$  via

$$p(y_3(t), y_4(t), y_5(t) | x_1(t), x_2(t), x_3(t), x_4(t)).$$

While we consider only the case of finite alphabets, a continuous model can be approximated arbitrarily well by choosing the alphabet size large enough as in [40, Chapter 7].

Source nodes 1 and 2 transmit signals  $x_1(t)$  and  $x_2(t)$  based on the messages they want to send. Relay nodes 3 and 4 decide what to transmit based only on the signals they have already received:

$$x_3(t) = f_{3,t}(y_3(1), \dots, y_3(t-1)),$$

$$x_4(t) = f_{4,t}(y_4(1), \dots, y_4(t-1)),$$

where  $f_{3,t}(\cdot)$ ,  $f_{4,t}(\cdot)$ ,  $t \geq 1$  can be any functions. After a time block  $1 \leq t \leq T$ , the destination node 5 needs to decode both the messages sent by nodes 1 and 2, based on the signals  $\{y_5(1), y_5(2), \dots, y_5(T)\}$  it has received. The strict definitions of codes,

encoding functions, decoding functions, probability of error, and achievable rates are as standard in information theory. See, for example, [31] for details.

Denote by  $R_1$  and  $R_2$  the transmission rates of source nodes 1 and 2, respectively. A rate pair  $(R_1, R_2)$  is said to be achievable if both the messages can be decoded at destination node 5 with an arbitrarily small probability of error. We have the following theorem [41] characterizing the achievable rate pairs.

**Theorem 21.1** *For the two-source relay network defined above, any rate pair  $(R_1, R_2)$  satisfying the following five inequalities is achievable:*

$$R_1 < I(X_1; Y_3|X_3), \quad (21.9)$$

$$R_2 < I(X_2; Y_4|X_4), \quad (21.10)$$

and

$$R_1 < I(X_1, X_3; Y_5|X_2, X_4), \quad (21.11)$$

$$R_2 < I(X_2, X_4; Y_5|X_1, X_3), \quad (21.12)$$

$$R_1 + R_2 < I(X_1, X_3, X_2, X_4; Y_5), \quad (21.13)$$

for some joint distribution  $p(x_1, x_3)p(x_2, x_4)$ .

The constraints (21.9) and (21.10) can be understood similarly to (21.2) since in our scheme node 3 needs to decode the information sent by node 1 and node 4 needs to decode the information sent by node 2. The constraints (21.11)–(21.13) are for the decoding at node 5, which looks like an extension of (21.6)–(21.8), only different in that now with the help of the relays, there are two inputs  $(X_1, X_3)$  for source 1 and two inputs  $(X_2, X_4)$  for source 2. Therefore, the achievable rate region (21.9)–(21.13) is a natural combination of multiple relay and multiple access. Note also the cooperative feature embodied in the optimization over  $p(x_1, x_3)p(x_2, x_4)$ .

What is the coding scheme to achieve (21.9)–(21.13)? Obviously, it should have both the elements of the decode-and-forward scheme as well as the CDMA scheme. Among the several decode-and-forward schemes mentioned in the introduction, the sliding-window decoding is the simplest, while, as noted above, the backward decoding is the most involved and also induces excessive delays. However, it turns out that for the case of multiple sources, only with the backward decoding can the rate region (21.9)–(21.13) be achieved. Neither the sliding-window decoding scheme nor the successive decoding scheme can achieve the same region.

The essential reason for the difference is that backward decoding is a *one-block-decision scheme*, while both the sliding-window decoding and the successive decoding schemes are *multiple-block-decision schemes*. Specifically, for the one-level relay network in Figure 21.4, both the sliding-window decoding and the successive decoding need two consecutive blocks to make one decoding decision. The proof can be found in [41].

In the relay structure shown in Figure 21.4, since node 3 does not decode source 2, the signals transmitted by nodes 2 and 4 cause interference to it. Similarly, nodes 1 and 3 cause interference to node 4. These interferences can also be seen from (21.9) and (21.10). It is possible to change the relay structure to avoid such interferences. For example, we could let node 3 also decode source 2, so that nodes 2 and 4 no longer

cause interference to node 3. This may not be a wise choice, however, since depending on the network topology, it may be even harder for source 2 to reach node 3 than for it to reach the destination node 5.

A special multisource one-relay network has been considered in [42, 43], where multiple sources try to send to the same destination via the same relay node, and the relay node needs to decode all the sources. An achievable rate region using backward decoding was obtained in [43], where both the constraints for the relay and for the destination are like multiple access.

### 21.3.5 General Networks

In this section, we develop a general multisource, multideestination, multirelay scheme for general networks.

Consider a network of  $n$  nodes  $\mathcal{N} = \{1, 2, \dots, n\}$ . We consider the multisource, multicast problem, where there can be more than one source in the network. Each source originates at a single node and may have multiple destinations. Let  $\mathcal{M} = \{1, 2, \dots, m\}$  denote the set of sources. Any source  $k \in \mathcal{M}$  corresponds to a source node  $s^{(k)} \in \mathcal{N}$  and a set of destination nodes  $\mathcal{D}^{(k)} \subset \mathcal{N}$ . The communication task is to send the information of source  $k$  from the source node  $s^{(k)}$  to all the nodes in  $\mathcal{D}^{(k)}$  over the network. Note that the number  $m$  can be greater than  $n$  since multiple sources having different destinations can originate from the same node.

Consider a multirelay route  $\mathcal{N}^{(k)} \subseteq \mathcal{N}$  for each source  $k \in \mathcal{M}$ , where,  $\mathcal{N}^{(k)}$  is an ordered set of nodes starting with  $s^{(k)}$ . For any  $i, j \in \mathcal{N}^{(k)}$ , the order is defined by  $i \prec^{(k)} j$  if node  $i$  is upstream of node  $j$  along the route. Since all the nodes on the multirelay route will obtain the source information, the multicast task is fulfilled as long as the route is chosen such that  $\mathcal{D}^{(k)} \subset \mathcal{N}^{(k)}$ .

Consider a discrete memoryless network channel model described by

$$(\mathcal{X}_1 \times \dots \times \mathcal{X}_n, p(y_1, \dots, y_n | x_1, \dots, x_n), \mathcal{Y}_1 \times \dots \times \mathcal{Y}_n),$$

where  $\mathcal{X}_i$  and  $\mathcal{Y}_i$ ,  $i = 1, \dots, n$  are finite input and output alphabets, respectively, and

$$p(y_1, \dots, y_n | x_1, \dots, x_n)$$

is a probability distribution on  $\mathcal{Y}_1 \times \dots \times \mathcal{Y}_n$  for each  $(x_1, \dots, x_n)$ . At any time  $t = 1, 2, \dots$ , each node  $i \in \mathcal{N}$  sends  $x_i(t) \in \mathcal{X}_i$  into the channel and receives  $y_i(t) \in \mathcal{Y}_i$  from the channel. The distribution of the outputs  $(y_1(t), \dots, y_n(t))$  depends only on the inputs at the time  $t$  via

$$p(y_1(t), \dots, y_n(t) | x_1(t), \dots, x_n(t)).$$

We choose a multirelay route  $\mathcal{N}^{(k)} \subseteq \mathcal{N}$  for each source  $k \in \mathcal{M}$ , such that  $\mathcal{D}^{(k)} \subset \mathcal{N}^{(k)}$ . Along each route, we use the scheme of regular block Markov encoding/backward decoding. These  $m$  routes can be united into a joint backward decoding scheme, if

- (A1) It is possible to assign a nonnegative integer to each node in the network, such that along any multirelay route excluding the source node, the integers are strictly increasing.

We note that this assumption rules out two-way communication as well as other nonacyclic unions of routes, as we note in Remark 21.1.

For any source  $k \in \mathcal{M}$ , introduce an auxiliary random variable  $U_i^{(k)}$  with cardinality equal to  $|\mathcal{X}_i|$  for each node  $i \in \mathcal{N}^{(k)}$ . Loosely speaking,  $U_i^{(k)}$  stands for the information node  $i$  has of source  $k$ . Denote

$$\mathcal{U}^{(k)} = \{U_j^{(k)} : j \in \mathcal{N}^{(k)}\}.$$

Since  $\mathcal{N}^{(k)}$  is an ordered set of nodes as defined above,  $\mathcal{U}^{(k)}$  is an ordered list of random variables. Consequently, define

$$\begin{aligned}\mathcal{U}_{i-}^{(k)} &= \{U_j^{(k)} : j \prec i, j \in \mathcal{N}^{(k)}\}, \\ \mathcal{U}_{i+}^{(k)} &= \{U_j^{(k)} : i \prec j, j \in \mathcal{N}^{(k)}\}.\end{aligned}$$

For any node  $i \in \mathcal{N}$ , denote by  $\mathcal{M}_i := \{k : i \in \mathcal{N}^{(k)}\}$  the set of all the sources with the multirelay route passing through node  $i$ . For any  $\mathcal{S} \subseteq \mathcal{M}_i$ , let

$$\begin{aligned}\mathcal{U}_i^{(\mathcal{S})} &= \{U_i^{(k)} : k \in \mathcal{S}\}, \\ \mathcal{U}^{(\mathcal{S})} &= \bigcup_{k \in \mathcal{S}} \mathcal{U}^{(k)}, \\ \mathcal{U}_{i-}^{(\mathcal{S})} &= \bigcup_{k \in \mathcal{S}} \mathcal{U}_{i-}^{(k)}, \\ \mathcal{U}_{i+}^{(\mathcal{S})} &= \bigcup_{k \in \mathcal{S}} \mathcal{U}_{i+}^{(k)}.\end{aligned}$$

Then we have the following characterization of the  $m$  rates simultaneously achievable along the  $m$  multirelay routes by a joint backward decoding scheme [41].

**Theorem 21.2** *Under assumption (A1), a rate vector  $R^{(\mathcal{M})} = (R^{(1)}, R^{(2)}, \dots, R^{(m)})$  is achievable if there exists some product distribution*

$$\prod_{k \in \mathcal{M}} p(u_j^{(k)}, j \in \mathcal{N}^{(k)}),$$

and some functions

$$x_i = f_i(u_i^{(k)}, k \in \mathcal{M}_i), \quad i \in \mathcal{N},$$

such that for any node  $i \in \mathcal{N}$  and any  $\mathcal{S} \subseteq \mathcal{M}_i$ ,

$$\sum_{k \in \mathcal{S}} R^{(k)} < I(\mathcal{U}_{i-}^{(\mathcal{S})}; Y_i | \mathcal{U}_i^{(\mathcal{S})}, \mathcal{U}_{i+}^{(\mathcal{S})}, \mathcal{U}^{(\mathcal{M}_i \setminus \mathcal{S})}). \quad (21.14)$$

**Remark 21.1** Assumption (A1) is necessary for us to be able to merge multiple routes into a joint backward decoding scheme. To see this, consider the network in Figure 21.5,



**Figure 21.5** Two-way multirelay network.

where there are two routes:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$  and  $4 \rightarrow 3 \rightarrow 2 \rightarrow 1$ . Obviously, assumption (A1) does not hold for this network since it is impossible to assign integers  $j_2$  to node 2 and  $j_3$  to node 3 so that both  $j_2 < j_3$  and  $j_2 > j_3$  hold. Since in the backward decoding scheme, a node must decode at a faster frequency than the node after it, it creates a conflict in that node 2 needs to decode more frequently than node 3 (along the route  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ ), while node 3 too needs to decode more frequently than node 2 (along the route  $4 \rightarrow 3 \rightarrow 2 \rightarrow 1$ ).

### 21.3.6 Application to Sensor Networks

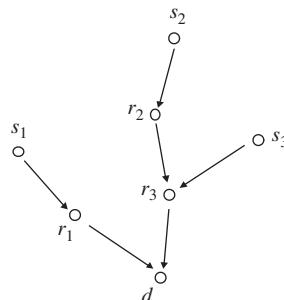
Some sensor networks consist of several sensor-equipped nodes collecting independent information that needs to be communicated to a designated collector or sink node; called the data downloading problem in [44]. Such sensor networks can accordingly be modeled as multisource, single-sink networks.

In the example depicted in Figure 21.6, three source nodes,  $s_1$ ,  $s_2$ , and  $s_3$ , desire to send information to the same destination node  $d$ , using, as relays, the nodes  $r_1$ ,  $r_2$ , and  $r_3$ . By Theorem 21.2, the rate vector  $(R^{(1)}, R^{(2)}, R^{(3)})$  is achievable if there exist some product distribution

$$p(u_{s_1}^{(1)}, u_{r_1}^{(1)}, u_d^{(1)}) p(u_{s_2}^{(2)}, u_{r_2}^{(2)}, u_{r_3}^{(2)}, u_d^{(2)}) p(u_{s_3}^{(3)}, u_{r_3}^{(3)}, u_d^{(3)}),$$

and some functions

$$\begin{aligned} x_{s_1} &= f_{s_1}(u_{s_1}^{(1)}), & x_{s_2} &= f_{s_2}(u_{s_2}^{(2)}), & x_{s_3} &= f_{s_3}(u_{s_3}^{(3)}), \\ x_{r_1} &= f_{r_1}(u_{r_1}^{(1)}), & x_{r_2} &= f_{r_2}(u_{r_2}^{(2)}), \\ x_{r_3} &= f_{r_3}(u_{r_3}^{(2)}, u_{r_3}^{(3)}), \\ x_d &= f_d(u_d^{(1)}, u_d^{(2)}, u_d^{(3)}), \end{aligned}$$



**Figure 21.6** Multisource, single-sink sensor network.

such that for node  $r_1$ ,

$$R^{(1)} < I(U_{s_1}^{(1)}; Y_{r_1}|U_{r_1}^{(1)}, U_d^{(1)});$$

for node  $r_2$ ,

$$R^{(2)} < I(U_{s_2}^{(2)}; Y_{r_2}|U_{r_2}^{(2)}, U_{r_3}^{(2)}, U_d^{(2)});$$

for node  $r_3$ ,

$$\begin{aligned} R^{(2)} &< I(U_{s_2}^{(2)}, U_{r_2}^{(2)}; Y_{r_3}|U_{r_3}^{(2)}, U_d^{(2)}, U_{s_3}^{(3)}, U_{r_3}^{(3)}, U_d^{(3)}), \\ R^{(3)} &< I(U_{s_3}^{(3)}; Y_{r_3}|U_{r_3}^{(3)}, U_d^{(3)}, U_{s_2}^{(2)}, U_{r_2}^{(2)}, U_{r_3}^{(2)}, U_d^{(2)}), \\ R^{(2)} + R^{(3)} &< I(U_{s_2}^{(2)}, U_{r_2}^{(2)}, U_{s_3}^{(3)}; Y_{r_3}|U_{r_3}^{(2)}, U_d^{(2)}, U_{r_3}^{(3)}, U_d^{(3)}); \\ R^{(1)} &< I(U_{s_1}^{(1)}, U_{r_1}^{(1)}; Y_d|U_d^{(1)}, U_{s_2}^{(2)}, U_{r_2}^{(2)}, U_{r_3}^{(2)}, U_d^{(2)}, U_{s_3}^{(3)}, U_d^{(3)}), \\ R^{(2)} &< I(U_{s_2}^{(2)}, U_{r_2}^{(2)}, U_{r_3}^{(2)}; Y_d|U_d^{(2)}, U_{s_1}^{(1)}, U_{r_1}^{(1)}, U_d^{(1)}, U_{s_3}^{(3)}, U_{r_3}^{(3)}, U_d^{(3)}), \\ R^{(3)} &< I(U_{s_3}^{(3)}, U_{r_3}^{(3)}; Y_d|U_d^{(3)}, U_{s_1}^{(1)}, U_{r_1}^{(1)}, U_d^{(1)}, U_{s_2}^{(2)}, U_{r_2}^{(2)}, U_{r_3}^{(2)}, U_d^{(2)}), \\ R^{(1)} + R^{(2)} &< I(U_{s_1}^{(1)}, U_{r_1}^{(1)}, U_{s_2}^{(2)}, U_{r_2}^{(2)}, U_{r_3}^{(2)}; Y_d|U_d^{(1)}, U_d^{(2)}, U_{s_3}^{(3)}, U_{r_3}^{(3)}, U_d^{(3)}), \\ R^{(1)} + R^{(3)} &< I(U_{s_1}^{(1)}, U_{r_1}^{(1)}, U_{s_3}^{(3)}, U_{r_3}^{(3)}; Y_d|U_d^{(1)}, U_d^{(3)}, U_{s_2}^{(2)}, U_{r_2}^{(2)}, U_{r_3}^{(2)}, U_d^{(2)}), \\ R^{(2)} + R^{(3)} &< I(U_{s_2}^{(2)}, U_{r_2}^{(2)}, U_{r_3}^{(2)}, U_{s_3}^{(3)}; Y_d|U_d^{(2)}, U_d^{(3)}, U_{s_1}^{(1)}, U_{r_1}^{(1)}, U_d^{(1)}), \\ R^{(1)} + R^{(2)} + R^{(3)} &< I(U_{s_1}^{(1)}, U_{r_1}^{(1)}, U_{s_2}^{(2)}, U_{r_2}^{(2)}, U_{r_3}^{(2)}, U_{s_3}^{(3)}, U_{r_3}^{(3)}; Y_d|U_d^{(1)}, U_d^{(2)}, U_d^{(3)}), \end{aligned} \tag{21.15}$$

while, for node  $d$ , the inequalities (21.15) hold.

The set of inequalities characterizing the achievable rate region may seem very complicated, especially if there are many multirelay routes crisscrossing each other in the network. However, the rules to follow when writing down the inequalities (21.14) for each node are actually quite simple. For any node, only the routes passing through it are of any concern, and all the other routes with the corresponding sources and inputs appear invisible. If a node  $i$  is on only one route, say,  $\mathcal{N}^{(k_i)}$  of the source  $k_i$ , then only one inequality applies:

$$R^{(k_i)} < I(\mathcal{U}_{i-}^{(k_i)}, Y_i|U_i^{(k_i)}, \mathcal{U}_{i+}^{(k_i)}),$$

where  $\mathcal{U}_{i-}^{(k_i)}$  and  $\mathcal{U}_{i+}^{(k_i)}$  are the inputs (corresponding to the source  $k_i$ ) of the upstream nodes and the downstream nodes, respectively. Actually,  $U_i^{(k_i)}$  can be equivalently replaced by  $X_i$  since node  $i$  has no other auxiliary inputs. On the other hand, for a node at the intersection of  $\ell > 1$  routes, its environment is similar to a multiple-access channel with  $\ell$  sources: with each source providing a set of inputs of the corresponding upstream nodes, while the inputs of the downstream nodes are known.

**Remark 21.2** *The advantage of applying the multisource, multirelay scheme to such multisource, single-sink networks, as the one shown in Figure 21.3, is obvious. Since multiple sources converge to a single sink, traffic gets concentrated and increases as one*

gets closer to the sink. If a traditional multihop scheme that does not exploit information theory is used, then the bottleneck of the whole network would be the area around the sink where the links carry the heaviest traffic. However, by utilizing a multirelay scheme, each node makes use of the inputs of all the upstream nodes. As one gets closer to the sink, there are more upstream nodes to help, which means higher received signal power and thus higher achievable rates. For the sink node especially, all the inputs of all the other nodes can be used. Importantly, and fortunately, we note that for networks with such a tree structure, condition (A1) does indeed hold, which means that joint backward decoding can be used.

## 21.4 WIRELESS NETWORK CODING

Network coding [45] has attracted a lot of research interest in recent years and is expected to result in basic changes to the communication strategies used in networks. Besides a wide variety of potential applications, the simplicity of the essential ideas of network coding also contributes to its success.

The basic idea of network coding can be explained using the simple network depicted in Figure 21.7, where node A has two bits of information  $b_1$  and  $b_2$  to transmit to nodes B and C, respectively. However, if node B already knows  $b_2$  and node C already knows  $b_1$ , then instead of transmitting two bits  $b_1$  and  $b_2$  separately, node A only needs to transmit one bit  $b_1 \oplus b_2$  to nodes B and C since node B can recover  $b_1$  by computing  $(b_1 \oplus b_2) \oplus b_2 = b_1$ , and similarly, node C can recover  $b_2$  by computing  $(b_1 \oplus b_2) \oplus b_1 = b_2$ .

An interesting observation of the above network coding scheme is that although only one bit is transmitted by node A, two different bits can be recovered at nodes B and C. Of course, this works only if nodes B and C have the appropriate side information. Hence, the success of network coding crucially depends on the availability of side information. Fortunately, in many communication networks, the presence of such side information is a common phenomenon. For example, in networks with multiple routes, side information may come from other routes.

We address a more general framework depicted in Figure 21.8, where, motivated by wireless communications, the channel dynamics at the physical layer is modeled as a broadcast channel  $(\mathcal{X}_0, p(y_1, y_2|x_0), \mathcal{Y}_1 \times \mathcal{Y}_2)$ , with one input  $x_0$  transmitted by node A, and two outputs  $y_1$  and  $y_2$  received by nodes B and C, respectively. Obviously, this includes the network in Figure 21.7 as a special case by setting  $y_1 = y_2 = x_0$ .

Consider the same problem where node A has two independent messages  $s_1$  and  $s_2$  to send to nodes B and C, respectively, while node B already knows  $s_2$ , and node C

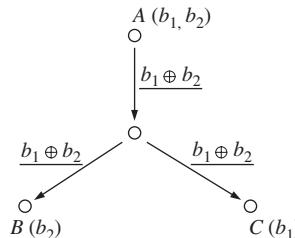
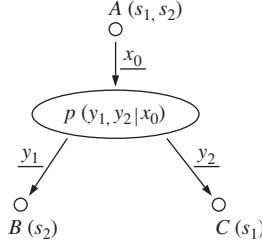


Figure 21.7 Idea of network coding.



**Figure 21.8** Broadcast channel with side information at the receivers.

already knows  $s_1$ . Now, an immediate question is whether the idea of network coding can be applied to this more general framework, and what the corresponding achievable rates are.

It turns out that in this setting, the idea of network coding can be applied with random binning, a classical and fundamental technique in multiuser information theory, and the corresponding achievable rates are any  $(R_1, R_2)$  satisfying

$$R_1 < I(X_0, Y_1), \quad (21.16)$$

$$R_2 < I(X_0, Y_2), \quad (21.17)$$

for any input distribution  $p(x_0)$ , where,  $R_1$  is the rate of sending  $s_1$  to node  $B$ , and  $R_2$  is the rate of sending  $s_2$  to node  $C$ .

Similarly, an interesting observation from (21.16) and (21.17) is that independent messages can be sent to two receivers, simultaneously, at their respective link capacities, by the same input. In addition, more generally than the network coding scheme used in Figure 21.7, the rates  $R_1$  and  $R_2$  can be different.

The achievability of (21.16) and (21.17) will be demonstrated in a more general setting discussed in Section 21.4.1. As applications of this generalized idea of network coding, two-way relay channels with one or two relays will be discussed in Section 21.4.2.

#### 21.4.1 Broadcast Channel with Side Information at Receivers

Consider a discrete memoryless broadcast channel with one transmitter and  $m$  receivers:

$$(\mathcal{X}_0, p(y_1, \dots, y_m | x_0), \mathcal{Y}_1 \times \dots \times \mathcal{Y}_m). \quad (21.18)$$

That is, at any time instant  $t = 1, 2, \dots$ , the transmitter sends  $X_0(t) \in \mathcal{X}_0$ , and each receiver  $i \in \{1, \dots, m\}$  receives  $Y_i(t) \in \mathcal{Y}_i$ , according to  $p(Y_1(t), \dots, Y_m(t) | X_0(t))$ .

Consider the problem where the transmitter wants to send independent messages to different receivers, while each receiver knows a priori the messages for the other receivers.

For brevity, the standard definitions of codes and achievable rates are omitted, except a special note that each receiver  $i \in \{1, \dots, m\}$  decodes based on  $(Y_i(1), \dots, Y_i(T))$  and  $W_{\{-i\}} = (W_1, \dots, W_{i-1}, W_{i+1}, \dots, W_m)$ , that is, the messages for the other receivers. We have the following theorem [46]:

**Theorem 21.3** *For the broadcast channel (21.18), with each receiver knowing a priori the messages for the other receivers, any rates  $(R_1, R_2, \dots, R_m)$  satisfying the following inequalities are simultaneously achievable:*

$$R_i < I(X_0; Y_i), \quad i = 1, 2, \dots, m, \quad (21.19)$$

for some  $p(x_0)$ .

Next, consider an extension to the case of correlated sources.

Consider  $m$  independent and identically distributed (i.i.d.) random processes  $\{S_i(t), t = 1, 2, \dots\}$ , for  $i = 1, \dots, m$ , with joint distribution  $p(s_1, s_2, \dots, s_m)$ . Suppose each random process  $\{S_i(t), t = 1, 2, \dots\}$  is available to all the receivers except receiver  $i$ , for  $i = 1, 2, \dots, m$ , while the transmitter knows all the  $m$  random processes. The communication task is for the transmitter to send to each receiver  $i$  the information about  $\{S_i(t), t = 1, 2, \dots\}$ . The following theorem characterizes the condition under which this can be done simultaneously for all the receivers.

**Theorem 21.4** *For the communication problem stated above, all the receivers can obtain their respective information through the broadcast channel simultaneously if for some  $p(x_0)$ ,*

$$H(S_i | S_{\{-i\}}) < I(X_0; Y_i), \quad i = 1, 2, \dots, m, \quad (21.20)$$

where  $S_{\{-i\}} := \{S_1, \dots, S_{i-1}, S_{i+1}, \dots, S_m\}$ .

### 21.4.2 Two-Way Relay Channel

The two-way relay channel [47] is often cited to explain the benefits of network coding for the wireless setting. In this channel, two sources communicate with each other with the help of a relay using the network-coding protocol. First, the relay linearly combines (or XORs) the two source messages it has decoded and broadcasts the result. Then the source nodes, knowing what they transmitted in the past, can recover each other's message by decoding the relay transmission and inverting the linear operation. Thus, the flow of information is uninterrupted despite the common channel between the relay and sources.

More general results can be obtained when applied in conjunction with the binning technique. Consider a network of three nodes 1, 2, 3, with the input–output dynamics modeled by the discrete memoryless channel:

$$(\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3, p(y_1, y_2, y_3 | x_1, x_2, x_3), \mathcal{Y}_1 \times \mathcal{Y}_2 \times \mathcal{Y}_3). \quad (21.21)$$

That is, at any time  $t = 1, 2, \dots$ , the outputs  $y_1(t)$ ,  $y_2(t)$ , and  $y_3(t)$  received by the three nodes, respectively, only depend on the inputs  $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$  transmitted by the three nodes at the same time, according to  $p(y_1(t), y_2(t), y_3(t) | x_1(t), x_2(t), x_3(t))$ .

Consider the two-way relay problem where node 1 and node 3 communicate with each other at rates  $R_1$  and  $R_3$ , respectively, with the help of the relay node 2.

We are interested in the simultaneously achievable rates  $(R_1, R_3)$ . Here, the standard definitions of codes and achievable rates are omitted, except to especially note that at any time  $t$ , each node  $i$  can choose its input  $x_i(t)$  based on the past outputs ( $y_i(t-1)$ ,  $y_i(t-2)$ ,  $\dots$ ,  $y_i(1)$ ) it has already received. The following result is obtained in [46].

**Theorem 21.5** For the two-way relay problem defined above, any rates  $(R_1, R_3)$  satisfying the following inequalities are simultaneously achievable:

$$R_1 < I(X_1, X_2; Y_3|X_3), \quad (21.22)$$

$$R_3 < I(X_2, X_3; Y_1|X_1), \quad (21.23)$$

and

$$R_1 < I(X_1; Y_2|X_2, X_3), \quad (21.24)$$

$$R_3 < I(X_3; Y_2|X_1, X_2), \quad (21.25)$$

$$R_1 + R_3 < I(X_1, X_3; Y_2|X_2), \quad (21.26)$$

for any  $p(x_1)p(x_2)p(x_3)$ .

As indicated by the product form of the joint input distribution above, a consequence of the network-coding strategy is that the inputs of different nodes must be independent, thus precluding the use of beamforming. However, if we use superposition coding at the relay for different sources, instead of binning different sources into the same signal, then the inputs of different nodes can be correlated, allowing the use of beamforming. Such a scheme for this two-way relay network was first studied in [48], and the following achievable region was obtained in [49]:

$$R_1 < I(X_1, X_2; Y_3|U_3, X_3), \quad (21.27)$$

$$R_3 < I(X_2, X_3; Y_1|U_1, X_1), \quad (21.28)$$

and

$$R_1 < I(X_1; Y_2|U_1, U_3, X_2, X_3), \quad (21.29)$$

$$R_3 < I(X_3; Y_2|U_1, U_3, X_1, X_2), \quad (21.30)$$

$$R_1 + R_3 < I(X_1, X_3; Y_2|U_1, U_3, X_2), \quad (21.31)$$

for any  $p(u_1)p(u_3)p(x_1|u_1)p(x_2|u_1, u_3)p(x_3|u_3)$ , where  $U_1$  and  $U_3$  are auxiliary random variables.

Depending on the channel parameters, neither of the two achievable regions (21.22)–(21.26) and (21.27)–(21.31) is always superior than the other. The advantage of the binning strategy is that all the relay power is fully used by both sources, while it is only partially used by either source if superposition coding is applied. However, superposition coding allows cooperation at the signal level, which can boost the received power by coherent transmission. More generally, we can consider the combination of these two schemes, especially for nonsymmetric situations.

The achievable rate region in Theorem 21.4 has a simple interpretation. Inequalities (21.22) and (21.23) are cut-set bounds without beamforming. Inequalities (21.24)–(21.26) are the multiple-access region if the relay is to fully decode both sources.

Now, an immediate question is whether this interpretation extends to the two-relay case. More specifically, we would like to know if the following rates are achievable in the two-relay setting:

$$R_1 < I(X_1, X_2, X_3; Y_4|X_4), \quad (21.32)$$

$$R_4 < I(X_2, X_3, X_4; Y_1|X_1), \quad (21.33)$$

and

$$R_1 < I(X_1; Y_2|X_2, X_3, X_4), \quad (21.34)$$

$$R_4 < I(X_3, X_4; Y_2|X_1, X_2), \quad (21.35)$$

$$R_1 + R_4 < I(X_1, X_3, X_4; Y_2|X_2), \quad (21.36)$$

and

$$R_1 < I(X_1, X_2; Y_3|X_3, X_4), \quad (21.37)$$

$$R_4 < I(X_4; Y_3|X_1, X_2, X_3), \quad (21.38)$$

$$R_1 + R_4 < I(X_1, X_2, X_4; Y_3|X_3), \quad (21.39)$$

where nodes 1 and 4 are the sources and nodes 2 and 3 are the relays.

Notice that (21.32) and (21.33) correspond to the cut-set bounds. Furthermore, (21.34)–(21.36) and (21.37)–(21.39) seem reasonable extensions of the multiple-access constraints to each relay node. Unfortunately, there is a fundamental difficulty in achieving (21.32)–(21.39). To achieve (21.34)–(21.36), node 3 needs to decode ahead of node 2 in order to help. However, the reverse is also needed for (21.37)–(21.39). This “deadlock” problem was identified in [41], as noted in Remark 21.1, when a backward decoding scheme was tried for achieving (21.32)–(21.39).

It turns out [49] that the deadlock problem can be resolved by adding an additional constraint to (21.32)–(21.39): At least one of the following inequalities hold:

$$R_1 < I(X_1; Y_2|X_2, X_3), \quad (21.40)$$

$$R_4 < I(X_4; Y_3|X_2, X_3), \quad (21.41)$$

$$R_1 + R_4 < \max\{I(X_1, X_4; Y_2|X_2, X_3), I(X_1, X_4; Y_3|X_2, X_3)\}. \quad (21.42)$$

Simply put into words, any one of these inequalities ensures that some relay can decode at least one of the most recently transmitted source messages. Intuitively, this requirement is reasonable. To start the flow of information, it is expected that at least one relay should decode a message being transmitted in the current block.

There are two ways in which a coding scheme satisfies the additional constraint: in the first case some relay decodes *one* source before the other relay, and in the second case some relay decodes *both* sources before the other relay. For each case, coding schemes can be developed that recover the region defined by (21.32)–(21.39). A key ingredient in some of the coding schemes is an offset-encoding strategy developed in [50] that gives more flexibility when combined with sliding-window decoding.

## 21.5 CONCLUDING REMARKS

We have presented an account of some of what can be considered information theoretically concerning transmission of data from source nodes to destination nodes in sensor networks. These include upper bounds, which address the impossibility results, relay schemes, which exploit “interferences,” and network coding or binning strategies, which exploit source correlation and side information. The general message is that there are fundamental limits in sensor networking and also wonderful opportunities to explore sophisticated communication schemes. Much remains to be done.

## ACKNOWLEDGMENTS

This material is based upon work partially supported by the NSERC of Canada, the USARO under Contract Nos. W911NF-08-1-0238 and W-911-NF-0710287, and NSF under Contract Nos. NSF ECCS-0701604, CNS-07-21992, NSF CNS 05-19535, NSF CNS-0626584, and CCR-0325716.

## REFERENCES

1. B. M. Sadler, "Fundamentals of energy-constrained sensor network systems," *IEEE Aerospace Electron. Syst. Mag.*, 2005.
2. W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Networking*, vol. 12, pp. 493–506, June 2004.
3. IEEE Protocol 802.11, "Standard for wireless LAN: Medium access control (MAC) and physical layer (PHY) specifications," IEEE, July 1996.
4. P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, pp. 388–404, Mar. 2000.
5. L.-L. Xie and P. R. Kumar, "A network information theory for wireless communication: Scaling laws and optimal operation," *IEEE Trans. Inform. Theory*, vol. 50, pp. 748–767, May 2004.
6. A. Jovicic, P. Viswanath, and S. R. Kulkarni, "Upper bounds to transport capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 50, pp. 2555–2565, Nov. 2004.
7. S. Ahmad, A. Jovicic, and P. Viswanath, "Outer bounds to the capacity region of wireless networks," *IEEE Trans. Inform. Theory*, vol. 52, pp. 2770–2776, June 2006.
8. L.-L. Xie and P. R. Kumar, "On the path-loss attenuation regime for positive cost and linear scaling of transport capacity in wireless networks," *Joint Special Issue of IEEE Trans. Inform. Theory and IEEE/ACM Trans. Networking Inform. Theory*, vol. 52, no. 6, pp. 2313–2328, June 2006.
9. O. Leveque and E. Telatar, "Information theoretic upper bounds on the capacity of large extended ad hoc wireless networks," *IEEE Trans. Inform. Theory*, vol. 51, no. 3, pp. 858–865, Mar. 2005.
10. A. Ozgur, O. Leveque, and D. Tse, "Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks," *IEEE Trans. Inform. Theory*, vol. 53, no. 10, pp. 3549–3572, Oct. 2007.
11. A. Ozgur, O. Leveque, and E. Preissmann, "Scaling laws for one- and two-dimensional random wireless networks in the low-attenuation regime," *IEEE Trans. Inform. Theory*, vol. 53, no. 10, pp. 3573–3586, Oct. 2007.
12. T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
13. D. Marco, E. J. Duarte-Melo, M. Liu, and D. L. Neuhoff, "On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data," *Inform. Process. Sensor Networks*, 2003.
14. H. Gupta, V. Navda, S. R. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2005.
15. A. F. Dana and B. Hassibi, "On the power efficiency of sensory and ad-hoc wireless networks," *IEEE Trans. Inform. Theory*, vol. 52, no. 7, pp. 2890–2914, 2006.

16. H. El Gamal, "On the scaling laws of dense wireless sensor networks," *IEEE Trans. Inform. Theory*, vol. 51, no. 3, pp. 1229–1234, 2005.
17. M. Gastpar and M. Vetterli, "Power, spatio-temporal bandwidth, and distortion in large sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 745–754, 2005.
18. E. J. Duarte-Melo and M. Liu, "Data-gathering wireless sensor networks: Organization and capacity," *Computer Networks (COMNET) Special Issue on Wireless Sensor Networks*, vol. 43, no. 4, pp. 519–537, 2003.
19. Y. Rachlin, R. Negi, and P. Khosla, "Sensing capacity for discrete sensor network applications," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, 2005.
20. S. C. Draper and G. W. Wornell, "Side information aware coding strategies for sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, pp. 966–976, 2004.
21. S. Adireddy and L. Tong, "Exploiting decentralized channel state information for random access," *IEEE Trans. Inform. Theory*, vol. 51, no. 2, pp. 537–561, 2005.
22. W. U. Bajwa, J. D. Haupt, A. M. Sayeed, and R. D. Nowak, "Joint source-channel communication for distributed estimation in sensor networks," *IEEE Trans. Inform. Theory*, vol. 53, no. 10, pp. 3629–3653, 2007.
23. E. C. van der Meulen, "Transmission of information in a  $T$ -terminal discrete memoryless channel," PhD thesis, Department of Statistics, University of California, Berkeley, 1968.
24. E. C. van der Meulen, "Three-terminal communication channels," *Adv. Appl. Prob.*, vol. 3, pp. 120–154, 1971.
25. T. Cover and A. El Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inform. Theory*, vol. 25, pp. 572–584, 1979.
26. T. Cover and S. K. Leung, "An achievable rate region for the multiple access channel with feedback," *IEEE Trans. Inform. Theory*, vol. 27, no. 3, pp. 292–298, 1981.
27. M. R. Aref, "Information flow in relay networks," PhD thesis, Stanford University, Stanford, CA, 1980.
28. P. Gupta and P. R. Kumar, "Towards an information theory of large networks: An achievable rate region," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1877–1894, Aug. 2003.
29. A. Reznik, S. R. Kulkarni, and S. Verdú, "Degraded Gaussian multirelay channel: Capacity and optimal power allocation," *IEEE Trans. Inform. Theory*, vol. 50, pp. 3037–3046, Dec. 2004.
30. G. Kramer, M. Gastpar, and P. Gupta, "Capacity theorems for wireless relay channels," in *Proc. 41th Annual Allerton Conf. on Commun., Control, and Computing*, Monticello, IL, Oct. 2003.
31. L.-L. Xie and P. R. Kumar, "An achievable rate for the multiple-level relay channel," *IEEE Trans. Inform. Theory*, vol. 51, pp. 1348–1358, Apr. 2005.
32. G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Trans. Inform. Theory*, vol. 51, pp. 3037–3063, Sept. 2005.
33. A. B. Carleial, "Multiple-access channels with different generalized feedback signals," *IEEE Trans. Inform. Theory*, vol. 28, pp. 841–850, Nov. 1982.
34. F. M. J. Willems and E. C. van der Meulen, "The discrete memoryless multiple-access channel with cribbing encoders," *IEEE Trans. Inform. Theory*, vol. 31, pp. 313–327, 1985.
35. C.-M. Zeng, F. Kuhlmann, and A. Buzo, "Achievability proof of some multiuser channel coding theorems using backward decoding," *IEEE Trans. Inform. Theory*, vol. 35, pp. 1160–1165, 1989.
36. P. Razaghi and W. Yu, "Parity forwarding for multiple-relay networks," in *Proceedings of the IEEE Symposium on Information Theory*, Seattle, WA, July 9–14, 2006.

37. B. Schein, "Distributed coordination in network information theory," PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2001.
38. R. Ahlswede, "Multi-way communication channels," in *Proceedings of the 2nd Int. Symp. Inform. Theory (Tsahkadsor, Armenian S.S.R.)*, Prague: Publishing House of the Hungarian Academy of Sciences, 1971, pp. 23–52.
39. H. Liao, "Multiple access channels," PhD thesis, Department of Electrical Engineering, University of Hawaii, Honolulu, 1972.
40. R. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
41. L.-L. Xie and P. R. Kumar, "Multi-source, multi-destination, multi-relay wireless networks," *IEEE Trans. Inform. Theory*, vol. 53, no. 10, pp. 3586–3895, Oct. 2007.
42. G. Kramer and A. J. van Wijngaarden, "On the white Gaussian multiple-access relay channel," in *Proceedings of the IEEE Symposium on Information Theory*, Sorrento, Italy, June 2000, p. 40.
43. G. Kramer, M. Gastpar, and P. Gupta, "Information-theoretic multi-hopping for relay networks," in *International Zurich Seminar on Communications*, ETH Zurich, Switzerland, Feb. 2004, pp. 192–195.
44. A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 755–764, Apr. 2005.
45. R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1204–1216, July 2000.
46. L.-L. Xie, "Network coding and random binning for multi-user channels," in *Proc. IEEE Canadian Workshop on Information Theory (CWIT)*, Edmonton, Canada, June 2007.
47. Y. Wu, P. A. Chou, and S.-Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," in *Proceedings of the 2005 Conf. on Information Sciences and Systems*, The Johns Hopkins University, Mar. 2005.
48. B. Rankov and A. Wittneben, "Achievable rate regions for the two-way relay channel," in *Proc. IEEE Int. Symposium on Information Theory (ISIT)*, Seattle, WA, July 2006.
49. J. Ponniah and L.-L. Xie, "An achievable rate for the two-way two-relay channel," *IEEE Int. Symp. Inform. Theory (ISIT)*, Toronto, July 2008, submitted for publication.
50. L. Sankar, G. Kramer, and N. B. Mandayam, "Offset encoding for multiaccess relay channels," *IEEE Trans. Inform. Theory*, vol. 53, no. 10, pp. 3814–3821, Oct. 2007.



---

**PART IV**

---

## **NOVEL TECHNIQUES FOR AND APPLICATIONS OF DISTRIBUTED SENSOR NETWORKS**



## CHAPTER 22

---

# Distributed Adaptive Learning Mechanisms

Ali H. Sayed and Federico S. Cattivelli

Electrical Engineering Department, University of California, Los Angeles, California

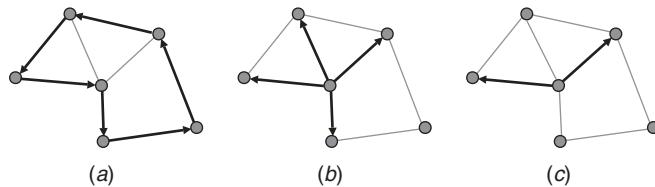
### 22.1 INTRODUCTION

Distributed networks linking sensors and actuators will form the backbone of future data communication and control networks. Applications will range from sensor networks to precision agriculture, environment monitoring, disaster relief management, smart spaces, target localization, as well as medical applications [1–5]. In all these cases, the distribution of the nodes in the field yields spatial diversity, which should be exploited alongside the temporal dimension in order to enhance the robustness of the processing tasks and improve the probability of signal and event detection.

Distributed processing techniques allow for the efficient extraction of temporal and spatial information from data collected at such distributed nodes by relying on local cooperation and data processing. For example, each node in the network could collect noisy observations related to a certain parameter of interest. The nodes would then interact with their neighboring nodes, as dictated by the network topology, in order to estimate the parameter. The objective is to arrive at an estimate that is as reliable as the one that would be obtained if each node had access to the information across the entire network.

In contrast, in the centralized approach to parameter estimation, the data from all nodes would be conveyed to a central processor where they would be fused and the vector of parameters estimated. Such an approach requires sufficient communications resources to transmit the data back and forth between the nodes and the central processor, which would limit the autonomy of the network besides adding a critical point of failure in the network due to the presence of a central node [1, 6].

This chapter describes recent development in distributed processing over adaptive networks. The presentation covers adaptive algorithms that allow neighboring nodes to communicate with each other at every iteration. At each node, estimates exchanged with neighboring nodes are fused and promptly fed into the local adaptation rules. In this way, an adaptive network is obtained where the structure as a whole is able to respond in real time to the temporal and spatial variations in the statistical profile of the data.



**Figure 22.1** Three modes of cooperation: (a) incremental, (b) diffusion, and (c) probabilistic diffusion.

Different adaptation or learning rules at the nodes, allied with different cooperation protocols, give rise to adaptive networks of various complexities and potential.

Obviously, the effectiveness of any distributed implementation depends on the modes of cooperation that are allowed among the nodes. Figure 22.1 illustrates three such modes of cooperation.

In an incremental mode of cooperation (see Fig. 22.1a), information flows in a sequential manner from one node to the adjacent node. This mode of operation requires a cyclic pattern of collaboration among the nodes and has the advantage that for the last node in the cycle, the data from the entire network are used to update the desired parameter estimate, thereby offering excellent estimation performance. Moreover, for every measurement, every node needs to communicate with only one neighbor. However, incremental cooperation has the disadvantage of requiring the definition of a cycle, and network processing has to be faster than the measurement process since a full communication cycle is needed for every measurement. This may become prohibitive for large networks. Incremental networks are also less robust to node and link failures.

An alternative protocol is the diffusion implementation (see Fig. 22.1b) where every node communicates with all of its neighbors as dictated by the network topology. This approach has no topology constraints and is more robust to node and link failure (see, e.g., [7]). It will have some performance degradation compared to an incremental solution, and also every node will need to communicate with its neighbors for every measurement, possibly requiring more energy than the incremental case. Note, however, that when omnidirectional communications are used, the energy required to transmit to one neighbor may be the same as that required to transmit to every neighbor.

The communication in the diffusion solution can be reduced by allowing each node to communicate only with a subset of its neighbors. This mode of cooperation is denoted probabilistic diffusion (see Fig. 22.1c). The choice of which subset of neighbors to communicate with can be randomized according to some performance criterion.

This chapter describes several developments in distributed processing over adaptive networks based on the works [8–18]. The resulting adaptive learning rules rely on local data at the individual nodes and on collaborations among neighboring nodes in order to exploit the space–time dimension of the data more fully. The ideas are illustrated by considering algorithms of the least-mean-squares (LMS) type, although more general adaptation rules are also possible including least-squares rules and Kalman-type rules [11, 15, 16, 18]. Both incremental and diffusion strategies are considered in the sequel.

### 22.1.1 Notation

In the remainder of the chapter we use boldface italic letters for random quantities and normal font for nonrandom (deterministic) quantities. We also use capital letters for

matrices and small letters for vectors. For example,  $\mathbf{d}$  is a random quantity and  $d$  is a realization or measurement for it, and  $R$  is a covariance matrix while  $w$  is a weight vector. The notation  $*$  denotes complex-conjugation for scalars and complex-conjugate transposition for matrices. The index  $i$  is used to denote iterations or time instants, and the indices  $k$  and  $\ell$  are used to denote different nodes in a network with a total of  $N$  nodes.

## 22.2 MOTIVATION

We motivate adaptive networks by examining an application in the context of data modeling. Thus, consider a set of  $N$  sensors scattered over a geographical area and observing some physical phenomenon of interest. Each node  $k$  collects a measurement  $d_k(i)$  at time  $i$ . It is assumed that these measurements arise from an autoregressive (AR) model of the form:

$$d_k(i) = \sum_{m=1}^M \beta_m d_k(i-m) + v_k(i), \quad (22.1)$$

where  $v_k(i)$  denotes additive zero-mean noise and the coefficients  $\{\beta_m\}$  represent the underlying model. If we define the  $M \times 1$  parameter vector

$$w^o = \text{col}\{\beta_1, \beta_2, \dots, \beta_M\}, \quad (M \times 1)$$

and the  $1 \times M$  regression vector

$$u_{k,i} = [d_k(i-1) \quad d_k(i-2) \quad \dots \quad d_k(i-M)],$$

then we can express the measurement equation (22.1) at each node  $k$  in the equivalent form:

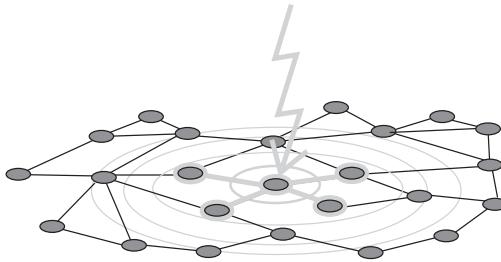
$$d_k(i) = u_{k,i} w^o + v_k(i). \quad (22.2)$$

The objective is to estimate the AR model coefficients  $\{\beta_m\}$  or  $w^o$  from measurements  $\{d_k(i)\}$  at all nodes. In other words, assuming that the  $\{d_k(i)\}$  are realizations of a random variable  $\mathbf{d}_k$  and the  $\{u_{k,i}\}$  are realizations of a random vector  $\mathbf{u}_k$ , the objective is to find the vector  $w$  that minimizes the mean-square error:

$$\frac{1}{N} \sum_{k=1}^N E|\mathbf{d}_k - \mathbf{u}_k w|^2.$$

One could employ individual adaptive filters at the nodes, with each node  $k$  estimating  $w^o$  independently of the other nodes by relying solely on its local data  $\{d_k(i), u_{k,i}, i \geq 0\}$ . In this case, each node will end up with a local estimate for  $w^o$  and the quality of this estimate will be dictated by the quality of the data at node  $k$  [such as the local signal-to-noise ratio (SNR) and noise conditions].

However, in situations where a multitude of nodes has access to data, and assuming some form of collaboration is allowed among the nodes, it is more useful to seek solutions that can take advantage of node cooperation. In addition, since the statistical profile of the data may vary with time and space, it is useful to explore cooperative



**Figure 22.2** Schematic representation of adaptive network consisting of an interconnected system of adaptive nodes interacting with each other and with information flowing through the network in real time.

strategies that are inherently adaptive. For example, the noise and SNR conditions at the nodes may vary in time and space, and the model parameters  $\{\beta_m\}$  themselves may vary with time as well. Under such conditions, it is helpful to endow the network of nodes with learning abilities so that it can function as an adaptive entity in its own right. By doing so, one would end up with an adaptive network where all nodes respond to data in real time through local and cooperative processing, as well adapt to variations in the statistical properties of the data—see Figure 22.2. It is expected that such cooperative adaptive schemes will result in improved performance over the decoupled individual filters in a noncooperative implementation, and cooperation should help equalize the effect of varying SNR conditions across the network.

This chapter illustrates these ideas with several algorithms. We start by describing incremental adaptive networks in Section 22.3, followed by diffusion networks in Section 22.4. In the process, we motivate and introduce the incremental LMS algorithm and two versions of the diffusion LMS algorithm: combine-then-adapt (CTA) and adapt-then-combine (ATC) diffusion LMS. We also comment on the performance of the algorithms via analysis and computer simulations.

### 22.3 INCREMENTAL ADAPTIVE SOLUTIONS

Consider a network with  $N$  nodes and assume initially that at least one cyclic path can be established across the network. The cyclic path should enable information to be moved from one node to a neighboring node around the network and back to the initial node (see Fig. 22.3). Obviously, some topologies may permit several possibilities for selecting such cyclic trajectories.

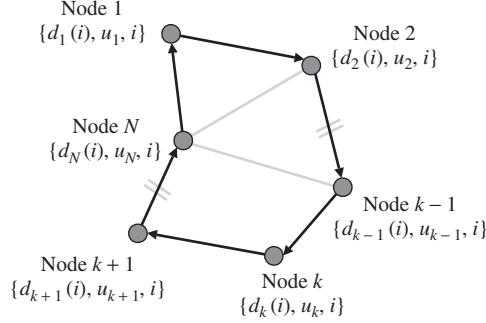
Assume further that each node  $k$  has access to time realizations  $\{d_k(i), u_{k,i}\}$  of zero-mean data  $\{\mathbf{d}_k, \mathbf{u}_k\}$ ,  $k = 1, \dots, N$ , where each  $\mathbf{d}_k$  is a scalar and each  $\mathbf{u}_k$  is a  $1 \times M$  (row) regression vector. We denote the  $M \times M$  covariance matrices of the regression data by

$$R_{u,k} \triangleq E\mathbf{u}_k^*\mathbf{u}_k \quad (\text{at node } k), \quad (22.3)$$

and the  $M \times 1$  cross-covariance vectors by

$$R_{du,k} \triangleq E\mathbf{d}_k\mathbf{u}_k^* \quad (\text{at node } k) \quad (22.4)$$

where  $E$  is the expectation operator. Observe that  $\{R_{u,k}, R_{du,k}\}$  depend on  $k$  and, therefore, for generality, we are allowing the statistical profile of the data to vary spatially



**Figure 22.3** Incremental network with  $N$  active nodes accessing space–time data.

across the nodes. The special case of uniform statistical profile would correspond to assuming  $R_{u,k} = R_u$  and  $R_{du,k} = R_{du}$  for all  $k$ .

Our objective is to develop a mechanism that would allow the nodes to cooperate with each other in order to estimate some unknown  $M \times 1$  vector  $w^o$ . We focus here on the mean-square error criterion and assume that the network seeks a vector  $w^o$  that solves the following estimation problem:

$$w^o = \arg \min_w \left( \frac{1}{N} \sum_{k=1}^N E |\mathbf{d}_k - \mathbf{u}_k w|^2 \right). \quad (22.5)$$

In other words, the optimal solution,  $w^o$ , should be such that it minimizes the average mean-square error (MSE) across the network. We shall refer to the optimal minimum cost as the resulting MSE network performance, namely,

$$\text{MSE}_{\text{network}} \triangleq \frac{1}{N} \sum_{k=1}^N E |\mathbf{d}_k - \mathbf{u}_k w^o|^2. \quad (22.6)$$

Likewise, we shall denote the MSE performance at an individual node  $k$  by

$$\text{MSE}_k \triangleq E |\mathbf{d}_k - \mathbf{u}_k w^o|^2. \quad (22.7)$$

Note that since  $N$  denotes the size of the network and is independent of the unknown  $w$ , then the optimization problem (22.5) is also equivalent to

$$w^o = \arg \min_w \sum_{k=1}^N E |\mathbf{d}_k - \mathbf{u}_k w|^2, \quad (22.8)$$

where the factor  $1/N$  has been removed. We shall denote the cost function in (22.8) by

$$J(w) \triangleq \sum_{k=1}^N E |\mathbf{d}_k - \mathbf{u}_k w|^2. \quad (22.9)$$

It is worth noting that  $J(w)$  decouples into a sum of individual cost functions, namely,

$$J(w) = \sum_{k=1}^N J_k(w), \quad (22.10)$$

where each individual  $J_k(w)$  is given by

$$J_k(w) \triangleq E |\mathbf{d}_k - \mathbf{u}_k w|^2. \quad (22.11)$$

In optimization problems involving such decoupled cost functions, incremental methods have been used to seek the solution in a distributed manner [19–22], as we now explain.

### 22.3.1 Steepest Descent Solution

To begin with, the traditional iterative steepest descent solution for determining  $w^o$  in (22.8) takes the form

$$w_i = w_{i-1} - \mu [\nabla_w J(w_{i-1})]^* \quad (22.12)$$

where  $\mu > 0$  is a step-size parameter and  $w_i$  is an estimate for  $w^o$  at iteration  $i$ . Moreover,  $\nabla_w J$  denotes the complex gradient of  $J(w)$  with respect to  $w$ , which is given by

$$\nabla_w J(w) = \sum_{k=1}^N (R_{u,k} w - R_{du,k})^*.$$

Substituting into (22.12) leads to

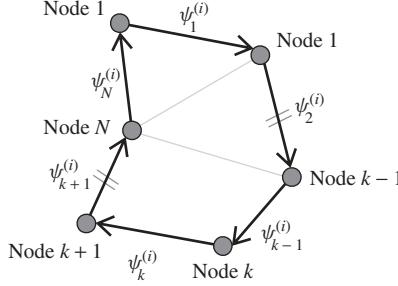
$$w_i = w_{i-1} + \mu \sum_{k=1}^N (R_{du,k} - R_{u,k} w_{i-1}). \quad (22.13)$$

Thus observe that each iteration step in (22.13) involves evaluating a sum of  $N$  terms, namely,

$$\sum_{k=1}^N (R_{du,k} - R_{u,k} w_{i-1})$$

and adding the result to  $w_{i-1}$  in order to arrive at  $w_i$ . This same result can be achieved by splitting the update into  $N$  separate steps whereby each step adds one term,  $R_{du,k} - R_{u,k} w_{i-1}$ , at a time and gives an intermediate value, say as follows:

$$\begin{aligned} \psi_0^{(i)} &\leftarrow w_{i-1}, \\ \psi_1^{(i)} &= \psi_0^{(i)} + \mu(R_{du,1} - R_{u,1} w_{i-1}), \\ \psi_2^{(i)} &= \psi_1^{(i)} + \mu(R_{du,2} - R_{u,2} w_{i-1}), \\ \psi_3^{(i)} &= \psi_2^{(i)} + \mu(R_{du,3} - R_{u,3} w_{i-1}), \\ &\vdots \\ \psi_N^{(i)} &= \psi_{N-1}^{(i)} + \mu(R_{du,N} - R_{u,N} w_{i-1}), \\ w_i &\leftarrow \psi_N^{(i)}. \end{aligned} \quad (22.14)$$



**Figure 22.4** Cycle covering nodes 1 through  $N$ .

Observe that we are denoting the intermediate value at node  $k$  by  $\psi_k^{(i)}$ , and we are using  $\psi_0^{(i)}$  to denote the initial condition at a virtual node 0. The procedure (22.14) defines a *cycle* visiting every node only once. At every iteration (or time  $i$ ), the information cycles through all  $N$  nodes. Each  $\psi_k^{(i)}$  represents a *local estimate* of  $w^o$  at node  $k$  and time  $i$ , and the process assumes that each node  $k$  has access to  $\psi_{k-1}^{(i)}$ , which is the local estimate of  $w^o$  at node  $k - 1$  – see Figure 22.4.

The procedure (22.14) can be described more compactly as follows:

$$\begin{aligned}\psi_0^{(i)} &\leftarrow w_{i-1}, \\ \psi_k^{(i)} &= \psi_{k-1}^{(i)} + \mu [R_{du,k} - R_{u,k} w_{i-1}], \quad k = 1, \dots, N, \\ w_i &\leftarrow \psi_N^{(i)}.\end{aligned}\tag{22.15}$$

Note, in particular, that the iteration for  $\psi_k^{(i)}$  is over the spatial index  $k$ . Note further that the update for  $\psi_k^{(i)}$  requires knowledge of  $w_{i-1}$ , which enters into the computation of the update direction in (22.15), namely,

$$R_{du,k} - R_{u,k} w_{i-1}.$$

The implication of this fact is that all  $N$  nodes will need to have access to  $w_{i-1}$ , which requires communicating the  $w_{i-1}$  to all nodes at each time  $i$ .

### 22.3.2 Incremental Solution

A distributed solution can be motivated by resorting to an approximation whereby the estimate  $w_{i-1}$  at each node in (22.15) is replaced by its local estimate, thus leading to what is known as an incremental solution. Specifically, if each node  $k$  relies solely on the local estimate  $\psi_{k-1}^{(i)}$  received from node  $k - 1$ , as opposed to requiring also  $w_{i-1}$ , then an incremental version of algorithm (22.15) would result in the following form:

$$\begin{aligned}\psi_0^{(i)} &\leftarrow w_{i-1}, \\ \psi_k^{(i)} &= \psi_{k-1}^{(i)} + \mu [R_{du,k} - R_{u,k} \psi_{k-1}^{(i)}], \quad k = 1, \dots, N, \\ w_i &\leftarrow \psi_N^{(i)},\end{aligned}\tag{22.16}$$

where  $w_{i-1}$  in (22.15) has been replaced by  $\psi_{k-1}^{(i)}$  in the update for  $\psi_k^{(i)}$ . Such incremental techniques have been studied extensively in the literature, and especially in the optimization literature and in works on distributed computational algorithms (e.g., [6, 19, 20, 22, 23]).

It is instructive to compare the performance of the steepest descent algorithm (22.13) or (22.15) and its incremental version (22.16) [9]. Thus, recall that the desired vector  $w^o$  in (22.8) is the solution to the normal equations [24, 25]:

$$\left( \sum_{k=1}^N R_{u,k} \right) w^o = \sum_{k=1}^N R_{du,k}. \quad (22.17)$$

We assume that the coefficient matrix is positive-definite,

$$\sum_{k=1}^N R_{u,k} > 0,$$

so that a unique solution  $w^o$  exists. Let

$$\tilde{w}_i = w^o - w_i$$

denote the weight error vector at iteration  $i$ . Subtracting  $w^o$  from both sides of the steepest descent recursion (22.13) leads to

$$\tilde{w}_i = \left[ I - \mu \sum_{k=1}^N R_{u,k} \right] \tilde{w}_{i-1}. \quad (22.18)$$

This recursion describes the dynamics of the weight error vector; it is seen that the evolution of the weight error vector is governed by the modes of the coefficient matrix:

$$F_{\text{sd}} \stackrel{\Delta}{=} \left[ I - \mu \sum_{k=1}^N R_{u,k} \right] \quad (\text{steepest descent}).$$

Let us now examine the evolution of  $\tilde{w}_i$  when evaluated by means of the incremental implementation (22.16). Subtracting  $w^o$  from both sides of (22.16) gives, for  $k = N$ ,

$$\tilde{\psi}_N^{(i)} = \tilde{\psi}_{N-1}^{(i)} - \mu \left[ R_{du,N} - R_{u,N} \psi_{N-1}^{(i)} \right], \quad (22.19)$$

where

$$\tilde{\psi}_k^{(i)} = w^o - \psi_k^{(i)}.$$

Replacing  $\psi_{N-1}^{(i)}$  in (22.19) by its update in terms of  $\psi_{N-2}^{(i)}$  [as given by (22.16)], and continuing in this manner, some algebra will show that the evolution for the weight error vector is now described by a recursion of the form:

$$\tilde{w}_i = \left[ \prod_{k=1}^N (I - \mu R_{u,k}) \right] \tilde{w}_{i-1} + O(\mu^2), \quad (22.20)$$

where  $O(\mu^2)$  denotes terms that are independent of  $\tilde{w}_{i-1}$  and of the order of  $\mu^2$  or higher powers in  $\mu$ . In the special case when the statistical profile is uniform across all nodes, that is,  $R_{u,k} = R_u$  and  $R_{du,k} = R_{du}$ , it can be verified that the driving term denoted by  $O(\mu^2)$  in (22.20) becomes zero.

Therefore, the evolution of the weight error vector  $\tilde{w}_i$  that is generated by the incremental solution (22.16) is governed by the modes of the coefficient matrix:

$$F_{\text{inc}} \triangleq \left[ \prod_{k=1}^N (I - \mu R_{u,k}) \right] \quad (\text{incremental}).$$

In order to illustrate the difference in the dynamics of both implementations, consider the special case of uniform statistical profile across all nodes. Then

$$F_{\text{sd}} = (I - \mu N R_u), \quad F_{\text{inc}} = (I - \mu R_u)^N$$

from which we find that the  $M$  modes of convergence of the algorithms are given by

$$\begin{aligned} \text{modes}^{\text{sd}} &= \{1 - \mu N \lambda_m\}, \\ \text{modes}^{\text{inc}} &= \{(1 - \mu \lambda_m)^N\}, \quad m = 1, 2, \dots, M \end{aligned}$$

in terms of the eigenvalues  $\{\lambda_m\}$  of  $R_u$ . In this case, a necessary condition for convergence in the steepest descent case (22.18) is

$$\mu < \frac{2}{N \lambda_{\max}} \quad (\text{steepest descent}),$$

whereas a necessary condition for convergence in the incremental case (23.20) is

$$\mu < \frac{2}{\lambda_{\max}} \quad (\text{incremental})$$

where  $\lambda_{\max}$  is the maximum eigenvalue of  $R_u$ . These conditions indicate that the incremental solution (22.16) converges over a wider range of the step size.

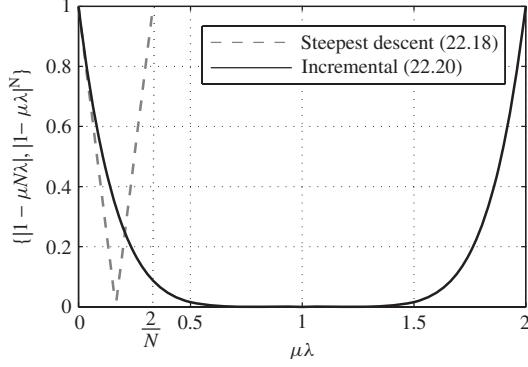
Figure 22.5 shows the magnitudes of the modes of convergence for the case  $R_u = \lambda I$  and  $N = 6$  nodes, as a function of  $\mu\lambda$ , both for the steepest descent and incremental algorithms (22.18) and (22.20), respectively. Note that for very small  $\mu\lambda$ ,  $(1 - \mu\lambda)^N \approx 1 - N\mu\lambda$ , and both algorithms have similar performance. As we increase  $\mu\lambda$ , the steepest descent algorithm has faster convergence, though it quickly becomes unstable when  $\mu\lambda = 2/N$ . For larger step sizes, the incremental algorithm has a faster convergence rate than the steepest descent solution. Note further that the stability range for the incremental algorithm is wider, leading to a more robust implementation.

Still both algorithms tend to exhibit the same behavior for diminishing step sizes. This can be seen from the weight error recursion (22.20) for the incremental solution, where the coefficient matrix can be expressed as

$$\left[ \prod_{k=1}^N (I - \mu R_{u,k}) \right] = \left[ I - \mu \sum_{k=1}^N R_{u,k} \right] + O(\mu^2)$$

so that for vanishingly small step sizes,

$$F_{\text{inc}} = F_{\text{sd}} + O(\mu^2)$$



**Figure 22.5** Modes of convergence for algorithms (22.15) and (22.16) for  $N = 6$ .

and the weight error vectors  $\{\tilde{w}_i\}$  from both algorithms (22.15) and (22.16) evolve along similar dynamics. This same conclusion follows from examining the update equation for the weight estimates directly [9]. Indeed, note from (22.11) that

$$[\nabla J_k(w)]^* = -R_{du,k} + R_{u,k}w. \quad (22.21)$$

Inspecting (22.21) we note that the following property holds for a scalar  $\mu$  and any two column vectors  $x$  and  $y$ :

$$[\nabla J_k(x - \mu y)]^* = [\nabla J_k(x)]^* - \mu [\nabla J_k(y)]^* - \mu R_{du,k}, \quad (22.22)$$

where  $\nabla J_k$  is computed relative to  $w$ .

Now, if we iterate the incremental solution (22.16) starting with  $\psi_0^{(i)} = w_{i-1}$ , we get

$$\psi_{k-1}^{(i)} = w_{i-1} - \mu \sum_{\ell=1}^{k-1} [\nabla J_\ell(\psi_{\ell-1}^{(i)})]^*. \quad (22.23)$$

Substituting (22.23) into (22.16) gives

$$\psi_k^{(i)} = \psi_{k-1}^{(i)} - \mu \left[ \nabla J_k \left( w_{i-1} - \mu \sum_{\ell=1}^{k-1} [\nabla J_\ell(\psi_{\ell-1}^{(i)})]^* \right) \right]^*.$$

Using relation (22.22) with the choices

$$x = w_{i-1}, \quad y = \sum_{\ell=1}^{k-1} [\nabla J_\ell(\psi_{\ell-1}^{(i)})]^*$$

leads to

$$\psi_k^{(i)} = \underbrace{\psi_{k-1}^{(i)} - \mu ([\nabla J_k(w_{i-1})]^*)^*}_{\text{steepest descent as in (22.15)}} + \underbrace{\mu^2 \left( \left[ \nabla J_k \left( \sum_{\ell=1}^{k-1} [\nabla J_\ell(\psi_{\ell-1}^{(i)})]^* \right) \right]^* + R_{du,k} \right)}_{\text{extra term due to incremental procedure}}. \quad (22.24)$$

Therefore, the incremental algorithm can be written as a sum of the steepest descent update plus extra terms. As  $\mu \rightarrow 0$ , the  $\mu$  term dominates the  $\mu^2$  term and the incremental algorithm (22.16) and the steepest descent algorithm (22.15) tend to exhibit the same behavior.

### 22.3.3 Adaptation: Incremental LMS

Now note that the incremental algorithm (22.16) requires knowledge of the second-order moments  $\{R_{u,k}, R_{du,k}\}$ . An adaptive implementation of (22.16) can be obtained by replacing these second-order moments by local instantaneous approximations, say of the LMS type, as follows:

$$R_{du,k} \approx d_k(i)u_{k,i}^*, \quad R_{u,k} \approx u_{k,i}^*u_{k,i}. \quad (22.25)$$

Obviously, more involved approximations are possible and they would lead to alternative adaptive implementations. Using the approximations (22.25) leads to the *incremental LMS* algorithm derived in [9, 13], where we additionally allow for the step sizes to vary across the nodes.

*Incremental LMS* Start with  $w_{-1} = 0$ . For each time  $i \geq 0$ , repeat:

$$\begin{aligned} \psi_0^{(i)} &= w_{i-1}, \\ \psi_k^{(i)} &= \psi_{k-1}^{(i)} + \mu_k u_{k,i}^*(d_k(i) - u_{k,i}\psi_{k-1}^{(i)}), \quad k = 1, \dots, N, \\ w_i &= \psi_N^{(i)}. \end{aligned} \quad (22.26)$$

One question is how well the adaptive algorithm (22.26) performs. A detailed mean-square and stability analysis of the algorithm is performed in [9, 13]. The analysis relies on the following assumptions on the data:  $\{\mathbf{d}_k(i), \mathbf{u}_{k,i}\}$ :

1. The unknown vector  $w^o$  relates  $\{\mathbf{d}_k(i), \mathbf{u}_{k,i}\}$  as

$$\mathbf{d}_k(i) = \mathbf{u}_{k,i}w^o + \mathbf{v}_k(i) \quad (22.27)$$

where  $\mathbf{v}_k(i)$  is some white noise sequence with variance  $\sigma_{v,k}^2$  and independent of  $\{\mathbf{d}_\ell(j), \mathbf{u}_{\ell,j}\}$  for all  $\ell, j$ .

2.  $\mathbf{u}_{k,i}$  is independent of  $\mathbf{u}_{\ell,i}$  for  $k \neq \ell$  (spatial independence).
3. For every  $k$ , the sequence  $\{\mathbf{u}_{k,i}\}$  is independent over time (time independence).
4. The regressors  $\{\mathbf{u}_{k,i}\}$  arise from a source with circular Gaussian distribution with covariance matrix  $R_{u,k}$ .

It is worth noting that for linear data models of the form (22.27), the solution  $w^o$  of the mean-square-error criterion in (22.5) coincides with the desired unknown vector in (22.27) [24, 25].

The following results are simplifications of the general expressions derived in [9] assuming sufficiently small step sizes. Define the error signals:

$$\tilde{\psi}_k^{(i)} \triangleq w^o - \psi_k^{(i)}, \quad (22.28)$$

$$\mathbf{e}_{a,k}(i) \triangleq \mathbf{u}_{k,i}\tilde{\psi}_{k-1}^{(i)} \quad (22.29)$$

where (22.28) denotes the weight error vector and (22.29) denotes the a priori local error, both at node  $k$  and time  $i$ . Observe that we are now denoting  $\tilde{\psi}_k^{(i)}$  and  $e_{a,k}(i)$  by boldface italic letters to highlight the fact that they are random quantities whose variances we are interested in evaluating.

For each node  $k$ , the mean-square deviation (MSD) and the excess mean-square error (EMSE) are defined as the steady-state values of the variances of these error quantities, namely,

$$\eta_k \triangleq E \left\| \tilde{\psi}_k^{(\infty)} \right\|^2 \quad (\text{MSD}), \quad (22.30)$$

$$\zeta_k \triangleq E |e_{a,k}(\infty)|^2 \quad (\text{EMSE}). \quad (22.31)$$

In the case of small step sizes, simplified expressions for the MSD and EMSE can be described as follows. For each node  $k$ , introduce the eigendecomposition

$$R_{u,k} = U_k \Lambda_k U_k^*,$$

where  $U_k$  is unitary and  $\Lambda_k$  is a diagonal matrix with the eigenvalues of  $R_{u,k}$ :

$$\Lambda_k = \text{diag}\{\lambda_{k,1}, \lambda_{k,2}, \dots, \lambda_{k,M}\} \quad (\text{node } k).$$

Define further the quantities:

$$D \triangleq 2 \sum_{k=1}^N \mu_k \Lambda_k \quad (\text{diagonal matrix}),$$

$$b_k \triangleq \text{diag}\{\Lambda_k\} \quad (\text{column vector}),$$

$$a \triangleq \sum_{k=1}^N \mu_k^2 \sigma_{v,k}^2 b_k \quad (\text{column vector}),$$

$$q \triangleq \text{col}\{1, 1, \dots, 1\}$$

where  $\sigma_{v,k}^2$  denotes the noise variance at node  $k$ . Then, according to the results from [8, 9]:

$$\eta_k \approx a^T D^{-1} q \quad (\text{MSD}), \quad (22.32)$$

$$\zeta_k \approx a^T D^{-1} b_k \quad (\text{EMSE}), \quad (22.33)$$

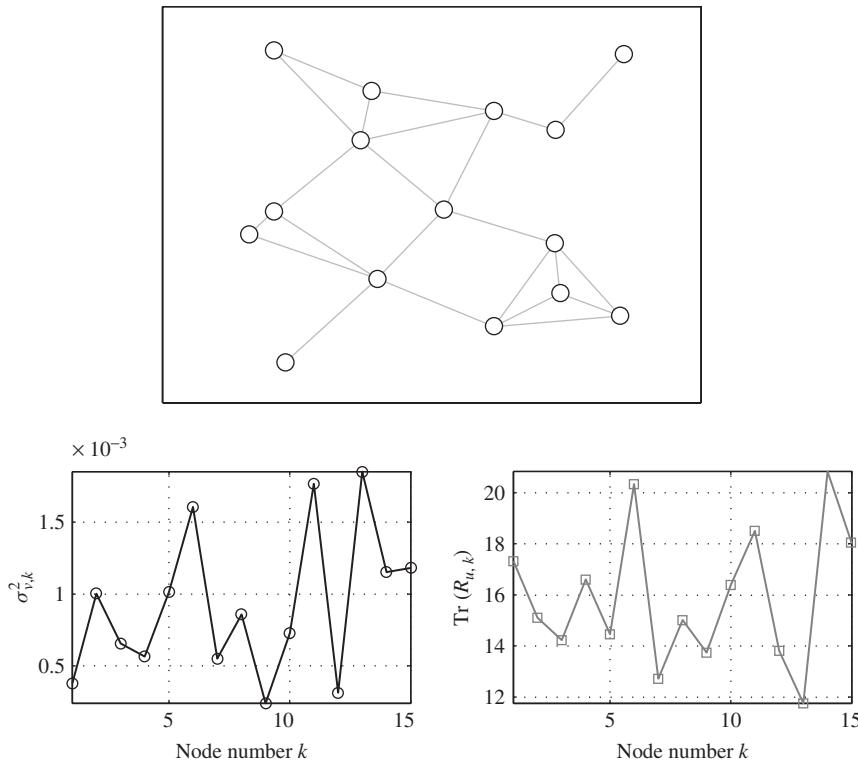
or, more explicitly,

$$\eta_k \approx \frac{1}{2} \sum_{j=1}^M \left( \frac{\sum_{\ell=1}^N \mu_{\ell}^2 \sigma_{v,\ell}^2 \lambda_{\ell,j}}{\sum_{\ell=1}^N \mu_{\ell} \lambda_{\ell,j}} \right), \quad (22.34)$$

$$\zeta_k \approx \frac{1}{2} \sum_{j=1}^M \left( \lambda_{k,j} \cdot \frac{\sum_{\ell=1}^N \mu_{\ell}^2 \sigma_{v,\ell}^2 \lambda_{\ell,j}}{\sum_{\ell=1}^N \mu_{\ell} \lambda_{\ell,j}} \right). \quad (22.35)$$

Moreover, the mean-square performance at each node is given by

$$\text{MSE}_k = \zeta_k + \sigma_{v,k}^2. \quad (22.36)$$



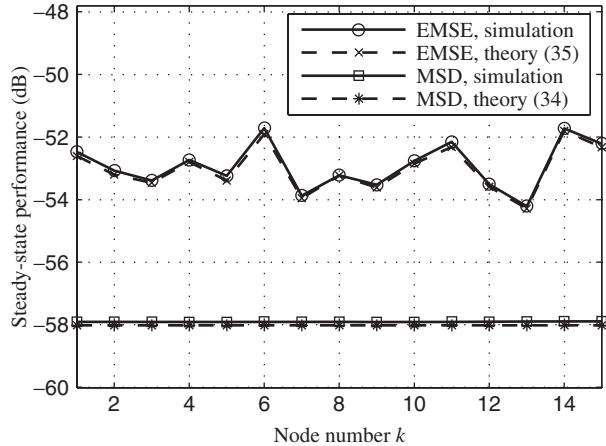
**Figure 22.6** Network topology (top), noise variances  $\sigma_{v,k}^2$  (bottom, left), and trace of regressor covariances  $\text{Tr}(R_{u,k})$  (bottom, right) for  $N = 15$  nodes.

The fact that the expression (22.32) for the MSD is independent of  $k$  reveals an interesting behavior. Namely, there is an equalization effect on the MSD throughout the network.

In order to illustrate the adaptive network performance, we present a simulation example in Figures 22.6 and 22.7. Figure 22.6 depicts the network topology with  $N = 15$  nodes, together with the network statistical profile. The regressors are zero-mean Gaussian, independent in time and space, with covariance matrices  $R_{u,k}$ . The background noise power is denoted by  $\sigma_{v,k}^2$ . Figure 22.7 shows the steady-state performance for the incremental LMS algorithm (22.26), using a uniform  $\mu = 0.01$ . The results were averaged over 200 experiments, and the steady-state values were calculated by averaging the last 50 samples after convergence. The figure shows the simulated steady-state MSD and EMSE for every node in the network and compares them with the theoretical results from (22.34) and (22.35).

## 22.4 DIFFUSION ADAPTIVE SOLUTIONS

The adaptive incremental solution (22.26) requires a cyclic trajectory across the entire network, which can limit its application and make the procedure less robust to node and link failures. In particular, observe that the updates progress sequentially from one



**Figure 22.7** Steady-state performance of the incremental LMS algorithm (22.6); theory and simulation.

node to another so that the generation of  $\psi_k^{(i)}$  at node  $k$  can only happen after  $\psi_{k-1}^{(i)}$  has been generated at node  $k-1$ .

However, when more communication resources are available, we should be able to take advantage of the network connectivity and devise more sophisticated cooperation rules among the nodes. For instance, node  $k$  does not need to rely solely on information from node  $k-1$ ; it should also be able to rely on information from other nodes in its neighborhood. In addition, it should be possible for all nodes in the network to undergo updates simultaneously whenever possible without being limited by sequential processing.

Thus observe from (22.26) that the update for each node  $k$  relies on receiving the local estimate  $\psi_{k-1}^{(i)}$  from its neighbor  $k-1$ . What if the network topology allows cooperation between node  $k$  and several other neighboring nodes? In this case, one could consider providing node  $k$  with a local estimate that is not only based on what node  $k-1$  has to offer but also on the information that the other neighboring nodes can offer. For example, one could consider replacing the local estimate  $\psi_{k-1}^{(i)}$  in the incremental iteration (22.26) by some linear combination of the local estimates at the neighbors of node  $k$ , say, replace  $\psi_{k-1}^{(i)}$  by

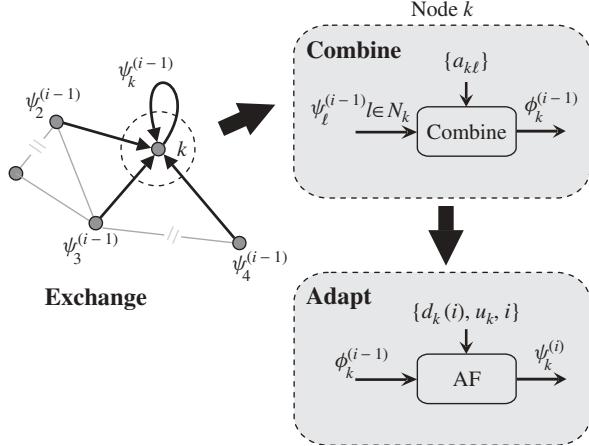
$$\phi_k^{(i-1)} \triangleq \sum_{\ell \in \mathcal{N}_k} a_{k\ell} \psi_\ell^{(i-1)}. \quad (22.37)$$

The coefficients  $\{a_{k\ell}\}$  are scaling factors that add up to one:

$$\sum_{\ell \in \mathcal{N}_k} a_{k\ell} = 1 \quad (\text{for each node } k), \quad (22.38)$$

and the notation  $\mathcal{N}_k$  denotes the set of all nodes lying in the neighborhood of node  $k$  (including  $k$  itself), that is, it is the set of all nodes  $\ell$  that can communicate with node  $k$ :

$$\mathcal{N}_k = \{\text{set of nodes connected to } k \text{ including itself}\}. \quad (22.39)$$



**Figure 22.8** Network with CTA diffusion strategy.

More generally, the neighborhood  $\mathcal{N}_k$  could also vary with time, say as  $\mathcal{N}_{k,i}$ , but we are going to continue to work with  $\mathcal{N}_k$  in this chapter for simplicity of exposition. We comment on choices for the combination coefficients  $\{a_{k\ell}\}$  later in Section 22.4.3.

#### 22.4.1 Adaptation: Node-Based Diffusion

Using (22.37), one can then consider replacing the incremental update (22.26) by the following recursion proposed in [10, 14]—see Figure 22.8:

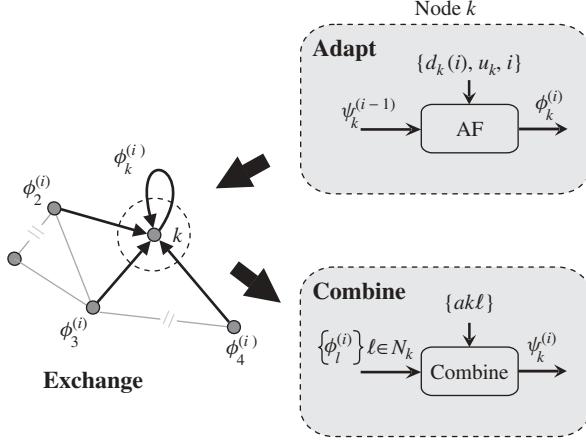
**CTA Diffusion LMS** Start with  $\{\psi_\ell^{(-1)} = 0\}$  for all  $\ell$ . For each time  $i \geq 0$  and for each node  $k$ , repeat:

$$\begin{aligned} \phi_k^{(i-1)} &= \sum_{\ell \in \mathcal{N}_k} a_{k\ell} \psi_\ell^{(i-1)} \quad (\text{CTA version}), \\ \psi_k^{(i)} &= \phi_k^{(i-1)} + \mu_k u_{k,i}^* \left[ d_k(i) - u_{k,i} \phi_k^{(i-1)} \right]. \end{aligned} \quad (22.40)$$

Note that the cyclic update through the nodes has been removed. Now, instead, at each iteration  $i$ , every node  $k$  performs a two-step procedure: an initial aggregation step to evaluate the aggregate (intermediate) estimate  $\phi_k^{(i-1)}$  and a subsequent adaptation step to update the local node estimate to  $\psi_k^{(i)}$ . The aggregation step combines estimates  $\{\psi_\ell^{(i-1)}\}$  from the *previous* time step  $i-1$ . In this way, all nodes across the network can perform their diffusion updates at the same time.

We refer to the above algorithm as *diffusion LMS* or, more specifically, as the combine-then-adapt (CTA) diffusion LMS version. The term diffusion is used to highlight the fact that information is being shared (or diffused) among the nodes in the neighborhood and, more generally, among the nodes in the entire network. This is because the aggregation step incorporates information from other neighborhoods into  $\phi_k^{(i-1)}$ .

A useful alternative to the diffusion algorithm (22.40) is to perform the adaptation step first followed by the aggregation step, say (see Fig. 22.9).



**Figure 22.9** Network with ATC diffusion strategy.

**ATC Diffusion LMS** Start with  $\{\psi_\ell^{(-1)} = 0\}$  for all  $\ell$ . For each time  $i \geq 0$  and for each node  $k$ , repeat:

$$\begin{aligned}\phi_k^{(i)} &= \psi_k^{(i-1)} + \mu_k u_{k,i}^* \left[ d_k(i) - u_{k,i} \psi_k^{(i-1)} \right], \\ \psi_k^{(i)} &= \sum_{\ell \in \mathcal{N}_k} a_{k\ell} \phi_\ell^{(i)} \quad (\text{ATC version}).\end{aligned}\tag{22.41}$$

We refer to (22.41) as the ATC diffusion LMS algorithm. Analysis and simulations show that ATC outperforms CTA. Intuitively, this is because the ATC version performs the adaptation step first, which incorporates the current data into the local weight estimates,  $\phi_\ell^{(i)}$ , before combining them. We should note that for both versions, the local weight vector estimate at node  $k$  and time  $i$  is taken as  $\psi_k^{(i)}$ .

One could also consider other diffusion schemes whereby the aggregation step involves more general functions of the local estimates, say as [10]:

$$\begin{aligned}\phi_k^{(i-1)} &= f_k \left[ \psi_\ell^{(i-1)}; \ell \in \mathcal{N}_k \right], \\ \psi_k^{(i)} &= \phi_k^{(i-1)} + \mu_k u_{k,i}^* \left[ d_k(i) - u_{k,i} \phi_k^{(i-1)} \right]\end{aligned}\tag{22.42}$$

for some local combiner  $f_k(\cdot)$ . The combiners  $f_k(\cdot)$  can be nonlinear or even time variant, to reflect, for instance, changing topologies or to respond to non-stationary environments. For illustration purposes we continue to focus on the linear combination structures defined by (22.40) and (22.41).

### 22.4.2 Mean-Square-Error Optimization

The CTA and ATC diffusion LMS algorithms so described can be motivated formally in the same manner as the incremental LMS algorithm by starting from a mean-square cost function as follows. Consider node  $k$  and assume each node  $\ell$  in its neighborhood

has some initial estimate for the weight vector  $w$ , say  $\{\psi_\ell, \ell \in \mathcal{N}_k\}$ . We then formulate at node  $k$  the problem of estimating the weight vector  $w$  that solves

$$\min_w \left( \delta \sum_{\ell \in \mathcal{N}_k} c_{k\ell} \|w - \psi_\ell\|^2 + E |\mathbf{d}_k - \mathbf{u}_k w|^2 \right), \quad (22.43)$$

where  $\delta > 0$  is a regularization parameter and the  $\{c_{k\ell}\}$  are some weighting coefficients that add up to one:

$$\sum_{\ell \in \mathcal{N}_k} c_{k\ell} = 1.$$

The second term in (22.43) is the same function  $J_k(w)$  used earlier in (22.11); this term involves only local information and seeks that value of  $w$  that helps match  $\mathbf{d}_k$  to  $\mathbf{u}_k w$  in the mean-square-error sense. The first term in the cost function (22.43) penalizes the distance between the solution  $w$  and the prior information represented by the available local estimates  $\{\psi_\ell\}$ . This is a useful term because it incorporates global information from other neighborhoods in the network; this is because the estimates  $\{\psi_\ell\}$  are expected to have been influenced by data across the other neighborhoods.

Now note that the cost function in (22.43) decouples into the sum of two individual cost functions:

$$\delta \sum_{\ell \in \mathcal{N}_k} c_{k\ell} \|w - \psi_\ell\|^2$$

and

$$E |\mathbf{d}_k - \mathbf{u}_k w|^2.$$

Thus, as before, an incremental approach can be used to carry out the optimization at node  $k$ . Let  $\{\psi_k^{(i)}, i \geq 0\}$  denote the successive iterates at node  $k$  that result from applying a steepest descent approach to minimizing (22.43). Then the traditional steepest descent solution, with the gradient vector of the cost function evaluated at the prior iterate  $\psi_k^{(i-1)}$ , is given by

$$\psi_k^{(i)} = \psi_k^{(i-1)} + \mu \left( R_{du,k} - R_{u,k} \psi_k^{(i-1)} \right) - \mu \delta \sum_{\ell \in \mathcal{N}_k} c_{k\ell} \left( \psi_k^{(i-1)} - \psi_\ell \right).$$

We can split this update into two incremental update steps, say as

$$\phi_k^{(i)} = \psi_k^{(i-1)} + \mu \left( R_{du,k} - R_{u,k} \psi_k^{(i-1)} \right), \quad (22.44)$$

$$\psi_k^{(i)} = \phi_k^{(i)} - \mu \delta \sum_{\ell \in \mathcal{N}_k} c_{k\ell} \left( \psi_k^{(i-1)} - \psi_\ell \right), \quad (22.45)$$

with the intermediate variable denoted by  $\phi_k^{(i)}$ . And, just like we replaced  $w_{i-1}$  of (22.15) by the local estimate  $\psi_{k-1}^{(i)}$  in (22.16), we can also replace  $\psi_k^{(i-1)}$  in (22.45)

by the local estimate  $\phi_k^{(i)}$  from (22.44). This approximation leads to

$$\phi_k^{(i)} = \psi_k^{(i-1)} + \mu \left( R_{du,k} - R_{u,k} \psi_k^{(i-1)} \right), \quad (22.46)$$

$$\psi_k^{(i)} = \phi_k^{(i)} - \mu \delta \sum_{\ell \in \mathcal{N}_k} c_{k\ell} (\phi_k^{(i)} - \psi_\ell^{(i)}). \quad (22.47)$$

Recall that the  $\{\psi_\ell\}$  in (22.47) are local estimates at the nodes  $\ell$  in the neighborhood of  $k$ . One useful way to approximate these estimates is by replacing them by the values  $\{\phi_l^{(i)}\}$  that are available at time  $i$  at these nodes, so that iteration (22.47) becomes

$$\begin{aligned} \psi_k^{(i)} &= \phi_k^{(i)} - \mu \delta \sum_{\ell \in \mathcal{N}_k} c_{k\ell} (\phi_k^{(i)} - \phi_\ell^{(i)}) \\ &= (1 - \mu \delta + \mu \delta c_{kk}) \phi_k^{(i)} + \sum_{\ell \in \mathcal{N}_k - \{k\}} \mu \delta c_{k\ell} \phi_\ell^{(i)}. \end{aligned}$$

Introduce the coefficients

$$a_{kk} = (1 - \mu \delta + \mu \delta c_{kk}), \quad a_{k\ell} = \mu \delta c_{k\ell}, \quad k \neq \ell.$$

Note that the  $\{a_{k\ell}\}$  defined in this manner add up to one. Note also that  $a_{k\ell} = c_{k\ell}$  if we set  $\delta = \mu^{-1}$ . Then we obtain

$$\begin{aligned} \phi_k^{(i)} &= \psi_k^{(i-1)} + \mu \left( R_{du,k} - R_{u,k} \psi_k^{(i-1)} \right), \\ \psi_k^{(i)} &= \sum_{\ell \in \mathcal{N}_k} a_{k\ell} \phi_\ell^{(i)}. \end{aligned}$$

If we apply the instantaneous approximations (22.25), and make the step size node dependent, then we arrive at the ATC diffusion LMS algorithm (22.41).

The CTA version (22.40) can be obtained in a similar manner if we simply reverse the order by which the incremental split was done in (22.44) and (22.45).

### 22.4.3 Combination Rules

There are several ways by which the combination weights  $\{a_{k\ell}\}$  can be selected. We list here some examples that have been used in the literature in the context of graph problems. We also motivate the case where the weights  $\{a_{k\ell}\}$  can be adapted as well; in this case, the network gains another level of adaptation, and the nodes are able to give less or more weight selectively to their neighbors according to their performance and reliability.

One of the simplest choices for the  $\{a_{k\ell}\}$  is to average the neighboring estimates. Thus, let  $n_k$  denote the degree of node  $k$ , which is defined as the number of incident links at the node (including a link from the node onto itself). In other words,  $n_k$  is the size of the neighborhood of  $k$ :

$$\begin{aligned} n_k &\stackrel{\Delta}{=} \text{number of neighbors of node } k \text{ including itself} \\ &= |\mathcal{N}_k|. \end{aligned}$$

Then we may select (see, e.g., [26])

$$a_{k\ell} = \frac{1}{n_k} \quad \text{for each } \ell \in \mathcal{N}_k.$$

In this case, each node is assigned the same weight and

$$\sum_{\ell \in \mathcal{N}_k} a_{k\ell} = 1.$$

This scheme exploits network connectivity rather fully, leading to robust algorithms. If links or nodes eventually fail, the adaptive network can still react by relying on the remaining topology.

The so-called Laplacian rule is described as follows. In graph theory, the entries of the  $N \times N$  Laplacian matrix  $\mathcal{L}$  of a graph with  $N$  nodes is defined as [27]

$$\mathcal{L}_{k\ell} = \begin{cases} -1 & \text{if } k \neq \ell \text{ are linked,} \\ n_k - 1 & \text{for } k = \ell, \\ 0 & \text{otherwise.} \end{cases}$$

Note that for  $k = \ell$ , the entry of the Laplacian matrix is the number of incident links on node  $k$ . The Laplacian of a graph has several important properties. For example, it is always a nonnegative-definite matrix and the number of times that 0 occurs as an eigenvalue is equal to the number of connected components in the graph. The weights  $A = [a_{k\ell}]$  in the Laplacian rule are chosen as follows (see, e.g., [28, 29]):

$$A = I_N - \gamma \mathcal{L}$$

for some constant  $\gamma$ . A possible choice is  $\gamma = n_{\max}$  where  $n_{\max}$  denotes the maximum degree across the network. In this case we get

$$a_{k\ell} = \begin{cases} 1/n_{\max} & \text{if } k \neq \ell \text{ are linked,} \\ 1 - (n_k - 1)/n_{\max} & \text{for } k = \ell, \\ 0 & \text{otherwise.} \end{cases}$$

Another choice is the maximum-degree weights rule (e.g., [30]) which uses  $\gamma = 1/N$ , or equivalently,

$$a_{k\ell} = \begin{cases} 1/N & \text{if } k \neq \ell \text{ are linked,} \\ 1 - (n_k - 1)/N & \text{for } k = \ell, \\ 0 & \text{otherwise.} \end{cases}$$

In this case, all links are assigned weights  $1/N$  and each node complements the sum of the weights to 1.

The so-called Metropolis rule is described in [29] and motivated by earlier works on sampling methods [31, 32]. Let  $n_k$  and  $n_\ell$  denote the degrees of nodes  $k$  and  $\ell$ , respectively. Then  $a_{k\ell}$  is selected as follows:

$$a_{k\ell} = \begin{cases} 1/\max(n_k, n_\ell) & \text{if } k \neq \ell \text{ are linked,} \\ 1 - \sum_{\ell \in \mathcal{N}_k - \{k\}} a_{k\ell} & \text{for } k = \ell, \\ 0 & \text{otherwise.} \end{cases}$$

In this case, the weighting assigned to a link is dependent on the degree of the node (i.e., on the number of incident links into that node).

Another choice is the relative-degree rule from [11], which does not yield symmetric weight matrices but generally yields better performance as illustrated in the examples further ahead:

$$a_{k\ell} = \begin{cases} n_\ell / \sum_{m \in \mathcal{N}_k} n_m & \text{if } k \text{ and } \ell \text{ are linked or } k = \ell, \\ 0 & \text{otherwise.} \end{cases} \quad (22.48)$$

In this case, every neighbor is weighted according to its degree. Reference [11] also suggests an optimal design procedure for the combination matrix  $A$  that is aimed at enhancing the network mean-square-error performance.

In the above rules, the combination weights are largely dictated by the sizes of the neighborhoods (or by the node degrees). When the neighborhoods vary with time, the degrees will also vary. However, for all practical purposes, these combination schemes are not adaptive in the sense that the schemes do not learn which nodes are more or less reliable so that the weights can be adjusted accordingly.

An adaptive combination rule along these lines can be motivated by the analysis results of [12]. The rule allows the network to assign convex combination weights to the local estimates and the aggregate estimate. Moreover, the combination weights can be adjusted adaptively so that the network can respond to node conditions and assign smaller weights to nodes that are subject to higher noise levels. For example, one possibility could be as follows [10]. Consider a set of coefficients  $b_{k\ell}$  that add up to one when node  $k$  is excluded. These coefficients could be obtained from the coefficients  $a_{k\ell}$  as follows:

$$b_{k\ell} = \begin{cases} \frac{a_{k\ell}}{\sum_{\ell \in \mathcal{N}_k - \{k\}} a_{k\ell}} & \text{if } k \neq \ell \text{ are linked,} \\ 0 & \text{otherwise.} \end{cases}$$

Now, we combine the local estimates at the neighbors of node  $k$ , say as before, but excluding node  $k$  itself. This step results in an intermediate estimate:

$$\bar{\psi}_k^{(i-1)} = \sum_{\ell \in \mathcal{N}_k - \{k\}} b_{k\ell} \psi_\ell^{(i-1)}.$$

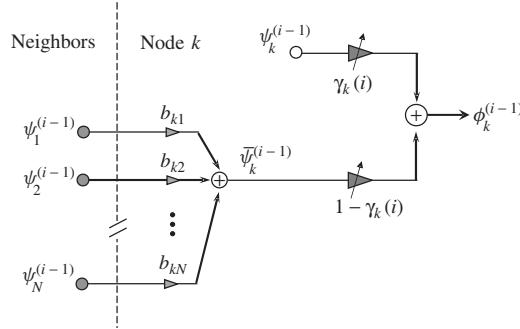
Then this aggregate estimate is combined *adaptively* with the local estimate at node  $k$  to provide the desired combination (compare with (22.40) and see Fig. 22.10):

$$\phi_k^{(i-1)} = \gamma_k(i) \psi_k^{(i-1)} + [1 - \gamma_k(i)] \bar{\psi}_k^{(i-1)},$$

where the coefficient  $\{\gamma_k(i)\}$  is adapted in order to improve performance (such as reducing the mean-square error further whenever possible) [10, 12]. The idea is that the selection of  $\gamma_k$  will give more or less weight to the local weight as opposed to the combination from the neighbors depending on which source of information is more reliable (or less noisy); we forgo the details of adapting the coefficient  $\gamma_k$ . Once this is done, we may continue to the adaptation step:

$$\psi_k^{(i)} = \phi_k^{(i-1)} + \mu u_{k,i}^* [d_k(i) - u_{k,i} \phi_k^{(i-1)}].$$

Alternatively, one could consider adapting all the coefficients  $\{a_{k\ell}\}$  in the diffusion schemes (22.40) and (22.41) directly.



**Figure 22.10** Example of network with an *adaptive* diffusion strategy.

#### 22.4.4 Simulation Examples

In order to illustrate the adaptive network performance, we present a simulation example. Figure 22.6 depicts the network topology with  $N = 15$  nodes, together with the network statistical profile. The regressors are zero-mean Gaussian, independent in time and space, with covariance matrices  $R_{u,k}$ . The background noise power is denoted by  $\sigma_{v,k}^2$ . Figure 22.11 shows the learning behavior of several algorithms in terms of the network EMSE and MSD. These are evaluated as

$$\zeta^{\text{network}}(i) = \frac{1}{N} \sum_{k=1}^N \zeta_k(i) \quad (\text{EMSE}),$$

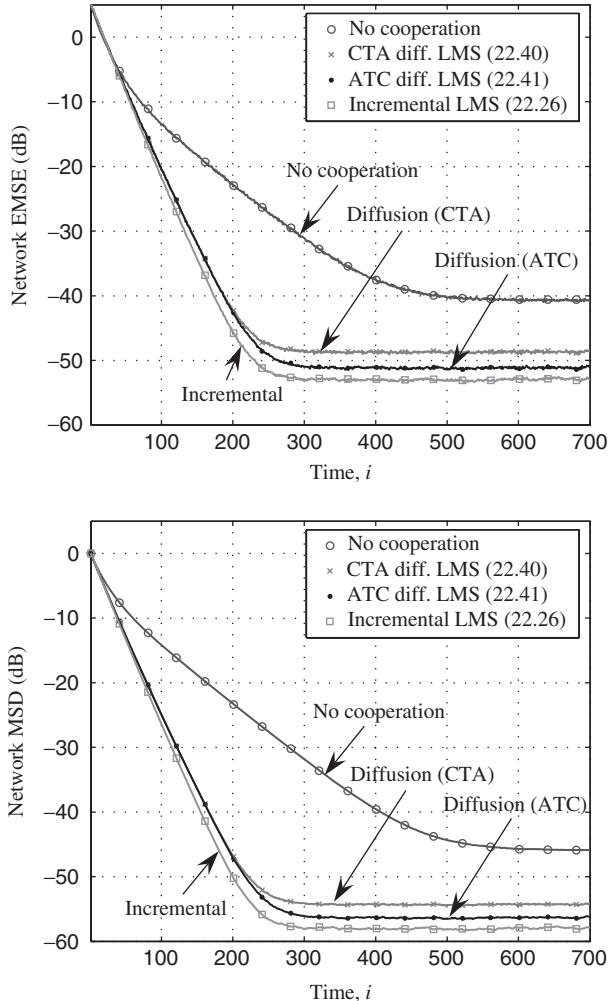
$$\eta^{\text{network}}(i) = \frac{1}{N} \sum_{k=1}^N \eta_k(i) \quad (\text{MSD}),$$

by averaging the corresponding curves across all nodes. For the diffusion and no-cooperation cases, a value of  $\mu_k = 0.01$  was used, whereas for the incremental LMS algorithm, the value was  $\mu_k = 0.01/N$ ; this is because the incremental algorithm uses  $N$  LMS-type iterations for every measurement time. The relative-degree weights (22.48) were used in the diffusion algorithms. The curves were averaged over 200 experiments, and the steady-state values were calculated by averaging the last 50 samples after convergence.

Note how the incremental and diffusion algorithms (22.26), (22.40), and (22.41) significantly outperform the noncooperative case (where each node runs an individual filter). Also note that the ATC algorithm (22.41) outperforms the CTA version (22.40). Also shown is the incremental diffusion LMS algorithm (22.26), which outperforms the diffusion solutions. This behavior by the incremental solution is expected since the incremental algorithm uses data from the entire network at every iteration. Figure 22.12 shows the steady-state network EMSE and MSD for every node in the network.

#### 22.4.5 Cooperation Enhances Stability

We illustrate in this section a useful property of the diffusion algorithms, namely, that cooperation does not only enhance performance but it also enhances stability relative to the noncooperative solution with individual filters at the nodes. Let us focus on the CTA version (22.40) of diffusion LMS.



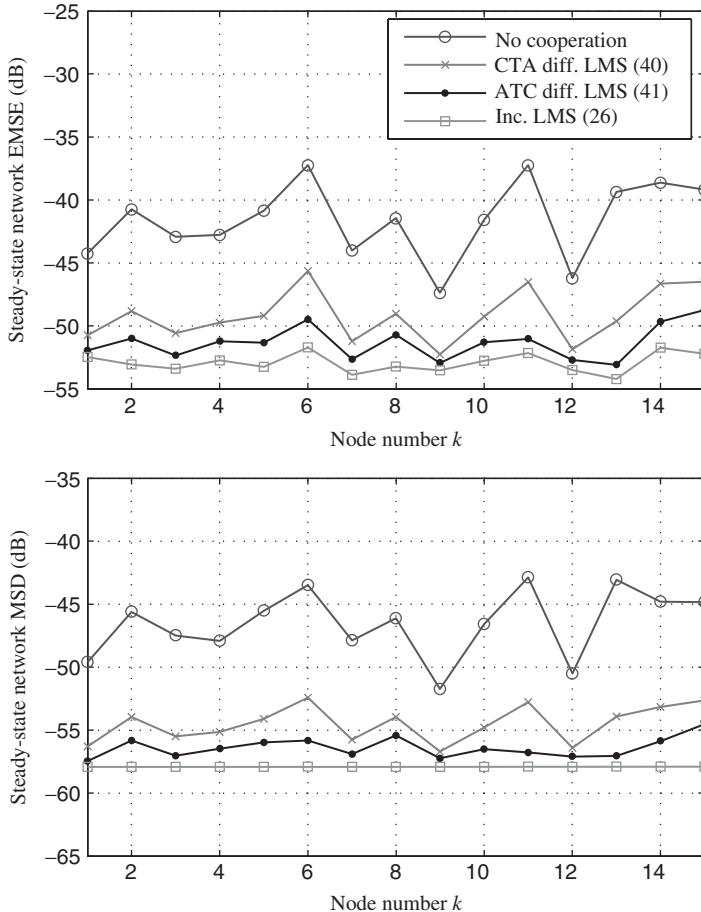
**Figure 22.11** Transient network EMSE (top) and MSD (bottom) for LMS without cooperation, CTA diffusion LMS, ATC diffusion LMS, individual LMS filters (no cooperation), and incremental LMS.

The coefficients  $a_{k\ell}$  give rise to an  $N \times N$  combination matrix  $A = [a_{k\ell}]$ , which carries information about the network topology: A nonzero entry  $a_{k\ell}$  means that nodes  $k$  and  $\ell$  are connected. Note that  $A$  is a stochastic matrix, namely, it satisfies

$$Aq = q$$

where  $q \stackrel{\Delta}{=} \text{col}\{1, \dots, 1\}$ . Let  $X \otimes Y$  denote the Kronecker product of the matrices  $X$  and  $Y$ . Note in particular that if  $X$  and  $Y$  are both  $M \times M$ , then their Kronecker product is  $M^2 \times M^2$ . Moreover,

$$I_m \otimes X = \text{diag}\underbrace{\{X, X, \dots, X\}}_{m \text{ times}}.$$



**Figure 22.12** Steady-state EMSE (top) and MSD (bottom) per node, for LMS without cooperation, CTA diffusion LMS, ATC diffusion LMS, and incremental LMS.

Introduce the global quantities

$$\tilde{\psi}^i \triangleq \text{col}\{\tilde{\psi}_1^{(i)}, \tilde{\psi}_2^{(i)}, \dots, \tilde{\psi}_N^{(i)}\},$$

$$\mathcal{M} \triangleq \text{diag}\{\mu_1 I_M, \mu_2 I_M, \dots, \mu_N I_M\},$$

$$\mathcal{A} \triangleq A \otimes I_M,$$

$$\mathcal{R}_u \triangleq \text{diag}\{R_{u,1}, R_{u,2}, \dots, R_{u,N}\}$$

where  $\{\mathcal{M}, \mathcal{A}, \mathcal{R}_u\}$  are  $NM \times NM$  matrices. Then, under the data assumptions described earlier in (22.27), some straightforward algebra will show that the mean of the extended weight error vector evolves according to the following dynamics:

$$E\tilde{\psi}^i = (I_{NM} - \mathcal{M}\mathcal{R}_u)\mathcal{A}E\tilde{\psi}^{i-1}. \quad (22.49)$$

For simplicity of notation, let

$$\mathcal{B} \triangleq I_{NM} - \mathcal{M}\mathcal{R}_u.$$

Then, expression (22.49) shows that the adaptive network will be stable in the mean if, and only if, the spectral radius of  $\mathcal{BA}$  is strictly less than one, that is,

$$|\lambda(\mathcal{BA})| < 1. \quad (22.50)$$

In the absence of cooperation (i.e., when the nodes evolve independently of each other and therefore  $\mathcal{A} = I_{NM}$ ), the mean-error vector would instead evolve according to

$$E\tilde{\psi}^i = (I_{NM} - \mathcal{M}\mathcal{R}_u)E\tilde{\psi}^{i-1}$$

with coefficient matrix  $\mathcal{B}$  alone. Thus, in the diffusion network case, convergence in the mean also depends on the network topology (as represented by  $\mathcal{A}$ ). Using matrix 2-norms we have

$$\|\mathcal{BA}\|_2 \leq \|\mathcal{B}\|_2 \cdot \|\mathcal{A}\|_2. \quad (22.51)$$

However, for any matrix  $X$  it holds that

$$|\lambda_{\max}(X)| \leq \|X\|_2 \quad (22.52)$$

with equality if  $X$  is Hermitian. Moreover, due to the block structure of  $\mathcal{R}_u$ ,  $\mathcal{B}$  is Hermitian, and recall that  $\mathcal{A} = A \otimes I_M$ . Hence, we have

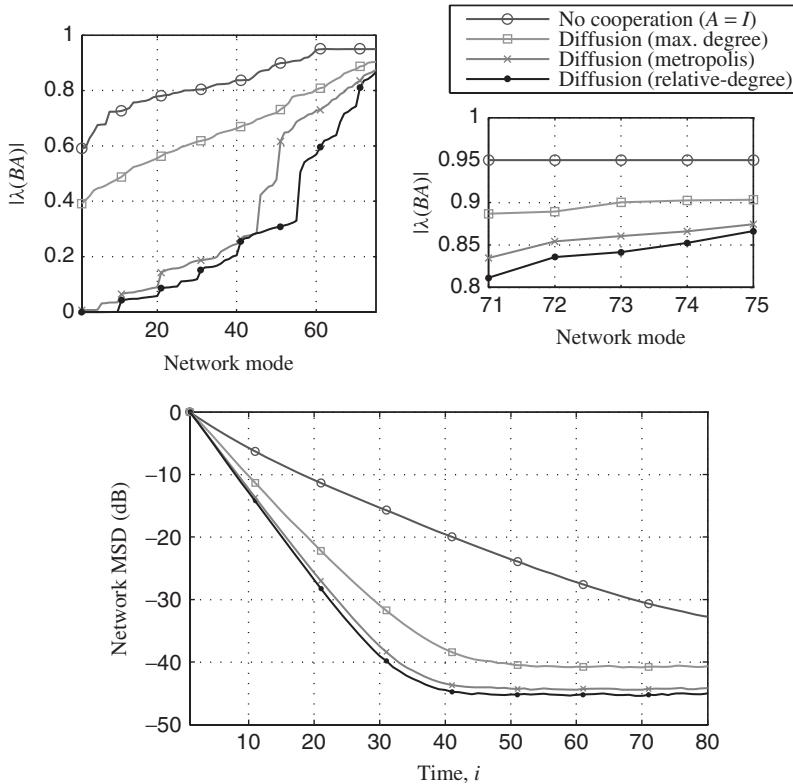
$$|\lambda_{\max}(\mathcal{BA})| \leq \|A\|_2 \cdot |\lambda_{\max}(\mathcal{B})|. \quad (22.53)$$

That is, the network mean stability depends on the local data statistics (represented by  $\mathcal{B}$ ) and on the cooperation strategy (represented by  $A$ ). Whenever a combiner rule is picked so that  $\|A\|_2 \leq 1$ , the cooperative scheme will enforce robustness over the noncooperative scheme. For combiners that render stochastic and symmetric matrices  $A$ , we have that  $\|A\|_2 = 1$ . As a result, we conclude that

$$|\lambda_{\max}(\mathcal{BA})| \leq |\lambda_{\max}(\mathcal{B})|. \quad (22.54)$$

In other words, the spectral radius of  $\mathcal{BA}$  is generally smaller than the spectral radius of  $\mathcal{B}$ . Hence, cooperation under the diffusion protocol (22.40) has a *stabilizing* effect on the network.

Figure 22.13 presents a simulation example for the network defined in Figure 22.6, and a value  $\mu_k = 0.05$ . Here we show the magnitude of the network modes (i.e., the magnitude of the eigenvalues of  $\mathcal{BA}$ ) for all nodes in the network, for a total of  $MN$  modes. We present the modes when there is no cooperation ( $A = I$ ), and when cooperation is introduced through the diffusion algorithms, for different choices of weighting matrices, namely, maximum-degree weights, Metropolis weights, and relative-degree weights. Note how cooperation significantly decreases the eigenmodes of the mean weight error evolution, as compared with the noncooperative scheme, thus yielding faster convergence. The top-right plot of Figure 22.13 zooms on the largest eigenmodes, which generally determine the convergence speed. Again the diffusion algorithms



**Figure 22.13** Comparison of diffusion schemes using different weighting matrices and the case where there is no cooperation. The simulation uses  $N = 15$ ,  $M = 5$  (75 modes in total),  $\mu_k = 0.05$  and network statistics as in Figure 22.5. All diffusion schemes use the ATC diffusion LMS algorithm (22.41). Plots include the network modes for different choices of weighting matrices (top left plot), a zoomed-in version showing the largest eigenmodes (top right plot), and the MSD learning curves (bottom plot).

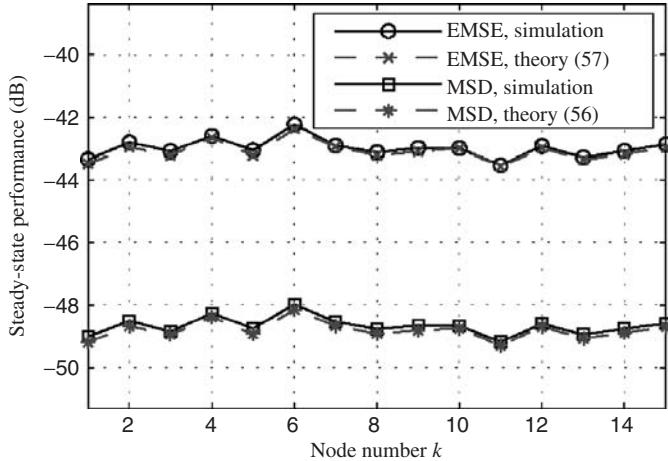
outperform the no-cooperation case. The bottom plot shows the MSD learning curves for different choices of weighting matrices.

#### 22.4.6 Mean-Square Performance

We may also examine the mean-square performance of the diffusion schemes. A detailed mean-square and stability analysis of the CTA diffusion algorithm (22.40) is performed in [10]. The following results are simplifications of the general expressions derived in [10]; The simplification assumes a uniform statistical profile across the network, that is,  $R_{u,k} = R_u$ ,  $R_{du,k} = R_{du}$ , and  $\sigma_{v,k}^2 = \sigma_v^2$  for all  $k$ , as well as uniform and sufficiently small step sizes,  $\mu_k = \mu$ . The results in [10] apply to the more general scenario of varying statistical profile and step sizes across the nodes.

Introduce the  $\text{vec}(\cdot)$  notation, which transforms an  $M \times M$  matrix  $X$  into an  $M^2 \times 1$  column vector  $x$  by stacking the columns of  $X$  on top of each other:

$$x = \text{vec}(X).$$



**Figure 22.14** Steady-state performance of the CTA diffusion LMS algorithm (22.40).

In the case of small step-size  $\mu$ , simplified expressions for the MSD and EMSE for the CTA diffusion algorithm (22.40) can be described as follows. Introduce the quantities:

$$D \triangleq I - A^T \otimes (I - \mu R_u^T) \otimes A^T \otimes (I - \mu R_u), \quad (22.55)$$

$$a \triangleq \mu^2 \sigma_v^2 \cdot \text{vec}(I_N \otimes R_u),$$

$$b_k \triangleq \text{vec}(\text{diag}(e_k) \otimes R_u),$$

$$q_k \triangleq \text{vec}(\text{diag}(e_k) \otimes I_M).$$

Then

$$\eta_k \approx a^T D^{-1} q_k \quad (\text{MSD}), \quad (22.56)$$

$$\zeta_k \approx a^T D^{-1} b_k \quad (\text{EMSE}), \quad (22.57)$$

where  $e_k$  denotes the  $M \times 1$  basis vector with 1 corresponding to the position of the  $k$ th node and zeros elsewhere. Observe how the network topology influences the performance through the matrix  $A$ , which appears in the expressions for the MSD and the EMSE.

Figure 22.14 shows the steady-state performance of the CTA diffusion LMS algorithm (22.40), both for simulation and the theoretical results of (22.56) and (22.57), for a network with  $N = 15$  nodes,  $\mu_k = 0.02$ , and uniform noise variances and regressor covariances across the nodes.

## 22.5 CONCLUDING REMARKS

This chapter describes several distributed and cooperative algorithms that endow networks with learning abilities. The algorithms address distributed estimation problems

that arise in a variety of applications, such as environment monitoring, target localization, and sensor network problems.

Although the chapter focused on algorithms of the LMS type, several other extensions are possible and have been pursued including algorithms of the least-squares type as well as Kalman filtering and smoothing procedures. The objective of this chapter has been to introduce the main ideas and to illustrate them by focusing on simpler algorithms for the benefit of clarity.

## ACKNOWLEDGMENTS

This material was based on work supported in part by the National Science Foundation under awards ECS-0601266 and EECS-0725441.

## REFERENCES

1. D. Estrin, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," In *Proc. ICASSP*, Salt Lake City, UT, May 2001, pp. 2033–2036.
2. D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayed, "Detection, classification, and tracking of targets," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 17–29, Mar. 2002.
3. I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.
4. L. A. Rossi, B. Krishnamachari, and C.-C. J. Kuo, "Distributed parameter estimation for monitoring diffusion phenomena using physical models," in *Proc. IEEE Conf. Sensor and Ad Hoc Comm. and Networks*, Santa Clara, CA, Oct. 2004, pp. 460–469.
5. D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, Aug. 2004.
6. J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 56–69, July 2006.
7. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *Proc. ICASSP*, Las Vegas, Apr. 2008, pp. 3285–3288.
8. A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. Fund. Electron. Commun. Computer Sci.*, vol. E90-A, no. 8, pp. 1504–1510, 2007.
9. C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.
10. C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, July 2008.
11. F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, May 2008.
12. J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1078–1090, Mar. 2006.
13. C. G. Lopes and A. H. Sayed, "Distributed adaptive incremental strategies: Formulation and performance analysis," in *Proc. ICASSP*, Vol. 3, Toulouse, France, May 2006, pp. 584–587.
14. C. G. Lopes and A. H. Sayed, "Diffusion least-mean-squares over adaptive networks," in *Proc. ICASSP*, Vol. 3, Honolulu, HI, Apr. 2007, pp. 917–920.

15. A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," in *Proc. Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Oct. 2006, pp. 233–237.
16. F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "A diffusion RLS scheme for distributed estimation over adaptive networks," in *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Helsinki, Finland, June 2007, pp. 1–5.
17. F. S. Cattivelli and A. H. Sayed, "Diffusion mechanisms for fixed-point distributed Kalman smoothing," in *Proc. EUSIPCO*, Lausanne, Switzerland, Aug. 2008.
18. F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering: Formulation and performance analysis," in *Proc. IAPR Workshop on Cognitive Information Processing*, Santorini, Greece, June 2008.
19. D. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. Optim.*, vol. 7, no. 4, pp. 913–926, Nov. 1997.
20. A. Nedic and D. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM J. Optim.*, vol. 12, no. 1, pp. 109–138, 2001.
21. J. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Automatic Control*, vol. AC-31, no. 9, pp. 650–655, Sept. 1986.
22. B. T. Poljak and Y. Z. Tsyplkin, "Pseudogradient adaptation and training algorithms," *Automatic Remote Control*, vol. 12, pp. 83–94, 1978.
23. M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.
24. A. H. Sayed, *Fundamentals of Adaptive Filtering*, Hoboken, NJ: Wiley, 2003.
25. A. H. Sayed, *Adaptive Filters*, Hoboken, NJ: Wiley, 2008.
26. V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. Joint 44th IEEE Conf. on Decision and Control and European Control Conf. (CDC-ECC)*, Seville, Spain, Dec. 2005, pp. 2996–3000.
27. W. Kocay and D. L. Kreher, *Graphs, Algorithms and Optimization*, Hoboken, NJ: Chapman & Hall/CRC Press, 2005.
28. D. S. Scherber and H. C. Papadopoulos, "Locally constructed algorithms for distributed computations in ad-hoc networks," in *Proc. Information Processing in Sensor Networks (IPSN)*, Berkeley, CA, Apr. 2004, pp. 11–19.
29. L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, Sept. 2004.
30. L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. Information Processing in Sensor Networks*, Los Angeles, CA, Apr. 2005, pp. 63–70.
31. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, 1953.
32. W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

## CHAPTER 23

---

# Routing for Statistical Inference in Sensor Networks

A. Anandkumar<sup>1</sup>, A. Ephremides<sup>2</sup>, A. Swami<sup>3</sup>, and L. Tong<sup>1</sup>

<sup>1</sup>Cornell University

<sup>2</sup>University of Maryland

<sup>3</sup>U.S. Army Research Laboratory

### 23.1 INTRODUCTION

Routing in communication networks, both wireline and wireless, has been a subject of extensive and in-depth study over the last few decades. It is a subject that is fairly well understood. Its state-of-the-art status can be summarized as follows: If a well-defined performance measure can be translated to a link metric, then there are low-complexity, efficient, robust, fast-converging, and often distributed algorithms for finding the optimal routes. Note the important distinction regarding the possibility of mapping the performance measure to a link metric. For example, on the Internet, if end-to-end latency is the performance measure, then the link metric is delay over the link. Bellman-Ford types of algorithms then perform very well and quickly discover the best routes [1]. By contrast, on the traditional circuit-switched voice telephone network, where the performance measure is blocking probability, there is no known link metric that captures the performance measure and, hence, up to this day we only have heuristic routing algorithms for assigning routes to accepted calls.

At this point it is also important to note that the routing problem, being basically a discrete optimization task, has always a default solution that consists of the exhaustive search over the finite number of possible routes. The only reason this solution is unattractive is the prohibitive complexity of this search as the network size increases.

Another example of successful mapping of a performance measure to a link metric that allows the use of efficient algorithms is energy consumption in a wireless network. The energy consumption on a single link is then the right metric. That link energy consumption, depending on the assumptions on the network operation, consists of the transmission energy (proportional to the transmission power needed to reach the destination at a given rate and bit-error-rate target for chosen modulation and coding schemes, as well as to the channel attenuation), the energy expended for reception at the receiving end of the link, and, finally, the residual energy at the battery of the node at that end.

What all routing problems to date share is the traditional integer program (IP) paradigm of store-and-forward, which treats the source packets as “sacrosanct” monoliths that must be carried through the network intact until they are received at the destination node. Already, the idea of network coding has shown how it is possible to improve performance if this paradigm is reconsidered [2]. In this chapter we will examine a different issue that arises in specialized routing that shows equally well the inadequacy of traditional packet forwarding.

Our focus will be the case of wireless sensor networks. The unique characteristic of such networks is that the performance measure is typically associated with the “mission” of the network. For example, if the sensor network is deployed for the purpose of detecting the presence of a target, then the objective is to maximize the probability of correct detection, subject to the usual constraint of the false alarm rate. In other words, the mission of the network is statistical inference. Thus, the collected measurement data at the source sensors need not be forwarded to the fusion center (i.e., the ultimate destination node) in their entirety. Of course, such complete forwarding remains an option (just as the exhaustive search over all possible routes was an option in ordinary routing). But it is an inefficient option that is highly undesirable in networks that must also prolong their lifetime as much as possible. Wireless sensor networks often do not have the possibility of recharging or replacing the node batteries, and thus it is important to “compress” the source data as much as possible. For statistical inference we do know that the collected data often map into a “sufficient statistic” that consists of substantially fewer bits than the original data [3]. In addition, intermediate nodes that are chosen to route the sufficient statistic information from a neighboring node make their own measurements as well and therefore need to combine the received information with their own data to form a collective sufficient statistic for further forwarding. Thus, the problem of routing is intimately intertwined with the process of statistical inference in a novel way. It calls for a distributed computation of a global sufficient statistic that is based on all the collected measurements but with as little energy expenditure as possible for the needed data exchange among the nodes.

In fact, this distributed computation process under energy efficiency constraints is a prime example of cross-layer optimization in wireless networks. It couples the process of routing with the physical layer (where the energy expenditure occurs) and the application layer (where the statistical inference takes place). And it is an example of a totally new and unexplored aspect of wireless sensor operation. At the same time it raises some fundamental issues of energy consumption for distributed computation of a function and introduces the trade-off between communication and computation that has been examined before in different contexts.

In this chapter we aim at a comprehensive presentation of this new aspect of wireless sensor networking and at a unified study of routing, inference, and energy consumption. Inherent in this presentation is the notion of combinatorial optimization (which remains the underpinning element of the routing task) and of spatial information modeling (which defines the information dependencies in the data the sensor nodes gather). The treatment is self-contained but depends on several pieces of recent work that is referred to and summarized at appropriate parts of the chapter.

## 23.2 SPATIAL DATA CORRELATION

In many realistic scenarios the sensor measurements are correlated, and our framework takes this into account. Examples of correlated signals include temperature and

humidity sensors and magnetometric sensors tracking a moving vehicle. Acoustic data are rich in spatial correlations due to the presence of echoes caused by multipath reflections. Moreover, in general, spatial signals are acausal in contrast to temporal signals. In the literature, the two are usually distinguished by referring to acausal signals as random fields and to causal signals as random processes. An example of exploiting correlation in a causal propagation setting can be found in [4, 5].

The model for spatially correlated data crucially affects in-network processing of raw data. Various assumptions on correlation have been made in the literature. Joint Gaussian distributions and distance-based correlation function have been widely assumed due to their simplicity [6–9]. Alternatively, diffusion-based [10] and joint-entropy-based models [11] have also been employed. The use of remote-sensing data, proposed in [12], may not meet the resolution requirements. The model proposed in [13] is a special case of a Markov random field (MRF).

Markov random fields, as a class of parametric models for spatial data, were introduced by Besag [14, 15] and were known as conditional autoregressions in his works. Prior to these works, Hammersley and Clifford formulated their now famous theorem on the equivalence of MRF to a Gibbs field [16]. However, the manuscript was never published, and a sketch of the original proof can be found in [17], along with further discussion on the historical aspects of research on MRF.

The use of the MRF model for spatial data in sensor networks is relatively new (e.g., [18]), although it is widely used in image processing [19] and geostatistics [20]. This could be due to the complexity of the model for arbitrarily placed nodes. We will see that the use of an MRF model leads to the formation of “clusters” that are based on the statistical dependence, rather than other considerations such as residual energy [21, 22]. The notion of clustering has been used extensively in sensor networks, where nodes send their data to one member of the cluster, which then processes and forwards to the destination. However, here, the issues are complicated by the fact that measurements processed in these statistical “clusters” have to be further aggregated rather than simply being forwarded to the destination.

We assume that all the sensors know the MRF model. In practice, the dependency structure and the model parameters of the MRF model can be estimated by incorporating a training phase. The seminal work of Chow and Liu in [23] considers the problem of approximating an unknown distribution from its samples using a procedure for learning the tree model that maximizes the likelihood of the training samples among the set of all possible tree models. Recently, learning graphical models from data samples specifically for binary hypothesis testing has been considered in [24]. Their procedure learns each hypothesis model from both sets of training samples.

In this chapter, we employ the MRF model for spatial correlation, taking into account only its graphical dependency structure; but no parametric correlation function is assumed. Moreover, any general random field without special properties can be represented as an MRF with a complete dependency graph (called the *saturated models* [25]).

### 23.2.1 Notations and Basic Definitions

An undirected graph  $G$  is a tuple  $G = (V, E)$ , where  $V$  is the vertex set and  $E = \{(i, j)\}$  is the edge set. We allow graphs to have multiple or parallel edges but no loops. The neighborhood function  $\mathcal{N}(i; G)$  of a node  $i$  is the set of all other nodes having an edge with it in  $G$ . Let  $\text{Deg}(i)$  denote the degree of node  $i$ . A subgraph

induced by  $V' \subset V$  on  $G$  is denoted by  $G(V')$ , and a complete subgraph or a clique has edges between any two nodes in  $V'$ . A maximal clique is one that is not contained in any other clique. Throughout this chapter, a clique refers to a maximal clique, unless otherwise mentioned. For a directed graph (digraph), we denote the edges (arcs) by  $< i, j >$ , where the direction is from  $i$  to  $j$ , and node  $j$  belongs to the set of immediate successors of  $i$ , and  $i$  is in the set of immediate predecessor of  $j$ . The above graph functions are extended to sets, for example,  $(i, A)$  denotes the set of edges between  $i$  and members of  $A$ . For sets  $A$  and  $B$ , let  $A \setminus B = \{i : i \in A, i \notin B\}$  and let  $|\cdot|$  denote cardinality. For matrix  $\mathbf{A}$ ,  $A(i, j)$  is the element in the  $i$ th row and  $j$ th column and  $|\mathbf{A}|$  its determinant.

### 23.2.2 Definition and Properties of MRF

The MRF falls under the framework of acausal graphical models and satisfies conditional-independence properties, based on an undirected graph known as the *dependency graph* and is defined below.

**Definition 23.1 Markov Random Field** Let  $\mathbf{Y}_V = [Y_i, i \in V]^T$  denote the random vector of measurements in set  $V$ .  $\mathbf{Y}_V$  is a Markov random field with an (undirected) dependency graph  $G_d = (V, E_d)$ , if  $\forall i \in V$ ,

$$Y_i \perp \mathbf{Y}_{V \setminus \{i, \mathcal{N}(i)\}} | \mathbf{Y}_{\mathcal{N}(i)}, \quad (23.1)$$

where  $\perp$  denotes conditional independence.

In words, the above definition states that the value at any node, given the values at its neighbors, is conditionally independent of the rest of the network.

**23.2.2.1 Example: One-Dimensional MRF** A simple example is the first-order autoregressive (AR-1) process, given by

$$Y_t = A_{t-1} Y_{t-1} + \epsilon_{t-1}, \quad Y_{t-1} \perp \epsilon_{t-1}, \quad \forall t \in V = \{1, \dots, n\}. \quad (23.2)$$

Since  $Y_t$  is conditionally independent of the past, given the measurement  $Y_{t-1}$ , we write

$$Y_t \perp Y_{1, \dots, t-2} | Y_{t-1}, \quad 2 < t \leq n.$$

Similarly, we can write

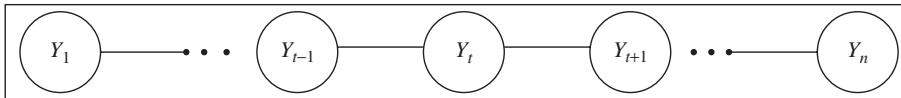
$$Y_{t+2, \dots, n} \perp Y_t | Y_{t+1}, \quad 1 \leq t < n.$$

This implies that

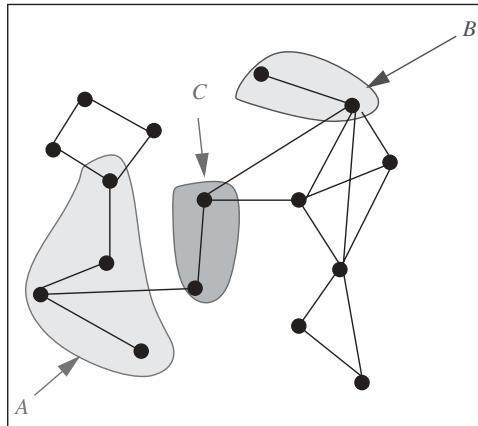
$$\begin{aligned} Y_t \perp \mathbf{Y}_{V \setminus \{t-1, t, t+1\}} | \{Y_{t-1}, Y_{t+1}\}, \quad \forall t = 2, \dots, n-1, \quad Y_1 \perp \mathbf{Y}_{V \setminus \{1, 2\}} | Y_2, \\ Y_n \perp \mathbf{Y}_{V \setminus \{n, n-1\}} | Y_{n-1}. \end{aligned}$$

Hence, we have the dependency graph with neighborhood function

$$\mathcal{N}(t) = \{t-1, t+1\}, \quad \text{for } t \neq 1, n, \quad \mathcal{N}(1) = 2, \quad \mathcal{N}(n) = n-1.$$



**Figure 23.1** Linear dependency graph for an MRF representation of autoregressive process of order 1.



**Figure 23.2** Global markov property:  $\mathbf{Y}_A$  is conditionally independent of  $\mathbf{Y}_B$  given  $\mathbf{Y}_C$ .

In other words, the dependency graph is a linear chain, as shown in Figure 23.1. Hence, the conditional independence relations of the AR-1 process have a simple graphical representation that is not apparent in (23.2). However, the dependency graph does not capture all the information of the AR-1 process, in particular, that the process is causal. On the other hand, the dependency graph can be used to model more general acausal dependencies, typically found in spatial random fields.

**23.2.2.2 Properties of General MRF** For a Markov random field, in fact, three types of Markov properties can be defined:

1. Local markov property:  $Y_i \perp \mathbf{Y}_{V \setminus (i \cup \mathcal{N}(i))} | \mathbf{Y}_{\mathcal{N}(i)}$ ,  $\forall i \in V$ .
2. Global markov property:  $\mathbf{Y}_A \perp \mathbf{Y}_B | \mathbf{Y}_C$ , where  $A, B, C$  are disjoint sets.  $A, B$  are nonempty and  $C$  separates  $A, B$ . See Figure 23.2.
3. Pairwise markov property:  $Y_i \perp Y_j | \mathbf{Y}_{V \setminus \{i, j\}} \iff (i, j) \notin E$ .

In Definition 23.1, we have used the local Markov property. We can immediately see that the global Markov property implies the local Markov property, since we can set

$$A = \{i\}, B = V \setminus \{\mathcal{N}(i)\}, C = \mathcal{N}(i).$$

Similarly, the global Markov property implies the pairwise Markov property since we can set

$$A = \{i\}, B = \{j\}, C = V \setminus \{i, j\}, \quad \forall (i, j) \notin E_d.$$

The three properties can be shown to be equivalent under the positivity condition [25]. The positivity condition is as follows: For all  $A \subset V$  with samples  $\mathbf{y}_A, \mathbf{y}_{V \setminus A}$  such that  $f(\mathbf{y}_A), f(\mathbf{y}_{V \setminus A}) > 0$ , the conditional is also positive

$$f(\mathbf{y}_A | \mathbf{y}_{V \setminus A}) > 0,$$

where  $f$  is the density function. An equivalent condition for positivity is

$$(f(\mathbf{y}_V) = 0) \Rightarrow (f(y_i) = 0), \quad \forall i \in V.$$

An example that does *not* satisfy positivity is the fully correlated case:  $Y_1 = Y_2 \dots = Y_n$ . In this case, the joint likelihood is zero whenever all the samples are not equal, but the marginal likelihood is not necessarily zero.

The Hammersley–Clifford theorem [17] states that for an MRF  $\mathbf{Y}_V$  with dependency graph  $G_d = (V, E_d)$ , the joint probability density function (pdf)  $f$ , under the positivity condition, can be expressed as

$$-\log f(\mathbf{Y}_V; \Upsilon) = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{Y}_c), \quad (23.3)$$

where  $\mathcal{C}$  is a collection of (maximal) cliques in  $G_d$ , the functions  $\psi_c$ , known as *clique potentials*, are real valued, nonnegative, and not zero everywhere on the support of  $\mathbf{Y}_c$ . Thus, the tuple  $\Upsilon = \{G_d, \mathcal{C}, \psi\}$  specifies the MRF in (23.3). We assume that the normalization constant is already incorporated in the potential functions in order to ensure that we have a valid pdf. For general potentials, finding the normalizing constant (called the *partition function*) is NP-hard, but approximate algorithms have been proposed in [26].

From (23.3), we see that the complexity of the likelihood function is vastly reduced for sparse dependency graphs; here, the conditional-independence relations in (23.1) results in the factorization of the joint likelihood into a product of components, each of which depends on a small set of variables. This form is already exploited by distributed algorithms such as belief propagation [27] for local inference of hidden measurements. In this chapter, we exploit the MRF model for a global inference problem, explained in Section 23.3.

In this chapter, we assume that the number of cliques  $|\mathcal{C}|$  of the MRF is polynomial in the number of nodes. This is satisfied by many graph families such as bounded-degree graphs [28]. Note that in (23.3), the set of cliques  $\mathcal{C}$  contains only those cliques with nonzero potentials. For example, for independent measurements,  $\mathcal{C}$  is the vertex set, and we have the likelihood function as a weighted sum function:

$$-\log f(\mathbf{Y}_V) = - \sum_{i \in V} \log f_i(Y_i), \quad \mathbf{Y}_V \sim \prod_{i \in V} f_i,$$

where  $f_i$  is the marginal pdf of  $Y_i$ . Besag's automodel [15] is a special MRF with only pairwise dependencies, and hence the clique set  $\mathcal{C}$  is the set of edges  $E_d$ . This leads to a simplified expression for the likelihood function,

$$-\log f(\mathbf{Y}_V; \{G_d, E_d, \psi\}) = \sum_{(i,j) \in E_d} \psi_{i,j}(Y_i, Y_j). \quad (23.4)$$

Multiparameter exponential family of conditional probabilities can be used to define such pairwise Markov random fields [29]. An example of Besag's model is the Ising model, which was first introduced to study phase transition in ferromagnetic materials.

**23.2.2.3 Gauss–Markov Random Field** The Gauss–Markov random field (GMRF) has some special properties. In this case, (23.3) is equivalent to (23.4) since the likelihood function of  $\mathbf{Y}_n \sim \mathcal{N}(\mathbf{0}, \Sigma)$  is given by

$$\log f(\mathbf{Y}_V; \mathbf{A}) = \frac{1}{2} \left( -n \log 2\pi + \log |\mathbf{A}| + \sum_{i \in V} A(i, i) Y_i^2 + \sum_{i, j \in V} A(i, j) Y_i Y_j \right), \quad (23.5)$$

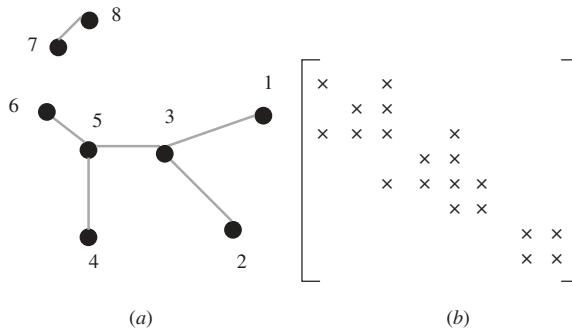
where  $\mathbf{A} := \Sigma^{-1}$  is the inverse of the covariance matrix. For a given dependency graph  $G_d = (V, E_d)$ , the GMRF should also satisfy (23.4). Hence, comparing the two equations (23.4) and (23.5), we have

$$A(i, j) = 0 \iff (i, j) \notin E_d.$$

Hence, there is a one-to-one correspondence between the nonzero elements of  $\mathbf{A}$  and the dependency graph edges  $E_d$  and is illustrated in Figure 23.3. Since  $\mathbf{A}$  is associated with the potentials, it is called the *potential matrix*. Hence, for the Gaussian distribution, we only need the edges of the dependency graph and not the higher order cliques. Moreover, for the Gaussian case, the edge potential  $\psi_{i,j}(Y_i, Y_j)$  in (23.4) reduces to the sum of squares and cross products of the measurements, weighted by the coefficients of the potential matrix  $\mathbf{A}$ . When the dependency graph is acyclic, we can additionally obtain a closed form for the elements of the potential matrix  $\mathbf{A}$ , in terms of the elements of the covariance matrix  $\Sigma$ .

**Fact 1 GMRF with Acyclic Dependency Graph** *The coefficients of the potential matrix  $\mathbf{A} := \Sigma^{-1}$ , with zero mean and covariance matrix  $\Sigma$  and acyclic dependency graph  $G_d = (V, E_d)$ , are*

$$A(i, i) = \frac{1}{\Sigma(i, i)} \left[ 1 + \sum_{j \in \mathcal{N}(i)} \frac{\Sigma(i, j)^2}{\Sigma(i, i)\Sigma(j, j) - \Sigma(i, j)^2} \right], \quad (23.6)$$



**Figure 23.3** One-to-one correspondence between dependency graph edges and nonzero elements of potential matrix for a GMRF. (a) Labeled simple undirected graph and (b)  $\times$ , nonzero elements of potential matrix.

$$A(i, j) = \begin{cases} \frac{-\Sigma(i, j)}{\Sigma(i, i)\Sigma(j, j) - \Sigma(i, j)^2} & \text{if } (i, j) \in E_d, \\ 0 & \text{otherwise.} \end{cases} \quad (23.7)$$

The determinant of the potential matrix of  $\mathbf{A}$  is given by

$$|\mathbf{A}| = \frac{1}{|\Sigma|} = \frac{\prod_{i \in V} \Sigma(i, i)^{\text{Deg}(i)-1}}{\prod_{\substack{(i, j) \in E_d \\ i < j}} [\Sigma(i, i)\Sigma(j, j) - \Sigma(i, j)^2]}. \quad (23.8)$$

In fact, for any MRF with acyclic dependency graph  $G_d$ , the joint pdf  $f_{\mathbf{Y}_V}$  can be expressed in terms of marginals at nodes  $f_{Y_i}$  and pairwise joint pdf's  $f_{Y_i, Y_j}$  as

$$f_{\mathbf{Y}_V}(\mathbf{Y}_V) = \prod_{i \in V} f_{Y_i}(y_i) \prod_{(i, j) \in E_d} \frac{f_{Y_i, Y_j}(y_i, y_j)}{f_{Y_i}(y_i)f_{Y_j}(y_j)}. \quad (23.9)$$

See [30] for details.

### 23.3 STATISTICAL INFERENCE OF MARKOV RANDOM FIELDS

The problem of distributed detection considers a set of sensors, one of them designated as the fusion center or the decision node, and all the sensor observations are ultimately routed (in some form) to it. This setup is relevant when we need to make a global decision on the phenomenon (contrasting to local inference algorithms such as belief propagation). We consider the binary hypothesis-testing problem with two given hypotheses, the null hypothesis  $\mathcal{H}_0$  and the alternative  $\mathcal{H}_1$ . We limit ourselves to only simple hypothesis testing, that is, the probability measures under both the hypotheses are known to all the sensors.

In statistical theory, a *sufficient statistic* is a well-behaved function of the data, which is as informative as the raw data for inference. Formally, a function  $T(Y)$  is said to be a sufficient statistic for model  $P_\theta$ , if conditioned on  $T(Y)$ ,  $Y \sim P_\theta$  does not depend on  $\theta$ . It is said to be *minimal* if it is a function of every other sufficient statistic for  $P_\theta$  [3]. A minimal sufficient statistic for inference represents the maximum possible reduction in dimensionality of the raw data, without destroying information about the underlying phenomenon [3]. The log-likelihood ratio (LLR) is the minimal sufficient statistic for hypothesis testing [31]. Let  $f(\mathbf{Y}_V; \mathcal{H}_j)$  be the pdf of the measurements  $\mathbf{Y}_V$  under hypothesis  $j$ . The optimal decision rule at the fusion center is a threshold test based on the LLR:

$$\text{LLR}(\mathbf{Y}_V) := \log \frac{f(\mathbf{Y}_V; \mathcal{H}_0)}{f(\mathbf{Y}_V; \mathcal{H}_1)}. \quad (23.10)$$

The result is also true for the  $M$ -ary hypothesis-testing problem, where the LLR vector

$$\left[ \log \frac{f(\mathbf{Y}_V; \mathcal{H}_0)}{f(\mathbf{Y}_V; \mathcal{H}_1)}, \dots, \log \frac{f(\mathbf{Y}_V; \mathcal{H}_0)}{f(\mathbf{Y}_V; \mathcal{H}_{M-1})} \right]^T$$

is minimally sufficient.

### 23.3.1 Form of Log-Likelihood Ratio for MRF

In this chapter, we assume that the measurement samples are drawn from distributions specified by distinct MRFs, defined on the same node set. In particular, we consider

$$\mathcal{H}_0: \Upsilon_0 = \{G_0(V), \mathcal{C}_0, \psi_0\} \quad \text{vs.} \quad \mathcal{H}_1: \Upsilon_1 = \{G_1(V), \mathcal{C}_1, \psi_1\}. \quad (23.11)$$

From (23.3) and (23.10), the LLR is given by the difference of the respective clique potentials:

$$\text{LLR}(\mathbf{Y}_V) = \sum_{a \in \mathcal{C}_1} \psi_{1,a}(\mathbf{Y}_a) - \sum_{b \in \mathcal{C}_0} \psi_{0,b}(\mathbf{Y}_b). \quad (23.12)$$

It is easily seen that the LLR can be expressed as the sum of potentials of an “effective” Markov random field  $\Upsilon = \{G_d, \mathcal{C}, \phi\}$  specified as follows: The effective dependency graph  $G_d = (V, E_d)$ , has the edge set  $E_d = E_0 \cup E_1$ ; the effective clique set is  $\mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1$ , with only the resulting maximal cliques retained; the effective potential functions  $\phi_c$  are given by

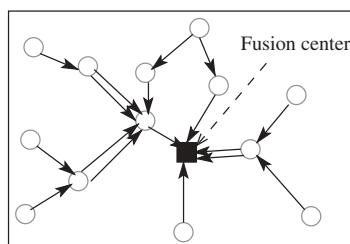
$$\phi_c(\mathbf{Y}_c) := \sum_{a \in \mathcal{C}_1, a \subset c} \psi_1(\mathbf{Y}_a) - \sum_{b \in \mathcal{C}_0, b \subset c} \psi_0(\mathbf{Y}_b), \quad \forall c \in \mathcal{C}. \quad (23.13)$$

Therefore, the LLR has a succinct form, which will be used in the rest of this chapter:

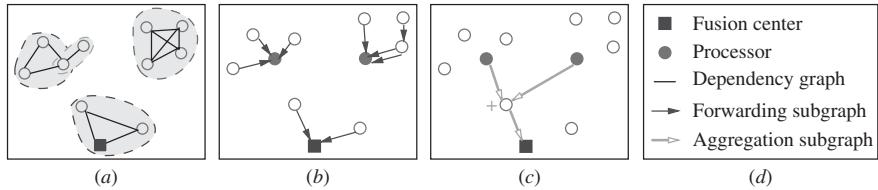
$$\text{LLR}(\mathbf{Y}_V; \Upsilon) = \sum_{c \in \mathcal{C}} \phi_c(\mathbf{Y}_c). \quad (23.14)$$

## 23.4 OPTIMAL ROUTING FOR INFERENCE WITH LOCAL PROCESSING

By optimal routing for inference, we mean the fusion scheme that minimizes the total costs of routing under the constraint that the likelihood function in (23.14) is delivered to the fusion center (Fig. 23.4). Such a scheme involves computing the likelihood function consisting of clique potential functions, each depending only on the measurements in the clique. Hence, these clique potential functions can be computed independently at various nodes. To exploit the Markovian structure of the underlying hypotheses, we consider a class of data fusion schemes that perform local processing within the cliques of the MRF. Specifically, an aggregation scheme involves the following considerations,



**Figure 23.4** Routing to designated fusion center is represented through a digraph. Each arc represents the routing path of one packet, carrying one real number.



**Figure 23.5** Schematic of dependency graph of Markov random field and stages of data aggregation. The forwarding and aggregation subgraphs transport raw and aggregated data. (a) Maximal cliques of dependency graph, (b) forwarding subgraph computes clique potentials, (c) aggregation subgraph adds computed potentials, and (d) legend.

namely each clique potential is assigned a computation site or a processor; measurements of the clique members are then transported to its processor to enable computation of the clique potentials. These values are then summed up and delivered to the fusion center. See Figure 23.5.

#### **23.4.1 Network and Communication Model**

We assume the presence of a medium-access control (MAC) that eliminates collisions or interferences among the nodes. The network is connected, that is, communication is feasible via a multihop route between any two nodes in the network. We assume that communication is bidirectional. We consider the unicast mode of routing, where a packet from a node is routed to a single destination, and the intermediate nodes do not perform any processing or store the packet for future use.

In our formulation, the processing costs are assumed constant, thus ignored in the optimization. Usually, the routing costs reflect transmission energy, but it could also represent, for example, delay, bandwidth, or a combination of these considerations. We represent the routing of a real number by a packet. A symmetric routing cost function is assumed and is denoted by  $C_{i,j} > 0$  between  $i$  and  $j$ . The metric closure on graph  $G$ , is defined as the complete graph where the cost of each edge  $(i, j)$  in the metric closure is the cost of the shortest path between  $i$  and  $j$  in  $G$  [32, p. 58]. Henceforth, we only consider the metric closure of the communication graph, denoted by  $G_c$ , and denote the metric costs by  $\bar{C}_{i,j}$ . There is no loss of generality since the edges of the metric closure can be replaced with the corresponding shortest paths. For any graph  $G \subset G_c$ , let  $C(G)$  denote the total cost of its links:

$$C(G) := \sum_{e \in E} C_e, \quad (23.15)$$

where  $C_e$  is the cost of the link  $e$  and  $E$  is the set of links in  $G$ ; if a link is used  $m$  times, then  $E$  contains  $m$  parallel links to incorporate the costs in our formulation.

In our formulation all real numbers are quantized with sufficiently high precision to ignore the quantization error and all nodes function as both sensors and routers. Quantization is indeed an important issue for detection and communication. However, even in the classical distributed setup, optimal quantization is not tractable for the correlated case. The recent works on this topic consider conditionally independent and identically distributed (i.i.d.) measurements with a fixed network topology of bounded-height tree [33] or a tandem network [34].

### 23.4.2 Formulation of Minimum Cost Fusion

Recall the succinct form of LLR in (23.14):

$$\text{LLR}(\mathbf{Y}_V; \Upsilon) = \sum_{c \in \mathcal{C}} \phi_c(\mathbf{Y}_c). \quad (23.16)$$

Hence, the LLR consists of the sum of the clique potential functions  $\phi$  and is amenable to localized processing within the cliques of the MRF. Hence, we propose a hierarchical order of processing the LLR. In the first stage, raw data are forwarded to compute all the potential functions at various nodes in the network. In the second stage, the computed values are summed up and delivered to the fusion center.

For the first stage of LLR computation, each clique potential function  $\phi_c$  is assigned a unique computation site, known as the *processor* for clique  $c$ , denoted by  $\text{Proc}(c)$ . Once the processor for clique  $c$  is assigned, measurement  $Y_i$  of each clique member  $i \subset c$  (other than the processor) is routed to  $\text{Proc}(c)$  along a path of feasible communication links. Since we are considering unicast mode of communication, the minimum cost is along the shortest path represented by the link  $\langle i, \text{Proc}(c) \rangle \in G_c$  with cost  $\overline{C}(i, \text{Proc}(c))$ , where  $G_c$  is the metric closure of the communication graph. The set of all links used by a fusion scheme in the first stage of computation to forward raw data to the processors is called the *forwarding* subgraph, denoted by  $\text{FG}$ ,

$$\text{FG} := \{\langle i, \text{Proc}(c) \rangle : i \subset c, i \neq \text{Proc}(c), c \in \mathcal{C}\}.$$

In the second stage of LLR computation, all the computed potential functions are summed up to obtain the LLR, which is then delivered to the fusion center. The set of links used by a fusion scheme in the second stage of LLR computation to sum up the computed potential values is known as the *aggregation* subgraph, denoted by  $\text{AG}$ . The tuple with the forwarding and aggregation subgraphs of a fusion scheme is referred to as the *fusion digraph*,  $G_f := \{\text{FG}, \text{AG}\}$ . A schematic of a fusion scheme is shown in Figure 23.5. The total routing costs of a fusion scheme is given by

$$C(G_f) = C(\text{FG}) + C(\text{AG}).$$

Hence, any fusion scheme in our setup is specified by a processor-assignment mapping  $\text{Proc}$  and a fusion digraph  $G_f = \{\text{FG}, \text{AG}\}$ , and we represent the scheme by the tuple  $\Gamma := \{\text{Proc}, \text{FG}, \text{AG}\}$ . Note that we do not explicitly specify the sequence in which data is transported and processed by a fusion scheme; we impose constraints to ensure that such a feasible sequence exists.

We first need the constraint that the scheme delivers the LLR to the fusion center

$$\text{AggVal}(v_0 \Gamma) = \text{LLR}(\mathbf{Y}_V; \Upsilon), \quad (23.17)$$

where  $\text{AggVal}(i; \Gamma)$  is the value at node  $i$  at the end of fusion.

**23.4.2.1 Local Processor Assignment** We now make the following additional assumption, which simplifies the fusion scheme: Each clique potential function  $\phi_c$  is assigned a “local” processor, which is one of the clique members:

$$\text{Proc}(c) \subset c, \quad \forall c \in \mathcal{C}. \quad (23.18)$$

The local processor assignment also implies that local knowledge of potential function parameters is sufficient, that is, each sensor  $i$  only needs to know the potential functions  $\phi_c$  of the cliques  $c$  to which it belongs, and, hence, the storage requirement at the sensors is considerably reduced. In practice, the potential function parameters are sent to the nodes by the fusion center after empirically estimating the joint pdf of the measurements. Through this, the nodes also implicitly receive information about their clique memberships. Hence, local processor assignment can also reduce the communication overhead during the learning stage. Localized processing can be especially efficient when the dependency graph of the Markov random field is a proximity graph, where edges are based on local point configuration [35]. We now formally define the minimum-cost fusion scheme  $\Gamma^*$ , which minimizes the total routing costs:

$$\Gamma^* := \arg \min_{\Gamma} C(G_f), \quad (23.19)$$

subject to the constraints in (23.17) and (23.18). Hence, the problem of minimum cost fusion takes the metric closure of communication graph and the maximal cliques of the dependency graph as inputs and provides a processor assignment and fusion digraph as outputs. An example of the problem of minimum cost fusion is illustrated in Figure 23.6, with the communication graph in Figure 23.6a and the chain dependency graph in Figure 23.6c, which are independent of one another. The resulting metric closure of communication graph in Figure 23.6b and cliques of dependency graph are taken as the inputs for the problem of minimum-cost fusion.

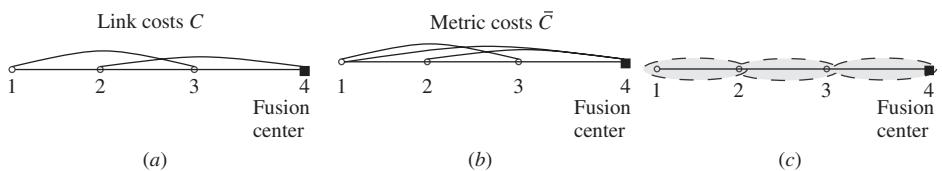
**23.4.2.2 0–1 Integer Programming Formulation** We now write a 0–1 integer program whose optimal solution provides the minimum cost fusion scheme in (23.19) for computing the LLR and delivering it to the fusion center  $v_0$ . We can map any valid fusion digraph  $G_f = \{FG, AG\}$  and the processor assignment mapping Proc to variables  $\mathbf{y}$  and  $\mathbf{z}$ , defined as

$$z(j, c) := I[\text{Proc}(c) == j], \quad y(i, j) := I[i < j \in AG],$$

where  $I$  is the indicator function. Once the processor assignment is fixed, the set of shortest paths from clique members to the processors minimizes the routing costs in the forwarding subgraph. Hence, we can set the forwarding subgraph as

$$FG \leftarrow \left\{ i < j : I \left( \sum_{c:i \subset c} z(j, c) \geq 1 \right) \right\},$$

where we ensure that every node  $i$  forwards its measurement to node  $j$ , whenever  $j$  is the processor of cliques  $c$  that contain node  $i$  along the link in the metric closure



**Figure 23.6** Inputs to the problem of minimum cost fusion for inference. (a) communication graph, (b) metric closure of communication graph, and (c) cliques of dependency graph.

(which has the same cost as the shortest path). Hence, the total routing costs of the fusion digraph can be expressed as

$$C(G_f) = C(\text{FG}) + C(\text{AG}) = \frac{1}{2} \sum_{i,j \in V} \left[ I \left( \sum_{c:i \in c} z(j, c) \geq 1 \right) + y(i, j) \right] \bar{C}(i, j),$$

where the factor of  $\frac{1}{2}$  ensures that each edge is counted only once. We now write a constraint equivalent to the local processor constraint in (23.21) and ensuring that at least one processor is selected,

$$\sum_{j \subset c} z(j, c) \geq 1, \quad \forall c \in \mathcal{C}.$$

We now need a constraint on the aggregation subgraph to ensure that the sum of the potential functions is delivered to the fusion center, and, hence, the constraint in (23.17) is satisfied. To this end, we define that  $A$  separates  $B$  if  $A \cap B \neq \emptyset$  and  $A \cup B \neq V$ . We consider all sets  $S \subset V$  separating the union of the set of processors and the fusion center. A cut edge of set  $S$  is one that has exactly one endpoint in  $S$ . As illustrated in Figure 23.7, since all the values at the processors contained within  $S$  can be summed up to a single packet, for the information to flow out of  $S$  (or into  $S$ ), at least one cut edge of  $S$  is needed. Hence, we write the constraint that

$$\sum_{i \in S, j \notin S} y(i, j) \geq 1, \quad \forall S \subset V \text{ separating } \left\{ \bigcup_{c \in \mathcal{C}} \text{Proc}(c) \cup v_0 \right\}.$$

We now have the integer program (IP)

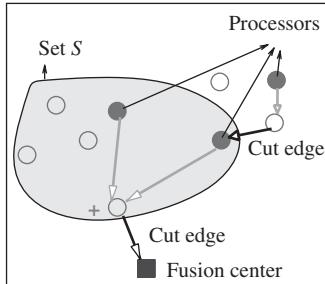
$$\frac{1}{2} \min_{\mathbf{y}, \mathbf{z}} \sum_{i,j \in V} [I(\sum_{c:i \in c} z(j, c) \geq 1) + y(i, j)] \bar{C}(i, j) \quad (\text{IP-1}), \quad (23.20)$$

$$\text{s.t. } \sum_{j \subset c} z(j, c) \geq 1, \quad \forall c \in \mathcal{C}, \quad \text{let } \text{Proc}(c) := \{j : z(j, c) = 1\}, \quad (23.21)$$

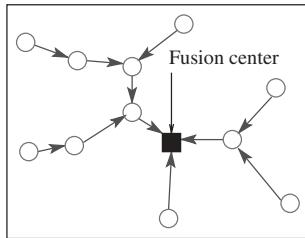
$$\sum_{i \in S, j \notin S} y(i, j) \geq 1, \quad \forall S \subset V \text{ separating } \left\{ \bigcup_{c \in \mathcal{C}} \text{Proc}(c) \cup v_0 \right\}, \quad (23.22)$$

$$y(i, j), z(j, c) \in \{0, 1\}, \quad (23.23)$$

where s.t. means “such that”. Hence, the optimal solutions to (23.19) and (23.20) are the same.



**Figure 23.7** Cut edges of a set  $S$  separating the set of processors and fusion center.



**Figure 23.8** Minimum spanning tree is energy optimal for detection of i.i.d. data.

### 23.4.3 Special Case: i.i.d. Measurements

In the special case when the measurements are i.i.d. conditioned on either hypothesis, the LLR in (23.10) is the sum of the log-likelihoods of individual sensor measurements, that is,

$$\text{LLR}(\mathbf{Y}_V) = \sum_{v \in V} \text{LLR}(Y_v), \quad \mathbf{Y}_v \stackrel{\text{i.i.d.}}{\sim} \mathcal{H}_0 \quad \text{or} \quad \mathcal{H}_1. \quad (23.24)$$

The minimum-energy routing in this case is given by the directed minimum spanning tree (DMST), with the directions toward the fusion center. See Figure 23.8. The sum function can be calculated hierarchically along DMST, starting at the leaves and ending at the fusion center. The minimum spanning tree  $\text{MST}(V)$  over a node set  $V$  is defined as the tree of minimum total length that spans all the nodes in  $V$ . The i.i.d. case, in fact, turns out to be a lower bound for the cost of optimal fusion in a general Markov random field.

**Lemma 23.1 Lower Bound on  $C(G_f^*)$**  *The total routing cost for optimal fusion in (23.19) is no less than that of the minimum spanning tree (MST), based on the routing cost function, that is,*

$$C(\text{MST}(V)) \leq C(G_f^*(V)). \quad (23.25)$$

### 23.4.4 Minimum Spanning-Tree-Based Heuristic

We first propose a simple heuristic (AggMST), based on the minimum spanning tree. Here, we separate the design of processor selection and aggregation tree. We arbitrarily assign a clique member as the clique processor and then exploit the fact that it is feasible to compute the sum of the potentials along the MST. Of course, here only the processors have useful information in the form of potential functions, and the other nodes just forward the aggregated information. This heuristic is simple to implement since there are efficient distributed algorithms for finding the MST [36, 37].

We specify the AggMST scheme in Figure 23.9. For a clique  $c$ , the processor is assigned arbitrarily to the clique member with the lowest index (line 3). Other suitable factors such as residual energy can instead be used for the assignment. The shortest path routes from other members of  $c$  to the processor are added to the forwarding subgraph FG (line 5), and the raw data is routed along these links to enable the computation of the clique potentials. Note that the construction of the FG can be implemented in a localized manner whenever the dependency graph is local (e.g.,  $k$  nearest-neighbor graph, disk graph). The aggregation subgraph AG is  $\text{DMST}(V)$ , the minimum spanning

**Require:**  $V = \{v_0, \dots, v_{|V|-1}\}$ ,  $v_0$ : Fusion center,  $\mathcal{C} = \{c_0, \dots, c_{|\mathcal{C}|-1}\}$ : maximal cliques of MRF,  $\text{DMST}(V)$ : Minimum spanning tree, direct toward  $v_0$

- 1:  $\text{SP}(i, j) =$  (Directed) shortest path from  $i$  to  $j$
- 2: **for**  $j \leftarrow 0, |\mathcal{C}| - 1$  **do**
- 3:    $\text{Proc}(c_j) \leftarrow \min_{v_i \in c_j} v_i$   $\triangleright$  Arbitrary processor assignment
- 4:   **if**  $|c_j| > 1$  **then**
- 5:     Add  $\overline{C}(c_j \setminus \text{Proc}(c_j), \text{Proc}(c_j))$  to FG
- 6:     **end if**
- 7: **end for**
- 8:  $\text{AG} \leftarrow \text{DMST}(V)$ ,  $\Gamma \leftarrow \{\text{Proc}, \text{FG}, \text{AG}\}$
- 9: **return**  $\Gamma$

**Figure 23.9** Heuristic for aggregation in a Markov random field (AggMST).

tree, directed toward the fusion center (line 9) and potentials are added hierarchically along AG.

We now quantify the performance of the AggMST scheme for a special scenario that allows us to utilize the lower bound in Lemma 23.1.

**Theorem 23.1 Approximation** *For the case when the routing costs are Euclidean and the dependency graph is a subgraph of the Euclidean MST, the AggMST scheme has an approximation ratio of 2.*

*Proof* The MST in the lower bound (Lemma 23.1) is Euclidean since the transmission costs are Euclidean. Since the dependency graph is a subgraph of the Euclidean MST, all the links in AggMST are contained in the Euclidean MST. Hence, we have the approximation ratio of 2. To show that the bound is tight, we note that the case of extended equilateral triangles on the Euclidean plane achieves this bound.

### 23.4.5 Overview of Steiner Tree

In this section, we briefly define the Steiner tree and study its properties. These will be employed to describe our results in the subsequent sections. The material in this section is mainly from [38]. We first define the *Steiner minimal tree* [32, p. 148] below.

**Definition 23.2 Steiner Tree** *Let  $G$  be an undirected graph with nonnegative edge weights. Given a set  $L \subset V$  of terminals, a Steiner tree (ST) is the tree  $T \subset G$  of minimum total edge weight such that  $T$  includes all vertices in  $L$ .*

Finding the Steiner tree is NP-hard, and there has been extensive work on finding approximation algorithms. A 0–1 integer program to find the Steiner tree can be written as

$$\min_y \quad \frac{1}{2} \sum_{i,j \in V} y(i, j) \overline{C}(i, j), \quad (23.26)$$

$$\text{s.t.} \quad \sum_{i \in S, j \notin S} y(i, j) \geq 1, \forall S \subset V \text{ separating } L, y(i, j) \in \{0, 1\}, \quad (23.27)$$

where we say that  $A$  separates  $B$  if  $A \cap B \neq \emptyset$  and  $A \cup B = V$ . This condition ensures that all the terminals are connected, as illustrated in Figure 23.7.

**Require:**  $V = \{v_0, \dots, v_{|V|-1}\}$ ,  $v_0$ : Fusion center,  $\mathcal{C} = \{c_0, \dots, c_{|\mathcal{C}|-1}\}$ : maximal cliques of MRF,

- 1:  $G_c =$  Metric closure of comm. graph,  $\overline{\mathcal{C}} =$  Link costs in  $G_c$ ,
- 2:  $\text{ST}(G, \mathcal{L}) = \delta\text{-approx. Steiner tree on } G$ , terminal set  $\mathcal{L}$
- 3:  $G', V_c \leftarrow \text{Map}(G_c; \overline{\mathcal{C}}, \mathcal{C})$
- 4:  $\text{DST} = \text{ST}(G', V_c \cup v_0)$  and directed toward  $v_0$
- 5:  $\Gamma \leftarrow \text{RevMap}(\text{DST}; V_c, V, \mathcal{C})$
- 6: **return**  $\Gamma$

**Figure 23.10**  $\delta$ -approximate minimum cost aggregation scheme  $\Gamma$  with processor assignment and fusion digraph via Steiner tree reduction.

A simple *MST heuristic* approximates the Steiner tree over  $G$  and terminal set  $L$  with the minimum spanning tree spanning the set  $L$ , over the metric closure of  $G$ . The MST heuristic has an approximation bound of 2 [39]. The best known approximation bound for Steiner tree on graphs is 1.55, derived in [40]. The Steiner tree can be generalized to group Steiner tree, introduced by Reich and Widmayer [41].

**Definition 23.3 Group Steiner Tree** *Let  $G$  be an undirected graph with nonnegative edge weights. Given groups of vertices  $g_i \subset V$  of terminals, a group Steiner tree is the tree  $T \subset G$  of minimum total edge weight such that  $T$  includes at least one vertex from each group  $g_i$ .*

Since the group Steiner tree is a generalization of the Steiner tree, it is also NP-hard. For a group Steiner tree, polylogarithmic (in the number of groups) approximation algorithms have been proposed [42]. A series of polynomial-time heuristics are described in [43] with worst-case ratio of  $O(|g|^\epsilon)$  for  $\epsilon > 0$ .

#### 23.4.6 Steiner Tree Reduction

In this section, we show that optimal fusion has a Steiner tree reduction. We specify the graph transformations required for such a reduction and finally obtain a valid fusion scheme with processor assignment and fusion digraph. We also show that the Steiner tree reduction is approximation factor preserving. This implies that any approximation algorithm for Steiner tree provides the same ratio for minimum cost fusion.

#### 23.4.7 Simplified Integer Program

We first note that if the processor assignment is already predetermined and not part of the routing cost optimization, then we can easily characterize the optimal solution. In practice, a predetermined processor assignment might be enforced by considering other factors such as processing capabilities or residual energies of different nodes. In this case, the forwarding subgraph is also predetermined by the shortest paths to the processors. The optimal aggregation subgraph is the Steiner tree with the set of processors and the fusion center as the terminals. This is because the sum of the potential function values at the processors is computed optimally along the Steiner tree.

We next consider a modified cost optimization problem, where we ignore the routing costs of the forwarding subgraph, incurred in transporting the raw measurements to a processor. In [44, Lemma 3], we show that the minimum cost aggregation subgraph is the group Steiner tree [41], with nodes in each clique of the MRF forming a group.

The presence of processor assignment in cost optimization in (23.20) makes the problem harder than the above versions. It influences the costs of both the forwarding and aggregation subgraphs in a fusion scheme. It is not immediately clear that there is a Steiner tree reduction for (23.20). In fact, if we directly relax the integers to  $\mathbf{y}, \mathbf{z} \geq 0$  in (23.20), the program is nonlinear. We now use the local processor assignment constraint in (23.21) to write an equivalent integer program with a linear relaxation. Let  $\mathbf{z}^*$  be the optimal solution to (23.20). We have

$$\begin{aligned} \sum_{i,j \in V} I \left[ \sum_{c:i \subset c} z^*(j, c) \geq 1 \right] \bar{C}(i, j) &= \sum_{i,j \in V} I \left[ \sum_{c:i, j \subset c} z^*(j, c) \geq 1 \right] \bar{C}(i, j), \\ &= \sum_{i,j \in V} \sum_{c:i, j \subset c} z^*(j, c) \bar{C}(i, j), \\ &= \sum_{\substack{c \in \mathcal{C} \\ |c| > 1}} \sum_{i,j \subset c} z^*(j, c) \bar{C}(i, j), \end{aligned} \quad (23.28)$$

where the first equality is from local processor assignment constraint, the second equality is due to the fact that we need to assign only one processor and that there is a unique maximal clique  $c$ , if it exists, containing both  $i$  and  $j$ . Note that if the local assignment constraint is removed, then  $j$  might be assigned as the processor to many cliques  $c$  and hence, the equality does not hold. Interchanging the sums in the last equality is possible since the terms are nonzero when there is a clique  $c$  containing both  $i$  and  $j$ , and this implies that  $|c| > 1$ . Hence, we can now write an equivalent IP for minimum cost fusion under local processor assignment:

$$\min_{\mathbf{y}, \mathbf{z}} \left[ \sum_{\substack{c \in \mathcal{C} \\ |c| > 1}} \sum_{i,j \subset c} z(j, c) \bar{C}(i, j) + \sum_{i,j \in V} y(i, j) \bar{C}(i, j) \right] \quad (\text{IP-2}), \quad (23.29)$$

subject to the same constraints (23.21)–(23.23). Upon relaxation of the integer constraints, IP-2 is a linear program.

We now show that a Steiner tree on the transformed communication graph is the optimal solution to IP-2 in (23.29). To this end, we define an operation Map( $G_c$ ) in Figure 23.11, which involves adding new virtual clique-representative nodes  $v_c$  for each nontrivial clique ( $|c| > 1$ ) and adding edges between  $v_c$  and all the members of clique  $c$  with costs,

$$\bar{C}(v_c, j) := \sum_{i \subset c} \bar{C}(i, j), \quad \forall j \subset c.$$

The above cost represents the cost incurred in the forwarding subgraph upon assigning a node  $j$  as the processor for clique  $c$ . Let the set of all added clique representative vertices be  $V'$ . Hence, IP-2 in (23.29) is now equivalent to

$$\frac{1}{2} \min_{\mathbf{y}, \mathbf{z}} \sum_{v_c \in V', j \in V} z(j, c) \bar{C}(v_c, j) + \sum_{i,j \in V} y(i, j) \bar{C}(i, j),$$

```

1: function Map( $G_c(V)$ ;  $\overline{C}, \mathcal{C}$ )
2:    $\mathcal{N}(v; G) =$  Neighborhood of  $v$  in  $G$ 
3:   Initialize  $G' \leftarrow G_c$ ,  $V_c \leftarrow 0$ ,  $n \leftarrow |V|$ 
4:   for  $j \leftarrow 0, |\mathcal{C}| - 1$  do ▷ Let  $V$  and  $\mathcal{C}$  be ordered
5:     if  $|c_j| > 1$  then
6:        $V_c \leftarrow v_{n-1+j}$ , Add new node  $v_{n-1+j}$  to  $G'$ ,
7:       for all  $v_i \subset c_j$  do
8:         Add node  $v_i$  to  $\mathcal{N}(v_{n-1+j}; G')$ 
9:          $\overline{C}(v_{n-1+j}, v_i; G') \leftarrow \sum_{v_k \subset c_j, k \neq i} \overline{C}(v_i, v_k; G_c)$ 
10:      end for
11:    else ▷ For trivial cliques
12:       $V_c \leftarrow v_i$ , for  $v_i \subset c_j$ 
13:    end if
14:   end for
15: return  $G', V_c$ 
16: end function

```

**Figure 23.11**  $\text{Map}(G_c; \overline{C}, \mathcal{C})$  adds nodes corresponding to each nontrivial clique and returns the expanded graph  $G'$  and node set representing cliques  $V_c$ .

subject to the same constraints (23.21)–(23.23). For the final step, we define the set of nodes  $V''$  to account for trivial cliques

$$V'':=\{i : i \in V, i \subset c, \text{ for some } c \in \mathcal{C}, |c|=1\}.$$

The set of clique representative nodes is  $V_c:=V' \cup V''$ , the set of newly added virtual nodes and the trivial cliques. We now write the equivalent IP, which is the Steiner tree with the set of clique representatives  $V_c$  and the fusion center  $v_0$  as the terminals,

$$\begin{aligned} \frac{1}{2} \min_{\mathbf{x}} \quad & \sum_{i,j \in V} x(i, j) \overline{C}(i, j), \\ \text{s.t.} \quad & \sum_{i \in S, j \notin S} x(i, j) \geq 1, \forall S \subset V \cup V' \text{ separating } \{V_c \cup v_0\}, \quad x(i, j) \in \{0, 1\}. \end{aligned} \tag{23.30}$$

$$(23.31)$$

The equivalence holds since in the above Steiner tree, each clique representative node  $v_c \in V'$  has to be connected to at least one clique member, and, hence, the local processor assignment constraint in (23.21) is satisfied, and the constraint (23.31), which ensures that all the terminals  $V' \cup v_0$  are connected, implies that all the processors and the fusion center are connected and, hence, the constraint in (23.22) is satisfied. Hence, the optimal solution to minimum cost routing for inference is a Steiner tree on the transformed graph  $\text{Map}(G_c)$ .

In order to obtain the fusion scheme, we need another transformation after finding the Steiner tree in (23.30) on the transformed graph  $\text{Map}(G_c)$ . We first direct the Steiner tree toward the fusion center, denoted by DST. The reverse mapping RevMap(DST) in Figure 23.12 assigns the unique immediate successor of every clique-representative node  $v_c$  in DST as the processor of the clique  $c$ . The edges from the representative nodes in DST are replaced by links in the metric closure from other

```

function RevMap( $G'; V_c, V, \mathcal{C}$ )
   $\mathcal{N}_s(v; G), \mathcal{N}_p(v; G) =$  Imm. successor, predecessor of  $v$ 
  Initialize  $G \leftarrow G', n \leftarrow |V|$ 
  for all  $v_j \in V_c$  do
    if  $j > n - 1$  then
       $k \leftarrow j - n + 1,$ 
       $\text{Proc}(c_k) \leftarrow \mathcal{N}_s(v_j; G'),$  for  $c_k \in \mathcal{C},$ 
       $V_j \leftarrow c_k \setminus \text{Proc}(c_k),$  Replace  $< v_j, \text{Proc}(c_k) >$  in  $G$  with edges  $< V_j, \text{Proc}(c_k) >$ , mark
    them
    if  $\mathcal{N}_p(v_j; G) \neq 0$  Replace  $< \mathcal{N}_p(v_j), v_j >$  in  $G$  with edges  $< \mathcal{N}_p(v_j), \text{Proc}(c_k) >$ 
    end if
  else
     $\text{Proc}(c_l) \leftarrow v_j,$  for  $v_j \subset c_l$  ▷ For trivial cliques
  end if
  end for
   $\text{FG} \leftarrow \text{Marked edges of } G, \text{AG} \leftarrow G \setminus \text{FG}$ 
   $\Gamma \leftarrow \{\text{Proc, FG, AG}\}$ 
return  $\Gamma$ 
end function

```

**Figure 23.12**  $\text{RevMap}(G; V_c, V, \mathcal{C})$  maps tree  $G'$  to fusion scheme  $\Gamma$  with processor assignment Proc, forwarding and aggregation subgraphs FG, AG.

clique members to the processor and added to the forwarding subgraph of the fusion scheme. All other edges, not belonging to representative nodes in DST, are assigned as the aggregation subgraph.

In the above discussion, we have shown that the optimal solution is a Steiner tree involving transformations Map and RevMap, summarized in Figure 23.10. We now prove in addition that the above Steiner tree reduction is approximation factor preserving. To this end, we state the conditions under which the reduction preserves the approximation ratio [38, A.3.1].

**Definition 23.4 Approximation-Factor Preserving Reduction** *Let  $\Pi_1$  and  $\Pi_2$  be two minimization problems, with  $\text{OPT}_{\Pi_i}$  denoting the values of their optimal solutions. An approximation factor preserving reduction from  $\Pi_1$  to  $\Pi_2$  consists of two polynomial time algorithms,  $f$  and  $g$ , such that*

- For any instance  $I_1$  of  $\Pi_1$ ,  $I_2 = f(I_1)$  is an instance of  $\Pi_2$  such that

$$\text{OPT}_{\Pi_2}(I_2) \leq \text{OPT}_{\Pi_1}(I_1). \quad (23.32)$$

- For any solution  $t$  of  $I_2$ ,  $s = g(I_1, t)$  is a solution of  $I_1$  such that

$$\text{obj}_{\Pi_1}(I_1, s) \leq \text{obj}_{\Pi_2}(I_2, t). \quad (23.33)$$

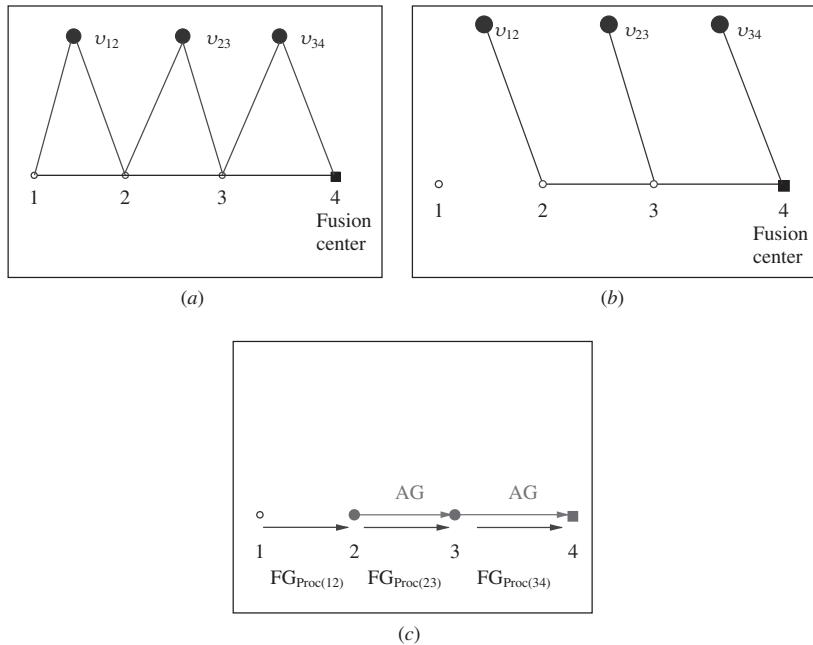
We now note that AggApprox results in a feasible fusion and runs in polynomial time since there are polynomial number of cliques. For any feasible solution to Steiner tree, replacement of links in line 9 of RevMap in Figure 23.12 reduces the sum cost, and hence, (23.33) holds.

The approximation ratio preserving Steiner tree reduction implies that any approximation algorithm for Steiner tree provides the same approximation ratio for minimum

cost fusion, when applied with the above transformations. Since currently the best known ratio for Steiner tree is 1.55, it is also the best possible approximation for minimum cost fusion for inference.

### 23.4.8 Chain Dependency Graph

We now illustrate the optimal fusion scheme through Steiner tree reduction for the simple example of a chain dependency graph in Figure 23.13, where the link communication costs and the metric closure are implicit and not shown. For this simple example, we can intuitively see that the optimal scheme first forwards raw data in the direction of fusion center. Upon computing the potential functions at the processors, the values are added along the chain, starting with the farthest processor. In Figure 23.13c, this optimal fusion scheme with forwarding and aggregation subgraphs is shown along with the values transported along the links. We now illustrate that the Steiner tree with transformations provides the same optimal solution. In Figure 23.13a, the expanded communication graph  $\text{Map}(G_c)$  is shown with added clique-representative nodes and edges. The added edges represent the costs in the forwarding subgraph on choosing a node as a processor. In Figure 23.13b, the optimal Steiner tree on the expanded graph is shown with the clique-representative nodes and the fusion center as terminals. Using RevMap, the Steiner tree is mapped to a fusion scheme by first directing the tree toward the fusion center and then assigning the immediate successor of clique-representative

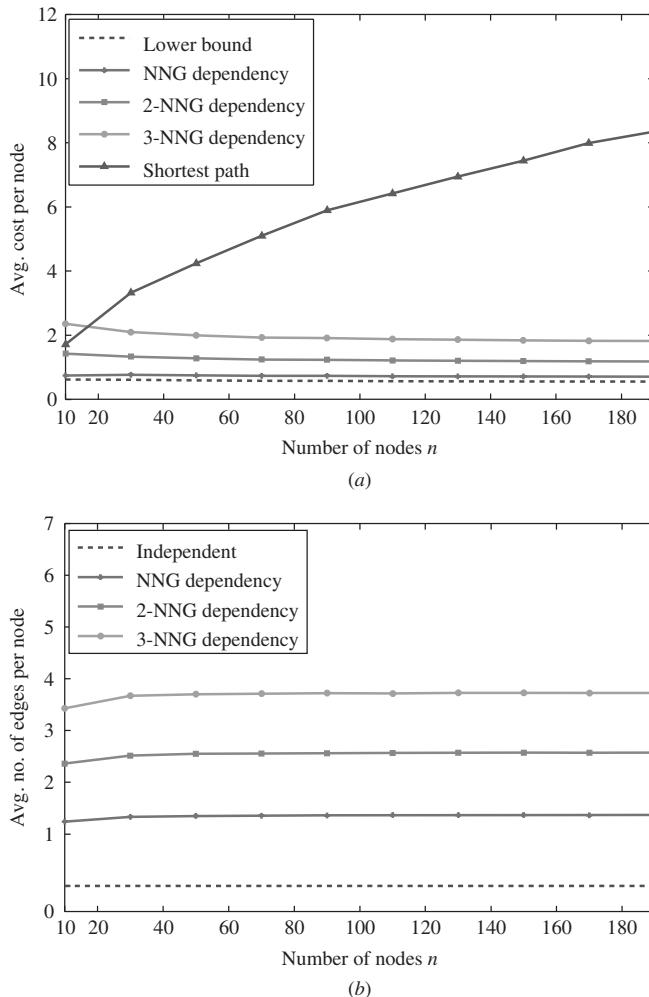


**Figure 23.13** Minimum cost fusion through Steiner reduction for chain dependency graph: (a) Expanded communication graph  $\text{Map}(G_c)$  with costs under different scenarios of processor selection. (b) Steiner tree on  $\text{Map}(G_c)$  with nodes  $v_{12}$ ,  $v_{23}$ ,  $v_{34}$  and fusion center 4 as terminals. (c) Fusion scheme with processor assignment, forwarding and aggregation subgraphs after RevMap.

nodes as processors. Hence, the member closer to the fusion center is chosen as the processor in this example. The edges from clique-representative nodes are replaced with forwarding subgraph edges, and we can see that the costs are conserved. The remaining edges in the Steiner tree form the aggregation subgraph. Hence, the RevMap operation provides the optimal fusion scheme shown in Figure 23.13c.

### 23.4.9 Discussion

We now plot some simulation results in Figure 23.14. We see that savings due to aggregation are considerable compared to shortest path routing for  $k$ -nearest neighbor graphs ( $k$ -NNG), at low values of  $k$ . These graphs are probably the best candidates,



**Figure 23.14** Simulation results for  $k$  nearest-neighbor dependency graphs. Uniform random placement of nodes. 500 runs. Constant density node placement. Routing cost on link  $(i, j) \propto \text{dist}(i, j)^2$ : (a) Average routing cost per node and (b) average number of edge potentials calculated.

after the independent data case, for in-network processing of the likelihood function. We also observe that there is direct correspondence between the number of cliques and the routing cost for fusion. Hence, it appears that the number of cliques is a good measure for judging the effectiveness of in-network processing. The gap between the heuristics and the lower bound represents the overhead arising due to correlation. A dense dependency graph has high routing costs due to the complexity of its likelihood function. This is unlike the case of compression with the aim of routing all the raw data to a destination, where a dense dependency graph (more correlation) implies redundancy and hence reduction in routing costs.

The use of localized processing constraint and unicast mode of communication are crucial to obtaining the above Steiner tree reduction. They lead to the separation of costs of routing raw measurements (in the forwarding subgraph) to compute different potential functions. On the other hand, in the absence of these constraints, the edge costs in the forwarding graph are no longer independent, and finding the optimal scheme requires the use of hyperedges. However, once a scheme is designed under the unicast setup, the broadcast nature of the wireless medium could be exploited to further reduce costs by broadcasting raw data from each node to all its processors.

We have so far considered minimum cost routing for optimal inference. A relaxation of this problem is where we only select a subset of measurements for routing and fusion, and we aim to achieve optimal trade-off between routing costs and end detection performance. This problem requires first the characterization of the detection performance, and one possibility is to use the detection error exponent, which is the asymptotic rate of exponential decay of error probability. It will be interesting to explore if this problem has reduction to well-known optimization problems, as it turned out in the case of optimal inference with local processing.

### 23.5 CONCLUSION AND FUTURE WORK

In this chapter, we have presented an instance of cross-layer design where information from the application layer is used to reduce the routing costs for a statistical inference application. Our approach combines the rich fields of statistical inference, graphical models, and approximation algorithms for network design. The joint study of these rich fields has so far been only sparsely explored, and this chapter is an effort in this direction. We exploit the data reduction in the sufficient statistic from the statistical inference literature to reduce the routing costs and formulate the minimum cost fusion scheme that computes and delivers the likelihood function to the fusion center. We employ the Markov random field model for spatial correlation and obtain the likelihood function as the sum of potential functions over the cliques of the MRF from the famous Hammersley–Clifford theorem. This structure of the likelihood function enables a two-stage in-network processing and delivery scheme. In the first stage, a processor is selected for each clique, which locally computes the potential function from the raw data. In the second stage, the values at the processors are summed up and delivered to the fusion center. We employ the machinery of approximation algorithms to prove a Steiner tree reduction, enabling us to use any Steiner tree approximation algorithm for minimum cost fusion. Our simulations show a significant saving in cost due to in-network processing compared to routing all the data to the fusion center for proximity-based sparse dependency graph models. Our results demonstrate that

inference in sensors networks can no longer be treated as well-separated problems in signal processing and networking.

### 23.6 BIBLIOGRAPHIC NOTES

Markov random fields, also known as conditional autoregressions (CAR), were introduced by Besag [14, 15]. Detailed exposition on the MRF can be found in [45–47]. For use of the MRF model in sensor networks, see [18], which deals with belief propagation (BP), also known as the sum-product algorithm. It has been applied to sensor network applications such as multitarget multisensor data association and to self-localization in [48] and [49]. However, the goal of belief propagation is to find the marginal pdf or the maximum-posterior-marginal (MPM) estimator locally, in contrast to our goal of obtaining an optimal global decision at the fusion center. In [50], a dynamic programming approach to resource management for object tracking is proposed. However, the possibility of data fusion, enroute, is not considered. In [51, 52], a decision-theoretic approach to local inference with single-bit communication is considered, and the network topology is predefined by a directed acyclic graph. Another local inference problem is consensus propagation [53], which is an asynchronous distributed protocol for averaging numbers across a network and has been applied to sensor networks in [54, 55]. The requirement of every node knowing the final value of the function is imposed.

It is beyond the scope of this chapter to provide an extensive review of the works on routing. An overview of routing for mobile wireless networks can be found in a number of surveys [56]. Correlated data gathering has been considered in [57–61]. But these schemes focus on compression, with the aim of routing all the measurements to a designated sink. Efficient aggregation schemes have been studied in [21, 62–65], but without taking into account the spatial correlation among the measurements. For example, in [65], it is assumed that multiple incoming packets at a node can be processed to a single outgoing packet; this holds only for some special functions such as sum, maximum and the like. A survey of in-network processing of various functions may be found in [66, 67]. In [68], a link metric for detection is proposed based on the model of one-dimensional Gauss–Markov random process.

### REFERENCES

1. D. P. Bertsekas and R. Gallager, *Data Networks*, Englewood Cliffs, NJ: Prentice Hall, 1992.
2. R. Koetter and M. Medard, “An algebraic approach to network coding,” *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, 2003.
3. H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer, 1994.
4. Y. Sung, S. Misra, L. Tong, and A. Emphremides, “Cooperative routing for signal detection in large sensor networks,” *IEEE J. Sel. Area Commun.*, vol. 25, no. 2, pp. 471–483, 2007.
5. Y. Sung, L. Tong, and H. Poor, “Neyman-Pearson detection of Gauss-Markov signals in noise: Closed-form error exponent and properties,” *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1354–1365, Apr. 2006.
6. A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong, “Model-driven data acquisition in sensor networks,” in *VLDB*, 2004.

7. R. Cristescu and M. Vetterli, "On the optimal density for real-time data gathering of spatio-temporal processes in sensor networks," in *IPSN*, 2005, pp. 159–164.
8. D. Marco, E. Duarte-Melo, M. Liu, and D. Neuhoff, "On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data," in *Proc. IPSN*, 2003, pp. 1–16.
9. S. Yoon and C. Shahabi, "The clustered aggregation technique leveraging spatial and temporal correlations in wireless sensor networks," *ACM Trans. Sensor Networks*, vol. 3, no. 1, 2007.
10. J. Faruque and A. Helmy, "RUGGED: RoUting on finGerprint Gradients in sEnsor Networks," in *IEEE/ACS Intl. Conf. on Pervasive Services (ICPS)*, 2004, pp. 179–188.
11. S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, 2004, pp. 28–35.
12. D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann, "An evaluation of multiresolution search and storage in resource-constrained sensor networks," in *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2003.
13. A. Jindal and K. Psounis, "Modeling spatially correlated data in sensor networks," *ACM Trans. Sensor Networks*, vol. 2, no. 4, pp. 466–499, 2006.
14. J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. Roy. Stat. Soc.*, vol. 36, no. B, pp. 192–225, 1974.
15. J. Besag, "Statistical analysis of non-lattice data," *Statistician*, vol. 24, no. 3, pp. 179–195, 1975.
16. J. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," unpublished manuscript, 1971.
17. P. Clifford, "Markov random fields in statistics," *Disord. Phys. Syst.*, pp. 19–32, 1990.
18. M. Cetin, L. Chen, J. Fisher, A. Ihler, O. Kreidl, R. Moses, M. Wainwright, J. Williams, and A. Willsky, "Graphical models and fusion in sensor networks," in *Wireless Sensor Networks: Signal Processing & Comm. Perspectives*, Hoboken, NJ: Wiley, 2007, pp. 215–250.
19. S. Li, *Markov Random Field Modeling in Computer Vision*, London: Springer-Verlag, 1995.
20. N. Cressie, *Statistics for Spatial Data*, New York: Wiley, 1993.
21. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. W. Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.
22. T. Kwon and M. Gerla, "Clustering with power control," in *Military Communications Conference Proceedings, IEEE*, Vol. 2, 1999.
23. C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Trans. Inform. Theory*, vol. 14, no. 3, pp. 462–467, 1968.
24. S. Sanghavi, V. Tan, and A. Willsky, "Learning graphical models for hypothesis testing," in *IEEE Workshop on Stat. Signal Proc.*, 2007.
25. S. Lauritzen, *Graphical Models*. Clarendon, 1996.
26. K. Jung and D. Shah, "Approximate message-passing inference algorithm," in *IEEE ITW*, 2007, pp. 224–229.
27. J. Pearl, *Probabilistic Reasoning in Intelligent Systems—Networks of Plausible Inference*. Morgan Kaufmann, 1988.
28. D. Eppstein, "All maximal independent sets and dynamic dominance for sparse graphs," in *Proc. of ACM-SIAM Symp. on Discrete Algorithms*, 2005, pp. 451–459.
29. N. Cressie and S. Lele, "New models for Markov random fields," *J. Appl. Prob.*, vol. 29, no. 4, pp. 877–884, 1992.
30. R. Cowell, *Probabilistic Networks and Expert Systems*, Springer, 1999.

31. E. Dynkin, "Necessary and sufficient statistics for a family of probability distributions," *Trans. Math. Stat. Prob.*, vol. 1, pp. 23–41, 1961.
32. B. Wu and K. Chao, *Spanning Trees and Optimization Problems*, Chapman & Hall, 2004.
33. W. P. Tay, J. N. Tsitsiklis, and M. Z. Win, "Data fusion trees for detection: Does architecture matter?" *IEEE Trans. Inform. Theory*, 2007, submitted for publication.
34. W. P. Tay, J. N. Tsitsiklis, and M. Z. Win, "On the sub-exponential decay of detection error probabilities in long tandems," *IEEE Trans. Inform. Theory*, 2007, submitted for publication.
35. L. Devroye, "The expected size of some graphs in computational geometry." *Comp. Math. App.*, vol. 15, no. 1, pp. 53–64, 1988.
36. R. Gallager, P. Humblet, and P. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Trans. Prog. Lang. Syst. (TOPLAS)*, vol. 5, no. 1, pp. 66–77, 1983.
37. P. Humblet, "A distributed algorithm for minimum weight directed spanning trees," *IEEE Trans. Commun.*, vol. 31, no. 6, pp. 756–762, 1983.
38. V. Vazirani, *Approximation Algorithms*, Springer, 2001.
39. L. Kou, G. Markowsky, and L. Berman, "A fast approximation algorithm for Steiner trees," *Acta Informatica*, vol. 15, pp. 141–145, 1981.
40. G. Robins and A. Zelikovsky, "Improved Steiner tree approximation in graphs," in *Proc. of ACM-SIAM Symposium on Discrete Algorithms*, 2000, pp. 770–779.
41. G. Reich and P. Widmayer, "Beyond Steiner's problem: A vlsi oriented generalization," in *Proc. of Intl. Workshop on Graph-Theoretic Concepts in Computer Science*, 1990, pp. 196–210.
42. N. Garg, G. Konjevod, and R. Ravi, "A polylogarithmic approximation algorithm for the group Steiner tree problem," *J. Algorithms*, vol. 37, no. 1, pp. 66–84, 2000.
43. C. S. Helvig, G. Robins, and A. Zelikovsky, "Improved approximation bounds for the group Steiner problem," in *DATE '98: Proceedings of the Conference on Design, Automation and Test in Europe*, 1998, pp. 406–413.
44. A. Anandkumar, L. Tong, A. Swami, and A. Ephremides, "Minimum cost data aggregation with localized processing for statistical inference," in *Proc. of IEEE INFOCOM*, Phoenix, AZ, Apr. 2008, pp. 780–788.
45. P. Brémaud, *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*, Springer, 1999.
46. H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*, London: Chapman and Hall, 2005.
47. X. Guyon, *Random Fields on a Network: Modeling, Statistics, and Applications*, Springer, 1995.
48. L. Chen, M. Wainwright, M. Cetin, and A. Willsky, "Multitarget-multisensor data association using the tree-reweighted max-product algorithm," *Proc. SPIE*, vol. 5096, 2003, pp. 127–138.
49. A. Ihler, J. Fisher III, R. Moses, and A. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 809–819, 2005.
50. J. Williams, J. Fisher III, and A. Willsky, "An approximate dynamic programming approach to a communication constrained sensor management problem," in *Intl. Conf. on Information Fusion*, Vol. 1, 2005.
51. O. Kreidl and A. Willsky, "Inference with minimal communication: A decision-theoretic variational approach," in *Advances in Neural Information Processing Systems*, 2006.
52. O. Kreidl and A. Willsky, "Efficient message-passing algorithms for optimizing decentralized detection networks," in *IEEE Conf. on Decision and Control*, 2006.

53. C. Moallemi and B. Van Roy, "Consensus propagation," vol. 52, no. 11, pp. 4753–4766, 2006.
54. A. Dimakis, A. Sarwate, and M. Wainwright, "Geographic gossip: Efficient aggregation for sensor networks," in *Proceedings of the Fifth International Conference on Information Processing in Sensor Networks*, 2006, pp. 69–76.
55. R. Olfati-Saber, E. Franco, E. Frazzoli, and J. Shamma, "Belief consensus and distributed hypothesis testing in sensor networks," in paper presented at the Workshop on Network Embedded Sensing and Control, Notre Dame University, Oct, 2005.
56. K. Akkaya and M. Younis, "A survey of routing protocols in wireless sensor networks," *Elsevier Adhoc Networks*, vol. 3, pp. 325–349, 2005.
57. R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer, "Network correlated data gathering with explicit communication: NP-completeness and algorithms," *IEEE/ACM Trans. Networking (TON)*, vol. 14, no. 1, pp. 41–54, 2006.
58. P. von Rickenbach and R. Wattenhofer, "Gathering correlated data in sensor networks," paper presented at the Joint Workshop on Foundations of Mobile Computing, 2004, pp. 60–66.
59. A. Goel and D. Estrin, "Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk," *Algorithmica*, vol. 43, no. 1, pp. 5–15, 2005.
60. H. Gupta, V. Navda, S. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," in *Proc. of ACM Intl. Symposium on Mobile Ad Hoc Networking and Computing*, 2005, pp. 402–413.
61. A. Scaglione and S. Servetto, "On the interdependence of routing and data compression in multi-hop sensor networks," in *Proc. MobiCom 2002*, Atlanta, GA, Sept. 2002.
62. S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TinyDB: An acquisitional query processing system for sensor networks," *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 122–173, 2005.
63. J. Gehrke and S. Madden, "Query processing in sensor networks," *IEEE Pervasive Comput.*, vol. 3, no. 1, pp. 46–55, 2004.
64. C. Intanagonwiwat, R. Govindan, and D. Esterin, "Directed diffusion: A scalable and robust paradigm for sensor networks," in *Proc. 6th ACM/Mobicom Conference*, Boston, MA, 2000, pp. 56–67.
65. B. Krishnamachari, D. Estrin, and S. Wicker, "Modeling data centric routing in wireless sensor networks," in *INFOCOM*, New York, 2002.
66. A. Giridhar and P. Kumar, "Toward a theory of in-network computation in wireless sensor networks," *Commun. Mag. IEEE*, vol. 44, no. 4, pp. 98–107, 2006.
67. R. Rajagopalan and P. Varshney, "Data aggregation techniques in sensor networks: A survey," *Commun. Surveys Tutorials IEEE*, vol. 8, no. 4, pp. 48–63, 2006.
68. S. Misra, L. Tong, and A. Ephremides, "Application dependent shortest path routing in ad-hoc sensor networks," in *Wireless Sensor Networks: Signal Processing & Comm. Perspectives*, Hoboken, NJ: Wiley, 2007, pp. 277–310.

---

## CHAPTER 24

---

# Spectral Estimation in Cognitive Radios

Behrouz Farhang-Boroujeny

ECE Department, University of Utah, Salt Lake City

The demand for ubiquitous wireless services has been on the rise in the past and is expected to remain the same in the future. As a result, the vast majority of the available spectral resources have already been licensed. It thus appears that there is little or no room to add any new services, unless some of the existing licenses are discontinued. On the other hand, studies have shown that vast portions of the licensed spectra are rarely used [1, 2]. This has initiated the idea of cognitive radio (CR), where secondary (i.e., unlicensed) users are allowed to transmit and receive data over portions of spectra when primary (i.e., licensed) users are inactive. This is done in a way that the secondary users (SUs) are transparent to the primary users (PUs). For this, SUs need to sense the spectrum, and this involves some sort of spectral analysis.

The term *spectral analysis* refers to any signal processing method that may be used to estimate the power distribution, known as power spectral density (PSD), of a signal across the frequency axis. Spectral estimation methods can be broadly divided in two classes: parametric and nonparametric [3]. In parametric spectral estimation, the input process is modeled as a cascade of a white random process and a linear time invariant system that is characterized by a transfer function with a predetermined order. The observed samples of the input are used to estimate the parameters of the model and accordingly obtain an estimate of the PSD. This method, although it works well when a correct model of the input is selected, may be inappropriate in cases where an accurate model of the input is unknown. This is the case in CR scenarios where user activities are highly random and in many cases unpredictable. Moreover, the complexity of parametric spectral estimators is often a concern. Nonparametric spectral estimation, on the other hand, does not involve any signal model and usually involves moderate or low computational complexity. Because of these advantages, it has been generally accepted that nonparametric spectral estimation methods should be used for spectrum sensing in CR systems [4].

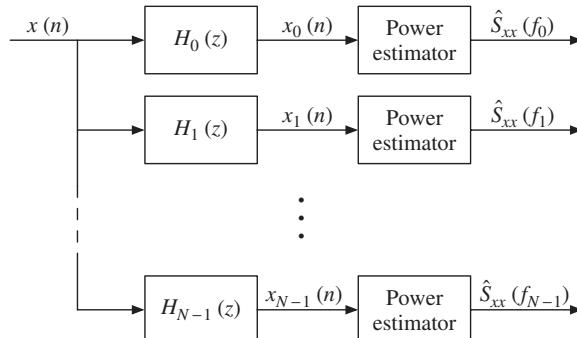
In this chapter, the problem of nonparametric spectral estimation will be cast in a filter bank formulation frame work. This approach allows us to present a unified study of a wide class of spectral estimation methods. The filter bank formulation suggests that the spectrum of a signal  $x(n)$  may be obtained by partitioning the signal into a

number of narrowband signals and measuring the power of each narrowband signal as an estimate of the signal spectrum over the respective band. The partitioning of the signal is done through a filter bank. This simple point of view turns out to be what the widely used *periodogram* spectral estimator (PSE) [3], as well as the more advanced *multitaper* spectral estimator of Thompson [5], do. More recently, Farhang-Boroujeny [6] has shown that filter banks that may be used for multicarrier communication in a cognitive radio setting may also be used to perform the task of spectral estimation. In that case, the task of spectrum sensing comes at virtually no additional cost.

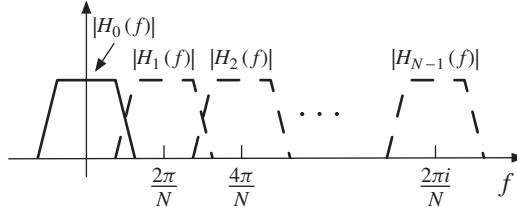
It is worth noting that beside the above methods, there exist other nonparametric spectral estimation methods that do not follow the filter bank formulation, for example, the Blackman–Tukey method and the minimum variance spectral estimation method [3]. On the other hand, parametric spectral estimation methods are often found useful in identifying line spectra within a wideband spectrum. Other methods such as multiple signal classification (MUSIC) have been proposed and also found very helpful for finding line spectra [7]. It is believed that in the application of cognitive radios, one's interest is more on looking at narrowbands of the spectrum and identify the presence or absence of the signal activities (the primary users) in each band, without much care about a more microscopic details (e.g., the presence of line spectra within a band). On this basis, the scope of this chapter is limited to the spectral estimators that fall within the class of filter-bank-based methods. Moreover, we note that in the application of interoperable (cognitive) radios, there exists also interest to look into the details of signal activity within each band, for example, to identify the modulation type [8, 9]. However, this does not fall within the area of spectral estimation methods and, hence, is beyond the scope of this chapter.

## 24.1 FILTER BANK FORMULATION OF SPECTRAL ESTIMATORS

Figure 24.1 presents a block diagram of a filter-bank-based spectral estimator. The filters  $H_0(z), H_1(z), \dots, H_{N-1}(z)$  make a bank of filters that decompose the input signal  $x(n)$  into  $N$  narrowband signals  $x_0(n), x_1(n), \dots, x_{N-1}(n)$ ; see Figure 24.2. For  $i = 0, 1, \dots, N - 1$ , the respective power estimator takes a time average of  $|x_i(n)|^2$  as an estimate of the signal power over the  $i$ th band of the filter bank. Normalizing the power estimates with respect to the width of the bands results in estimates of the PSD



**Figure 24.1** Block diagram of spectral estimators based on filter banks.



**Figure 24.2** Graphical presentation of a filter bank.

of  $x(n)$ ,  $S_{xx}(f)$ . In Figure 24.1, the frequency  $f_i$  is the center frequency of the  $i$ th band, and  $\hat{S}_{xx}(f_i)$  denotes the estimate of  $S_{xx}(f)$  at  $f = f_i$ .

Figure 24.2 presents a set of magnitude responses of the filters  $H_0(z)$ ,  $H_1(z)$ , ...,  $H_{N-1}(z)$ . We note that in this figure we have used the shorthand notation  $H_i(f)$  to represent  $H_i(e^{j2\pi f})$ . The same notation is used throughout this chapter for other functions as well, for example,  $\hat{S}_{xx}(f)$  for  $\hat{S}_{xx}(e^{j2\pi f})$  in Figure 24.1.

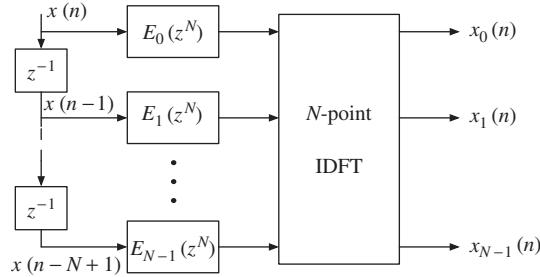
One may notice that in Figure 24.2,  $f_i = 2\pi i/N$  and  $H_i(f) = H_0(f - f_i)$ . In other words, the zeroth-band filter  $H_0(f)$  may be used to generate the rest of the filters in the filter bank. This is a special case of filter banks that is called *uniform* discrete Fourier transform (DFT) *filter bank*. The adjective DFT refers to the fact that the set of filters  $H_i(z)$ 's can be implemented jointly in an efficient DFT-based structure, called *polyphase*. The polyphase realization of a uniform DFT filter bank is discussed in the next section. Also, since the whole filter bank is implemented based on the filter  $H_0(z)$ ,  $H_0(z)$  is called the *prototype filter*. Also, to simplify notations, we drop the subscript 0 from  $H_0(z)$ , that is, we use  $H(z)$  to refer to the prototype filter.

The number of bands,  $N$ , in the filter bank determines the number of samples of  $S_{xx}(f)$  that the underlying spectral estimator obtains. It thus determines the *frequency resolution* (i.e., quantization along the frequency axis) of the spectral estimator. It is also important to note that each estimate of  $S_{xx}(f)$  is obtained by averaging the signal energy at the output of the respective filter. The variance of the estimates can be controlled and is reduced as the number of samples used to obtain each power estimate increases. Moreover, since in practice  $H(z)$  is a nonideal filter with a nonzero stopband gain, there will be always some signal *leakage* among different bands within a filter bank. Such leakage results in some *bias* in the spectral estimates. In particular, when estimating  $S_{xx}(f)$  over a low-power band of  $x(n)$ , leakage from high-power bands of  $x(n)$  may introduce a significant bias in the desired estimate. Clearly, to reduce the leakage, and thus increase the spectrum analyzer ability to distinguish between low and high power bands, one should use a prototype filter with a stopband gain that is as small as possible. In this chapter, we use the terminology *spectral dynamic range* to quantify the ratio of maximum and minimum spectral powers that are distinguishable in a spectrum analyzer.

## 24.2 POLYPHASE REALIZATION OF UNIFORM FILTER BANKS

Let

$$H(z) = \sum_{k=0}^{M-1} h(k)z^{-k} \quad (24.1)$$



**Figure 24.3** Polyphase realization of an  $N$ -band filter bank.

be the prototype filter of an  $N$ -band uniform filter bank. The  $i$ th band of the filter bank has the center frequency  $2\pi i/N$  and, thus, its transfer function is obtained by replacing  $z$  in (24.1) by  $zW_N^i$ , where  $W_N = e^{-j2\pi/N}$ . This leads to the transfer function

$$H_i(z) = \sum_{k=0}^{M-1} h(k) W_N^{-ik} z^{-k}. \quad (24.2)$$

If we let  $M = KN$ , (24.2) can be rearranged as

$$H_i(z) = \sum_{l=0}^{N-1} z^{-l} E_l(z^N) W_N^{-il}, \quad (24.3)$$

where

$$E_l(z) = h(l) + h(l+N)z^{-1} + \cdots + h(l+(K-1)N)z^{-(K-1)} \quad (24.4)$$

is the  $l$ th polyphase component of  $H(z)$ .

Direct application of (24.3) leads to the polyphase structure shown in Figure 24.3 [10]. Here, the  $N$ -band filters of the filter bank share the same structure (hardware or software). The computational complexity of the structure is equivalent to that of realization of the prototype filter and one inverse discrete Fourier transform (IDFT) of size  $N$ . Moreover, when  $N$  is a proper composite number, the IDFT can be realized efficiently using the fast Fourier transform (FFT) technique.

### 24.3 PERIODOGRAM SPECTRAL ESTIMATOR

The periodogram spectral estimator (PSE) is the most basic and simplest member of the class of nonparametric spectral estimators. It obtains an estimate of the spectrum  $S_{xx}(f)$  of a random process  $x(n)$ , based on  $N$  samples of one realization of it, as [3]

$$\hat{S}_{\text{PSE}}(f_i) = \left| \sum_{k=0}^{N-1} h_i(k) x(n-k) \right|^2, \quad (24.5)$$

where  $\{x(n-k), k = 0, 1, \dots, N-1\}$  is the sample set,  $h_i(n) = w(n)e^{j2\pi f_i n}$ , and  $w(n)$  is a window function. Clearly, if  $w(n)$ 's are chosen such that they are the

coefficients of a finite impulse response (FIR) low-pass filter,  $h_i(n)$  will be a bandpass filter with the center frequency  $f_i$ . Also, if we choose  $f_i = i/N$ ,  $i = 0, 1, \dots, N - 1$ , the filters  $h_i(n)$  define a filter bank with the prototype filter  $h(n) = w(n)$  and, hence, the scalar polyphase components  $E_l(z) = w(l)$ .

### 24.3.1 Common Window Functions

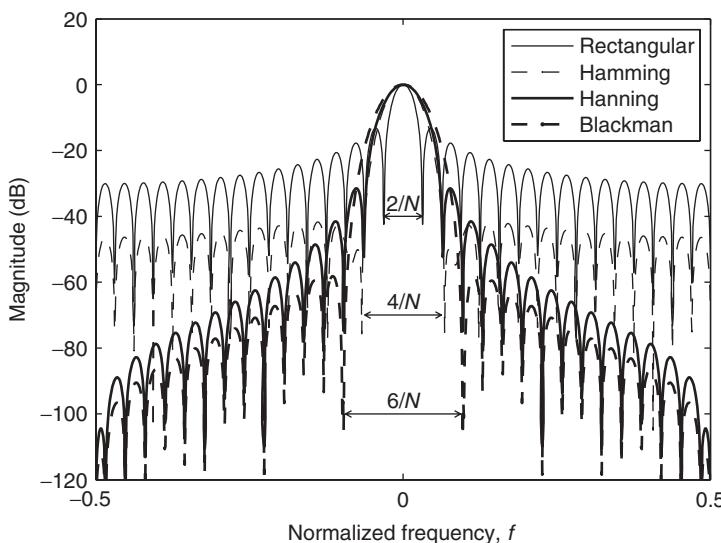
In its simplest form  $w(n) = 1/\sqrt{N}$ , for  $n = 0, 1, \dots, N - 1$ . This is a rectangular window that is characterized with a sinc magnitude response. The sinc pulse is not desirable in the application of interest here since its relatively large side lobes result in significant spectral leakage and, therefore, a limited spectral dynamic range. By replacing the rectangular window with a window function that smoothly decays on both sides, a prototype filter with much smaller side lobes is obtained [3]. There exist a wide range of window functions from which one may choose. Among them Hamming, Hanning, and Blackman are the most popular and widely used window functions. They are, respectively, defined as

$$\text{Hamming: } w(n) = 0.54 - 0.46 \cos \frac{2\pi n}{N-1}, \quad (24.6)$$

$$\text{Hanning: } w(n) = 0.5 - 0.5 \cos \frac{2\pi n}{N-1}, \quad (24.7)$$

$$\text{Blackman: } w(n) = 0.42 - 0.5 \cos \frac{2\pi n}{N-1} + 0.08 \cos \frac{4\pi n}{N-1}. \quad (24.8)$$

The magnitude of frequency responses of these window functions, for  $N = 31$ , along with that of rectangular window, are presented in Figure 24.4. Comparing the responses, we find that (i) the rectangular window has the narrowest main lobe (equal to  $2/N$ ) while its side lobes are largest in magnitude; (ii) Hamming and Hanning windows



**Figure 24.4** Frequency responses of various window functions for  $N = 31$ .

achieve much lower side lobes at the cost of a wider main lobe (equal to  $4/N$ ); and (iii) Blackman window further improves the side lobes at the cost of further expansion of the width of the main lobe (equal to  $6/N$ ).

The above window functions are very limited in controlling the width of the main lobe and the size of the side lobes of the frequency response. The size of the side lobes and the width of the main lobe are determined by the window type, and once window type is selected, one can only control the width of the main lobe by changing the window length  $N$ . Also, it is not clear whether any of these window functions is optimal in any sense. Next, we introduce and study a class of window functions that provides a great degree of flexibility and the designed windows satisfy a certain optimality feature.

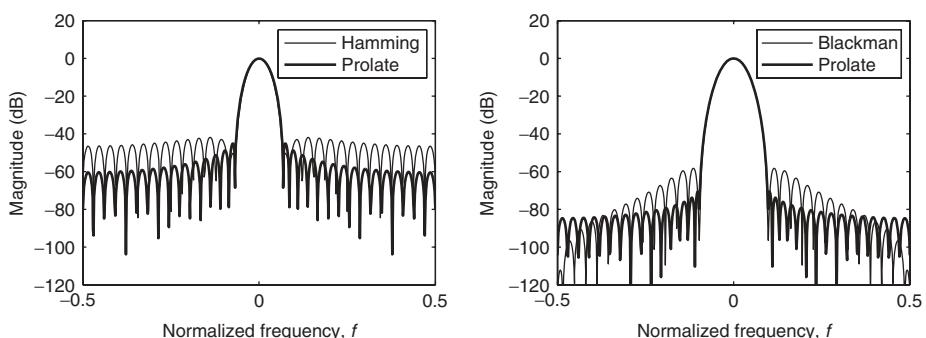
### 24.3.2 Prolate Sequences: Class of Optimal Window Functions

The process of choosing a window  $w(n)$  with a target main lobe width  $\Delta f$  and minimizing the side lobes may be formulated as the following optimization problem: *Given a bandwidth  $\Delta f$ , design a low-pass FIR filter of length  $M$  whose main lobe is within the range  $(-\Delta f/2, \Delta f/2)$  and has minimum stopband energy.* The coefficients of the designed filter,  $w(n)$ , constitute a sequence that is called prolate. It may be also viewed as a window function and thus referred to as *prolate window*.

We defer the solution to the above problem to the next section where we find this to be a specialized case of the multitaper spectral estimator (MTSE). However, we note that the desired window happens to be the eigenvector associated with the maximum eigenvalue of the matrix  $\mathbf{R}$  that is defined in the next section; see (24.17) and note that for the case PSE the prototype filter length  $M = N$ . Also, to provide the reader a sense of how such optimized design compares with the window functions that were introduced above, in Figure 24.5, we have presented magnitude responses of Hamming and Blackman window functions along with those of prolate windows of the same length and the same main lobe width. As expected, the prolate designs are indeed superior.

### 24.3.3 Spectrum Smearing

The above studies reveal that the reduction in the size of side lobes, which translates to reduction of bias in the spectral estimates, is achieved at the cost of an increase in the width of the main lobe. For instance, while a rectangular window of length  $N$



**Figure 24.5** Comparison of prolate window with Hamming and Blackman window functions. Both cases are for a window length  $N = 31$ .

has a main lobe of width  $2/N$ , Hamming and Hanning windows of the same length have a main lobe of width  $4/N$ . On the other hand, widening the main lobe of a window function results in a spectral average (power estimation) over a wider band. The result of such averaging, is a smearing effect in the detected spectra—the sharp peaks as well as sharp transitions will be smoothed. Hence, the choice of different windows provides a trade-off between the ability of seeing sharp peaks and/or transitions (which may be viewed as frequency resolution) and estimation bias [3, 7].

#### 24.3.4 Spectral Averaging

Since each sample of the estimate in (24.5) is based on a single output sample of the corresponding filter, the estimates are very coarse, that is, have a very low precision. In fact, if the output sample  $x_i(n) = \sum_{k=0}^{N-1} h_i(k)x(n-k)$  is assumed to be a complex-valued Gaussian random variable, it will be found that  $|x_i(n)|^2$  has a standard deviation that is equal to its mean [3].

There are two common methods that may be used for improving the precision of the spectral estimates.

- *Averaging Across Time Axis* In this method multiple successive and possibly overlapping blocks of  $x(n)$  are processed and samples of  $|x_i(n)|^2$ , across the time axis,  $n$ , are averaged. This method is called *weighted overlapped segment averaging* (WOSA) [11].
- *Averaging Across Frequency Axis* In this method, the window length is increased and adjacent samples of  $|x_i(n)|^2$ , across the frequency axis,  $i$ , are weighted and averaged. Note that by increasing  $N$ , one increases the frequency resolution of the estimates, and averaging across frequency axis may bring us back to the same frequency resolution before increasing  $N$ . This method has been referred to a *frequency smoothing*. Also, an effective implementation of it that applies effective weights by swinging between the time and frequency domains has been proposed in [12] and is referred to as *lag weighting*. It is shown in [12] that the combination of WOSA and lag weighting results in an effective spectral estimation method.

#### 24.3.5 Numerical Example of PSE

In order to have a common basis for comparison of the various spectral estimation methods that are discussed in this chapter, here we begin with introducing the construction of a test signal. We lay down the details of this construction to allow interested readers to regenerate the same process and reproduce the same results, if they wish. We use this test signal to examine the performance of the various window functions in this section and repeat the same experiment for other methods that are presented in the subsequent sections. These numerical experiments also serve to reveal some aspects of the spectral estimation methods that otherwise would be hard to explain.

As the test signal, we construct a random process with the PSD shown in Figure 24.6. This is a multiband process with a 60 dB variation of signal power across three passband signals. There is also a white background noise at -80 dB below the peak of the spectrum. Variation of the signal power over a relatively wide dynamic range, 80 dB, will allow us to test the capabilities as well as the shortcomings of the different methods.

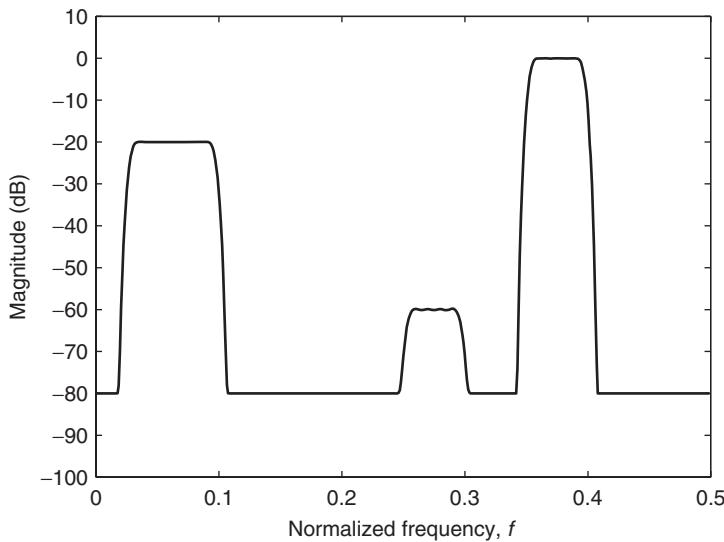


Figure 24.6 PSD of the test signal.

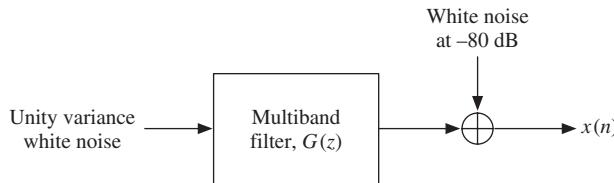


Figure 24.7 Test signal generator.

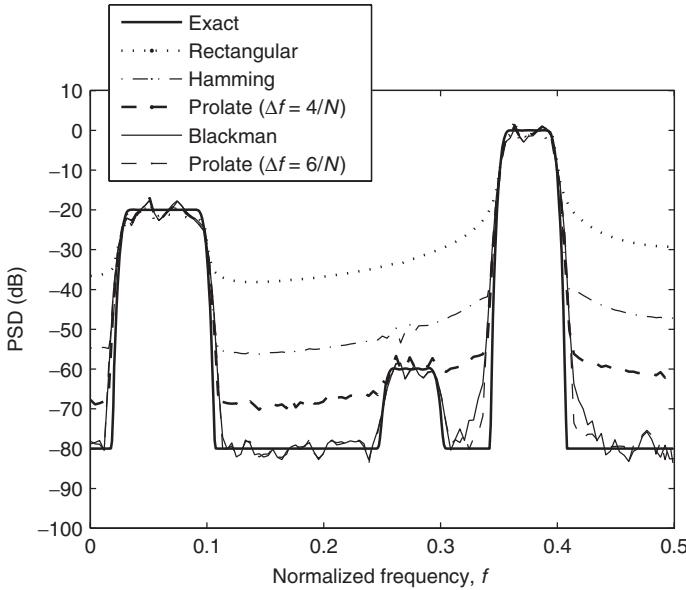
Figure 24.7 presents the block diagram that is used to generate the test signal. The multiband filter  $G(z)$  is a finite impulse response (FIR) filter that we generate in MATLAB using the following instructions:

```

band1=[0.025 0.1]*2; band2=[0.25 0.3]*2; band3=[0.35 0.4]*2;
g0=sqrt(0.1)*fir1(200,band1,'bandpass')+sqrt(0.001)*fir1(200,
    band2,'bandpass')+fir1(200,band3,'bandpass');
g=conv(g0,g0);
    
```

The first line defines the three passbands. The second line generates an FIR filter with the specified passband and the gains that are square root of the desired gain. The third line, by convolving the designed filter with itself, fixes the passband gains to the desired levels and pushes down the stopband gains to a level significantly below the background noise. To generate a test signal with the desired PSD, samples of  $x(n)$  are taken after the transient period of  $G(z)$ .

In Figure 24.8, we have presented a few snapshots of the PSE using rectangular, Hamming, Blackman, and two choices of prolate window functions. The window length is  $N = 256$ , and the spectra are averages over six nonoverlapping, consecutive windows of  $x(n)$ . The prolate windows are designed for the main-lobe widths of  $\Delta f = 4/N$  and



**Figure 24.8** Comparison of different window function in a periodogram spectral estimator.

$6/N$ . These designs have similar stopband attenuation as those shown in Figure 24.5;  $-60$  and  $-80$  dB, respectively. Also, from Figure 24.4 we observe that the rectangular window has a very poor stopband, at around  $-30$  dB, the Hamming window stopband is around  $-45$  dB, and the Blackman window has a stopband comparable to that of prolate window with  $\Delta f = 6/N$ .

From Figure 24.8, we make the following observations:

- The rectangular, Hamming, and prolate window with  $\Delta f = 4/N$  are incapable of seeing the second passband (at  $-60$  dB) in the spectrum.
- The prolate window with  $\Delta f = 6/N$  and Blackman window, on the other hand, are able to see the second passband clearly. However, the wider main lobe of these window functions results in more pronounced smearing effect at the sides of each passband. Moreover, the smearing effect in the case of the Blackman window is more observed at the two sides of the third passband. This may be attributed to the relatively larger side lobes at the beginning of the stopband of the Blackman window; see Figure 24.5.

## 24.4 MULTITAPER SPECTRAL ESTIMATOR

From our discussions so far, each band of a spectrum analyzer of Figure 24.1 is characterized by a frequency response that is determined by the underlying prototype filter. Ideally, the prototype filter should have a flat gain over its passband and should drop to zero abruptly out of this band. However, this is not possible in practice, due to the limited length of the prototype filter. The prototype filters (the window functions) that were introduced in the previous section, in particular, had a relatively short length, equal to the number of bands in the filter bank. They thus suffer either

from side lobes with relatively large amplitude or a wide main lobe. The first has the impact of introducing significant leakage from the portions of the spectrum that are far from the desired band. A wide main lobe, on the other hand, results in significant leakage from adjacent bands, hence, significant smearing of the estimated spectra.

The MTSE resolves the above problems by adopting much longer prototype filters. Longer prototype filters allow narrower transition width and reduce the size of the side lobes, hence, reduce the leakage from both adjacent bands and other portions of the spectrum. Moreover, to improve on the precision of the spectral estimates (i.e., to reduce the variance of the estimates), the MTSE uses multiple prototype filters and averages the instantaneous energy of the associated filter bank outputs [5]. The concept based on which MTSE prototype filters are constructed can be viewed as a generalization of prolate filters that were introduced above and, thus, may be explained as follows.

For a given frequency resolution  $\Delta f$ , the prototype filters are designed optimally (as discussed below) to pick signal energy from the frequency range  $(-\Delta f/2, +\Delta f/2)$ , while achieving maximum rejection of the out of band energy, thus, minimizing the size of the side lobes. Filters designed in this way, as noted earlier, are known as *prolate filters* and their coefficients form sequences that are called *discrete-time prolate spherical sequences* or *Slepian sequences* [13–16]. The Slepian sequences constitute a set of orthogonal vectors that may be used to expand the time series<sup>1</sup>  $\{x(n), x(n-1), \dots, x(n-M+1)\}$ , or, equivalently, the vector  $\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-M+1)]^T$ , over the frequency band  $(f_i - \Delta f/2, f_i + \Delta f/2)$ . Mathematically, this may be written as

$$\mathbf{x}(n) \approx \sum_{k=0}^{K-1} \kappa_k(f_i) \mathbf{D} \mathbf{q}_k, \quad (24.9)$$

where  $K < N$  is the number of prolate sequences used for the expansion,  $\kappa_k(f_i)$ 's are the expansion coefficients,  $\mathbf{q}_k$ 's, a set of orthogonal basis vectors, are the Slepian sequences,  $\mathbf{D}$  is a diagonal matrix with the diagonal elements of  $1, e^{j2\pi f_i}, \dots, e^{j2\pi(M-1)f_i}$ , and the equality sign is replaced by  $\approx$  (denoting approximately equal to) because the expansion is incomplete, since  $K < N$ . The size of  $K$  is determined by the bandwidth  $\Delta f$  and the expected spectral dynamic range that MTSE has to resolve. This will be discussed in great details as we proceed in the next few sections.

The reader may realize the similarity of (24.9) with the Fourier series expansion and note that

$$\kappa_k(f_i) = (\mathbf{D} \mathbf{q}_k)^H \mathbf{x}(n), \quad (24.10)$$

where the superscript H denotes Hermitian transpose. Accordingly, an estimate of the signal energy over the frequency band  $(f_i - \Delta f/2, f_i + \Delta f/2)$  is given by

$$\hat{S}_{xx}(f_i) = \frac{1}{K} \sum_{k=0}^{K-1} |\kappa_k(f_i)|^2. \quad (24.11)$$

Alternatively, one may view  $\kappa_k(f_i)$ 's as the outputs of a set of bandpass filters with coefficient vectors  $\mathbf{D} \mathbf{q}_k$ 's. Moreover,  $\mathbf{q}_k$ 's may be thought of as a set of prototype filters

<sup>1</sup>Here, we have chosen to write time sequences in reverse order, as it simplifies the derivations in terms of filters. This minor change in presentation has no effect on the spectral estimates.

that are used to construct a set of filter banks. The averaged output energy of the same numbered subbands is then used to obtain a spectral estimate of  $x(n)$ . This is exactly how the Thomson's multitaper (MT) method works [5].

#### 24.4.1 Derivation of Slepian Sequences

The process of finding the Slepian sequences can be cast in the following filter design problem: *Given a bandwidth  $\Delta f$ , design a set of  $K$  prototype low-pass filters whose main lobes are within the range  $(-\Delta f/2, \Delta f/2)$  and have minimum stopband energy. Moreover, the designed filters are selected so that their coefficients form a set of orthogonal vectors.*

A solution to this problem follows from the eigenvalue/eigenvector minimax theorem. We present a particular form of this theorem that best relates to the design of the Slepian sequences.

**Theorem 24.1 Minimax Theorem<sup>2</sup>** *The distinct eigenvalues  $\lambda_0 > \lambda_1 > \dots > \lambda_{M-1}$  of the correlation matrix  $\mathbf{R}$  of an observation vector  $\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-M+1)]^T$ , and their corresponding eigenvectors,  $\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_{M-1}$ , may be obtained through the following optimization procedure:*

$$\lambda_{\max} = \lambda_0 = \max_{\|\mathbf{q}_0\|=1} E[|\mathbf{q}_0^T \mathbf{x}(n)|^2] \quad (24.12)$$

where  $\|\mathbf{q}_i\| = \sqrt{\mathbf{q}_i^T \mathbf{q}_i}$  denotes the norm of the vector  $\mathbf{q}_i$ , and for  $i = 1, 2, \dots, M-1$

$$\lambda_i = \max_{\|\mathbf{q}_i\|=1} E[|\mathbf{q}_i^T \mathbf{x}(n)|^2], \quad \text{subject to } q_i^T q_j = 0, \text{ for } 0 \leq j < i. \quad (24.13)$$

Alternatively, the following procedure may also be used to obtain the eigenvalues (and the associated eigenvectors) of the correlation matrix  $\mathbf{R}$ , in the ascending order:

$$\lambda_{\min} = \lambda_{M-1} = \min_{\|\mathbf{q}_{M-1}\|=1} E[|\mathbf{q}_{M-1}^T \mathbf{x}(n)|^2], \quad (24.14)$$

and for  $i = M-2, \dots, 1, 0$

$$\lambda_i = \min_{\|\mathbf{q}_i\|=1} E[|\mathbf{q}_i^T \mathbf{x}(n)|^2], \quad \text{subject to } \mathbf{q}_i^T \mathbf{q}_j = 0, \text{ for } i < j \leq M-1. \quad (24.15)$$

A proof of the minimax theorem, in the above form, can be found in [18].

Using the filter design problem, mentioned above, and the minimax theorem, the Slepian sequences may be obtained by taking the following steps:

- Let  $x(n)$  be a random process with power spectral density

$$\Phi(f) = \begin{cases} 1, & -\Delta f/2 \leq f \leq \Delta f/2, \\ 0, & \text{otherwise.} \end{cases} \quad (24.16)$$

<sup>2</sup>In matrix algebra literature, the minimax theorem is usually stated using the Hermitian form  $\mathbf{q}_i^T \mathbf{R} \mathbf{q}_i$  (or  $\mathbf{q}_i^H \mathbf{R} \mathbf{q}_i$ , when the underlying process is complex valued), instead of  $E[|\mathbf{q}_i^T \mathbf{x}(n)|^2]$ ; see [17], for example. The above minimax formulation is from [18]. This method has been adopted from [19].

- Construct the  $M$ -by- $M$  correlation matrix  $\mathbf{R}$  of  $x(n)$ . This is the symmetric Toeplitz matrix whose first row consists of the correlation coefficients of  $x(n)$ , namely,

$$\phi(k) = \Delta f \operatorname{sinc}(\Delta f k), \quad \text{for } k = 0, 1, \dots, M - 1. \quad (24.17)$$

- The first  $K$  eigenvectors of  $\mathbf{R}$ , that is,  $\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_{K-1}$ , are the desired Slepian sequences.

Let us elaborate. We note that  $\lambda_0 = E[|\mathbf{q}_0^T \mathbf{x}(n)|^2]$  is the output power of a FIR filter with the coefficient vector  $\mathbf{q}_0$ . Moreover, according to (24.12),  $\mathbf{q}_0$  is selected such that to maximize the output power of the desired filter. On the other hand, when  $x(n)$  is chosen to satisfy (24.16), using the Rayleigh's relation [20] (equivalent of signal power in the time and frequency domain), (24.12) can be rearranged as

$$\begin{aligned} \lambda_0 &= \max_{Q_0(f)} \int_{-0.5}^{0.5} |Q_0(f)|^2 \Phi(f) df \\ &= \max_{Q_0(f)} \int_{-\Delta f/2}^{\Delta f/2} |Q_0(f)|^2 df \\ &= 1 - \min_{Q_0(f)} \int_{\Delta f/2}^{1-\Delta f/2} |Q_0(f)|^2 df, \end{aligned} \quad (24.18)$$

where the last identity follows from the Paseval's identity  $\mathbf{q}_0^T \mathbf{q}_0 = \int_0^1 |Q_0(f)|^2 df = 1$ . Accordingly, one finds that the maximization process in (24.12) results in a filter in which the stopband energy  $\int_{\Delta f/2}^{1-\Delta f/2} |Q_0(f)|^2 df$  is minimized, that is, it attains the maximum attenuation of the side lobes.

Following the same argument,  $\mathbf{q}_1$  will also be the coefficient vector of a low-pass filter whose stopband begins at  $\Delta f/2$  and achieves the minimum stopband energy, subject to the constraint  $\mathbf{q}_0^T \mathbf{q}_1 = 0$ . Clearly, this will result in a filter whose stopband attenuation will not be as good as that of  $\mathbf{q}_0$ . Proceeding further, one finds that subsequent filters,  $\mathbf{q}_2, \mathbf{q}_3, \dots$ , will experience more loss in their stopband attenuation because of more constraints.

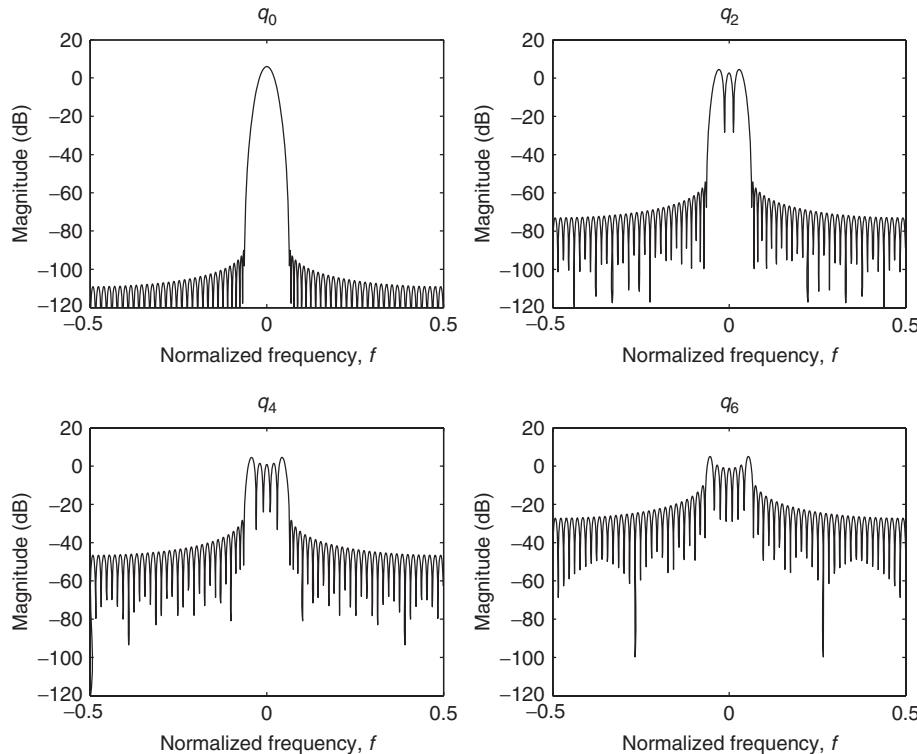
From the above discussions, the Slepian sequences define the coefficients of a set of prototype filters with certain optimal properties. In particular, the good stopband behavior of these filters makes them a desirable candidate in the application of nonparametric spectral estimation, particularly in applications where a wide spectral dynamic range is required. For obvious reasons that follow from the above derivations, the term *eigen-filters* is often used to refer to these filters. The term *prolate filters* has also been used. In this chapter, we have chosen to use the term prolate filters.

It is also worth noting that the orthogonality condition imposed on the prolate filters is to assure that under the condition where  $S_{xx}(f)$  variation over each subband is negligible, the set of outputs from various filter banks that correspond to the same subband will be uncorrelated. Hence, averaging the energy of the signals from the filter banks results in spectral estimates with minimum variance. This may be related to the concept of the effective degrees of freedom (EDF) that is introduced in Section 24.4.4 and further studied in Appendix A.

#### 24.4.2 Example of Prolate Filters

In order to better understand the capabilities as well as shortcomings of the MTSE, without any loss in the generality of the conclusions that will be derived, we continue our discussion with a numerical example. We choose  $N = 8$  and set  $\Delta f = 1/N$ . Here, as in the case of PSE,  $N$  denotes the number of frequency bands within the principal frequency range  $0 \leq f \leq 1$  or, equivalently,  $-0.5 \leq f \leq 0.5$ . This means we divide the principal frequency range into  $N$  mutually exclusive bands,<sup>3</sup> each of width  $\Delta f = 1/N$ . We set the length of prolate filters as  $M = KN$ , with  $K = 8$ , and note that  $K = M/N = M\Delta f$ . Moreover, since  $M$  is the length of signal samples that will be processed by the MTSE,  $K = M\Delta f$  is called the *time-bandwidth product*.

Figure 24.9 presents the magnitude responses of the first few prolate filters of this design. For brevity of the presentation, only the even-numbered filters are shown. This figure reveals the following facts. Only the first few prolate filters have good stopband attenuation. Thomson [5] has identified the number of useful prolate filters as  $K$ . However, as is evident from the results of Figure 24.9, the stopband responses of the



**Figure 24.9** Magnitude responses of the first few prolate filters of length  $M = 64$  with the target frequency resolution  $\Delta f = \frac{1}{8}$ . Only the even-numbered filters are shown. The odd-numbered filters have responses that fall in between the presented ones. Note that the number of lobes within the passband  $\Delta f$  is equal to the prolate filter number plus 1.

<sup>3</sup>Note that although the choice of  $\Delta f = 1/N$  that leads to a set of mutually exclusive bands seems natural and has been considered mostly in the literature it is not a requirement of MT method and is not necessarily the best choice. More on this will be presented in Section 24.4.6.

prolate filters deteriorate very fast in the higher numbered filters. This, as discussed in [5], imposes some limitation on the number of usable prolate filters. The adaptive MTSE method discussed below suggests a solution (though computationally expensive) to this problem. A more manageable solution to this problem is discussed in Section 24.4.6.

### 24.4.3 Adaptive MTSE

A naive implementation of MTSE may be based on (24.11), that is, direct averaging of the instantaneous power of the output signals from multiple filter banks realized using the first  $K$  prolate filters. Moreover, noting that  $\kappa_k(f_i)$  is an output sample of the eigenfilter  $Q_k(f - f_i)$ , we recall from the theory of linear time-invariant systems [20] that

$$E[|\kappa_k(f_i)|^2] = \int_{-1/2}^{1/2} S_{xx}(f) |Q_k(f - f_i)|^2 df. \quad (24.19)$$

Although, in practice, MTSE always obtains the estimates of samples of  $S_{xx}(f)$  over a grid of frequencies, it is convenient here to write (24.11) in terms of the continuous frequency  $f$  as

$$\hat{S}_{xx}(f) = \frac{1}{K} \sum_{k=0}^{K-1} \hat{S}_{xx}^{(k)}(f), \quad (24.20)$$

where  $\hat{S}_{xx}^{(k)}(f) = |\kappa_k(f)|^2$ , and  $\kappa_k(f)$  is obtained through an equation similar to (24.10). Also, (24.19) implies that

$$S_{xx}^{(k)}(f) \stackrel{\text{def}}{=} E[\hat{S}_{xx}^{(k)}(f)] = S_{xx}(f) \star |Q_k(f)|^2, \quad (24.21)$$

where  $\star$  denotes convolution. Note that, here, the convolution is *circular* because  $S_{xx}(f)$  and  $|Q_k(f)|^2$  are periodic functions of  $f$ . In (24.21) the convolution has two effects. First, the nonzero width of the main lobe of  $|Q_k(f)|^2$  has a smearing effect on the estimated spectra. Second, the nonzero side lobes of  $|Q_k(f)|^2$  introduce some leakage and, thus, bias in the estimates.

We note that while the leakage from high-power bands to the low-power bands can introduce significant bias in the estimates, the leakage from low-power bands to the high-power bands are less important. This is because in the former case a portion of a large quantity (high power) is added to a small quantity (low power), while in the latter a portion of a small quantity is added to a large quantity. Noting this, Thomson [5] suggested a weighted average of the spectral samples from different prolate filters and proposed an iterative procedure that adjusts the weighting coefficients for each band to strike a balance between the bias and the variance of the estimates. The procedure proposed by Thomson seeks the joint solution of the following pair of equations for each value of  $f$ :

$$\hat{S}(f) = \frac{\sum_{k=0}^{K-1} |d_k(f)|^2 \hat{S}_k(f)}{\sum_{k=0}^{K-1} |d_k(f)|^2}, \quad (24.22)$$

where

$$d_k(f) = \frac{\sqrt{\lambda_k} S(f)}{\lambda_k S(f) + B_k(f)}, \quad (24.23)$$

and  $S(f)$  is the exact power spectrum (that we seek to estimate!) and  $B_k(f)$  is the leakage (power) that the  $k$ th filter bank picks up at the frequency  $f$  (also unknown). Thomson has proposed the following iterative algorithm to obtain  $\hat{S}(f)$ :

1. Start with a coarse estimate  $\hat{S}(f)$ .
2. Use the present estimate  $\hat{S}(f)$  to obtain estimates of  $B_k(f)$  by evaluating the integrals

$$\hat{B}_k(f) = \int_{-1/2}^{1/2} \hat{S}(\nu) |Q_k(\nu - f)|^2 d\nu \quad \text{for } k = 0, 1, \dots, K-1, \quad (24.24)$$

where  $\int$  means the integration excludes the interval  $(f - \Delta f/2, f + \Delta f/2)$ .

3. Use  $\hat{S}(f)$  and  $\hat{B}_k(f)$  in (24.23) to obtain estimates  $\hat{d}_k(f)$ , for  $k = 0, 1, \dots, K-1$ .
4. Use  $\hat{d}_k(f)$  in (24.22) to obtain a new estimate of  $\hat{S}(f)$ .
5. Proceed through steps 2 to 4 until  $\hat{S}(f)$  converges.

This is a rather computationally expensive procedure. It was, thus, revisited in [21], where the authors suggested a much simpler method for a coarse estimation of  $\hat{B}_k(f)$ . This method reduces the complexity of the adaptive MTSE considerably, at the cost of some mild bias in the estimates. However, still the complexity may remain a concern in the application of cognitive radios, where the available resources at each cognitive node may be limited and the processing time is also a concern.

#### 24.4.4 Effective Degrees of Freedom

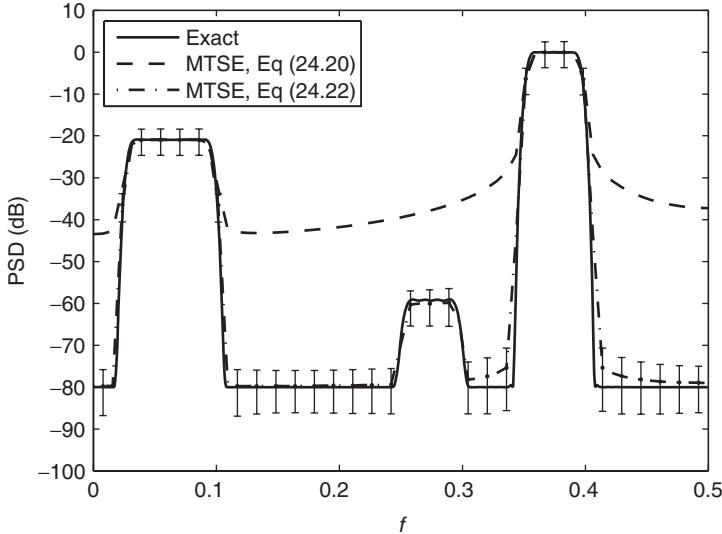
Thomson [5] has noted that a useful by-product of the above procedure is the function

$$v(f) = 2 \sum_{k=0}^{K-1} |d_k(f)|^2, \quad (24.25)$$

which has the following interpretation. If for a particular frequency  $f$ ,  $m$  out of  $K$   $d_k(f)$ 's are equal to 1 and the rest are equal to 0, the weighted mean (24.22) will have a chi-square distribution with  $2m$  degrees of freedom. Also, we note that the variance of the spectral estimates decreases as  $m$  increases; recall that the main reason for introduction of the MTSE was to reduce the variance of the estimates by introducing a number of independent estimates of  $S_{xx}(f)$ . When some or all of  $d_k(f)$ 's are fractional numbers, Thomson argues that still one may use  $v(f)$  as a parameter that characterizes the accuracy of the estimates and thus has referred to  $v(f)$  as a *stability measure* for the estimates. Thomson has also referred to  $v(f)$  as the *sensitivity function*. In this chapter, we follow the degrees of freedom terminology used for the chi-square distributions [20], and accordingly refer to  $v(f)$  as the *effective degrees of freedom (EDF)*. Further discussion on EDF can be found in Appendix A.

#### 24.4.5 Numerical Example of MTSE

Figure 24.10 presents a statistical evaluation of 10,000 snapshots of the MTSE when applied to the test signal that was introduced in Section 24.3.5. The mean values of the



**Figure 24.10** Example of power spectral density (PSD) of a random signal and statistical evaluation of 10,000 independent snapshots of the MTSE. The vertical lines indicate the 95% confidence intervals.

results of the estimators based on both (24.20) and (24.22) are presented. As predicted above, a direct averaging based on (24.20) results in a very limited spectral dynamic range and, thus, fails to identify the lower level portions of the spectrum. On the other hand, the adaptive averaging based on (24.22) can easily cover the desired spectral dynamic range. However, this will be at the cost of some loss in the precision of the estimates at lower level parts of the spectrum. This is also shown in Figure 24.10 by presenting the vertical lines that indicate the intervals over which 95% of the estimates fall; the 95% confidence intervals [3]. Note that these intervals increase over the bands where the PSD is lower. This phenomenon can be easily explained if we note that spectral leakages lead to smaller values of  $d_k(f)$ 's (hence, a smaller EDF) at the frequencies where PSD is lower. This is demonstrated further in Figure 24.11, where a plot of  $v(f)$  is presented. Note that while at higher levels of PSD the desired maximum EDF of 16 is achieved, the EDF reduces to an average of 6 or lower at other parts of the spectrum.

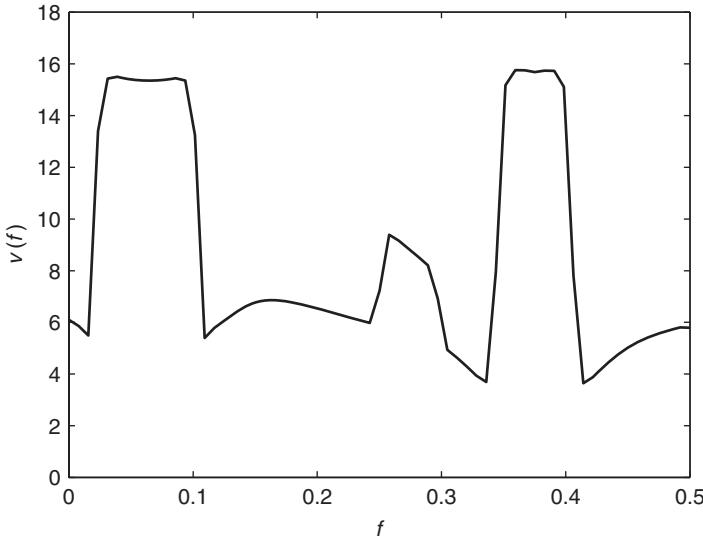
#### 24.4.6 Power Transfer Function

Consider the following (nonadaptive) estimate of  $S_{xx}(f)$ :

$$\hat{S}_{xx}^L(f) = \frac{1}{L} \sum_{k=0}^{L-1} \hat{S}_{xx}^{(k)}(f). \quad (24.26)$$

We let  $L$  takes values of 1 to  $K$ , and refer to  $\hat{S}_{xx}^L(f)$  as the  $L$ th order estimate of  $S_{xx}(f)$ . Taking expectations on both sides of (24.26) and using (24.21), we obtain

$$S_{xx}^L(f) = S_{xx}(f) \star P_L(f), \quad (24.27)$$



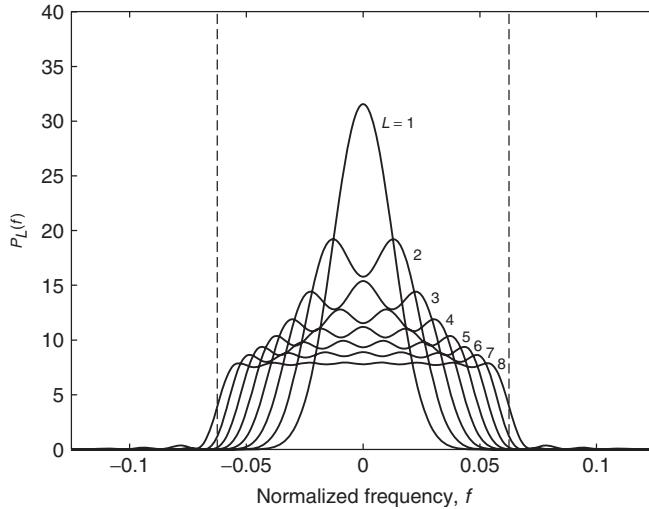
**Figure 24.11** Average value of the EDF,  $v(f)$ , as a function of the frequency  $f$ . The averaging is performed over 10,000 independent snapshots of the MTSE.

where  $S_{xx}^L(f) = E[\hat{S}_{xx}^L(f)]$  and

$$P_L(f) = \frac{1}{L} \sum_{k=0}^{L-1} |Q_k(f)|^2. \quad (24.28)$$

We refer to  $P_L(f)$  as the *Lth order power transfer function* of the MTSE.

It is instructive to explore the plots of the power transfer function  $P_L(f)$  for various choices of the order  $L$ . Figure 24.12 presents plots of  $P_L(f)$  for the prolate filters that were presented in Section 24.4.2. The vertical dashed lines show the  $\pm\Delta f/2$  boundaries. The plots reveal the following interesting fact. Recall that each prolate filter (Slepian sequence) contributes to part of the signal energy within the band of interest. From Figure 24.12, one may infer that the first prolate filter picks signal energy from the middle of the band. The second prolate filter picks signal components from the neighborhood of the center of the band and expands the passband of  $P_L(f)$  slightly. The subsequent prolate filters each cover more of the frequency range  $(-\Delta f/2, \Delta f/2)$ . The full band is almost covered when the number of prolate filters is  $L = K = M\Delta f$ , the time–bandwidth product. However, the side lobes of  $P_L(f)$  also increase as  $L$  increases, and as discussed above, one may need to keep  $L$  to a small value to avoid significant signal leakage and, hence, bias in the spectral estimates. On the other hand, when  $L$  is small, as is clearly seen from the plots in Figure 24.12, the MTSE may miss seeing part of each band within the filter banks, that is, part of the band  $(f_i - \Delta f/2, f_i + \Delta f/2)$  for different choices of  $f_i$ . As a result, MTSE may fail to detect a narrowband signal that may be present in a hidden part of the spectrum. Extending this, we may argue that the adaptive MTSE may fail to detect low-power narrowband signals within a wideband spectrum when they are hidden in portions of the frequency band that are not covered by  $P_L(f)$  when the adaptive MTSE chooses a small value for  $L$ .



**Figure 24.12** Plots of the power transfer function  $P_L(f)$  for different choice of the order  $L$ . The design parameter are  $N = 8$ ,  $\Delta f = 1/N$ , and  $M = KN = 8 \times 8 = 64$ .

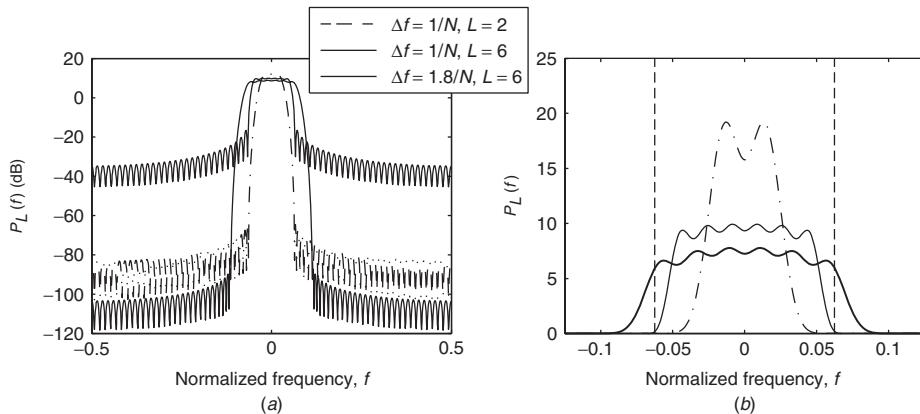
#### 24.4.7 Modified (Nonadaptive) MTSE

The above observations suggest the following modification for more effective design of prolate filters as well as a more efficient implementation of the MTSE. The goal here to resort to a nonadaptive estimation method, to cut the computational complexity, while using a fair number of prolate filters with small side lobes, to improve on the spectral dynamic range.

In the designs used to generate the results of Figures. 24.10 and 24.11, we let  $\Delta f = 1/N$  to divide the full band of the spectrum to  $N$  mutually exclusive bands. Although this is not a restriction in MT method, it seems to be the most natural choice and appears to have been more adopted in the past literature. Here, we propose to design the prolate filters with a choice of  $\Delta f > 1/N$ . Figure 24.13 compares a few results of the designs for the choices of  $\Delta f = 1/N$  and  $\Delta f = 1.8/N$ , through the power transfer function  $P_L(f)$ . All designs are based on the same prolate filter length  $M = KN = 8 \times 8 = 64$ . The design with  $\Delta f = 1.8/N$  and  $L = 6$  achieves a stopband of around  $-100$  dB; hence, it can cover a very wide spectral dynamic range (say, 90 dB), without resorting to the adaptive method of Section 24.4.3. The design with  $\Delta f = 1/N$  and  $L = 6$ , on the other hand, has a very poor stopband. Reducing  $L$  to 2, in the latter case, to improve on the stopband of  $P_L(f)$ , the stopband drops to around  $-80$  dB, still 20 dB poorer than the design with  $\Delta f = 1.8/N$  and  $L = 6$ . Moreover, the bandwidth of  $P_L(f)$  reduces to about one half of  $\Delta f$ , thus, resulting in an MTSE that is blind to about 50% of the bands of interest. We note that the superior performance of the design based on the extended value  $\Delta f = 1.8/N$  is achieved at the cost of some mild leakage from the adjacent bands. This seems to be a reasonably good compromise choice.

### 24.5 FILTER BANK SPECTRAL ESTIMATOR

Multicarrier communications have been emphasized as one of the major candidates for CR systems [22]. Also, although the conventional orthogonal frequency division



**Figure 24.13** Plots of power transfer function  $P_L(f)$  for three designs of prolate filters. All designs are based on prolate filter length  $M = 64$ . (a) Has decibel scale on the vertical axis to allow us to see the stopband attenuations. On the other hand (b) has a linear scale on the vertical axis and is zoomed on the passband of the filters to allow us to see the coverage of the passband and also leakage to adjacent bands. The vertical dashed lines show the  $\pm\Delta f/2$  boundaries.

multiplexing (OFDM) has been adopted by the IEEE 802.22 standard [23], other studies have shown that in cases where multiple nodes in a network may transmit concurrently in an FDM mode, filter-bank-based multicarrier communication systems can perform significantly better [24–26]. When a filter bank multicarrier technique is exploited as the physical layer of a CR network, the same filter bank can also be used for channel sensing. Hence, in such systems, channel sensing will be done at virtually no cost.

In this section, we begin with a brief review of the filter bank multicarrier communication techniques and then show how the filter banks used for signal demodulation at the receiver can also be used for spectral estimation.

#### 24.5.1 Filter Bank Multicarrier Communication Techniques

In the past, three different filter bank multicarrier communication techniques have been proposed. Pioneering work on filter bank multicarrier communication techniques was done by Chang [27] and Saltzberg [28] in the mid-1960s. Saltzberg showed that by proper design of a transmit pulse shape in a multichannel QAM system, and by introducing a half symbol space delay between the in-phase and quadrature components of QAM symbols, it is possible to achieve a baud-rate spacing between adjacent subcarrier channels and still recover the information symbols, free of intersymbol interference (ISI) and intercarrier interference (ICI). This leads to the *maximum spectral efficiency*. Further progress was made by Hirosaki [29] who showed that the transmitter and receiver parts of this modulation method could be implemented efficiently in a polyphase structure. The method was called orthogonally multiplexed QAM (OQAM) in [29]. OQAM has later been referred to as OFDM-OQAM, with the acronym OQAM standing for *offset QAM*, reflecting the fact that the in-phase and quadrature components of each QAM symbol are time offset with respect to each other.

In the 1990s, the advancements in digital subscriber line (DSL) technology motivated more activities in the development of other filter-bank-based multicarrier communication techniques that could better suit the DSL channels. Early development

in this area is an American National Standards Institute (ANSI) contribution by Tzannes et al., which was later expanded and called discrete wavelet multitone (DWMT) [30]. In [31], it was shown that DWMT uses cosine-modulated filter banks that are more frequently used for signal compression. The name cosine modulated multitone (CMT) was later adopted for this class of modulators [32]. It was also noted that in CMT, each subcarrier channel transmits a PAM symbol using vestigial sideband (VSB) modulation [31].

Filtered multitone (FMT) is another multicarrier modulation technique that was specifically developed for DSL applications [33]. As opposed to OFDM-OQAM and CMT, which allow for overlapping of adjacent subcarrier bands, in FMT subcarrier bands are disjoint. It is thus less bandwidth efficient than CMT and OFDM-OQAM.

### 24.5.2 Prototype Filter

Even though the OFDM-OQAM, CMT, and FMT are three different multicarrier techniques (with the differences noted above), they share the same prototype filter. Since in the FMT each subcarrier channel is a conventional narrowband channel, the optimum transmitter and receiver filters are a pair of matched root-Nyquist filters. It turns out that in the OFDM-OQAM and CMT, also to satisfy the conditions required for perfect separation of subcarrier streams, the prototype filters at the transmit and receive sides should be a pair of matched root-Nyquist filters [28, 31]. In the cases of OFDM-OQAM the root-Nyquist filter is designed such that the zero crossing occurs at the intervals of  $N$  samples, where  $N$  is the maximum number of subcarriers. In that case,  $H(z)$  and  $G(z) = H(z)H(z^{-1})$  are called root-Nyquist ( $N$ ) and Nyquist ( $N$ ) filters, respectively [34]. In addition,  $G(z)$  is usually normalized to have a middle tap of unity. Hence, in the time domain,  $G(z)$  should satisfy the following constraints:

$$g(n) = \begin{cases} 1, & n = 0, \\ 0, & n = mN, m \neq 0. \end{cases} \quad (24.29)$$

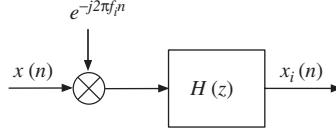
In the case of CMT,  $H(z)$  must be chosen to be a root-Nyquist ( $2N$ ) filter [31].

From a filter bank point of view, the demodulator in either of the above multicarrier systems performs the following task. For each subcarrier, the corresponding portion of the input signal is down-converted to baseband, low-pass filtered, and decimated. From a spectral analysis point of view, we are interested in the averaged signal powers of each of the decimated signal sequences. Hence, the spectral samples will be available at virtually no additional cost. Also, to be able to evaluate the variance of estimated averages, we are interested in the correlation properties of the decimated signal samples of each subcarrier band. We thus proceed with a discussion on the correlation properties/correlation coefficients of the demodulated subband signals.

### 24.5.3 Correlation Coefficients of Demodulated Signals

We are interested in the correlation coefficients of the process at the output of the system shown in Figure 24.14, where  $f_i$  is the carrier frequency of the  $i$ th band and  $H(z)$  is the prototype filter. Noting that the demodulator shifts the spectrum of  $x(n)$  to left by  $f_i$ , the power spectral density of  $x_i(n)$  is given by

$$S_{x_i x_i}(f) = S_{xx}(f + f_i)|H(f)|^2. \quad (24.30)$$



**Figure 24.14** Demodulation process of extracting baseband signal associated with  $i$ th subcarrier in a filter bank multicarrier receiver.

Moreover, if we assume that  $H(z)$  is a narrowband filter, thus, is nonzero only for values of  $f$  in a small range around zero, and for values of  $f$  over this range the approximation  $S_{xx}(f + f_i) \approx S_{xx}(f_i)$  holds, we obtain

$$S_{x_i x_i}(f) \approx S_{xx}(f_i) |H(f)|^2, \quad (24.31)$$

where, obviously,  $S_{xx}(f_i)$  is a constant. Replacing the approximation sign by an equality sign and rewriting the result in terms of the  $z$ -transform variable  $z$ , we get

$$\Phi_{x_i x_i}(z) = S_{xx}(f_i) H(z) H(z^{-1}). \quad (24.32)$$

The inverse  $z$  transform of  $\Phi_{x_i x_i}(z)$  gives the correlation coefficients of  $x_i(n)$ ,  $\phi_{x_i x_i}(k)$ . The correlation coefficients of the decimated version of  $x_i(n)$  are obtained by decimating the sequence  $\phi_{x_i x_i}(k)$ .

When  $H(z)$  is a root-Nyquist (N) filter, (24.32) implies that the autocorrelation coefficients  $\phi_{x_i x_i}(k)$  will resemble a Nyquist (N) sequence, say,  $g_N(n)$ , where the subscript  $N$  is to indicate explicitly the zero-crossing spacing of the autocorrelation coefficients. Therefore, the correlation matrix of the observation vector  $\mathbf{x}_i(n) = [x_i(n), x_i(n - L), \dots, x_i(n - (K - 1)L)]$ , where  $L$  is the sample spacing, is the Toeplitz matrix

$$\mathbf{R}_{x_i x_i} = S_{xx}(f_i) \mathbf{A}, \quad (24.33)$$

where

$$\mathbf{A} = \begin{bmatrix} g_N(0) & g_N(L) & \cdots & g_N((K-1)L) \\ g_N(-L) & g_N(0) & \cdots & g_N((K-2)L) \\ \vdots & \vdots & \ddots & \vdots \\ g_N(-(K-1)L) & g_N(-(K-2)L) & \cdots & g_N(0) \end{bmatrix}. \quad (24.34)$$

Moreover, if we make the common assumption that  $x_i(n)$  is a zero-mean Gaussian process<sup>4</sup>, the observation vector  $\mathbf{x}_i(n)$  will be a zero-mean Gaussian random vector with the covariance matrix  $\mathbf{R}_{x_i x_i}$ .

In the case of CMT, (24.31) and (24.34) should be slightly modified. Since each subcarrier in CMT is a VSB modulated PAM signal, the filter  $H(z)$  should be replaced by a VSB low-pass filter [31]:

$$H_{\text{VSB}}(z) = H(z e^{-j\pi/2N}), \quad (24.35)$$

<sup>4</sup>The assumption of Gaussian  $x_i(n)$ , irrespective of the distribution of the input,  $x(n)$ , is widely used in the literature. It is argued that since the filtering process linearly combines a (large) number of random variables, it follows from the central limit theorem that the result will be a Gaussian random variable [5].

where  $H(z)$  is a root-Nyquist ( $2N$ ) filter. Using (24.35), the modified correlation matrix that should be used for any further study of the CMT is obtained by replacing  $\mathbf{A}$  in (24.34) by

$$\mathbf{A} = \begin{bmatrix} g_{2N}(0) & g_{2N}(L) & \cdots & g_{2N}((K-1)L) \\ g_{2N}(-L) & \times \exp\left(-j\frac{\pi L}{2N}\right) & \cdots & \times \exp\left[j\frac{\pi L(K-1)}{2N}\right] \\ \vdots & \vdots & \ddots & \vdots \\ g_{2N}(-(K-1)L) & g_{2N}(-(K-2)L) & \cdots & g_{2N}(0) \\ \times \exp\left[-j\frac{\pi L(K-1)}{2N}\right] & \times \exp\left[-j\frac{\pi L(K-2)}{2N}\right] & \cdots & \end{bmatrix}. \quad (24.36)$$

Although the matrices  $\mathbf{A}$  in (24.34) and (24.36) are somewhat different, both matrices will share the same eigenproperties. In particular, if in (24.36)  $2N$  is replaced by  $N$ , the resulting  $\mathbf{A}$  will have exactly the same eigenvalues as those of  $\mathbf{A}$  in (24.34). Noting this, even though all the references to  $\mathbf{A}$  in the sequel are made to (24.34), the conclusions drawn are applicable to all three types of the filter bank multicarrier techniques that were introduced in Section 24.5.1.

#### 24.5.4 Sample Spacing and Effective Degrees of Freedom

An estimate of  $S(f_i)$  based on the observation vector  $\mathbf{x}_i(n) = [x_i(n), x_i(n-L), \dots, x_i(n-(K-1)L)]$  is obtained by evaluating the average

$$\hat{S}(f_i) = \frac{1}{K} \sum_{k=0}^{K-1} |x_i(n-kL)|^2. \quad (24.37)$$

Consider the case where the covariance matrix of  $\mathbf{x}_i(n)$  is given by (24.34), and  $L = N$ . In this case,  $\mathbf{A}$  reduces to the identity matrix,  $\mathbf{I}$ , and this indicates that the elements of  $\mathbf{x}_i(n)$  are a set of identically independently distributed (i.i.d.) complex-valued Gaussian random variables. The summation  $\sum_{k=0}^{K-1} |x_i(n-kL)|^2$  is thus a chi-square random variable with  $2K$  degrees of freedom. When  $L \neq N$ ,  $\mathbf{A}$  is no longer equal to the identity matrix and, hence, the elements of  $\mathbf{x}_i(n)$  will no longer be independent. This clearly reduces the degrees of freedom (or the EDF, to be more accurate) of the summation  $\sum_{k=0}^{K-1} |x_i(n-kL)|^2$ . We evaluate the EDF of  $\mathbf{A}$  by introducing and evaluating a sensitivity function similar to the one introduced in (24.25).

Since  $\mathbf{A}$  is a Hermitian matrix, one can find a unitary matrix  $\mathbf{U}$  such that  $\mathbf{A} = \mathbf{U}\Lambda\mathbf{U}^H$ , where  $\Lambda$  is a diagonal matrix with eigenvalues of  $\mathbf{A}$ ,  $\lambda_0, \lambda_1, \dots, \lambda_{K-1}$ , at its diagonal. Next, we define

$$\mathbf{x}_i^U(n) = \Lambda^{-1/2} \mathbf{U}^H \mathbf{x}_i(n) \quad (24.38)$$

and note that since  $\mathbf{x}_i(n)$  is a Gaussian vector,  $\mathbf{x}_i^U(n)$  is also Gaussian. Moreover, the covariance matrix of  $\mathbf{x}_i^U(n)$  is obtained as

$$\begin{aligned} E \left[ \mathbf{x}_i^U(n) (\mathbf{x}_i^U(n))^H \right] &= E \left[ \Lambda^{-1/2} \mathbf{U}^H \mathbf{x}_i(n) \mathbf{x}_i^H(n) \mathbf{U} \Lambda^{-1/2} \right] \\ &= S_{xx}(f_i) \Lambda^{-1/2} \mathbf{U}^H \mathbf{A} \mathbf{U} \Lambda^{-1/2} = S_{xx}(f_i) \mathbf{I}. \end{aligned} \quad (24.39)$$

This result shows that  $\mathbf{x}_i^U(n)$  is a vector of  $K$ -independent Gaussian variables, all with the same variance  $S_{xx}(f_i)$ .

On the other hand, we note that (24.38) can be rearranged as

$$\mathbf{x}_i(n) = \mathbf{U}\Lambda^{1/2}\mathbf{x}_i^U(n). \quad (24.40)$$

Using (24.40), (24.37) may be written as

$$\begin{aligned} \hat{S}(f_i) &= \frac{1}{K} \mathbf{x}_i^H(n) \mathbf{x}_i(n) \\ &= \frac{1}{K} (\mathbf{x}_i^U(n))^H \Lambda \mathbf{x}_i^U(n) \\ &= \frac{\sum_{k=0}^{K-1} \lambda_k |[\mathbf{x}_i^U(n)]_k|^2}{\sum_{k=0}^{K-1} \lambda_k}, \end{aligned} \quad (24.41)$$

where  $[\mathbf{x}_i^U(n)]_k$  denotes the  $k$ th element of  $\mathbf{x}_i^U(n)$  and the last identity is obtained by noting that

$$\sum_{k=0}^{K-1} \lambda_k = \text{tr} \left[ \frac{1}{S_{xx}(f_i)} \mathbf{A} \right] = K,$$

where  $\text{tr}[\cdot]$  denotes the trace of a matrix.

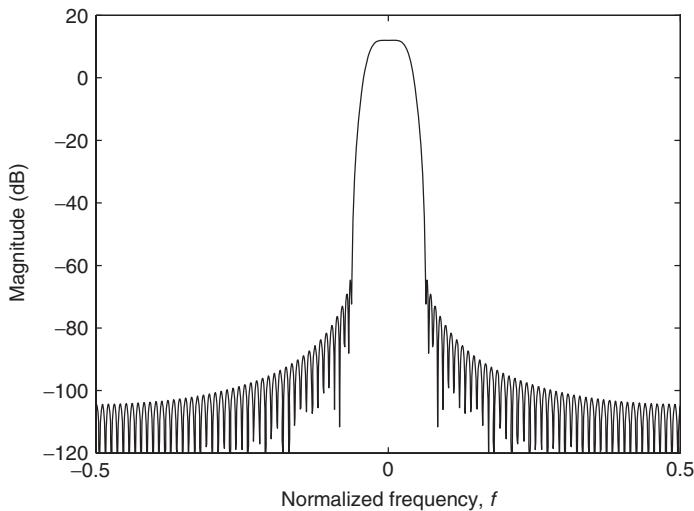
Now, let us look at the fraction on the right-hand side of (24.41). This, clearly, is a weighted mean-square of  $K$  i.i.d zero-mean complex Gaussian random variables with variance  $S_{xx}(f_i)$ . This is similar to (24.22) with  $|d_k(f)|^2$  replaced by  $\lambda_k$ . However, before using  $\lambda_k$ 's to define a similar sensitivity function, we should be reminded that  $|d_k(f)|^2$ 's, by definition, are smaller than or equal to 1, while  $\lambda_k$ 's are not. To explain the point, consider a pair of complex Gaussian random variables  $x$  and  $y = \lambda x$ , where  $\lambda$  is a constant. Clearly, both  $|x|^2$  and  $|y|^2$  have chi-square distributions with 2 degrees of freedom, irrespective of the value of  $\lambda$ . Similarly, if  $x_0$  and  $x_1$  are i.i.d complex Gaussian random variables and  $\lambda_0$  and  $\lambda_1$  are some constants, one will find that  $\lambda_0|x_0|^2 + \lambda_1|x_1|^2$  has a chi-square distribution with 4 degrees of freedom if  $\lambda_0 = \lambda_1 \neq 0$ , and has a chi-square distribution with 2 degrees of freedom if  $\lambda_0 \neq 0$  and  $\lambda_1 = 0$ . The EDF of  $\lambda_0|x_0|^2 + \lambda_1|x_1|^2$  will be a number between 2 and 4, if  $\lambda_0 \neq \lambda_1$  and both are nonzero. A good approximation to the EDF, here, following Thomson [5], is  $2/\lambda_{\max}(\lambda_0 + \lambda_1)$ , where  $\lambda_{\max} = \max(\lambda_0, \lambda_1)$ . Generalizing this result, we argue that the estimate  $\hat{S}(f_i)$  may be thought of as a random variable with a chi-square distribution whose EDF is given by

$$v = \frac{2}{\lambda_{\max}} \sum_{k=0}^{K-1} \lambda_k, \quad (24.42)$$

where  $\lambda_{\max}$  is the maximum of the eigenvalues  $\lambda_0, \lambda_1, \dots, \lambda_{K-1}$ .

### 24.5.5 Numerical Experiments

To learn about the typical prototype filters that can be designed and may be used in the application of interest, we designed a number of root-Nyquist filters using a MATLAB program that has been developed in [35]. The program that



**Figure 24.15** Magnitude responses of typical prototype filter with comparable response to the prolate filters shown in Figure 24.9. Filter length here is 97.

we have used is called `rNyquistM0.m` and is available at the author's website (<http://www.ece.utah.edu/~farhang/>). To present a result comparable with the prolate filters presented in Figure 24.9 (in particular, to achieve the same frequency resolution), we choose  $N = 16$ , a roll-off factor  $\alpha = 1$  and a filter order  $M - 1 = 6 \times N = 96$ . The magnitude response of the designed filter is shown in Figure 24.15. It may be noted that, here, the filter length is approximately 50% longer than the prolate filters of Figure 24.9, and it has a magnitude response very similar to the best prolate filter of Figure 24.9, that is,  $q_0$ . In particular, the width of its main lobe is equal to  $(1 + \alpha)/N = \frac{1}{8}$ . The increase in the filter length, here, may be thought as the price that one has to pay to satisfy the Nyquist condition. It is also worth noting that the same ratio between the lengths of the prototype filter and prolate filters will remain if one attempts to increase the frequency resolution of the spectral estimates by increasing the filters' lengths.

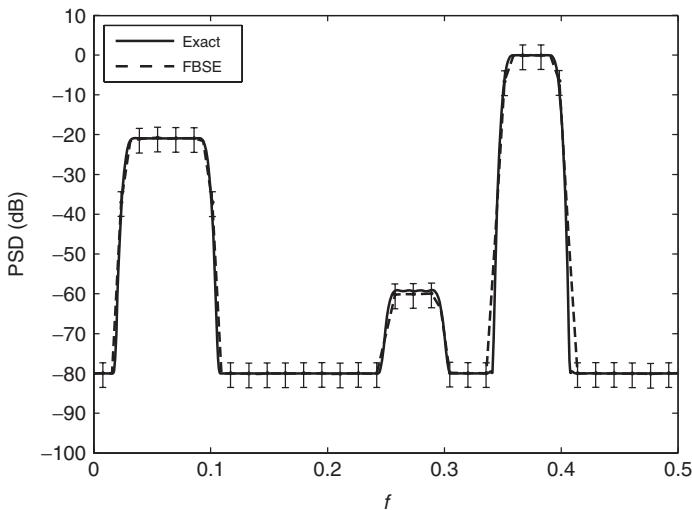
Next, we use the proposed filter bank spectral estimator (FBSE) and repeat the spectral estimation problem whose results for the MTSE were presented in Figure 24.10. We use the root-Nyquist filter design program `rNyquistM0.m` that was mentioned above to design a prototype filter for the FBSE. To achieve the same frequency resolution as in Figure 24.10, we use the parameters  $N = 256$  and  $\alpha = 1$ . Also, to achieve the same stopband attenuation as in Figure 24.15, hence, to be able to easily handle a spectral dynamic range of 60 dB (or, even better), we choose a filter order  $M - 1 = 6 \times N = 1536$ .

Before presentation of the spectral estimates, we use the designed prototype filter to construct the matrix  $\mathbf{A}$  and from there evaluate the EDF,  $v$ , for a number of choices of  $L$  and  $K$ . The results are presented in Table 24.1. From these results, we observe that for a given window length  $KL$  and  $L \leq N$ ,  $v$  remains almost a constant. An explanation to this observation is given in Appendix B.

Figure 24.16 presents the results of a statistical evaluation of 10,000 snapshots of the FBSE. The prototype filter used to implement the FBSE is the one discussed above

**TABLE 24.1** Effective Degrees of Freedom  $v$  for  $N = 256$  and Filter Order  $6N$  as  $L$  and  $K$  Vary

$L$	$K$	$KL$	$v$	$L$	$K$	$KL$	$v$	$L$	$K$	$KL$	$v$
256	1	256	2	128	1	128	2	64	1	64	2
	2	512	4		2	256	2.51		2	128	2.12
	4	1024	8		4	512	4.14		4	256	2.60
	8	2048	16		8	1024	8.02		8	512	4.17
					16	2048	16.00		16	1024	8.02
								32	2048	16.00	32
									32	1024	8.03
									64	2048	16.00



**Figure 24.16** Example of power spectral density (PSD) of random signal and statistical evaluation of 10,000 independent snapshots of an FBSE with similar frequency resolution to the MTSE results presented in Figure 24.10. Spectral estimates are obtained by averaging energy of eight signal samples at the output of each subband of the analysis filter bank. The vertical lines indicate the 95% confidence intervals.

and used to generate the results of Table 24.1. The parameter  $K$  is set equal to 8, equal to the number of prolate filters used to generate the results of Figure 24.10. Also, because of the conclusion drawn from the results of Table 24.1, we set  $L = N = 256$ . The input process is the one introduced earlier and used to generate Fig. 24.10. The 95% confidence intervals of the results are also presented.

## 24.6 DISTRIBUTED SPECTRUM SENSING

All the derivations, so far, assume that there is only one receiver that listens to the spectrum and decides presence or absence of the spectral holes. In a cognitive network clearly there exist multiple nodes, and each node may independently sense the spectrum, and the results from various nodes may be processed collectively, say, at a central node (a common base station) for a more reliable signal detection and estimation of the background noise level. In this section, we present the details of such a cooperative detector.

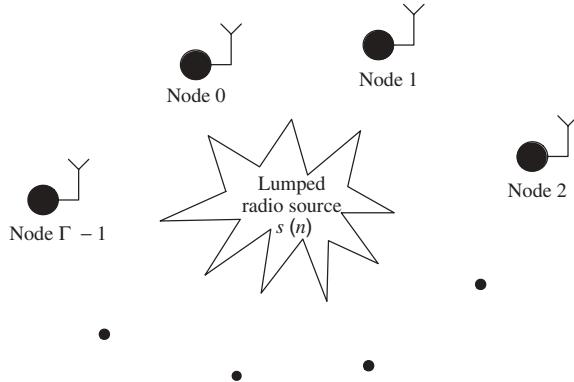


Figure 24.17 Network model.

### 24.6.1 Network Signal Model

We consider a network of  $\Gamma$  nodes and one lumped radio stimuli source, as in in Figure 24.17. Although, in reality, there are a number of radio sources around the network, we have lumped these together as a single source and represent it with  $s(n)$ .

We assume that similar and synchronized sets of filter banks are used at all the nodes for spectrum sensing. We concentrate on the signals from the  $i$ th band of the filter banks. Moreover, we assume that each band is sufficiently narrow in width such that the channel gain between the source  $s(n)$  and the node  $\gamma$  over the  $i$ th band can be approximated by a flat gain  $h_{\gamma,i}$ . Furthermore, we assume that there are  $K$  parallel filter banks at each nodes. This assumption directly addresses the case of MTSE and covers the cases FBSE and PSE, with some modification in notations. Hence, the signal samples at the  $i$ th output of the  $k$ th filter bank at the  $\gamma$ th node will be

$$x_{\gamma,k,i}(n) = h_{\gamma,i} s_{k,i}(n) + v_{\gamma,k,i}(n), \quad (24.43)$$

where  $s_{k,i}(n)$  is the filter bank output if the input to the filter bank was  $s(n)$  and  $v_{\gamma,k,i}(n)$  is the background (including thermal) noise. In the case of MTSE,  $x_{\gamma,k,i}(n)$  is the  $i$ th output of the  $k$ th filter bank at the  $\gamma$ th node. In the case of FBSE,  $x_{\gamma,k,i}(n) = x_{\gamma,i}(n - kL)$ . Relevant modifications to the case of PSE should be obvious.

Combining the above samples in a single matrix for further processing, and dropping the band and time indices  $i$  and  $n$ , for convenience of notations, we define the  $\Gamma$ -by- $K$  matrices  $\mathbf{X}$  and  $\mathbf{V}$  with the entries  $x_{\gamma,k}$  and  $v_{\gamma,k}$ , respectively, and note that

$$\mathbf{X} = \mathbf{h}\mathbf{s}^H + \mathbf{V}, \quad (24.44)$$

where  $\mathbf{h} = [h_0 h_1 \cdots h_{\Gamma-1}]^T$  and  $\mathbf{s} = [s_0 s_1 \cdots s_{K-1}]^H$ . We assume that the noise samples  $v_{\gamma,k}$  are a set of zero-mean i.i.d random variables with variance  $\sigma_v^2$ , the channel gains  $h_\gamma$  are a set of zero-mean i.i.d random variables with variance  $\sigma_h^2$ , the signal samples  $s_k$  are a set of zero-mean i.i.d random variables with variance  $\sigma_s^2$ , and the sets of random variables  $v_{\gamma,k}$ ,  $h_\gamma$ , and  $s_k$  are independent of one another. The goal here is to obtain estimates of the signal power  $E[|h_\gamma s_k|^2] = \sigma_h^2 \sigma_s^2$  and the noise power  $E[|v_{\gamma,k}|^2] = \sigma_v^2$ , based on the available signal samples  $x_{\gamma,k}$ . To this end, we proceed as follows.

We first note that since the sets  $v_{\gamma,k}$ ,  $h_\gamma$ , and  $s_k$  are independent of each other, the best estimates (but not necessarily achievable) of  $\sigma_h^2$ ,  $\sigma_s^2$ , and  $\sigma_v^2$ , based on the available signal samples, are the time averages

$$\hat{\sigma}_h^2 = \frac{1}{\Gamma} \sum_{p=0}^{\Gamma-1} |h_\gamma|^2, \quad (24.45)$$

$$\hat{\sigma}_s^2 = \frac{1}{K} \sum_{k=0}^{K-1} |s_k|^2, \quad (24.46)$$

$$\hat{\sigma}_v^2 = \frac{1}{\Gamma K} \sum_{\gamma=0}^{\Gamma-1} \sum_{k=0}^{K-1} |v_{\gamma,k}|^2. \quad (24.47)$$

Hence, the desired signal power  $E[|h_\gamma s_k|^2] = \sigma_h^2 \sigma_s^2$  is evaluated as

$$\hat{\sigma}_h^2 \hat{\sigma}_s^2 = \left( \frac{1}{\Gamma} \sum_{\gamma=0}^{\Gamma-1} |h_\gamma|^2 \right) \left( \frac{1}{K} \sum_{k=0}^{K-1} |s_k|^2 \right). \quad (24.48)$$

Note that the “hat” signs show the values are estimates.

Next, we define the  $K$ -by- $K$  matrix  $\mathbf{R} = E[\mathbf{X}^H \mathbf{X}]$  and note that using the approximation  $E[\mathbf{V}^H \mathbf{V}] \approx \hat{\sigma}_v^2 P \mathbf{I}$  (which follows since the elements of  $\mathbf{V}$  are zero mean and i.i.d), we obtain the estimate of  $\mathbf{R}$  as

$$\hat{\mathbf{R}} = \Gamma (\hat{\sigma}_h^2 \mathbf{s} \mathbf{s}^H + \hat{\sigma}_v^2 \mathbf{I}). \quad (24.49)$$

Let  $\hat{\lambda}_0, \hat{\lambda}_1, \dots, \hat{\lambda}_{K-1}$  be the eigenvalues of  $\hat{\mathbf{R}}$  in descending order. It is straightforward to show that  $\hat{\mathbf{q}}_0 = \hat{\sigma}_s^{-1} \mathbf{s}$  is the first eigenvector of  $\hat{\mathbf{R}}$ , with the associated eigenvalue

$$\hat{\lambda}_0 = \Gamma K \hat{\sigma}_h^2 \hat{\sigma}_s^2 + \Gamma \hat{\sigma}_v^2. \quad (24.50)$$

The rest of eigenvectors of  $\hat{\mathbf{R}}$  are any arbitrary set of  $K - 1$  unit-length vectors orthogonal to  $\hat{\mathbf{q}}_0$ . These eigenvectors share a common eigenvalue equal to  $\Gamma \hat{\sigma}_v^2$ . Also, since  $\text{tr}[\hat{\mathbf{R}}] = \sum_{k=0}^{K-1} \hat{\lambda}_k$ , we get

$$\text{tr}[\hat{\mathbf{R}}] = \Gamma K (\hat{\sigma}_h^2 \hat{\sigma}_s^2 + \hat{\sigma}_v^2). \quad (24.51)$$

Solving (24.50) and (24.51), we obtain  $\hat{\sigma}_h^2 \hat{\sigma}_s^2 = (K \hat{\lambda}_0 - \text{tr}[\hat{\mathbf{R}}]) / [\Gamma K (K - 1)]$  and  $\hat{\sigma}_v^2 = (\text{tr}[\hat{\mathbf{R}}] - \hat{\lambda}_0) / [\Gamma (K - 1)]$ .

From the above results, we conclude that to obtain the signal power  $E[|h_\gamma s_k|^2] = \sigma_h^2 \sigma_s^2$  and the noise power  $E[|v_{p,k}|^2] = \sigma_v^2$  from the observation matrix  $\mathbf{X}$ , one may take the following steps:

1. Form the  $K$ -by- $K$  matrix  $\mathbf{R} = E[\mathbf{X}^H \mathbf{X}]$ .
2. Evaluate  $\lambda_0$ , the largest eigenvalue of  $\mathbf{R}$ .

**TABLE 24.2** Estimated Values of  $P_{\text{sig}}$  and  $P_{\text{noise}}$  When  $\sigma_h^2 = \sigma_s^2 = 1$ ,  $\sigma_v^2 = 0.01$ , and  $K = 8$ 

	$\Gamma = 5$	$\Gamma = 10$	$\Gamma = 20$	$\Gamma = 50$	$\Gamma = 100$
Mean of $P_{\text{sig}}$	1.0001	0.9988	1.0004	0.9979	1.0017
Std of $P_{\text{sig}}$	0.590	0.486	0.426	0.388	0.370
Mean of $P_{\text{noise}}$	0.0080	0.0090	0.0095	0.0098	0.0099
Std of $P_{\text{noise}}$	0.189	0.126	0.087	0.054	0.038
Mean of $P_{\text{noise}}$					

3. Obtain the estimates of the desired signal power and the noise power as

$$P_{\text{sig}} = \frac{K\lambda_0 - \text{tr}[\mathbf{R}]}{\Gamma K(K-1)} \quad (24.52)$$

and

$$P_{\text{noise}} = \frac{\text{tr}[\mathbf{R}] - \lambda_0}{\Gamma(K-1)}. \quad (24.53)$$

### 24.6.2 Numerical Experiments

In order to evaluate the accuracy of the above power estimates, we simulate the signal model (24.44) and obtain the statistical distribution of the estimates obtained through (24.52) and (24.53). The channel gains  $h_\gamma$  and signal samples  $s_k$  are modeled as zero-mean complex-valued Gaussian and independent random variables with variance of unity. The samples  $v_{y,k}$  are also modeled as zero-mean complex-valued Gaussian and independent random variables with variance of  $\sigma_v^2 = 0.01$ . We choose  $K = 8$  and consider choices of  $\Gamma = 10, 20, 50$ , and  $100$ . For each choice of  $\Gamma$ , we obtain the means of the estimated signal power,  $P_{\text{sig}}$ , and the estimated noise power,  $P_{\text{noise}}$ , based on 100,000 repetitions of the experiment. The standard deviation (std) of the estimates, normalized with respect to the estimated means are also evaluated. The results presented in Table 24.2 show that for smaller values of  $\Gamma$ , the mean of  $P_{\text{noise}}$  is biased; it deviates from  $\sigma_v^2 = 0.01$  by a factor of  $1/\Gamma$ . However, the mean of  $P_{\text{sig}}$  remains approximately equal to its expected value,  $\sigma_s^2 = 1$ , independent of  $\Gamma$ . On the other hand, the accuracy of the estimates of  $P_{\text{noise}}$  are better than those of the estimates of  $P_{\text{sig}}$  and show more improvement as  $\Gamma$  increases.

## 24.7 DISCUSSION

In this chapter, we presented an overview of three possible choices for spectrum sensing in cognitive radios; namely, periodogram spectral estimator (PSE), multitaper spectral estimator (MTSE), and filter bank spectral estimator (FBSE). We noted that the three methods operate on the same principle. The band of interest is divided into a number of subbands and the signal energy in each subband is measured as an estimate of the power spectral density (PSD) over the subband. All three methods can also be implemented efficiently using polyphase filter bank structures. The difference lies in the prototype filters that are used by different methods.

The PSE uses a prototype filter whose length is equal to the number of subbands. Here, the prototype filter is essentially a window function that is applied to the signal samples before passing them through a Fourier transformer, an FFT block. The application of window is equivalent to using a prototype filter with a controlled stopband behavior. Different choices of window functions were reviewed. Moreover, we introduced the prolate sequences as a class of optimal window functions.

The MTSE was presented as an extension to PSE with multiple window functions/prototype filters that are optimally designed following the same principles as the prolate window function; namely, the optimization is set to minimize the stopband energy of the prototype filters. To allow selection of multiple prototype filters with acceptable responses, the MTSE uses prototype filters with a length longer than that of the PSE method. The use of multiple prototype filters, thus, multiple filter banks, provides multiple signals for each subband whose energy can be averaged to reduce the variance of the PSD estimates. Also, the use of prototype filters with longer length results in improved filter bank frequency responses, thus, less bias/spectral leakage in the PSD estimates.

The FBSE was presented as a low-cost (almost a no cost) candidate in cases where filter banks are used for communications among cognitive radio nodes. Similar to the MTSE, the FBSE uses prototype filters with relatively long length; hence, it also provides a great deal of flexibility in controlling the stopband responses and, therefore, can accommodate a wide spectral dynamic range; see Figures 24.10 and 24.16. In order to improve on the PSD estimates, it was proposed that the successive signal samples across time are squared and averaged. This is similar to the commonly used weighted overlapped segment averaging (WOSA) in the PSE; see Section 24.3.4.

In terms of spectral leakage, the MTSE and FBSE can be designed to perform about the same, albeit the MTSE uses a smaller number of samples because it uses a set of prototype filters in parallel. To provide some figures for the latter, consider the cases whose results were presented in Figures 24.10 and 24.16. In the case of MTSE, each snapshot is based on  $128 \times 8 = 1024$  signal samples. On the other hand, each snapshot of FBSE requires  $M + (K - 1)N = 1537 + 7 \times 256 = 3329$  signal samples. This is a threefold increase. To better understand these numbers and find out how significant they may be in the application of cognitive radios, consider a case where the cognitive radio examines a frequency band of 20 MHz and for that we have taken samples at a rate of 20 megasamples per second. With this choice of sampling rate, each snapshot of the MTSE involves a time window of  $1024 \times (1/20) \mu\text{s} = 51.2 \mu\text{s}$ . This value increases to  $166 \mu\text{s}$  in the case of FBSE. Some recent studies on cognitive radios suggest a sensing iteration of 5–10 ms, for example [22]. Noting that this is almost two orders of magnitude greater than the above snapshot periods, one may argue that the excess time required by the FBSE (compared to that of the MTSE) should not be a concern in the particular application of cognitive radios.

## APPENDIX A: EFFECTIVE DEGREE OF FREEDOM

In probability theory and statistics, if  $x_0, x_1, \dots, x_{K-1}$  are a set of zero-mean unit-variance independent Gaussian variables, then

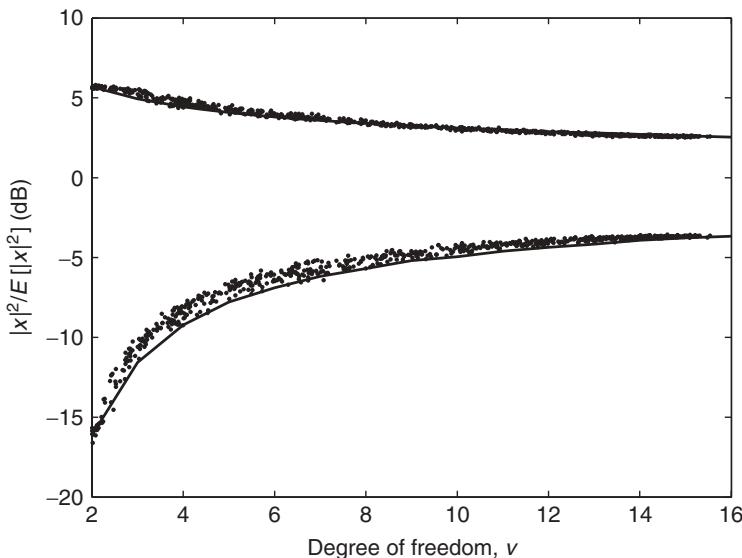
$$|x|^2 = \sum_{k=0}^{K-1} x_k^2 \quad (24.54)$$

has a distribution that is called chi-square with  $K$  degrees of freedom [20]. Obviously, in this definition, the degrees of freedom  $K$  is always an integer. When  $x_k$ 's are complex Gaussian variables, in (24.54),  $x_k^2$  is replaced by  $|x_k|^2$ , and  $|x|^2$  will have a chi-square distribution with  $2K$  degrees of freedom. The EDF  $v$ , which is defined in (24.42), extends this definition to noninteger numbers, which we define (following Thomson [5]) for the degrees of freedom of the random variable

$$|x|^2 = \sum_{k=0}^{K-1} \lambda_k x_k^2, \quad (24.55)$$

where  $\lambda_k$  are a set of fixed, but randomly selected, positive coefficients.

To evaluate how meaningful the definition (24.42) is, we perform the following test. In Figure 24.18 we have presented the boundary lines that a random variable  $|x|^2$  with a chi-square distribution of various degrees of freedom falls in between with a chance of 95%. The 0-dB level indicates the mean value of  $|x|^2$ . The lower plot shows the level below which 2.5% of the values of  $|x|^2$  fall. The upper plot shows the level above which 2.5% of the values of  $|x|^2$  fall. The full-line plots are obtained by linear interpolation between the points that correspond to the cases where  $|x|^2$  is a true chi-square with an integer degrees of freedom. Each pair of dots (for a fixed  $v$ ) are the results obtained by first selecting a set of randomly generated  $\lambda_k$ 's, then using them to generate 10,000 samples of  $|x|^2$  according to (24.55), and finally finding the boundaries of the 95% confidence interval, using histograms. Since these results match relatively well with the full-line plots, we argue that the interpretation of  $|x|^2 = \sum_k \lambda_k |x_k|^2$  as a random variable with a chi-square distribution whose EDF is given by (24.42) is a good engineering approximation and, thus, can be reliably used to obtain the estimates of the confidence intervals of the spectral estimates.



**Figure 24.18** The 95% boundary limits of a chi-square distribution with various degrees of freedom.

## APPENDIX B: EXPLANATION TO THE RESULTS OF TABLE 24.1

The results presented in Table 24.1 indicate that for  $L \leq N$  and a fixed window length  $W$ , the EDF  $v$  remains almost a constant, independent of  $L$ . The goal of this appendix is to explain this observation.

Using the minimax theorem that was introduced in Section 24.4.1, one can show that the largest eigenvalue of the correlation matrix  $\mathbf{R}$  of a random process  $x(n)$  is upper bounded by the maximum of the spectral density of  $x(n)$ , [18, 20], namely  $\lambda_{\max} \leq \max S_{xx}(f)$ . Moreover this bound becomes tight as the size of  $\mathbf{R}$  increases. Hence, the approximation  $\lambda_{\max} \approx \max S_{xx}(f)$  may be used. On the other hand, if  $x(n)$  is obtained by sampling a continuous-time signal, say,  $x(n) = x_a(nT)$ , where  $T$  is the sampling period, the magnitude of  $S_{xx}(f)$ , for any  $f$ , and, thus, its maximum increases proportional to  $1/T$ . Also, we note that for the results presented in Table 24.1, the underlying correlation matrix is  $\mathbf{A}$  (one can remove the factor  $S_{xx}(f_i)$  without affecting the results) and  $T$  is proportional to the size of the sample spacing interval  $L$ . In addition, noting that for  $L = N$ ,  $\mathbf{A} = \mathbf{I}$  and, hence,  $\lambda_{\max} = 1 (= N/N)$ , one may draw the conclusion that

$$\lambda_{\max} \approx \frac{N}{L}. \quad (24.56)$$

On the other hand, since  $\mathbf{A}$  is a matrix of size  $(W/L) \times (W/L)$  and has diagonal elements of 1, we get

$$\sum_i \lambda_i = \text{tr}[\mathbf{A}] = \frac{W}{L}. \quad (24.57)$$

Substituting (24.56) and (24.57) in (24.42), we obtain

$$v \approx \frac{2W}{N}, \quad (24.58)$$

which is independent of  $L$ .

## REFERENCES

1. R. W. Brodersen, A. Wolisz, D. Cabric, S. M. Mishra, and D. Willkomm, “CORVUS: A cognitive radio approach for usage of virtual unlicensed spectrum,” White paper, Berkeley, July 29, 2004, available: [http://bwrc.eecs.berkeley.edu/Research/MCMA/CR\\_White\\_paper\\_final1.pdf](http://bwrc.eecs.berkeley.edu/Research/MCMA/CR_White_paper_final1.pdf).
2. Federal Communications Commission, “Spectrum policy task force,” Report ET Docket no. 02-135, Nov. 2002.
3. J. S. Lim and A. V. Oppenheim (Eds.), *Advanced Topics in Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1988.
4. S. Haykin, “Cognitive radio: Brain-empowered wireless communications,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 3, pp. 201–220, Feb. 2005.
5. Thomson, D. J., “Spectrum estimation and harmonic analysis,” *Proc. IEEE*, vol. 70, no. 9, pp. 1055–1096, Sept. 1982.
6. B. Farhang-Boroujeny, “Filter bank spectrum sensing for cognitive radios,” *IEEE Trans. Signal Proc.*, vol. 56, no 5, May 2008, pp. 1801–1811.

7. S. Kay, *Modern Spectral Estimation: Theory and Application*, Englewood Cliffs, NJ: Prentice-Hall, 1987.
8. A. N. Mody, S. R. Blatt, D. G. Mills, T. P. McElwain, B. B. Thammakhoune, J. D. Niedzwiecki, M. J. Sherman, C. S. Myers, and Fiore, "Recent advances in cognitive communications," *IEEE Commun. Mag.*, vol. 45, no. 10, Oct. 2007, pp. 54–61.
9. J. Lunden and V. Koivunen, "Automatic radar waveform recognition," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 1, June 2007, pp. 124–136.
10. P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, NJ: Prentice Hall, 1993.
11. P. D. Welch, "The use of FFT for the estimation of power spectra: A method based on time averaging over short modified periodograms," *IEEE Trans. Audio Electroacoust.*, vol. AU-15, no. 2, pp. 70–73, June 1967.
12. A. H. Nuttall and G. C. Carter, "Spectral estimation using combined time and lag weighting," *IEEE Proc.*, vol. 70, no. 9, pp. 115–1125, Sept. 1982.
13. D. Slepian and H. O. Pollak, "Prolate spheroidal wave functions, Fourier analysis and uncertainty—I," *Bell Syst. Tech. J.*, vol. 40, pp. 43–64, 1961.
14. D. Slepian, "Prolate spheroidal wave functions, Fourier analysis and uncertainty—IV," *Bell Syst. Tech. J.*, vol. 43, pp. 3009–3057, 1964.
15. D. Slepian, "On bandwidth," *Proc. IEEE*, vol. 64, pp. 292–300, 1976.
16. D. Slepian, "Prolate spheroidal wave functions, Fourier analysis and uncertainty—V: The discrete case," *Bell Syst. Tech. J.*, vol. 57, pp. 1371–1429, 1978.
17. S. Haykin, *Adaptive Filter Theory*, 4th ed., Upper Saddle River, NJ: Prentice Hall, 2001.
18. B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*, Chichester, England: Wiley, 1998.
19. B. Farhang-Boroujeny and S. Gazor, "Selection of orthonormal transforms for improving performance of transform domain normalized LMS algorithm," *IEE Proc. F Commun. Radar Signal Process.*, vol. 139, no. 5, pp. 327–335, Oct. 1992.
20. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed., New York: McGraw-Hill, 1991.
21. A. Drosopoulos and S. Haykin, "Angle-of-arrival astimation in the presence of multipath," in *Adaptive Radar Signal Processing*, S. Haykin (Ed.), Hoboken, NJ: Wiley, 2007.
22. T. A. Weiss and F. K. Jondral, "Spectrum pooling: An innovative strategy for the enhancement of spectrum efficiency," *IEEE Commun. Mag.*, vol. 42, no. 3, pp. S8–S14, Mar. 2004.
23. The IEEE 802 LAN/MAN Standards Committee, 802.22 WG on Wireless Regional Area Networks (WRANs), available: <http://www.ieee802.org/22/>.
24. P. Amini, R. Kempfer, R. R. Chen, L. Lin, and B. Farhang-Boroujeny, "Filter bank multitone: A physical layer candidate for cognitive radios," paper presented at the Software Defined Radio Technical Conference, SDR 2005, Orange County, CA, Nov. 14–18, 2005.
25. P. Amini, R. Kempfer, and B. Farhang-Boroujeny, "A comparison of alternative filterbank multicarrier methods in cognitive radios," paper presented at the Software Defined Radio Technical Conference, SDR 2006, Orlando, FL, Nov. 13–17, 2006.
26. B. Farhang-Boroujeny and R. Kempfer, "Multicarrier communication techniques for spectrum sensing and communication in cognitive radios," *IEEE Commun. Mag.*, Special Issue on Cognitive Radios for Dynamic Spectrum Access, vol. 46, no 4, Apr. 2008, pp. 80–85.
27. R. W. Chang, "High-speed multichannel data transmission with bandlimited orthogonal signals," *Bell Syst. Tech. J.*, vol. 45, pp. 1775–1796, Dec. 1966.
28. B. R. Saltzberg, "Performance of an efficient parallel data transmission system," *IEEE Trans. Commun. Tech.*, vol. 15, no. 6, pp. 805–811, Dec. 1967.

29. B. Hirosaki, "An orthogonally multiplexed QAM system using the discrete Fourier transform," *IEEE Trans. Commun.*, vol. 29, no. 7, pp. 982–989, July 1981.
30. S. D. Sandberg and M. A. Tzannes, "Overlapped discrete multitone modulation for high speed copper wire communications," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 9, pp. 1571–1585, Dec. 1995.
31. B. Farhang-Boroujeny, "Multicarrier modulation with blind detection capability using cosine modulated filter banks," *IEEE Trans. Commun.*, vol. 51, no. 12, pp. 2057–2070, Dec. 2003.
32. L. Lin and B. Farhang-Boroujeny, "Cosine modulated multitone for very high-speed digital subscriber lines," *EURASIP J. Appl. Signal Process.*, vol. 2006, Article ID 19329, 2006.
33. G. Cherubini, E. Eleftheriou, S. Olcer, and J. M. Cioffi, "Filter bank modulation techniques for very high speed digital subscriber lines," *IEEE Commun. Mag.*, vol. 38, no. 5, pp. 98–104, May 2000.
34. B. Farhang-Boroujeny, *Signal Processing Techniques for Software Radios*, Morrisville, NC: Lulu Publishing House, 2008.
35. B. Farhang-Boroujeny, "A universal square-root Nyquist (M) filter design for digital communication systems," in *Proceedings of Software Defined Radio Technical Conference*, SDR 2006, Orlando, FL, Nov. 13–17, 2006.
36. V. J. Mathews, D. H. Youn, and N. Ahmed, "A unified approach to nonparametric spectrum estimation algorithms," *IEEE Trans. Acoust. Speech Signal Proc.*, vol. ASSP-35, no. 3, pp. 338–349, Mar. 1987.
37. T. Thong, "Practical considerations for a continuous time digital spectrum analyzer," in *Proc. ISCAS'89*, Vol. 2 pp. 1047–1050
38. T. P. Bronez, "On the performance advantage of multitaper spectral analysis," *IEEE Trans. Signal Process.*, vol. 40, no. 12, pp. 2941–2946, Dec. 1992.
39. A. T. Walden, E. McCoy, and D. B. Percival, "The variance of multitaper spectrum estimates for real Gaussian processes," *IEEE Trans. Signal Process.*, vol. 42, no. 2, pp. 479–482, Feb. 1994.
40. P. Stoica and T. Sundin, "On nonparametric spectral estimation," *Circuits Syst. Signal Process.*, vol. 18, no. 2, pp. 169–181, 1999.
41. J. W. Pitton, "Time-frequency spectrum estimation: An adaptive multitaper method," in *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, Pittsburgh, PA, Oct. 6–9, 1998, pp. 665–668.
42. D. G. Mestdagh, M. R. Isaksson, and P. Odling, "Zipper VDSL: A solution for robust duplex communication over telephone lines," *IEEE Commun. Mag.*, vol. 38, no. 5, pp. 90–96, May 2000.
43. S. Brandes, I. Cosovic, and M. Schnell, "Reduction of out-of-band radiation in OFDM based overlay systems," paper presented at the First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2005, Nov. 8–11, 2005, pp. 662–665.



## CHAPTER 25

---

# Nonparametric Techniques for Pedestrian Tracking in Wireless Local Area Networks

Azadeh Kushki and Konstantinos N. Plataniotis

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering University of Toronto, Toronto, Ontario, Canada

### 25.1 INTRODUCTION

Over the past five decades, the problem of target tracking has been studied extensively in military and civilian applications. More recently, advances in wireless communication technology have enabled user mobility within wireless networks, resulting in the dependency of users' communication, resource, and information needs on their physical location. This has sparked a new need for effective positioning and tracking of wireless terminals to allow the delivery of location-based services catered to changing user contexts. Well-known examples of positioning systems are the Global Positioning System (GPS) and cellular network-based systems [1] used for navigation and location-based emergency and commercial services.

The ubiquity of indoor wireless networks has also inspired location awareness in indoor environments in applications such as location-based network access, management, and security, automatic resource assignment, health monitoring, guidance of persons with disabilities, location-sensitive information delivery, and context awareness. Unfortunately, the positioning accuracy provided by existing cellular-based methods is not sufficient for such applications and coverage of the GPS is limited in indoor environments. In this light, a plethora of systems have been proposed to estimate and track location of people specifically in indoor environments. Such systems rely on input from various types of sensors, such as proximity sensors, radio frequency (RF) and ultrasound badges [2], visual sensors, and wireless local area network (WLAN) radio signals [3, 4] to carry out the estimation.

Wireless LAN positioning refers to the process of determining the physical coordinates of mobile network devices, such as laptops or personal digital assistants, using a WLAN infrastructure such as IEEE 802.11b/g. Among the above methods, WLAN positioning and tracking is especially favored for three reasons [5, 6]:

- *Cost Effectiveness* WLAN positioning is carried out by exploiting the dependency between the location of a mobile device and characteristics of signals

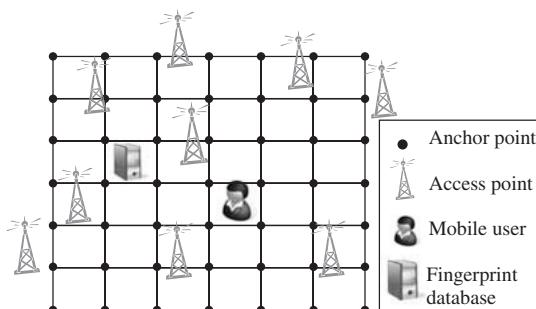
transmitted between the device and a set of physically distributed WLAN access points (APs). Signal characteristics used for positioning include time of arrival (ToA), time difference of arrival (TDoA), angle of arrival (AoA), and received signal strength (RSS). RSS is the feature of choice in WLAN positioning systems as it can be obtained directly from network interface cards (NIC), which are available on most handheld computers. This allows the implementation of positioning algorithms on top of existing WLAN infrastructures without the need for any additional hardware. The wide availability and ubiquitous coverage provided by WLANs make this type of positioning a particularly cost-effective solution for offering value-added location-specific services in commercial and residential indoor environments.

- *Scalability* WLAN sensors (access points) are ubiquitously deployed in commercial and residential environments. The wide availability of access points and the cost-effectiveness of positioning make WLAN positioning highly scalable.
- *Consensual Sensing and Tracking* Since indoor positioning involves direct monitoring of humans, it is imperative that this technology does not lead to infringement of user privacy rights [7]. WLAN positioning is especially favored in this regard since all sensing operations require the cooperation of the mobile user. Moreover, in terminal-based positioning, users must initiate positioning operations (e.g., by starting a software program on the device). They can also choose to terminate positioning services by shutting off wireless communications with the infrastructure. Lastly, since positioning operations can be fully implemented on mobile clients, no invasive sensing, processing, and storage is required in WLAN positioning.

Figure 25.1 depicts a typical WLAN positioning setup containing  $L$  WLAN access points. Since these APs may belong to different networks, their exact coordinates are generally unknown to the positioning system, rendering ranging-based positioning techniques inappropriate.

While the area of target tracking in classical applications is well studied and quite mature, the emerging area of LBS and indoor positioning in WLAN settings presents various novel and unique challenges. Specifically, two key challenges must be addressed.

- *Unknown RSS Position Dependency* WLAN positioning systems rely on the dependency of RSS on location of the mobile device. Characterization of the RSS position dependency, however, is a difficult task due to the complexity of the



**Figure 25.1** Problem setup.

indoor radio channel. As a result, WLAN positioning systems characterize the RSS position relationship implicitly, through the use of a method known as *fingerprinting* or *scene matching*. In such an approach, training RSS measurements are collected at a set of  $N$  anchor points with known coordinates. During the online operation of the system, the incoming readings from the mobile are matched against these fingerprints to obtain a position estimate. Therefore, in contrast to classical target tracking problems, an explicit form relating position to RSS measurements is unknown in the WLAN problem. Instead, training RSS values, collected at a set of spatially distributed anchor points, implicitly characterize the RSS position dependency [6, 8]. This representation renders the commonly used Kalman filter and its variants inapplicable to the WLAN tracking problem. Consequently, new developments in the area of stochastic filtering are needed to handle the unique characteristics of the WLAN tracking problem.

- *Unpredictable RSS Variations* Given the above characterization of the RSS position dependency, the second technical challenge in WLAN positioning is estimation in the presence of unpredictable variations in RSS measurements. The variations occur due to radio channel impediments such as interference and shadowing by moving objects [9] as well as movement of the wireless device. This unpredictable nature of the indoor propagation environment causes operating conditions to deviate from those learned based on location fingerprinting. To deal with such uncertainties, intelligent sensor selection and scene analysis are needed to anticipate and adapt to environmental conditions.

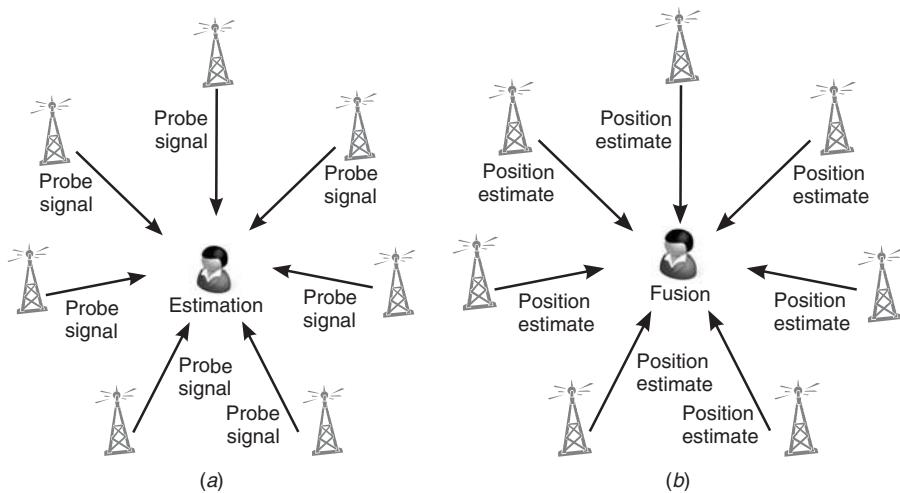
In this chapter, nonparametric techniques for addressing the aforementioned challenges in WLAN tracking are discussed. It must be noted here that additional challenges in fingerprinting include the dependence of RSS measurements on the particular network card used, orientation of the receiver, and calibration drift occurring over time. These issues are out of the scope of this chapter and will not be discussed further. The reader is referred to [6, 8, 10, 11] for further details on these topics.

The rest of this chapter is organized as follows. Section 25.2 discusses two architectures of positioning and tracking with measurements from multiple access points; Section 25.3 presents the methods of characterizing the RSS position dependency; Section 25.4 outlines nonparametric position estimation methods while Section 25.5 discusses Bayesian tracking techniques in WLANs. In Section 25.6, we present the design of a cognitive dynamic tracking system, and Section 25.7 illustrates the concepts through a real example. Finally, Section 25.8 concludes the chapter and provides directions for future work.

## 25.2 WLAN POSITIONING ARCHITECTURES

A WLAN positioning system relies on a set of measurements from spatially distributed wireless APs. Depending on how these sensor readings are used, two types of architectures have been proposed for positioning.

1. *Centralized Positioning (Measurement Fusion)* As shown in Figure 25.2a, in this type of architecture sensors (APs) act as simple observers that provide the mobile client with probe signals. The mobile client serves as the central node and uses RSS measurements obtained from the probe signals to carry out positioning



**Figure 25.2** Possible architectures for positioning in WLAN: (a) Centralized positioning (measurement fusion) and (b) decentralized positioning (estimation fusion).

operations. Centralized fusion systems generally have the best performance in terms of accuracy as they can use the complete observation set [12]. This set can be used for outlier filtering and faulty sensor detection [13]. Furthermore, communication needs are minimal for this type of architecture as position estimates and measurements need not be sent over wireless links. These advantages come at the cost of increased processing demand on the mobile client.

2. *Decentralized Positioning (Estimation Fusion)* This architecture is shown in Figure 25.2b. Here each AP measures the RSS from the mobile client and from a position estimate based on this local RSS measurements. While positioning in two dimensions generally requires measurements from three or more APs, it was shown in [14] that the asymmetry of the propagation environment allows for estimation using a single AP. This means that no communication between APs is needed to form the location position estimates. The local estimates are then sent to the mobile client for fusion. An advantage of such a hierarchical scheme is the reduced computations on the power-limited mobile device. Moreover, each local estimate can be augmented with a quality measure that can aid in sensor selection based on reliability and fidelity of local estimates. Lastly, this architecture provides a scalable positioning solution in environments with a large number of APs. The disadvantage of this approach is the need for communication of local AP estimates to the mobile device or central server.

We limit the scope of this chapter to centralized schemes and refer the reader to [14] for further reading on the decentralized formulation.

### 25.3 SIGNAL MODELS

Wireless LAN positioning systems rely on the dependency of RSS on location of the mobile device. Unfortunately, an explicit functional RSS position relationship is not

available in indoor environments because of the complexity of the indoor radio channel. This is due to severe multipath and shadowing conditions and non-line-of-sight propagation caused by the presence of walls, humans, and other rigid objects. Moreover, the IEEE 802.11 WLAN operates on the license-free band of frequency of 2.4 GHz, which is the same as cordless phones, microwaves, BlueTooth devices, and the resonance frequency of water. This leads to time-varying interference from such devices and signal absorption by the human body, further complicating the propagation environment. This type of environment gives rise to a many-to-many correspondence between RSS and spatial positions. To make matters worse, WLAN infrastructures are highly dynamic as APs<sup>1</sup> can easily be moved or discarded, in contrast to their counterparts in cellular systems that generally remain intact for long periods of time.

Existing WLAN positioning techniques use two approaches to modeling the RSS position dependency: radio propagation modeling and fingerprinting.

### 25.3.1 Radio Propagation Modeling

This approach entails the assumption of a prior theoretical model for the RSS position relationship and estimation of model parameters based on training data. Given an RSS measurement and this model, the distances from the mobile device to at least three APs are determined, and trilateration is used to obtain the position of the device.

As the radio signal travels through an ideal propagation medium (i.e., free space), the received signal power falls off inversely proportional to the square of the distance. Thus, given measurements of transmitted and received powers, the distance between the transmitter and the mobile device can be determined. In real environments, however, in addition to power dissipation (path loss), reflection, absorption, and scattering caused by obstacles between transmitter and receiver affect the radio signal. Due to uncertainties in the nature and location of the blocking objects, the effects of shadowing are often characterized statistically. Specifically, a log-normal distribution is assumed for the ratio of transmit-to-receive power. The combined effects of the path loss and shadowing can be expressed by the simplified model below [9]:

$$P_r(\text{dB}) = P_t(\text{dB}) + 10 \log_{10} K - 10\gamma \log_{10} \left( \frac{d}{d_0} \right) - \psi(\text{dB}). \quad (25.1)$$

In Eq. (25.1),  $P_r$  and  $P_t$  are the received and transmitted powers,  $K$  is constant relating to antenna and channel characteristics,  $d_0$  is a reference distance for antenna far-field, and  $\gamma$  is the path loss exponent and can be determined based on empirical measurements in a given environment [9]. Finally,  $\psi \sim \mathcal{N}(0, \sigma_\psi^2)$  reflects the effects of log-normal shadowing in the model.

In indoor areas, materials for walls and floors, number of floors, layout of rooms, location of obstructing objects, and size of each room have a significant effect on path loss. This makes it difficult to find a model applicable to general environments. Other limitations of model-based approaches include their dependence on prior topological information, assumption of isotropic RSS contours, and invariance to receiver orientation [10]. More importantly, model-based approaches assume exact knowledge of AP locations. This is impractical in large and ubiquitous deployments where multiple wireless networks, run by different operators, coexist.

<sup>1</sup>The term access point (AP) is used to refer to a WLAN base station in this chapter.

### 25.3.2 Fingerprinting-Based Methods

In order to overcome the above limitations of the propagation model, the second class of WLAN positioning systems use a training-based method known as *location fingerprinting*. These methods use training RSS measurements at spatially distributed anchor points with known coordinates to construct a *radio map* that implicitly characterizes the RSS position relationship. A radio map is a set  $\mathcal{R} = \{(\mathbf{p}_i, \mathbf{F}(\mathbf{p}_i)) | i = 1, \dots, N\}$  where  $\mathbf{p}_i = [p_x \ p_y]^T$  is the Cartesian coordinates of the  $i$ th anchor point,  $\mathbf{F}(\mathbf{p}_i) = [\mathbf{r}_i(1), \dots, \mathbf{r}_i(T)]$  is a fingerprint matrix [6], and  $T$  is the number of training samples collected at each anchor point. The vector  $\mathbf{r}_i(t) = [r_i^1(t), \dots, r_i^L(t)]$  contains the RSS measurements from  $L$  APs at time  $t$  at anchor point  $\mathbf{p}_i$ .

Traditionally, radio map construction has been performed prior to the operation of the positioning system. More recently, online determination of the training values has been proposed in [8, 15] to improve the reliability of fingerprints and promote resiliency to time variations in indoor radio environments.

Characterization of the indoor propagation environments through the radio map motivates the use of nonparametric techniques in positioning. The following sections discuss these tools in detail.

## 25.4 ZERO-MEMORY POSITIONING

Denote the true and estimated positions of the pedestrian carrying the mobile device at time  $k$  as  $\mathbf{p}(k)$  and  $\hat{\mathbf{p}}(k)$ . The objective of the WLAN tracking system is to determine a sequence of estimates of position over time,  $\hat{\mathbf{p}}(1), \dots, \hat{\mathbf{p}}(k)$ , given a sequence of RSS measurements  $\mathbf{R}(k) = \{\mathbf{r}(0), \dots, \mathbf{r}(k)\}$ .

Most existing research in WLAN positioning [4, 6, 8, 10, 16–19] has largely focused on the problem of *static* positioning where users are assumed to remain stationary. We shall refer to these methods as *zero-memory estimator* since only  $\mathbf{r}(k)$ , the RSS measurement at time  $k$ , is used to estimate the mobile position at time  $k$ .

Zero-memory positioning entails matching the observed RSS measurement to the training values in the radio map. While this matching can be performed using classification and pattern recognition methods, this chapter focuses on Bayesian estimation techniques. In particular, we consider minimum mean-squared error (MMSE) techniques. The MMSE position estimate minimizes the expected value of  $l_2$  norm of positioning error, that is,

$$\hat{\mathbf{p}}(k) = \arg \min_{\hat{\mathbf{p}}(k)} \mathbb{E}\{(\mathbf{p}(k) - \hat{\mathbf{p}}(k))^T (\mathbf{p}(k) - \hat{\mathbf{p}}(k))\}. \quad (25.2)$$

The MMSE position estimate is the expected value of the position conditioned on the current observation [6]:

$$\hat{\mathbf{p}}(k) = \mathbb{E}\{\mathbf{p}(k) | \mathbf{r}(k)\} = \int \mathbf{p}(k) f(\mathbf{p}(k) | \mathbf{r}(k)) d\mathbf{p}(k) \quad (25.3)$$

where  $\mathbf{p}_i$  is the  $i$ th anchor point. The key challenge in this estimation problem is that a parametric form for the density  $f(\mathbf{p}(k) | \mathbf{r}(k))$  is unavailable. Therefore, nonparametric techniques are used to estimate this density from the location fingerprints in the radio map. In contrast to parametric approaches, nonparametric methods do not assume a prior statistical distribution and rely solely on the structure present in the data to

estimate a density. Among the nonparametric density estimation techniques, the kernel density estimator has proved to be effective in non-line-of-sight propagation conditions in both cellular networks [1] as well as WLANs [6, 20, 21] and will be used to develop tools for WLAN positioning and tracking in the rest of this chapter.

We begin by noting that

$$f(\mathbf{p}(k)|\mathbf{r}(k)) = \frac{f(\mathbf{p}(k), \mathbf{r}(k))}{f(\mathbf{r}(k))} = \frac{f(\mathbf{p}(k), \mathbf{r}(k))}{\int f(\mathbf{p}(k), \mathbf{r}(k))d\mathbf{p}(k)}. \quad (25.4)$$

The kernel density estimate uses a set of training pairs  $\{(\mathbf{p}_i, \bar{\mathbf{r}}^i) | i = 1, \dots, N\}$ . The second element in the pair,  $\bar{\mathbf{r}}^i$  is an RSS representative value for each anchor point and is extracted from the fingerprint matrix. While various methods for extracting this value have been proposed, in the chapter we use the sample mean of the training record at each anchor point  $\bar{\mathbf{r}}_i = (1/T) \sum_{t=1}^T \mathbf{r}_i(t)$  for simplicity. The index  $t$  is used to differentiate between RSS samples collected during the training phase and the measurements observed in online operation of the system.

Given these training pairs, the kernel density estimate of  $f(\mathbf{p}(k), \mathbf{r}(k))$  is

$$\hat{f}(\mathbf{p}(k), \mathbf{r}(k)) = \sum_{i=1}^N K\left(\frac{\mathbf{r}(k) - \bar{\mathbf{r}}^i}{\Sigma_{\mathbf{r}}}\right) K\left(\frac{\mathbf{p}(k) - \mathbf{p}^i}{\Sigma_{\mathbf{p}}}\right). \quad (25.5)$$

The kernel  $k(\cdot)$  is a nonnegative and zero-mean function such that  $\int k(x) dx = 1$ . The diagonal matrices  $\Sigma_{\mathbf{r}}$  and  $\Sigma_{\mathbf{p}}$  represent the width of the kernel function in each dimension and are determined to minimize the asymptotic mean integrated error between the estimated and true densities. For a Gaussian kernel, a quick method for computing the diagonal elements of these matrices is given in [22] as

$$\sigma^* = \left(\frac{4}{2d+1}\right)^{1/d+4} \hat{\sigma} n^{-1/d+4}. \quad (25.6)$$

Here  $d$  is the dimension of the random vector and  $\hat{\sigma}^2 = (1/d) \sum_{l=1}^d \sigma_l^2$  is the average of marginal variances in each dimension. Note that we assumed that RSS values from different APs are conditionally independent.

Substituting the estimate of (25.5) into (25.4) and using a Gaussian kernel, we obtain the following result:

$$\hat{f}(\mathbf{p}(k)|\mathbf{r}(k)) = \frac{\hat{f}(\mathbf{p}(k), \mathbf{r}(k))}{\hat{f}(\mathbf{r}(k))} \quad (25.7)$$

$$= \frac{\sum_{i=1}^N \mathcal{N}(\mathbf{r}(k); \bar{\mathbf{r}}_i, \Sigma_{\mathbf{r}}) \mathcal{N}(\mathbf{p}(k); \mathbf{p}_i, \Sigma_{\mathbf{p}})}{\sum_{i=1}^N \mathcal{N}(\mathbf{r}(k); \bar{\mathbf{r}}_i, \Sigma_{\mathbf{r}})} \quad (25.8)$$

$$= \sum_{i=1}^N w_i(\mathbf{r}(k)) \mathcal{N}(\mathbf{p}(k); \mathbf{p}_i, \Sigma_{\mathbf{p}}), \quad (25.9)$$

where

$$w_i(\mathbf{r}(k)) = \frac{\mathcal{N}(\mathbf{r}(k); \bar{\mathbf{r}}_i, \Sigma_{\mathbf{r}})}{\sum_{i=1}^N \mathcal{N}(\mathbf{r}(k); \bar{\mathbf{r}}_i, \Sigma_{\mathbf{r}})}. \quad (25.10)$$

The MMSE estimate and its associated covariance are the first two moments of the Gaussian mixture in (25.7):  $\hat{\mathbf{p}}(k) = E\{\mathbf{p}(k)|\mathbf{r}(k)\}$  and  $\mathbf{P}(k) = E\{(\mathbf{p}(k) - \hat{\mathbf{p}}(k))(\mathbf{p}(k) - \hat{\mathbf{p}}(k))^T|\mathbf{r}(k)\}$ . These moments are given as [23]

$$\hat{\mathbf{p}}_r(k) = \sum_{i=1}^N w_i(\mathbf{r}(k)) \mathbf{p}_i, \quad (25.11)$$

$$\hat{\mathbf{P}}_r(k) = \sum_{i=1}^N w_i(\mathbf{r}(k)) (\Sigma_{\mathbf{p}} + (\mathbf{p}_i - \hat{\mathbf{p}}(k))(\mathbf{p}_i - \hat{\mathbf{p}}(k))^T). \quad (25.12)$$

In the above formulation, the posterior distribution is determined using a Gaussian mixture obtained from a spatially distributed set of points.

## 25.5 DYNAMIC POSITIONING SYSTEMS

Recall that the objective of a WLAN tracking system is to continuously determine a sequence of minimum mean-square estimates of position over time,  $\hat{\mathbf{p}}(1), \dots, \hat{\mathbf{p}}(k)$ , given a sequence of noisy RSS measurements  $\mathbf{R}(k) = \{\mathbf{r}(0), \dots, \mathbf{r}(k)\}$ . In the previous section, we discussed a nonparametric solution for zero-memory estimation. In this section, we shall discuss the problem of recursive MMSE state estimation using Bayesian filtering. This is a challenging problem. In contrast to classical target tracking problems, an explicit *measurement equation* relating position to RSS measurements is unknown. Instead, the location fingerprints stored in the radio map implicitly characterize the RSS position dependency. This lack of an explicit relation between the mobile's position and RSS measurements renders the direct application of commonly used filters, such as the Kalman filter and its variants, impractical. In this chapter, we focus on development of state-space filters based on the nonparametric description of the measurement equation for the WLAN tracking problem.

### 25.5.1 Bayesian Tracking Problem

Central to state-space filtering is the notion of a *state vector*. This vector contains the minimal set of parameters that are needed to describe the dynamic behavior of the system [24]. In tracking problems, the state vector contains parameters related to the kinematics of the target. Denote the state vector as  $\mathbf{x}(k) = [p_x(k) v_x(k) p_y(k) v_y(k)]^T$  where  $\mathbf{p}(k) = (p_x(k), p_y(k))$  and  $(v_x(k), v_y(k))$  are the position and velocity of the pedestrian carrying the mobile at time  $k$  in two-dimensional (2D) Cartesian coordinates. The position estimate is related to the state estimate as

$$\hat{\mathbf{p}}(k|k) = \mathbf{A}\hat{\mathbf{x}}(k|k) \quad (25.13)$$

where

$$\mathbf{A} \triangleq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (25.14)$$

Let  $\mathbf{R}(k) = \{\mathbf{r}(0), \dots, \mathbf{r}(k)\}$  be the observation record up to time  $k$ . In the statical sense, the posterior density  $f(\mathbf{x}(k)|\mathcal{R}(k))$  summarizes all the information about the

past and current states of the system and can be used to obtain an estimate of  $\mathbf{x}(k)$  given an objective function. For example, MMSE and maximum a posteriori (MAP) estimates of the state are generated as  $\hat{\mathbf{x}}^{\text{MMSE}}(k|k) = E\{\mathbf{x}(k)|\mathcal{R}(k)\}$  and  $\hat{\mathbf{x}}^{\text{MAP}}(k|k) = \max_{\mathbf{x}(k)} f(\mathbf{x}(k)|\mathcal{R}(k))$ , respectively. For the rest of this chapter, MMSE estimation of position is considered as positioning is carried in Cartesian coordinates. This estimate is denoted as  $\hat{\mathbf{x}}(k|j)$  from hereon, where the notation  $\hat{\mathbf{x}}(k|j)$  denotes the state estimate at time  $k$  given the observation record  $\mathcal{R}_j$ .

The MMSE estimate of the  $\mathbf{x}(k)$  is the conditional expectation computed from the posterior density

$$\hat{\mathbf{x}}(k) = \mathbb{E}\{\mathbf{x}(k)|\mathbf{R}(k)\} = \int \mathbf{x}(k) f(\mathbf{x}(k)|\mathbf{R}(k)) d\mathbf{x}(k).$$

The Bayesian filtering framework provides the means to calculate the posterior density in a recursive manner:

$$f(\mathbf{x}(k)|\mathbf{R}(k)) = f(\mathbf{x}(k)|\mathbf{r}(k), \mathbf{R}(k-1)) \quad (25.15)$$

$$= \frac{f(\mathbf{r}(k)|\mathbf{x}(k), \mathbf{R}(k-1))}{f(\mathbf{r}(k)|\mathbf{R}(k-1))} f(\mathbf{x}(k)|\mathbf{R}(k-1)) \quad (25.16)$$

$$= \frac{f(\mathbf{r}(k)|\mathbf{x}(k))}{f(\mathbf{r}(k)|\mathbf{R}(k-1))} f(\mathbf{x}(k)|\mathbf{R}(k-1)). \quad (25.17)$$

We have made two assumptions in the above derivation: (1)  $\mathbf{x}(k)$  is a Markov-I process and (2) the observation  $\mathbf{r}(k)$  depends on  $\mathbf{x}(k)$  but is conditionally independent from prior observations, that is,  $f(\mathbf{r}(k)|\mathbf{x}(k), \mathcal{R}(k-1)) = f(\mathbf{r}(k)|\mathbf{x}(k))$ . The first assumption is indeed true in the WLAN problem where  $\mathbf{x}(k)$  is pedestrian motion. The second assumption, however, cannot be as trivially justified because  $\mathbf{r}(k)$  depends on environmental conditions contributing to multipath and shadowing in addition to the current position of the mobile. Despite the violation of the independence assumption, Bayesian filtering has been shown to significantly improve the accuracy of positioning.

Bayesian state-space filters recursively compute the posterior density in (25.16) using two models: a state evolution model, describing the evolution of the state over time, and a measurement model relating the measurements to the state. The state-space formulation summarizes this information in two equations:

$$\mathbf{x}(k) = m(\mathbf{x}(k-1), \mathbf{w}(k)) \quad (25.18)$$

$$\mathbf{r}(k) = h(\mathbf{x}(k), \mathbf{v}(k)) \quad (25.19)$$

where  $\mathbf{w}(k)$  and  $\mathbf{v}(k)$  are the system and measurement noise processes. These equations and known noise statistics allow the computation of a recursive estimate of the state at each time step.

At each time step, two sources of information contribute to reducing the uncertainty in the state: the dynamic model (prior knowledge about the evolution of the system) and the measurements. The information from these sources are combined to produce a state estimate in two steps. First, the dynamic model is used to generate a *prediction* of the state. This prediction is a contribution of the dynamic model to the estimate at this time step. Next, the measurement equation is used to provide the contribution of

the measurements and fuse this contribution with the state prediction. Given the initial density  $f(\mathbf{x}(0)|\mathbf{R}(0))$ , the posterior density is calculated recursively in two steps:

1. *Prediction (dynamic model):*

$$f(\mathbf{x}(k-1)|\mathbf{R}(k-1)) \rightarrow f(\mathbf{x}(k)|\mathbf{R}(k-1)). \quad (25.20)$$

2. *Update (measurement equation):*

$$f(\mathbf{x}(k)|\mathbf{R}(k-1)) \rightarrow f(\mathbf{x}(k)|\mathbf{R}(k)). \quad (25.21)$$

The prediction density can be obtained through the Chapman–Kolmogorov equation:

$$f(\mathbf{x}(k)|\mathcal{R}(k-1)) = \int f(\mathbf{x}(k)|\mathbf{x}(k-1))f(\mathbf{x}(k-1)|\mathbf{R}(k-1))d\mathbf{x}(k-1) \quad (25.22)$$

where  $f(\mathbf{x}(k)|\mathbf{x}(k-1))$  is computed from the system of Eq. (25.18). The system equation describes the kinematic characteristics of pedestrian movements. While this motion is complex in the general case, a simplified model can be obtained for the WLAN tracking problem by considering the typical movement scenarios in indoor environments. These may include, for example, going from one office to another, going to the elevators, and so on. Since these types of motion are generally constrained by the structure of hallways, we will use a simple linear motion model in the subsequent development of filters. This model is determined from its continuous time counterpart and is shown below [1]:

$$\mathbf{x}(k) = \mathbf{F}\mathbf{x}(k-1) + \mathbf{w}(k) \quad (25.23)$$

where  $\mathbf{w}(k)$  is white Gaussian noise such that  $\mathbf{w}(k) \sim \mathcal{N}(0, \mathbf{Q})$ . The system matrix is given by

$$\mathbf{F} = \begin{pmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (25.24)$$

where  $\Delta$  is the sampling period used to sample the state. The noise covariance  $\mathbf{Q}$  matrix in the discrete-time model is determined from the continuous-time motion model as described in [25] and is given by

$$\mathbf{Q} = q \begin{pmatrix} \frac{\Delta^3}{3} & \frac{\Delta^2}{2} & 0 & 0 \\ \frac{\Delta^2}{2} & \Delta & 0 & 0 \\ 0 & 0 & \frac{\Delta^3}{3} & \frac{\Delta^2}{2} \\ 0 & 0 & \frac{\Delta^2}{2} & \Delta \end{pmatrix}. \quad (25.25)$$

The choice of the parameter  $q$  is guided by noting that changes in velocity over a period of length  $\Delta$  are of the order  $\sqrt{\Delta q}$  [23].

Given this linear Gaussian model, the measurement equation is used to perform the second step in computation of the posterior density. In particular, for a linear Gaussian measurement equation the Kalman filter provides the optimal MMSE estimate of the state. For nonlinear and non-Gaussian measurement equations, the posterior density cannot be obtained in closed form, and the extended Kalman filter, unscented Kalman filter, and particle filters are commonly used as suboptimal solutions.

Due to the lack of an accurate model for RSS position dependency, the measurement equation is unknown in WLAN tracking. That is, both the function  $h(\cdot, \cdot)$  and the statistical description of the noise process  $\mathbf{v}(k)$  are unknown in (25.19). Instead, the RSS position relation is modeled implicitly through a radio map constructed from training data. The measurement update step is therefore the source of difficulty in the WLAN tracking problem.

The rest of this section reviews two filters that use the nonparametric formulation discussed in the previous section to replace the measurement equation.

### 25.5.2 The Kalman Filter

The Kalman filter assumes a Gaussian posterior density at each time step. As such, only the first two moments  $\hat{\mathbf{x}}(k|k)$  and  $\mathbf{P}(k|k) = E\{(\mathbf{x}(k|k) - \hat{\mathbf{x}}(k|k))(\mathbf{x}(k|k) - \hat{\mathbf{x}}(k|k))^T\}$  are sufficient to provide a complete representation of the posterior density.

Given  $\hat{\mathbf{x}}(k-1|k-1)$ , the estimate at time  $k-1$ , the above motion linear Gaussian model is used to generate prediction density  $f(\mathbf{x}(k)|\mathcal{R}(k-1)) = \mathcal{N}(\hat{\mathbf{x}}(k|k-1), \mathbf{P}(k|k-1))$  where

$$\hat{\mathbf{x}}(k|k-1) = \mathbf{F}\hat{\mathbf{x}}(k-1|k-1), \quad (25.26)$$

$$\mathbf{P}(k|k-1) = \mathbf{F}\mathbf{P}(k-1|k-1)\mathbf{F}' + \mathbf{Q}. \quad (25.27)$$

The Kalman filter requires a linear Gaussian measurement model. Consequently, this filter cannot be directly applied to the WLAN tracking problem as an explicit form of RSS position is not available in indoor spaces. In order to use the Kalman filter, then a preprocessor is used to generate a *synthetic measurement*, allowing the Kalman filter to abstract the details of the RSS position relationship [1, 26, 27]. Specifically, instead of using the raw RSS measurements, the zero-memory estimator of the previous section is used to generate a position estimate  $\hat{\mathbf{p}}_r(k)$  and the corresponding covariance  $\hat{\mathbf{p}}_r(k)$  from the anchor point information according to (25.11) and (25.12). Assuming the estimation noise to be Gaussian, this estimate is linearly related to the true state vector:

$$\hat{\mathbf{p}}_r(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k) \quad (25.28)$$

where  $\mathbf{v}(k) \sim \mathcal{N}(0, \mathbf{P}_r(k))$  is assumed to be zero-mean, white Gaussian, and uncorrelated with system noise  $\mathbf{w}$ . The measurement matrix  $\mathbf{H}$  is

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (25.29)$$

Given the above synthetic linear and Gaussian measurement equation, the Kalman filter can now be applied to the problem of WLAN tracking. Unfortunately, although the Kalman filter is an optimum MMSE estimator for a linear Gaussian system, it produces a suboptimal result in the above scenario. This is due to the violation of

**Assumptions:**

- State-space equation:

$$\begin{aligned}\mathbf{x}(k) &= \mathbf{F}\mathbf{x}(k-1) + \mathbf{w}(k), \\ \hat{\mathbf{p}}_r(k) &= \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k).\end{aligned}$$

**Prediction:**

$$\begin{aligned}\hat{\mathbf{x}}(k|k-1) &= \mathbf{F}\hat{\mathbf{x}}(k-1|k-1), \\ \mathbf{P}(k|k-1) &= \mathbf{F}\mathbf{P}(k-1|k-1)\mathbf{F}^T + \mathbf{Q}.\end{aligned}$$

**Update:**

$$\begin{aligned}\hat{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(\hat{\mathbf{p}}_r(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1)), \\ \mathbf{P}(k|k) &= (1 - \mathbf{K}\mathbf{H})\mathbf{P}(k|k-1)(1 - \mathbf{K}\mathbf{H})^T + \mathbf{K}\mathbf{P}_r(k)\mathbf{K}^T, \\ \mathbf{K} &= \mathbf{P}(k|k-1)\mathbf{H}[\mathbf{H}\mathbf{P}(k|k-1)\mathbf{H}^T + \mathbf{P}_r(k)]^{-1}.\end{aligned}$$

**Figure 25.3** Outline of Kalman Filter algorithm.

assumptions on the measurement noise resulting from the fact that the residuals from the zero-memory estimator are not zero mean, white, and Gaussian in practice.

### 25.5.3 The Nonparametric Information Filter

The above approach forces the Kalman structure onto the WLAN problem. In this section, the nonparametric estimation of the measurement equation is used to obtain the posterior density of (25.16) directly [28].

This is accomplished through the use of training state vectors  $\mathbf{x}_i$  generated from the anchor points  $\mathbf{p}_i = [p_x \ p_y]^T$ . Note that in addition to the  $x$  and  $y$  coordinates included with the anchor points, the state vector also requires velocity information. Therefore, in generating  $\mathbf{x}_i$  from  $\mathbf{p}_i$ , missing velocity values must be determined. The velocity (and acceleration values when applicable) can be set to zero because the radio map does not provide any knowledge regarding pedestrian motion. Consequently,  $\mathbf{x}_i = [p_i^x \ 0 \ p_i^y \ 0]$ . The components of  $\Sigma_x$  corresponding to  $x$  and  $y$  positions are computed using (25.6). Since the fingerprints do not provide any information on the remaining components of the covariance matrix, these values are obtained based on the prior knowledge of pedestrian motion as specified by  $\mathbf{Q}$  shown in (25.25).

Now the pairs  $\{(\mathbf{x}_i, \bar{\mathbf{r}}_i) | i = 1, \dots, N\}$  can be used to generate the state estimate  $\hat{\mathbf{x}}_r(k)$  and its associated covariance  $\mathbf{P}_r(k)$  through the zero-memory estimator.

From the linear Gaussian model, we obtain  $f(\mathbf{x}(k)|\mathcal{R}(k-1)) = \mathcal{N}(\hat{\mathbf{x}}(k|k-1), \mathbf{P}(k|k-1))$ . Moreover, the zero-memory estimator provides  $f(\mathbf{x}(k)|\mathbf{r}(k)) \approx \mathcal{N}(\hat{\mathbf{x}}_r(k), \mathbf{P}_r(k))$ . Substituting the densities  $\mathcal{N}(\hat{\mathbf{x}}(k|k-1), \mathbf{P}(k|k-1))$  and  $\mathcal{N}(\hat{\mathbf{p}}(k), \mathbf{P}(k))$  into (25.16), we have

$$\begin{aligned}f(\mathbf{x}(k)|\mathcal{R}(k)) &= \frac{f(\mathbf{r}(k)|\mathbf{x}(k)) f(\mathbf{x}(k)|\mathcal{R}(k-1))}{f(\mathbf{r}(k)|\mathcal{R}(k-1))} \\ &= \frac{f(\mathbf{r}(k)|\mathbf{x}(k)) f(\mathbf{x}(k)|\mathcal{R}(k-1))}{\int f(\mathbf{r}(k)|\mathbf{x}(k)) f(\mathbf{x}(k)|\mathcal{R}(k-1)) d\mathbf{x}(k)} \\ &= \frac{f(\mathbf{r}(k))}{f(\mathbf{x}(k))} \frac{f(\mathbf{x}(k)|\mathbf{r}(k)) f(\mathbf{x}(k)|\mathcal{R}(k-1))}{\int \frac{f(\mathbf{r}(k))}{f(\mathbf{x}(k))} f(\mathbf{x}(k)|\mathbf{r}(k)) f(\mathbf{x}(k)|\mathcal{R}(k-1)) d\mathbf{x}(k)}.\end{aligned}$$

The two densities  $f(\mathbf{r}(k))$  and  $f(\mathbf{x}(k))$  represent the prior information. Since no subjective information is available about  $\mathbf{r}(k)$  and  $\mathbf{x}(k)$ , noninformative uniform priors

are chosen for these parameters [13]. Using the assumption of uniform priors, the above expression is simplified to find

$$f(\mathbf{x}(k)|\mathcal{R}(k)) \approx \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}) \quad (25.30)$$

where

$$\hat{\mathbf{x}}(k|k) = \mathbf{P}(k|k) (\mathbf{P}(k|k-1)^{-1} \hat{\mathbf{x}}(k|k-1) + \mathbf{P}_r^{-1} \hat{\mathbf{x}}_r(k)), \quad (25.31)$$

$$\mathbf{P}^{-1}(k|k) = \mathbf{P}(k|k-1)^{-1} + \mathbf{P}_r^{-1}(k). \quad (25.32)$$

The complete filter algorithm is shown in Figure 25.4 and the filter structure is depicted in Figure 25.5. The main novelty of this nonparametric information (NI) filter [28] is in the nonparametric description of the measurement equation: Instead of using a measurement equation, the NI filter uses an implicit representation using the anchor point data. Note that the NI filter has a similar form to the information filter in that model and measurement contributions are fused linearly using *filter gains* that are related to *information* content in the Fisher sense [30, 31]. In the case of the measurements, the inverse covariance is computed in a nonparametric manner of the spatially distributed anchor points.

As previously mentioned, the derivation of Eq. (25.16) assumed conditional independence of the measurements over time. This assumption is generally not true in practice due to the dependence of the RSS on environmental factors affecting radio propagation, including doors opening or closing, elevators moving between floors, and users present in the area. Moreover, RSS values are dependent on the type of network

**Inputs:**

Set of location fingerprints:  $\{(\mathbf{p}^1, \bar{\mathbf{r}}^1), \dots, (\mathbf{p}^N, \bar{\mathbf{r}}^N)\}$

RSS observation at time  $k$ :  $\mathbf{r}(k)$

**Outputs:**

State estimate at time  $k$ :  $\hat{\mathbf{x}}(k|k)$

Estimation covariance at time  $k$ :  $\mathbf{P}(k|k)$

**Preprocessing:**

Generate  $\mathbf{H}_x$  and  $\mathbf{x}^i$ ,  $i = 1, \dots, N$

**NI filter:**

**State update:**

$$\hat{\mathbf{x}}(k|k-1) = \mathbf{F}\hat{\mathbf{x}}(k-1|k-1)$$

$$\mathbf{P}(k|k-1) = \mathbf{F}\mathbf{P}(k-1|k-1)\mathbf{F}^T + \mathbf{Q}$$

**Measurement update**

Spatial processing

$$w^i(\mathbf{r}(k)) = \mathcal{N}(\mathbf{r}; \mathbf{r}^i, \mathbf{H}_r) \left( \sum_{i=1}^{N(k)} w^i(\mathbf{r}(k)) \right)^{-1}$$

$$\hat{\mathbf{x}}_r(k) = \sum_{i=1}^{N(k)} w^i(\mathbf{r}(k)) \mathbf{x}^i$$

$$\mathbf{P}_r(k) = \sum_{i=1}^{N(k)} w^i(\mathbf{r}(k)) (\mathbf{H}_r + (\hat{\mathbf{x}}_r(k) - \mathbf{x}^i)(\hat{\mathbf{x}}_r(k) - \mathbf{x}^i)^T)$$

Fusion

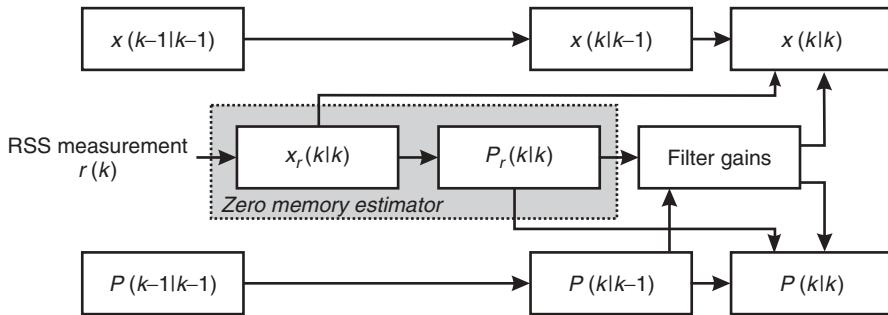
$$\mathbf{P}(k|k)^{-1} \hat{\mathbf{x}}(k|k) = \mathbf{P}(k|k-1)^{-1} \hat{\mathbf{x}}(k|k-1) + \mathbf{P}_r^{-1} \hat{\mathbf{x}}_r(k)$$

$$\mathbf{P}(k|k)^{-1} = \mathbf{P}(k|k-1)^{-1} + \mathbf{P}_r^{-1}(k)$$

**Initial conditions:**

$$\hat{\mathbf{x}}(0|0) = \hat{\mathbf{x}}_r(0), \quad \mathbf{P}(0|0) = \mathbf{P}_r(0)$$

**Figure 25.4** Nonparametric information (NI) filter.



**Figure 25.5** Overview of NI filter.

card used during measurements. Lastly, systematic errors of the spatial processor (e.g., bias [6]) causes the measurement noise to be correlated over time.

The Gaussian approximation of the measurement equation also results in suboptimality of the estimate  $\hat{p}(k|k)$  with respect to the MMSE criterion. The RSS measurements used in (25.7) are not the true RSS values but quantized values provided by the the network interface card. This leads to non-Gaussian quantization noise that is processed through the nonlinear relationship of (25.7). This type of processing generally leads to non-Gaussian posterior distributions.

## 25.6 COGNITION AND FEEDBACK

In traditional WLAN tracking systems, radio sensors monitor the environment in a passive manner. The nonstationary and noisy nature of the indoor radio environment, however, motivates the use of an *active* sensing and tracking paradigm to deal with deviations of the filter-operating conditions from those learned based on location fingerprinting. In particular, we shall explore the design of a *cognitive dynamic system* [32, 33] for WLAN tracking [28]. Cognitive dynamic systems learn rules of behavior through interactions with the environment to deal with uncertainties in a robust and reliable manner [32, 33]. Such systems have been proposed across a wide range of applications such as radio and radar systems [34, 35].

A cognitive tracking system is aware of the environment, anticipates future operating conditions, and adapts its sensing and estimation parameters accordingly. The main components of a cognitive tracking system are shown in Figure 25.6.

Such a system possesses the following aspects of cognition [31]:

- *Knowing* The characterization of the radio propagation environment is obtained through prior knowledge represented in the form of a radio map.
- *Perceiving* The WLAN tracking system senses and adapts to the nonstationary radio environment through adaptive scene analysis comprised of both AP and anchor point selection. Here, cognition is achieved through feedback—an essential ingredient of intelligent behavior [31]. In particular, positioning predictions from the Bayesian tracker are used to allow adaptive sensor selection strategies.
- *Conceiving* The tracking system deals with environmental uncertainties through Bayesian tracking. The recursive estimation framework allows for preservation

of knowledge over time and prediction and anticipation of future operating conditions.

- *Reliability* The cognitive design uses past estimation values in determining future system parameters. As with any feedback system, an erroneous position estimate can therefore propagate to future estimates, degrading positioning accuracy. A strategy for detection and mitigation of outlier estimates is therefore necessary.

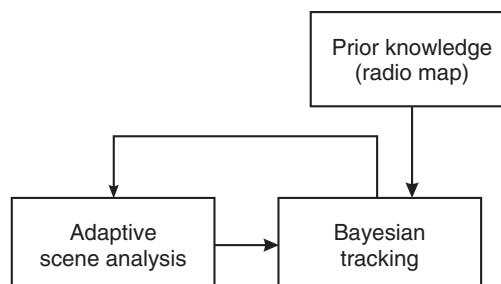
In the remainder of this section, incorporation of state estimates and predictions in selection of sensors and anchor points will be discussed. Moreover, the use of environmental interactions for verification of state estimates will be studied.

### 25.6.1 Adaptive Scene Analysis (Sensor Selection)

The WLAN tracking system builds its knowledge of the environment through RSS readings from available APs as well as prior knowledge provided by the location fingerprints. The nonstationary nature of the indoor propagation environment, however, causes operating conditions to deviate from those learned based on the location fingerprints. This situation motivates the design of an adaptive system that anticipates and adjusts its internal parameters based on current conditions. As shown in Figure 25.6, this is achieved through feedback: Predictions from the NI filter are used to adaptively determine the relevant portion of the RSS scene at each time step. The RSS scene is comprised of two components: anchor points and APs. Before discussing the particulars of anchor points and AP selection schemes, it is important here to distinguish two types of feedback used in this cognitive dynamic system:

- *Local Feedback* This type of feedback is used internally by the Kalman and NI filters to implement the recursive predictor–corrector structure. This allows the filter to adjust the relative weights of the prediction and measurement contributions in a *reactive* manner.
- *Global Feedback* This type of feedback connects two distinct modules of the tracking system, namely, radio sensing and Bayesian tracking. This allows for the adjustment of sensing operations based on information provided by the tracker in a *proactive* manner.

The anchor point and AP selection schemes used in the cognitive designs are discussed next.



**Figure 25.6** Overview of the cognitive design.

**25.6.1.1 Anchor Point Selection** Discrepancies between observations used by the Bayesian tracker and fingerprint values mean that anchor points that are physically far away from the observation point may erroneously contribute to the position estimate. To mitigate this issue, WLAN tracking systems intelligently choose a *region of interest* (ROI) for their operation, and only anchor points in the ROI are used for positioning by the zero-memory estimator. The ROI can be determined based on the observation that spatial points that are close in the physical space, receive coverage from similar sets of APs. Offline clustering of anchor points based on their respective covering APs or fingerprint values can be used to reduce the search space to a single cluster [17, 36]. Spatial filtering can also be performed during the online operation of the system using binary AP coverage vectors generated from RSS vectors or fingerprints [6]. These methods, however, rely directly on the radio map data and therefore they too are highly sensitive to changes in the radio propagation environment.

The alternative is to determine the ROI based on the one-step state predictions from the Bayesian tracker. Moreover, this tracker provides a measure of accuracy, that is, the prediction covariance, alongside the state prediction, which can be used to determine the size of the ROI. Specifically, the center of the ROI is chosen as the state prediction  $\hat{\mathbf{x}}(k|k-1)$ , and its size is determined by the confidence ellipsoid defined by  $\mathbf{P}(k|k-1)$ . The  $g$ -sigma confidence ellipsoid [23, 37] is defined as the locus of spatial points  $\mathbf{p}$  such that

$$(\mathbf{p} - \hat{\mathbf{p}}(k|k-1))^T \mathbf{P}(k|k-1)^{-1} (\mathbf{p} - \hat{\mathbf{p}}(k|k-1)) = g^2. \quad (25.33)$$

The lengths of the semiaxes of the above ellipsoid are  $g$  times the square root of the eigenvalues of the covariance matrix  $\mathbf{P}(k|k-1)$ . The parameter  $g$  controls the probability that the error vector lies within the ellipsoid defined in (25.33). For example, the probability that the error vector is inside the 3-sigma and 4-sigma ellipsoids is 98.89 and 99.97%.

**25.6.1.2 Access Point Selection** In general, estimation of a two-dimensional position requires measurements from at least three APs (although single AP positioning is also possible due to asymmetry of the propagation environment). In a typical WLAN environment, however, the number of available APs is much greater than three. Using all available APs increases the computational complexity of the positioning algorithm and can lead to degradations in positioning accuracy caused by biased and correlated estimates. This motivates the need for an AP selection component to choose a subset of the available APs for use in positioning. Selection methodologies include choosing a subset of APs with the highest observation RSS to maximize the probability of coverage[35], and use of divergence [6] and information-theoretic [17] measures to minimize redundancy between the selected APs and maximize information gained from the APs, respectively.

Interestingly, the statistical characteristics of APs change over space, reinforcing the need for an adaptive scene analysis method where APs are selected based on their properties over the ROI as opposed to the entire space. As an example consider the selection of APs based on Fisher criterion [38, 39] where the score of the  $a$ th AP is defined as the within-cluster to between-cluster scatter ratio evaluated over the anchor points in the ROI [28]:

$$\xi_a = \frac{\sum_{i=1}^{N(k)} (\sigma_i^a)^2}{\sum_{i=1}^{N(k)} (\bar{r}_i^a - \bar{r}^a)^2} \quad (25.34)$$

where

$$\bar{r}_i^a = \frac{1}{T} \sum_{t=1}^T r_i^a(t) \quad \text{and} \quad (\sigma_i^a)^2 = \frac{1}{T-1} \sum_{t=1}^T (r_i^a(t) - \bar{r}_i^a)^2$$

are the sample mean and variance of the training RSS from AP  $a$  at  $\mathbf{p}_i$ . Moreover,  $\bar{r}^a = \sum_{i=1}^{N(k)} r_i^a$  is the mean RSS value from this AP across anchor points. Lastly,  $N(k)$  is the number of anchor points in the ROI at time  $k$ . The set of  $d$  APs with highest scores  $\xi_a$  are selected for positioning.

### 25.6.2 Outlier Mitigation

As with any feedback system, an erroneous state estimate can propagate to future estimates, degrading positioning accuracy. This is of spacial concern in the WLAN tracking problem where RSS observations are known to be noisy.

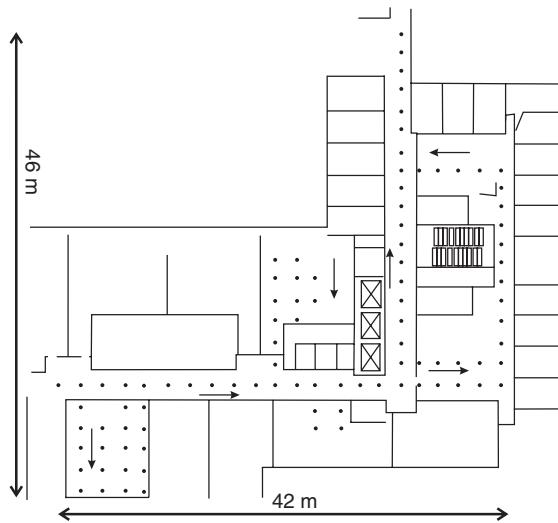
The nonparametric zero-memory formulation provides a means for detecting outlier RSS observations that can lead to degradation of positioning accuracy. In particular, the sum of the weights computed by the zero-memory estimator,  $\sum_{i=1}^{N(k)} w_i(\mathbf{r}(k))$  can serve as the detection criterion. Since  $w_i(\mathbf{r}(k)) = \mathcal{N}(\mathbf{r}; \mathbf{r}_i, \mathbf{H}_i)$ , we have  $0 < \sum_{i=1}^{N(k)} w_i(\mathbf{r}(k)) \leq N(k)$ . Small values of this sum indicate that the observation  $\mathbf{r}(k)$  does not match the location fingerprints for any of the anchor points in the selected region of space and may therefore be an outlier. In the case that an outlier is detected, the Bayesian tracker may skip the measurement update step and simply rely on state predictions.

## 25.7 TRACKING EXAMPLE

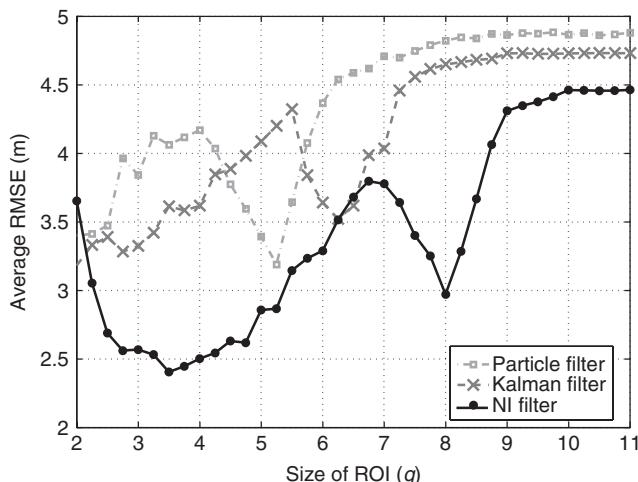
To illustrate the capabilities of the above approaches, we present tracking results obtained based on data collected from a real office environment. The data was collected during normal office hours on the fourth floor of an eight-story building at the University of Toronto. This building houses various faculty and graduate student offices as well as classrooms, meeting rooms, and laboratories. Figure 25.7 shows the layout of this environment. The dimensions of the experimentation site were 46 m by 42 m and a total of 55 IEEE 802.11b/g APs were detectable throughout the floor from various deployments, each providing only partial coverage of the environment. The average number of APs per point was 11.6. Out of these, three APs were selected using the Fisher criterion of (25.34) at each time step.

The RSS measurements used herein were collected using a Toshiba Satellite laptop with a Pentium M processor, an onboard Intel PRO/Wireless 2915ABG Network Adapter, and Windows XP operating system. RSS readings were obtained through a publicly available network sniffer software, NetStumbler (<http://www.netstumbler.com>), providing an RSS sampling rate of at most 2 samples/second. RSS measurements are reported as integers in the range  $(-100, 0)$  in unit of decibels relative to 1 milliwatt (dBm).

A total of 93 anchor points with a separation of 2 m in each direction were used. For each point, a total of 60 measurements were collected in two intervals of 30 s over 2 days. The testing set contains 34 tracks collected by two users and covers a variety



**Figure 25.7** Map of experimental area. Black dots represent survey points and arrows indicate the orientation of laptop during training.



**Figure 25.8** Average RMSE over 34 test tracks for the Kalman, nonparametric information, and particle filters versus the size of the ROI.

of user mobility scenarios. The average track length is 51 m and average speeds for the two users are 0.43 and 0.64 m/s.

Figure 25.8 compares the average root mean square error (ARMSE) resulting from tracking based on the Kalman and NI filters discussed here. Moreover, the ARMSE results are also shown for a particle filter with 500 particles. In order to ensure fair comparison of filter performances, the particle filter uses the kernel density estimate of the likelihood for determining the particle weights. All three filters use the cognitive design for anchor point and AP selection, and results depict the effect of the size of ROI on tracking accuracy.

The results indicate that adaptive scene analysis is effective in improving the tracking accuracy. For example, in the case of the NI filter, adaptive scene analysis results in an improvement of 2.05 m in positioning error or 46% (2.41 m for  $g = 3.5$  and 4.46 m for  $g = 11$ ). For small values of  $g$  ( $g < 2$ ), the confidence region excludes relevant anchor points and thus an increase in ARMSE is observed. For larger values of  $g$  ( $g > 6$ ) the size of the confidence region grows and irrelevant anchor points degrade the positioning accuracy. The figure also indicates that the NI filter outperforms the Kalman and particle counterparts, even when no cognition is used.

## 25.8 CONCLUSIONS

In this chapter, we discussed the problem of pedestrian tracking based on received signal strength from spatially distributed access points. Due to the complexity of the indoor propagation environment, an explicit form for the position RSS dependency is unknown. This motivates the use of nonparametric techniques for tracking. The starting point of the discussion in this chapter was the use of the nonparametric kernel density estimator (KDE) for memoryless positioning where a nonparametric position estimator based on the minimum mean square error (MMSE) was reviewed. This KDE technique has the advantage of providing a covariance matrix that is used to gauge the reliability of the position estimate. It is this feature that allowed the development of state-space filters, in particular the NI filter, which augment memoryless estimates with the knowledge of pedestrian motion dynamics. The motion model serves two purposes. First, it acts as a secondary source of information to complement the RSS measurement during estimation. Second, it provides state predictions that led to the development of a cognitive dynamic tracking system. This design uses global feedback to guide the selection of anchor points and APs used during estimation and to mitigate difficulties that arise due to discrepancies between training and testing conditions resulting from the time-varying nature of the environment.

While the application of cognitive dynamic systems has been limited to cognitive radio and radar to this time, the results of this chapter indicate that the introduction of cognition in the tracking system can lead to similar benefits. Specifically, it was shown that predictions from the NI filter can be used for adaptive radio scene analysis through determination of a spatial region of interest at each iteration of the filter. The proposed cognitive system is similar to cognitive radar in that sensing (AP selection) is carried out based on previous interactions with the environment. We extend the cognitive paradigm a step further by adapting the Bayesian target tracker to anticipated conditions.

The nonparametric nature of the methods presented in this chapter makes them particularly suitable for application to other areas where probabilistic distributions of sensor patterns are unknown or complicated by operations such as quantization performed to meet power and bandwidth constraints. Future research directions, therefore, include the extension of the methods proposed herein to distributed and constrained settings of sensor networks used for positioning and self-localization.

## REFERENCES

1. M. McGuire and K. Plataniotis, "Dynamic model-based filtering for mobile terminal location estimation," *IEEE Trans. Vehic. Technol.*, vol. 52, no. 4, pp. 1012–1031, 2003.

2. N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00)*, 2000, pp. 32–43.
3. M. Youssef and A. K. Agrawala, "Continuous space estimation for WLAN location determination systems," paper presented at the IEEE Thirteenth International Conference on Computer Communications and Networks, 2004.
4. P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of IEEE Infocom*, Vol. 2, 2000, pp. 775–784.
5. M. B. Kjaergaard, "A taxonomy for radio location fingerprinting," *Lecture Notes in Computer Science*, vol. 4718, pp. 139–156, 2007.
6. A. Kushki, K. Plataniotis, and A. Venetsanopoulos, "Kernel-based positioning in wireless local area networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 689–705, 2007.
7. A. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Comput.*, vol. 2, no. 1, pp. 46–55, 2003.
8. Y. Jie, Y. Qiang, and N. Lionel, "Learning adaptive temporal radio maps for signal-strength-based location estimation," *IEEE Trans. Mobile Comput.*, 2008.
9. A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.
10. K. Kaemarungsi and P. Krishnamurthy, "Properties of indoor received signal strength for WLAN location fingerprinting," in *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS)*, 2004, pp. 14–23.
11. M. Kjaergaard and C. Munk, "Solving rss client differences by hyperbolic location fingerprinting," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*, 2008.
12. Y. Zhu, *Multisensor Decision and Estimation Fusion*, Boston: Kluwer Academic, 2003.
13. J. Manyika and H. F. Durrant-Whyte, *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*, New York: Ellis Horwood, 1994.
14. A. Kushki, K. Plataniotis, and A. Venetsanopoulos, "Sensor selection for mitigation of RSS-based attacks in wireless local area network positioning," to appear in the *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008.
15. P. Krishnan, A. Krishnakumar, W.-H. Ju, C. Mallows, and S. Ganu, "A system for LEASE: Location estimation assisted by stationary emitters for indoor RF wireless networks," in *Proceedings of IEEE Infocom*, Vol. 2, 2004, pp. 1001–1011.
16. M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, 2005, pp. 205–218.
17. Y. Chen, Q. Yang, J. Yin, and X. Chai, "Power-efficient access-point selection for indoor location estimation," *IEEE Trans. Knowledge Data Eng.*, vol. 18, no. 7, pp. 877–888, 2006.
18. J. Pan, J. Kwok, Q. Yang, and Y. Chen, "Multidimensional vector regression for accurate and low-cost location estimation in pervasive computing," *IEEE Trans. Knowledge Data Eng.*, vol. 18, no. 9, pp. 1181–1193, 2006.
19. P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis, "On indoor position location with wireless LANs," in *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Vol. 2, 2002, pp. 720–724.
20. Z. Xiang, S. Song, J. Chen, H. Wang, J. Huang, and X. Gao, "A wireless LAN-based indoor positioning technology," *IBM J. Res. Devel.*, vol. 48, nos. 5/6, pp. 617–626, 2004.
21. T. Roos, P. Myllymki, H. Tirri, P. Misikangas, and J. Sievnen, "A probabilistic approach to WLAN user location estimation," *Int. J. Wireless Inform. Networks*, vol. 9, no. 3, pp. 155–164, 2002.

22. B. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, 1986.
23. Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, New York: Wiley, 2001.
24. S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 2002.
25. T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Prentice Hall, 2000.
26. A. Kushki, K. Plataniotis, and A. N. Venetsanopoulos, "Location tracking in wireless local area networks with adaptive radio maps," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 5, 2006, pp. 741–744.
27. I. Guvenc, C. Abdallah, R. Jordan, and O. Dedeoglu, "Enhancements to RSS based indoor tracking systems using Kalman filters," in *Proceedings of the International Signal Processing Conference and Global Signal Processing Expo*, 2003.
28. A. Kushki, K. Plataniotis, and A. Venetsanopoulos, "Cognitive dynamic tracking for indoor wireless local area networks," submitted to *IEEE Trans. Mobile Comput.*, 2007.
29. G. Antonini, "A discrete choice modeling framework for pedestrian walking behavior with application to human tracking in video sequences," PhD dissertation, École Polytechnique Fédérale de Lausanne, 2006.
30. A. G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*, Boca Raton, FL: CRC Press, 1998.
31. Z. Quan and A. H. Sayed, "Innovations-based sampling over spatially-correlated sensors," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 3, 2007, pp. 509–512.
32. S. Haykin, "Cognitive dynamic systems," *Proc. IEEE*, vol. 94, no. 11, pp. 1910–1911, 2006.
33. S. Haykin, "Cognitive dynamic systems," in *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 4, 2007, pp. 1369–1372.
34. S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, 2005.
35. S. Haykin, "Cognitive radar: A way of the future," *IEEE Signal Process. Mag.*, vol. 23, no. 1, pp. 30–40, 2006.
36. M. Youssef, A. Agrawala, and A. U. Shankar, "WLAN location determination via clustering and probability distributions," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, 2003, pp. 143–150.
37. H. L. V. Trees, *Detection, Estimation, and Modulation Theory*, New York: Wiley, 2001.
38. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed., New York: Wiley, 2001.
39. J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, New York: Cambridge University Press, 2004.



## CHAPTER 26

---

# Reconfigurable Self-Activating Ion-Channel-Based Biosensors

Vikram Krishnamurthy<sup>1</sup> and Bruce Cornell<sup>2</sup>

<sup>1</sup>University of British Columbia, Department of Electrical and Computer Engineering, Vancouver, Canada

<sup>2</sup>St. Leonards Australia, Surgical Diagnostics Ltd

### 26.1 INTRODUCTION

Biological ion channels are water-filled subnano-sized pores formed by protein molecules in the membranes of all living cells [1, 2]. Ion channels in cell membranes play a crucial role in living organisms. They selectively regulate the flow of ions into and out of a cell and regulate the cell's biochemical activities. In the past few years, there have been enormous strides in our understanding of the structure–function relationships in biological ion channels due to the combined efforts of experimental and computational biophysicists [3]. The 2003 Nobel Prize in Chemistry was awarded to R. MacKinnon for determining the crystallographic structures of several different ion channels, including the bacterial potassium channel [4, 5]. This chapter focuses on another recent advance in biological ion channels: The design of biosensors that exploit the selective conductivity of ion channels. Such ion-channel-based biosensors can detect target molecular species of interest across a wide range of applications. These include medical diagnostics, environmental monitoring, and general biohazard detection.

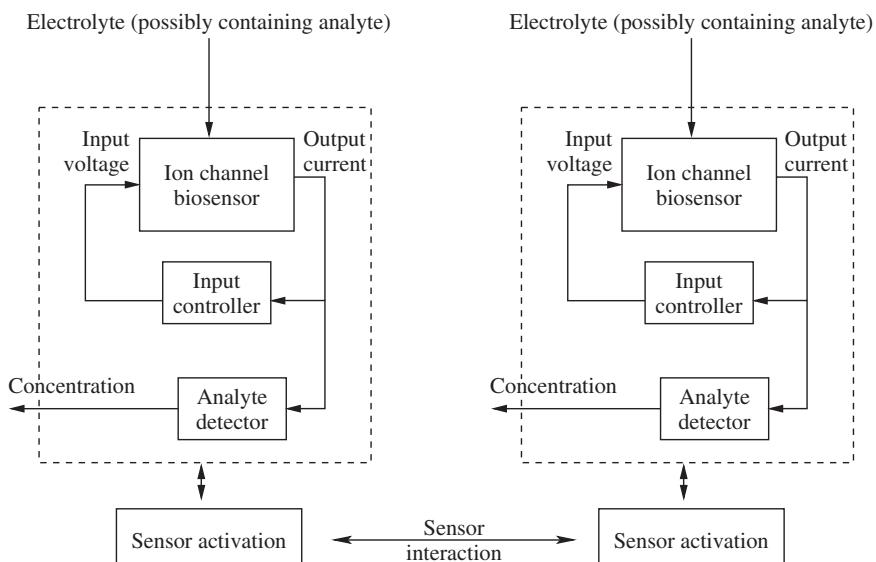
The ion-channel-based biosensors we focus on in this chapter are built using gramicidin A. Gramicidin A was one of the first antibiotics isolated in the 1940s [6, p. 130] and has a low molecular weight. In [7], a novel biosensor, which incorporates gramicidin ion channels into an artificial cell membrane was developed by our co-author (B. Cornell) and published in *Nature*; see also [8–10]. These are commercially available from Surgical Diagnostics.<sup>1</sup> This chapter describes how such ion channel biosensors work, how they can be modeled as a stochastic dynamical system, how their input can be dynamically adapted to minimize the detection error covariance, and finally how novel game-theoretic activation algorithms can be used for networks

<sup>1</sup>Surgical Diagnostics Ltd., Unit 2/12 Frederick St., St. Leonards NSW 2065 Australia. Email: BruceC@surgicaldiagnostics.com.

of such biosensors. Since the gramicidin channels move (diffuse) randomly in the outer membrane of the ion channel biosensor, we can also view the biosensor as a fully functioning nanomachine with moving parts. We refer the reader to [11] for an interesting overview of the interface between molecular biology (ion channels) and microelectronics.

We will use the following biological-related terminology: Target molecules, that is, molecules we wish to detect, are called *analytes*. The cell membrane is called a *lipid bilayer*. Roughly speaking, it is comprised of two layers of lipid (fat) that slide on each other. Each layer is called a *monolayer*.

Figure 26.1 gives a schematic of the logical organization of this chapter. Each box in Figure 26.1 demarcated in broken lines denotes a biosensor unit with a signal processing and control unit. The chapter begins with modeling each individual functional unit within the biosensor unit. We model the individual ion channel biosensor as a second-order linear system. Then we formulate the dynamics of the biosensor response to analyte. The presence of analyte decreases the admittance of the biosensor. We devise an input controller that optimizes the input excitation to the biosensor to minimize the covariance of the biosensor impedance (see Fig. 26.1). We present a sequential multihypothesis detection algorithm for the biosensor to detect the concentration of analyte. The ability of the biosensor to adapt its input excitation to minimize the covariance error gives it a reconfigurable functionality. Finally, we consider a network of several biosensors—the figure shows two such biosensors for illustrative purposes. How can the biosensors activate themselves in a decentralized fashion? We develop game-theoretic sensor activation algorithms that can be deployed by each biosensor. Roughly speaking, game theory can be viewed as a generalization of stochastic control where multiple controllers fight to optimize their individual revenues. We use the theory of global games to achieve decentralized activation. The theory of global games was



**Figure 26.1** Biosensor input excitation control, analyte detection, and activation control. The figure shows two biosensors in a network of biosensors.

first introduced in [12] as a tool for refining equilibria in economic game theory—see [13] for an excellent exposition—and has subsequently been used to model speculative currency attacks. Global games form an ideal tool for decentralized coordination of sensors; see [14]. In dense sensor networks (i.e., where a large number of sensors are present), it is natural to respond to the *decentralized awareness* of a sensor network with *decentralized information processing* for proper operation. The idea is that if each sensor or small group of biosensors can appropriately adapt its behavior to locally observed conditions, it can quickly self-organize into a functioning network, eliminating the need for difficult and costly centralized control.

The various signal processing and control tools used in this chapter range from classical to state of the art. The use of these concepts in the ion channel biosensor and network of biosensors is novel. Let us give some perspective on these tools. Modeling of the dynamics of the ion channel biosensor using an equivalent circuit with a resistor and capacitor in parallel has been studied extensively; see [10, 15]. The design of the input excitation to minimize the mean-square estimation error has been studied extensively in the systems identification literature under the area of optimal input design [16]. The use of sequential multihypothesis testing is an interesting area; see [17] for an excellent review, and also [18] for recent results in quickest time detection with exponential costs. The game-theoretic tools presented in this chapter are state of the art. Game theory is a natural tool for describing self-configuration of sensors since it models each sensor as a self-driving decision maker. An overview of methods and challenges in this area are given in [19, 20] and the references therein. We use the powerful concept of *global games* to devise biosensor activation algorithms that have a simple threshold Nash equilibrium structure. Despite the practicality and insight provided by global game models, and despite the popularity of game-theoretic approaches in the field of sensor networks, we are not aware of any other work applying global games in the realm of sensor networks.

## 26.2 BIOSENSORS BUILT OF ION CHANNELS

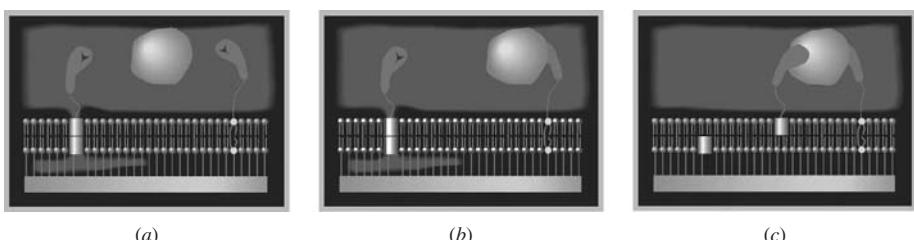
To understand the dynamics of the ion channel biosensor, we describe briefly its construction. The construction of the ion-channel-based biosensor developed by [7] involves sophisticated concepts in biochemistry. However, for our purposes its operation can be simply described as follows. First, an artificial “tethered” lipid monolayer is constructed containing tethered gramicidin channels. Tethered means that the inner layer of the membrane is fixed to a gold substrate (using a disulfide bond) and is no longer mobile. Then a second outer mobile monolayer comprising of lipids and gramicidin channels is introduced. These components “self-assemble” in water to form a lipid bilayer that mimics a cell membrane. A voltage of typically 100–300 mV is introduced across the membrane. The gramicidin channels act as subnano-sized pipes that move randomly along the outer monolayer of the bilayer. (The channels in the inner layer are tethered and hence cannot move). As the mobile gramicidin channels in the outer layer diffuse, occasionally a channel in the outer layer will align exactly with a channel at the inner level of the membrane, thereby forming a single longer pipe. Such a random event where two channels align is called the formation of a dimer. When a dimer forms, ions can travel along the two aligned pipes (channels), thereby resulting in a small current. Of course, at any given time, several such pairs

of pipes can align (forming dimers) or disassociate (breaking dimers) since the outer layer diffuses randomly. Therefore, the current recorded at the output of the biosensor is a random process.

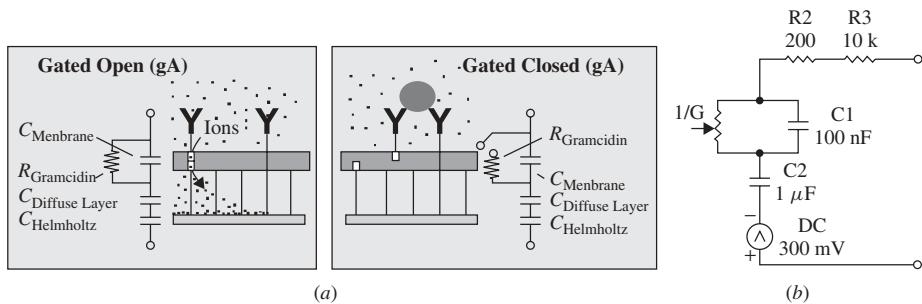
How does the biosensor respond to an analyte (target molecules)? In the construction of the biosensor, specific antibodies that recognize specific analyte molecules are attached to the mobile outer layer channels. Then electrolyte that may or may not contain the analyte is introduced. If no analyte molecules are present, the biosensor operates as above. On the other hand, if analyte is present in the electrolyte, then the arrival of analyte crosslinks antibodies attached to the mobile outer layer channels to those attached to membrane spanning lipid tethers. Due to the low density of tethered channels within the inner membrane, this anchors them distant, on average, from their immobilized inner layer channel partners. Gramicidin dimer conduction is thus prevented and the ionic admittance of the membrane decreases. Based on the resulting decrease in current (or equivalently, increase in resistance of the membrane), one can detect the presence of analyte. Based on how the resistance increases with time, one can also estimate the concentration of analyte. (Actually, the above description holds for the detection of large analyte molecules; when the biosensor is used to detect small analyte molecules, the resistance decreases with time; see [15]. The methods presented in this chapter apply to both types of behavior.)

The above operation of the ion channel biosensor is illustrated in Figure 26.2. As depicted in Figure 26.2a, mobile gramicidin ion channels (blue) diffuse by Brownian motion, forming and breaking dimers with tethered inner layer ion channels that traverse the membrane. Ions flow through the dimers from the bathing solution to the reservoir space between the membrane and the gold electrode surface. Antibody fragments (red) are linked to mobile ion channels and to tethered sites on the membrane surface. A diffusing target molecule is also shown (green). In Figure 26.2c the switch operates by the Brownian diffusion of the mobile ion channel, permitting crosslinking of the mobile gramicidins to the tethered lipid. This prevents the reformation of a conducting channel dimer and increases the membrane resistance.

The above ion channel biosensor can also be viewed as a switch that acts as a biological transistor. Figure 26.3a illustrates the equivalent circuit of the switch before and after the detection of analyte. Figure 26.3b details the components of the equivalent circuit. The resistor  $R = 1/G$  models the biosensor resistance and increases with the



**Figure 26.2** Schematic operation of ion channel biosensor. The pore of each gramicidin channel dimer is approximately 0.4 nm in radius and 2.5 nm long. The gold substrate of the tethered bilayer is shown in gold color. The antibodies are shown in red and the analyte molecule in green. Since the gramicidin channels move (diffuse) randomly within the lipid layers, the biosensor can be considered as a nanomachine with moving parts.



**Figure 26.3** Biosensor comprises of an ion channel switch that acts like a biological transistor. The left figure in (a) denotes the switched-on state when the ion channels are conducting, while the right figure denotes the switched-off state when the channels are not conducting. The equivalent circuit shown in (b) results in a second-order system.

presence of analyte.  $C_1$  denotes the capacitance of the membrane and is typically 100 nF.  $C_2$  denotes the interfacial capacitance of the gold substrate. Note that one face of the capacitor  $C_2$  is charged due to ions; the other face is due to electrons that form the output current of the biosensor. Thus  $C_2$  provides the interface between the biological sensor and the electrical instrumentation.  $R_2$  denotes the resistance of the electrolyte—this is typically known to be around 200  $\Omega$ . Finally,  $R_3$  denotes the input resistance of the amplifier at the next stage (which is ideally very large). The 300-mV bias voltage controls the value of the capacitor  $C_2$ . Typically,  $C_2$  can change from 1  $\mu\text{F}$  at 300 mV to 0.01  $\mu\text{F}$  at no bias. The biosensor is usually deployed with a 300-mV bias. The admittance transfer function of the equivalent circuit parametrized by  $G$  is

$$H(s; G) = \frac{I}{V_{\text{out}}} = \frac{s^2 + s a G}{s^2 R_2 + s(b_2 + b_1 G) + b_3 G}. \quad (26.1)$$

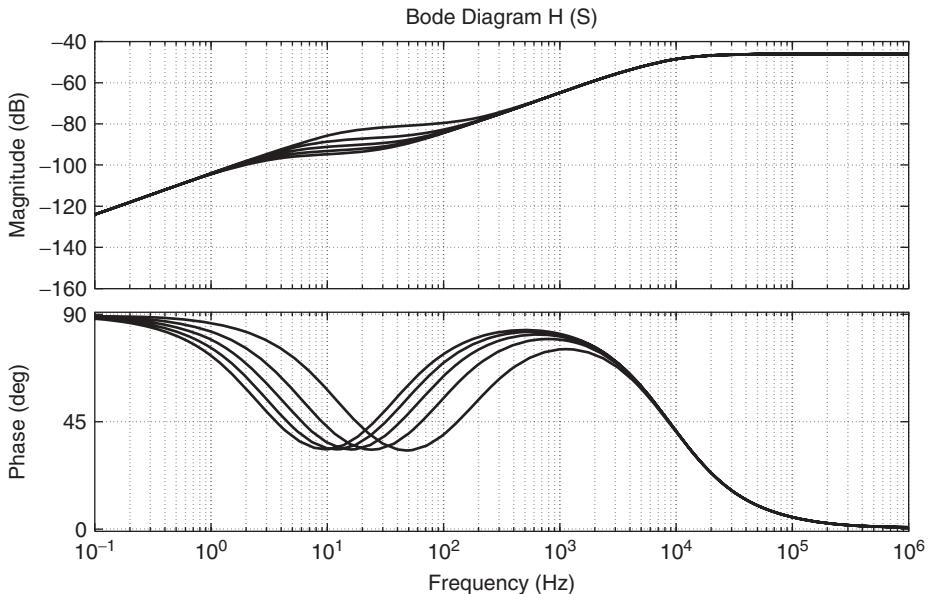
The constants in  $H(s)$  above are  $a = 1/C_1$ ,  $b_1 = R_2/C_1$ ,  $b_3 = 1/C_1C_2$ ,  $b_2 = 1/C_1 + 1/C_2$ .

Figure 26.4 illustrates the frequency response of the biosensor. When it detects analyte, its admittance  $G$  decreases and the phase response shifts to the left as illustrated in the figure. In Section 26.3 we address how this information can be used to design the input to the biosensor so as to minimize the error variance of the analyte estimate.

Before proceeding with the formulation of the dynamics of the biosensor response, we remark that the biosensor we consider below comprises of electrodes with area  $10^{-6} \text{ m}^2$ . Each such electrode contains several thousands of ion channels. Therefore, the measured current is the average effect of the formation and disassociation of thousands of dimers and is approximately continuous valued. In contrast, for microsize electrodes of area  $10^{-10} \text{ m}^2$  (or smaller), the current would be a finite-state process since only a few dimers would form and disassociate; such micro-sized electrodes are studied in our recent work [21].

### 26.2.1 Dynamics of Biosensor Response to Analyte

From an abstract point of view, we can describe the biosensor response to an analyte by a two-time-scale stochastic dynamical system: Let  $k = 1, 2 \dots$  denote the fast time



**Figure 26.4** Frequency response of the ion channel biosensor. Upper graph denotes the magnitude of the admittance  $H(jw; G)$  in decibels. Lower graph is the phase response of  $H(jw; G)$ . The different lines on the graphs indicate the magnitude and phase response of  $H(jw; G)$  for different values of cell membrane admittance  $1/G = 10 \text{ k}\Omega, 20 \text{ k}\Omega, 30 \text{ k}\Omega$ , and  $40 \text{ k}\Omega$ .

scale. Typically, with a sampling frequency of 1 kHz,  $k$  evolves on the millisecond time scale. Let  $n = 1, 2, \dots$  denote the slow time scale. We will refer to  $n$  as batch number. So for fixed large positive integer  $T$ , the  $n$ th batch on the slow time scale corresponds to times  $k \in [(n-1)T + 1, \dots, nT]$  in the fast time scale. Typically,  $T$  is chosen in the order of a few seconds.

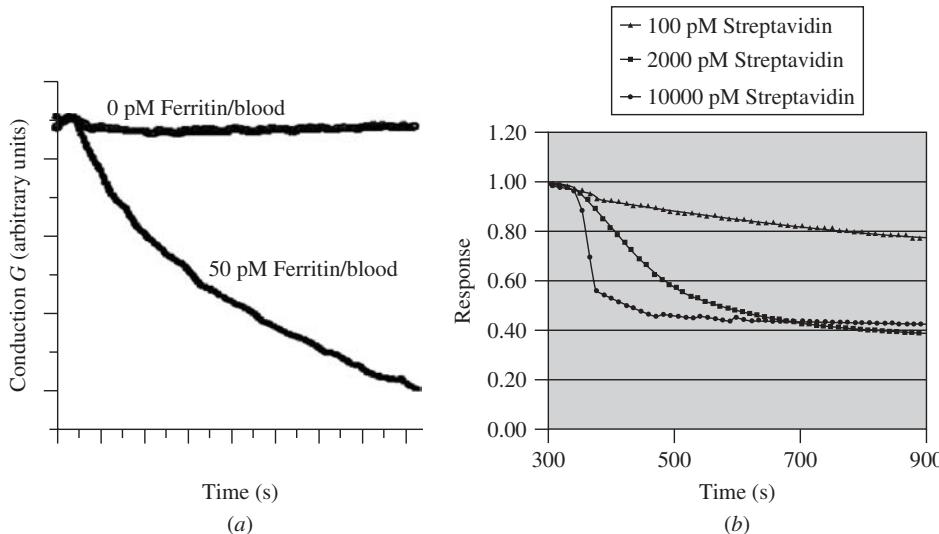
Let  $X$  denote the unknown concentration of analyte. From detailed experimental analysis of the biosensor and reaction rate dynamics analysis [10], it is known that the conductance  $G$  of the biosensor evolves on the slow time scale according to one of three different modes:

$$G_{n+1} = f^{\mathcal{M}}(G_n, X) + w_n, \quad (26.2)$$

where

$$\text{concentration mode } \mathcal{M} = \begin{cases} 1 & \text{if } X = 0 \text{ (no analyte)} : f^{\mathcal{M}}(G, X) = G, \\ 2 & \text{if } X \text{ is low} : f^{\mathcal{M}}(G, X) = G + \kappa_1, \\ 3 & \text{if } X \text{ is high} : f^{\mathcal{M}}(G, X) = \kappa_2 G + \kappa_3. \end{cases} \quad (26.3)$$

Here  $\kappa_1, \kappa_2, \kappa_3$  are constants, with  $|\kappa_2| < 1$ ;  $w_n$  is a zero-mean white Gaussian noise process with small variance and models our uncertainty in the evolution of  $G$ . The function  $f^{\mathcal{M}}$  models the fact that the admittance  $G_n$  of the membrane decreases according to one of three distinct modes depending on the concentration  $X$  of the analyte. For no analyte present ( $\mathcal{M} = 1$ ), the admittance remains constant. As described earlier, when analyte is present, the admittance  $G$  in (26.1) decreases due to analyte molecules binding to the antibodies in the biosensor. For low concentration ( $\mathcal{M} = 2 = \text{low}$ ), the



**Figure 26.5** Biosensor response to different analytes; see [9, 10] for details: (a) response to whole blood containing 50 pM ferritin and (b) response to streptavidin.

decrease in admittance is linear; for high concentration ( $\mathcal{M} = 3 = \text{high}$ ) the decrease is exponential.

Figure 26.5 shows examples of the biosensor response to two different analytes, namely, ferritin and streptavidin; see [9, 10] for details. Using streptavidin as the protein antigen and biotin as a low-molecular-weight receptor provides a rapid and clear demonstration of the different kinetic regimes of the sensor function. The streptavidin–biotin binding pair is one of the strongest and best characterized interactions available today and is used as a model system in Figure 26.5b to permit full characterization of the sensor.

Let  $u_k$  denote the applied input excitation voltage to the biosensor at time  $k$ . For practical purposes, in the ion channel biosensor, we restrict  $u_k$  to a pseudorandom binary sequence (PRBS) that is symmetric with  $u_k \in \{-300 \text{ mV}, 300 \text{ mV}\}$ . For simplicity of hardware implementation, we further restrict the PRBS sequence  $u_k$  to a two-state Markov chain with symmetric transition probability matrix:

$$\mathbf{P} = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}, \quad 0 \leq p \leq 1. \quad (26.4)$$

The measured current output (in discrete time) of the biosensor on each  $T$  length interval in the fast time scale is given by applying an antialiasing filter and then sampling the continuous time system (26.1). We use a Butterworth antialiasing filter and a bilinear transformation time discretization. Recall, [22] in a bilinear transformation discretization  $s$  is replaced in (26.1) with  $(2/T)(1 - z^{-1}/1 + z^{-1})$ , resulting in a discrete-time transfer function  $H(z^{-1}; G)$ . In input–output form, this yields a  $\Delta$ -order discrete-time ARMA process:

$$y_{k+1} = \sum_{i=1}^{\Delta} \alpha_i y_{k-i} + \beta_i u_{k-i}. \quad (26.5)$$

In our experiments we chose a third-order Butterworth antialiasing filter and  $\Delta = 5$ . The coefficients  $\alpha_i$ ,  $\beta_i$  are linear in  $G$ . Typically, in experiments with the biosensor,  $G$  is corrupted by ambient noise implying that the coefficients are of the form  $\alpha_i$ , and  $\beta_i$  are of the form  $(aG + e)$ , where  $a$  is some constant and  $e$  is noise. However, the noise components can be grouped into a single noise term and least-squares regression used to estimate the coefficients.

Before concluding this section, it is worthwhile remarking that maximum-likelihood parameter estimates of  $\kappa_1$ ,  $\kappa_2$ ,  $\kappa_3$  in (26.2) and the noise variances can be computed given the observations generated by (26.5). For example, the expectation maximization (EM) algorithm or recursive EM algorithm can be used [23–25].

### 26.3 JOINT INPUT EXCITATION DESIGN AND CONCENTRATION CLASSIFICATION FOR BIOSENSOR

When an analyte containing a specific molecule is introduced to the biosensor, the molecules of the analyte bind to the specific antibodies present in the biosensor. This results in a slow decrease in the admittance  $G$  of the biosensor or equivalently a decrease in the output current. By cleverly switching the applied input voltage  $u$  to the biosensor, we can infer the presence and concentration of analyte more rapidly. The goal of this section is to describe how to choose an optimal pseudorandom binary sequence of applied voltages  $u$  [parametrized by  $p$  in (26.4)] to the biosensor to minimize the detection time for the presence and concentration of analytes. We wish to simultaneously pick the optimal input  $u$  and minimize the time required to announce whether  $M = 1$ , 2, or 3. In its full generality, we are dealing with a change detection problem [26]—initially there is no analyte and then later there may be analyte. However, since the admittance of the biosensor changes on a slow time scale, it suffices to treat the problem as a batch of sequential multihypothesis detection problems and an input optimization problem. In other words we have a combined optimal input design and sequential multiple hypothesis problem. Moreover, by exploiting the two-time-scale nature of the biosensor dynamics and the characteristics of its response, we can nicely decouple the problem into optimizing  $u$  and then doing a sequential multihypothesis test.

#### 26.3.1 Optimal Input Excitation Design

First consider the problem of optimizing the PRBS input  $u$ . Our approach is based on optimal input design for open-loop experiments and is well known in the systems identification literature [16, Section 13.3]. The open-loop design is approximately justified due to the two-time-scale nature of the biosensor dynamics described above, and because we adjust the parameter  $p$  batchwise on the slow time scale  $n$ . Assuming that a consistent estimator is available for  $G$ , then the asymptotic covariance of the estimate when a prediction error method is applied for estimation is (see [16, Eq. (13.26)])

$$\overline{M} = \kappa \int_{-\pi}^{\pi} |H(e^{j\omega}; G)|^2 S_u(e^{j\omega}; p) d\omega + M_e.$$

Here  $\kappa$  and  $M_e$  denote terms independent of  $u$ . Also  $S_u(e^{j\omega}; p)$  denotes the spectral density of the input PRBS sequence  $u$  with transition probabilities given above.

The covariance  $E\{u_0 u_k\}$  of the Markov chain is straightforwardly evaluated as  $|u|^2 (2p - 1)^k$ , where  $|u|$  denotes the magnitude of binary symmetric signal  $u_k$ . In discrete time, the spectral density of the input PRBS (Markov chain)  $u$  is

$$S_u(z; p) = \frac{2|u|^2}{1 - (2p - 1)z^{-1}}. \quad (26.6)$$

The optimal choice of input that minimizes the asymptotic covariance of the estimate of  $G$  is

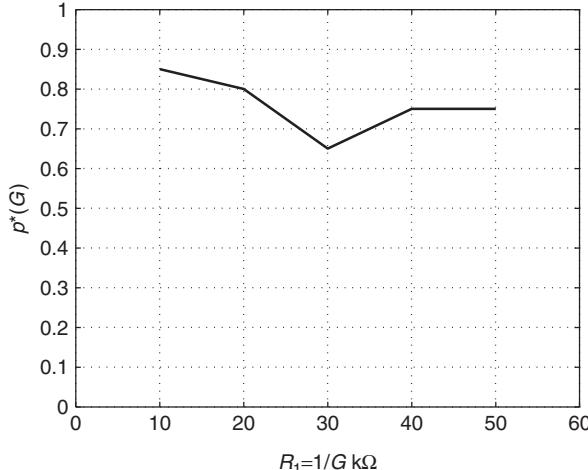
$$p^*(G) = \operatorname{argmax}_p \int_{-\pi}^{\pi} |H(e^{j\omega}; G)|^2 S_u(e^{j\omega}; p) d\omega. \quad (26.7)$$

The above optimization problem is easily solved numerically. Intuitively, the above statement means that the optimal PRBS choice focuses its energy on the part of the Bode plot that changes the most in Figure 26.4. Figure 26.6 shows how the optimal probability  $p^*(G)$  of the PRBS input  $u$  varies with admittance  $G$ . The really nice property of  $p^*(G)$  is that it varies very little with  $G$ . Therefore,  $p^*(G)$  varies very little with  $\mathcal{M}$ —since  $G$  evolves according to  $\mathcal{M}$ . As a result we can decouple the optimal input design and the sequential multihypothesis test.

### 26.3.2 Sequential Multihypothesis Test for Analyte Concentration

Having designed the optimal input  $u$  above, we now briefly describe the sequential multihypothesis test for classifying the analyte concentration  $\mathcal{M}$  based on measurements from the biosensor (26.2). The sequential multihypothesis test is specified by the following ingredients:

1. The estimated concentration mode  $\hat{\mathcal{M}}_n \in \{1, 2, 3, 4\}$  computed in the  $n$ th batch. Several methods can be deployed for computing the estimate  $\hat{\mathcal{M}}_n$  using the



**Figure 26.6** Optimal probability  $p^*(G)$  for pseudorandom binary sequence (PRBS) input excitation for each  $R_1 = 1/G$  value is plotted, and it can be concluded that the optimal probability is not sensitive to the channel resistance.

model (26.2), (26.5), and  $y$ . For example, using least-squares estimation of  $G$  in (26.5) and hence concentration  $X$ , we can then infer  $\hat{\mathcal{M}}$  by another least-squares estimation using (26.2). Alternatively, a maximum-likelihood estimate or minimum mean-square error estimate  $\hat{\mathcal{M}}_n$  can be computed using sequential Markov chain Monte Carlo methods.

2. The probability distribution of the estimated concentration, which we denote as

$$f_i(\hat{\mathcal{M}}) = P(\hat{\mathcal{M}}|\mathcal{M} = i), \quad i = 1, 2, 3. \quad (26.8)$$

3. At each time  $n$  on the slow time scale, the analyte detector (see Fig. 26.1) in the biosensor unit must choose one of five possible actions. Denote this set of actions as

$$\begin{aligned} \mathcal{A} = \{ &1 \text{ (continue)}, 2 \text{ (announce } \mathcal{M} = 1\text{)}, 3 \text{ (announce } \mathcal{M} = 2\text{)}, \\ &4 \text{ (announce } \mathcal{M} = 3\text{)} \}. \end{aligned}$$

Let  $a_n \in \mathcal{A}$  denote the action chosen at time  $n$ . If  $a_n = 1$  (continue), then an additional batch of observations are taken by the biosensor. If action  $a_n = \text{announce } \mathcal{M}_i$ , then the biosensor announces that the concentration of analyte is  $i$  and the stops taking further measurements at time  $n$ .

4. Costs of announcing decisions and making observations: Let  $c(\mathcal{M}, a_n)$  denote the user-specified cost of taking action  $a_n$  when the true concentration is  $\mathcal{M}$ . These costs are defined as follows:

- Let  $M$  denote the cost of taking a new batch of measurements. So  $c(\mathcal{M}, a = 1) = M$  is the cost of taking an additional batch of measurements and delaying the announcement.
- Let  $L_i$  denote the cost of wrongly announcing concentration  $i$  (i.e., picking action  $u = i$ ) when the actual concentration is  $\mathcal{M} = j$ .

The costs  $M$  and  $L_i$  are user defined and chosen to reflect the importance of the decisions. For example, in time-sensitive applications, when concentration classification needs to be made rapidly,  $M$  is chosen large.

Given the above ingredients, the aim is to minimize the total expected cost  $\sum_{n=1}^{\infty} \mathbf{E}\{c(\mathcal{M}, a_n)\}$  based on the estimated concentration modes  $\hat{\mathcal{M}}_n$ ,  $n = 1, 2, \dots$ . Thus, one is trading off the delay in announcing a decision versus the accuracy of the decision.

The optimal solution to the above optimization problem is obtained via stochastic dynamic programming [27]. Denote the posterior probability distribution of the concentration given the estimates  $\hat{\mathcal{M}}_1, \dots, \hat{\mathcal{M}}_n$  as

$$\pi_n(i) = P(\mathcal{M} = i | \hat{\mathcal{M}}_1, \dots, \hat{\mathcal{M}}_n). \quad (26.9)$$

Then via Bayes' rule, the posterior distribution is updated as

$$\pi_{n+1}(i) = \frac{\pi_n(i) f_i(\hat{\mathcal{M}}_n)}{\sum_{j=1}^4 \pi_n(j) f_j(\hat{\mathcal{M}}_n)}. \quad (26.10)$$

The posterior distribution  $\pi$  is called the *information state* in [28, 29] and forms the state space for the stochastic dynamic programming equation below. Let  $a(\pi) \in \mathcal{A}$  denote the optimal action taken when the posterior distribution is  $\pi$ . Then the solution to the sequential multihypothesis test is given by the following stochastic dynamic programming functional equation (also called Bellman's equation) [28, p. 256],[27]:

Choose optimal action

$$a(\pi) = \operatorname{argmin}_{a \in \mathcal{A}} Q(\pi, a)$$

where

$$J(\pi) = \min_{a \in \mathcal{A}} Q(\pi, a),$$

and

$$\begin{aligned} Q(\pi, a) = & \left\{ [(1 - \pi(1))L_1, [(1 - \pi(2))L_2, [(1 - \pi(3))L_3, \right. \\ & \left. M + \mathbf{E}_{\hat{\mathcal{M}}} \left[ J \left( \frac{\pi(i)f_i(\hat{\mathcal{M}})}{\sum_{j=1}^4 \pi(j)f_j(\hat{\mathcal{M}})} \right) \right] \right\}. \end{aligned} \quad (26.11)$$

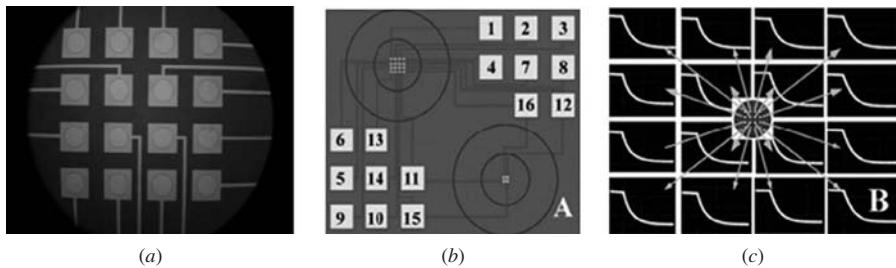
The above dynamic programming equation allows for the four possible actions: if  $a = 2$  is taken,  $\mathcal{M} = 1$  is announced, the cost of error is the probability of error  $1 - \pi(1)$  times the cost  $L_1$ ; similarly for actions 3 and 4. For action  $a = 1$ , a measurement cost of  $M$  is incurred and the posterior distribution is updated. Recall the above dynamic programming equation is an off-line procedure that constructs a lookup table for picking the optimal action for each possible state  $\pi$ , see [27–29] for textbook treatments.

As it stands, (26.11) does not directly lead to practical methodologies since  $Q(\pi, a)$  needs to be evaluated on an uncountable set of probability distributions  $\pi$ . However, the structure of (26.9) and (26.11) can be exploited in many useful cases to characterize the solution; see, for example [30, 31] where the more complex case of stochastic sensor scheduling for partially observed Markov decision processes is considered. We refer to [17, 27, 28, 31] where structural solutions are given to this dynamic programming problem. For example, [27, p. 59] in the case where we only wish to test for the presence of analyte, that is,  $\mathcal{M} \in \{1(\text{no analyte}), 2(\text{analyte present})\}$ , then given the posterior distribution  $\pi$ , the solution to (26.11) has the following simple threshold structure: [Note  $\pi_n(2) = P(\mathcal{M} = \text{analyte present} | \hat{\mathcal{M}}_1, \dots, \hat{\mathcal{M}}_n)$  below; see (26.9)]

$$a(\pi) = \begin{cases} \text{announce no analyte} & \text{if } \pi(2) < \underline{\pi}, \\ \text{announce analyte present} & \text{if } \pi(2) < \bar{\pi}, \\ \text{take an additional measurement} & \text{if } \underline{\pi} < \pi(2) < \bar{\pi}. \end{cases} \quad (26.12)$$

Here the constants  $\underline{\pi}$  and  $\bar{\pi}$  where  $0 \leq \underline{\pi} \leq \bar{\pi} \leq 1$  can be computed numerically.

*Summary:* This section described how the input voltage to the biosensor can be controlled to minimize the covariance in the estimate of the biosensor admittance  $G$ . Then we presented a sequential detection scheme for detecting the concentration  $\mathcal{M}$ .



**Figure 26.7** Biosensor array with 16 electrodes. Such arrays can be used to detect cocktails of analytes. In Section 26.3, we use a global game formulation to dynamically activate the electrodes in the biosensor.

## 26.4 DECENTRALIZED DEPLOYMENT OF DENSE NETWORK OF BIOSENSORS

The ion channel biosensor has been incorporated into a single chip that contains an array of 16 electrodes as illustrated in Figure 26.7. Each of the 16 electrodes in the array can be used to detect a possibly different analyte, and therefore the sensor can detect a cocktail of analytes. Figure 26.7c illustrates the current versus time in each of the 16 electrodes of the biosensor when a single analyte is detected. The current decays exponentially due to the increase in resistance. Naturally, by averaging over 16 electrodes, one can obtain a more accurate estimate of the analyte.

**Reusability:** The next level of design is a network that is comprised of a large number of biosensor chips. Each biosensor can be equipped with a transceiver to communicate its inference to a base station. This section deals with the design of networks of biosensors. A critical design issue for the biosensor network is lack of reusability of the biosensors. If the affinity of the receptor is high, then once an analyte binds to a receptor, it can take several hours for the molecule to come off the receptor. Note that a high affinity receptor is essential for a highly sensitive biosensor. So each electrode in a biosensor can realistically be used only once.<sup>2</sup>

Therefore, if all the biosensors were deployed simultaneously, even small concentrations of analyte can render them useless for subsequent sensing. So in situations where small concentrations of analyte are unimportant, but large concentrations of analyte are important, it is essential not to activate all the biosensors at once. (A typical example is a biotoxin or chemical bomb that disperses large concentrations of analyte versus the relative harmlessness of small concentrations of chemicals that might occur naturally). It makes sense to initially activate some sensors (or electrodes) to obtain a coarse estimate and then if required, deploy further sensors to obtain a high-resolution result. Based on this theme, in this section we devise novel decentralized algorithms for dynamic activation of biosensors.

In dense sensor networks (i.e., where a large number of sensors are present), it is natural to respond to the *decentralized awareness* of a sensor network with *decentralized information processing* for proper operation. The idea is that if each sensor or small group of biosensors can appropriately adapt their behavior to locally observed conditions, they can quickly self-organize into a functioning network, eliminating the

<sup>2</sup>There are also practical reasons for using an electrode only once. With biological specimens, the potential for bacterial and yeast cross contamination strongly favors the use once and discard philosophy.

need for difficult and costly centralized control. Our goal of this section is to develop a *global games* approach [14, 32] to biosensor activation in a dense sensor network. To give the reader a clear yet rapid treatment of the key ideas, our description of the assumptions and theory below is idealized. For further details on global games for sensor activation, we refer the reader to [14].

Before proceeding with the details, let us first illustrate a typical global game using the analogy of patrons that can visit a night club. Consider a night club comprising of a large number (actually a continuum) of patrons. Each patron receives noisy information  $Y$  about the quality of music  $X$  playing at a night club. Based on this noisy information the patron can choose either to *go* or *not go* to the night club. If a patron chooses not to go to the night club, he receives no reward. If the patron goes to the night club, he receives a reward  $X + f(\alpha)$  where  $\alpha \in [0, 1]$  is the proportion of patrons that decided to go to the night club. Thus the better the music quality, the higher the reward to the patron if he goes to the night club. On the other hand,  $f(\alpha)$  is typically a quasi-concave function with  $f(0) = 0$ . The reasoning is: If too few patrons decide to go, that is,  $\alpha$  is small, then  $f(\alpha)$  is small due to lack of social interaction. If too many patrons go to the night club, that is,  $\alpha$  is large, then  $f(\alpha)$  is also small due to the crowded nature (congestion) of the night club. Each patron is rational and knows that other patrons who choose to go to the night club will also receive the reward  $X + f(\alpha)$ . Each patron also knows that other patrons know that he knows this, and so on, ad infinitum. So each patron can predict rationally (via Bayes' rule; see Section 26.4.2) given its measurement  $Y$ , what proportion  $\alpha$  of patrons will choose to go to the nightclub. How should the agent decide rationally whether to go or not to go to the night club to maximize his reward?

The above night club problem is an example of a global game [13, 32] and is an ideal method for decentralized coordination among sensors. The example draws immediate parallels with decentralized sensor activation in large-scale sensor networks. Consider a biosensor network where each sensor has the capability of transmitting its inference about the analyte to a base station. Due to the lack of reusability of electrodes described above, assume that each sensor initially deploys only one electrode to get a coarse measurement of the presence of analyte. We call this the Low\_Res Mode (low-resolution mode). (Alternatively, a group of sensors could deploy initially a single coarse sensor). Next based on the measurements at the one electrode, an estimate  $Y$  of the quality (or importance) of analyte concentration  $X$  present in the data is computed using the detection schemes presented in the previous section. (For simplicity, we assume here that the importance of an analyte is proportional to its concentration. Alternatively, one can choose  $X$  as the true signal-to-noise ratio (SNR) of the analyte measurement and  $Y$  as the estimated signal to noise ratio). Each sensor then decides whether or not to switch to the High\_Res (high-resolution) mode where additional electrodes are deployed to obtain further more accurate measurements of the analyte. We denote sensors that switch to the High\_Res mode as *activated sensors*. Assume that the activated sensors transmit their inference about the analyte and its concentration to a base station. Let  $\alpha$  denote the fraction of sensors that decide to activate themselves. If too few biosensors decide to activate themselves, then the combined information from the sensors at the base station is not sufficiently accurate. (Assume that the base station averages the measurements of the sensors—so the more sensors that transmit, the lower the variance and the more accurate the inference). If too many sensors activate themselves, then network congestion (assuming some sort of

multiaccess communication scheme) results in wasted battery energy. Also since the electrode in a biosensor cannot be reused once analyte binds to it, deploying too many biosensors reduces the effectiveness of the sensor network for future high-threat measurements. How should each biosensor decide in a decentralized manner whether or not to switch to the High\_Res mode to maximize its utility  $X + f(\alpha)$ ?

**Aim** Suppose each biosensor  $i$  uses the following simple threshold activation policy:

If measurement  $y^{(i)} < y^*$ , remain in “Low\_Res” mode.

If measurement  $y^{(i)} \geq y^*$ , switch to “High\_Res” mode.

Here  $y^*$  is a threshold value that is either specified by design or computed by each sensor. Figure 26.8 illustrates such a threshold policy implemented at a sensor  $i$ .

Such a threshold activation policy is completely decentralized and straightforward to implement. Our goal below is to show under what conditions in a dense sensor network, such a simple threshold policy is optimal (or more specifically a Nash equilibrium). We will use the theory of global games to derive these conditions.

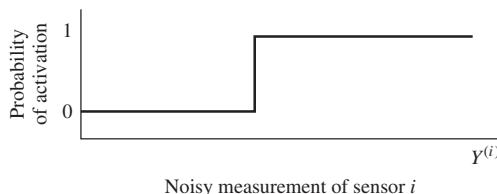
The theory of global games was first introduced in [12] as a tool for refining equilibria in economic game theory; see [13] for an excellent exposition. The term *global* refers to the fact that players at each time can play any game selected from a subclass of all games, which adds an extra dimension to standard game-play (wherein players act to maximize their own utility in a fixed interactive environment). Global games model the incentive of sensors (players) to act together or not. The incentive of a sensor to act is either damped or stimulated by the average level of activity of other sensors (which we denoted as  $\alpha$  above). This is typical in a sensor network where one seeks a trade-off between the cost of measurement (reusability of a sensor) and accuracy of measurement. The results of this section hold for quite general classes of sensor networks; see [14].

#### 26.4.1 Global Game Formulation

Based on the above discussion, each biosensor chooses action (the notation  $u$  used in this section is unrelated to  $u$  in Section 26.2 and Section 26.3).

$$u \in S = \{\text{High\_Res}, \text{ Low\_Res}\}. \quad (26.13)$$

Assume that a sensor only transmits data when in the activated mode.



**Figure 26.8** Threshold policy implemented at sensor  $i$ . “0” denotes the Low\_Res mode and “1” denotes the High\_Res mode.  $Y^{(i)}$  denotes the noisy measurement of the importance of the data (concentration in this case).

**Sensor Class** To allow for sensor diversity, we consider multiple *classes* of biosensors. We assume that each of the sensors can be classified into one of  $I$  possible classes, where  $I$  denotes a positive integer. Let  $\mathcal{I} = \{1, 2, \dots, I\}$  denote the set of possible sensor classes. Let

$$r_J \text{ denote the proportion of sensors of class } J \in \mathcal{I}, \text{ so } \sum_{J \in \mathcal{I}} r_J = 1. \quad (26.14)$$

All sensors of a given class  $J \in \mathcal{I}$  are assumed to be functionally identical, having the same measurement noise distribution and utility function. Since the noise distribution and reward parameters will never be known precisely, it is reasonable to perform this type of classification; it is simply a rough division of sensors based on the specific analytes or cocktail of analytes they can detect.

**Environment Quality  $X$  and Estimate  $Y$**  Let  $X$  denote the actual concentration of analyte. We assume that each sensor can obtain an unbiased measurement of  $X$ , although the quality of these measurements may vary between sensors due to variable noise conditions. Sensor  $i$ 's estimate of  $X$  can be written as

$$Y^{(i)} = X + W^{(i)}, \quad (26.15)$$

where  $W^{(i)}$  is a random variable representing measurement noise (assumed independent between sensors). We assume all sensors in a given class  $J \in \mathcal{I}$  have the same prior probability distribution for  $X$  and noise distribution. So with  $p_{W_J}(w)$  denoting the measurement noise density at all sensors in class  $J$ , we have

$$\text{for all } i \in J, p_{Y^{(i)}|X,i}(y|x) = p_{Y^{(i)}|X,J}(y|x) = p_{W_J}(y - x). \quad (26.16)$$

Define the cumulative distribution function of the noise at class  $J$  sensors as  $\Phi_{W_J}(w)$ . Denote the noise variance of class  $J$  sensors as

$$\sigma_J^2 = \text{var}\{W^{(i)}\}, \quad i \in J, \quad J \in \mathcal{I}. \quad (26.17)$$

**Proportion of “Active” Sensors**  $\alpha$  Let  $\alpha_J$  represent the proportion of sensors of class  $J \in \mathcal{I}$  that are active, that is, in high-resolution mode, at a given time. Define the activity vector profile

$$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_I), \quad \alpha_J \in [0, 1], \quad J \in \mathcal{I}. \quad (26.18)$$

Thus the proportion of all active sensors, which we denote as  $\alpha$  is  $\alpha = \sum_{J=1}^I r_J \alpha_J \in [0, 1]$ .

**Reward Function and Mode Selection Policy of Each Sensor** The task of the sensor network is to ensure that its end users are sufficiently informed of the environment, without expending more effort than required. We consider an autonomous decision model, where each sensor supports this goal by independently deciding whether to pick action  $u \in S = \{\text{High\_Res, Low\_Res}\}$ , based only on its measurement  $Y^{(i)}$ .

This reduces costly active message passing for coordination between sensors.

Given the activity vector profile  $\alpha$ , each sensor  $i$  chooses action  $u \in \{\text{High\_Res}, \text{Low\_Res}\}$  so as to maximize the expected value of a local reward:

$$C^{(i)}(X, \alpha, u) = \begin{cases} h_J(X, \alpha) = X + f_J(\alpha), & \text{if } u = \text{High\_Res} \\ 0, & \text{if } u = \text{Low\_Res} \end{cases}, \quad \forall \text{ sensors } i \in \text{ class } J. \quad (26.19)$$

That is, we assume that the reward function of each individual sensor  $i$  in class  $J$  is the same; and  $f_J(\alpha)$  is the reward earned by each sensor in class  $J$  when the proportion of active sensors is  $\alpha$ ; details are given below. We assume  $f_J(\alpha)$  is continuously differentiable with respect to each  $\alpha_J$ ,  $J \in \mathcal{I}$ .

Given its observation  $Y^{(i)}$  in (26.15), the goal of each sensor  $i$  is to execute a (possibly randomized) strategy to optimize its local reward. That is, sensor  $i$  seeks to compute policy

$$\mu^{(i)} : Y^{(i)} \rightarrow S \text{ to maximize } E[C^{(i)}(X, \alpha, \mu)], \quad (26.20)$$

where  $\mu = \{\mu^{(i)} : i = 1, 2, \dots\}$  is a collection of strategies of all sensors.

The term  $f_J(\alpha)$  in (26.19) represents the trade-off between network transmission throughput versus global mean-square error. Typically,  $f_J(\alpha)$  is chosen so as to increase for small values of  $\alpha_J$  (close to zero) and decrease for large  $\alpha_J$  (close to one). In [32],  $f(\alpha)$  (assuming a single sensor class) is chosen as a quasi-concave function of  $\alpha$ .

In summary, we have formulated the biosensor activation problem as a global game. Each biosensor (player) can choose one of two actions: Low\_Res mode or High\_Res mode. It chooses its mode by greedily optimizing its utility function defined in (26.19). The vector  $\alpha$  is an interaction term between the different biosensors—it denotes how many biosensors in each class pick the High\_Res mode.

#### 26.4.2 Threshold Biosensor Activation Policies

We are interested in characterizing conditions under which a simple threshold strategy deployed at each biosensor is optimal in a local sense [with respect to (26.20)] and is a Nash equilibrium for the entire network.

**Definition 26.1 Threshold Strategies** *For any sensor  $i$ , let  $y^{(i)}$  denote a realization of the random observation  $Y^{(i)}$ , in (26.15). Then a threshold mode selection strategy  $\mu^{(i)}$  is characterized by*

$$\mu^{(i)}(y^{(i)}) = \begin{cases} \text{High\_Res} & \text{if } y^{(i)} > y^{*(i)}, \\ \text{Low\_Res} & \text{if } y^{(i)} \leq y^{*(i)}. \end{cases} \quad (26.21)$$

Here the constant  $y^{*(i)} \in \mathbb{R}$  is called the threshold or switching point.

For a class of sensors  $J$ , a symmetric threshold strategy  $\mu_J$  is a threshold strategy (26.21) such that all sensors  $i \in J$  have the same switching point  $y^{*(i)} = y_J^*$ .

The fact that we want biosensors to autonomously decide when to activate themselves does not imply that a sensor should switch to “activate” and transmit information

whenever its measurement  $Y^{(i)}$  is sufficiently large, since it may be better to exploit signal correlation by remaining idle and relying on others to act instead. The optimal strategy  $\mu$  depends on the expected behavior of the other sensors' behavior through  $\alpha$ , the proportion of sensors choosing action High\_Res. This in turn depends on the strategy of the other sensors. We are therefore interested in determining a collection of strategies for each sensor that is simultaneously optimal, that is, in Nash equilibrium. (In other words, if a sensor unilaterally departs from a Nash equilibrium it is worse off). Let us now define a Nash equilibrium.

**Definition 26.2 Threshold Nash Equilibrium** *A collection of strategies  $\mu$  is a Nash equilibrium if each  $\mu^{(i)}$  is optimal [in the sense of (26.20)] for sensor  $i$  given activity vector profile  $\alpha$ . If the Nash equilibrium policy for every sensor in class  $J$  is a threshold policy (26.21) with the same threshold point  $y^{*(i)}$  for all sensors  $i \in J$ , then a symmetric threshold Nash equilibrium is obtained.*

If we can prove that the Nash equilibrium is comprised of threshold strategies, then the result is of practical importance since each biosensor can implement its optimal activation policy straightforwardly—only a single threshold parameter  $y_J^*$  needs be computed or specified for each biosensor class  $J \in \mathcal{I}$ . The main aim of the remainder of this section is to determine conditions under which the Nash equilibrium is a collection of symmetric threshold strategies. The proof of existence of such a structured Nash equilibrium profile is in two steps [33]: The first step involves showing that a Nash equilibrium exists among the class of randomized policies. This typically involves the use of an appropriate fixed-point theorem, the Glicksberg fixed-point theorem in our case. The proof of this is identical to that in [32] and is omitted. The second step is to prove that the Nash equilibrium comprises of threshold policies. We focus on this second aspect in the following theorem (proofs are in [14]).

**Theorem 26.1 Optimality of Pure Policies** *Consider each sensor  $i$  in class  $J$  with reward function  $C^{(i)}$  (26.19). Then the following properties hold for the Nash equilibrium strategy  $\mu^{(i)}$  (see Definition 26.2):*

(i)  $\mu^{(i)}$ ,  $i \in J$ , is a pure (nonrandomized) policy with

$$\mu^{(i)}(y^{(i)}) = \begin{cases} \text{High\_Res} & \text{if } E[h_J(X, \alpha)|Y^{(i)} = y^{(i)}] > 0, \\ \text{Low\_Res} & \text{if } E[h_J(X, \alpha)|Y^{(i)} = y^{(i)}] < 0. \end{cases} \quad (26.22)$$

(ii) A necessary and sufficient condition for the Nash equilibrium policy  $\mu_J = \{\mu^{(i)}, i \in J\}$  to be a pure threshold policy in the sense of Definition 26.2 is the following: Suppose for each class  $J$  of sensors, there exists a unique switching point  $y_J^* \in \mathbb{R}$  such that for every sensor  $i \in J$

$$E[h_J(X, \alpha)|Y^{(i)} = y^{(i)}] > 0 \iff y^{(i)} > y_J^*. \quad (26.23)$$

Furthermore the optimal switching point  $y_J^*$  for sensors in class  $J$  is the solution of the functional equation

$$E[h_J(X, \alpha)|Y^{(i)} = y_J^*] = 0, \quad \forall i \in J. \quad (26.24)$$

The first claim of the above theorem states that the Nash equilibrium obtained by maximizing the utility at each sensor  $i$  is a *pure* policy—that is, for a fixed value  $y^{(i)}$ , the optimal action  $u$  is to pick either High\_Res or Low\_Res with probability 1. (In contrast, a *randomized* policy would pick these actions with some nondegenerate probability.) So a graph depicting  $\mu^{(i)}(y^{(i)})$  plotted versus  $y^{(i)}$  consists of piecewise constant segments at values High\_Res or Low\_Res and an arbitrary number of jumps between these constant values. Such a control is often referred to as a “bang-bang” controller in the optimal controls literature [34]. The second claim of the theorem gives further structure to the Nash equilibrium policy, that is, it is a step or threshold function of the observation  $y^{(i)}$ . That is, the graph of  $\mu^{(i)}(y^{(i)})$  only jumps once upwards from Low\_Res to High\_Res. The condition (26.23) means that  $E[h_J(X, \alpha)|Y^{(i)} = y^{(i)}]$  as a function of  $y^{(i)}$  crosses zero only once at some unique point  $y_J^*$  and that this crossing too is upwards, that is, the function is less than zero before  $y_J^*$  and greater than zero after  $y_J^*$ . This *single upcrossing condition*, however, is difficult to verify in practice, apart from the special case of a single sensor class, uniformly distributed noise with uniformly distributed prior considered in [32]. In Section 26.4.4, we will establish the existence and uniqueness of such Nash equilibrium threshold policies for the uniform and Gaussian noise cases.

Clearly, a sufficient condition for a single upcrossing is that the function  $E[h_J(X, \alpha)|Y^{(i)} = y]$  is monotone increasing in  $y$ , and a necessary condition is that at the crossing point  $y_J^*$ , the derivative is positive (so that it is an upcrossing as opposed to a downcrossing). Therefore from Theorem 26.1 we directly have:

**Corollary 26.1** (i) A sufficient condition for a Nash equilibrium to comprise of threshold policies is that  $E[h_J(X, \alpha)|Y^{(i)} = y]$  in (26.27) is monotonically increasing in  $y$ , that is,

$$\frac{d}{dy} E[h_J(X, \alpha)|Y^{(i)} = y] > 0 \text{ for all } y.$$

(ii) A collection of threshold policies with switching points  $y_J^*$ ,  $J \in \mathcal{I}$ , is not a Nash equilibrium if

$$\frac{d}{dy} E[h_J(X, \alpha)|Y^{(i)} = y] < 0 \text{ at } y = y_J^*,$$

where  $E[h_J(X, \alpha)|Y^{(i)} = y^{(i)}]$  is defined in (26.27).

Statement (i) of the above corollary does not yield conditions that can be easily verified apart from the single-class sensor case ( $\mathcal{I} = \{1\}$ ). Statement (ii) will be used to give conditions under which the Nash policy is not threshold.

In summary, we have shown above that if each biosensor deploys a simple threshold strategy as in Definition 26.1 (see also Fig. 26.8) to pick its mode  $u$  (Low\_Res or High\_Res), then the overall sensor network performance achieves a Nash equilibrium. This means that there is no incentive for an individual sensor to unilaterally depart from this threshold policy since if it does so, its utility would decrease. Such competitively optimal behavior of the entire network given simple policies implemented at individual sensors is a pleasing aspect of the global games formulation.

### 26.4.3 How Does a Biosensor Predict Behavior of Other Biosensors?

Implicit in the above formulation, each biosensor needs to predict the activity profile vector  $\alpha$  (proportion of other biosensors that are active), based on its own noisy observations  $Y^{(i)}$  of the environment. The Bayesian prediction of  $\alpha$  and reward (26.27) below are identical to what is done in stochastic control of partially observed systems such as partially observed Markov decision processes. The a posteriori distribution  $p_{X|Y^{(i)}, J}$  in (26.27) below can be viewed as the “information state” [29], and the reward  $E[h_J(X, \alpha)|Y^{(i)} = y^{(i)}]$  in (26.27) is expressed in terms of the information state.

Sensors in the same class  $J \in \mathcal{I}$  use the same threshold strategy  $\mu_J(y_J^*)$  for mode selection (see Definition 26.1). So the proportion of sensors in class  $J$  that choose action High\_Res is equal to the proportion of sensors that receive a signal larger than  $y_J^*$ . Let  $\alpha_J(X)$  denote the proportion of sensors of class  $J \in \mathcal{I}$  selecting action High\_Res (relative to the total number of sensors in class  $J$ ) given the environmental quality  $X$ . Since we are considering an infinite number of sensors that behave independently, by Kolmogorov’s strong law of large numbers,  $\alpha_J(X)$  is also (with probability 1) the conditional probability that a class  $J$  sensor receives signal  $y^{(i)} > y_J^*$  given  $X$ ; see [32] for details. That is,

$$\alpha(x) = (\alpha_1(x), \dots, \alpha_I(x)), \quad \alpha_J(X) = \Pr(y^{(i)} > y_J^* | X), \quad J \in \mathcal{I}. \quad (26.25)$$

In Section 26.4.4 we will often omit the explicit  $X$  dependence of  $\alpha$  for convenience.

**Lemma 26.1** *For a threshold policy with threshold point  $y_J^*$ , the proportion of sensors  $\alpha_J(X)$  in class  $J$  that choose action High\_Res given  $X$  is [where  $\Phi_{W_J}$  below denotes the cdf of the noise at class  $J$ ; see (26.16)]:*

$$\alpha_J(x) = \int_{y_J^*}^{\infty} p_{Y^{(i)}|X, J}(y|x) dy = 1 - \Phi_{W_J}(y_J^* - x), \quad J \in \mathcal{I}. \quad (26.26)$$

(Therefore the proportion of all sensors that choose action High\_Res is  $\alpha(x) = \sum_{J \in \mathcal{I}} r_J \alpha_J(x)$  where  $r_J$  is the proportion of class  $J$  sensors,  $J \in \mathcal{I}$ ).

As a result, the conditional expectation of reward  $h_J$  (26.19) for each sensor  $i \in J$ ,  $J \in \mathcal{I}$  is

$$E[h_J(X, \alpha)|Y^{(i)} = y^{(i)}] = E[X|Y^i = y^{(i)}] + \int_{-\infty}^{\infty} f_J(\alpha(x)) p_{X|Y^{(i)}, J}(x|y^{(i)}) dx. \quad (26.27)$$

*Proof* Given  $X = x$ , the probability of  $y^{(i)} > y_J^*$  for any sensor  $i$  of class  $J$  is  $\int_{y_J^*}^{\infty} p_{Y^{(i)}|X, J}(y|x) dy$ . Then using (26.16), (26.26) follows. Also, (26.27) follows from Bayes’ rule.

To summarize, in this section, we have described how each sensor can predict the mode (behavior) of other sensors based on its own observation. This ability to predict the proportion of active sensors, based only on a noisy signal and assuming rationality of other sensors, is central to the global game approach; see [13, 32]. It implies that sensors need only measure  $Y^{(i)}$ , and not  $\alpha(X)$  in order to make a mode selection decision. This allows sensors to act simultaneously, without a costly coordination and

consultation phase. The conditional prediction of  $\alpha(X)$  (26.26) and the cost (26.27) will be used in the threshold Nash equilibrium results in the next section.

#### 26.4.4 Nash Equilibrium Threshold Strategies for Sensor Activation

The goal of this section is to present conditions that guarantee the existence of simple threshold Nash equilibrium for each sensor. The single upcrossing condition (26.23) of Theorem 26.1 and statement (i) of Corollary 26.1 are sufficient conditions for a threshold Nash policy, but are difficult to verify. In this section we give easily verifiable sufficient conditions under which the threshold strategies for each sensor are Nash equilibria. Such strategies are simple to implement and learn since sensors need only follow a simple threshold rule (26.21) for mode selection. Then for the prior of  $X$  and  $W^{(i)}$  being uniformly distributed (Section 26.4.5), we will examine these conditions.

**Theorem 26.2** *A sufficient condition for a Nash equilibrium to comprise of threshold policies is that the following conditions (a) and (b) hold:*

(a) *The observation probabilities  $p_{W_J}(y - x)$ ,  $J \in \mathcal{I}$  [see (26.16)] for class  $J$  sensors satisfy*

$$\frac{p_{W_J}(y - x)}{p_{W_J}(y - x')} \text{ is increasing in } y \text{ for } x \geq x' \quad (26.28)$$

(b) *The reward  $f_J(\alpha)$  satisfies [where  $\alpha = (\alpha_1, \dots, \alpha_J)$ ]*

$$\sum_{K \in \mathcal{I}} \frac{df_J(\alpha)}{d\alpha_K} \mathbf{I} \left[ \frac{df_J(\alpha)}{d\alpha_K} \leq 0 \right] \max_x p_{W_K}(y_k^* - x) \geq -1, \text{ for all } \alpha_J \in [0, 1], J \in \mathcal{I}. \quad (26.29)$$

*Discussion* A nice feature of conditions (26.28), (26.29) is that they do not depend on the statistics of the prior distribution. Even nicer, the left-hand side of (26.29) is independent of the variable  $x$ ; it only depends on the vector variable  $\alpha = (\alpha_1, \dots, \alpha_I)$  where each component  $\alpha_J \in [0, 1]$ . The maximum of the density function  $\max_x p_{W_K}(y_k^* - x)$  is trivially obtained in most cases. Thus (26.29) can be easily checked for most types of probability density functions. It does not even require computing the posteriori distribution in closed form.

We should stress that (26.28) and (26.29) are sufficient conditions for  $E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}]$  to be increasing in  $y$  and thus for  $E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}]$  to cross 0 upwards at some unique point  $y^{(i)} = y_j^*$ . The proof uses ideas in stochastic dominance and the monotone likelihood ratio. The intuition behind the proof is as follows: It is well known [35] that a sufficient condition for  $E[h_J(X, \alpha) | Y^{(i)} = y^{(i)}]$  to be increasing in  $y$ , is that  $h_J(X, \alpha(X))$  is monotone increasing in  $X$  and  $p_{X|Y^{(i)}, J}(x|y)$  is first order stochastically increasing in  $y^{(i)}$ . Condition (26.29) is a sufficient condition for  $h_J(X, \alpha(X))$  to be increasing in  $X$  (simply by taking derivatives w.r.t.  $X$ ). Condition (26.28) is sufficient for  $p_{X|Y^{(i)}, J}(x|y)$  to be stochastically increasing in  $y$ . To deal with the conditioning with respect to  $Y^{(i)}$  in (26.22), we need to use a stochastic order that remains invariant with respect to conditional expectations. The monotone likelihood ratio order [31] is an ideal choice since it is invariant with respect to conditioning. We also use a nice result developed by Whitt [36] that relates stochastic ordering of likelihood distributions to stochastic ordering of the posterior distributions.

In summary, Theorem 26.2 gave easily verifiable conditions on the noise and utility function that guarantees that it is locally optimal (i.e., a Nash equilibrium) for each biosensor to deploy a threshold policy to pick its mode. Below, we verify these conditions for uniformly distributed noise. In [14], Gaussian noise is also considered.

### 26.4.5 Nash Equilibrium Threshold Strategies for Uniform Observation Noise

This section develops the Nash equilibrium threshold strategy for the case where there is a uniform (noninformative) prior on  $X$ , which is observed in uniform noise. So for each sensor  $i \in J$ ,

$$X \sim \text{Uniform}(-A, A), \quad W^{(i)} \sim \text{Uniform}(-\sqrt{3}\sigma_J, \sqrt{3}\sigma_J), \\ \text{where } A > \sqrt{3}\sigma_J \text{ for all } J \in \mathcal{I}. \quad (26.30)$$

(Choosing  $\pm\sqrt{3}\sigma_J$  implies that the noise variance is  $\sigma_J^2$  as defined in (26.17)). If biosensors are deployed in an unknown environment, there may be no prior knowledge of the distribution of analyte concentration  $X$ , and the distribution of observation noise may be unknown. In this case, one may take a conservative approach, by assuming (26.30) holds. This approach is taken, in the context of players who must decide whether to go to a nightclub (given the quality of music and having decreased utility if the club is either too empty or too crowded) in [32]. However, [32] only allows for one sensor class; we extend the results to multiple classes here. The following is our main result for the threshold Nash equilibrium.

**Theorem 26.3** *Consider the sensor network with local reward (26.19), which defines  $f_J$ , observations (26.15) and uniform noise with diffuse prior (26.30). Let  $r_K$  denote the proportion of sensors of class  $K \in \mathcal{I}$ . Then the collection of threshold strategies for sensors of class  $J \in \mathcal{I}$  is a Nash equilibrium if (where  $\mathbf{I}$  below denotes indicator function):*

$$\sum_{K \in \mathcal{I}} \frac{df_J(\alpha)}{d\alpha_K} \mathbf{I} \left[ \frac{df_J(\alpha)}{d\alpha_K} \leq 0 \right] \frac{1}{2\sqrt{3}\sigma_K} \geq -1, \quad \text{for all } \alpha_K \in [0, 1], K \in \mathcal{I}. \quad (26.31)$$

Here the unique switching point  $y_J^*$  of the threshold strategy (Definition 26.1) satisfies the functional equation

$$y_J^* = -\frac{1}{2\sqrt{3}\sigma_J} \int_{y_J^* - \sqrt{3}\sigma_J}^{y_J^* + \sqrt{3}\sigma_J} f_J(\alpha(x)) dx, \text{ where} \\ \alpha_K(x) = \left[ \frac{x - y_K^* + \sqrt{3}\sigma_K}{2\sqrt{3}\sigma_K} \right]_0^1, J, K \in \mathcal{I} \quad (26.32)$$

The notation  $[x]_0^1$  above denotes the operation  $\min\{1, \max\{0, x\}\}$ .

A collection of threshold strategies (26.21) with switching points  $y_J^*$ ,  $J \in \mathcal{I}$  is not a Nash equilibrium if

$$f_J(\alpha(y_J^* - \sqrt{3}\sigma_J)) - f_J(\alpha(y_J^* + \sqrt{3}\sigma_J)) > 2\sqrt{3}\sigma_J. \quad (26.33)$$

*Remark* In the special case when the reward  $f_J(\alpha)$  depends only on the proportion of all active sensors  $\alpha = \sum_{J \in \mathcal{I}} r_J \alpha_J$ , that is,  $f_J(\alpha) = f_J(\alpha)$ , then (26.31), becomes

$$\frac{df_J(\alpha)}{d\alpha} \geq \frac{-1}{\sum_{K \in \mathcal{I}} \frac{r_K}{2\sqrt{\sigma_K}}} \quad (26.34)$$

Also (26.32) and (26.33) hold with  $\alpha(x)$  replaced by  $\alpha(x) = \sum_{K \in \mathcal{I}} r_K \alpha_K(x)$ .

The threshold strategy switching point  $y_J^*$  above is essentially an indifference point; if a sensor receives a signal  $Y^{(i)} = y_J^*$ , then it will be indifferent between its two modes since it expects to receive an expected reward of zero either way. The equilibrium condition is really that this point is the only zero crossing of the expected reward  $E[h(X, \alpha | Y^{(i)} = y^{(i)})]$  as a function of  $y^{(i)}$ . The solution to Eq. (26.32) is easily obtained numerically, for example, by Newton's method.

To gain some intuition into the above result, note that the larger  $\sqrt{3}\sigma_J$  (or equivalently the noise variance is), the less restrictive the right-hand side of (26.31) becomes. This means that threshold strategies form a Nash equilibrium for sufficiently large noise variance. On the other hand, as  $\sqrt{3}\sigma_J \rightarrow 0$ , the sufficient condition only holds for nondecreasing  $f_J(\alpha)$ . To obtain further intuition, consider the case where the index set  $\mathcal{I} = \{1\}$ , that is, there is only one class of sensors.

**Corollary 26.2** *Under the conditions of Theorem 26.3, for a sensor network consisting of a single sensor class [i.e.  $\mathcal{I} = \{1\}$  in (26.14); also denote  $f_1$  as  $f$ , and  $\sigma_1$  as  $\sigma$ ]. Then (i) A threshold Nash equilibrium strategy with switching point  $y^* = -\int_0^1 f(\alpha) d\alpha$  exists if  $df/d\alpha \geq -2\sqrt{3}\sigma$ . [This follows from (26.32) and (26.34)]. (ii) Threshold strategies are not Nash equilibria if  $f(0) - f(1) > 2\sqrt{3}\sigma$ . [This follows from (26.33)].*

Statement (i) is to be compared to Remark 1 of [32, p. 162] where for the case  $c = 1$  and quasi-concave  $f(\cdot)$ , a sufficient condition for a threshold Nash policy to exist is  $\sup_\alpha f(\alpha) - f(1) \leq 2\sqrt{3}\sigma$ . The interpretation in both cases is that if the noise variance  $\sigma^2$  is large, then a threshold Nash equilibrium exists. Statement (ii) says that if the noise variance is small, then a threshold Nash does not exist. Thus there is a *phase transition* in the global behavior of the system as the variance of the noise decreases. This interpretation results in a fascinating behavior. Another interpretation of the above corollary is that the reward  $f$  cannot fall too much from its beginning to its minimum value or from its maximum to its ending value. In the words of [32], there cannot be too much “congestion” in the system, relative to the noise.

In summary, Theorem 26.3 gives conditions on the utility function that guarantees the existence of a threshold Nash equilibrium for uniformly distributed noise. It also gives an explicit expression for the switching point  $y_J^*$  of the threshold policy. Therefore, these sensor activation policies can be easily implemented for activation control of the biosensor; see Figure 26.1.

## 26.5 DISCUSSION AND EXTENSIONS

This chapter began with a description of the modeling of a novel ion channel biosensor. We then derived results for the optimal input voltage to the biosensor to minimize the covariance of the estimation error. Also a sequential multihypothesis test was proposed

to detect the analyte concentration. Finally, a novel global games-based activation scheme was proposed for a network of biosensors. We gave conditions under which if each biosensor deploys a simple threshold policy, the global behavior of the network achieves a Nash equilibrium. In summary, this chapter has described the interaction, design, and analysis of the various functional blocks depicted in Figure 26.1.

There are several extensions of the results of this chapter that can be pursued as discussed below. The global games paradigm for sensor activation is interesting since it generalizes the traditional paradigm of sensors learning from data to sensors learning from other sensors and data. Extensions of the approach to Gaussian noise and the analysis of phase transitions in the behavior of the Nash equilibrium are studied in [14]. Threshold Nash equilibria also arise in channel access schemes such as multipacket reception protocols [37]. An alternative approach to the global games approach (which deals with Nash equilibria) is to consider the set of correlated equilibria. The set of correlated equilibria (see [38]) describes a condition of competitive optimality between sensors. It can be more preferable in certain applications than the well-known Nash equilibrium since it directly considers the ability of sensors to coordinate their actions. This coordination can lead to higher performance than if each sensor was required to act in isolation; see [39, 40].

When employing antibodies or other well-defined receptors, much of the sensitivity advantages of stochastic sensing occurs when using stochastic detection in conjunction with the spatial analysis across an electrode arrays. Exploiting the arrival statistics of a target molecule on the array, in addition to the stochastic ion current properties within each element, permits a rapid estimate of the spontaneous concentration.

When using a large number of channels, the benefit of measurement redundancy in an array is the classical  $\sqrt{N}$ , where  $N$  is the number of independently read electrodes in the array. It is worthwhile examining if the dynamics of the fluid flow (given by a partial differential equation) of the analyte can be exploited to devise a more efficient detector for the analyte and estimator for its concentration.

## REFERENCES

1. S. H. Chung, O. Andersen, and V. Krishnamurthy (Eds.), *Biological Membrane Ion Channels: Dynamics, Structure and Applications*. Springer-Verlag, 2007.
2. B. Hille, *Ionic Channels of Excitable Membranes*, 3rd ed., Sunderland, MA: Sinauer Associates, 2001.
3. V. Krishnamurthy and S. H. Chung, “Large-scale dynamical models and estimation for permeation in biological membrane ion channels,” *Proc. IEEE*, vol. 95, pp. 853–880, May 2007.
4. D. A. Doyle, J. M. Cabral, R. A. Pfuetzner, A. Kuo, J. M. Gulbis, S. L. Cohen, B. T. Chait, and R. MacKinnon, “The structure of the potassium channel: Molecular basis of  $K^+$  conduction and selectivity,” *Science*, vol. 280, pp. 69–77, 1998.
5. R. Dutzler, E. B. Campbell, M. Cadene, B. T. Chait, and R. MacKinnon, “X-ray structure of a ClC chloride channel at 3.0 Å reveals the molecular basis of anion selectivity,” *Nature*, vol. 415, pp. 287–294, 2002.
6. A. Finkelstein, *Water Movement through Lipid Bilayers, Pores and Plasma Membranes*, New York: Wiley-Interscience, 1987.
7. B. Cornell, V. L. Braach-Maksvytis, L. G. King, P. D. Osman, B. Raguse, L. Wieczorek, and R. J. Pace, “A biosensor that uses ion-channel switches,” *Nature*, vol. 387, pp. 580–583, June 1997.

8. B. Cornell, G. Krishna, P. Osman, R. Pace, and L. Wieczorek, "Tethered bilayer lipid membranes as a support for membrane-active peptides," *Biochem. Soc. Trans.*, vol. 29, no. 4, p. 613, 2001.
9. B. Cornell, "Optical biosensors: Present and future," in *Membrane Based Biosensors*, F. Lighler and C. Taitt (Eds.), Elsevier, 2002, p. 457.
10. G. Woodhouse, L. King, L. Wieczorek, P. Osman, and B. Cornell, "The ion channel switch biosensor," *J. Mol. Recognition*, vol. 12, p. 1, 1999.
11. E. Neher, "Molecular biology meets microelectronics," *Nature Biotechnol.*, vol. 19, Feb. 2001.
12. H. Carlsson and E. van Damme, "Global games and equilibrium selection," *Econometrica*, vol. 61, no. 5, pp. 989–1018, Sept. 1993.
13. S. Morris and H. S. Shin, "Global games: Theory and applications," in *Advances in Economic Theory and Econometrics: Proceedings of Eight World Congress of the Econometric Society*, Cambridge University Press, 2000.
14. V. Krishnamurthy, "Self-configuration in dense sensor networks via global games," *IEEE Trans. Signal Proc.*, 2008 (preprint).
15. F. Separovic and B. Cornell, "Gated ion channel-based biosensor device," in *Biological Membrane Ion Channels*, S. H. Chung, O. Andersen, and V. Krishnamurthy (Eds.), Springer-Verlag, 2007, pp. 595–621.
16. L. Ljung, *System Identification*, 2nd ed., Upper Saddle River, NJ: Prentice Hall, 1999.
17. D. P. Malladi and J. L. Speyer, "A generalized Shirayev sequential probability ratio test for change detection and isolation," *IEEE Trans. Auto. Control*, vol. 44, pp. 1522–1534, Aug. 1999.
18. H. V. Poor, "Quickest detection with exponential penalty for delay," *Ann. Stastist.*, vol. 26, no. 6, pp. 2179–2205, 1998.
19. A. J. Goldsmith and S. B. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," *IEEE Wireless Commun.*, vol. 9, pp. 8–27, 2002.
20. A. B. MacKenzie and S. B. Wicker, "Game theory and the design of self-configuring, adaptive wireless networks," *IEEE Commun. Mag.*, pp. 126–131, Nov. 2001.
21. V. Krishnamurthy, K. Luk, B. Cornell, and D. Martin, "Gramicidin ion channel based nano-biosensors: Construction, stochastic dynamical models and statistical detection algorithms," *IEEE Sensors J.*, vol. 7, no. 9, pp. 1281–1288, Sept. 2007.
22. G. Franklin, D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3rd ed., Englewood Cliffs, NJ: Prentice Hall, 1997.
23. R. J. Elliott and V. Krishnamurthy, "Exact finite-dimensional filters for maximum likelihood parameter estimation of continuous-time linear Gaussian systems," *SIAM J. Control Optimization*, vol. 35, no. 6, pp. 1908–1923, Nov. 1997.
24. V. Krishnamurthy and G. Yin, "Recursive algorithms for estimation of hidden Markov models and autoregressive models with Markov regime," *IEEE Trans. Inform. Theory*, vol. 48, no. 2, pp. 458–476, Feb. 2002.
25. V. Krishnamurthy, "On-line estimation of dynamic shock-error models," *IEEE Trans. Automatic Control*, vol. 35, no. 5, pp. 1129–1134, May 1994.
26. M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes—Theory and Applications, Information and System Sciences Series*, Englewood Cliffs, NJ: Prentice Hall, 1993.
27. S. Ross, *Introduction to Stochastic Dynamic Programming*, San Diego, CA: Academic, 1983.
28. D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Vols. 1 and 2, Belmont, MA: Athena Scientific, 2000.
29. P. R. Kumar and P. Varaiya, *Stochastic Systems—Estimation, Identification and Adaptive Control*, Englewood Cliffs, NJ: Prentice-Hall, 1986.

30. V. Krishnamurthy, "Algorithms for optimal scheduling and management of hidden Markov model sensors," *IEEE Trans. Signal Proc.*, vol. 50, no. 6, pp. 1382–1397, June 2002.
31. V. Krishnamurthy and D. Djonin, "Structured threshold policies for dynamic sensor scheduling—A partially observed Markov decision process approach," *IEEE Trans. Signal Process.*, vol. 55, no. 10, pp. 4938–4957, Oct. 2007.
32. L. Karp, I. H. Lee, and R. Mason, "A global game with strategic substitutes and complements," *Games and Economic Behavior*, vol. 60, pp. 155–175, 2007.
33. D. Fudenberg and J. Tirole, *Game Theory*, Cambridge: MA, MIT Press, 1991.
34. D. E. Kirk, *Optimal Control Theory*, Dover, 2004.
35. A. Muller and D. Stoyan, *Comparison Methods for Stochastic Models and Risk*, Hoboken, NJ: Wiley, 2002.
36. W. Whitt, "Multivariate monotone likelihood ratio and uniform conditional stochastic order," *J. Appl. Prob.*, vol. 19, p. 695–701, 1982.
37. M. Ngo and V. Krishnamurthy, "Game theoretic cross layer transmission policies in multipacket reception wireless networks," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1911–1927, May 2007.
38. R. J. Aumann, "Correlated equilibrium as an expression of bayesian rationality," *Econometrica*, vol. 55, no. 1, pp. 1–18, 1987.
39. V. Krishnamurthy, M. Maskery, and M. Ngo, "Game theoretic activation and transmission scheduling in unattended ground sensor networks: A correlated equilibrium approach," in *Wireless Sensor Networks: Signal Processing and Communication Perspectives*, A. Swami, Q. Zhao, Y. W. Hong, and L. Tong (Eds.), Hoboken, NJ: Wiley, 2007.
40. V. Krishnamurthy, M. Maskery, and G. Yin, "Decentralized activation in a ZigBee-enabled unattended ground sensor network: A correlated equilibrium game theoretic analysis," *IEEE Trans. Signal Process.*, 2008 (preprint).



## CHAPTER 27

---

# Biochemical Transport Modeling, Estimation, and Detection in Realistic Environments

Mathias Ortner and Arye Nehorai

Department of Electrical and Systems Engineering, Washington University, St. Louis, Missouri

### 27.1 INTRODUCTION

*New Numerical Approach* We present<sup>1</sup> in this chapter a new approach for computing and using a numerical forward physical dispersion model relating the source to the measurements given by an array of biochemical sensors in realistic environments. The approach presented here provides a modeling framework that accounts for complex geometries and allows full use of software-simulated random wind turbulence. The key point of our approach is that we decouple the “fluid simulation” part from the “transport computation.” In particular, we show on a simple but realistic example how to incorporate numerical simulations including random effects. The approach we propose is generic enough to incorporate additional random effects, including chemical reactions, temperature effects, and radioactive decaying. The Monte Carlo approach we employ is based on a Feynman–Kac representation and therefore does not require solving the problem on the entire domain but only at the sensor positions. The required computational time is thereby limited.

*Illustrating Example: Monitoring Biochemical Events* We take monitoring biological or chemical events as an illustrating example. The prospect of a biological or chemical attack is a prominent security issue. Release in a closed space such as subways or buildings could result in especially high doses and impact [3–5]. In our previous work, we presented detection and estimation techniques for simple scenarios where analytical solutions to the transport equations are available (see [6–10]). In [11] we considered numerical solutions given by finite-element methods. Refer to [12–17] for additional work on the subject, including ways of considering the impact of turbulence on urban diffusion.

<sup>1</sup>We originally presented this work in [1] and [2].

**Inverse Problem** Localization of the source is important in order to predict the cloud propagation in space and time and consequently make quick emergency decisions. We consider cases involving multiple sources and propose to use a generic Bayesian approach to solve the inverse problem based on a suitable prior model. This framework allows us to consider multiple sources with unknown initial diffusion times and release intensities. The estimator we consider is the a posteriori probability distribution of the source locations, with a major consequence: In the case of uncertainties due to a lack of measurements or the presence of several diffusive sources, the computed a posteriori distribution shows all relevant hypotheses.

**Detection Framework** We propose a sequential detector. The difficulty of the detection task arises from the unknown parameters. As in real applications the starting time of the spread and the original concentration and location of the initial delivery are unknown, we propose using a sequential generalized likelihood ratio test (GLRT) as advocated in [18]. Such a detector relies on setting a threshold usually fixed by deciding on the average time before a false alarm. We propose in this chapter a bound on this average time and show that the bound is a good approximation using numerical examples.

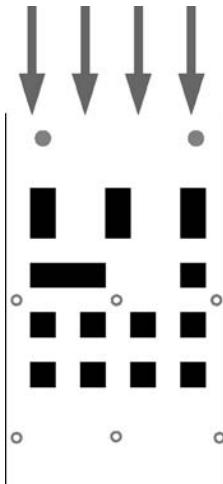
**Chapter Organization** This chapter is organized as follows: in Section 27.2 we detail the physical dispersion model and present the measurement model. In Section 27.3 we present the setup for numerical approximations of the dispersion through Monte Carlo simulations of the associated stochastic diffusion, assuming known arbitrary geometry. We also propose an ad hoc approach to account for stochastic wind turbulence based on the result of numerical fluid computations. In Section 27.4 we explain the Bayesian model employed for localizing several sources and detail the Metropolis–Hastings algorithm used to sample the posterior density. The results from Sections 27.3 and 27.4 are illustrated by a toy example representing an urban environment. In Section 27.5 we develop the sequential detector. In Section 27.5.3, we analyze the false alarm behavior and propose a bound based on geometric considerations. In Section 27.5.4, we detail how performance measures such as the probability of detection can be computed.

## 27.2 PHYSICAL AND STATISTICAL MODELS

In this section we discuss the physical model we use to describe the dispersion of a chemical substance in a fluid. We employ the framework provided by [7, 10]. We also detail the statistical sensor measurement model. The goal is to determine a numerical relationship between the contaminant sources and sensor measurements.

### 27.2.1 Assumptions

We assume the geometry of the setup to be known, including the locations of the biochemical sensors and boundaries of the problem (the building geometries in the case of urban environments). We also assume a knowledge of the wind distribution in the considered setup. For instance, in the presented examples we assume that the wind has a known main direction, and we suppose the availability of software that is capable of computing the resulting wind distribution over the area, including vorticities for



**Figure 27.1** Example of a dispersion scenario: an urban environment (135 m by 225 m) modeled by a set of rectangles. Wind is shown to come from the north at a speed of 54 km/h. The rectangles stand for buildings. Six sensors (shown as circles) are placed in the area, and a chemical substance is instantaneously released at two points (shown as disks).

modeling turbulent effects. We assume also that we know the diffusion properties of the contaminant (diffusion coefficient) and that the sensors have been calibrated, resulting in a known noise variance.

Figure 27.1 shows a toy example illustrating an urban environment that is modeled by a set of rectangles with two release sources and six sensors. Note that the setup in the figure is considered two dimensional (2D) for computational simplicity, although the framework described in the rest of this section is presented in three dimensions (3D).

### 27.2.2 Physical Dispersion Model

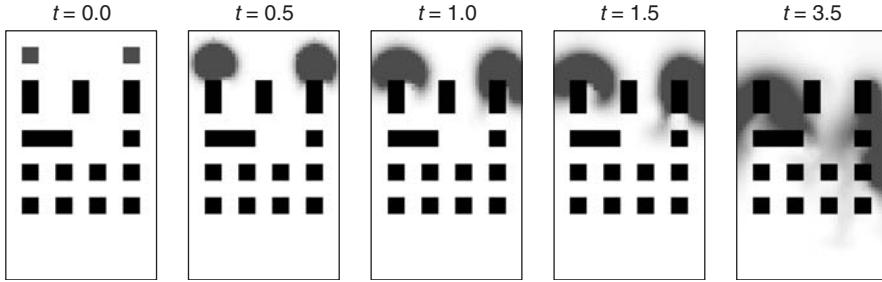
**27.2.2.1 Advection Model** We consider a bounded open domain  $D \subset R^3$ . Let  $\mathbf{r} = (x, y, z)$  be a point in  $D$ . Denote by  $c(\mathbf{r}, t)$  the dispersive substance concentration at a point  $\mathbf{r}$  and time  $t$ . The transport equation in the presence of a wind field  $\mathbf{v}(\mathbf{r}, t) \in R^3$  is given by the following equation [6] when the medium is assumed to be incompressible [ $\text{div}(\mathbf{v}) = 0$ ]:

$$\frac{\partial c}{\partial t} = \text{div}(\mathcal{K}\nabla c) - \nabla c \cdot \mathbf{v} \quad (27.1)$$

where  $\mathcal{K}$  is a  $3 \times 3$  matrix of conduction (or diffusivity). We suppose that  $\mathcal{K}$  is a function of the space variables.

To consider realistic scenarios, we need to use a wind field that includes turbulent effects resulting in either random or time-dependent wind fields  $\mathbf{v}$ . We discuss this point in more detail in Section 27.2.3.1. Figure 27.2 presents a diffusion simulation on the example presented in Figure 27.1.

**27.2.2.2 Boundary Conditions** Let  $\partial D$  be the boundaries of the domain  $D$ . We assume [1] two kinds of domain boundary conditions and divide  $\partial D$  into two disjoint



**Figure 27.2** Simulated diffusion of a contaminant at different time instants computed by dedicated software [19] that includes realistic turbulence computations. The gray level corresponds to the concentration of a contaminant.

subsets denoted by  $\partial D_N$  and  $\partial D_D$  corresponding to Neumann and Dirichlet conditions. The former means  $\nabla c(\mathbf{r}) \cdot \mathbf{n}(\mathbf{r}) = 0$ , for all  $\mathbf{r} \in \partial D_N$  and the latter  $c(\mathbf{r}) = 0$ , for all  $\mathbf{r} \in \partial D_D$ , where  $\mathbf{n}(\mathbf{r})$  is the normal vector at any point  $\mathbf{r}$  belonging to  $\partial D$ . Neumann conditions describe boundaries that do not affect the substance concentration, while Dirichlet conditions correspond to boundaries within the domain  $D$  and the “outside world.”

**27.2.2.3 Initial Substance Distribution and Sources** We assume that the sources have released a certain amount of a substance into the environment when the diffusion begins (instantaneous sources) and denote by  $c_0(\mathbf{r})$  the substance concentration at time  $t = 0$  (release time). This formulation covers most of the usual cases [1], due to the linearity of the advection–diffusion equation.

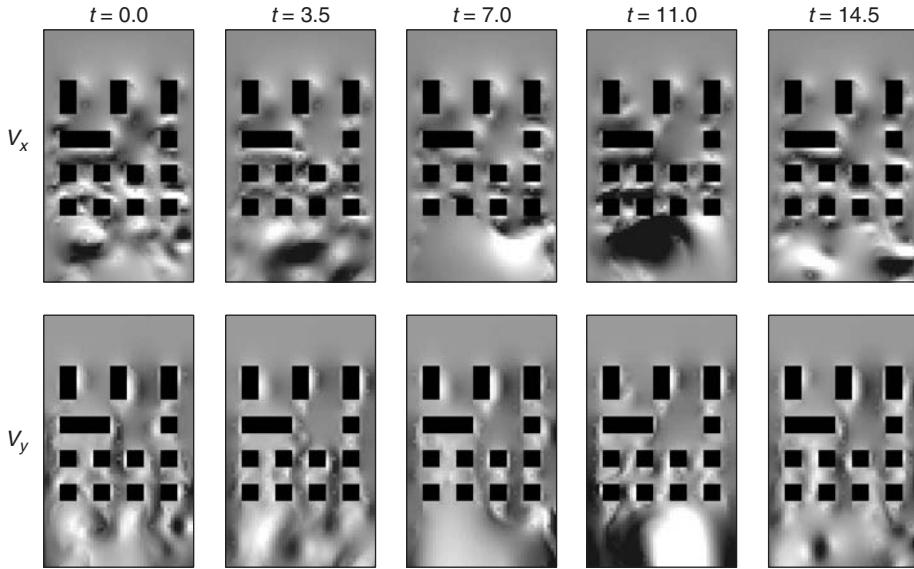
### 27.2.3 Measurement Model

To model the measurements, we suppose a spatially distributed array of  $m$  biochemical sensors located at known positions  $\mathbf{r}_1, \dots, \mathbf{r}_m$ . We assume that each sensor takes measurements at times  $t_0, \dots, t_n$ . Referring to our earlier work [7], we adopt the following measurement model:

$$y(\mathbf{r}_i, t_j) = c(\mathbf{r}_i, t_j) + \epsilon_{i,j}, \quad \epsilon \sim \mathcal{N}(0, \sigma_e^2), \quad i = 1, \dots, m, \quad j = 0, \dots, n \quad (27.2)$$

where  $\epsilon$  represents measurement noise and modeling errors. In the remainder of this chapter, we assume that  $\sigma_e$  is known from a calibration step.

**27.2.3.1 Fluid Simulations, Transport Model, and Inclusion of Random Effects** A major modeling issue in Eq. (27.1) is the assumed knowledge of the transport term  $\mathbf{v}$  due to the wind. The wind actually is a function of both the position and time variables. In urban areas, as shown in [20], random effects generated by wind turbulence may have a huge impact on the dispersal behavior of the biochemical contaminant. Precise knowledge of this variable is impossible to obtain even if many wind sensors are available owing to the chaotic nature of the wind. One way to model the chaotic nature of turbulence is to use a deterministic representation of the wind describing the ergodic effect of random turbulence. Eddy diffusivities that



**Figure 27.3** Temporal snapshots of wind distribution at different times given by dedicated software [19] using the scenario presented in Figure 27.1, with a wind value of 35 km/h. (Top)  $x$  component and (bottom)  $y$  component.

approximate the turbulent effect by a higher diffusion coefficient [21] are a first possible model. Another idea is to use Reynolds decomposition  $\mathbf{v} = \bar{\mathbf{v}} + \xi$  and consider random fluctuations of the wind around the mean [22].

We proposed in our work [1] a different approach, one that allows us to include more realistic wind descriptions for specific scenarios based on wind computation provided by a dedicated software [19]. For instance, in the outdoor problem of Figure 27.1, we suppose that the wind comes from a main direction. We then employ the flow solver to compute the wind distribution over space and time. Figure 27.3 shows some samples of the obtained wind field at different times using this software. In our case, we use the default values provided by the Gerris flow solver. For the scenario of Figure 27.1 (135 m by 225 m) we assume that the wind is coming from the north at a speed of 54 km/h. It results in a mean wind field of 2 km/h horizontally and 62 km/h vertically. The average standard deviation obtained is 60 km/h horizontally and 80 km/h vertically. The numerical forward computation technique we propose allows making full use of these snapshots as detailed in Section 27.3.3.5.

## 27.3 TRANSPORT MODELING USING MONTE CARLO APPROXIMATION

In this section we propose a numerical approach to solve the transport equation (27.1) in the presence of turbulence. The proposed approach is generic and uses random walks to compute the solution of a diffusion equation at a precise location and time. The interesting point is that these random walks are started from the point of interest (e.g., the sensor) and go backward in time with respect to the physical interpretation of the diffusion equation. As a consequence, the problem is only solved at the important

locations (sensor positions), saving computational time. Of course, the random walks depend on the considered diffusion equation and boundary conditions.

For simplicity, in the remainder of this section we will consider a one-dimensional (1D) setup. Extensions to 2D or 3D frameworks are straightforward. Note that in order to present a generic framework, we use specific notations and replace the concentration function  $c$  by  $u$ .

### 27.3.1 Stochastic Diffusion

We present here the general mathematical link between diffusion equations and their associated stochastic process. This link can be seen as the relationship between the macro- and the microscopic descriptions of the physical phenomena.

**27.3.1.1 Diffusion Equation** Consider the following general diffusion operator  $G$  acting on a function  $u(x, t)$ :

$$Gu = \frac{1}{2}\sigma^2(x)\frac{d^2u}{dx^2} + b(x)\frac{du}{dx}. \quad (27.3)$$

We are interested in solving the diffusion equation:

$$\frac{du}{dt} = Gu - Ku, \quad u(x, 0) = u_0(x) \quad (27.4)$$

where  $K$  is a real-valued function that might depend on the variable location  $x$ . In the case of the heat equation,  $K(x)$  stands for an additional cooling or heating effect at  $x$ .

In our biochemical case, the variable of interest is the concentration ( $u = c$ ). The transport term  $b(x)$  then describes the wind effect ( $b = -\mathbf{v}$ ), whereas the diffusion term  $\sigma$  represents the chemical diffusion effect ( $\sigma = \sqrt{2K}$ ). In the case of the chemical diffusion,  $-Ku$  is a source/sink term that represents radioactive decaying, chemical reaction, or incorporates a divergence term for compressible fluids. For the sake of simplicity we assume that the fluid is incompressible  $K = 0$ , although the framework can be applied similarly for nonnull  $K$ .

**27.3.1.2 Stochastic Process** We introduce the following stochastic diffusion process in the Ito sense to represent the microscopic description of Eq. (27.4). As we will detail, this process can be seen as the evolution equation of individual particles going backward with respect to the real physical phenomena. We consider  $X_t$  given by

$$\begin{aligned} X_0 &= x_0 \\ dX_t &= b(X_t) dt + \sigma(X_t) dW_t \end{aligned} \quad (27.5)$$

where  $W_t$  represents a Brownian motion. The position value of  $X_t$  at time  $t$  has a probability distribution  $\mathbf{P}^{x_0,t}(X_t \in \cdot)$  that depends on the initial location  $x_0$  and time  $t$ . We use  $\mathbf{E}^{x_0,t}[\cdot]$  to denote the corresponding expectation family. The relationship between this stochastic process and the diffusion problem described by Eq. (27.4) and (27.3) arises from the property described in Eq. (27.6).

**27.3.1.3 Feynman–Kac Formula** The Feynman–Kac formula relates the continuous macroscopic transport description (27.4) to the stochastic microscopic equation (27.5) through the following property. Under certain regularity conditions on  $\sigma(\cdot)$  and  $b(\cdot)$ , the following function is a solution to the diffusion problem (27.4):

$$u(x_0, t) = \mathbf{E}^{x_0, t} \left[ \exp \left( - \int_0^t K(X_\tau) d\tau \right) u_0(X_t) \right].$$

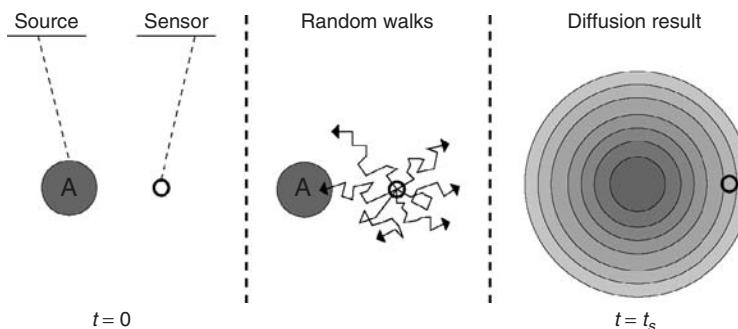
In the case of  $K = 0$  (incompressible flow), we obtain:

$$u(x_0, t) = \mathbf{E}^{x_0, t} [u_0(X_t)]. \quad (27.6)$$

For a given initial condition  $x \rightarrow u_0(x)$ , replacing  $x_0, t$  by a sensor coordinates  $x_s, t_s$  (sensor position and sample time), the Feynman–Kac formula (27.6) asserts that the value of the solution  $u(x_s, t_s)$  at a given location  $x_s$  and time  $t_s$  is given by the expectation of the initial condition  $\mathbf{E}^{x_s, t_s} [u_0(X_{t_s})]$ . This expectation uses the probability distribution  $\mathbf{P}^{x_s, t_s}(X_{t_s} \in \cdot)$  of  $X_{t_s}$ , the location of the process  $(X_t)$  starting from  $x_s$  after time  $t_s$ .

The important point is that the stochastic process is started from the sensor location  $x_s$ : The formula reflects a backward property compared with usual forward computation methods going from the source to the sensor. In our application the transport term  $b$  is given by  $b = -v$  [see Eq. (27.3) and (27.4)]. The process is accordingly drifted by the opposite of the wind and goes backward with respect to the real physical transport phenomena. Another interesting point is that the random walk is independent of the initial condition: One can compute the solutions associated with different initial conditions by using the same set of random-walk samples.

We present in Figure 27.4 an illustration of the Feynman–Kac property. Suppose that the initial condition is given by an indicator function  $u_0(x) = \mathbf{1}(x \in A)$ . The Feynman–Kac formula states that  $u(x_s, t_s) = \mathbf{P}^{x_s, t_s}(X_{t_s} \in A)$ , that is, that the solution



**Figure 27.4** Illustration of the Feynman–Kac formula (27.6). (Left) The initial condition ( $t = 0$ ) is given by the indicator function of set  $A$ , i.e.,  $u_0(x) = \mathbf{1}(x \in A)$ . The disk represents  $A$ ; the circle shows the sensor location  $x_s$ . (Middle) Illustration of the analytical result of a heat equation at  $t = t_s$ . (Right) Realizations of the stochastic process starting from the sensor location  $x_s$ . According to the Feynman–Kac formula, the result of the diffusion in  $x_s$  after time  $t_s$  is given by the probability that the random walks of the last image hit the set  $A$ :  $u(x_s, t_s) = \mathbf{P}(X_{t_s} \in A)$ .

value  $u$  at  $x_s$  and time  $t_s$  is given by the probability that the process  $X_t$  starting from  $x_s$  hits the set  $A$  after time  $t_s$ . Thus, to obtain the solution of the heat equation at a particular location and time, one can launch random walks starting from this point and compute the empirical probability of hitting the set  $A$  after the considered time.

**27.3.1.4 Boundary Conditions** The behavior of the stochastic process depends on the boundary conditions. For Dirichlet boundaries, the process can be killed, or stopped. For Neumann conditions, the process can be reflected and for mixed conditions more generic Feynman–Kac representations need to be considered [23]. We present later how to handle Dirichlet and Neumann conditions.

**27.3.1.5 Approximate Stochastic Diffusion and Sensor Measurements** To implement the Feynman–Kac formula in practice we need to consider simulations of the process  $(X_t)$ . More specifically, we need to simulate the random process  $(X_t)$  starting from  $x$  and then compute the empirical expectation associated with Eq. (27.6). After collecting  $N$  samples  $X_t^1, \dots, X_t^N$  of the process location at time  $t$ , we calculate a natural estimate of  $u(x, t)$  by

$$\hat{u}(x, t)_{u_0} = \frac{\sum_{i=1}^N u_0(X_t^i)}{N}. \quad (27.7)$$

This result demonstrates a direct relationship between the solution at a fixed position and a given time  $(x, t)$  and the initial condition  $x \rightarrow u_0(x)$ . Once a sample  $(X_t^1, \dots, X_t^N)$  has been obtained, the solution  $\hat{u}(x, t)_{u_0}$  can be computed for different functions  $x \rightarrow u_0(x)$  using Eq. (27.7) without requiring further generation of new samples.

In the context of locating a chemical source using an array of  $m$  sensors, we need to launch  $N$  random walks starting from each of the  $m$  sensors in order to obtain the concentration evolution at each of the sensor locations.

The first advantage of the proposed approach to solve the forward problem over other classical numerical tools like finite-element methods is that it allows a stochastic modeling of the wind. Second, this method is highly suitable for parallel computing: The random walks are independent and can be therefore generated by different computers. Another advantage is that this approach computes the solution only at specific points and time of interest and does not try to solve the entire problem. As a consequence, even when we consider a large setup (e.g., a real city), the processing load evolves linearly with the number of sensors and the considered diffusion duration. The major drawback is the needed computational time: The method is more computationally intensive in small setups than finite elements (in practice, we use  $N = 10,000$  or  $N = 50,000$ ). However, as already stated, the approach is particularly useful for large setups, in terms of domain size and dimension. Another point is that the forward simulation can be done ahead of time as we show it in Section 27.3.3.2.

## 27.3.2 Simulation and Convergence

We now focus on the simulations of the stochastic process involved in Eq. (27.5). The discretization of a stochastic process is a research field in itself and has been the subject of a huge body of literature, especially in the past 10 years (see, e.g., [24]).

Depending on the type of process to be approximated, convergence results may exist or not. In our case, the infinitesimal generator is very simple (it essentially corresponds to the heat equation), and one can expect some nice convergence results. However, the boundary conditions complicate the problem. In this framework, we use a reflected stochastic diffusion to model the Neumann boundary conditions. Different schemes have been proposed (e.g., by Lepingle [25] or by Costantini et al. [23]). The weak convergence of the approach we use has been proved only recently by Bossy et al. [26]. We detail later the simulation procedure for the case of the Neumann boundary conditions, while restricting the presentation here to the simple case of infinite domains without boundaries.

*Infinite Domain:* In the case of an infinite domain  $D$  (without any boundary conditions), the following Euler scheme is known to converge in the weak sense [24]. Define a time discretization parameter  $h$  giving the associated discrete times  $t_k = hk \in [0, T]$ , where  $k$  is an integer. Sampling the process described in Eq. (27.5) is straightforward and is obtained by the following iterative procedure:

$$\begin{aligned} X_0^h &= x_0 \\ X_{t_{k+1}}^h &= X_{t_k}^h + b(X_{t_k}^h)h + \sigma(X_{t_k}^h)(W_{t_{k+1}} - W_{t_k}) \end{aligned} \quad (27.8)$$

where  $W_{t_{k+1}} - W_{t_k}$  is simulated by a Gaussian random variable with mean 0 and variance  $h$ . The weak error  $\mathbf{E}[f(X_T^h)] - \mathbf{E}[f(X_T)]$  for  $f$  in a class of smooth functions can be expanded in terms of powers of  $h$ , provided some regularity conditions on  $f$  or some conditions on  $X$  are satisfied [26].

*Neumann Conditions* In the case of an open bounded domain  $D$ , the scheme presented here needs to be adapted. The weak convergence of the following algorithm has been recently provided by Bossy et al. [26]. This algorithm is based on symmetric reflections of the random walk against the boundaries:

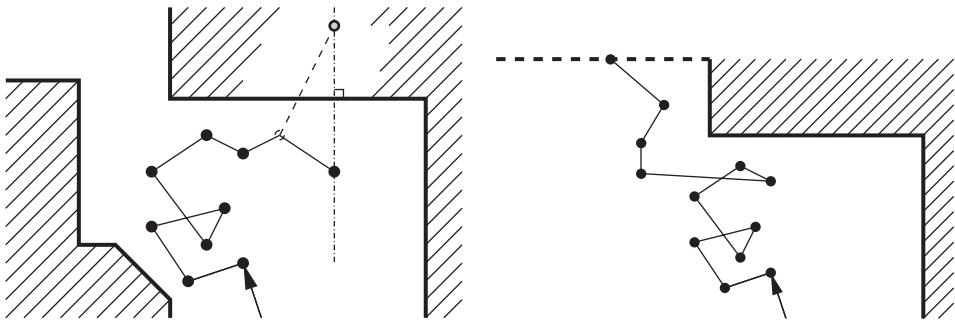
$$X_0^h = x,$$

For  $X_{t_k}^h \in \overline{D}$ ,

$$\begin{aligned} 1: \quad Y_{t_{k+1}}^h &= X_{t_k}^h + b(X_{t_k}^h)h + \sigma(X_{t_k}^h)(W_{t_{k+1}} - W_{t_k}), \\ 2: \quad X_{t_{k+1}}^h &= \begin{cases} S_{\partial D}^\gamma(Y_{t_{k+1}}^h) & \text{if } Y_{t_{k+1}}^h \notin \overline{D} \\ X_{t_{k+1}}^h & \text{otherwise.} \end{cases} \end{aligned} \quad (27.9)$$

Here  $S_{\partial D}^\gamma(x)$  represents the mirror point of the point  $x$  with respect to the boundary  $\partial D$  taken using the  $\gamma$  direction. The vector  $\gamma$  depends on the boundary condition:  $\nabla u(t, x) \cdot \gamma = 0$  for  $(t, x) \in [0, T] \times \partial D$ . In our case  $\gamma(x) = \mathbf{n}(x)$ , resulting in a symmetric reflection (see Fig. 27.5). In order to be consistent, this definition supposes that there are no ambiguities on the border on which to reflect ( $h$  thus needs to be small enough). When a bad case is encountered (the symmetric point lies outside of  $\overline{D}$ ), one can resimulate  $Y_{t_{k+1}}^h$ . Note that according to [26], the probability of such events goes exponentially to zero with  $h$ .

Through conditions on the smoothness of  $D$ ,  $\partial D$ ,  $b$ , and  $\sigma$ , Bossy et al. [26] have shown the following result: for  $f$  smooth enough (see the original work) and compatible



**Figure 27.5** Illustration of Neumann and Dirichlet boundaries. (Left) Trajectory of a reflected process illustrating the numerical scheme used to account for Neumann boundary conditions. In gray, the originally proposed point replaced by its mirror point with respect to the boundary. (Right) Trajectory of an absorbed realization illustrating the effect of a Dirichlet boundary.

with the problem:

$$|\mathbf{E}[f(X_T^h)] - \mathbf{E}[f(X_T)]| \leq \text{const}(T)C(f)h$$

where const is uniform in  $x$  and  $f$ , and  $C(f)$  depends on the sum of  $\infty$  norm of the differential of  $f$  until order 5 (see [26] for details.)

**Dirichlet Conditions** The simplest way [27], to account for Dirichlet boundaries, is to stop the random walk when it hits such a boundary. The Feynman–Kac formula then becomes  $u(x, t) = \mathbf{E}^x[u_0(X_t)\mathbf{1}(t < \tau)]$ ,  $\tau$  being the stopping time associated with the event: “ $X_t$  crosses a Dirichlet boundary.” Note that we denote by  $\mathbf{1}$  the indicator function.

### 27.3.3 Application to Transport Problem

We present here the numerical method for implementing the particular case of our transport problem (27.1). We also present its consequences on the likelihood of the measurements. We propose a way to include turbulent flow within the computations.

**27.3.3.1 Stochastic Transport Model** By identifying the generic diffusion Eq. (27.4) with the transport equation (27.1), we obtain, in 2D or 3D:

$$\mathbf{b}(\mathbf{r}) = -\mathbf{v}, \quad \sigma = \sqrt{2\mathcal{K}}$$

where we used the square root for a definite positive matrix. Note again that the wind is reversed: the Feynman–Kac formula works in a backward mode as discussed earlier. The discrete scheme in Eq. (27.8) results in the following iterations:

$$\mathbf{X}_{t_{k+1}}^h = \mathbf{X}_{t_k}^h - \mathbf{v}(\mathbf{X}_{t_k}^h)h + \sqrt{2\mathcal{K}(\mathbf{X}_{t_k}^h)}(\mathbf{W}_{t_{k+1}} - \mathbf{W}_{t_k}) \quad (27.10)$$

where  $\mathbf{W}_{t_{k+1}} - \mathbf{W}_{t_k}$  is obtained by the generation of two or three (depending on the dimension) independent and identically distributed normal variables, with mean 0 and variance  $h$ . In practice we consider an isotropic and homogeneous matrix and the square root correspond to the scalar one.

**27.3.3.2 Monte Carlo Approximations** To obtain a handy version of the empirical distribution of the particles after time  $h$ , we use a discrete version of the domain  $D$ . Denote by  $\Lambda$  a set of sites in  $D$  associated with a grid of points indexed by  $z$  and  $|\Lambda| = \text{card}(\Lambda)$  the number of elements in  $\Lambda$ . We partition  $D$  into small squares  $\Delta D(z)$  (pixel like):

$$D = \bigcup_{z \in \Lambda} \Delta D(z) \quad \text{and} \quad \Delta D(z) \cap \Delta D(z') = \emptyset, \quad \forall z \neq z'.$$

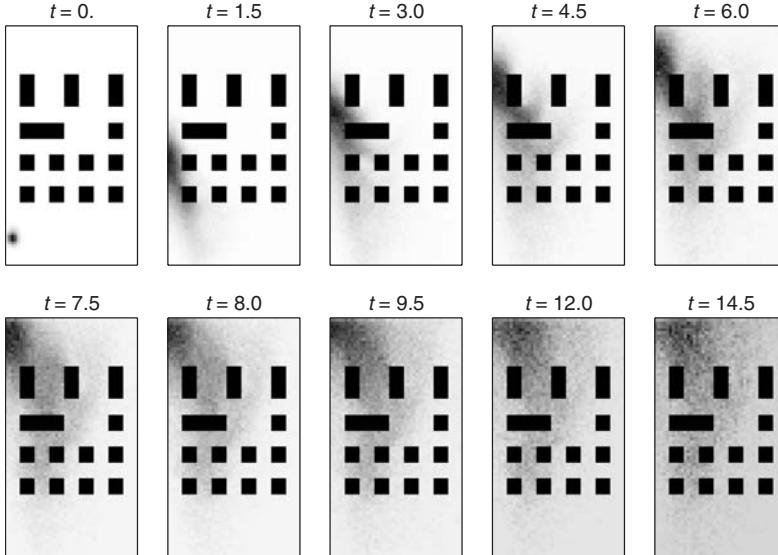
For given sensor location and time  $(\mathbf{r}_i, t_j)$ , diffusion matrix  $\mathcal{K}$ , and wind distribution  $\mathbf{v}$ , the Monte Carlo simulations give  $N$  final points, denoted by  $X_{\mathbf{r}_i, t_j}^1, \dots, X_{\mathbf{r}_i, t_j}^N$ . Let  $p_{i,j,z}$  be the average number of such points falling in the element  $\Delta D(z)$ :

$$p_{i,j,z} = \frac{1}{N} \sum_{k=1}^N \mathbf{1}(X_{\mathbf{r}_i, t_j}^k \in \Delta D(z)).$$

For a given initial concentration value function  $\mathbf{r} \rightarrow c_0(\mathbf{r})$ , the Feynman–Kac formula (27.6) yields

$$c_{i,j} = \sum_{z \in \Lambda} p_{i,j,z} c_0(z) \quad (27.11)$$

where  $c_{i,j}$  is the calculated estimate of the concentration at the location  $\mathbf{r}_i$  and time  $t_j$ . We present two results of the computations of these transport probabilities  $p_{i,j,z}$  on Figure 27.6. For a given sensor ( $i = 1$ , bottom left on Figure 27.1) we illustrate the



**Figure 27.6** Illustration of the  $p_{i,j,z}$  for  $i = 1$  (bottom left sensor in Fig. 27.1) and several successive times  $t_j$ . The mapping  $z \rightarrow p_{i,j,z}$  corresponds to the empirical probability density of the origin  $z$  of a particle arriving at the  $i$ th sensor after a time  $t_j$ . In each image, the higher  $p_{i,j,z}$  the darker the corresponding pixel. The gray scale is adapted for each image, resulting in the overall darkening.

probabilities  $p_{i,j,z}$  for different times  $t_j$ . The figure illustrates the notion of backward evolution, the  $p_{1,j,z}$  indeed standing for the probability that a particle arriving at the first sensor was launched from a site  $z$  at a time  $t_j$  earlier.

**27.3.3.3 Unit Response** Consider a time discretization parameter  $\delta_t$  and regularly spaced discrete times  $(t_0, \dots, t_j, \dots) = (0, \dots, j\delta_t, \dots)$ . The sequence  $(p_{i,j,z})_{0 \leq j \leq n}$  can be seen as the unit response of the sensor located at  $r_i$  to a unit instantaneous substance release at the site indexed by  $z$ . In the remainder of this chapter, we will consider  $\delta_t = 1$  and denote by  $t \in \{0, \dots, n\}$  the time index for the impulse response [see Eq. (27.12)].

**27.3.3.4 Likelihood** Denote by  $\mathbf{Y}$  all the measurements  $y_{i,t}$  lumped into a single  $mn$ -dimensioned vector. Let  $\mathbf{C}$  be the vector of all initial concentration  $c_0(z)$  in every point  $z \in \Lambda$ . By assuming independent measurements and a Gaussian noise, Eq. (27.11) yields the likelihood  $f(\mathbf{Y}/\sigma_e, \mathbf{C})$ :

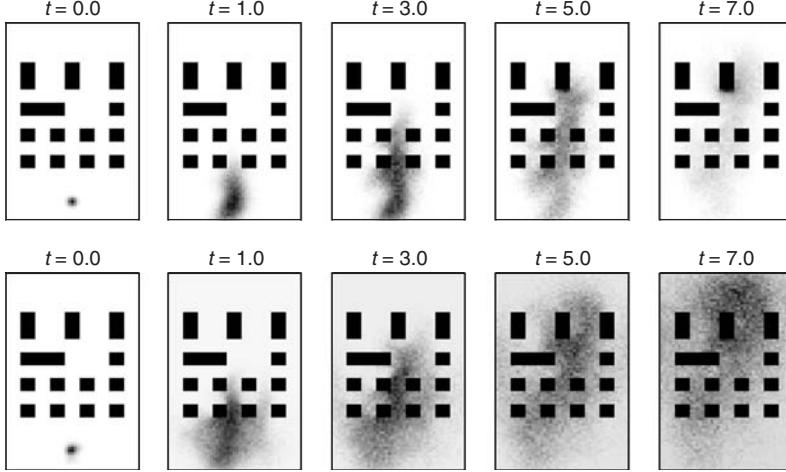
$$f = \frac{1}{(\sqrt{2\pi}\sigma_e)^{mn}} \exp \left[ -\frac{1}{2\sigma_e^2} \sum_{i=1}^m \sum_{t=0}^{n-1} \left( y_{i,t} - \sum_{z \in \Lambda} p_{i,t,z} c_0(z) \right)^2 \right]. \quad (27.12)$$

**27.3.3.5 Wind Turbulence Modeling** We present here an ad hoc procedure to account for wind turbulence.

To illustrate the procedure, we focus on the urban environment example presented in Figure 27.1. We compute a solution to the Navier–Stokes equation for the wind distribution using a dedicated program called Gerris [19]. This computation is made possible because of the assumption that the wind in this example is mainly coming from the north. We then obtain several snapshots of the wind distribution such as those shown in Figure 27.3.

A common way to account for turbulence is to use a mean wind field [20], averaging the obtained snapshots. However, our preliminary results indicated that using a mean field would be inappropriate especially for outdoor applications. For instance, at a location where the turbulence is large, the average wind can be null if the wind direction variability is large enough. Another solution is to decompose the wind into a mean field and a Gaussian random variable [22]. However, even if the turbulence values are chaotic in time, there is a strong spatial correlation between neighboring points. Supposing independence between neighboring points might therefore be incoherent while estimating the correlation might not be a simple issue.

We now propose a simple way of incorporating turbulent flow behavior in the computations of the random walks. We take advantage of the stochastic formulation of the numerical computations we proposed in this section. Let  $\{\mathbf{v}_1, \dots, \mathbf{v}_l\}$  be  $l$  spatial snapshots of the wind distribution over the whole setup, provided by a software. For instance, we use [19] to compute 30 such snapshots under the constant main wind direction assumption. We propose to replace the wind drift  $\mathbf{v}$  in Eq. (27.10) by one snapshot randomly selected among  $\{\mathbf{v}_1, \dots, \mathbf{v}_l\}$ . In order to account for the variability of the turbulent flow, we randomly change the snapshot used during the stochastic random walk (27.5). Each snapshot is used during a random time  $v$  generated according to an



**Figure 27.7** Comparison of the effect of a mean wind field (top) and the proposed stochastic wind approach (bottom) on the transport probabilities  $p_{i,t,z}$ . We present the transport probabilities associated with the second sensor (bottom center, in Fig. 27.1). (Top) Results obtained using a mean wind field obtained using dedicated software [19]. (Bottom) Results obtained using our stochastic wind modeling, which appears to increase the dispersive effect.

exponential distribution with mean  $\lambda$ :

$$\nu \sim f_{\exp}(\nu; \lambda) = \begin{cases} \lambda e^{\lambda \nu} & \nu \geq 0 \\ 0 & \nu \geq 0 \end{cases} \quad (27.13)$$

and once this random duration has expired, we uniformly select a new snapshot of the wind among the  $l$  possible snapshots before generating a new random duration and iterating the process. The parameter  $\lambda$  can be seen as the expected duration of a turbulence flurry, in practice, we took  $\lambda = 1s$ .

During a random walk of duration  $T$ , we use approximatively  $T/\lambda$  different snapshots. The first advantage of this approach comes from the implicitly modeled spatial correlation. The second advantage is that even if turbulence has a null average at one location, we still can account for large wind values. We present a result in Figure 27.7. On top, we show the transport probabilities  $p_{i,t,z}$  computed using a mean wind field approximation. The results show that the main direction of the wind is taken into account since the cloud of possible original locations goes toward the north with time. However, the bottom result obtained using our approach shows that using randomly selected snapshots of the wind increases the predicted diffusivity, a result that is in line with the eddy diffusivity framework [21].

## 27.4 LOCALIZING THE SOURCE(S)

We describe briefly in this section the Bayesian approach we employed [1] for inferring the source location from the measurements. This task is useful for predicting the cloud

evolution in space and time dispersion by applying the transport model to the estimated source(s) location(s).

### 27.4.1 Inverse Problem and Random Field

We develop in this section a setup to introduce a Bayesian regularization term for solving the inverse problem. This Bayesian term allows us to consider distributed and therefore multiple sources. Additionally, the unknown initial time can be considered as a model parameter and be estimated from the measurements.

#### 27.4.1.1 Bayesian Model

*Likelihood* We consider a set of sites  $\Lambda$  and associate with each site an unknown initial concentration value  $c_0(z) = \mu_z$ . The purpose of the source localization is to estimate the initial values  $\mu_z$  using a set of measurements  $\{\mathbf{y}_{t_0}, \dots, \mathbf{y}_n\}$ . In the following, we assume that  $t_0$ , the initial release time, is approximatively known from a detection process. The likelihood of the measurements being given the set of values  $(\mu_z)_{z \in \Lambda}$  is then given by:

$$f\left(\frac{\mathbf{y}}{\sigma_e, (\mu_z)_{z \in \Lambda}}\right) = \frac{1}{(\sqrt{2\pi}\sigma_e)^{mn}} \exp\left[-\frac{1}{2\sigma_e^2} \sum_{i=1}^m \sum_{t=0}^{n-1} \left(y_{i,t_0+t} - \sum_{z \in \Lambda} p_{i,t,z} \mu_z\right)^2\right]. \quad (27.14)$$

*Prior Model* We use the following mixture as a prior model for the  $(\mu_z)$ . We state that  $\mu_z$  should be equal to 0 with a probability  $1 - \rho$  and uniformly distributed in  $[c_{\min}, c_{\max}]$  with a probability  $\rho$ . The prior term can therefore be written as

$$f_{\text{prior}}\left(\frac{(\mu_z)_{z \in \Lambda}}{\rho}\right) = \sum_{z \in \Lambda} (1 - \rho)\mathbf{1}[\mu_z = 0] + \rho\mathbf{1}[\mu_z > 0, \mu_z \in [c_{\min}, c_{\max}]]. \quad (27.15)$$

The mixing parameter  $\rho$  should be chosen according to the size of the domain  $D$  and the number of sites  $|\Lambda|$ . A way to choose  $\rho$  is to make a prior decision about the average surface of the release. In practice, we took  $\rho = 0.01$ , meaning that we state that the source surface is expected to be 1% of the overall domain area.

*Posterior Density* The likelihood of the measurements, the prior model, and Bayes formula result in the following a posteriori distribution:

$$\begin{aligned} f_{\text{post}}((\mu)_{z \in \Lambda} / \mathbf{y}_{t_0}, \dots, \mathbf{y}_n, \sigma_e, \rho, t_0) \\ = C (2\pi\sigma_e^2)^{-q/2} ((1 - \rho)\Upsilon + \rho\Psi) \\ \times \exp\left(-\frac{\sum_{i=1}^m \sum_{t=0}^{n-1} (\sum_{z \in \Lambda} \mu_z p_{i,t,z} - y_{i,t+t_0})^2}{2\sigma_e^2}\right) \end{aligned} \quad (27.16)$$

where  $\Upsilon = \text{card}\{z \in \Lambda: \mu_z = 0\}$  and  $\Psi = \text{card}\{s \in \Lambda: \mu_s > 0\}$ , and  $C$  is the normalizing constant. Note that according to the Bayes formula,  $C$  is the inverse of the expectation of the likelihood under the prior model  $C^{-1} = B = \int f(\mathbf{y}/\boldsymbol{\mu}) f_{\text{prior}}(\boldsymbol{\mu}) d\boldsymbol{\mu}$ . The value of  $B$  is usually called the Bayesian evidence [28] and can be used for model selection.

*Estimator* For each site we consider the following two values:

$$\mathbf{P}_{\text{post}}(\mu_z > 0), \quad \mathbf{E}_{\text{post}}(\mu_z | \mu_z > 0). \quad (27.17)$$

These estimators, respectively, correspond to the posterior probability of having a source at the location  $z$ , and the posterior conditional expectation of the source concentration, knowing that there is a source. The former provides the probability that there was a source at each location, whereas the latter gives the estimated intensity in that case.

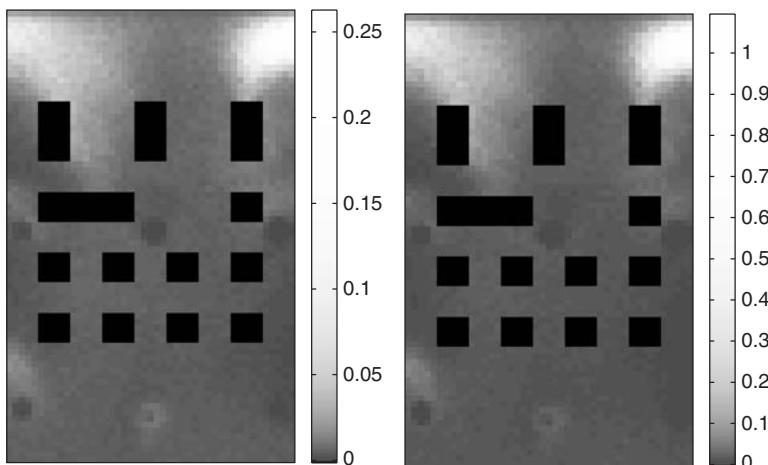
*Estimating Initial Time of Release* Equation (27.16) assumes that the initial time of the transport phenomena is exactly known. However, this will hardly be the case, and we propose for estimating this unknown initial time to compute the Bayesian evidence [ $C^{-1}$  in Eq. (27.16)] for several time hypotheses and then consider the maximum obtained value.

### 27.4.2 Algorithm

We employ [1] a Monte Carlo Markov chain method to sample the posterior distribution and more precisely a Metropolis–Hastings approach. This kind of sampler is especially suitable for our case since we do not know the normalizing constant  $C$  in Eq. (27.16). Note that  $C$  is actually another value we want to compute since it provides the Bayesian evidence.

### 27.4.3 Results

We present 27.8 the result of the sampler. On the left, we show the posterior probability of having a source in each considered location (note that the gray scale is logarithmic).



**Figure 27.8** Results of the source localization estimation given by the Bayesian approach. (Left)  $\mathbf{P}_{\text{post}}(\mu_z > 0)$ , the probability of having a source in each location (logarithmic scale). (Right)  $\mathbf{E}_{\text{post}}(\mu_z | \mu_z > 0)$ , the average source intensity conditioned on the presence of a source. Note that the real initial concentration level used was  $c_0 = 5$ . The result on the left provides for each location the probability that a source was present. The result on the right provides the estimated intensity in the case there was a source.

The true locations of the sources were correctly found. On the right we show the a posteriori expectation of the initial intensity in each location conditioned by the event that there is a release. Note that in the locations where the probability of having a source is high, the estimated intensity is close to the real value (we recall that we used an initial intensity  $c_0 = 5$ ). For that particular example, we assumed the initial time to be known ( $t = t_0$ ). In the following section, we provide a result using the Bayesian evidence, for selecting a relevant initial time hypothesis. These results show that the random-field approach is powerful for finding several sources.

## 27.5 SEQUENTIAL DETECTION

In this section we describe a framework [2] for detecting a biochemical release using the incoming measurements. The goal is to be able to detect a release as soon as possible.

### 27.5.1 Discussion

In sequential analysis, the measurements are considered as an incoming flow, and the goal is to select the hypothesis of interest as soon as possible. In our case, designing such a detector is of the utmost importance since we would like to assess the presence of a dispersive contaminant in a fluid with the smallest possible delay. For a detailed review of sequential detection, see Lai [29].

The problem of detecting a biochemical attack is complicated by the fact that the starting time is unknown. As a result, we need to focus on sequential change detection, which deals with the problem of detecting an abrupt change in the distribution of measurements. This topic has been actively researched in the last decades, both in theory [29–31, 36] and for different applications, including target detection [32] as well as Internet security [33].

Fundamental work on change-point detection has been done by Girshik and Rubin [34], Page [35], Roberts [36], and Shiryaev [37]. The two major competitive procedures mostly used today are the Shiryaev–Roberts–Girshik–Rubin [38] algorithm and Page’s cumulative sum (CUSUM) algorithm [39]. Both are known to be optimal when the observations are independent and identically distributed (i.i.d.) in prechange and postchange distribution.

In our chemical setup, although the change-point detection framework fits our goal well, the usual change-point detectors face some limitations since we do not know the location of the hypothetical release nor its initial concentration. Moreover, the successive measurements are independent but not identically distributed, making analytical results hard to provide.

### 27.5.2 A Sequential Detector

In detection theory, a natural idea when dealing with unknown parameters is to use the likelihood of the best hypothesis under each assumption [40] resulting in GLRT.

We consider three unknown parameters: the initial time  $\delta$ , the location  $s \in \Lambda$ , and the intensity  $\mu$  of an impulse substance source. We assume that the variance  $\sigma_e$  of the noise is known through a calibration step. Let  $\mathbf{y}_t = (y_{1,t}, \dots, y_{m,t})^T$  be the vector of  $m$

measurements given by the  $m$  sensors at time  $t$ . We then obtain the following sequential generalized likelihood ratio:

$$\tilde{L}_n(\mathbf{y}_0, \dots, \mathbf{y}_n) = \max_{0 \leq \delta \leq n} \max_{s \in \Lambda} \sup_{\mu \geq 0} \frac{f_{s,\mu}^1(\mathbf{y}_\delta, \dots, \mathbf{y}_n)}{f^0(\mathbf{y}_\delta, \dots, \mathbf{y}_n)}. \quad (27.18)$$

Denoting by  $\gamma = n - \delta + 1$  the number of measurements available at time  $n$  under the hypothesis that the release occurred at time  $\delta$  and obtain the following expression:

$$f_{s,\mu}^1(\mathbf{y}_\delta, \dots, \mathbf{y}_n) = \frac{1}{(\sqrt{2\pi}\sigma_e)^{m\gamma}} \exp \left[ -\frac{1}{2\sigma_e^2} \sum_{i=1}^m \sum_{t=0}^{\gamma-1} (y_{i,\delta+t} - \mu p_{i,t,s})^2 \right]. \quad (27.19)$$

**27.5.2.1 Sufficient Statistics** Incorporating the maximum-likelihood estimator in the detector expression, we obtain [2] the following ratio value:

$$\begin{aligned} l_n^{\delta,s}(\mathbf{y}_\delta, \dots, \mathbf{y}_n) &= \left( \max \left\{ 0, T_\gamma^{\delta,s}(\mathbf{y}_\delta, \dots, \mathbf{y}_n) \right\} \right)^2, \\ T_n^{\delta,s}(\mathbf{y}_\delta, \dots, \mathbf{y}_n) &= \frac{\sum_{i=1}^m \sum_{t=0}^{\gamma} p_{i,t,s} y_{i,t+\delta}}{\sqrt{\sum_{i=1}^m \sum_{t=0}^{\gamma} p_{i,t,s}^2}}. \end{aligned} \quad (27.20)$$

This expression is well known [40], as it corresponds to a matched filter. In the case of a block detector with known initial release location  $s$  and time  $\delta$ , the distributions of the probabilities of false alarm and detection are straightforward to derive since under both hypotheses the statistic  $T_\gamma^{\delta,s}(\mathbf{y}_\delta, \dots, \mathbf{y}_n)$  is normally distributed. We derived a recursive formulation [2] of the test that is useful in practice.

**27.5.2.2 Resulting Tests** We recapitulate in this section the resulting test. We consider the stopping time  $\tau$ :

$$\tau = \inf\{n \geq 0 \text{ s.t. } L_n \geq \eta\}, \quad L_n = \max_{n-\gamma_m+1 \leq \delta \leq n} \max_{s \in \Lambda} l_n^{\delta,s}$$

where  $\eta$  is the test threshold,  $l_n^{\delta,s}$  is provided by eq. (27.20), and s.t. is “such that.” Alternatively, we note  $L_n = \max_{k \in \mathbb{K}} l^k$  where  $k \in \mathbb{K}$  describe the possible release time–position hypothesis couples  $k = (\delta, s)$  and  $l^k = l^{s,\delta}$ . We now focus on how to select a threshold.

### 27.5.3 Threshold and False Alarm Rate

**27.5.3.1 Average Run Length** In a classical detection framework, the threshold is fixed through the probability of false alarm corresponding to the probability that a positive hypothesis is wrongly selected, that is, the probability that measurements given by the null hypothesis make the statistic cross the threshold. In the sequential detection case like in the Wald test, the threshold is fixed in a similar way, although the false alarm probability is approximated. However, in a change-point detection framework the goal is to keep on testing while new measurements are arriving. Instead of fixing a

false alarm probability, the usual approach is to decide the average run length (ARL) before a false alarm. We denote  $\tau_0$  as the quantity of interest:

$$\tau_0 = \mathbb{E}_{\mathcal{H}_0}[\tau],$$

which is the expected duration before a false alarm.

A naive way of fixing  $\eta$  is to use direct Monte Carlo simulations of  $\tau$  under  $\mathcal{H}_0$ . However, the desired value of  $\tau_0$  is usually very high, and the required number of simulations is therefore huge. Note that, as usual, fixing the test threshold requires a knowledge on the rare-event regime of the test behavior under  $\mathcal{H}_0$ . We provide [2] an analytical result in terms on a bound on the average run length before false alarm.

#### 27.5.4 Performance

In this section we focus on performance measures of the test. We examine the probability of detection and show how to compute the minimum signal intensity level that achieves a desired performance as a function of the release location. We also consider the average delay before detection.

**27.5.4.1 Probability of Detection** The probability of detection  $\mathbf{P}_d$  corresponds to the probability that an attack has been correctly detected. We use a lower bound on the probability of detection for a given positive hypothesis  $\mathcal{H}_1(\delta_1, s_1)$  meaning that an instantaneous release has taken place at time  $\delta_1$  and location  $s_1$ :

$$\mathbf{P}_d(s_1, \eta) \geq \mathbf{P}_{\mathcal{H}_1(\delta_1, s_1)} \left[ \max_{\delta_1 \leq n \leq \delta_1 + \gamma_m} \max_{\delta_1 \leq \delta \leq n} l_n^{\delta, s} \geq \eta \right]. \quad (27.21)$$

**27.5.4.2 Minimum Signal Level** We call minimum signal level the required level to achieve a desired detection performance, under a fixed threshold. Under  $\mathcal{H}_1$ , the distribution of the random variables  $(y_{\delta_1}, \dots, y_n)$  and hence of  $l_n^{\delta, s}$  depends on the unknown intensity of the release  $\mu$ . For instance, for  $\mu = 0$  the probability of detecting the release is low since it corresponds to the probability of false alarm, which is tuned to be very low.

An important issue is then quantifying the ability of the system to detect a release depending on the initial substance concentration. We propose to compute  $\mu_{\min}(s)$ , the minimum signal level in site  $s$  such that a minimum detection performance is achieved for a given experiment duration:

$$\mathbf{P}_d(s, \eta) \geq P_{\min},$$

with, for instance,  $P_{\min} = 95\%$ . The accuracy of  $\mu_{\min}$  depends on the expression used to approximate the detection probability. We discuss that point in our work [2].

**27.5.4.3 Expected Delay Before Detection** The last characteristic of interest is the expected delay between a release event and the actual detection. Let  $\tau_s$  be the stopping time associated with the event for which a release has been correctly detected and  $D(s)$  the associated delay:

$$\tau_{(\delta, s)} = \inf\{n \geq 1 \mid l_n^{\delta, s} \geq \eta_s\}, \quad D(s) = \mathbf{E}_{\mathcal{H}_1(\delta, s)}[\tau_{\delta, s} - \delta \mid \tau < \infty].$$

We use Monte Carlo simulations to compute the expected delay for a given release intensity  $\mu$ . Note that the expression of  $G_s$  is particularly adapted to numerical computations.

### 27.5.5 Simulations

We first present results corresponding to the outdoor setup described in Figure 27.1 with six sensors and two initial release locations. We take the noise as  $\sigma_e = 0.3$ .

**27.5.5.1 Online Detection** In Figure 27.9 we present an example of a detection scenario. The first six rows correspond to measurements by the six sensors. Until time  $t = 90$  the measurements are given by the null hypothesis ( $\sigma_e = 0.3$ ). After time  $t = 90$ , we use the measurements predicted by the model. The last row shows the test statistic  $T$  and the threshold computed to achieve a false alarm rate of  $\alpha = 10^{-5}$ . Note that this simulation included two sources, whereas the detector has been designed under a single-source hypothesis.

**27.5.5.2 Performance Measures** In Figure 27.10 we present the computed performances with  $\sigma_e = 0.3$ . On the left, we present the minimum concentration level  $\mu_{\min}$  to achieve  $P_{\min} = 95\%$  for each location hypothesis. The middle and right pictures show, respectively, the expected delay before detection with  $\mu = \mu_{\min}$  at each location and  $\mu = 20$ . These results show that turbulence has a significant impact on detection performance since in the area within the central buildings a low concentration level of substance will not fit the detection requirement. A release in that area will also need more time, on the average, to be detected.

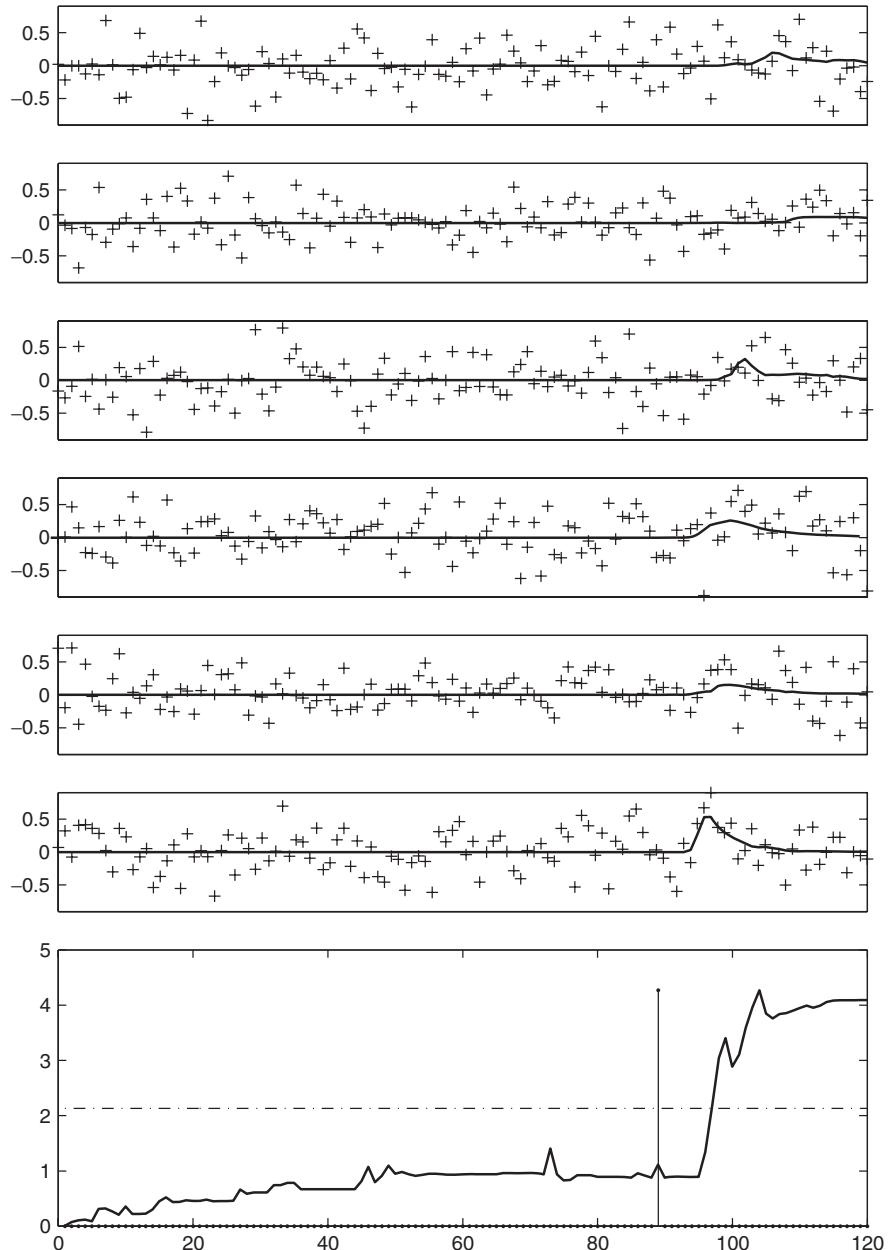
## 27.6 CONCLUSION

We have presented a new way to compute chemical transport equations in realistic environments and proposed a Bayesian framework to solve the inverse problem. The results are potentially useful for array optimal design.

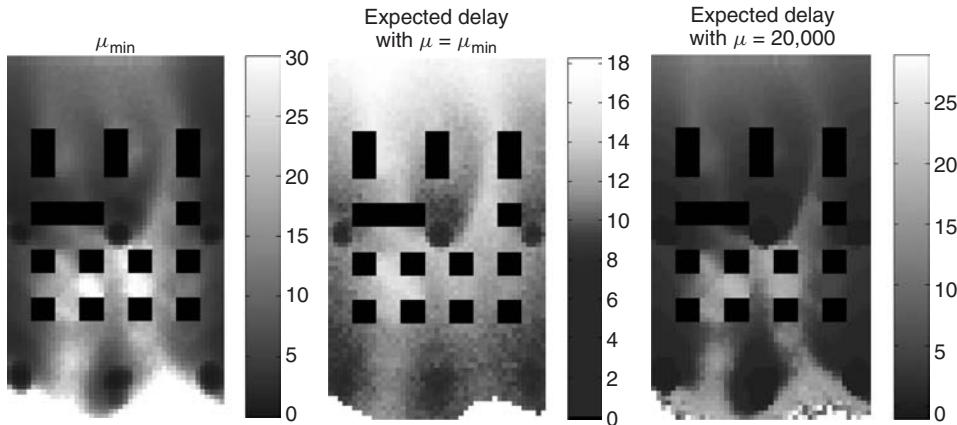
Assuming a main wind direction for the external incoming flow and a known geometry, we developed Monte Carlo simulations of the stochastic process associated with the transport equation. The proposed method allows the inclusion of a realistic stochastic wind distribution accounting for turbulence that proved to be powerful in practice.

We then integrated this method into an array signal processing setup and presented a Bayesian framework to localize the releasing sources, useful for cases where the amount of measurements is too low, resulting from uncertainties concerning the source parameters. The presented framework allows us to localize several sources and to represent uncertainties in the source location. We also provided a way to get an estimate of the initial release time through the Bayesian evidence.

We presented a sequential detector useful for our problem. The sequential detector faces a major challenge. As the initial time and location release are unknown, we obtain a nontrivial expression of the average run length (ARL) before false alarm. We provided a bound on the ARL and showed in Monte Carlo simulations that the bound is useful in practice. We also provided performance measures, such as the minimum release intensity, to achieve a detection probability and the expected delay before detection. These measures may be used for optimal design of the sensor array.



**Figure 27.9** Online detection by an array of sensors illustrated by a simulated release on the framework of Figure 27.1 at time  $t = 90$ . The horizontal axis corresponds to time. The top six figures show the simulated measurements for each of the six sensors (see Fig. 27.1). The noiseless measurements (solid lines) are given by the null hypothesis until  $t = 90$ . At time  $t = 90$ , a chemical diffusion has occurred and we use the measurements given by the diffusion simulation (see Figs. 27.1 and 27.2) augmented with white noise ( $\sigma_e = 0.3$ ). The bottom figure shows the test value (solid line) and threshold (dashed line), computed to achieve a false alarm rate  $\alpha = 10^{-5}$ . The vertical line of the last row ( $t = 90$ ) corresponds to the chemical release instant. The release is detected when the test value is above the threshold ( $t = 98$ ).



**Figure 27.10** Detection performance. (Left) Minimum initial concentration level of a point source required at each location to achieve a detection probability of  $P_{\min} = 95\%$  as a function of the source position. (Middle) Expected delay before detection for each release location hypothesis using  $\mu_s = \mu_{\min}(s)$ . (Right) Expected delay before detection using  $\mu_s = 20$ .

Although the results are presented in the specific framework of biochemical detection, the bound we obtain on the sequential matched filter can be used for different applications.

## REFERENCES

1. M. Ortner, A. Nehorai, and A. Jeremic, "Biochemical transport modeling and bayesian source estimation in realistic environments," *IEEE Trans. Signal Process.*, vol. 55, pp. 2520–2532, June 2007.
2. M. Ortner and A. Nehorai, "A sequential detector for biochemical release in realistic environments," *IEEE Trans. Signal Process.*, vol. 55, pp. 4173–4182, July 2007.
3. "DHS and national academies highlight role of media in terrorism response," available: <http://www.nae.edu/>.
4. J. Fitch, E. Raber, and D. Imbro, "Technology challenges in responding to biological or chemical attacks in the civilian sector," *Science*, vol. 32, pp. 1350–1354, Nov. 2003.
5. H. Banks and C. Castillo-Chavez, "Bioterrorism: Mathematical modeling applications in homeland security," Philadelphia: Society for Industrial and Applied Mathematics, 2003.
6. A. Nehorai, B. Porat, and E. Paldi, "Detection and localization of vapor-emitting sources," *IEEE Trans. Signal Process.*, vol. SP-43, pp. 243–253, Jan. 1995.
7. B. Porat and A. Nehorai, "Localizing vapor-emitting sources by moving sensors," *IEEE Trans. Signal Process.*, vol. SP-44, pp. 1018–1021, Apr. 1996.
8. A. Jeremić and A. Nehorai, "Design of chemical sensor arrays for monitoring disposal sites on the ocean floor," *IEEE J. Ocean. Eng.*, vol. 23, pp. 334–343, 1998.
9. A. Jeremić and A. Nehorai, "Landmine detection and localization using chemical sensor array processing," *IEEE Trans. Signal Process.*, vol. SP48, May 2000.
10. T. Zhao and A. Nehorai, "Detecting and estimating biochemical dispersion of a moving source in a semi-infinite medium," *IEEE Trans. Signal Process.*, 2005, in revision.

11. A. Jeremić and A. Nehorai, "Detection and estimation of biochemical sources in arbitrary 2D environments," in *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Philadelphia, PA, Mar. 2005.
12. A. Gershman and V. Turchin, "Nonwave field processing using sensor array approach," *Signal Process.*, vol. 44, no. 2, pp. 197–210, June 199.
13. A. Pardo, S. Marco, and J. Samitier, "Nonlinear inverse dynamic models of gas sensing systems based on chemical sensor arrays for quantitative measurements," *IEEE Trans. Instrum. Meas.*, vol. 47, no. 3, pp. 644–651, June.
14. H. Ishida, T. Nakamoto, and T. Moriizumi, "Remote sensing and localization of gas/odor source and distribution using mobile sensing system," in *Proceeding of the 2002 45th Midwest Symposium on Circuits and Systems*, Vol. 1, Aug. 2002, pp. 52–55.
15. J. Matthes, L. Groll, and H. Keller, "Source localization based on pointwise concentration measurements," *Sensors Actuators A: Phys.*, vol. 115, no. 1, pp. 32–37, Sept. 2004.
16. H. Niska, M. Rantamäki, T. Hiltunen, A. Karppinen, J. Kukkonen, J. Ruuskanen, and M. Kolehmainen, "Valuation of an integrated modelling system containing a multi-layer perceptron model and the numerical weather prediction model hirlam for the forecasting of urban airborne pollutant concentrations." *Atmospher. Environ.*, vol. 39, no. 35, pp. 6524–6536, 2005.
17. A. Venestanos, T. Huld, P. Adams, and J. Bartzis, "Source, dispersion and combustion modeling of an accidental release of hydrogen in an urban environment." *J. Hazard. Mater.*, vol. 105, pp. 1–25, 2003.
18. H. Chang and T. Lai, "Importance sampling for generalized likelihood ratio procedures in sequential analysis," *Sequential Analysis*, to appear.
19. "Gerris Flow Solver," available: <http://gfs.sourceforge.net>.
20. K. Radics, J. Bartholy, and R. Pongrácz, "Modeling studies of wind field on urban environment," *Atmospher. Chem. Phys. Discuss.*, vol. 2, 2002.
21. K. P. U. D. Baldocchi, T. Meyers, and K. Wilson, "Correction of eddy-covariance measurements incorporating both advective effects and density fluxes source," *Boundary-Layer Meteorol.*, vol. 97, no. 3, pp. 487–511, 2000.
22. J. Anderson, *Fundamentals of Aerodynamics*, New York: Mc Graw-Hill, 1984.
23. C. Costantini, B. Pachiarotti, and F. Sartoretto, "Numerical approximation for functionals of reflecting diffusion processes," *SAM J. Appl. Math.*, vol. 58, no. 1, pp. 73–102, 1998.
24. D. Talay, "Simulations of stochastic differential systems," in *Probabilistic Methods in Applied Physics*, Series Lecture Notes in Physics 451, Springer Verlag, 1995.
25. D. Lépingle, "Euler scheme for reflected stochastic differential equations," *Math. Comput. Simul.*, vol. 38, pp. 119–126, 1995.
26. M. Bossy, E. Gobet, and D. Talay, "Symmetrized Euler scheme for an efficient approximation of reflected diffusions," *J. Appl. Prob.*, vol. 41, no. 3, pp. 877–889, 2004.
27. E. Gobet, "Efficient schemes for the weak approximation of killed diffusions," *Stochastic Process. Applicat.*, vol. 7, pp. 167–197, 2000.
28. J. Ruanaidh and W. Fitzgerald, *Numerical Bayesian Methods Applied to Signal Processing*, Statistics and Computing, New York: Springer, 1996.
29. T. Lai, "Sequential analysis: Some classical problems and new challenges," *Statist. Sinica*, vol. 11, pp. 303–408, 2001.
30. A. Tartakovskiy, "Asymptotic properties of CUSUM and Shiryaev's procedures for detecting a change in a nonhomogeneous Gaussian process," *Math. Methods Statist.*, vol. 4, no. 4, 1995.
31. A. G. Tartakovskiy, "Asymptotic optimality of certain multihypothesis sequential tests: Non-iid case," *Statist. Inf. Stochast. Process.*, vol. 1, pp. 265–295, 1998.

32. A. Tartakovsky, S. Kligys, and A. Petroc, "Adaptative sequential algorithms for detecting targets in a heavy IR clutter," in *SPIE Proceedings: Signal and Data Processing of Small Targets*, Vol. 3809, Denver, CO, 1999.
33. B. Blažek, H. Kim, B. Rozovskii, and A. Tartakovsky, "A novel approach to detection of 'denial-of-service' attacks via adaptive sequential and batch-sequential change-point detection methods," in *IEEE Workshop on Information Assurance and Security United States Military Academy West Point*, June 2001.
34. M. Girshik and H. Rubin, "A bayes approach to a quality control model," *Ann. Math. Statist.*, vol. 23, pp. 114–125, 1952.
35. E. S. Page, "A test for a change in a parameter occurring at an unkown point," *Biometrika*, 1955.
36. S. Roberts, "Control chart tests based on geometric moving averages," *Technometrics*, vol. 1, pp. 239–250, 1959.
37. A. Shiryaev, "On optimum methods in quickest detection problems," *Theory Prob. Its Appl.*, vol. 8, pp. 22–46, 1963.
38. M. Pollack, "Optimal detection of a change in distribution," *Ann. Statist.*, vol. 13, pp. 206–227, 1986.
39. G. Lorden, "Procedures for reacting to a change in distribution," *Ann. Math. Statist.*, vol. 42, pp. 1897–1908, 1971.
40. S. M. Kay, *Fundamentals of statistical signal processing*, Vol. II: *Detection Theory*, Upper Saddle River, NJ: Prentice Hall, 1998.
41. A. S. Monin and A. M. Yaglom, *Statistical Fluid Mechanics*, Cambridge MA: MIT Press, 1975.
42. G. Winkler, *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction*, Springer-Verlag, 2003.
43. C. Robert and G. Casella, *Monte Carlo Statistical Methods*, New York: Springer-Verlag, 1999.
44. H. Pikkarainen, "State estimation approach to nonstationary inverse problems: Discretization error and filtering problem," *Inverse Problems*, vol. 22, pp. 365–379, 2006.
45. S. Roberts, "Control chart tests based on geometric moving averages," *Technometrics*, vol. 1, pp. 239–250, 1959.
46. A. Willsky and H. Jones, "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems," *IEEE Trans. on Automat. Control*, pp. 108–112, Feb. 1976.
47. M. Basseville and A. Benveniste, "Design and comparatice strudy of some sequential jump detection algorithms for digital signals," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-31, pp. 521–535, June 1983.



## CHAPTER 28

---

# Security and Privacy for Sensor Networks

Wade Trappe<sup>1</sup>, Peng Ning<sup>2</sup>, and Adrian Perrig<sup>3</sup>

<sup>1</sup>Rutgers University, North Brunswick, New Jersey

<sup>2</sup>North Carolina State University, Raleigh, NC

<sup>3</sup>Carnegie Mellon University, Pittsburgh, PA

### 28.1 INTRODUCTION

Remote sensing applications are becoming an increasingly important area for research and development due to the critical need for applications that will perform environmental monitoring, provide security assurance, assist in health-care services, and facilitate factory automation. Over the past decade there has been a concerted effort to develop networks of sensors that can monitor different phenomena and drive these remote sensing applications. The sensors in the sensor network make measurements, such as local temperature or barometric pressure, and communicate this data with appropriate applications via the sensor network.

Unfortunately, these sensor systems are typically deployed in unattended environments, without explicit forms of physical protection. As a result, providing security mechanisms for sensor networks is of critical importance since sensors will ultimately be used to assist in our daily lives. Since sensor systems involve applications connected to a network of devices that monitor physical phenomena, there is a broad array of threats that may be posed against sensor applications. These attacks may range from threats that seek to corrupt the basic processes of measurement (and hence affect the reliability of application-level decisions that use such information), to attacks targeted at the basic link-level wireless connectivity, to attacks where an adversary uses knowledge of the routing functionality to its advantage.

These threats may constitute risks to the security and privacy of sensor communications. Since most sensor networks actively monitor their surroundings and share such information over a wireless medium, it is often easy for an adversary to deduce information by monitoring communication traffic that was not intended to be shared outside those users connected to the network. Such unwanted information leakage often results in privacy breaches as the context or information gathered by the sensor network might be sensitive to those administering the network or to entities being monitored by the sensor network.

This chapter will explore a variety of security and privacy challenges facing sensor networks, and then look at defense strategies that may be employed to protect against

these threats. Since the amount of threats is enormous, we will begin by providing a high-level discussion of different attacks or solutions and then focus our discussion on a specific sampling of topics.

## 28.2 SECURITY AND PRIVACY CHALLENGES

We begin our discussion by examining the threats facing sensor networks using the well-known CIA model [1] for security analysis. Specifically, we consider threats that seek to disrupt the (C) confidentiality, (I) integrity, and (A) availability of sensor services. Here, confidentiality is concerned with ensuring that sensor communications and associated information is private and kept confidential from adversaries. Integrity, on the other hand, is related to ensuring that the proper operation of the sensor network and the validity of its messages are maintained. Finally, availability is concerned with guaranteeing that the services provided by the sensor network are maintained in spite of adversarial threats. In all cases, threats facing the sensor network may exist outside the sensor network or come from within the sensor network.

In an outsider attack, the attacker node is not an authorized participant of the sensor network. As the sensor network communicates over a wireless channel, a passive attacker can easily eavesdrop on the network's communications in an attempt to steal private or sensitive information. A broad range of monitoring applications that utilize sensor networks, such as inventory or supply chain management, are particularly attractive for competitors to attempt to monitor. The leakage of strategic information, such as inventory stock, can place companies at competitive advantage or disadvantage. Another outsider threat might involve an adversary altering or spoofing packets, to infringe on the authenticity of communications. Such a threat may seek to convey incorrect sensor readings to a monitoring application. An adversary may also seek to disrupt the communication functionality by transmitting its own interfering signals to jam the network. Another form of outsider attack is to disable sensor nodes. To this end, an attacker can inject useless packets to drain the receiver's battery; he or she may capture and physically destroy nodes. Furthermore, benign node failures may result from nonadversarial factors, such as battery depletion, device faults, and catastrophic climate events. A failed node is often indistinguishable from a disabled node. Therefore, although benign node failure is not really an attack, addressing benign node failures is inseparable from addressing disabled nodes, and thus should be considered as part of a holistic security solution.

Insider attacks involve node compromise, where a sensor node is physically manipulated or reprogrammed to operate in a manner that makes the sensor act in a variety of malicious manners. In contrast to disabled nodes, compromised nodes actively seek to disrupt or paralyze the network. A compromised node may exist in the form of a subverted sensor node (i.e., a captured sensor node that has been reprogrammed by the attacker); or it can be a more powerful device such as a laptop, with more computational and memory resources and a more powerful radio. A compromised node has the following properties:

1. The device is running some malicious code that is different from the code running on a legitimate node and seeks to steal secrets from the sensor network or disrupt its normal functioning.

2. The device has a radio compatible with the sensor nodes such that it can communicate with the sensor network.
3. The device is an authorized participant in the sensor network. Assuming that communication is encrypted and authenticated through cryptographic primitives, the device must be in possession of the secret keys of a legitimate node such that it can participate in the secret and authenticated communications of the network.

We now turn to look at generic confidentiality, integrity, and availability threats. We will describe different defense strategies that have been investigated. Later, we shall select a single confidentiality, integrity, and availability threat and explore these in more detail.

### 28.2.1 Confidentiality-Related Threats

Ensuring confidentiality extends beyond merely performing end-to-end encryption or link encryption, to managing cryptographic keying material and preventing information leakage via traffic analysis. In the context of ensuring that information communicated or gathered by a sensor network is only understood by those intended to understand such information, the objective of achieving confidentiality also includes ensuring the privacy of all parties associated with a sensor system.

Ensuring the secrecy of sensed data is important for protecting data from eavesdroppers. As a starting point, we can use standard encryption procedures (e.g., the AES block cipher) and a shared secret key between the communicating parties to achieve secrecy. Encrypting data is generally not a sufficient solution as it does not provide strong guarantees that the adversary does not learn any information about the plaintext. In order to achieve stronger confidentiality, a system designer should strive for *semantic* security, which provides a guarantee that an eavesdropping adversary cannot reliably answer yes–no questions about an encrypted message. A basic technique to achieve semantic security is randomization: Before encrypting the message with a chaining encryption function (e.g., DES-CBC), the sender precedes the message with a random bit string. This prevents the attacker from inferring the plaintext of encrypted messages if it knows plaintext–ciphertext pairs encrypted with the same key. Such approaches to achieving semantic security were proposed in SPINS [2] and TinySec [3].

A key requirement to supporting encryption services using symmetric cryptography is the distribution of keying material. The general approach employed is to assume that the network is preconfigured with symmetric keys. The challenge with such a solution, though, lies in the fact that having a unique (shared) symmetric key between every pair of nodes requires that the network maintain a number of keys that is quadratic in the amount of nodes in the network. The cost associated with such a solution becomes prohibitive as the size of the network increases. There has been considerable research recently in predistribution schemes where the assumption that each node has a guaranteed shared key with any other node is relaxed. Random key predistribution protocols have been proposed where there is a collection of keys assigned to the network, and each sensor node has a random subset of these keys [4–6]. In such an approach, nodes can only securely communicate if they both have a common shared key, otherwise their communication is not possible. Additional protocol functions, such as revoking keys and updating keys have also been explored in the context of key predistribution. Improved efficiency and performance of key predistribution is possible

if the system designer has knowledge of the topology associated with the deployment of the sensor network [7]. One general concern with key predistribution schemes that employ less keys than needed to ensure a unique key for each pair of communicators is that it may be possible for attackers to compromise a small subset of nodes in order to reconstruct the key pool, and thus an adversary may be able to read all messages crossing the network using an attack against a limited amount of nodes.

More recently, there has been work on supporting public key mechanisms for sensor devices [8, 9]. The computational and communication efficiency of elliptic curve cryptography [10] suggests that elliptic curve cryptography could be a realistic approach to providing public key cryptography (and hence support confidentiality and also digital signatures) on sensor platforms. Current implementations of elliptic curve cryptography, such as TinyECC, have focused on optimizations that make elliptic curve suitable for the MICAz, Imote, and other sensor platforms.

Encryption itself is not sufficient for protecting the privacy of data, as an eavesdropper can perform traffic analysis on the overheard ciphertext, and this can release sensitive information about the data. In addition to encryption, privacy of sensed data also needs to be enforced through access control policies at the base station to prevent misuse of information. Consider, for example, a person locator application. Sensors are implanted in an office building to sense the location of people, and the information is sent to a Web server to answer requests to locate a person. Thus, access control has to be enforced at the Web server to prevent misuse of information by unintended parties. Additionally, by conducting traffic analysis, an adversary may be able to infer the context associated with sensor communications [11, 12].

An adversary armed with knowledge of the network deployment, routing algorithms, and the base station (data sink) location can infer the location of a valuable asset being monitored or the temporal patterns of interesting events by merely monitoring the arrival of packets at the sink, thereby allowing the adversary to remotely track the spatiotemporal evolution of a sensed event. By exploiting the delay-tolerant nature of many sensor networks, it is possible for sensor networks to employ adaptive buffering at intermediate nodes on the source–sink routing path to obfuscate temporal information from an adversary [12]. Other privacy-enhancing mechanisms intended to mask the location of the original source of a sensor communication will be explored in Section 28.5.

### 28.2.2 Integrity-Related Threats

Above the networking layer, the sensor network is usually involved in feeding sensor readings to one or more application-level services. In order to reduce the amount of communication traversing the sensor network, data aggregation is often performed, whereby redundant observations are dropped from the communication flow, and nonredundant data is more succinctly gathered into a smaller amount of messages being routed to the network sink. Data aggregation is one of the most important sensor network services. Since data aggregation and other in-network processing involve sensor nodes collecting readings from neighboring nodes, performing computations prior to forwarding the result, they can be the target of attacks seeking to subvert the integrity of forwarded data. For example, in a temperature-monitoring application involving many temperature sensors connected to an air-conditioning/heating unit, a single corrupt node reporting a large temperature value could significantly throw off the average temperature in the building and thus cause the actuating application to continually run the air conditioner.

In order to overcome such threats, secure aggregation techniques have been proposed [13, 14]. Here, the goal is to obtain an assessment of the measured data's integrity, filter potential outliers, and provide a relatively accurate estimate of the real-world quantity being measured. Additionally, it might be desirable to identify anomalous activity by sensor nodes and blacklist their data at the application. We will examine an architecture for assessing sensor data integrity in Section 28.3. Recently, other sensor applications, such as localization [15] and time synchronization, have been targeted for attacks. Corrupt nodes or corrupt communications can significantly disrupt the proper operation of time synchronization protocols in a sensor network. As another example, there has been significant focus recently on ensuring that sensor localization services are trustworthy [16–18]. Adversaries may seek to disrupt the proper operation of localization services through cryptographic and noncryptographic attacks, and robust localization algorithms have been developed to cope with such threats.

Another serious integrity threat facing sensor networks is the fact that the nodes themselves are susceptible to capture and reprogramming. All internal sensor network attacks originate from the execution of malicious code on a platform. Such malicious code can corrupt routing updates, selectively drop messages, mount slander or framing attacks,<sup>1</sup> collude with other malicious nodes, and the like. Thus, verifying the integrity of code that is executing on a node is a first line of defense that is essential to preventing attacks and ensuring the validity of information crossing the sensor network. For example, the integrity of the forwarding code ensures the integrity of the forwarded packet. Recent work has focused on providing software-based attestation for sensor networks. In SWATT [19], techniques are presented whereby the memory of embedded devices can be checked externally by a verifying entity. The approaches used in SWATT are desirable for sensor networks as SWATT employs a software-only approach to establishing the absence of malicious code or changes in device memory. SWATT does not need physical access to the devices memory, yet provides memory content attestation similar to a trusted computing base without the need for trusted hardware. In fact, software-based attestation methods used in the Pioneer system [20] have been shown to provide a dynamic root of trust on conventional x86 architectures at a level comparable to that provided by AMD SVM and the Intel TXT. As an application of software attestation for sensor networks, recent work presented in [21] has involved porting the Pioneer primitive to the TI MSP430 [the central processing unit (CPU) used in Telos rev.B sensor nodes] and was used to provide trustworthy updates of software within a sensor network [21].

### 28.2.3 Availability-Related Threats

There are a broad range of denial-of-service (DoS) attacks that may be launched against the sensor network. These DoS attacks can be targeted at different layers of the networking stack.

At the most basic level, one may seek to disrupt the connectivity between sensor nodes by exploiting the properties of the physical or medium access control layer. We shall examine such jamming attacks in more detail in Section 28.4. At the networking layer, the attacker can inject additional communications packets in an attempt to flood the network with bogus traffic. Authentication mechanisms, such as those described in [2, 3, 22, 23] can help cope with the injection of false communications into the network. For example, in [23], a thorough protocol description is presented that involves

<sup>1</sup>A slander attack involves a subverted node claiming that a legitimate node is malicious.

the authentication of sensors in a hierarchical sensor network that includes aggregators and gateway nodes that connect to the Internet and provide delivery of sensor data to a monitoring application. A common strategy used to provide authentication in sensor networks is to employ a variation of the TESLA protocol [24, 25] to provide authentication of messages. One concern, though, in using authentication systems employing the principle of delayed-key disclosure is that they are susceptible to DoS attacks targeted at queues that store messages prior to key release. Recent modifications to TESLA have been proposed to cope with such DoS threats [26, 27].

One particularly pernicious threat facing sensor networks are Sybil attacks, where a malicious node illegitimately claims multiple identities [28, 29]. The Sybil attack can be exploited at different layers to cause service disruption. At the MAC layer, by presenting multiple identities the malicious node can claim a dominating fraction of the shared radio resource, so legitimate nodes are left with little chance to transmit. At the routing layer, the Sybil attacker can lure network traffic to go through the same physical malicious entity. Imagine a simple routing protocol where a node chooses an upstream neighbor as the next hop with equal probability. By claiming to be a large number of identities, with high probability a Sybil identity will be selected as the next hop. Therefore, a sinkhole is created and the attacker can hence do selective forwarding. One defense against Sybil attacks is to ensure that entities prove their identity using lightweight authentication mechanisms [28]. Other approaches involve using radio channel assignment or verifying the location of communicating entities [29]. More recently, there has been work on exploiting the properties of multipath propagation to uniquely identify transmitters [30] and may be applied as a mechanism for detecting Sybil attacks when cryptographic tools are not possible.

Another availability threat may be launched against the networking and routing layers. An adversary can mount attacks to disrupt routing availability by exploiting the specifics of a routing protocol to cause packets to not be delivered to the intended recipient [31]. For example, if an adversary has compromised sensor nodes within the network, then the adversary may merely drop packets that are routed through it or may perform selective forwarding.

### 28.3 ENSURING INTEGRITY OF MEASUREMENT PROCESS

Sensors measure values corresponding to underlying physical phenomena. These measurements may be governed by a set of related physical properties. The fact that there are many external factors that influence a sensor reading means that there are many avenues that an adversary may use to cleverly subvert the validity of measured data. *An adversary may cause an intentional change in one property and as a result have an effect on the property the actual measurement is monitoring.* We label these attacks as process of measurement (PoM) attacks.

Let us look at some examples to illustrate the ease with which PoM attacks may be conducted, as well as highlight their potential impact. Beginning with temperature sensors, we may change the tension of the spring coil in a thermostat to affect the temperature at which the switch makes contact, or we may modify the resistance of a thermistor to change the measured value of the temperature. For affecting the wind speed reported by an anemometer, we may adjust the friction of the rotating spindle, thereby causing the amount of revolutions to increase/decrease and alter the calibration

used to infer wind speed. Measurements of time may also be affected, for example, by fluctuating the temperature around a clock crystal to affect clock drift.

At the simplest level, these attacks affect the traffic crossing the sensor network. False event measurements will be propagated in a sensor network, resulting in wastage of network resources. More significantly, false data will ultimately reach an application where this data will corrupt scientific experiments or trigger false actuation. The consequences of action, or inaction, based upon false readings can have implications more dramatic than merely turning on the heater or air conditioning in a room. To cite a real-world example with both economic and environmental impact, recent reports have uncovered a practice of gas station owners to bypass regulatory monitoring systems by relocating sensors installed in gas tanks to detect fuel leaks [32]—essentially, altering the calibration of the sensor system to produce erroneous measurements.

The nature of PoM attacks is quite different from other security attacks on sensor networks. Traditional security and cryptographic protocols are ineffective against PoM threats. Rather, they protect the communication network from attacks, not the environment from attacks. Physically guarding these sensors so as to prevent PoM attacks is not an option either. Deployments of sensor networks are typically in remote or hostile environments and hence call for unattended sensor nodes. Overall, physically guarding or providing constant surveillance of sensors is a solution that simply does not scale and, hence, mechanisms to detect and/or cope with PoM attacks must be part of the sensor network design.

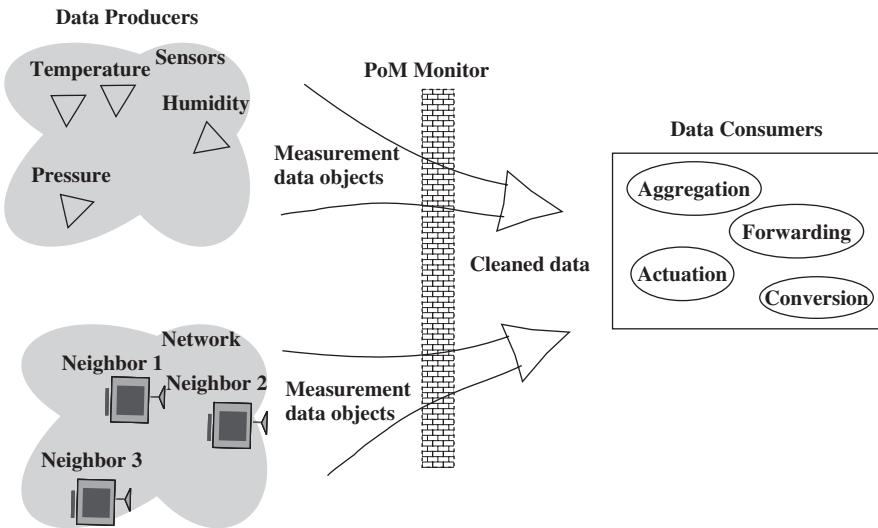
Mechanisms must be integrated into the sensor system to ensure the quality of the measured data. We describe a framework for sensor networks that would provide trustworthy data to applications.

### 28.3.1 Process of Measurement Monitor Overview

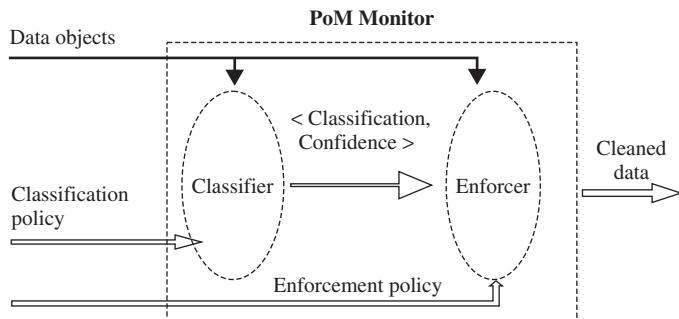
Existing sensor models must include components that will assist the network in guaranteeing the quality of sensor data. Thus, a query to a sensor database should return a quality indicator reflecting the mechanisms in place to assure measurement integrity. This is fundamentally different from traditional databases or data services, where the integrity of *data entry* is never in doubt. To accomplish this, a lightweight data monitoring layer, which we shall called the PoM monitor, can be introduced to detect unreliable measurement data and take appropriate actions to clean the data. A schematic of the envisioned sensor data flow architecture is depicted in Figure 28.1. Such a flow can fit in easily with existing sensor programming models [33].

Since we are only concerned with sensed data, our discussion will only focus on the *data path* in the sensor network. A simplified data path model consists of two entities: *data producers*, such as sensors and other network nodes, and *data consumers*, such as in-network processing functions (data aggregation functions, actuation units, forwarding logic, sensor databases, and the like [34]). The PoM monitor would sit between the data producers and the data consumers, validating the data and cleaning up if necessary. We emphasize that the PoM monitor would only provide mechanisms that verify/clean data, and that the applications would specify which techniques to use or even whether to disable the PoM monitor.

The PoM monitor plays a role analogous to reference monitors from computer security [35] and is comprised of a collection of filtering mechanisms. A modular approach to constructing PoM monitor filters involves threat classification and policy enforcement. The logical flow of a PoM monitor is shown in Figure 28.2. Although we have



**Figure 28.1** Data flow on a sensor node. The PoM monitor is a layer on every sensor node that takes measurement data objects from either sensors or other nodes, classifies data, and filters them to provide clean objects.



**Figure 28.2** PoM monitor has two logical components: the classifier and the enforcer.

depicted separation between classification and enforcement, these two functionalities might actually reside in the same routine. The classifier takes data objects, applies consistency checking, and concludes whether the data is reliable or not. In either case, be it reliable or unreliable, the classifier also publishes an associated confidence level, which may be determined using the underlying classification model. Hence, the classification result is a two-attribute tuple, with the format of  $\langle \text{Classification}, \text{Confidence} \rangle$ . Both the data objects and the classification results are passed to the enforcer, which forwards cleansed data to the data consumer. We note that the introduction of the confidence field may affect the way in which application queries are constructed.

We present an example to show how a simple PoM attack can be launched, and how PoM monitors can defend against these attacks in a realistic sensor scenario. Suppose we deploy five sensor nodes to monitor a room's environment. These nodes are connected to a temperature sensor (which senses the temperature around the node) and a

light sensor (which senses the sunlight level around the node). After the measurements are obtained, they send the sensed data,  $\langle T, S \rangle$ , to the aggregation point. The aggregation node takes all the measured temperatures, calculates the average, and turns on the air conditioner if the average is above a threshold.

In this application, an adversary can launch a simple PoM attack by moving a sensor node close to the window where the node will be exposed to more sunlight than normal. As a result, the node will have a higher temperature and luminance reading than normal. Without the PoM monitor, the high-temperature reading from the manipulated sensor may boost the average at the aggregation point above the threshold, causing the air conditioner to unnecessarily turn on.

The PoM monitor can defend against this PoM attack. Each sensor node would run the PoM monitor. Let us step through the process by first focusing on the attacked sensor node. The data measured by the sensors,  $\langle T, S \rangle$ , are passed to the classifier, which will verify whether these readings reflect the actual physical environment. The classifier can use various ways to verify the data. For example, it can first check the relationship between the temperature and the luminance for any discrepancy between these two variables. Since these two measurements are taken at the same spot, the physical law governing the two phenomena should hold. Next, since temperature is partially dependent on sunlight, the classifier checks whether the luminance measurement reflects the actual sunlight level where the node is supposed to be. In order to do so, the classifier must have some historical data about the typical luminance level of that location, perhaps considering different seasonal trends and the time of the day. If the current measurement is inconsistent with the historical profile, the classifier would conclude that the measurement mechanism of the sensor has been interfered with, and the measurement is not trustworthy. Suppose the classifier assigns a classification result  $\langle H_1, 0.8 \rangle$ , where  $H_1$  signifies an inconsistent measurement, and 0.8 denotes the confidence level of the classifier. The classifier then passes its judgment to the enforcer, which decides what to do with the data. The enforcer has two logical components, a *policy table* and a list of *filters*. A filter specifies the action that can take place on the data, such as purging, cleansing, tagging, and the like. The policy table specifies which filter to use and under what conditions. We note that the policy table is given by the application. Table 28.1 shows an example policy table. Here, according to the table, the enforcer will purge the data, rather than send to the aggregation point.

Similarly, the other four nodes will also clean the data using their PoM monitors. Suppose that their final results are  $\langle 60, H_0, 0.9 \rangle$ ,  $\langle 60, H_0, 0.9 \rangle$ ,  $\langle 63, H_0, 0.9 \rangle$ ,  $\langle 120, H_0, 0.8 \rangle$ , where the first term of the results is the temperature reading. All these results will be fed to the aggregation node. Before the aggregation and actuation take place, the PoM monitor on the aggregation node kicks in. Again, the classifier first checks whether these data are trustworthy. This time it can run an outlier detection algorithm to find out which readings are not in agreement. If the outlier detector marks the reading of 120 as an outlier with confidence 0.95, the classifier can pass the

**TABLE 28.1 A PoM Policy Table**

Condition	Filter
$H_1$ and confidence level $\geq 0.7$	Purging
$H_1$ and confidence level $< 0.7$	Tagging

classification result,

$$\langle [60, 60, 60, 120], [H_0, H_0, H_0, H_1], [0.9, 0.9, 0.9, 0.95] \rangle,$$

to the enforcer. The policy table on the aggregation node may drop the inconsistent reading, namely 120, and send the other three trustworthy readings to the aggregation function, which is the data consumer.

### 28.3.2 Measurement Classifiers

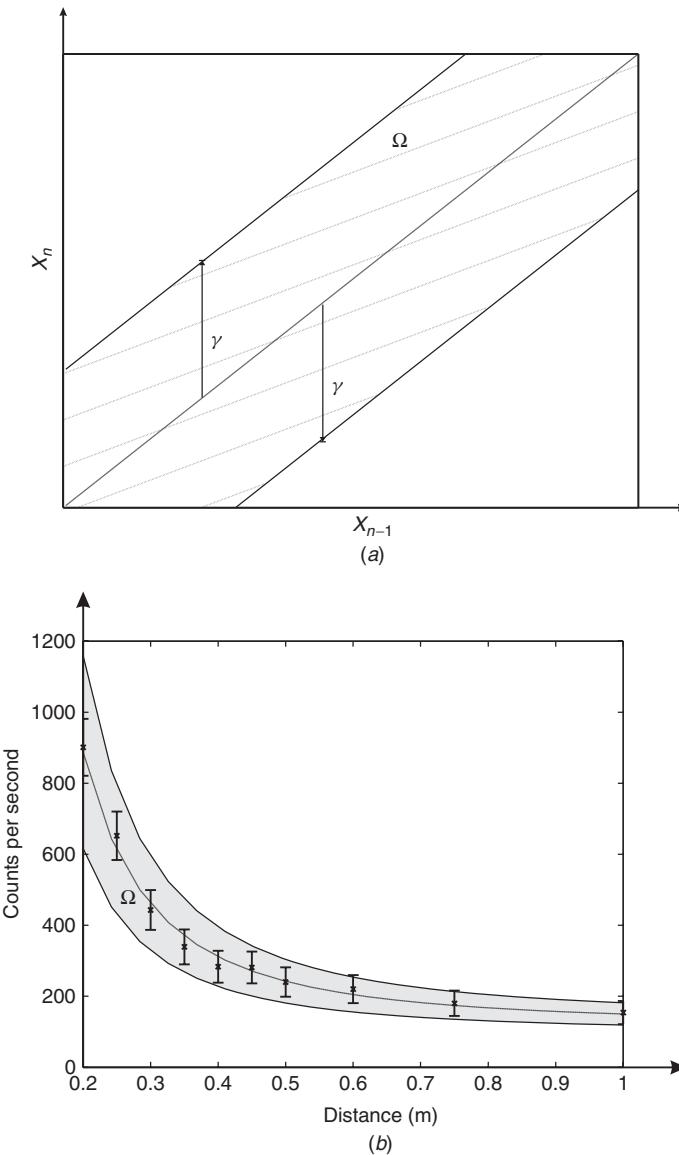
The first stage to coping with false measurement data is the classification of such data as suspicious. The diagnosis of measurement data should be based on a series of consistency checks [36, 37]. Consistency checking, in its simplest form, is merely a sanity check on the quality of the data. It is unreasonable to accept a temperature reading of 10,000 K. More generally, the idea behind consistency checking involves determining whether one or more measurements fall inside or outside of a region of validity. If it falls in the valid region, we classify that data as reliable and label the classification by the null hypothesis,  $H_0$ . On the other hand, unreliable data is classified as  $H_1$ .

In order to formally set up the notion of consistency checking, suppose that we have one or more measurements that have been gathered into a measurement vector  $\mathbf{X}$ . This vector resides in  $N$ -dimensional space, and we may define the consistency region  $\Omega$  in  $N$ -dimensional space by all  $\mathbf{X}$  that satisfy the consistency policy specified by the application. Many different strategies may be used to define the consistency region, and we depict the consistency region for two such strategies in Figure 28.3. We may employ outlier detection schemes to identify suspect data [38], or we may use a rule-based method to perform classification [36, 37, 39]. Finally, augmenting the classification is a measurement of the confidence of the classification. The quantification of the confidence level is done using the underlying data models provided in the classification policy. The trick behind defining a consistency check is choosing an appropriate measurement vector and consistency policy. We now discuss several consistency checking strategies.

**28.3.2.1 Physical Sanity Consistency Checks** Physical sanity consistency checks are the simplest of consistency checks. Here, the data vector is simply a single measurement  $\mathbf{X} = X$ . The consistency policy defines a range of physically reasonable values for  $X$ . For example, if we are using a temperature sensor, we might define the region  $\Omega$  by  $\Omega = [0, 1683]$  K to be the valid temperature region. In this case, a negative value or a value larger than the upper bound would indicate that something is amiss in a sensor reading. As a result, the measurement should be declared suspect and the filter enforcement phase should act accordingly. Here the confidence level produced can be conclusively set to 1.

**28.3.2.2 Temporal Consistency Checks** Temporal consistency checks involve gathering a sequence of measurements in time and employing past measurements to judge the quality of a current measurement. Here, we are sampling a single physical phenomenon over time to define a discrete-time series  $X_k$ . We collect the samples to form the measurement vector

$$\mathbf{X} = [X_n, X_{n-1}, \dots, X_{n-N+1}].$$



**Figure 28.3** Two examples of consistency checks and their consistency regions  $\Omega$ : (a) a simple temporal consistency check and (b) multimodal consistency check for monitoring radioactive material.

We may view the state space of  $\mathbf{X}$  by examining a lag plot of the data. The underlying physical properties of the process  $X_k$  allow us to define the consistency region  $\Omega$ . For example, in Figure 28.3a, we present a lag plot for  $N = 2$ , that is,  $X_n$  versus  $X_{n-1}$ , and define the consistency region  $\Omega$  by

$$\Omega = \{X_n : |X_n - X_{n-1}| < \gamma\}$$

for an appropriately chosen threshold  $\gamma$ . We declare that the data  $X_n$  violates temporal consistency if  $X_n \notin \Omega$ . This consistency check simply asks whether the discontinuity between the current measurement and the past measurement is too large to be physically possible. In this simple form of temporal consistency checking, the confidence level is dependent on how far  $X_n$  is beyond the threshold. More complex classes of consistency checks that lead to more general regions  $\Omega$  may be devised.

We emphasize that the use of a temporal consistency check strategy should be exercised with care. In many sensing applications, it is precisely measurement discontinuities that are of interest as they might signify a catastrophic event: A sudden shift in temperature might signify a spontaneous conflagration in a building, or a spurious spike in the acoustic monitoring of a building might indicate the presence of a trespasser. Further, temporal phenomena require an underlying sampling process that is designed to capture the salient features of the phenomena. The careful selection of an appropriate sample rate is necessary for the reliable operation of temporal consistency checking. In fact, natural and gradual transitions in the physical process can be viewed as sudden discontinuities if the sampling process is not sufficiently frequent.

**28.3.2.3 Multimodal Consistency Checks** Most current sensing paradigms focus on a single physical property at a time. This single measurement process may become a target of attack for an adversary. Many physical properties, however, exhibit correlation with other physical properties, which may be used to corroborate the values produced by a different type of measurement.

It is therefore natural to explore multimodal consistency checks. For example, suppose a device measures two different phenomena,  $X_1$  and  $X_2$ , that have a functional relationship  $\psi$ , which may be described via a physical law or learned empirically. Based on the observed value of  $X_1$ , we should expect that  $X_2$  falls within a certain confidence region. In particular, measurement  $X_1$  should have some confidence region governed by the measurement noise under ideal (nonattack) scenarios. Considering measurement errors for  $X_2$ , and using the function  $\psi$ , we may map the confidence region in  $X_1$  space to a corresponding confidence region in  $X_2$  space. Taken together, these confidence regions allow us to define a consistency region  $\Omega$ . If the measured value of  $X_2$  does not fall within this consistency region, then we declare that there is a multimodal inconsistency. Another example of a consistency check is provided in Figure 28.3, which depicts the expected relationship between counts monitored by a Geiger counter and distance of the Geiger counter sensor from the radioactive material. By calculating the distance between the material and the Geiger counter, and employing the calibration, one can define a consistency region in  $(1/r^2, C)$  space. Multimodality is important in this case as it captures the inherent relationship between two properties and, should the sensor be moved, a consistency check involving both properties immediately calibrates the data and lessens the likelihood of false alarm.

### 28.3.3 Measurement Enforcers

Once an inconsistency is detected, an appropriate action must be taken by the PoM monitor. Collectively, we refer to these actions as *filter enforcement* strategies. The choice of an appropriate enforcement strategy is specified by the sensing application and may vary for different nodes. The PoM monitor employs PoM enforcement policies to relate the data, their classifications, and their confidence levels to appropriate

enforcement filter operations. For example, translating the notion of access control matrices [35] to the PoM domain provides us with the notion of a PoM enforcement policy table, such as depicted in Table 28.1. Efficiently representing the PoM policy table is critical to the deployment of the PoM monitor on a resource-constrained platform like a sensor node.

**28.3.3.1 Measurement Purging, Marking, and Passthrough** The most natural filter response is to discard the measured data and not let suspicious data propagate further in the network. Data objects passed from the classification modules to the enforcement module are augmented by a classification field and a confidence field  $i(X)$ , describing the perceived confidence in classifying whether the data is consistent or not. Measurement purging drops all data whose classification is  $H_1$  and whose confidence  $i(X)$  is greater than a threshold. This enforcement policy should be used when the primary concern is energy consumption, and the network has sufficient redundancy in its deployment to be able to cope with missing data. The necessary redundancy for message purging introduces an interesting systems-level trade-off: Energy might be saved since poor-quality messages are purged, yet the extra nodes that are activated in order to provide redundancy means that more nodes are expending energy.

Another simple filter response is to mark and let data passthrough. It may be that nodes may not have enough information to reliably decide to act on the data, and hence it might be desirable to pass the data through the PoM monitor with the confidence assertions. In the hierarchical view of measurement assurance, such an operation will facilitate upstream filtering. In this case, the filter receives one or more data objects  $X_1, \dots, X_n$ , along with their classifications  $C_j$  and confidence levels  $i(X_j)$ . The filtered object is a vector of values

$$\mathbf{Z} = \langle [X_1, \dots, X_n], [C_1, \dots, C_n], [i(X_1), \dots, i(X_n)] \rangle.$$

**28.3.3.2 Measurement Cleansing** A more general sensor scenario involves data consumers desiring data objects that are compositions of the gathered measurements. Consider a sensor network in which the sensor application requires readings of a parameter vector  $\mathbf{Z}$ . To guarantee measurement assurance, the network administrator may have deployed sensors that measure a set of nonindependent samples  $X_j$ , which may be related to the desired parameter  $\mathbf{Z}$  through a functional or statistical relationship. When the readings are relatively clean, this relationship is enough for the sensor to determine  $\mathbf{Z}$ . In the presence of PoM attacks, however, this relationship may be exploited to provide data consumers significantly erroneous values for  $\mathbf{Z}$ . In this case, it is desirable for the PoM monitor to apply data cleansing before passing an estimate of  $\mathbf{Z}$ .

*Relationship-Based Cleansing* Suppose that the measured data vector  $\mathbf{X} = [X_1, \dots, X_N]$  has been sent to the enforcer, along with an indication that there is an inconsistency in the data. Rather than simply purge the entire measurement vector, we may try to identify which of the  $X_j$  is corrupted, correct it, and then use the cleaned vector to determine  $\mathbf{Z}$ . One approach for identifying which reading is corrupted is to take  $\mathbf{X}$  and remove a single  $X_j$  reading and employ an estimation technique, such as Wiener filtering or curve fitting, to estimate the removed measurement. Repeating this process for each  $X_j$ , we get a prediction error vector  $\mathbf{e}$  whose  $j$ th element is the error produced when estimating  $X_j$  using the other readings. If any error values are significantly large,

then we may conclude the corresponding  $X_j$  is corrupted. We replace that  $X_j$  with its estimate and then calculate  $\mathbf{Z}$  using the cleansed version of  $\mathbf{Z}$ .

It might seem natural to combine classification and cleansing into a single procedure. Although true from an algorithmic viewpoint, we must realize that for sensors it is desirable to reduce energy and computation wherever possible. The approach of first detecting an inconsistency in the classifier, and then letting the enforcer locate the inconsistency and correct it is desirable on resource-constrained architectures. Most of the time data objects will not be attacked, and it is wasteful to make calls to estimator functions when the data is clean. Instead, we use estimators only when an inconsistency is suspected.

**Robust Estimation** Measurement attacks might alter values enough to cause significant damage to actuators. For example, a decision based upon the average is easily thrown off by a single, corrupt reading. Consequently robust estimates of the values should be forwarded to the data consumer. The application of the median as a robust estimator for the mean is a simple example of robust statistical methods that have found application in data aggregation in sensor networks [13].

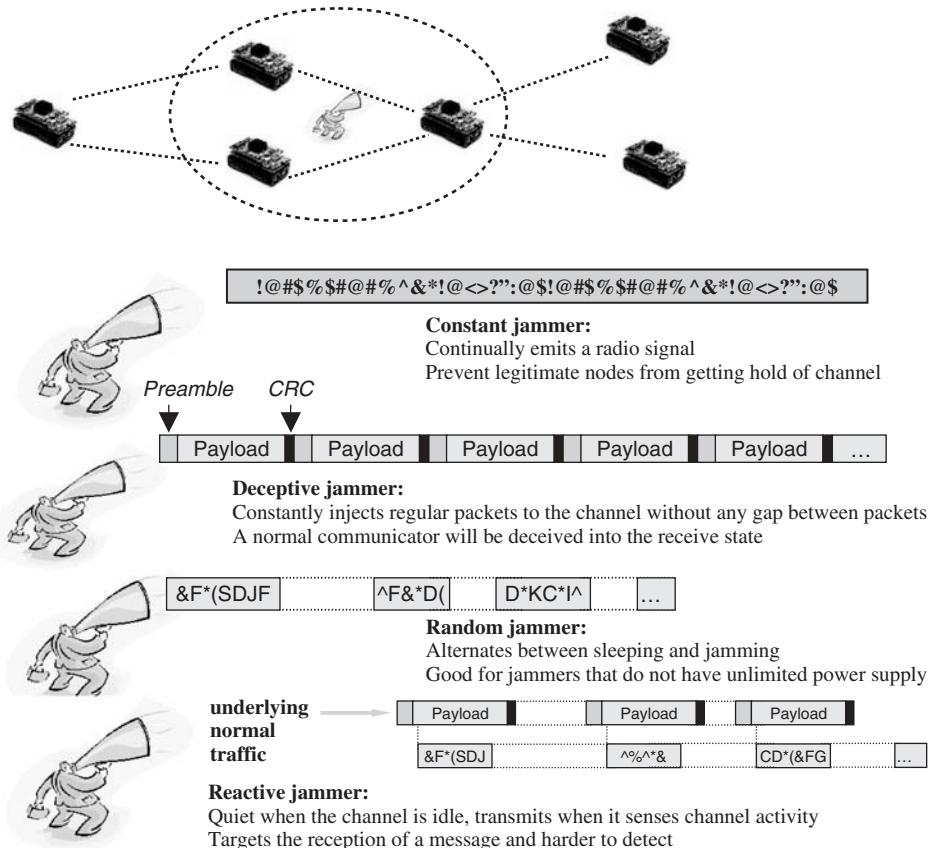
For the more general case where  $\mathbf{Z}$  has a functional or statistical relationship with  $\mathbf{X} = [X_1, \dots, X_N]$ , it is necessary to employ robust methods. *The challenge here is not merely devising robust estimation algorithms, as there are several well-known techniques in statistics that might be employed [40]. Rather, the challenge lies in carefully designing such algorithms to make them suitable for deployment in lightweight computing platforms.* For example, least median of squares (LMS) estimators and the random subset reweighting least-squares algorithms [40] can tolerate many measurement outliers but are too computationally demanding for sensor devices, and lightweight alternatives are needed.

In the end, the utility of a sensor system is closely tied to the ability of the sensors to provide accurate measurements to applications or individuals that make decisions. Although cryptographic methods can secure sensor communications against attacks during transmission, it is possible to subvert the measurement process. In order to cope with such threats, it is necessary to have accurate models of the underlying physical process so that appropriate data filtering techniques can be employed to effectively filter out false data readings. Information assurance in sensor systems will thus require a deep understanding of the actual sensed phenomena as well as traditional security mechanisms.

## 28.4 AVAILABILITY ATTACKS AGAINST THE WIRELESS LINK

There are many different attack strategies that an adversary can use to jam wireless communications [41–43], as depicted in Figure 28.4 While it is impractical to cover all the possible attack models that might exist, in this chapter, we review a wide range of jammers that have proven to be effective:

- **Constant Jammer** The constant jammer continually emits a radio signal, and it can be implemented either using a waveform generator that continuously sends a radio signal [44] or by using a normal wireless device that continuously sends out random bits to the channel without following any MAC layer etiquette [41].



**Figure 28.4** Jamming attacks target a sensor's ability to transmit or receive packets. Different jamming models accomplish the objective of blocking communications through different strategies.

Normally, the underlying MAC protocol allows legitimate nodes to send out packets only if the channel is idle. Thus, a constant jammer can effectively prevent legitimate traffic sources from getting hold of channel and sending packets.

- **Deceptive Jammer** Instead of sending out random bits, the deceptive jammer constantly injects regular packets to the channel without any gap between subsequent packet transmissions. As a result, a normal communicator will be deceived into believing there is a legitimate packet and will be duped to remain in the receive state. For example, in TinyOS, if a preamble is detected, a node remains in the receive mode, regardless of whether that node has a packet to send or not. Even if a node has packets to send, it cannot switch to the send state because a constant stream of incoming packets will be detected.
- **Random Jammer** Instead of continuously sending out a radio signal, a random jammer alternates between sleeping and jamming. Specifically, after jamming for a while, it turns off its radio, and enters a “sleeping” mode. It will resume jamming after sleeping for some time. During its jamming phase, it can either behave like a constant jammer or a deceptive jammer. This jammer model tries to take energy

conservation into consideration, which is especially important for those jammers that do not have unlimited power supply.

- *Reactive Jammer* The three models discussed above are active jammers in the sense that they try to block the channel irrespective of the traffic pattern on the channel. Active jammers are usually effective because they keep the channel busy all the time. As we shall see in the following section, these methods are relatively easy to detect. An alternative approach to jamming wireless communication is to employ a reactive strategy. The reactive jammer stays quiet when the channel is idle, but starts transmitting a radio signal as soon as it senses activity on the channel. One advantage of a reactive jammer is that it is harder to detect.

In [41], the above four jammer models were implemented using Berkeley Motes, which employed a ChipCon CC1000 RF transceiver and TinyOS as the operating system. The MAC layer was bypassed so that the jammer can blast on the channel irrespective of other activities that are taking place. The level of interference a jammer causes is governed by several factors, such as the distance between the jammer and a normal wireless node, the relative transmission power of the jammer and normal nodes, and the MAC protocol employed by normal nodes. The MAC protocol decides whether the channel is idle if the measured signal strength value is lower than a threshold. Many MAC protocols, such as the one in TinyOS release 1.1.1, use a fixed threshold value, while others, such as BMAC [45], adapt the threshold value based on the measured signal strength values, when the channel is idle. As a result, these two different categories of MAC protocols will decide the channel is jammed differently. We briefly summarize the results of the experiments, which involved three parties: *A*, *B*, and *X*, where *A* and *B* are normal wireless nodes with *A* being the sender, *B* the receiver, and *X* a jammer using one of the four models. More details of the experimental setup can be found in [41].

A jammer can interfere with normal communications between two legitimate communicators in two ways: preventing the sender from sending out packets or preventing the receiver from receiving packets. Hence, the resulting packet send ratio (PSR) and packet delivery ratio (PDR) measure the effectiveness of a jammer. The experiments showed that all four jammers are quite effective in interfering with normal communications. The constant jammer can successfully prevent a node from sending out packets, if that node employs a MAC protocol with a fixed threshold. Irrespective of the MAC protocol, the PDR, however, is very poor because the packets that manage to get sent out (such as the case where BMAC is employed) are corrupted anyway. The deceptive jammer, on the other hand, can completely block the send operations in any case because the sender will be forced to stay in receive mode all the time. The random jammer alternates between sleeping and jamming. While sleeping, the network operations will be normal; while jamming, it behaves just like a constant jammer or a deceptive jammer, depending on which mode it operates in. Finally, the reactive jammer does not affect the send ratio, but all the packets are corrupted, resulting in a zero PDR.

#### 28.4.1 Detecting Jamming Attacks in Sensor Networks

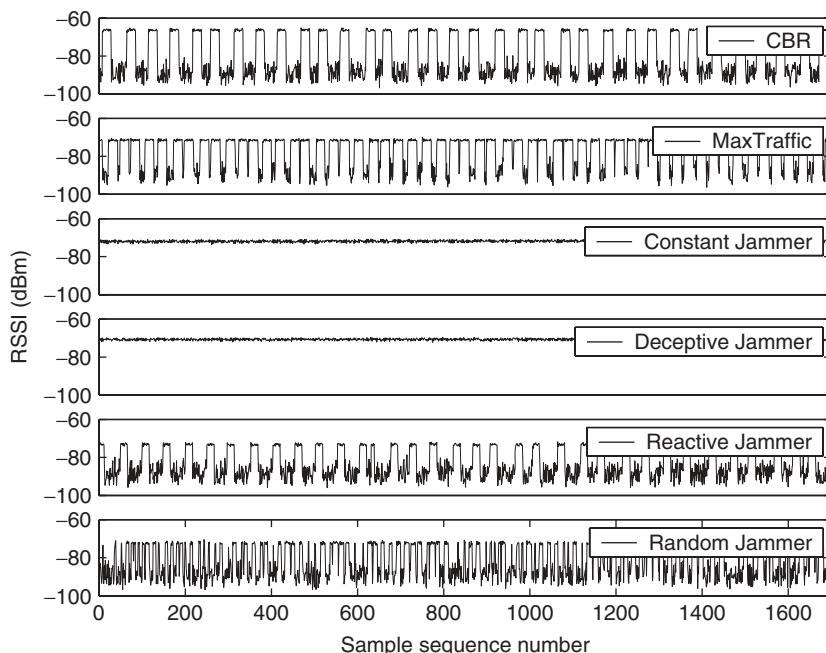
Detecting jamming attacks is important because it is the first step towards overcoming jamming. Detecting radio interference attacks is challenging as it involves discriminating between legitimate and adversarial causes of poor connectivity. In particular,

legitimate scenarios for poor-connectivity, such as congestion and device failures, may be difficult to differentiate from jamming.

There are several statistics that naturally lend themselves to detecting jamming, such as signal strength, carrier sensing time, and packet delivery ration. We will look at these different measurements and discuss how they are not effective in detecting a jamming attack. In order to repair the ability to detect a jamming attack, more sophisticated methods are needed, and we will discuss one possibility involving multimodal methods.

**28.4.1.1 Signal Strength** One natural measurement that can be employed to detect jamming is signal strength. The rationale behind using this measurement is that the signal strength distribution may be affected by the presence of a jammer. Two natural approaches to detecting jamming using signal strength involve comparing average signal magnitude versus a threshold calculated from the ambient noise levels and classifying the *shape* of a window of signal samples.

In order to illustrate the effect that a jammer would have on the received signal strength, we present results of several experiments conducted with the MICA2 Mote platform in Figure 28.5. These experiments are described in more detail in [41]. In the first two experiments, we have two Motes, a sender A and a receiver B, which are 30 inches apart from each other. The top two plots correspond to normal, or benign, traffic scenarios where the source A transmits at a constant traffic rate (CBR) of 5.28 kbps, while the second plot corresponds to A transmitting at its maximum send rate, corresponding to a raw traffic rate of 6.46 kbps. The bottom four plots correspond to four different jamming scenarios in which we introduced a jammer. Throughout



**Figure 28.5** RSSI readings as a function of time in different scenarios. RSSI values were sampled every 1 ms.

these four jamming scenarios, A is a CBR source. Looking at raw time series data, it is clear that any statistic based solely upon a moving average of the RSSI values would be hard pressed to discriminate between a normal traffic scenario and a reactive jammer scenario. Further, the shape of the RSSI time series for normal traffic scenarios and the reactive jammer are too similar to rely on spectral discrimination techniques for discrimination. Further analysis of these methods, and the difficulties associated with using signal strength readings, may be found in [41]. Overall, these results suggest the following important observation: We may not be able to use simple statistics, such as average signal strength or energy, to discriminate jamming scenarios from normal traffic scenarios because it is not straightforward to devise a threshold that can separate these two scenarios.

**28.4.1.2 Carrier Sensing Time** A jammer can prevent a legitimate source from sending out packets because the channel might appear constantly busy to the source, and hence it might seem possible to use carrier sensing time as a means to determine whether a device is jammed. In [41], the authors explored this possibility. We observed that using carrier sensing time is suitable when the following two conditions are true: the jammer is nonreactive or nonrandom, and the underlying MAC protocol determines whether a channel is idle by comparing the noise level with a fixed threshold. If these two conditions are true, then carrier sensing time is an efficient way of discriminating a jammed scenario from a normal ill-functioned scenario, such as congestion, because the sensing time will be bounded, though large, in a congested situation, but is unbounded in a jammed situation. Overall, the carrier sensing time alone cannot be used to detect all the jamming scenarios.

**28.4.1.3 Packet Delivery Ratio** Similarly, the PDR may be used to detect the presence of jamming, as the jammer can effectively corrupt transmissions, leading to a much lower PDR. Since a jamming attack will degrade the channel quality surrounding a node, the detection of a radio interference attack essentially boils down to determining whether the communication node can send or receive packets in the way it should have had the jammer not been present. More formally, let us consider the PDR between a sender and a receiver, who are within radio range of each other, assuming that the network only contains these two nodes and that they are static. As noted earlier, an effective jammer results in a very poor PDR, close to 0, which indicates that PDR may be a good candidate in detecting jamming attacks. We would like to point out that a nonaggressive jammer, which only marginally affects the PDR, does not cause noticeable damage to the network quality and does not need to be detected or defended against.

Next, we need to investigate how much PDR degradation can be caused by nonjamming, normal network dynamics, such as congestion, failures at the sender side, and the like. The studies in [41] showed that, even in a highly congested situation where a raw traffic rate of 19.38 kbps is offered to MICA2 radio whose maximum bandwidth capacity is 12.364 kbps at a 100% duty cycle, the PDR measured by the receiver is still around 78%. As a result, a simple thresholding mechanism based on the PDR value can be used to differentiate a jamming attack, regardless of the jamming model, from a congested network condition. Though PDR is quite effective in discriminating jamming from congestion, it is not as effective for other network dynamics, such as a sender battery failure, or the sender moving out of the receiver's communication range,

because these dynamics can result in a sudden PDR drop in much the same way as a jammer does. Specifically, if the sender's battery drains out, it stops sending packets, and the corresponding PDR is 0%.

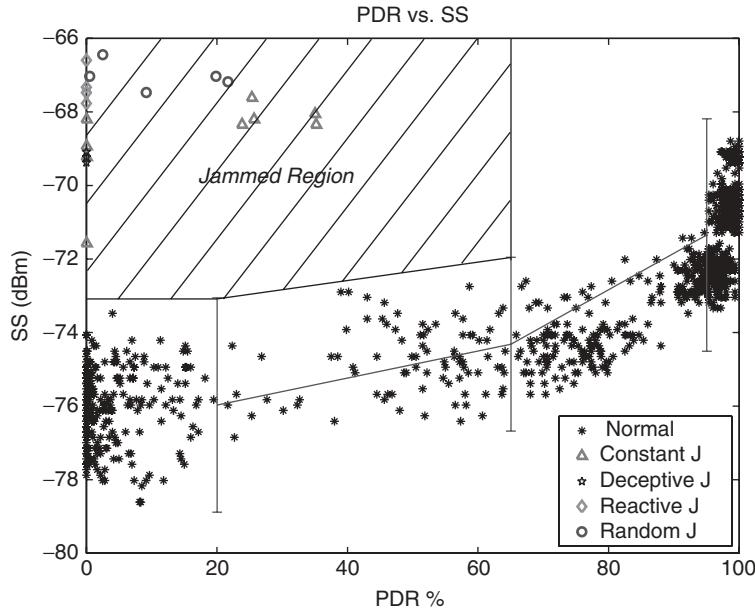
Consequently, compared to signal strength and carrier sensing time, PDR is a more powerful statistic in that it can be used to differentiate a jamming attack from a congested network scenario, for different jammer models. However, it still cannot differentiate the jamming attack from other network dynamics that can disrupt the communication between the sender and the receiver.

**28.4.1.4 Advanced Detection Strategies** Rather than use such basic statistical methods, multimodal strategies, such as combining PDR with signal strength readings, appear to be promising. In a normal scenario with no interference, a high signal strength corresponds to a high PDR. However, if the signal strength is low, that is, the strength of the signal is comparable to noise levels, the PDR will be also low. On the other hand, a low PDR does not necessarily imply a low signal strength: It may be that all of a node's neighbors have died (perhaps from consuming battery resources or device faults), or it may be that the node is jammed. The key observation here is that, in the first case, the signal strength is low, which is consistent with a low PDR measurement, while in the jammed case, the signal strength should be high, which contradicts the fact that the PDR is low.

Using these observations, a multimodal consistency check may be defined [41]. During a guaranteed time of noninterfered network operation, a table (PDR, SS) of typical packet delivery ratios and signal strength (SS) values are measured. From this data, one can calculate an upper bound for the maximum SS that would have produced a particular PDR value in a nonjammed scenario. Using this bound, the (PDR, SS) plane is partitioned into a benign region and a jammed region. To illustrate how such a detection scheme might operate, we present the results of our investigation, which was conducted using MICA2 Motes, in Figure 28.6. We varied source–receiver separation for four different jammers. The PDR and SS readings were averaged over a sufficient time window to remove anomalous fluctuations (e.g., hardware-related or fading-related variations). We found the 99% SS confidence levels for different regions and defined the jammed region to be the region of (PDR, SS) that is above the 99% signal strength confidence intervals and whose PDR values are less than 65%. As can be seen in Figure 28.6, the (PDR, SS) values for all jammers distinctly fall within the jammed region, suggesting that classification is feasible.

## 28.4.2 Mapping Jammed Areas

Following the detection of whether a node is jammed, it is desirable for the network to map out regions of the sensor network that are jammed. By having a map of jammed areas, network services can use this knowledge to influence routing, power management, and higher layer planning. A protocol for mapping out the jammed regions of a sensor network was presented in [46]. In this section, jamming detection is performed by monitoring channel utilization. Once the sensors observe that their channel utility is below a preset threshold, they conclude that they are jammed. Following detection, the jammed nodes bypass their MAC layer temporarily and broadcast JAMMED messages, announcing the fact that they are jammed. These JAMMED messages will not be able to be received by other jammed neighbors. However, those neighbors on the



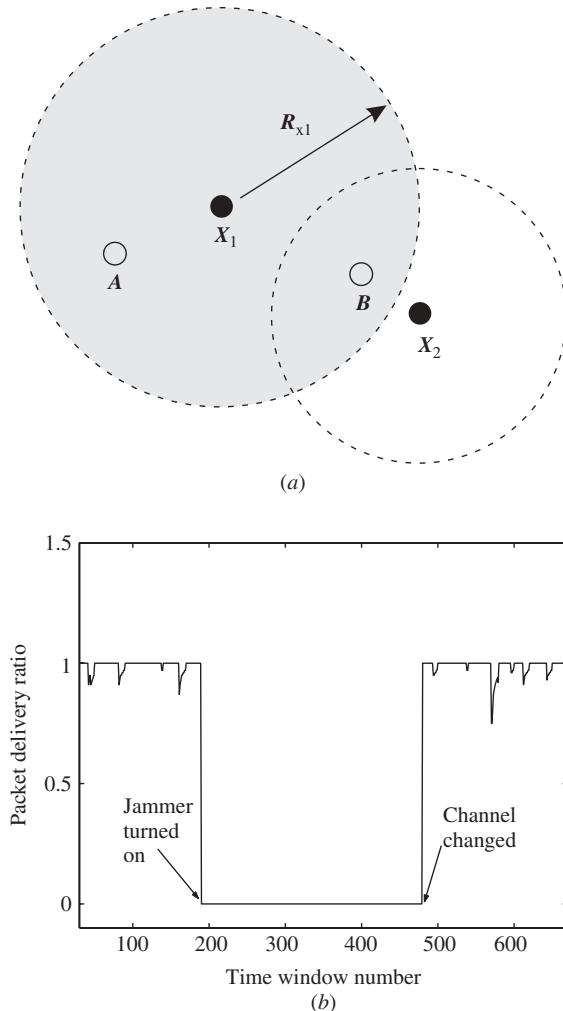
**Figure 28.6** (PDR, SS) measurements indicating the relationship between PDR and signal strength, and the (PDR, SS) values for different jammers. The shaded region is the jammed region.

boundary of the jammed region, but are not themselves jammed, will be able to hear the JAMMED messages, though potentially at a higher error rate. Once nonjammed sensors receive JAMMED messages, they initiate the mapping procedure. These non-jammed nodes exchange and merge information describing which nodes they have witnessed as jammed, where those jammed sensors are located, along with neighbor information. By continuing the exchange of information regarding witnessed jammed nodes, the network will eventually be able to map out the boundary of a jammed area.

#### 28.4.3 Recovering the Network: Channel Surfing

There are a variety of different strategies that may be used to overcome radio-level interference. At one level, one could attempt to adjust the transmission power in order to out-compete the jammer [47], or employ stronger error correcting codes in order to have resilience to decoding errors introduced by the jammer [48], or even move sensor nodes away from the interference source and reestablish connectivity at a safe distance [49]. For the remainder of this section, we shall focus on a different defense strategy, known as channel surfing, where the network seeks to reestablish connectivity by adjusting its channel allocation.

Channel surfing is motivated by frequency hopping modulation. Unlike frequency hopping, which is a PHY layer modulation method involving continual changing of the carrier frequency, the changing of frequencies in channel surfing is on demand and operates at the link layer. Let us examine a simple two-party communication scenario, as depicted in Figure 28.7a, where adversary  $X_1$  or  $X_2$  has disrupted communication between  $A$  and  $B$ . In channel surfing, both  $A$  and  $B$  change their channel assignment to



**Figure 28.7** (a) Jammed two-party radio communication and (b) PDR measurements from channel surfing prototype.

a new channel in order to avoid  $X$ 's interference. Changing channels in the two-party scenario is fairly straightforward. A prototype was built using the Berkeley MICA2 platform, in which  $A$  and  $B$  detect whether they are jammed, switch to a new channel (channel assignment is done using a pseudorandom generator), and reestablish connectivity when they detect each other's presence on the new channel [44]. Example results depicting the PDR for the two-party channel surfing prototype are presented in Figure 28.7b.

Extending the notion of channel surfing to more general network scenarios, such as wireless local area networks (LANs) or ad hoc networks, is significantly more challenging. An initial outline of such channel surfing strategies was presented in [44] and more detailed implementation studies were presented in [50]. Implementing these basic strategies is a difficult task as reliably coordinating multiple devices switching to

new channels faces the usual challenges of distributed computing: asynchrony, latency, and scalability. Two variations of channel surfing have been proposed: *coordinated channel switching* and *spectral multiplexing*.

In a coordinated channel switch, the entire network changes its channel to a new channel. In such a scheme, when a node detects that it is jammed, it switches channels and sends beacons to announce its presence on the new channel. Boundary nodes, which are not jammed but are neighbors of jammed nodes, will detect the absence of their neighbors on the original channel and probe the next channel to see if their neighbors are still nearby. If a node detects beacons on the new channel, it will switch back to the original channel and transmit a broadcast message informing the entire network to switch to the new channel.

Performing a coordinated channel switch across an entire network incurs significant latency as the scale of the network increases and, as a result, the network may be in an unstable phase where some devices are on an old channel while others are waiting on the new channel. To alleviate the latency problem, we can have only jammed regions switch channels and have nodes on the boundary of a jammed region serve as *relay* nodes between different spectral zones.

Many commercial wireless sensor networks are susceptible to both intentional and nonintentional radio interference. Such interference is likely to become more prevalent as sensor networks become increasingly pervasive. It is therefore critical to develop methods that can make sensor networks coexist with each other and even survive external interference. These defense mechanisms, however, must be distributed, easy to scale, and have low false positives. The methods described in the section serve as an overview to several potential directions for coping with interference. There are many other methods that can be employed to cope with interference, ranging from physically moving sensor nodes away from the interference source, to adapting transmission power levels or physical layer modulation techniques.

## 28.5 ENSURING PRIVACY OF ROUTING CONTEXTS

Privacy may be defined as the guarantee that information, in its general sense, is observable or decipherable by only those who are intentionally meant to observe or decipher it. The phrase “in its general sense” is meant to imply that there may be types of information besides the message content that are associated with a message transmission. Consequently, the privacy threats that exist for sensor networks may be categorized into two broad classes: content-oriented security/privacy threats and contextual privacy threats. Content-oriented security and privacy threats are issues that arise due to the ability of the adversary to observe and manipulate the exact content of packets being sent over the sensor network, whether these packets correspond to actual sensed data or sensitive lower layer control information. In contrast to content-oriented security, the issue of contextual privacy is concerned with protecting the *context* associated with the measurement and transmission of sensed data. For many scenarios, general contextual information surrounding the sensor application, especially the location of the message originator, are sensitive and must be protected. This is particularly true when the sensor network monitors valuable assets since protecting the asset’s location becomes critical.

In order to facilitate the discussion and analysis of source location privacy in sensor networks, a generic asset monitoring application known as the *Panda-Hunter Game*

was introduced in [11]. In the Panda-Hunter Game, a large array of panda detection sensor nodes have been deployed by the Save-The-Panda Organization to monitor a vast habitat for pandas [51]. As soon as a panda is observed, the corresponding *source* node will make observations and report data periodically to the *sink* via multihop routing techniques. The adversary is a hunter who tries to capture the panda by back-tracing the routing path until it reaches the source. During the lifetime of the network, the sensor nodes continually send data, and the hunter may use this to his advantage to track and hunt the panda. In the formulation of the Panda-Hunter Game, it is assumed that the source includes its ID in the encrypted messages, and that only the sink can tell a node's location from its ID. The objective of routing protocols that enhance source location privacy is to extend the amount of time needed for an adversary to trace back (in a hop-by-hop manner) to the source. At the same time, the sensor network should maintain desirable levels of message latency while minimizing the total number of messages transmitted.

A carefully designed routing technique can be effective in protecting the source's location. In this section, we classify the existing routing protocols into two broad classes: flooding and single-path routing. In each class, we examine the privacy protection capabilities of many routing protocols and examine several enhancements to boost privacy. In order to stress the effectiveness of various routing protocols, in this section, we base our discussion on a simpler scenario where the panda pops up at a random location and then stays there until captured. We further assume that the hunter always starts monitoring the network at the sink.

We examine the privacy preserving capabilities of several popular routing protocols, namely baseline flooding, probabilistic flooding, and baseline single-path routing.

### 28.5.1 Flooding

Many sensor networks employ flooding to disseminate data and control messages [52–55]. In flooding, a message originator transmits its message to each of its neighbors, who in turn retransmit the message to each of their neighbors. In order to control the energy consumption, in practice, every node in the network only forwards the same message at most once. When a message reaches an intermediate node, the node first checks whether it has received and forwarded that message before.

**28.5.1.1 Baseline Flooding** Flooding is known to have a high energy consumption. At first glance, one may think that flooding can provide strong privacy protection since almost every node in the network will participate in data forwarding, and that the hunter may be led to the wrong source. However, it should be emphasized that flooding provides the least possible privacy protection since it allows the hunter to track and reach the panda after a minimum number of messages.

The poor privacy performance of flooding is due to the fact that the set of all paths produced by the flooding of a single message contains the shortest path, and hence the first message that the hunter receives while waiting around the sink will correspond to a message that follows the shortest path. As a result the hunter will be able to jump to the forwarding node on the last hop in the shortest path. Now, while the hunter is sitting at this new position, the source produces the next message. Due to the fact that the hunter is on the shortest path, the hunter will subsequently receive the next message via the subpath of the source–sink shortest path. Thus, the hunter can jump to

the previous forwarding node on the source–sink shortest path. Ultimately, the hunter will capture every message on the shortest path, and reach the source via the shortest path.

**28.5.1.2 Probabilistic Flooding** Probabilistic flooding [56–58] was first proposed as an optimization of flooding to reduce energy consumption. In probabilistic flooding, only a subset of nodes within the entire network will participate in data forwarding, while the others simply discard the messages they receive. The ratio of the nodes that participate in data forwarding is referred to as the *forwarding probability* ( $P_{\text{forward}}$ ).

In addition to its energy efficiency, probabilistic flooding can improve the privacy. Imagine there exists a path {1, 2, 3, 4, sink}, and the hunter is waiting for a new message at node 4. In flooding, the subsequent message will certainly arrive at node 4, though after some delay. However, in probabilistic flooding, the subsequent message may not arrive at node 4 because the nodes before it may decide not to forward. As a result, the source will likely have to transmit more messages in order for the hunter to work his way back to the source.

In probabilistic flooding, the hunter will often not stay on the shortest path between the source and sink since there is a positive probability that a message will not be delivered on the shortest path. Let us pick a random node within the network and a random path connecting that node and the source. If this path has length  $l$ , and we use  $P_{\text{path}}$  to represent the probability of the node getting a message from this particular path, then

$$P_{\text{path}} = P_{\text{forward}}^l. \quad (28.1)$$

If the shortest path fails, there is a high likelihood that *at least* one longer path will succeed, drawing the hunter away from the shortest path, putting the hunter on a less efficient path. It should be noted, however, that the improvement that probabilistic flooding provides is not unrestricted. There is a natural probabilistic pull that draws the hunter back toward shorter paths. To see this, suppose the hunter migrated and followed a longer path of length  $l_2$ . Further, if there is a shorter path with length  $l_1$  that passes through the node that he is now at, then applying (28.1), we have  $P_{\text{path}_2} = P_{\text{forward}}^{l_2}$  and  $P_{\text{path}_1} = P_{\text{forward}}^{l_1}$ , and hence  $P_{\text{path}_2} < P_{\text{path}_1}$ . Thus there is a higher likelihood that the hunter will drift back toward a shorter path, and therefore the hunter will ultimately receive the majority of his new messages from a set of reasonably short paths.

**28.5.1.3 Flooding with Fake Messages** Flooding cannot provide privacy protection because the hunter can easily identify the shortest path between the source and the sink, allowing him to back trace to the source location. One of the reasons this happens is due to the fact that we only have one source in the network. This observation suggests that one promising approach we can take to alleviate the risk of a source location privacy breach is to introduce more sources that inject fake messages into the network.

In order to discuss the possible gain of fake messaging, we assume that these messages are of the same length as the real messages, and that they are encrypted as well. Therefore, the hunter cannot tell the difference between a fake message and a real one. One challenge with implementing fake messaging is how to inject fake messages.

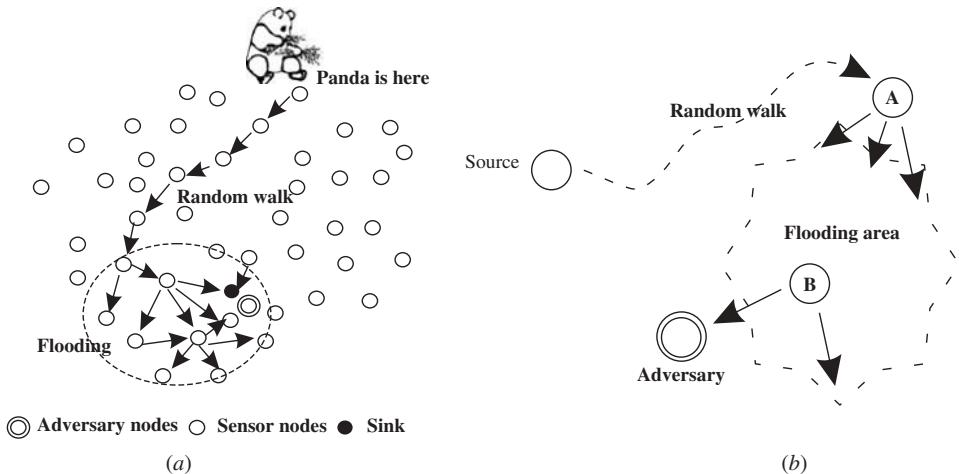
The use of fake sources to obfuscate the location of the original source was examined in [11], and two different injection strategies were proposed:

- *Short-Lived Fake Source* This method does not require any additional overhead. We propose to use the constant  $P_{fake}$  to govern the fake source percentage. For any node within the network, after it receives a real message, it generates a random number  $q$  that is uniformly distributed between 0 and 1. If  $q < P_{fake}$ , then this node will produce a fake packet and flood it to the network. We can tune the value of  $P_{fake}$  to balance the trade-off between energy consumption and privacy protection. In this strategy, a node may be a fake source for a short period of time, as governed by  $P_{fake}$ .
- *Persistent Fake Source* The basic idea of this method is that once a node decides to become a fake source, it will keep generating fake messages regularly so that the hunter can be further misled. A node decides whether or not to become a fake source based upon the value  $P_{fake}$  as described above. The location of the fake source(s) can have a bearing on the performance. For example, a fake source that is on the opposite side of the sink may work better than a fake source that is along the path from the sink to the source.

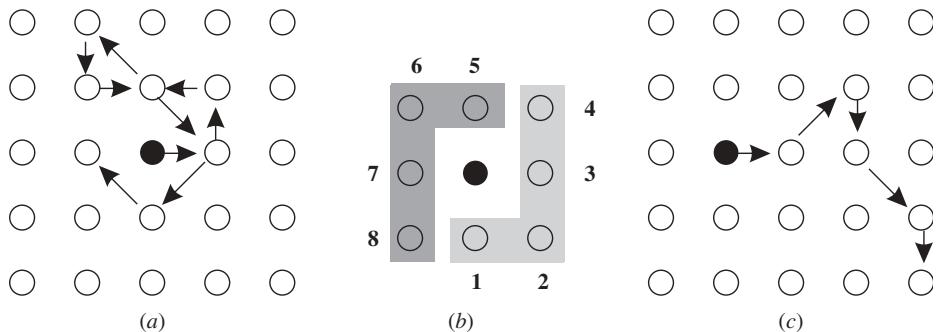
It was observed that if the fake messages are injected into the network at the same rate as the real messages, then the adversary will likely oscillate between the real source and the fake source, and cannot make progress toward either of them. If the fake messages are injected at a slower rate, then the hunter will be drawn toward the real source. On the other hand, if the fake messaging rate is higher than the real messaging rate, then the hunter will be kept at the fake source, thereby protecting the true source's location.

**28.5.1.4 Phantom Flooding** Probabilistic flooding is not very effective in privacy protection because shorter paths are more likely to deliver more messages. Therefore, we would like to entice the hunter toward a fake source, called the phantom source. In phantom flooding, every message experiences two phases: (1) a walking phase during which the message is unicasted a distance of  $h_{walk}$  hops, and (2) a subsequent flooding phase meant to deliver the message to the sink. The algorithm is illustrated in Figure 28.8a.

During the walking phase, the idea of “random” walk is used to determine where to walk to next, in which a message walks to a random neighbor of the current node. However, if the network is more or less uniformly deployed, and we let those nodes randomly choose one of their neighbors with equal probability, then the resulting random walk has a large chance that the path will loop around the source's spot and end at a random location not far from the source (illustrated in Fig. 28.9a). In order to avoid this phenomenon, we need to introduce bias into the walking process, and thus a *directed walk* was proposed in [11]. In the directed walk approach, we separate the neighbors into two groups so that those nodes whose directions are opposite to each other do not belong to the same group, as illustrated in Figure 28.9b. During the first step of the directed walk, the node randomly picks one group, and later steps will only choose neighbor nodes from that specific group. This method can remove the paths that loop back upon themselves in the random walk. As a result, the routing can leave



**Figure 28.8** Illustration of phantom flooding: (a) phantom flooding protocol and (b) example scenario.



**Figure 28.9** From random walk to directed walk: (a) random walk, (b) neighbor grouping method, and (c) directed walk.

the source area and reach a random location (illustrated in Fig. 28.9c). Directed walk requires a node knows the relative position of its neighbors. Such knowledge can be obtained by using ranging [59–61] and angle of arrival (AoA) [62] measurements.

Phantom flooding can significantly improve source location privacy because messages will probably take different, divergent paths during the walking phase. As a result, after the hunter hears a message, it may take a long time before he receives the next message. Further, by having the messages originate from a location far from the source, the hunter will be led away from the true source. In the example shown in Figure 28.8b, the hunter is already pretty close to the source before he receives the next message. This new message goes through the random-walk phase and reaches node A, and then goes through the flooding phase. The hunter receives this message from node B, and it will be duped to move to node B, which is actually farther away from the source than its current location.

## 28.5.2 Single-Path Routing

Flooding is the simplest data-forwarding protocol in sensor networks, but it usually consumes large amounts of energy. In order to extend the network lifetime, a considerable amount of research effort has been devoted to developing energy-efficient routing techniques that allow a node to forward packets to only one of its neighbors. This family of routing techniques is referred to as *single-path routing*. Single-path routing techniques usually require either extra hardware support or a preconfiguration phase. For example, in [63], Karp and Kung propose to use the location information of a node, its neighbors, and the destination to calculate a greedy single routing path. In [62], Niculescu and Nath propose trajectory-based routing, which uses the location information associated with a node and its neighbors to create a routing path along a specified trajectory. Location information can be obtained by either using Global Positioning System (GPS) or other means. In other schemes, such as [55, 64], initialization and reconfiguration phases are required to set up and maintain the routing path from the source to the sink.

In general, the amount of transmissions required for single-path routing is  $O(h)$ , where  $h$  is the amount of nodes along the routing path from the source to the sink. This is a dramatic improvement compared to the flooding protocol. However, although single-path protocols conserve energy, they are rather poor at protecting the source's location. Since only the nodes that are on the routing path forward messages, the hunter can track the path easily and can locate the panda within  $h$  moves. Therefore, based on these observations, it is necessary to incorporate modifications into baseline single-routing in order to achieve improved privacy performance.

**28.5.2.1 Alternating Single-Path Routing** Single-path routing only uses one path to forward data, and the hunter can thus easily back trace the path to the source. Inspired by this observation, one may try to alternate between multiple routing paths [65] to improve the privacy. Suppose we pick a fixed set of routing paths, and alternate between them. A hunter might start out on a path, and eventually have a period where he does not see any messages since the source has switched to an alternate path. Rather than return to the sink and try a new trace back from the sink, a good strategy for the hunter to employ is to wait on the path that he has followed until the source randomly resumes the path that he has followed. It is clear that, on average, the privacy level should be increased by a factor equal to the number of alternating paths from which the source can choose. For example, if the source has two alternating paths, the average number of messages that will be transmitted by the source before the hunter can trace his way back to the panda is twice that of baseline single-path routing.

**28.5.2.2 Phantom Single-Path Routing** Let us look at the behavior of alternating single-path routing again. It starts with the hunter following one path out of a set of alternate routing paths. Then, at a random time the source switches to a different path. If the source never returns to that path, and none of the alternate paths intersect the original path, then the hunter becomes stranded. With this observation in mind, phantom single-path routing was proposed in [11].

Similar to phantom flooding, phantom single-path routing seeks to pull the hunter away from the source and consists of directed walk in a random direction, followed

by shortest path routing. The parameter  $h_{\text{walk}}$ , describing the number of hops in the walking phase, has a bearing on the privacy level of this approach. A larger value of  $h_{\text{walk}}$  can potentially create a larger and more divergent family of source–sink paths. As a result, in realistic scenarios with thousands of sensors, the family of divergent paths can be substantially large, and the probability of sending messages over precisely the same path will decrease dramatically. Even though the hunter knows the routing technique, it is difficult for him to track the source location, regardless of whether he waits at a location until the arrival of the next new message or returns to his previous position after a certain time period.

It was observed in [11] that an adversary must witness more messages for phantom shortest path than for phantom flooding in order to track down the location of the source. This behavior is due to the fact that, when we perform routing after the random walk, there is a high likelihood that the resulting single paths from subsequent phantom sources will not significantly intersect and hence the hunter may miss messages. On the other hand, the resulting floods from subsequent phantom sources will still result in packets arriving at the hunter, allowing him to make progress. Overall, phantom routing techniques have been shown to be desirable since they only marginally increase communication overhead, while achieving significant protection in the source’s location privacy.

Sensor networks will be deployed to monitor valuable assets. In many scenarios, an adversary may be able to back trace message routing paths to the event source, which can be a serious privacy breach for many monitoring and remote sensing application scenarios. By appropriately modifying the underlying routing protocols that are employed, it is possible to obfuscate the location of a source sensor. Currently, phantom routing techniques, such as outlined in this section, have been shown to be promising for enhancing source location privacy, and many modifications to the basic phantom routing strategy have been proposed. Other privacy aspects related to the operation of a sensor network, such as protecting the context surrounding the time at which a sensed event occurred, can also be addressed at the routing layer. Recent work in [12] has examined the use of internal network buffers to randomly delay packets before forwarding so as to obfuscate temporal information related to sensor events from an eavesdropping adversary. The many contexts surrounding the deployment of sensor networks, plus an increasing societal awareness of privacy issues, suggests that developing customized privacy enhancement solutions to sensor systems will become increasingly important.

## 28.6 CONCLUSION

Ensuring the security of sensor networks is a challenge that is at least as vast and complicated as the challenge facing traditional communication networks. The resource limitations and operational requirements facing the system designer have required that a new collection of tools be developed over the past decade in order to face the foreseeable threats that adversaries may launch against a sensor network or its monitoring applications. In this chapter we have examined a variety of different confidentiality, integrity, and availability threats that may be faced by sensor networks as they are deployed in unattended scenarios. We briefly surveyed different strategies that have been proposed for several of these threats.

We then focused on three specific sensor security topics. First, we explored an architecture that can assist in ensuring the integrity of the measurement process. Since sensor nodes monitor physical phenomena, there are a variety of noncryptographic threats that may be employed to subvert the process of measuring and delivering measurements to a monitoring application. The framework involves a combination of secure aggregation techniques with message marking as well as defining appropriate policies specifying how and when data should be filtered.

Next, we examined a class of availability attacks that involve jamming the communication between sensor nodes. We summarized a collection of different jammer models that have been proposed recently. In order to cope with such a threat, it is first necessary to detect the presence of jamming. Ultimately, jamming detection is best accomplished by using a combination of statistics, including packet delivery rate and received signal strength. Following the detection of jamming, it is often desirable to map out the jammed area, and then reconstruct network connectivity. One method for recovering network communication functionality involves adjusting the channel allocation across the sensor network.

Lastly, we noted that the routing of sensor messages may lead to a variety of contextual privacy breaches. In scenarios where an adversary monitors sensor network messages in hopes of tracking a valuable asset that the network is monitoring, conventional routing techniques, such as single-path routing and flooding, can lead to an adversary employing a hop-by-hop trace-back strategy to rapidly identify the location of the asset being monitored. We describe a variety of routing obfuscation protocols that have been proposed to cope with such threats.

Looking forward, these unique threats facing wireless sensor networks will require a continual evolution of defense strategies. Although the computing resources available to sensor networks are increasing as technology matures, adversaries are likely to have more resources at their disposal than individual sensor nodes, and thus it will be important to develop security solutions that provide the strongest degree of protection possible while minimizing the total communication and computational resources expended.

## REFERENCES

1. M. Bishop, *Computer Security: Art and Practice*, Reading, MA: Addison Wesley, 2003.
2. A. Perrig, R. Szewczyk, D. Tygar, V. Wen, and D. Culler, “SPINS: Security protocols for sensor networks,” *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.
3. C. Karlof, N. Sastry, and D. Wagner, “Tinysec: A link layer security architecture for wireless sensor networks,” in *Sensys ’04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 162–175.
4. L. Eschenauer and V. Gligor, “A key-management scheme for distributed sensor networks,” in *CCS ’02: Proceedings of the 9th ACM Conference on Computer and Communications Security*, 2002, pp. 41–47.
5. H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks,” in *SP ’03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*, 2003, pp. 197–213.
6. D. Liu, P. Ning, and R. Li, “Establishing pairwise keys in distributed sensor networks,” *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 1, pp. 41–77, 2005.

7. D. Liu and P. Ning, "Improving key predistribution with deployment knowledge in static sensor networks," *ACM Trans. Sen. Netw.*, vol. 1, no. 2, pp. 204–239, 2005.
8. A. Liu and P. Ning, "Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008)*, 2008.
9. A. Liu and P. Ning, "Tinyecc library," available: <http://discovery.csc.ncsu.edu/software/TinyECC/>.
10. I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*, Cambridge University Press, 1999.
11. P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Enhancing source-location privacy in sensor network routing," in *ICDCS'05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, 2005, pp. 599–608.
12. P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, "Temporal privacy in wireless sensor networks," in *ICDCS'07: Proceedings of the 27th IEEE International Conference on Distributed Computing Systems*, 2007, pp. 23–30.
13. B. Przydatek, D. Song, and A. Perrig, "SIA: Secure information aggregation in sensor networks," in *SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, 2003, pp. 255–265.
14. D. Wagner, "Resilient aggregation in sensor networks," in *SASN '04: Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, 2004, pp. 78–87.
15. K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: A quantitative comparison," *Comput. Networks*, vol. 43, no. 4, pp. 499–518, 2003.
16. Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, 2005.
17. D. Liu, P. Ning, and W. Du, "Attack-resistant location estimation in sensor networks," in *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, 2005.
18. S. Capkun and J. P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, 2005.
19. A. Seshadri, A. Perrig, L. vanDoorn, and P. Khosla, "Swatt: Software-based attestation for embedded devices," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2004.
20. A. Seshadri, M. Luk, E. Shi, A. Perrig, L. Doorn, and P. Khosla, "Pioneer: Verifying integrity and guaranteeing execution of code on legacy platforms," in *Proceedings of ACM Symposium on Operating Systems Principles (SOSP)*, 2005.
21. A. Seshadri, M. Luk, A. Perrig, L. van Doorn, and P. Khosla, "Scuba: Secure code update by attestation in sensor networks," in *ACM Workshop on Wireless Security (WiSe 2006)*, 2006.
22. S. Zhu, S. Xu, S. Setia, and S. Jajodia, "LHAP: A lightweigth hop-by-hop authentication protocol for ad-hoc networks," in *International Workshop on Mobile and Wireless Network (MWN 2003)*, 2003.
23. M. Bohge and W. Trappe, "An authentication framework for hierarchical ad hoc sensor networks," in *Proc. of the 2003 ACM Workshop on Wireless Security*, 2003, pp. 79–87.
24. A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA broadcast authentication protocol," in *RSA Cryptobytes*, 2002.
25. D. Liu and P. Ning, "Multilevel  $\mu$  tesla: Broadcast authentication for distributed sensor networks," *Trans. Embedded Comput. Syst.*, vol. 3, no. 4, pp. 800–836, 2004.

26. P. Ning, A. Liu, and W. Du, "Mitigating dos attacks against broadcast authentication in wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 4, no. 1, pp. 1–35, 2008.
27. Q. Li and W. Trappe, "Reducing delay and enhancing dos resistance in multicast authentication through multigrade security," *IEEE Trans. Inform. Forens. Security*, vol. 1, no. 2, pp. 190–204, 2006.
28. Q. Zhang, P. Wang, D. Reeves, and P. Ning, "Defending sybil attacks in sensor networks," in *Proceedings of the International Workshop on Security in Distributed Computing Systems (SDCS-2005)*, 2005, pp. 185–191.
29. J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: Analysis and defenses," in *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, 2004, pp. 259–268.
30. L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe, "Fingerprints in the ether: Using the physical layer for wireless authentication," in *Proceedings of the IEEE International Conference on Communications*, 2007, pp. 4646–4651.
31. C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications*, 2003, pp. 113–127.
32. M. Talev, "Officials guard against leaks at gas stations," *Los Angeles Times*, Aug. 19, 2002.
33. Berkeley Intel Research, "Tiny application sensor kit," available: <http://berkeley.intel-research.net/task/>.
34. S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: A tiny aggregation service for ad-hoc sensor networks," in *Proceedings of the Usenix Symposium on Operating Systems Design and Implementation*, 2002.
35. C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*, Upper Saddle River, NJ: Prentice Hall, 2003.
36. D. Beneventano, S. Bergamaschi, S. Lodi, and C. Sartori, "Consistency checking in complex object database schemata with integrity constraints," *IEEE Trans. Knowledge Data Eng.*, vol. 10, no. 4, pp. 576–598, 1998.
37. F. Bry and R. Manthey, "Checking consistency of database constraints: A logical basis," in *Proceedings of the Twelfth International Conference on Very Large Data Bases*, 1986, pp. 13–20.
38. V. Barnett and T. Lewis, *Outliers in Statistical Data*, New York: Wiley, 1994.
39. M. Stonebraker, "Implementation of integrity constraints and views by query modification," in *SIGMOD '75: Proceedings of the 1975 ACM SIGMOD International Conference on Management of Data*, 1975, pp. 65–78.
40. P. Rousseeuw and A. Leroy, "Robust regression and outlier detection," Hoboken, NJ: Wiley-Interscience, 2003.
41. W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *MobiHoc '05: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2005, pp. 46–57.
42. Y. Law, P. Hartel, J. den Hartog, and P. Havinga, "Link-layer jamming attacks on s-mac," in *Proceedings of the 2nd European Workshop on Wireless Sensor Networks (EWSN 2005)*, 2005, pp. 217–225.
43. A. Wood and J. Stankovic, "Denial of service in sensor networks," *IEEE Computer*, vol. 35, no. 10, pp. 54–62, Oct. 2002.
44. W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel surfing and spatial retreats: Defenses against wireless denial of service," in *Proceedings of the 2004 ACM Workshop on Wireless Security*, 2004, pp. 80–89.

45. J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *SenSys '04: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, ACM Press, 2004, pp. 95–107.
46. A. Wood, J. Stankovic, and S. Son, "JAM: A jammed-area mapping service for sensor networks," in *24th IEEE Real-Time Systems Symposium*, 2003, pp. 286–297.
47. W. Xu, "On adjusting power to defend wireless networks from jamming," in *Proceedings of the First Workshop on the Security and Privacy of Emerging Ubiquitous Communication Systems*, 2007.
48. G. Noubir and G. Lin, "Low-power DoS attacks in data wireless lans and countermeasures," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 29–30, 2003.
49. K. Ma, Y. Zhang, and W. Trappe, "Mobile network management and robust spatial retreats via network dynamics," in *Proceedings of the 1st International Workshop on Resource Provisioning and Management in Sensor Networks (RPMSN05)*, 2005.
50. W. Xu, W. Trappe, and Y. Zhang, "Channel surfing: Defending wireless sensor networks from interference," in *IPSN '07: Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, 2007, pp. 499–508.
51. "WWWF—the conservation organization," available: <http://www.panda.org/>.
52. H. Lim and C. Kim, "Flooding in wireless ad-hoc networks," *IEEE Computer Commun.*, 2000.
53. Z. Cheng and W. Heinzelman, "Flooding strategy for target discovery in wireless networks," in *Proceedings of the Sixth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2003)*, 2003.
54. C. L. Barrett, S. J. Eidenbenz, L. Kroc, M. Marathe, and J. P. Smit, "Parametric probabilistic sensor network routing," in *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, 2003.
55. C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MobiCOM)*, Aug. 2000.
56. P. Th. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec, "Lightweight probabilistic broadcast," *ACM Trans. Computer Syst. (TOCS)*, vol. 21, no. 4, pp. 341–374, Nov. 2003.
57. D. Braginsky and D. Estrin, "Rumor routing algorithim for sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.
58. B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, June 2002, pp. 194–205.
59. D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proceedings of the IEEE GLOBECOM 2001*, Nov. 2001.
60. A. Savvides, C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *International Conference on Mobile Computing and Networks (MobiCOM)*, 2001, pp. 166–179.
61. P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of IEEE Infocom 2003*, 2000, pp. 775–784.
62. D. Niculescu and B. Nath, "Trajectory based forwarding and its applications," in *Proceedings of the Ninth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MobiCOM)*, Sept. 2003, pp. 260–272.
63. B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MobiCOM)*, Aug. 2000.

64. D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imielinski and H. Korth (Eds.), Kluwer Academic, 1996, pp. 153–181.
65. D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 4, pp. 11–25, Oct. 2001.



## INDEX

---

- 0–1 Integer programming formulation, 734–736  
2 × 2 Space–time diversity waveform design, 213–216  
Alamouti coding, 213–214  
complementary sequences, 213–214  
polarization diversity and radar detection, 215–216  
polarization diversity code design, 214–215  
2D Ultrasound imaging technology  
limitations, 368–372  
of current beamforming structure, 368–371  
3 × 3 Waveform scheduling, 228–229  
3D/4D Experimental system with planar phase array probe, 401–403  
3D/4D Ultrasound system technology, *See* Digital 3D/4D ultrasound imaging array  
4 × 4 Space–time diversity waveform design, 217–220  
4 × 4 polarization diversity radar detection, 219–220  
4 × 4 waveform scheduling, 217–218  
perfect reconstruction and separation, conditions for, 218–219  
6 × 6 Waveform scheduling, 228–229
- Access point selection, 798–799  
ACF, *See* Autocorrelation function (ACF)  
Acoustic array processing for speech enhancement, 231–264, *See also* Speech enhancement, acoustic array processing for  
Acoustic beamforming for hearing aid applications, 269–296  
Active wireless sensing (AWS), 117  
with wideband MIMO transceivers, 156–165  
angle–delay matched filtering, 159–160  
downlink, 163–165  
space–time communication architecture, 157–159  
uplink communication, 160–163  
Adaptive beamforming, 102–107, 245–246, 270  
adaptive noise canceler (ANC), 270  
advanced techniques, 106–107  
traditional techniques, 104–106  
for ultrasound systems, 375–376
- Adaptive cooperation strategies, 565–568  
incremental relaying, 566–568  
selective relaying, 565–566  
Adaptive-filter-based methods, 241–242  
Adaptive mixing, 263  
Adaptive MTSE, 762–763  
Adaptive scene analysis, WLAN, 797–799  
Additive white Gaussian noise (AWGN), 591  
Affine projection algorithm, 238  
Aggregation subgraph, 733  
Alamouti coding, 213–214  
Allen Telescope Array (ATA), 350, 360  
Ambiguity function, 14, 17  
Amplify-and-forward (AF) protocol, 561–565  
Analog approaches, to energy-efficient decentralized estimation, 476–484, *See also under* Energy-efficient decentralized estimation  
Analytes, 806  
Analytical known modulus algorithm (AKMA), 184  
Anchor nodes as location constraints, 416–417  
Anchor point selection, 798  
Angle–delay matched filtering, 159  
Angle difference of arrival (ADOA), 413  
Angular power spectrum, 130  
Antenna configuration, aperture synthesis, 348–352  
Aperture synthesis, correlation arrays, 344–366  
antenna configuration, 348–352  
Allen Telescope Array (ATA), 350  
Atacama Large Millimeter/submillimeter Array (ALMA), 349  
Australia Telescope Compact Array, 349  
Combined Array for Research in Millimeter-wave Astronomy (CARMA), 349  
Kohonen self-organized neural network algorithm, 350  
Very Large Array (VLA), 349  
aperture plane phased arrays, 361–362  
calibration, 357–358  
self-calibration, 357  
centimeter-wave arrays of parabolic reflectors, 360–361  
data acquisition and correlation, 346–348  
Earth rotation aperture synthesis, 344

- Aperture synthesis, correlation arrays (*Continued*)  
   focal plane arrays, 362–363  
   future directions, 362–364  
   imaging, 352–357  
     CLEAN algorithm, 354–355  
     CLEAN component, 355  
     CLEAN stripes, 355  
     deconvolution, 354–356  
     gridding, 354  
     maximum entropy method (MEM), 355  
   Molonglo Observatory Synthesis Telescope (MOST), 357  
   mosaicking, 356–357  
   natural weighting, 353  
   weighting, 353–354  
 interferometer measures, 344  
   visibility, 344  
 Low-Frequency Array (LOFAR), 361  
 millimeter arrays, 361  
 optical and infrared wavelengths, array processing at, 364  
 radio frequency interference (RFI), mitigation, 358–359  
 Square Kilometer Array (SKA), 363–364  
 submillimeter arrays, 361  
 Approximation-factor preserving reduction, 741  
 Array aperture, 31  
 Array processing, *See also under* Speech enhancement, acoustic array processing for digital communication systems, 173–203  
 IT-based estimation for multiple users exploiting, 191–201, *See also* Continuous transmission, IT-based estimation for multiple users exploiting; Packet transmission, IT-based estimation for multiple users exploiting  
 Astronomy, array processing in, 343–365, *See also* Correlation arrays  
 Atacama Large Millimeter submillimeter Array (ALMA), 349, 361  
 Australia Telescope Compact Array, 349  
 Australian Square Kilometer Array Pathfinder (ASKAP), 363  
 Autocorrelation function (ACF), 12  
 Autocovariance function, 34  
 Autoregressive moving-average (ARMA), 40  
 Autoregressive random process, 40  
 Availability-related threats, for sensor networks, 859–860  
 AWS, *See* Active wireless sensing (AWS)  
  
 Background noise suppression, in speech enhancement, 254–255  
 Barker codes, Kronecker products of, 223  
 Bartlett method, 37  
 Barycentric coordinates, 548  
 Base state model, 508  
 Baseline flooding, 877–878  
 Bayesian model, biochemical transport, 844–845  
   algorithm, 845  
   estimating initial time of release, 845  
   estimator, 845  
   likelihood, 844  
   posterior density, 844  
   prior model, 844  
 Bayesian tracking problem, 790–793  
 Beam power pattern for multifocus transmit beamformer  
   for linear phase array, 378–380  
   for planar phase array, 382–384  
 Beamforming, 242–249, *See also under* Hearing aid applications, acoustic beamforming for adaptive beamforming, 245–246 basics, 242–243 beamformer structures, 243–246 broadside array, 242 delay-and-sum beamformer, 243–244 echo cancellation and, 247–249 echo-to-noise ratio (ENR), 249 endfire arrays, 242 filter-and-sum beamformer, 244–245 generalized echo and interference canceler (GEIC), 248 generalized side-lobe and acoustic echo canceler (GSAEC), 248 robustness aspects, 247 speech recognition tests with digit loops, 247 steering direction, 242 structure of 2D ultrasound imaging limitations, 368–371 simplified beamforming structure, 370  
 Binary mask approach to underdetermined BSS, 312–320 clustering (Step 3), 319–320  
   k-means clustering algorithm, 319–320 feature extraction (Step 2), 314–319  
   feature vectors for k-means clustering, 316–317  
   feature vectors for multiple sensors, 317–318  
   features in MENUET, 317–318  
   modified features, 319  
 MAP-based two-stage approach and, experimental comparison, 328–335  
   experimental conditions, 328–329  
   sparseness assumption and anechoic assumption, validity, 331–332  
 separated signal reconstruction (Step 5), 320  
 separation (Step 4), 320  
   separation results, 329–331  
     with four sensors, 330–331  
     with three 2-D sensors, 329–330  
     with two sensors, 329  
   signal transformation to time–frequency domain (Step 1), 312–314  
 Binaural beamforming, 286–296  
   binaural multichannel Wiener filter, 289–290  
   configuration, 287–289  
   ITF cost function (SDW-MWF-ITF), 292–293

- microphone signals and output signals, 287–288  
 notation, 287–289  
 partial noise estimation (SDW-MWF- $\eta$ ), 280–281  
 performance comparison, 294–296  
 performance measures, 288–289  
 setup and performance measures, 293–294  
**Binaural hearing**, 270  
**Binaural multichannel Wiener filter**, 289–290  
**Biochemical transport**, 831–851  
 localizing the source(s), 843–846, *See also*  
     Bayesian model, biochemical transport  
     inverse problem and random field, 844–846  
 measurement model, 834–835  
     fluid simulations, 834–835  
     random effects, 834–835  
     transport model, 834–835  
 modeling, estimation, and detection in realistic environments, 831–851  
     detection framework, 832  
     inverse problem, 832  
     new numerical approach, 831  
 physical and statistical models, 832–835  
     assumptions, 832–833  
 physical dispersion model, 833–834  
     advection model, 833  
     boundary conditions, 833–834  
     initial substance distribution and sources, 834  
 sequential detection, 846–849  
     average run length, 847–848  
     expected delay before detection, 848–849  
     minimum signal level, 848  
     performance, 848–849  
     probability of detection, 848  
     threshold and false alarm rate, 847–848  
 simulations, 849  
     online detection, 849  
     performance measures, 849  
 transport modeling using Monte Carlo  
     approximation, 835–843, *See also* Monte Carlo approximation, biochemical transport modeling using  
**Biological ion channels**, 8051827, *See also*  
     Reconfigurable self-activating ion-channel-based biosensors  
**Blackman–Tukey method**, 35, 750  
**Blind method (BM)** in digital communications systems, 179  
**Blind source separation (BSS)**, 270–272, 303–337,  
*See also* Underdetermined blind source separation using acoustic arrays  
**Blind system identification (BSI)**, 304  
**Boussinesq equation**, 24  
**Broadcasting** in sensor networks, 662–663  
**Butterfly network**, 647  
  
**Canonical model**, 147  
**Capon's method**, 37, 46  
**Cayley–Menger determinants**, 548  
  
 CDMA, *See* Code division multiple access (CDMA) systems  
 Centimeter-wave arrays of parabolic reflectors, 360–361  
 Chain dependency graph, 742–743  
 Channel capacity, single-antenna systems, 142  
 Channel state information (CSI), 117  
 Classical beamforming, 44  
 Clique potentials, 728  
 Closed-form estimators, 420–424  
     classical multidimensional scaling, 420–421  
     procrustes alignment, 423–424  
     robust angulation using subspace techniques (RAST), 422  
     subspace-based multiangulation, 422–423  
 Closed-form SER expressions, 582–584  
 Cluster, 67–69  
 Clutter noise model, 444  
**CMT**, *See* Covariance matrix tapering (CMT)  
 Code division multiple access (CDMA) systems, 117, 212, 671  
     transceivers, 134  
 Coding gain, 151  
 Coding network vectors, 651–652  
 Coding vector, 650  
 Cognitive radios (CR), 749–779, *See also* Spectral estimation in cognitive radios  
 Coherence time, 121  
 Combination rules, 712–715  
 Combined Array for Research in Millimeter-wave Astronomy (CARMA), 361  
 Comfort noise, 252, 263  
 Computational auditory scene analysis (CASA), 270, 272, 305  
 Conditional distribution of measurements, in collaborative sensor networks, 444–445  
 Conjugate-symmetric transmit waveforms, 223–225  
 Consensus algorithms in sensor networks, 539–542  
     classification, 540–542  
     consensus in higher dimensions, 541, 544–545  
     leader–follower algorithms, 541, 545–548, *See also individual entry*  
     localization in sensor networks, 541  
     zero-dimension consensus, 541, 542–544  
     dimension, 540  
 Consensus problems, 533–534  
 Constant jammer, 868–869  
 Constant modulus algorithm (CMA), 192–195  
 Constant-velocity model, 443  
 Constellation rotation transmission (CRT), 184–185  
 Continuous transmission, IT-based estimation for multiple users exploiting, 191–199  
     source separation in flat channels, 191–195  
     based on TVTP-KMA, 192  
     multistage canceller based on the CMA (MSC-CMA), 194  
     multiuser CMA (MU-CMA), 194

- Continuous transmission, IT-based estimation for multiple users exploiting (*Continued*)  
source separation in frequency-selective channels, 195–198  
TVTP-KMA for, 195
- Conventional beamformer, 93
- Cooperative relay protocols, 561–568, *See also*  
Adaptive cooperation strategies; Fixed cooperation strategies  
cooperation protocols, 562  
cooperative communications, 561–562  
amplify-and-forward (AF) protocol, 561  
decode-and-forward (DF) protocol, 561  
fixed relaying, 561  
selective relaying, 561
- Cooperative sensor communications, 559–606, *See also*  
Cooperative relay protocols; Symbol error rate (SER) analysis  
energy efficiency in, 589–598  
cooperative transmission, 593–596  
direct transmission, 592–593  
multirelay scenario, 597–598  
performance analysis and optimum power allocation, 592–596  
system model, 590–592  
numerical examples, 600–606  
optimum power allocation, 575–577
- Coordinated turn rate model, 443
- Correlation arrays, 343–361  
aperture synthesis, 344–366, *See also individual entry*
- Correlation coefficients of demodulated signals, 768–770
- Correlogram, 35
- Cosine modulated multitone (CMT), 768
- COST 259 model, 69
- Coupon collector problem, 653–657
- Covariance matrix tapering (CMT), 105
- Cramér–Rao bounds (CRBs) for localization, 409, 414–417  
anchor nodes as location constraints, 416–417  
angle of arrival and angle difference of arrival, 415–416  
numerical examples, 417–420  
hybrid measurement systems, 419–420  
time of arrival versus time difference of arrival, 417–419  
received signal strength and received signal strength difference, 416  
time of arrival, time difference of arrival, and distances, 415
- Crucial function, 23
- Cyclostationarity in communications systems, 184–185
- Data-dependent superimposed training (DDST), 182
- Data-dependent waveform design, 226–228  
 $3 \times 3$  waveform scheduling, 228–229
- $6 \times 6$  waveform scheduling, 228–229  
reduced rank optimization, 227–228
- Data-independent beamformer, 274
- Decentralized deployment biosensors network, 816–826  
behavior prediction of other biosensors, 823–824  
global game formulation, 818–820  
‘active’ sensors, proportion, 819  
environment quality X and estimate Y, 819  
mode selection, 819–820  
reward function, 819–820  
sensor class, 819
- Nash equilibrium threshold strategies, 824–826  
reusability, 816  
threshold biosensor activation policies, 820–822
- Decentralized estimation scheme, 472
- Decentralized information processing, 807
- Deceptive jammer, 869
- Decision-directed approach, 259
- Decode-and-forward (DF) protocol, 561, 565
- Degenerate unmixing estimation technique (DUET), 304
- Degrees of freedom (DoF), 118
- Delay-and-sum beamformer, 243–244
- Delay–Doppler domain, 211–212
- Delay–Doppler scattering function, 120
- Delay spread, 119
- Demodulated signals, correlation coefficients of, 768–770
- Denial-of-service (DoS) attacks, for sensor networks, 859
- Dependency graph, 726
- Dereverberation, 255
- Diagonal loading, 104
- Diffusion adaptive solutions, 707–720  
combination rules, 712–715  
cooperation enhances stability, 715–719  
mean-square-error optimization, 710–712  
mean-square performance, 719–720  
node-based diffusion, 709–710  
ATC diffusion LMS, 710  
CTA diffusion LMS, 709  
simulation examples, 715
- Digital 3D/4D ultrasound imaging array, 367–404, *See also* 2D ultrasound imaging technology;  
Planar array ultrasound imaging system, experimental
- 3D visualization methods, current technology  
concept, 371–372  
freehand scanning, 372  
mechanical scanning, 371–372  
computing architecture and implementation issues, 392–393
- fully digital ultrasound system architecture,  
technological challenges for, 392–393
- next-generation technology, 372–392  
3D adaptive beamforming, 372

- adaptive beamforming structure for ultrasound systems, 375–376  
beam power pattern, 378–380  
digitizing large-size planar arrays, 373–374  
Fraunhofer's 3D and 4D visualization schemes, 372  
multifocus transmit beamformer for linear and planar phase arrays, 376–384  
multifocus transmit beamformer for linear phase array, 376–378  
multifocus transmit beamformer for planar phase array, 380–382  
PC-based computing architecture, 372  
receiving ultrasound beamformer for linear phase array, 384–388  
synthetic aperture processing, 372  
ultrasound beamforming structure for line and planar arrays, 374–375
- Digital approaches, to energy-efficient decentralized estimation, 472–476, *See also under* Energy-efficient decentralized estimation
- Digital communication systems, 173–203  
array processing for, 173–203  
basic elements of, 174  
implicit training (IT) for, 173–203, *See also individual entry*  
link model, 175–178  
open research problems, 201–203  
IT strategies, combination, 202  
multipacket reception challenge, 202–203  
parameter estimation problems, 202  
practical implementations, 201–202  
parameter estimation methods, 178–180  
system parameters, 175–178  
transmission paths, 177
- Digital subscriber line (DSL) technology, 767
- Digitizing large-size planar arrays, synthetic aperture processing for, 373–374
- Directed minimum spanning (DMST), 736
- Direction-of-arrival (DoA) estimation, 30, 92–102, 240–241  
4.2.7 unknown noise fields, 101–102  
imperfectly calibrated arrays, 96–97  
partly calibrated arrays, 97–99  
rapidly moving sources, 100–101  
signal model, 92–93  
time-varying arrays, 99–100  
traditional techniques, 93–96
- Dirichlet conditions, 840
- Discrete Fourier transform (DFT) representation, 35, 306–307
- Discrete-path model, 120
- Discrete-time Fourier transform (DTFT), 33
- Discrete wavelet multitone (DWMT), 768
- Dispersion relation, 21
- Distributed adaptive learning mechanisms, 695–721  
diffusion mode of cooperation, 696, 707–720, *See also Diffusion adaptive solutions*
- incremental mode of cooperation, 696, *See also Incremental adaptive solutions*  
motivation, 697–698  
notation, 696–697  
probabilistic diffusion mode of cooperation, 696
- Distributed algorithms in sensor networks, 533–553, *See also Consensus algorithms in sensor networks; Localization in sensor networks*  
distributed algorithms, 537–538  
distributed detection, 538–539  
linear system of equations, 551–553  
convergence, 552–553  
iteration matrix, design of, 552  
Markov chains, 536–537  
matrix theory, 536  
doubly stochastic, 536  
irreducible, 536  
nonnegative, 536  
primitive, 536  
row substochastic, 536  
row(column)-stochastic, 536  
strict diagonally dominant, 536  
preliminaries, 535–538  
spectral graph theory, 535  
balanced, 535  
directed graph, 535  
strongly connected, 535
- Distributed source coding (DSC), 609–639, *See also Multiterminal source coding*  
applications, 631–638  
code designs, 619–631  
multiterminal source code design, 629–631  
quadratic Gaussian case, 628–629  
Slepian–Wolf code, 619–624  
Wyner–Ziv (WZ) coding, 624–629  
Wyner–Ziv code, 624–629  
multiterminal video coding, 637–638  
Slepian–Wolf coding, 610–611  
theoretical background, 610–618  
Wyner–Ziv (WZ) coding, 609, 611–613
- Distributed spectrum sensing in cognitive radios, 773–776  
network signal model, 774–776  
numerical experiments, 776
- Diversity gain, 151
- Doppler spread, 119
- Double-directional channel model, 68
- Doubly selective MIMO channels, 153–154
- Dynamic positioning systems, WLAN, 790–796  
Bayesian tracking problem, 790–793  
Kalman filter, 793–494  
nonparametric information filter, 794–796
- Earth rotation aperture synthesis, 344
- Echo cancellation, 247–249
- Echo-to-noise ratio (ENR), 249
- Effective degrees of freedom (EDF), 763, 770–771, 777–778

- Eigenbeam model, 147  
 Eigenspace  
   CDMA transceivers, 155–156  
   Eigenspace-based beamformer, 105  
   STF transceivers, 152–153  
 Encryption, 858  
 Energy efficiency in cooperative sensor networks, 589–598, *See also under* Cooperative sensor communications  
 Energy-efficient decentralized estimation, 469–494  
   analog approaches, 476–484  
     coherent MAC, 478, 479  
     estimation diversity, 481–484  
     optimal power allocation, 478–481  
     orthogonal MAC, 477–479  
   analog versus digital, 470–471, 485–487  
   digital approaches, 472–476  
     power scheduling, 474–476  
     randomized quantization, 473–474  
   extension to vector model, 487–492  
     coherent MAC, 488–489, 490–492  
     noiseless channel case, 490–491  
     noisy channel case, 491–492  
     orthogonal MAC, 487–490  
   multiple-access (MAC) protocols, 471  
   system model, 471–472  
 Envelope estimation, speech enhancement, 261–262  
 Equivalent measurement-based algorithm, OOSM, 522–523  
 Estimation of signal parameters via rotational invariance technique (ESPRIT), 97  
 Euler–Bernoulli beam equation, 23  
 Expected error, 430–431  
 Experimental implementation of synthetic aperture algorithm (ETAM), 373–374  
 Explicit training (ET) paradigm in digital communications systems, 178–179  
 Extended Kalman filter (EKF), 499, 502–503  
 Extended Saleh–Valenzuela models (ESVMs), 70  
  
 Fake messages, flooding with, 878–879  
   persistent fake source, 879  
   short-lived fake source, 879  
 Fast Fourier transform (FFT), 32  
 FBSE, *See Filter bank spectral estimator (FBSE)*  
 Federal Aviation Administration (FAA) radar systems, 500  
 Feedback, pedestrian tracking in WLAN, 796–799  
   access point selection, 798–799  
   anchor point selection, 798  
   global feedback, 797  
   local feedback, 497  
   outlier mitigation, 799  
 Feynman–Kac formula, 837–838  
 Filter-and-sum beamformer, 244–245  
 Filter bank formulation  
   of spectral estimators, 750–751  
   polyphase realization of uniform filter banks, 751–752  
 Filter bank methods, 37  
 Filter bank multicarrier communication techniques, 767–768  
 Filter bank spectral estimator (FBSE), 766–773  
   correlation coefficients of demodulated signals, 768–770  
   digital subscriber line (DSL) technology, 767  
   effective degrees of freedom, 770–771  
   filter bank multicarrier communication techniques, 767–768  
   numerical experiments, 771–773  
   prototype filter, 768  
   sample spacing, 770–771  
 Filtered multitone (FMT), 768  
 Filtering, in multisensor data fusion, 499–511, *See also* Tracking filters, in multisensor data fusion  
 Fingerprinting, 785  
 Fingerprinting-based methods, 788  
 Finite impulse response (FIR), 38  
 Finite scatterer model, 72–73  
 Finite set statistics (FISST), 506  
 Fisher information matrix (FIM), 414  
 Fixed beamforming, 270  
   for binaural hearing, 270  
   for monaural hearing, 270  
 Fixed cooperation strategies, 562–565  
   amplify-and-forward protocol, 562–565  
   decode-and-forward protocol, 565  
 Fixed relaying, 561  
 Flat channels, source separation in, 191–195  
 Flooding, 877–880  
   baseline flooding, 877–878  
   with fake messages, 878–879  
     persistent fake source, 879  
     phantom flooding, 879–880  
     short-lived fake source, 879  
   probabilistic flooding, 878  
 Focal plane arrays, 362–363  
 Form of log-likelihood ratio for MRF, 731  
 Forwarding subgraph, 733  
 Four-corners diagram, 14  
 Fourier–Stieltjes integral, 12  
 Freehand scanning, in 3D visualization method, 372  
 Frequency-domain generalized side-lobe canceler, 276–280  
   relative transfer function (RTF) estimation, 279–280  
   suboptimal GSC, 277–279  
 Frequency-multiplexed training (FMT), 178  
 Frequency-selective channels, source separation in, 125, 195–198  
   linear space-time equalizer, 197  
   TVTP-KMA for source separation, 195–196  
 Frequency smoothing, 755  
 Fusion digraph, 733

- Gaussian mixture model (GMM) fitting, 319  
 Gauss–Markov random field (GMRF), 729–730  
   with acyclic dependency graph, 729  
 Gauss–Newton type, 48  
 Generalized echo and interference canceler (GEIC), 248  
 Generalized Fourier transform, 12  
 Generalized pseudo-Bayesian (GPB) algorithms, 507  
 Generalized side-lobe and acoustic echo canceler (GSAEC), 248  
 Generalized side-lobe canceler (GSC), 245–246, 270  
   mitigating leakage problem of, 284–285  
 Generations, network coding, 651–652  
 Geometrically based stochastic channel models (GSCMs), 67–70  
   drawbacks, 69  
 Global feedback, pedestrian tracking in WLAN, 797  
 Global games approach, 807, 817–820  
 Global variables, 16  
   frequency–wavenumber spectrum, 16  
   spatiotemporal correlation functions, 16  
 Golay complementary sequences, Kronecker products of, 221–223  
 Grating lobes, 244  
 Group Steiner tree, 738  
 Group velocities, 21
- Hammersley–Clifford theorem, 728  
 Hamming distance, 612  
 Harmonic-plus-noise model (HNM), 259  
 Harmonizable stochastic processes, 12–14  
   moment function, 12–14  
   wide-sense stationary, 14  
 Hearing aid applications, acoustic beamforming for, 269–296, *See also* Binaural beamforming;  
   Monaural beamforming; Noise reduction techniques  
 Higher dimensions, consensus in, 541, 544–545  
 Hybrid measurement systems, 419–420
- Identity-aware sensor networks, 664–665  
 Imperfectly calibrated arrays, 96–97  
 Implicit training (IT) for digital communication systems, 178–179, 180–186  
   classification, 180–186  
   constellation rotation transmission (CRT), 184–185  
   data-dependent superimposed training (DDST), 182  
   superimposed training, 181–183  
   time-varying transmitted power (TVTP), 183–184  
   training sequence synchronization (TSS), 181–182  
   cyclostationarity in, 184–185  
   for multiple users, 191–199, *See also under* Array processing
- for a single user, 186–191, *See also under* Single user, IT-based estimation for  
 Incremental adaptive solutions, 698–707  
   incremental LMS, 705–707  
   incremental solution, 701–705  
   steepest descent solution, 700–701  
 Incremental relaying, 566–568  
 Increment process, 12  
 Independent component analysis (ICA) techniques, 271  
 Indirect MT source coding, 616–618  
 Information filter, 502  
 Information-theoretic studies of wireless sensor networks, 669–689, *See also* Relay schemes  
   channel state information (CSI), 673  
   dense wireless sensor networks, 672  
   joint source–channel communication perspective, 673  
   wireless network coding, 684–688  
   broadcast channel with side information at receivers, 685–686  
 Information vector, 651  
 Integrity-related threats, for sensor networks, 858–859  
 Interacting multiple-model (IMM) estimator, 500, 507–510  
   algorithm, 508–510  
   interaction, 508–509  
   mode-conditioned filtering, 509  
   overall state estimate and covariance, 509–510  
   probability evaluation, 509  
   modeling assumptions, 508  
   base state model, 508  
   mode (modal state), 508  
   mode jump process, 508  
 Interaural level difference (ILD), 269  
 Interaural time difference (ITD), 269  
 Interaural transfer function (ITF), 289–296  
   ITF cost function (SDW-MWF-ITF), 292–293  
 Interference undernulling, 104  
 Inverse short-time Fourier transform (iSTFT), 273  
 Ion-channel-based biosensors, 805–827, *See also* Reconfigurable self-activating ion-channel-based biosensors
- Jamming attacks detection in sensor networks, 870–873  
   advanced detection strategies, 873  
   carrier sensing time, 872  
   packet delivery ratio (PDR), 872–873  
   signal strength, 871–872  
 Joint probabilistic data association (JPDA), 440, 445, 500, 514  
 Joint single-target tracking and classification, 448–452  
   class-based resampling scheme, 449–452  
   algorithm, 449–450  
   Kernel smoothing of belief state, 450–452

- Joint single-target tracking and classification  
*(Continued)*
- Kernel-based resampling, 451–452
  - related work, 448–449
- Kalman filter, 501–502, 793–494
- Kirkwood–Rihaczek (KR) spectrum, 13, 22
- k-means algorithm, 335–336
- Known modulus algorithm (KMA), 192–195
- Kohonen self-organized neural network algorithm, 350
- Korteweg–de Vries (KdV) equation, 24
- Kronecker products, 49, 72
  - of Barker codes, 223
  - conjugate-symmetric transmit waveforms, 223–225
  - Golay codes and half-band filters combination, 225–226
  - of Golay complementary sequences, 221–223
  - waveform families based on, 220–226
- Lag weighting, 755
- Laplacian rule, 713
- Leader-based tracking in sensor networks, 441
- Leader–follower algorithms, 541, 545–548
  - multiple anchors,  $n > 1$ , 546–548
  - one anchor,  $n = 1$ , 545–546
    - iteration matrix, 545–546
- Levy–Desplanques–Hadamard theorem, 536
- Line arrays, ultrasound beamforming structure for, 374–375
- Linear phase array probe, portable 2D/3D
  - experimental system with, 399–401
  - B-scan results, 399–401
  - dead zone label, 399
  - resolution array label, 399
  - volumetric 2D/3D imaging, 401
- Linear phase arrays
  - beam power pattern for multifocus transmit beamformer for, 378–380
  - multifocus receiving beamformer for, 384–392
  - multifocus transmit beamformer for, 376–384
    - receiving ultrasound beamformer for, 384–388
- Linear predictive coding (LPC), 261
- Linearly constrained minimum variance (LCMV) criterion, 244, 274
- Local feedback, pedestrian tracking in WLAN, 797
- Local processor assignment, 733–734
- Localization algorithms, sensor networks, 420–427
  - See also* Closed-form estimators; Statistically based estimators
  - related localization literature, 426–427
- Localization in sensor networks, 548–551
  - assumption, 549–550
    - (A0) nondegeneracy, 549
    - (A1) anchor nodes, 549
    - (A2) convexity, 549
    - (A3) triangulation, 549
- barycentric coordinates, 548
- Cayley–Menger determinants, 548
- convergence, 551
- convex hull, 548
- distributed localization algorithm, 550
- generalized volume, 548
- iteration matrix for localization, 550
- Loève spectrum, 13
- Local variables, 17
  - frequency–wavenumber spectrum, 16
  - spatiotemporal correlation functions, 16
- Long Wavelength Array (LWA), 361
- Lossless MT networks, Slepian–Wolf coding for, 633
- Loudspeaker–enclosure–microphone (LEM) systems, 232
- Low-Frequency Array (LOFAR), 361
- Magnitude subtraction, 254
- MAP-based two-stage approach to underdetermined BSS, 321–328, *See also* Binary mask approach to underdetermined BSS
  - blind source recovery (Step 2), 323–327
    - complex-valued  $l_1$ -norm minimization, 326–327
    - constrained  $l_1$ -norm minimization, 325
    - real-valued  $l_1$ -norm minimization, 325–326
    - shortest path algorithm and N–M source removal approach, 326
  - sparseness-based source model, 324–325
- blind system identification—hierarchical clustering (Step 1), 321–323
  - experimental conditions with, 332
  - permutation indeterminacy, 327
  - scaling indeterminacy, 327–328
  - separation procedures, 321
  - separation results, 332–335
- Markov chains, 53, 536–537
  - absorbing state, 536
- Markov random field (MRF), 725–730
  - definition, 726–730
  - dependency graph, 726
  - form of log-likelihood ratio for, 731
  - Gauss–Markov random field (GMRF), 729–730
  - general MRF, properties, 727–729
  - one-dimensional MRF, 726–727
  - properties, 726–730
  - statistical inference of, 730–731
- Matched-filter beamformer (MBF), 274
- Matrix theory, 536
- Maximum a posteriori (MAP) estimation, 425–426
- Maximum entropy method (MEM), 355
- Maximum-likelihood (ML) estimation, 424–425
- Maximum signal-to-noise ratio (MSNR)
  - beamformer, 274
- Mean-square-error optimization, 710–712
- Mean-square performance, 719–720
- Measurement model, tracking filters, 501

- Measurement-to-measurement association, in multisensor data fusion, 517–521
- Measurement-to-track association, in multisensor data fusion, 512–517
- Mechanical scanning, in 3D visualization method, 371–372
- Metropolis rule, 713
- Millimeter arrays, 361
- MIMO channels, 63–70
- physical channel models, 65–70
  - geometrically based stochastic models, 67–70
  - ring models, 65–67
  - stochastic models, 70
- MIMO radio propagation, 59–86
- measured channel characteristics, 75–86
  - analytical model parameterization, 75–79
  - stationarity, 81–86
  - temporal variations, 79–81
- propagation models, 64–75
- analytical models, 70–75
  - physical channel models, 65–70
- space–time propagation environment, 60–64
- channel capacity, 63–64
- Minimum cost fusion, formulation, 733–736
- Minimum mean-square error (MMSE), 271
- Minimum spanning-tree-based heuristic, 736–737
- Minimum variance distortionless response (MVDR)
- criterion, 103, 244, 270, 274–276
  - derivation, 274–275
  - MSNR and, equivalence between, 275–276
- Mobile collector model, 655
- Mobile node model, 655–657
- Modified (nonadaptive) MTSE, 766
- Molonglo Observatory Synthesis Telescope (MOST), 357, 361–362
- Monaural beamforming, 272–286, *See also*
- Multichannel Wiener filter (MWF)
  - data-independent beamformer, 274
  - frequency-domain generalized side-lobe canceler, 276–280
  - matched-filter beamformer (MBF), 274
  - maximum signal-to-noise ratio (MSNR)
    - beamformer, 274  - minimum-variance distortionless response (MVDR) beamformer, 274–276
- problem formulation, 272–274
- acoustical transfer function (ATF), 273
  - inverse short-time Fourier transform (iSTFT), 273
- Monaural hearing, 270
- Monte Carlo approximation, biochemical transport modeling using, 835–843
- Feynman–Kac formula, 837–838
- approximate stochastic diffusion and sensor measurements, 838
  - boundary conditions, 838
- Dirichlet conditions, 840
- infinite domain, 839
- Neumann conditions, 839
- simulation and convergence, 838–840
- likelihood, 842
- stochastic diffusion, 836–838
- diffusion equation, 836
  - stochastic process, 836
- stochastic transport model, 840
- unit response, 842
- wind turbulence modeling, 842–843
- Monte Carlo methods, 439–466
- Mosaicking, 356–357
- Moving-average process, 40
- MRF, *See* Markov random field (MRF)
- MTSE, *See* Multitaper spectral estimator (MTSE)
- Multiantenna transceivers, 115–118
- CDMA, 134–138
  - OFDM, 138–140
  - STF, 140–142
- Multichannel echo cancellation, 236–239, *See also*
- under* Speech enhancement, acoustic array processing for
- Multichannel speech enhancement system, 232
- Multichannel Wiener filter (MWF), 270–271, 280–286
- implementation, 282–283
  - linearly constrained adaptive beamformer, 280
  - mitigating leakage problem of GSC, 284–285
  - MMSE criterium, 281
  - speech distortion regularized GSC (SDR-GSC), 285
  - speech-distortion-weighted MWF (SDW-MWF), 282
  - TF-GSC and, 283–284
- Multidimensional scaling, 420–421
- Multifocus receiving beamformer for linear and planar phase arrays, 384–392
- Multifocus transmit beamformer
- for linear phase arrays, 376–380
  - for planar phase array, 380–382
- Multipath wireless channel modeling, 118–133
- nonselective multiantenna MIMO channels, 125–130
  - single-antenna channels, 119–125
  - time- and frequency-selective MIMO channels, 130–133
  - path partitioning in angle–delay–Doppler, 131–133
  - physical model, 130–131
  - sampling in angle–delay–Doppler, 131
- Multiple hypothesis tracker (MHT), 500, 514–515
- Multiple-input multiple-output-(MIMO) radars, 212, *See also* MIMO channels; MIMO radio
- propagation
- Multiple packet reception (MPR), 173, 199–201
- Multiple signal classification (MUSIC) algorithm, 32, 240, 750
- Multisensor data fusion, 499–527
- data association, 511–521

- Multisensor data fusion (*Continued*)
- filtering, 499–511, *See also* Tracking filters, in multisensor data fusion
  - fusing multisensor data, advantages, 525–527
  - fusion architectures, 511–512
    - centralized tracking, 511
    - decentralized tracking, 512
    - distributed tracking, 512
  - IMM estimator with KF comparison, 524–525
  - IMM-L and IMM-CT estimators comparison, 527
  - measurement-to-measurement association, 517–521
    - track formation with multiple sensors, 518
    - track formation with single sensor, 517–518
    - track-to-track association, 518–521
  - measurement-to-track association, 512–517
    - joint probabilistic data association (JPDA), 514
    - multidimensional (S-D) assignments, 516–517
    - multiple hypothesis tracker (MHT), 514–515
    - nearest neighbor (NN), 513
    - probabilistic data association (PDA), 513–514
    - strongest neighbor (SN), 513
    - two-dimensional assignment, 515–516
    - validation regions, 512
  - out-of-sequence measurements (OOSM), 521–524, *See also individual entry*
  - tracking with multiple sensors, 510
    - parallel updating, 511
    - sequential updating, 510
  - Multitaper spectral estimator (MTSE), 38, 46, 750, 757–766, 776–777
    - adaptive MTSE, 762–763
    - effective degrees of freedom (EDF), 763
    - modified (nonadaptive) MTSE, 766
    - power transfer function, 764–766
    - Slepian sequences, derivation, 759–760
      - eigen filters, 759
      - minimax theorem, 759
      - prolate filters, 760–762
  - Multitarget tracking and classification in
    - collaborative sensor networks, 441–443
    - crossing of tracked target with unknown target, 460–464
    - crossing of two tracked targets, 459–460
    - sensor selection, 456–458
      - expected information gain, 457–458
    - via sequential Monte Carlo methods, 439–466, *See also* Sequential Monte Carlo (SMC) methods
      - class-based resampling scheme, extension, 455
      - clutter noise model, 444
      - conditional distribution of measurements, 444–445
      - constant-velocity model, 443
      - coordinated turn rate model, 443
      - JMS formulation, 488–489, 455
      - leader-based tracking, 488–489, 455–456
      - optimal sampling density, 453–454
  - problem formulation, 440–445
  - sensing model, 444
  - system description, 440–445
  - target dynamics, 443
  - target-originated measurements, 444
  - Multitarget tracking, sensor data fusion application to, *See* Multisensor data fusion
  - Multiterminal source coding, 613–618, 629–631
    - direct MT source coding, 613–616
    - indirect MT source coding, 616–618
  - Multiterminal video coding, 637–638
  - Multivariate complex normal (MVCN) model, 74
  - Murchison Widefield Array (MWA), 361
  - Musical noise phenomenon, 254, 258
  - MWF, *See* Multichannel Wiener filter (MWF)
  - Nash equilibrium threshold strategies
    - for sensor activation, 824–825
    - for uniform observation noise, 825–826
  - Network coding for sensor networks, 645–666
    - broadcasting, 662–663
    - butterfly network, 647
    - coupon collector problem, 653–657
    - data collection, 655–657
      - mobile collector model, 655
      - mobile node model, 655–657
    - decentralized operation, 488–489, 660–662
    - description, 646–648
    - distributed function computation, 648
    - distributed storage and sensor network data persistence, 657–660
      - growth codes, 657–659
      - regenerating codes, 659–660
    - dynamically changing network, 648
    - identity-aware sensor networks, 664–665
    - implementation, 649–653
      - generations and coding vectors, 651–652
      - information vector, 651
      - operation over finite field, 650
      - randomized network coding, 650–651
      - subspace coding, 652–653
    - multipath diversity, 662–663
    - network, channel, and source coding, 663–664
    - restricted resources, 648
    - untuned radios, 661–662
  - Network coding, wireless, 684–688
    - broadcast channel with side information at receivers, 685–686
    - two-way relay channel, 686–688
  - Neumann conditions, 839
  - Node-based diffusion, 709–710
  - Noise reduction techniques, 270–272
    - blind source separation, 271
    - computational auditory scene analysis, 272
    - fixed beamforming, 270
      - for binaural hearing, 270
      - for monaural hearing, 270
    - multichannel Wiener filtering, 271

- Noise reference signals, 277  
 Noise subspace, 240  
 Nondispersive wave equation, 23  
 Nonparametric information (NI) filter, 794–796  
 Nonparametric techniques for pedestrian tracking in WLAN, 783–801  
     adaptive scene analysis (sensor selection), 797–799  
     cognition and feedback, 796–799  
         conceiving, 796  
         knowing, 796  
         perceiving, 796  
         reliability, 797  
     dynamic positioning systems, 790–796, *See also individual entry*  
     signal models, 786–788  
         fingerprinting-based methods, 788  
         radio propagation modeling, 787  
     zero-memory positioning, 788–790  
 Nonselective MIMO systems, 144–152  
     channel capacity, 147–148  
     eigenspace/beamspace signaling, 148–149  
     marginal and joint channel statistics, 145–147  
     space–time coding, 149–152  
 Nonselective multiantenna MIMO channels, 125–130  
     channel statistics and DoF, 127–130  
     path partitioning in angle 127–130  
     physical model, 126  
     sampling in angle, 126  
     spatial characteristics, 125–130  
     virtual channel representation, 126–127  
 Normalized least-mean squares (NLMS) algorithm, 237, 277  
  
 One-dimensional MRF, 726–727  
 Operation over finite field, 650  
 Optical and infrared wavelengths, array processing at, 364  
 Optimal routing, 731–744  
     for inference with local processing, 731–744  
     0–1 integer programming formulation, 734–736  
     approximation-factor preserving reduction, 741  
     chain dependency graph, 742–743  
     group Steiner tree, 738  
     i.i.d. measurements, 736  
     local processor assignment, 733–734  
     minimum cost fusion, formulation, 733–736  
     minimum spanning-tree-based heuristic, 736–737  
     network and communication model, 732  
     simplified integer program, 738–742  
     Steiner tree, 737–738  
     Steiner tree reduction, 738  
 Orthogonal frequency division multiplexing (OFDM), 116  
     transceivers, 138–140  
 Orthogonal space–time block code (OSTBC), 213  
  
 Outlier mitigation, 799  
 Out-of-sequence measurements (OOSM), in multisensor data fusion, 500, 521–524  
     multistep-lag OOSM, 522–524  
     equivalent measurement-based algorithm, 522–523  
     smoothing-based algorithm, 523–524  
     one-step-lag OOSM, 521–522  
 Overlap-add-method, 235  
  
 Packet delivery ratio (PDR), 872–873  
 Packet transmission, IT-based estimation for  
     multiple users exploiting, 199–201  
     multiple packet reception, 199–201  
     multipacket reception model, 199–200  
     signal processing for packet separation, 200–201  
     throughput versus complexity trade-offs, 201  
 Parameter estimation methods in digital communications systems, 178–180  
     blind method (BM), 179  
     explicit training (ET) paradigm, 178  
 Particle filter, 504–506  
     prediction, 505  
     reselection, 505–506  
     update, 505  
 Partly calibrated arrays, 97–99  
 Pedestrian tracking in WLAN, 783–801, *See also under* Nonparametric techniques for pedestrian tracking in WLAN  
 Periodic time-varying transmitted power (PTVTP), 183–184  
 Periodogram spectral estimator (PSE), 35, 750, 752–757, 776–777  
     common window functions, 753–754  
     prolate sequences, 754  
     spectral averaging, 755  
         averaging across frequency axis, 755  
         averaging across time axis, 755  
         spectrum smearing, 754–755  
 Permutation indeterminacy, 327  
 Persistent fake source, 879  
 Phantom flooding, 879–880  
 Phantom single-path routing, 881–882  
 Phased arrays, 361–362  
     aperture plane phased arrays, 361–362  
 Physical dispersion model, biochemical transport, 833–834  
     advective model, 833  
 Pisarenko’s method, 52  
 Pitch impulse generation, speech enhancement, 261–262  
 Planar array ultrasound imaging system,  
     experimental, 394–403  
     3D beamformer software, 396  
     multinode computing cluster, 395  
     performance, 398–403  
     system overview, 394–398

- Planar arrays, ultrasound beamforming structure for, 374–375
- Planar phase arrays  
3D/4D experimental system with, 401–403  
beam power pattern for multifocus transmit beamformer for, 382–384  
multifocus receiving beamformer for, 384–392  
multifocus transmit beamformer for, 380–382  
receiving ultrasound beamformer for, 388–392
- Plasma frequency, 24
- Plasma waves, 24
- Point-to-point MIMO wireless communication systems, 133–156  
nonselective MIMO systems, 144–152  
single-antenna systems, 133–144  
time- and frequency-selective systems, 152–156  
doubly selective MIMO channels, capacity of, 153–155  
eigenspace–CDMA transceivers, 155–156  
eigenspace–STF transceivers, 152–153
- Polarization diversity code design, 214–215
- Polyphase-based analysis system, 235
- Polyphase realization of uniform filter banks, 751–752
- Portable 2D/3D experimental system with linear phase array probe, 399–401
- Positioning architectures, WLAN, 785–786  
centralized positioning, 785–786  
decentralized positioning, 786
- Postechoes, 236
- Potential matrix, 729
- Power spectral density (PSD), 29, 749, 776–777
- Power subtraction, 253
- Power transfer function, 764–766
- Probabilistic data association (PDA), 500, 513–514
- Probabilistic flooding, 878
- Probability hypothesis density (PHD) method, 499, 506–507
- Process of measurement (PoM) attacks, sensor networks, 860–861  
classifier, 862  
enforcer, 862
- Procrustes alignment, 423–424
- Prolate filters, 760–762
- Prolate window, 754
- Prototype filter, 768
- Prototype low-pass filter, 236
- PSE, *See* Periodogram spectral estimator (PSE)
- Pseudo-optimal step size, 239
- Quadratic Gaussian case, 628–629
- Radar-phased arrays, 47
- Radar waveform diversity sets, unitary design of, 211–229, *See also* Data-dependent waveform design
- $2 \times 2$  space–time diversity waveform design, 213–216, *See also individual entry*
- $4 \times 4$  space–time diversity waveform design, 217–220, *See also individual entry*
- waveform families based on Kronecker products, 220–226, *See also* Kronecker products
- Radio frequency interference (RFI), mitigation, 358–359
- Radio propagation modeling, 787
- RAKE receiver, 135
- Random jammer, 869–870
- Randomized network coding, 650–651
- Rank reduction (RARE), 97
- Rauch–Tung–Streibel (RTS) backward recursion, 524
- Reactive jammer, 870
- Received signal strength difference (RSSD), 413
- Receiving ultrasound beamformer  
for linear phase array, 384–388  
for planar phase array, 388–392
- Reconfigurable self-activating ion-channel-based biosensors, 805–827  
activation control, 806  
analyte detection, 806  
biosensor response to analyte, dynamics, 809–812  
biosensors built of ion channels, 807–812  
concentration classification, 812–816  
decentralized deployment biosensors network, 816–826, *See also individual entry*  
input excitation control, 806  
joint input excitation design, 812–816  
optimal input excitation design, 812–813  
sequential multihypothesis test for analyte concentration, 813–816
- Recursive least-squares (RLS) algorithm, 238
- Recursive Wiener filtering, 255
- Relative-degree rule, 714
- Relative error decomposition, 427–434  
anchor evaluation, 433–434  
definitions, 428–430  
expected error, 430–431
- Relative transfer function (RTF) estimation, 279–280
- Relay schemes, 674–684  
application to sensor networks, 682–684  
general networks, 680–682  
multiple relays, 675–677  
multiple sources, 677  
multiple-block-decision schemes, 679  
oneblock-decision scheme, 679  
relay channel, 674–675  
two-source relay channel, 677–680
- Residual echo suppression, 255
- Residual interferences in speech enhancement, suppression, 252–259
- Reverberant speech  
sparseness of, 312  
speeches in, 305–307

- Reversible jump Markov chain Monte Carlo (RJMCMC), 55
- Rissanen Schwartz's method, 55
- Robust angulation using subspace techniques (RAST), 422
- ROOT MUSIC, 52
- Routing for statistical inference in sensor networks, 723–745, *See also* Optimal routing  
spatial data correlation, 724–730  
basic definitions, 725–726  
Markov random field (MRF), 725–730, *See also individual entry*  
notations, 725–726
- Saleh–Valenzuela model, 70
- Sample matrix inverse (SMI) beamformer, 104
- Scaling indeterminacy, 327–328
- Scene matching, 785
- Schur–Hadamard product, 73
- Second-order cone programming (SOCP), 321
- Secure biometrics, Slepian–Wolf coding for, 633
- Security and privacy for sensor networks, 855–883  
availability attacks against wireless link, 868–876  
constant jammer, 868–869  
deceptive jammer, 869  
random jammer, 869–870  
reactive jammer, 870  
challenges, 856–860  
availability-related threats, 859–860  
confidentiality-related threats, 857–858  
denial-of-service (DoS) attacks, 859  
encryption, 858  
integrity-related threats, 858–859  
Sybil attacks, 860  
detecting jamming attacks, 870–873, *See also*  
Jamming attacks detection in sensor networks  
ensuring integrity of measurement process, 860–868  
mapping jammed areas, 873–874  
measurement classifiers, 864–866  
multimodal consistency checks, 866  
physical sanity consistency checks, 864  
temporal consistency checks, 864–866  
measurement cleansing, 867–868  
relationship-based cleansing, 867–868  
robust estimation, 868  
measurement enforcers, 866–868  
filter enforcement strategies, 866  
measurement purging, marking, and  
passthrough, 867  
measurement monitoring, 861–864  
data flow, 862  
PoM monitor, 862–863  
process of measurement (PoM) attacks, 860  
recovering the network, 874–876  
channel surfing, 874–876  
coordinated channel switching, 876
- spectral multiplexing, 876
- routing contexts, ensuring privacy of, 876–882  
flooding, 877–880, *See also individual entry*  
single-path routing, 881–882
- Selective relaying, 561, 565–566
- Self-localization of sensor networks, 409–435  
algorithm classifications, 411  
centralized versus distributed, 411  
iterative versus closed form, 411  
relative versus absolute, 411  
statistical basis, 411
- error decomposition, 427–434, *See also* Relative error decomposition; Transformation error decomposition
- localization algorithms, 420–427, *See also*  
Localization algorithms, sensor networks
- measurement types, 412–420  
basic measurement systems, 412  
Cramér–Rao bounds (CRBs) for localization, 414–417, *See also individual entry*  
nuisance parameters, 413  
time-of-arrival measurements, 412
- performance bounds, 411–420  
source localization, 410
- Sensor array processing  
robustness issues in, 91–107  
adaptive beamforming, 102–107  
direction-of-arrival estimation, 92–102
- Sensor calibration, in speech enhancement  
procedure, 249–251, *See also under* Speech enhancement, acoustic array processing for
- Sensor data fusion application to multitarget tracking, 499–527, *See also* Multisensor data fusion
- Sensor networks  
multitarget tracking in, 439–466, *See also under*  
Multitarget tracking and classification in  
collaborative sensor networks  
self-localization of, 409–435, *See also*  
Self-localization of sensor networks
- Sequential Monte Carlo (SMC) methods, 446–448,  
*See also under* Multitarget tracking and  
classification in collaborative sensor networks  
resampling procedure, 447–448  
sequential importance sampling, 447
- Shadowing, 69
- Shiryaev–Roberts–Girshik–Rubin algorithm, 846
- Short-lived fake source, 879
- Short-time Fourier signaling (STF), 140–142  
transceivers, 140–142
- Short-time Fourier transform (STFT) processing, 272
- Signal cancelation effect, 247
- Signal processing for packet separation, 200–201
- Signal self-nulling, 104
- Signal subspace, 240
- Signal-to-interference-plus-noise ratio (SINR), 103
- Signal-to-noise ratios (SNRs), 50
- Simplified integer program, 738–742

- Single-antenna channels, 119–125  
     channel statistics and DoF, 123–125  
     path partitioning in delay–Doppler, 123–125  
     physical discrete-path model, 120–121  
     sampling in delay–Doppler, 121–123  
     virtual channel representation, 121–123
- Single-antenna systems, 133–144  
     CDMA transceivers, 134–138  
     CDMA versus OFDM/STF signaling, 144  
     channel capacity, 142–144  
     OFDM transceivers, 138–140  
     STF transceivers, 140–142
- Single-bounce model, 66
- Single-path routing, 881–882  
     alternating single-path routing, 881  
     phantom single-path routing, 881–882
- Single user, IT-based estimation for, 186–191  
     channel equalization, 190–191  
         CRT-based, 191  
         ST-based, 190–191  
         TVTP-based, 191
- channel estimation, 190–191  
     CRT-based, 190  
     ST-based, 190  
     TVTP-based, 190
- data detection, 191
- DC offset estimation and compensation, 188–189  
     CRT-based, 189  
     ST-based, 188  
     TVTP-based, 189
- discrete-time representation, 187–188
- frame and training sequence synchronization, 189–190  
     CRT-based, 190  
     ST-based, 189  
     training sequence synchronization (TSS), 189  
     TVTP-based, 190
- frequency offset estimation and compensation, 189  
     CRT-based, 189  
     ST-based, 189  
     TVTP-based, 189
- system model, 187–188
- Singular value decomposition (SVD), 63
- Slepian sequences, derivation, 759–760
- Slepian–Wolf (SW) coding, 610–611  
     design, 619–624  
         encoding, 488–489, 621–622  
         decoding, 488–489, 622  
     lossless MT networks, 633  
     for secure biometrics, 633  
     for two sources, 611
- Smoothing-based algorithm, OOSM, 523–524
- Source separation  
     in flat channels, 191–195  
     in frequency-selective channels, 195–198
- Space–time coding, 149
- Space–time multipath environment, 60–61
- Sparserness of speech sources, 307–312
- sparseness of reverberant speech, 312
- sparsest representation with STFT, 311–312
- Sparserness-based source model, 324–325
- Spatial postfiltering, 256–259
- Spectral estimation in cognitive radios, 749–779  
     distributed spectrum sensing, 773–776, *See also*  
         Distributed spectrum sensing in cognitive  
             radios  
     effective degree of freedom, 777–778  
     filter bank formulation, 750–751  
         polyphase realization of uniform filter banks,  
             751–752  
         prototype filter, 751  
     filter bank spectral estimator (FBSE), 766–773,  
         *See also individual entry*  
     multitaper spectral estimator, 757–766, *See also*  
         *individual entry*  
     periodogram spectral estimator (PSE), 752–757,  
         *See also individual entry*
- Spatial spectrum estimation, 29–56  
     fundamentals, 33–34  
     model, 41  
     nonparametric methods, 44–46  
         Capon’s method, 46  
         classical beamforming, 44  
         multitaper method, 46  
     number of impinging signals, 54–56  
         information-theoretic approaches, 54–55  
     RJMCMC, 55–56  
     parametric methods, 47–54  
         Bayesian method based in MCMC, 53  
         deterministic model of signals, 47  
         ESPRIT, 52  
         MUSIC, 51  
         stochastic maximum-likelihood method, 49  
         subspace fitting method, 50  
         temporal spectrum estimation, 34–41
- Spectral floor, 254
- Spectral graph theory, 535
- Spectrum smearing, 754–755
- Speech distortion regularized GSC (SDR-GSC), 285
- Speech enhancement, acoustic array processing for, 231–264, *See also* Beamforming  
     acoustic environments, 232–233
- multichannel echo cancellation, 236–239  
     adaptation algorithms, 237–238  
     adaptation control, 238–239  
     affine projection algorithm, 238  
     pseudo-optimal step size, 239  
     recursive least-squares (RLS) algorithm, 238
- multichannel speech enhancement system, 232
- multichannel systems, special problems of, 239
- postprocessing, 252–264  
     background noise suppression, 254–255  
     dereverberation, 255  
     residual echo suppression, 255  
     residual interferences, suppression, 252–259  
     spatial postfiltering, 256–259

- sensor calibration, 249–251  
     adaptation control, 250–251  
     adaptive calibration, 249–250
- signal processing in subband domain, 233–236  
     DFT-based analysis–synthesis system, 234–235  
     polyphase-based analysis system, 235  
     postechoes, 236  
     prototype low-pass filter, 236
- signal reconstruction, postprocessing, 259–264  
     adaptive mixing, 263  
     comfort noise, 263  
     envelope estimation, 261–262  
     pitch impulse generation, 261–262  
     speech synthesis, 262–263
- speaker localization, 240–242  
     adaptive-filter-based methods, 241–242  
     DoA estimation, basic concepts, 240–241  
     generalized cross-correlation (GCC) function, 241  
     multiple signal classification (MUSIC) algorithm, 240  
     speech signal properties, 232–233
- Speech-distortion-weighted MWF (SDW-MWF), 271, 282–283
- Spread-spectrum signaling, 134–138
- Square Kilometer Array (SKA), 363–364
- Stationary manifold, 14
- Statistical inference in sensor networks, 723–745,  
*See also* Routing for statistical inference in sensor networks
- Statistical inference of MRF, 730–731
- Statistically based estimators, 424–426  
     maximum a posteriori (MAP) estimation, 425–426  
     maximum-likelihood (ML) estimation, 424–425
- Steepest descent solution, 700–701
- Steering direction, beamforming, 242
- Steiner tree, 737–738  
     reduction, 738
- Stochastic diffusion, biochemical transport, 836–838
- Stochastic transport model, 840
- Stochastic wavefields, 15–19  
     estimation, 19  
     generalized coherences, 18  
     Hilbert space, 18  
     linear spatiotemporal systems, 19  
     second-order moment functions, 15–18  
         frequency–wavenumber spectrum, 16  
         spatiotemporal correlation functions, 16  
         stationary and homogeneous manifold, 17–18
- Subband domain, speech signal processing in, 233–236, *See also under* Speech enhancement, acoustic array processing for
- Submillimeter arrays, 361
- Suboptimal GSC, 277–279
- Subspace-based methods, 32
- Subspace-based multiangulation, 422–423
- Subspace coding, 652–653
- Sum-rate bound, 618
- Superimposed training (ST), 181–183  
     data-dependent superimposed training (DDST), 182  
     training sequence synchronization (TSS), 181–182
- Sybil attacks, 860
- Symbol error rate (SER) analysis, 568–589  
     for AF cooperative communications, 577–585  
         MGF approach, 577–579  
         simple MGF expression for harmonic mean, 579–582
- AF optimum power allocation, 584–585  
     asymptotically tight approximation, 582–584  
     closed-form SER expressions, 582–584
- DF and AF cooperation gains, comparison, 585–589
- for DF cooperative communications, 568–577  
     optimal power allocation and, 575–577
- SER upper bound and asymptotically tight approximation, 571–574
- Synthesized beam, 344
- System model, tracking filters, 501
- Target-originated measurements, in collaborative sensor networks, 444
- Temporal spectrum estimation, 34–41  
     nonparametric methods, 34–38  
         Blackman-Tukey spectrum estimator, 35–37  
         Capon’s method, 37  
         correlogram, 34–35  
         multitaper method, 38  
         periodogram, 35  
         parametric methods, 39–41
- Threshold biosensor activation policies, 820–822  
     optimality of pure policies, 821–822  
     threshold Nash equilibrium, 821
- Threshold Nash equilibrium, 821–825
- Time-bandwidth product, 761
- Time difference of arrival (TDOA), 413–417
- Time-multiplexed training (TMT), 178
- Time-selective channels, 125
- Time-varying arrays, 99–100
- Time-varying transmitted power (TVTP), 183–184  
     disadvantage, 183  
     periodic TVTP operation, 183
- Tracking filters, in multisensor data fusion, 500–511  
     extended Kalman filter, 502–503  
     information filter, 502  
     interacting multiple-model (IMM) estimator, 507–510, *See also individual entry*  
         Kalman filter, 501–502  
         measurement model, 501  
         particle filter, 504–506  
             prediction, 505  
             reselection, 505–506  
             update, 505
- probability hypothesis density method, 506–507  
     prediction, 506–507  
     update, 507

- Tracking filters, in multisensor data fusion  
*(Continued)*
- system model, 501
  - unscented Kalman filter, 503–504
    - recursion, 503–504
    - sigma point generation, 503
- Track-to-track association, in multisensor data fusion, 518–521
- tracklet fusion, 520–521
  - for tracks with dependent errors, 518–520
- Training sequence synchronization (TSS), 181–182
- Transformation error decomposition, 427–434
  - anchor evaluation, 433–434
  - definitions, 428–430
  - expected error, 430–431
- Transport capacity, 670
- Two-way relay channel, 686–688
- Ultrasound beamforming structure for line and planar arrays, 374–375
- Ultrasound systems, adaptive beamforming structure for, 375–376
- Underdetermined blind source separation using acoustic arrays, 303–337, *See also* MAP-based two-stage approach
- binary mask approach to, 312–320, *See also under* Binary mask approach to underdetermined BSS
  - discrete Fourier transform (DFT) representation, 306–307
  - k*-means clustering, initial values for, 335–336
  - MAP-based two-stage approach, 304
  - objective, 307
  - sparseness-based, 488–489, 307–312
  - speeches in reverberant environments, 305–307
    - discrete time-domain representation, 305–306
- Unitary design of radar waveform diversity sets, 211–229, *See also* Radar waveform diversity sets, unitary design of
- Unscented Kalman filter (UKF), 499, 503–504
  - recursion, 503–504
  - sigma point generation, 503
- Untuned radios in sensor networks, 661–662
- Uplink communication, 160–163
  - sensing capacity, 161–162
- Van Cittert–Zernicke theorem, 346
- Vector function, 21
- Very Large Array (VLA), 349, 360
- Very Long Baseline Array (VLBA), 360
- Very Long Baseline Interferometry (VLBI), 360
- Vestigial sideband (VSB) modulation, 768
- Virtual channel model, 73
- Volumetric 2D/3D imaging, 401
- Water-filling constant, 64
- Wave dispersion, 19–26
- applications, 23–24
- Euler–Bernoulli beam equation, 23
  - long water waves, 24
  - nondispersive wave equation, 23
  - plasma waves, 24
- dispersion relation, 21
- group velocities, 21
  - linear systems, 20
  - moment fuctions, 21–22
    - homogeneous driving force field, 22
    - stationary, 22
  - phase, 21
- Wavefields, 11–26
  - harmonizable stochastic processes, 12–14
  - stochastic wavefields, 15–19
  - wave dispersion, 19–26
- Wavefront, 41
- W-disjoint orthogonality (WDO), 311
- Weichselberger coupling matrix, 78
- Weichselberger model, 73–74
  - coupling modes, 80
- Weighted overlapped segment averaging (WOSA), 755, 777
- Wideband MIMO transceivers, 156–162
- Wide-sense stationary (WSS), 14
- Wide-sense stationary uncorrelated scattering (WSSUS) model, 118
- Wiener filter, 253
- Wind turbulence modeling, 842–843
- Wireless information retriever (WIR), 156
- Wireless local area network (WLAN), 783–801
  - advantages, 783–784
    - consensual sensing and tracking, 784
    - cost effectiveness, 783–784
    - scalability, 784
  - challenges, 784
    - unknown RSS position dependency, 784–785
    - unpredictable RSS variations, 785
  - pedestrian tracking in, 783–801, *See also under* Nonparametric techniques for pedestrian tracking in WLAN
  - positioning architectures, 785–786
    - centralized positioning, 785–786
    - decentralized positioning, 786
- Wold's theorem, 33
- Wyner–Ziv (WZ) coding, 609, 611–613
  - for compress-forward relaying and receiver cooperation, 635–637
  - design, 624–629
    - binary symmetric case, 625–628
    - with side information, 611
    - successive WZ coding, 629
    - video coding, 633–635
- Zero-dimension consensus, 541, 542–544
  - convergence, 543–544
  - iteration matrix, design of, 543
- Zero-memory positioning, WLAN, 788–790