Optimal State Estimation

Optimal State Estimation

Kalman, H_∞, and Nonlinear Approaches

Dan Simon
Cleveland State University

WILEY-INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

Copyright © 2006 by John Wiley & Sons, Inc. All rights reserved.

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

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008 or online at http://www.wiley.com/go/permission.

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

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

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic format. For information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication is available.

ISBN-13 978-0-471-70858-2
ISBN-10 0-471-70858-5

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

CONTENTS

Acknowledgments xiii

Acronyms xv

List of algorithms xvii

Introduction xxi

PART I INTRODUCTORY MATERIAL

1 Linear systems theory 3

1.1 Matrix algebra and matrix calculus 4

1.1.1 Matrix algebra 6

1.1.2 The matrix inversion lemma 11

1.1.3 Matrix calculus 14

1.1.4 The history of matrices 17

1.2 Linear systems 18

1.3 Nonlinear systems 22

1.4 Discretization 26

1.5 Simulation 27

1.5.1 Rectangular integration 29

1.5.2 Trapezoidal integration 29

1.5.3 Runge–Kutta integration 31

1.6 Stability 33

1.6.1 Continuous-time systems 33
1.6.2 Discrete-time systems 37
1.7 Controllability and observability 38
1.7.1 Controllability 38
1.7.2 Observability 40
1.7.3 Stabilizability and detectability 43
1.8 Summary 45
Problems 45

2 Probability theory 49

2.1 Probability 50
2.2 Random variables 53
2.3 Transformations of random variables 59
2.4 Multiple random variables 61
2.4.1 Statistical independence 62
2.4.2 Multivariate statistics 65
2.5 Stochastic Processes 68
2.6 White noise and colored noise 71
2.7 Simulating correlated noise 73
2.8 Summary 74
Problems 75

3 Least squares estimation 79

3.1 Estimation of a constant 80
3.2 Weighted least squares estimation 82
3.3 Recursive least squares estimation 84
3.3.1 Alternate estimator forms 86
3.3.2 Curve fitting 92
3.4 Wiener filtering 94
3.4.1 Parametric filter optimization 96
3.4.2 General filter optimization 97
3.4.3 Noncausal filter optimization 98
3.4.4 Causal filter optimization 100
3.4.5 Comparison 101
3.5 Summary 102
Problems 102

4 Propagation of states and covariances 107

4.1 Discrete-time systems 107
4.2 Sampled-data systems 111
4.3 Continuous-time systems 114

4.4 Summary 117
Problems 117

PART II THE KALMAN FILTER

5 The discrete-time Kalman filter 123

5.1 Derivation of the discrete-time Kalman filter 124
5.2 Kalman filter properties 129
5.3 One-step Kalman filter equations 131
5.4 Alternate propagation of covariance 135
5.4.1 Multiple state systems 135
5.4.2 Scalar systems 137
5.5 Divergence issues 139
5.6 Summary 144
Problems 145

6 Alternate Kalman filter formulations 149

6.1 Sequential Kalman filtering 150
6.2 Information filtering 156
6.3 Square root filtering 158
6.3.1 Condition number 159
6.3.2 The square root time-update equation 162
6.3.3 Potter's square root measurement-update equation 165
6.3.4 Square root measurement update via triangularization 169
6.3.5 Algorithms for orthogonal transformations 171
6.4 U-D filtering 174
6.4.1 U-D filtering: The measurement-update equation 174
6.4.2 U-D filtering: The time-update equation 176
6.5 Summary 178
Problems 179

7 Kalman filter generalizations 183

7.1 Correlated process and measurement noise 184
7.2 Colored process and measurement noise 188
7.2.1 Colored process noise 188
7.2.2 Colored measurement noise: State augmentation 189
7.2.3 Colored measurement noise: Measurement differencing 190
7.3 Steady-state filtering 193
7.3.1 α-β filtering 199
7.3.2 α-β-γ filtering 202
7.3.3 A Hamiltonian approach to steady-state filtering 203
7.4 Kalman filtering with fading memory 208

7.5 Constrained Kalman filtering 212
7.5.1 Model reduction 212
7.5.2 Perfect measurements 213
7.5.3 Projection approaches 214
7.5.4 A pdf truncation approach 218
7.6 Summary 223
Problems 225

8 The continuous-time Kalman filter 229

8.1 Discrete-time and continuous-time white noise 230
8.1.1 Process noise 230
8.1.2 Measurement noise 232
8.1.3 Discretized simulation of noisy continuous-time systems 232
8.2 Derivation of the continuous-time Kalman filter 233
8.3 Alternate solutions to the Riccati equation 238
8.3.1 The transition matrix approach 238
8.3.2 The Chandrasekhar algorithm 242
8.3.3 The square root filter 246
8.4 Generalizations of the continuous-time filter 247
8.4.1 Correlated process and measurement noise 248
8.4.2 Colored measurement noise 249
8.5 The steady-state continuous-time Kalman filter 252
8.5.1 The algebraic Riccati equation 253
8.5.2 The Wiener filter is a Kalman filter 257
8.5.3 Duality 258
8.6 Summary 259
Problems 260

9 Optimal smoothing 263

9.1 An alternate form for the Kalman filter 265
9.2 Fixed-point smoothing 267
9.2.1 Estimation improvement due to smoothing 270
9.2.2 Smoothing constant states 274
9.3 Fixed-lag smoothing 274
9.4 Fixed-interval smoothing 279
9.4.1 Forward–backward smoothing 280
9.4.2 RTS smoothing 286
9.5 Summary 294
Problems 294

10 Additional topics in Kalman filtering **297**

10.1 Verifying Kalman filter performance 298
10.2 Multiple-model estimation 301
10.3 Reduced-order Kalman filtering 305
10.3.1 Anderson's approach to reduced-order filtering 306
10.3.2 The reduced-order Schmidt–Kalman filter 309
10.4 Robust Kalman filtering 312
10.5 Delayed measurements and synchronization errors 317
10.5.1 A statistical derivation of the Kalman filter 318
10.5.2 Kalman filtering with delayed measurements 320
10.6 Summary 325
Problems 326

PART III THE H_∞ FILTER

11 The H_∞ filter **333**

11.1 Introduction 334
11.1.1 An alternate form for the Kalman filter 334
11.1.2 Kalman filter limitations 336
11.2 Constrained optimization 337
11.2.1 Static constrained optimization 337
11.2.2 Inequality constraints 339
11.2.3 Dynamic constrained optimization 341
11.3 A game theory approach to H_∞ filtering 343
11.3.1 Stationarity with respect to x_0 and w_k 345
11.3.2 Stationarity with respect to $\hat{x}$ and y 347
11.3.3 A comparison of the Kalman and H_∞ filters 354
11.3.4 Steady-state H_∞ filtering 354
11.3.5 The transfer function bound of the H_∞ filter 357
11.4 The continuous-time H_∞ filter 361
11.5 Transfer function approaches 365
11.6 Summary 367
Problems 369

12 Additional topics in H_∞ filtering **373**

12.1 Mixed Kalman/H_∞ filtering 374
12.2 Robust Kalman/H_∞ filtering 377
12.3 Constrained H_∞ filtering 381
12.4 Summary 388
Problems 389

PART IV NONLINEAR FILTERS

13 Nonlinear Kalman filtering **395**

13.1 The linearized Kalman filter 397
13.2 The extended Kalman filter 400
13.2.1 The continuous-time extended Kalman filter 400
13.2.2 The hybrid extended Kalman filter 403
13.2.3 The discrete-time extended Kalman filter 407
13.3 Higher-order approaches 410
13.3.1 The iterated extended Kalman filter 410
13.3.2 The second-order extended Kalman filter 413
13.3.3 Other approaches 420
13.4 Parameter estimation 422
13.5 Summary 425
Problems 426

14 The unscented Kalman filter **433**

14.1 Means and covariances of nonlinear transformations 434
14.1.1 The mean of a nonlinear transformation 434
14.1.2 The covariance of a nonlinear transformation 437
14.2 Unscented transformations 441
14.2.1 Mean approximation 441
14.2.2 Covariance approximation 444
14.3 Unscented Kalman filtering 447
14.4 Other unscented transformations 452
14.4.1 General unscented transformations 452
14.4.2 The simplex unscented transformation 454
14.4.3 The spherical unscented transformation 455
14.5 Summary 457
Problems 458

15 The particle filter **461**

15.1 Bayesian state estimation 462
15.2 Particle filtering 466
15.3 Implementation issues 469
15.3.1 Sample impoverishment 469
15.3.2 Particle filtering combined with other filters 477
15.4 Summary 480
Problems 481

Appendix A: Historical perspectives **485**

Appendix B: Other books on Kalman filtering **489**

Appendix C: State estimation and the meaning of life **493**

References 501

Index 521

ACKNOWLEDGMENTS

The financial support of Sanjay Garg and Donald Simon (no relation to the author) at the NASA Glenn Research Center was instrumental in allowing me to pursue research in the area of optimal state estimation, and indirectly led to the idea for this book. I am thankful to Eugenio Villaseca, the Chair of the Department of Electrical and Computer Engineering at Cleveland State University, for his encouragement and support of my research and writing efforts. Dennis Feucht and Jonathan Litt reviewed the first draft of the book and offered constructive criticism that made the book better than it otherwise would have been. I am also indebted to the two anonymous reviewers of the proposal for this book, who made suggestions that strengthened the material presented herein. I acknowledge the work of Sandy Buettner, Joe Connolly, Classica Jain, Aaron Radke, Bryan Welch, and Qing Zheng, who were students in my Optimal State Estimation class in Fall 2005. They contributed some of the problems at the end of the chapters and made many suggestions for improvement that helped clarify the subject matter. Finally I acknowledge the love and support of my wife, Annette, whose encouragement of my endeavors has always been above and beyond the call of duty.

D. J. S.

ACRONYMS

ACR	Acronym
ARE	Algebraic Riccati equation
CARE	Continuous ARE
DARE	Discrete ARE
EKF	Extended Kalman filter
erf	Error function
FPGA	Field programmable gate array
GPS	Global Positioning System
HOT	Higher-order terms
iff	If and only if
INS	Inertial navigation system
LHP	Left half plane
LTI	Linear time-invariant
LTV	Linear time-varying
MCMC	Markov chain Monte Carlo
MIMO	Multiple input, multiple output
$N(a, b)$	Normal pdf with a mean of a and a variance of b
pdf	Probability density function

PDF	Probability distribution function
QED	Quod erat demonstrandum (i.e., "that which was to be demonstrated")
RHP	Right half plane
RMS	Root mean square
RPF	Regularized particle filter
RTS	Rauch–Tung–Striebel
RV	Random variable
SIR	Sampling importance resampling
SISO	Single input, single output
SSS	Strict-sense stationary
SVD	Singular value decomposition
TF	Transfer function
$U(a, b)$	Uniform pdf that is nonzero on the domain $[a, b]$
UKF	Unscented Kalman filter
WSS	Wide-sense stationary

LIST OF ALGORITHMS

Chapter 1: Linear systems theory
Rectangular integration 29
Trapezoidal integration 31
Fourth-order Runge–Kutta integration 32

Chapter 2: Probability theory
Correlated noise simulation 74

Chapter 3: Least squares estimation
Recursive least squares estimation 86
General recursive least squares estimation 88

Chapter 5: The discrete-time Kalman filter
The discrete-time Kalman filter 128

Chapter 6: Alternate Kalman filter formulations
The sequential Kalman filter 151
The information filter 156
The Cholesky matrix square root algorithm 160
Potter's square root measurement-update algorithm 166
The Householder algorithm 171
The Gram–Schmidt algorithm 172
The U-D measurement update 175
The U-D time update 177

Chapter 7: Kalman filter generalizations
The general discrete-time Kalman filter 186
The discrete-time Kalman filter with colored measurement noise 191
The Hamiltonian approach to steady-state Kalman filtering 207
The fading-memory filter 210

Chapter 8: The continuous-time Kalman filter
The continuous-time Kalman filter 235
The Chandrasekhar algorithm 244
The continuous-time square root Kalman filter 247
The continuous-time Kalman filter with correlated noise 249
The continuous-time Kalman filter with colored measurement noise 251

Chapter 9: Optimal smoothing
The fixed-point smoother 269
The fixed-lag smoother 278
The RTS smoother 293

Chapter 10: Additional topics in Kalman filtering
The multiple-model estimator 302
The reduced-order Schmidt–Kalman filter 312
The delayed-measurement Kalman filter 324

Chapter 11: The H_∞ filter
The discrete-time H_∞ filter 353

Chapter 12: Additional topics in H_∞ filtering
The mixed Kalman/H_∞ filter 374
The robust mixed Kalman/H_∞ filter 378
The constrained H_∞ filter 385

Chapter 13: Nonlinear Kalman filtering
The continuous-time linearized Kalman filter 399
The continuous-time extended Kalman filter 401
The hybrid extended Kalman filter 405
The discrete-time extended Kalman filter 409
The iterated extended Kalman filter 411
The second-order hybrid extended Kalman filter 416
The second-order discrete-time extended Kalman filter 419
The Gaussian sum filter 421

Chapter 14: The unscented Kalman filter
The unscented transformation 446
The unscented Kalman filter 448
The simplex sigma-point algorithm 454
The spherical sigma-point algorithm 455

Chapter 15: The particle filter

The recursive Bayesian state estimator 465
The particle filter 468
Regularized particle filter resampling 473
The extended Kalman particle filter 478

INTRODUCTION

This book discusses mathematical approaches to the best possible way of estimating the state of a general system. Although the book is firmly grounded in mathematical theory, it should not be considered a mathematics text. It is more of an engineering text, or perhaps an *applied* mathematics text. The approaches that we present for state estimation are all given with the goal of eventual implementation in software.[1] The goal of this text is to present state estimation theory in the most clear yet rigorous way possible, while providing enough advanced material and references so that the reader is prepared to contribute new material to the state of the art. Engineers are usually concerned with eventual implementation, and so the material presented is geared toward discrete-time systems. However, continuous-time systems are also discussed for the sake of completeness, and because there is still room for implementations of continuous-time filters.

Before we discuss optimal state estimation, we need to define what we mean by the term *state*. The states of a system are those variables that provide a complete representation of the internal condition or status of the system at a given instant of time.[2] This is far from a rigorous definition, but it suffices for the purposes of

[1] I use the practice that is common in academia of referring to a generic third person by the word *we*. Sometimes, I use the word *we* to refer to the reader and myself. Other times, I use the word *we* to indicate that I am speaking on behalf of the control and estimation community. The distinction should be clear from the context. However, I encourage the reader not to read too much into my use of the word *we*; it is more a matter of personal preference and style rather than a claim to authority.

[2] In this book, we use the terms *state* and *state variable* interchangably. Also, the word *state* could refer to the entire collection of state variables, or it could refer to a single state variable. The specific meaning needs to be inferred from the context.

this introduction. For example, the states of a motor might include the currents through the windings, and the position and speed of the motor shaft. The states of an orbiting satellite might include its position, velocity, and angular orientation. The states of an economic system might include per-capita income, tax rates, unemployment, and economic growth. The states of a biological system might include blood sugar levels, heart and respiration rates, and body temperature.

State estimation is applicable to virtually all areas of engineering and science. Any discipline that is concerned with the mathematical modeling of its systems is a likely (perhaps inevitable) candidate for state estimation. This includes electrical engineering, mechanical engineering, chemical engineering, aerospace engineering, robotics, economics, ecology, biology, and many others. The possible applications of state estimation theory are limited only by the engineer's imagination, which is why state estimation has become such a widely researched and applied discipline in the past few decades. State-space theory and state estimation was initially developed in the 1950s and 1960s, and since then there have been a huge number of applications. A few applications are documented in [Sor85]. Thousands of other applications can be discovered by doing an Internet search on the terms "state estimation" and "application," or "Kalman filter" and "application."

State estimation is interesting to engineers for at least two reasons:

- Often, an engineer needs to estimate the system states in order to implement a state-feedback controller. For example, the electrical engineer needs to estimate the winding currents of a motor in order to control its position. The aerospace engineer needs to estimate the attitude of a satellite in order to control its velocity. The economist needs to estimate economic growth in order to try to control unemployment. The medical doctor needs to estimate blood sugar levels in order to control heart and respiration rates.

- Often an engineer needs to estimate the system states because those states are interesting in their own right. For example, if an engineer wants to measure the health of an engineering system, it may be necessary to estimate the internal condition of the system using a state estimation algorithm. An engineer might want to estimate satellite position in order to more intelligently schedule future satellite activities. An economist might want to estimate economic growth in order to make a political point. A medical doctor might want to estimate blood sugar levels in order to evaluate the health of a patient.

There are many other fine books on state estimation that are available (see Appendix B). This begs the question: Why yet another textbook on the topic of state estimation? The reason that this present book has been written is to offer a pedagogical approach and perspective that is not available in other state estimation books. In particular, the hope is that this book will offer the following:

- A straightforward, bottom-up approach that assists the reader in obtaining a clear (but theoretically rigorous) understanding of state estimation. This is reminiscent of Gelb's approach [Gel74], which has proven effective for many state estimation students of the past few decades. However, many aspects of Gelb's book have become outdated. In addition, many of the more recent books on state estimation read more like research monographs and are not entirely accessible to the average engineering student. Hence the need for the present book.

- Simple examples that provide the reader with an intuitive understanding of the theory. Many books present state estimation theory and then follow with examples or problems that require a computer for implementation. However, it is possible to present simple examples and problems that require only paper and pencil to solve. These simple problems allow the student to more directly see how the theory works itself out in practice. Again, this is reminiscent of Gelb's approach [Gel74].

- MATLAB-based source code[3] for the examples in the book is available at the author's Web site.[4] A number of other texts supply source code, but it is often on disk or CD, which makes the code subject to obsolescence. The author's e-mail address is also available on the Web site, and I enthusiastically welcome feedback, comments, suggestions for improvements, and corrections. Of course, Web addresses are also subject to obsolescence, but the book also contains algorithmic, high-level pseudocode listings that will last longer than any specific software listings.

- Careful treatment of advanced topics in optimal state estimation. These topics include unscented filtering, high-order nonlinear filtering, particle filtering, constrained state estimation, reduced-order filtering, robust Kalman filtering, and mixed Kalman/H_∞ filtering. Some of these topics are mature, having been introduced in the 1960s, but others of these topics are recent additions to the state of the art. This coverage is not matched in any other books on the topic of state estimation.

Some of the other books on state estimation offer some of the above features, but no other books offer *all* of these features.

Prerequisites

The prerequisites for understanding the material in this book are a good foundation in linear systems theory and probability and stochastic processes. Ideally, the reader will already have taken a graduate course in both of these topics. However, it should be said that a background in linear systems theory is more important than probability. The first two chapters of the book review the elements of linear systems and probability that are essential for the rest of the book, and also serve to establish the notation that is used during the remainder of the book.

Other material could also be considered prerequisite to understanding this book, such as undergraduate advanced calculus, control theory, and signal processing. However, it would be more accurate to say that the reader will require a moderately high level of mathematical and engineering maturity, rather than trying to identify a list of required prerequisite courses.

[3]MATLAB is a registered trademark of The MathWorks, Inc.
[4]http://academic.csuohio.edu/simond/estimation – if the Web site address changes, it should be easy to find with an internet search.

Problems

The problems at the end of each chapter have been written to give a high degree of flexibility to the instructor and student. The problems include both written exercises and computer exercises. The written exercises are intended to strengthen the student's grasp of the theory, and deepen the student's intuitive understanding of the concepts. The computer exercises are intended to help the student learn how to apply the theory to problems of the type that might be encountered in industrial or government projects. Both types of problems are important for the student to become proficient at the material. The distinction between written exercises and computer exercises is more of a fuzzy division rather than a strict division. That is, some of the written exercises include parts for which some computer work might be useful (even required), and some of the computer exercises include parts for which some written analysis might be useful (even required).

A solution manual to all of the problems in the text (both written exercises and computer exercises) is available from the publisher to instructors who have adopted this book. Course instructors are encouraged to contact the publisher for further information about out how to obtain the solution manual.

Outline of the book

This book is divided into four parts. The first part of the book covers introductory material. Chapter 1 is a review of the relevant areas of linear systems. This material is often covered in a first-semester graduate course taken by engineering students. It is advisable, although not strictly required, that readers of this book have already taken a graduate linear systems course. Chapter 2 reviews probability theory and stochastic processes. Again, this is often covered in a first-semester graduate course. In this book we rely less on probability theory than linear systems theory, so a previous course in probability and stochastic processes is not required for the material in this book (although it would be helpful). Chapter 3 covers least squares estimation of constants and Wiener filtering of stochastic processes. The section on Wiener filtering is not required for the remainder of the book, although it is interesting both in its own right and for historical perspective. Chapter 4 is a brief discussion of how the statistical measures of a state (mean and covariance) propagate in time. Chapter 4 provides a bridge from the first three chapters to the second part of the book.

The second part of the book covers Kalman filtering, which is the workhorse of state estimation. In Chapter 5, we derive the discrete-time Kalman filter, including several different (but mathematically equivalent) formulations. In Chapter 6, we present some alternative Kalman filter formulations, including sequential filtering, information filtering, square root filtering, and U-D filtering. In Chapter 7, we discuss some generalizations of the Kalman filter that make the filter applicable to a wider class of problems. These generalizations include correlated process and measurement noise, colored process and measurement noise, steady-state filtering for computational savings, fading-memory filtering, and constrained Kalman filtering. In Chapter 8, we present the continuous-time Kalman filter. This chapter could be skipped if time is short since the continuous-time filter is rarely implemented in practice. In Chapter 9, we discuss optimal smoothing, which is a way to estimate

the state of a system at time τ based on measurements that extend beyond time τ. As part of the derivation of the smoothing equations, the first section of Chapter 9 presents another alternative form for the Kalman filter. Chapter 10 presents some additional, more advanced topics in Kalman filtering. These topics include verification of filter performance, estimation in the case of unknown system models, reduced-order filtering, increasing the robustness of the Kalman filter, and filtering in the presence of measurement synchronization errors. This chapter should provide fertile ground for students or engineers who are looking for research topics or projects.

The third part of the book covers H_∞ filtering. This area is not as mature as Kalman filtering and so there is less material than in the Kalman filtering part of the book. Chapter 11 introduces yet another alternate Kalman filter form as part of the H_∞ filter derivation. This chapter discusses both time domain and frequency domain approaches to H_∞ filtering. Chapter 12 discusses advanced topics in H_∞ filtering, including mixed Kalman/H_∞ filtering and constrained H_∞ filtering. There is a lot of room for further development in H_∞ filtering, and this part of the book could provide a springboard for researchers to make contributions in this area.

The fourth part of the book covers filtering for nonlinear systems. Chapter 13 discusses nonlinear filtering based on the Kalman filter, which includes the widely used extended Kalman filter. Chapter 14 covers the unscented Kalman filter, which is a relatively recent development that provides improved performance over the extended Kalman filter. Chapter 15 discusses the particle filter, another recent development that provides a very general solution to the nonlinear filtering problem. It is hoped that this part of the book, especially Chapters 14 and 15, will inspire researchers to make further contributions to these new areas of study.

The book concludes with three brief appendices. Appendix A gives some historical perspectives on the development of the Kalman filter, starting with the least squares work of Roger Cotes in the early 1700s, and concluding with the space program applications of Kalman filtering in the 1960s. Appendix B discusses the many other books that have been written on Kalman filtering, including their distinctive contributions. Finally, Appendix C presents some speculations on the connections between optimal state estimation and the meaning of life.

Figure I.1 gives a graphical representation of the structure of the book from a prerequisite point of view. For example, Chapter 3 builds on Chapters 1 and 2. Chapter 4 builds on Chapter 3, and Chapter 5 builds on Chapter 4. Chapters 6–11 each depend on material from Chapter 5, but are independent from each other. Chapter 12 builds on Chapter 11. Chapter 13 depends on Chapter 8, and Chapter 14 depends on Chapter 13. Finally, Chapter 15 builds on Chapter 3. This structure can be used to customize a course based on this book.

A note on notation

Three dots between delimiters (parenthesis, brackets, or braces) means that the quantity between the delimiters is the same as the quantity between the previous set of identical delimiters in the same equation. For example,

$$\begin{aligned}(A+BCD)+(\cdots)^T &= (A+BCD)+(A+BCD)^T \\ A+\left[B(C+D)\right]^{-1}E\left[\cdots\right] &= A+\left[B(C+D)\right]^{-1}E\left[B(C+D)\right] \qquad \text{(I.1)}\end{aligned}$$

Figure I.1 Prerequisite structure of the chapters in this book.

PART I

INTRODUCTORY MATERIAL

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

CHAPTER 1

Linear systems theory

> Finally, we make some remarks on why *linear* systems are so important. The answer is simple: because we can solve them!
>
> —Richard Feynman [Fey63, p. 25-4]

This chapter reviews some essentials of linear systems theory. This material is typically covered in a linear systems course, which is a first-semester graduate level course in electrical engineering. The theory of optimal state estimation heavily relies on matrix theory, including matrix calculus, so matrix theory is reviewed in Section 1.1. Optimal state estimation can be applied to both linear and nonlinear systems, although state estimation is much more straightforward for linear systems. Linear systems are briefly reviewed in Section 1.2 and nonlinear systems are discussed in Section 1.3. State-space systems can be represented in the continuous-time domain or the discrete-time domain. Physical systems are typically described in continuous time, but control and state estimation algorithms are typically implemented on digital computers. Section 1.4 discusses some standard methods for obtaining a discrete-time representation of a continuous-time system. Section 1.5 discusses how to simulate continuous-time systems on a digital computer. Sections 1.6 and 1.7 discuss the standard concepts of stability, controllability, and observability of linear systems. These concepts are necessary to understand some of the optimal state estimation material later in the book. Students with a strong

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

background in linear systems theory can skip the material in this chapter. However, it would still help to at least review this chapter to solidify the foundational concepts of state estimation before moving on to the later chapters of this book.

1.1 MATRIX ALGEBRA AND MATRIX CALCULUS

In this section, we review matrices, matrix algebra, and matrix calculus. This is necessary in order to understand the rest of the book because optimal state estimation algorithms are usually formulated with matrices.

A scalar is a single quantity. For example, the number 2 is a scalar. The number $1 + 3j$ is a scalar (we use j in this book to denote the square root of -1). The number π is a scalar.

A vector consists of scalars that are arranged in a row or column. For example, the vector

$$\begin{bmatrix} 1 & 3 & \pi \end{bmatrix} \tag{1.1}$$

is a 3-element vector. This vector is a called a 1×3 vector because it has 1 row and 3 columns. This vector is also called a row vector because it is arranged as a single row. The vector

$$\begin{bmatrix} -2 \\ \pi^2 \\ j \\ 0 \end{bmatrix} \tag{1.2}$$

is a 4-element vector. This vector is a called a 4×1 vector because it has 4 rows and 1 column. This vector is also called a column vector because it is arranged as a single column. Note that a scalar can be viewed as a 1-element vector; a scalar is a degenerate vector. (This is just like a plane can be viewed as a 3-dimensional shape; a plane is a degenerate 3-dimensional shape.)

A matrix consists of scalars that are arranged in a rectangle. For example, the matrix

$$\begin{bmatrix} -2 & 3 \\ 0 & \pi^2 \\ j & 0 \end{bmatrix} \tag{1.3}$$

is a 3×2 matrix because it has 3 rows and 2 columns. The number of rows and columns in a matrix can be collectively referred to as the dimension of the matrix. For example, the dimension of the matrix in the preceding equation is 3×2. Note that a vector can be viewed as a degenerate matrix. For example, Equation (1.1) is a 1×3 matrix. A scalar can also be viewed as a degenerate matrix. For example, the scalar 6 is a 1×1 matrix.

The rank of a matrix is defined as the number of linearly independent rows. This is also equal to the number of linearly independent columns. The rank of a matrix A is often indicated with the notation $\rho(A)$. The rank of a matrix is always less than or equal to the number of rows, and it is also less than or equal to the number of columns. For example, the matrix

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \tag{1.4}$$

has a rank of one because it has only one linearly independent row; the two rows are multiples of each other. It also has only one linearly independent column; the two columns are multiples of each other. On the other hand, the matrix

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \tag{1.5}$$

has a rank of two because it has two linearly independent rows. That is, there are no nonzero scalars c_1 and c_2 such that

$$c_1 \begin{bmatrix} 1 & 3 \end{bmatrix} + c_2 \begin{bmatrix} 2 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix} \tag{1.6}$$

so the two rows are linearly independent. It also has two linearly independent columns. That is, there are no nonzero scalars c_1 and c_2 such that

$$c_1 \begin{bmatrix} 1 \\ 2 \end{bmatrix} + c_2 \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{1.7}$$

so the two columns are linearly independent. A matrix whose elements are comprised entirely of zeros has a rank of zero. An $n \times m$ matrix whose rank is equal to $\min(n,m)$ is called full rank. The nullity of an $n \times m$ matrix A is equal to $[m - \rho(A)]$.

The transpose of a matrix (or vector) can be taken by changing all the rows to columns, and all the columns to rows. The transpose of a matrix is indicated with a T superscript, as in A^T.[1] For example, if A is the $r \times n$ matrix

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & & \vdots \\ A_{r1} & \cdots & A_{rn} \end{bmatrix} \tag{1.8}$$

then A^T is the $n \times r$ matrix

$$A^T = \begin{bmatrix} A_{11} & \cdots & A_{r1} \\ \vdots & & \vdots \\ A_{1n} & \cdots & A_{rn} \end{bmatrix} \tag{1.9}$$

Note that we use the notation A_{ij} to indicate the scalar in the ith row and jth column of the matrix A. A symmetric matrix is one for which $A = A^T$.

The hermitian transpose of a matrix (or vector) is the complex conjugate of the transpose, and is indicated with an H superscript, as in A^H. For example, if

$$A = \begin{bmatrix} 1 & 2j & 3-j \\ 4j & 5+j & 1-3j \end{bmatrix} \tag{1.10}$$

then

$$A = \begin{bmatrix} 1 & -4j \\ -2j & 5-j \\ 3+j & \\ 1+3j & \end{bmatrix} \tag{1.11}$$

A hermitian matrix is one for which $A = A^H$.

[1]Many papers or books indicate transpose with a prime, as in A', or with a lower case t, as in A^t.

1.1.1 Matrix algebra

Matrix addition and subtraction is simply defined as element-by-element addition and subtraction. For example,

$$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 4 & 1 \\ 1 & -1 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 6 & 4 \\ 4 & 1 & -1 \end{bmatrix} \tag{1.12}$$

The sum $(A + B)$ and the difference $(A - B)$ is defined only if the dimension of A is equal to the dimension of B.

Suppose that A is an $n \times r$ matrix and B is an $r \times p$ matrix. Then the product of A and B is written as $C = AB$. Each element in the matrix product C is computed as

$$C_{ij} = \sum_{k=1}^{r} A_{ik} B_{kj} \qquad i = 1, \ldots, n \qquad j = 1, \ldots, p \tag{1.13}$$

The matrix product AB is defined only if the number of columns in A is equal to the number of rows in B. It is important to note that matrix multiplication does not commute. In general, $AB \neq BA$.

Suppose we have an $n \times 1$ vector x. We can compute the 1×1 product $x^T x$, and the $n \times n$ product xx^T as follows:

$$\begin{aligned} x^T x &= \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \\ &= x_1^2 + \cdots + x_n^2 \\ xx^T &= \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix} \\ &= \begin{bmatrix} x_1^2 & \cdots & x_1 x_n \\ \vdots & \ddots & \vdots \\ x_n x_1 & \cdots & x_n^2 \end{bmatrix} \end{aligned} \tag{1.14}$$

Suppose that we have a $p \times n$ matrix H and an $n \times n$ matrix P. Then H^T is a $n \times p$ matrix, and we can compute the $p \times p$ matrix product HPH^T.

$$\begin{aligned} HPH^T &= \begin{bmatrix} H_{11} & \cdots & H_{1n} \\ \vdots & \ddots & \vdots \\ H_{p1} & \cdots & H_{pn} \end{bmatrix} \begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix} \begin{bmatrix} H_{11} & \cdots & H_{p1} \\ \vdots & \ddots & \vdots \\ H_{1n} & \cdots & H_{pn} \end{bmatrix} \\ &= \begin{bmatrix} \sum_{j,k} H_{1j} P_{jk} H_{1k} & \cdots & \sum_{j,k} H_{1j} P_{jk} H_{pk} \\ \vdots & \ddots & \vdots \\ \sum_{j,k} H_{pj} P_{jk} H_{1k} & \cdots & \sum_{j,k} H_{pj} P_{jk} H_{pk} \end{bmatrix} \end{aligned} \tag{1.15}$$

This matrix of sums can be written as the following sum of matrices:

$$
\begin{aligned}
HPH^T &= \begin{bmatrix} H_{11}P_{11}H_{11} & \cdots & H_{11}P_{11}H_{p1} \\ \vdots & \ddots & \vdots \\ H_{p1}P_{11}H_{11} & \cdots & H_{p1}P_{11}H_{p1} \end{bmatrix} + \cdots + \\
&\quad \begin{bmatrix} H_{1n}P_{nn}H_{1n} & \cdots & H_{1n}P_{nn}H_{pn} \\ \vdots & \ddots & \vdots \\ H_{pn}P_{nn}H_{1n} & \cdots & H_{pn}P_{nn}H_{pn} \end{bmatrix} \\
&= H_1P_{11}H_1^T + \cdots + H_nP_{nn}H_n^T \\
&= \sum_{j,k} H_jP_{jk}H_k^T
\end{aligned} \tag{1.16}
$$

where we have used the notation that H_k is the kth column of H.

Matrix division is not defined; we cannot divide a matrix by another matrix (unless, of course, the denominator matrix is a scalar).

An identity matrix I is defined as a square matrix with ones on the diagonal and zeros everywhere else. For example, the 3×3 identity matrix is equal to

$$
I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1.17}
$$

The identity matrix has the property that $AI = A$ for any matrix A, and $IA = A$ (as long the dimensions of the identity matrices are compatible with those of A). The 1×1 identity matrix is equal to the scalar 1.

The determinant of a matrix is defined inductively for square matrices. The determinant of a scalar (i.e., a 1×1 matrix) is equal to the scalar. Now consider an $n \times n$ matrix A. Use the notation $A^{(i,j)}$ to denote the matrix that is formed by deleting the ith row and jth column of A. The determinant of A is defined as

$$
|A| = \sum_{j=1}^{n} (-1)^{i+j} A_{ij} |A^{(i,j)}| \tag{1.18}
$$

for any value of $i \in [1, n]$. This is called the Laplace expansion of A along its ith row. We see that the determinant of the $n \times n$ matrix A is defined in terms of the determinants of $(n-1) \times (n-1)$ matrices. Similarly, the determinants of $(n-1) \times (n-1)$ matrices are defined in terms of the determinants of $(n-2) \times (n-2)$ matrices. This continues until the determinants of 2×2 matrices are defined in terms of the determinants of 1×1 matrices, which are scalars. The determinant of A can also be defined as

$$
|A| = \sum_{i=1}^{n} (-1)^{i+j} A_{ij} |A^{(i,j)}| \tag{1.19}
$$

for any value of $j \in [1, n]$. This is called the Laplace expansion of A along its jth column. Interestingly, Equation (1.18) (for any value of i) and Equation (1.19) (for any value of j) both give identical results. From the definition of the determinant

we see that

$$\begin{aligned}
\det[A_{11}] &= A_{11} \\
\det\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} &= A_{11}A_{22} - A_{12}A_{21} \\
\det\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} &= A_{11}(A_{22}A_{33} - A_{23}A_{32}) - \\
&\quad A_{12}(A_{21}A_{33} - A_{23}A_{31}) + \\
&\quad A_{13}(A_{21}A_{32} - A_{22}A_{31})
\end{aligned} \tag{1.20}$$

Some interesting properties of determinants are

$$|AB| = |A||B| \tag{1.21}$$

assuming that A and B are square and have the same dimensions. Also,

$$|A| = \prod_{i=1}^{n} \lambda_i \tag{1.22}$$

where λ_i (the eigenvalues of A) are defined below.

The inverse of a matrix A is defined as the matrix A^{-1} such that $AA^{-1} = A^{-1}A = I$. A matrix cannot have an inverse unless it is square. Some square matrices do not have an inverse. A square matrix that does not have an inverse is called singular or invertible. In the scalar case, the only number that does not have an inverse is the number 0. But in the matrix case, there are many matrices that are singular. A matrix that does have an inverse is called nonsingular or invertible. For example, notice that

$$\begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ -2/3 & 1/3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{1.23}$$

Therefore, the two matrices on the left side of the equation are inverses of each other. The nonsingularity of an $n \times n$ matrix A can be stated in many equivalent ways, some of which are the following [Hor85]:

- A is nonsingular.
- A^{-1} exists.
- The rank of A is equal to n.
- The rows of A are linearly independent.
- The columns of A are linearly independent.
- $|A| \neq 0$.
- $Ax = b$ has a unique solution x for all b.
- 0 is not an eigenvalue of A.

The trace of a square matrix is defined as the sum of its diagonal elements:

$$\mathrm{Tr}(A) = \sum_i A_{ii} \tag{1.24}$$

The trace of a matrix is defined only if the matrix is square. The trace of a 1×1 matrix is equal to the trace of a scalar, which is equal to the value of the scalar. One interesting property of the trace of a square matrix is

$$\mathrm{Tr}(A) = \sum_i \lambda_i \tag{1.25}$$

That is, the trace of a square matrix is equal to the sum of its eigenvalues.

Some interesting and useful characteristics of matrix products are the following:

$$\begin{aligned} (AB)^T &= B^T A^T \\ (AB)^{-1} &= B^{-1} A^{-1} \\ \mathrm{Tr}(AB) &= \mathrm{Tr}(BA) \end{aligned} \tag{1.26}$$

This assumes that the inverses exist for the inverse equation, and that the matrix dimensions are compatible so that matrix multiplication is defined. The transpose of a matrix product is equal to the product of the transposes in the opposite order. The inverse of a matrix product is equal to the product of the inverses in the opposite order. The trace of a matrix product is independent of the order in which the matrices are multiplied.

The two-norm of a column vector of real numbers, also called the Euclidean norm, is defined as follows:

$$\begin{aligned} ||x||_2 &= \sqrt{x^T x} \\ &= \sqrt{x_1^2 + \cdots + x_n^2} \end{aligned} \tag{1.27}$$

From (1.14) we see that

$$xx^T = \begin{bmatrix} x_1^2 & \cdots & x_1 x_n \\ \vdots & \ddots & \vdots \\ x_n x_1 & \cdots & x_n^2 \end{bmatrix} \tag{1.28}$$

Taking the trace of this matrix is

$$\begin{aligned} \mathrm{Tr}(xx^T) &= x_1^2 + \cdots + x_n^2 \\ &= ||x||_2^2 \end{aligned} \tag{1.29}$$

An $n \times n$ matrix A has n eigenvalues and n eigenvectors. The scalar λ is an eigenvalue of A, and the $n \times 1$ vector x is an eigenvector of A, if the following equation holds:

$$Ax = \lambda x \tag{1.30}$$

The eigenvalues and eigenvectors of a matrix are collectively referred to as the eigendata of the matrix.[2] An $n \times n$ matrix has exactly n eigenvalues, although

[2]Eigendata have also been referred to by many other terms over the years, including characteristic roots, latent roots and vectors, and proper numbers and vectors [Fad59].

some may be repeated. This is like saying that an nth order polynomial equation has exactly n roots, although some may be repeated. From the above definitions of eigenvalues and eigenvectors we can see that

$$\begin{aligned} Ax &= \lambda x \\ A^2x &= A\lambda x \\ &= \lambda(Ax) \\ &= \lambda(\lambda x) \\ &= \lambda^2 x \end{aligned} \tag{1.31}$$

So if A has eigendata (λ, x), then A^2 has eigendata (λ^2, x). It can be shown that A^{-1} exists if and only if none of the eigenvalues of A are equal to 0. If A is symmetric then all of its eigenvalues are real numbers.

A symmetric $n \times n$ matrix A can be characterized as either positive definite, positive semidefinite, negative definite, negative semidefinite, or indefinite. Matrix A is:

- *Positive definite* if $x^T Ax > 0$ for all nonzero $n \times 1$ vectors x. This is equivalent to saying that all of the eigenvalues of A are positive real numbers. If A is positive definite, then A^{-1} is also positive definite.

- *Positive semidefinite* if $x^T Ax \geq 0$ for all $n \times 1$ vectors x. This is equivalent to saying that all of the eigenvalues of A are nonnegative real numbers. Positive semidefinite matrices are sometimes called nonnegative definite.

- *Negative definite* if $x^T Ax < 0$ for all nonzero $n \times 1$ vectors x. This is equivalent to saying that all of the eigenvalues of A are negative real numbers. If A is negative definite, then A^{-1} is also negative definite.

- *Negative semidefinite* if $x^T Ax \leq 0$ for all $n \times 1$ vectors x. This is equivalent to saying that all of the eigenvalues of A are nonpositive real numbers. Negative semidefinite matrices are sometimes called nonpositive definite.

- *Indefinite* if it does not fit into any of the above four categories. This is equivalent to saying that some of its eigenvalues are positive and some are negative.

Some books generalize the idea of positive definiteness and negative definiteness to include nonsymmetric matrices.

The weighted two-norm of an $n \times 1$ vector x is defined as

$$||x||_Q^2 = \sqrt{x^T Qx} \tag{1.32}$$

where Q is required to be an $n \times n$ positive definite matrix. The above norm is also called the Q-weighted two-norm of x. A quantity of the form $x^T Qx$ is called a quadratic in analogy to a quadratic term in a scalar equation.

The singular values σ of a matrix A are defined as

$$\begin{aligned} \sigma^2(A) &= \lambda(A^T A) \\ &= \lambda(AA^T) \end{aligned} \tag{1.33}$$

If A is an $n \times m$ matrix, then it has $\min(n, m)$ singular values. AA^T will have n eigenvalues, and A^TA will have m eigenvalues. If $n > m$ then AA^T will have the same eigenvalues as A^TA plus an additional $(n - m)$ zeros. These additional zeros are not considered to be singular values of A, because A always has $\min(n, m)$ singular values. This knowledge can help reduce effort during the computation of singular values. For example, if A is a 13×3 matrix, then it is much easier to compute the eigenvalues of the 3×3 matrix A^TA rather than the 13×13 matrix AA^T. Either computation will result in the same three singular values.

1.1.2 The matrix inversion lemma

In this section, we will derive the matrix inversion lemma, which is a tool that we will use many times in this book. It is also a tool that is frequently useful in other areas of control, estimation theory, and signal processing.

Suppose we have the partitioned matrix $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ where A and D are invertible square matrices, and the B and C matrices may or may not be square. We define E and F matrices as follows:

$$\begin{aligned} E &= D - CA^{-1}B \\ F &= A - BD^{-1}C \end{aligned} \tag{1.34}$$

Assume that E is invertible. Then we can show that

$$\begin{aligned} &\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} A^{-1} + A^{-1}BE^{-1}CA^{-1} & -A^{-1}BE^{-1} \\ -E^{-1}CA^{-1} & E^{-1} \end{bmatrix} \\ &= \begin{bmatrix} I + BE^{-1}CA^{-1} - BE^{-1}CA^{-1} & -BE^{-1} + BE^{-1} \\ CA^{-1} + CA^{-1}BE^{-1}CA^{-1} - DE^{-1}CA^{-1} & -CA^{-1}BE^{-1} + DE^{-1} \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ CA^{-1} - (D - CA^{-1}B)E^{-1}CA^{-1} & (D - CA^{-1}B)E^{-1} \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \end{aligned} \tag{1.35}$$

Now assume that F is invertible. Then we can show that

$$\begin{aligned} &\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} F^{-1} & -A^{-1}BE^{-1} \\ -D^{-1}CF^{-1} & E^{-1} \end{bmatrix} \\ &= \begin{bmatrix} AF^{-1} - BD^{-1}CF^{-1} & -BE^{-1} + BE^{-1} \\ CF^{-1} - CF^{-1} & -CA^{-1}BE^{-1} + DE^{-1} \end{bmatrix} \\ &= \begin{bmatrix} (A - BD^{-1}C)F^{-1} & 0 \\ 0 & (D - CA^{-1}B)E^{-1} \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \end{aligned} \tag{1.36}$$

Equations (1.35) and (1.36) are two expressions for the inverse of $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$. Since these two expressions are inverses of the same matrix, they must be equal. We therefore conclude that the upper-left partitions of the matrices are equal, which gives

$$F^{-1} = A^{-1} + A^{-1}BE^{-1}CA^{-1} \tag{1.37}$$

Now we can use the definition of F to obtain

$$(A - BD^{-1}C)^{-1} = A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} \tag{1.38}$$

This is called the matrix inversion lemma. It is also referred to by other terms, such as the Sherman–Morrison formula, Woodbury's identity, and the modified matrices formula. One of its earliest presentations was in 1944 by William Duncan [Dun44], and similar identities were developed by Alston Householder [Hou53]. An account of its origins and variations (e.g., singular A) is given in [Hen81]. The matrix inversion lemma is often stated in slightly different but equivalent ways. For example,

$$(A + BD^{-1}C)^{-1} = A^{-1} - A^{-1}B(D + CA^{-1}B)^{-1}CA^{-1} \tag{1.39}$$

The matrix inversion lemma can sometimes be used to reduce the computational effort of matrix inversion. For instance, suppose that A is $n \times n$, B is $n \times p$, C is $p \times n$, D is $p \times p$, and $p < n$. Suppose further that we already know A^{-1}, and we want to add some quantity to A and then compute the new inverse. A straightforward computation of the new inverse would be an $n \times n$ inversion. But if the new matrix to invert can be written in the form of the left side of Equation (1.39), then we can use the right side of Equation (1.39) to compute the new inverse, and the right side of Equation (1.39) requires a $p \times p$ inversion instead of an $n \times n$ inversion (since we already know the inverse of the old A matrix).

■ EXAMPLE 1.1

At your investment firm, you notice that in January the New York Stock Exchange index decreased by 2%, the American Stock Exchange index increased by 1%, and the NASDAQ stock exchange index increased by 2%. As a result, investors increased their deposits by 1%. The next month, the stock exchange indices changed by −4%, 3%, and 2%, respectively, and investor deposits increased by 2%. The following month, the stock exchange indices changed by −5%, 1%, and 5%, respectively, and investor deposits increased by 2%. You suspect that investment changes y can be modeled as $y = g_1x_1 + g_2x_2 + g_3x_3$, where the x_i variables are the stock exchange index changes, and the g_i are unknown constants. In order to determine the g_i constants you need to invert the matrix

$$A = \begin{bmatrix} -2 & 1 & 2 \\ -4 & 3 & 2 \\ -5 & 1 & 5 \end{bmatrix} \tag{1.40}$$

The result is

$$\begin{aligned} A^{-1} &= \frac{1}{6}\begin{bmatrix} 13 & -3 & -4 \\ 10 & 0 & -4 \\ 11 & -3 & -2 \end{bmatrix} \\ g &= A^{-1}\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} \\ &= \frac{1}{6}\begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} \end{aligned} \tag{1.41}$$

This allows you to use stock exchange index changes to predict investment changes in the following month, which allows you to better schedule personnel and computer resources. However, soon afterward you find out that the NASDAQ change in the third month was actually 6% rather than 5%. This means that in order to find the g_i constants you need to invert the matrix

$$A' = \begin{bmatrix} -2 & 1 & 2 \\ -4 & 3 & 2 \\ -5 & 1 & 6 \end{bmatrix} \tag{1.42}$$

You are tired of inverting matrices and so you wonder if you can somehow use the inverse of A (which you have already calculated) to find the inverse of A'. Remembering the matrix inversion lemma, you realize that $A' = A + BD^{-1}C$, where

$$\begin{aligned} B &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \\ C &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \\ D &= 1 \end{aligned} \tag{1.43}$$

You therefore use the matrix inversion lemma to compute

$$\begin{aligned} (A')^{-1} &= (A + BD^{-1}C)^{-1} \\ &= A^{-1} - A^{-1}B(D + CA^{-1}B)^{-1}CA^{-1} \end{aligned} \tag{1.44}$$

The $(D + CA^{-1}B)$ term that needs to be inverted in the above equation is a scalar, so its inversion is simple. This gives

$$\begin{aligned} (A')^{-1} &= \begin{bmatrix} 4.00 & 1.00 & -1.00 \\ 3.50 & -0.50 & -1.00 \\ 2.75 & -0.75 & -0.50 \end{bmatrix} \\ g &= (A')^{-1} \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0.5 \\ 0.25 \end{bmatrix} \end{aligned} \tag{1.45}$$

In this example, the use of the matrix inversion lemma is not really necessary because A' (the new matrix to invert) is only 3×3. However, with larger matrices, such as 1000×1000 matrices, the computational savings that is realized by using the matrix inversion lemma could be significant.

$\nabla\nabla\nabla$

Now suppose that A, B, C, and D are matrices, with A and D being square. Then it can be seen that

$$\begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I & -A^{-1}B \\ 0 & I \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & D - CA^{-1}B \end{bmatrix} \tag{1.46}$$

This means that

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = |A||D - CA^{-1}B| \tag{1.47}$$

Similarly, it can be shown that

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = |D||A - BD^{-1}C| \tag{1.48}$$

These formulas are called product rules for determinants. They were first given by the Russian-born mathematician Issai Schur in a German paper [Sch17] that was reprinted in English in [Sch86].

1.1.3 Matrix calculus

In our first calculus course, we learned the mathematics of derivatives and integrals and how to apply those concepts to scalars. We can also apply the mathematics of calculus to vectors and matrices. Some aspects of matrix calculus are identical to scalar calculus, but some scalar calculus concepts need to be extended in order to derive formulas for matrix calculus.

As intuition would lead us to believe, the time derivative of a matrix is simply equal to the matrix of the time derivatives of the individual matrix elements. Also, the integral of a matrix is equal to the matrix of the integrals of the individual matrix elements. In other words, assuming that A is an $m \times n$ matrix, we have

$$\begin{aligned} \dot{A}(t) &= \begin{bmatrix} \dot{A}_{11}(t) & \cdots & \dot{A}_{1n}(t) \\ \vdots & \ddots & \vdots \\ \dot{A}_{n1}(t) & \cdots & \dot{A}_{nn}(t) \end{bmatrix} \\ \int A(t)\,dt &= \begin{bmatrix} \int A_{11}(t)\,dt & \cdots & \int A_{1n}(t)\,dt \\ \vdots & \ddots & \vdots \\ \int A_{n1}(t)\,dt & \cdots & \int A_{nn}(t)\,dt \end{bmatrix} \end{aligned} \tag{1.49}$$

Next we will compute the time derivative of the inverse of a matrix. Suppose that matrix $A(t)$, which we will denote as A, has elements that are functions of time. We know that $AA^{-1} = I$; that is, AA^{-1} is a constant matrix and therefore has a time derivative of zero. But the time derivative of AA^{-1} can be computed as

$$\frac{d}{dt}(AA^{-1}) = \dot{A}A^{-1} + A\frac{d}{dt}(A^{-1}) \tag{1.50}$$

Since this is zero, we can solve for $d(A^{-1})/dt$ as

$$\frac{d}{dt}(A^{-1}) = -A^{-1}\dot{A}A^{-1} \tag{1.51}$$

Note that for the special case of a scalar A, this reduces to the familiar equation

$$\begin{aligned} \frac{d}{dt}(1/A) &= \frac{\partial(1/A)}{\partial A}\frac{dA}{dt} \\ &= -\dot{A}/A^2 \end{aligned} \tag{1.52}$$

Now suppose that x is an $n \times 1$ vector and $f(x)$ is a scalar function of the elements of x. Then

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \partial f/\partial x_1 & \cdots & \partial f/\partial x_n \end{bmatrix} \tag{1.53}$$

Even though x is a column vector, $\partial f/\partial x$ is a row vector. The converse is also true – if x is a row vector, then $\partial f/\partial x$ is a column vector. Note that some authors define this the other way around. That is, they say that if x is a column vector then $\partial f/\partial x$ is also a column vector. There is no accepted convention for the definition of the partial derivative of a scalar with respect to a vector. It does not really matter which definition we use as long as we are consistent. In this book, we will use the convention described by Equation (1.53).

Now suppose that A is an $m \times n$ matrix and $f(A)$ is a scalar. Then the partial derivative of a scalar with respect to a matrix can be computed as follows:

$$\frac{\partial f}{\partial A} = \begin{bmatrix} \partial f/\partial A_{11} & \cdots & \partial f/\partial A_{1n} \\ \vdots & \ddots & \vdots \\ \partial f/\partial A_{m1} & \cdots & \partial f/\partial A_{mn} \end{bmatrix} \tag{1.54}$$

With these definitions we can compute the partial derivative of the dot product of two vectors. Suppose x and y are n-element column vectors. Then

$$\begin{aligned} x^T y &= x_1 y_1 + \cdots + x_n y_n \\ \frac{\partial (x^T y)}{\partial x} &= \begin{bmatrix} \partial (x^T y)/\partial x_1 & \cdots & \partial (x^T y)/\partial x_n \end{bmatrix} \\ &= \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix} \\ &= y^T \end{aligned} \tag{1.55}$$

Likewise, we can obtain

$$\frac{\partial (x^T y)}{\partial y} = x^T \tag{1.56}$$

Now we will compute the partial derivative of a quadratic with respect to a vector. First write the quadratic as follows:

$$\begin{aligned} x^T A x &= \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix} \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \\ &= \begin{bmatrix} \sum_i x_i A_{i1} & \cdots & \sum_i x_i A_{in} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \\ &= \sum_{i,j} x_i x_j A_{ij} \end{aligned} \tag{1.57}$$

Now take the partial derivative of the quadratic as follows:

$$\begin{aligned} \frac{\partial (x^T A x)}{\partial x} &= \begin{bmatrix} \partial (x^T A x)/\partial x_1 & \cdots & \partial (x^T A x)/\partial x_n \end{bmatrix} \\ &= \begin{bmatrix} \sum_j x_j A_{1j} + \sum_i x_i A_{i1} & \cdots & \sum_j x_j A_{nj} + \sum_i x_i A_{in} \end{bmatrix} \\ &= \begin{bmatrix} \sum_j x_j A_{1j} & \cdots & \sum_j x_j A_{nj} \end{bmatrix} + \begin{bmatrix} \sum_i x_i A_{i1} & \cdots & \sum_i x_i A_{in} \end{bmatrix} \\ &= x^T A^T + x^T A \end{aligned} \tag{1.58}$$

If A is symmetric, as it often is in quadratic expressions, then $A = A^T$ and the above expression simplifies to

$$\frac{\partial(x^T A x)}{\partial x} = 2x^T A \qquad \text{if } A = A^T \tag{1.59}$$

Next we define the partial derivative of a vector with respect to another vector. Suppose $g(x) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_m(x) \end{bmatrix}$ and $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$. Then

$$\frac{\partial g}{\partial x} = \begin{bmatrix} \partial g_1/\partial x_1 & \cdots & \partial g_1/\partial x_n \\ \vdots & & \vdots \\ \partial g_m/\partial x_1 & \cdots & \partial g_m \partial x_n \end{bmatrix} \tag{1.60}$$

If either $g(x)$ or x is transposed, then the partial derivative is also transposed.

$$\begin{aligned} \frac{\partial g^T}{\partial x} &= \left(\frac{\partial g}{\partial x}\right)^T \\ \frac{\partial g}{\partial x^T} &= \left(\frac{\partial g}{\partial x}\right)^T \\ \frac{\partial g^T}{\partial x^T} &= \frac{\partial g}{\partial x} \end{aligned} \tag{1.61}$$

With these definitions, the following important equalities can be derived. Suppose A is an $m \times n$ matrix and x is an $n \times 1$ vector. Then

$$\begin{aligned} \frac{\partial(Ax)}{\partial x} &= A \\ \frac{\partial(x^T A)}{\partial x} &= A \end{aligned} \tag{1.62}$$

Now we suppose that A is an $m \times n$ matrix, B is an $n \times n$ matrix, and we want to compute the partial derivative of $\mathrm{Tr}(ABA^T)$ with respect to A. First compute ABA^T as follows:

$$\begin{aligned} ABA^T &= \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix} \begin{bmatrix} B_{11} & \cdots & B_{1n} \\ \vdots & \ddots & \vdots \\ B_{n1} & \cdots & B_{nn} \end{bmatrix} \begin{bmatrix} A_{11} & \cdots & A_{m1} \\ \vdots & \ddots & \vdots \\ A_{1n} & \cdots & A_{nn} \end{bmatrix} \\ &= \begin{bmatrix} \sum_{j,k} A_{1k} B_{kj} A_{1j} & \cdots & \sum_{j,k} A_{1k} B_{kj} A_{mj} \\ \vdots & & \vdots \\ \sum_{j,k} A_{mk} B_{kj} A_{1j} & \cdots & \sum_{j,k} A_{mk} B_{kj} A_{mj} \end{bmatrix} \end{aligned} \tag{1.63}$$

From this we see that the trace of ABA^T is given as

$$\mathrm{Tr}(ABA^T) = \sum_{i,j,k} A_{ik} B_{kj} A_{ij} \tag{1.64}$$

Its partial derivative with respect to A can be computed as

$$\begin{aligned}
\frac{\partial \mathrm{Tr}(ABA^T)}{\partial A} &= \begin{bmatrix} \partial \mathrm{Tr}(ABA^T)/\partial A_{11} & \cdots & \partial \mathrm{Tr}(ABA^T)/\partial A_{1n} \\ \vdots & & \vdots \\ \partial \mathrm{Tr}(ABA^T)/\partial A_{m1} & \cdots & \partial \mathrm{Tr}(ABA^T)/\partial A_{mn} \end{bmatrix} \\
&= \begin{bmatrix} \sum_j A_{1j}B_{1j} + \sum_k A_{1k}B_{k1} & \cdots & \sum_j A_{1j}B_{nj} + \sum_k A_{1k}B_{kn} \\ \vdots & \ddots & \vdots \\ \sum_j A_{mj}B_{1j} + \sum_k A_{mk}B_{k1} & \cdots & \sum_j A_{mj}B_{nj} + \sum_k A_{mk}B_{kn} \end{bmatrix} \\
&= \begin{bmatrix} \sum_j A_{1j}B_{1j} & \cdots & \sum_j A_{1j}B_{nj} \\ \vdots & & \vdots \\ \sum_j A_{mj}B_{1j} & \cdots & \sum_j A_{mj}B_{nj} \end{bmatrix} + \\
&\quad \begin{bmatrix} \sum_k A_{1k}B_{k1} & \cdots & \sum_k A_{1k}B_{kn} \\ \vdots & & \vdots \\ \sum_k A_{mk}B_{k1} & \cdots & \sum_k A_{mk}B_{kn} \end{bmatrix} \\
&= AB^T + AB
\end{aligned} \tag{1.65}$$

If B is symmetric, as it often is in partial derivatives of the form above, then this can be simplified to

$$\frac{\partial \mathrm{Tr}(ABA^T)}{\partial A} = 2AB \qquad \text{if } B = B^T \tag{1.66}$$

A number of additional interesting results related to matrix calculus can be found in [Ske98, Appendix B].

1.1.4 The history of matrices

This section is a brief diversion to present some of the history of matrix theory. Much of the information in this section is taken from [OCo96].

The use of matrices can be found as far back as the fourth century BC. We see in ancient clay tablets that the Babylonians studied problems that led to simultaneous linear equations. For example, a tablet dating from about 300 BC contains the following problem: "There are two fields whose total area is 1800 units. One produces grain at the rate of 2/3 of a bushel per unit while the other produces grain at the rate of 1/2 a bushel per unit. If the total yield is 1100 bushels, what is the size of each field?"

Later, the Chinese came even closer to the use of matrices. In [She99] (originally published between 200 BC and 100 AD) we see the following problem: "There are three types of corn, of which three bundles of the first, two of the second, and one of the third make 39 measures. Two of the first, three of the second, and one of the third make 34 measures. And one of the first, two of the second and three of the third make 26 measures. How many measures of corn are contained in one bundle of each type?" At that point, the ancient Chinese essentially use Gaussian elimination (which was not well known until the 19th century) to solve the problem.

In spite of this very early beginning, it was not until the end of the 17th century that serious investigation of matrix algebra began. In 1683, the Japanese

mathematician Takakazu Seki Kowa wrote a book called "Method of Solving the Dissimulated Problems." This book gives general methods for calculating determinants and presents examples for matrices as large as 5×5. Coincidentally, in the same year (1683) Gottfried Leibniz in Europe also first used determinants to solve systems of linear equations. Leibniz also discovered that a determinant could be expanded using any of the matrix columns.

In the middle of the 1700s, Colin Maclaurin and Gabriel Cramer published some major contributions to matrix theory. After that point, work on matrices became rather regular, with significant contributions by Etienne Bezout, Alexandre Vandermonde, Pierre Laplace, Joseph Lagrange, and Carl Gauss. The term "determinant" was first used in the modern sense by Augustin Cauchy in 1812 (although the word was used earlier by Gauss in a different sense). Cauchy also discovered matrix eigenvalues and diagonalization, and introduced the idea of similar matrices. He was the first to prove that every real symmetric matrix is diagonalizable.

James Sylvester (in 1850) was the first to use the term "matrix." Sylvester moved to England in 1851 to became a lawyer and met Arthur Cayley, a fellow lawyer who was also interested in mathematics. Cayley saw the importance of the idea of matrices and in 1853 he invented matrix inversion. Cayley also proved that 2×2 and 3×3 matrices satisfy their own characteristic equations. The fact that a matrix satisfies its own characteristic equation is now called the Cayley–Hamilton theorem (see Problem 1.5). The theorem has William Hamilton's name associated with it because he proved the theorem for 4×4 matrices during the course of his work on quaternions.

Camille Jordan invented the Jordan canonical form of a matrix in 1870. Georg Frobenius proved in 1878 that all matrices satisfy their own characteristic equation (the Cayley Hamilton theorem). He also introduced the definition of the rank of a matrix. The nullity of a square matrix was defined by Sylvester in 1884. Karl Weierstrass's and Leopold Kronecker's publications in 1903 were instrumental in establishing matrix theory as an important branch of mathematics. Leon Mirsky's book in 1955 [Mir90] helped solidify matrix theory as a fundamentally important topic in university mathematics.

1.2 LINEAR SYSTEMS

Many processes in our world can be described by state-space systems. These include processes in engineering, economics, physics, chemistry, biology, and many other areas. If we can derive a mathematical model for a process, then we can use the tools of mathematics to control the process and obtain information about the process. This is why state-space systems are so important to engineers. If we know the state of a system at the present time, and we know all of the present and future inputs, then we can deduce the values of all future outputs of the system.

State-space models can be generally divided into linear models and nonlinear models. Although most real processes are nonlinear, the mathematical tools that are available for estimation and control are much more accessible and well understood for linear systems. That is why nonlinear systems are often approximated as linear systems. That way we can use the tools that have been developed for linear systems to derive estimation or control algorithms.

A continuous-time, deterministic linear system can be described by the equations

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{1.67}$$

where x is the state vector, u is the control vector, and y is the output vector. Matrices A, B, and C are appropriately dimensioned matrices. The A matrix is often called the system matrix, B is often called the input matrix, and C is often called the output matrix. In general, A, B, and C can be time-varying matrices and the system will still be linear. If A, B, and C are constant then the solution to Equation (1.67) is given by

$$\begin{aligned} x(t) &= e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)\,d\tau \\ y(t) &= Cx(t) \end{aligned} \tag{1.68}$$

where t_0 is the initial time of the system and is often taken to be 0. This is easy to verify when all of the quantities in Equation (1.67) are scalar, but it happens to be true in the vector case also. Note that in the zero input case, $x(t)$ is given as

$$x(t) = e^{A(t-t_0)}x(t_0), \qquad \text{zero input case} \tag{1.69}$$

For this reason, e^{At} is called the state-transition matrix of the system.[3] It is the matrix that describes how the state changes from its initial condition in the absence of external inputs. We can evaluate the above equation at $t = t_0$ to see that

$$e^{A0} = I \tag{1.70}$$

in analogy with the scalar exponential of zero.

As stated above, even if x is an n-element vector, then Equation (1.68) still describes the solution of Equation (1.67). However, a fundamental question arises in this case: How can we take the exponential of the matrix A in Equation (1.68)? What does it mean to raise the scalar e to the power of a matrix? There are many different ways to compute this quantity [Mol03]. Three of the most useful are the following:

$$\begin{aligned} e^{At} &= \sum_{j=0}^{\infty} \frac{(At)^j}{j!} \\ &= \mathcal{L}^{-1}[(sI - A)^{-1}] \\ &= Qe^{\hat{A}t}Q^{-1} \end{aligned} \tag{1.71}$$

The first expression above is the definition of e^{At}, and is analogous to the definition of the exponential of a scalar. This definition shows that A must be square in order for e^{At} to exist. From Equation (1.67), we see that a system matrix is always square. The definition of e^{At} can also be used to derive the following properties.

$$\begin{aligned} \frac{d}{dt}e^{At} &= Ae^{At} \\ &= e^{At}A \end{aligned} \tag{1.72}$$

[3]The MATLAB function EXPM computes the matrix exponential. Note that the MATLAB function EXP computes the element-by-element exponential of a matrix, which is generally not the same as the matrix exponential.

In general, matrices do not commute under multiplication but, interestingly, a matrix always commutes with its exponential.

The first expression in Equation (1.71) is not usually practical for computational purposes since it is an infinite sum (although the latter terms in the sum often decrease rapidly in magnitude, and may even become zero). The second expression in Equation (1.71) uses the inverse Laplace transform to compute e^{At}. In the third expression of Equation (1.71), Q is a matrix whose columns comprise the eigenvectors of A, and $\hat{A}$ is the Jordan form[4] of A. Note that Q and $\hat{A}$ are well defined for any square matrix A, so the matrix exponential e^{At} exists for all square matrices A and all finite t. The matrix $\hat{A}$ is often diagonal, in which case $e^{\hat{A}t}$ is easy to compute:

$$\begin{aligned} \hat{A} &= \begin{bmatrix} \hat{A}_{11} & 0 & \cdots & 0 \\ 0 & \hat{A}_{22} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \hat{A}_{nn} \end{bmatrix} \\ e^{\hat{A}t} &= \begin{bmatrix} e^{\hat{A}_{11}} & 0 & \cdots & 0 \\ 0 & e^{\hat{A}_{22}} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & e^{\hat{A}_{nn}} \end{bmatrix} \end{aligned} \tag{1.73}$$

This can be computed from the definition of $e^{\hat{A}t}$ in Equation (1.71). Even if the Jordan form matrix $\hat{A}$ is not diagonal, $e^{\hat{A}t}$ is easy to compute [Bay99, Che99, Kai80]. We can also note from the third expression in Equation (1.71) that

$$\begin{aligned} \left[e^{At}\right]^{-1} &= e^{-At} \\ &= Qe^{-\hat{A}t}Q^{-1} \end{aligned} \tag{1.74}$$

(Recall that A and $-A$ have the same eigenvectors, and their eigenvalues are negatives of each other. See Problem 1.10.) We see from this that e^{At} is always invertible. This is analogous to the scalar situation in which the exponential of a scalar is always nonzero.

Another interesting fact about the matrix exponential is that all of the individual elements of the matrix exponential e^A are nonnegative if and only if all of the individual elements of A are nonnegative [Bel60, Bel80].

■ EXAMPLE 1.2

As an example of a linear system, suppose that we are controlling the angular acceleration of a motor (for example, with some applied voltage across the motor windings). The derivative of the position is the velocity. A simplified motor model can then be written as

[4]In fact, Equation (1.71) can be used to define the Jordan form of a matrix. That is, if e^{At} can be written as shown in Equation (1.71), where Q is a matrix whose columns comprise the eigenvectors of A, then $\hat{A}$ is the Jordan form of A. More discussion about Jordan forms and their computation can be found in most linear systems books [Kai80, Bay99, Che99].

$$\begin{aligned} \dot{\theta} &= \omega \\ \dot{\omega} &= u + w_1 \end{aligned} \tag{1.75}$$

The scalar w_1 is the acceleration noise and could consist of such factors as uncertainty in the applied acceleration, motor shaft eccentricity, and load disturbances. If our measurement consists of the angular position of the motor then a state space description of this system can be written as

$$\begin{aligned} \begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ w_1 \end{bmatrix} \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x + v \end{aligned} \tag{1.76}$$

The scalar v consists of measurement noise. Comparing with Equation (1.67), we see that the state vector x is a 2×1 vector containing the scalars θ and ω.

$\nabla\nabla\nabla$

■ **EXAMPLE 1.3**

In this example, we will use the three expressions in Equation (1.71) to compute the state-transition matrix of the system described in Example 1.2. From the first expression in Equation (1.71) we obtain

$$\begin{aligned} e^{At} &= \sum_{j=0}^{\infty} \frac{(At)^j}{j!} \\ &= (At)^0 + (At)^1 + \frac{(At)^2}{2!} + \frac{(At)^3}{3!} + \cdots \\ &= I + At \end{aligned} \tag{1.77}$$

where the last equality comes from the fact that $A^k = 0$ when $k > 1$ for the A matrix given in Example 1.2. We therefore obtain

$$\begin{aligned} e^{At} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & t \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{1.78}$$

From the second expression in Equation (1.71) we obtain

$$\begin{aligned} e^{At} &= \mathcal{L}^{-1}[(sI - A)^{-1}] \\ &= \mathcal{L}^{-1}\left(\begin{bmatrix} s & -1 \\ 0 & s \end{bmatrix}^{-1} \right) \\ &= \mathcal{L}^{-1} \begin{bmatrix} 1/s & 1/s^2 \\ 0 & 1/s \end{bmatrix} \\ &= \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{1.79}$$

In order to use the third expression in Equation (1.71) we first need to obtain the eigendata (i.e., the eigenvalues and eigenvectors) of the A matrix. These are found as

$$\begin{aligned} \lambda(A) &= \{0,0\} \\ v(A) &= \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\} \end{aligned} \tag{1.80}$$

This shows that

$$\begin{aligned} \hat{A} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \\ Q &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{1.81}$$

Note that in this simple example A is already in Jordan form, so $\hat{A} = A$ and $Q = I$. The third expression in Equation (1.71) therefore gives

$$\begin{aligned} e^{At} &= Qe^{\hat{A}t}Q^{-1} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{1.82}$$

$\nabla\nabla\nabla$

1.3 NONLINEAR SYSTEMS

The discussion of linear systems in the preceding section is a bit optimistic, because in reality linear systems do not exist. Real systems always have some nonlinearities. Even a simple resistor is ultimately nonlinear if we apply a large enough voltage across it. However, we often model a resistor with the simple linear equation $V = IR$ because this equation accurately describes the operation of the resistor over a wide operating range. So even though linear systems do not exist in the real world, linear systems theory is still a valuable tool for dealing with nonlinear systems.

The general form of a continuous-time nonlinear system can be written as

$$\begin{aligned} \dot{x} &= f(x,u,w) \\ y &= h(x,v) \end{aligned} \tag{1.83}$$

where $f(\cdot)$ and $h(\cdot)$ are arbitrary vector-valued functions. We use w to indicate process noise, and v to indicate measurement noise. If $f(\cdot)$ and $h(\cdot)$ are explicit functions of t then the system is time-varying. Otherwise, the system is time-invariant. If $f(x,u,w) = Ax + Bu + w$, and $h(x,v) = Hx + v$, then the system is linear [compare with Equation (1.67)]. Otherwise, the system is nonlinear.

In order to apply tools from linear systems theory to nonlinear systems, we need to linearize the nonlinear system. In other words, we need to find a linear system

that is approximately equal to the nonlinear system. To see how this is done, let us start with a nonlinear vector function $f(\cdot)$ of a scalar x. We expand $f(x)$ in a Taylor series around some nominal operating point (also called a linearization point) $x = \bar{x}$, defining $\tilde{x} = x - \bar{x}$:

$$f(x) = f(\bar{x}) + \left.\frac{\partial f}{\partial x}\right|_{\bar{x}} \tilde{x} + \frac{1}{2!}\left.\frac{\partial^2 f}{\partial x^2}\right|_{\bar{x}} \tilde{x}^2 + \frac{1}{3!}\left.\frac{\partial^3 f}{\partial x^3}\right|_{\bar{x}} \tilde{x}^3 + \cdots \tag{1.84}$$

Now suppose that x is a 2×1 vector. This implies that $f(x)$ is a nonlinear function of two independent variables x_1 and x_2. The Taylor series expansion of $f(x)$ becomes

$$\begin{aligned} f(x) \;=\; & f(\bar{x}) + \left.\frac{\partial f}{\partial x_1}\right|_{\bar{x}} \tilde{x}_1 + \left.\frac{\partial f}{\partial x_2}\right|_{\bar{x}} \tilde{x}_2 + \\ & \frac{1}{2!}\left(\left.\frac{\partial^2 f}{\partial x_1^2}\right|_{\bar{x}} \tilde{x}_1^2 + \left.\frac{\partial^2 f}{\partial x_2^2}\right|_{\bar{x}} \tilde{x}_2^2 + 2\left.\frac{\partial^2 f}{\partial x_1 x_2}\right|_{\bar{x}} \tilde{x}_1\tilde{x}_2\right) + \\ & \frac{1}{3!}\left(\left.\frac{\partial^3 f}{\partial x_1^3}\right|_{\bar{x}} \tilde{x}_1^3 + \left.\frac{\partial^3 f}{\partial x_2^3}\right|_{\bar{x}} \tilde{x}_2^3 + 3\left.\frac{\partial^3 f}{\partial x_1^2 x_2}\right|_{\bar{x}} \tilde{x}_1^2\tilde{x}_2 + 3\left.\frac{\partial^3 f}{\partial x_1 x_2^2}\right|_{\bar{x}} \tilde{x}_1\tilde{x}_2^2\right) + \cdots \end{aligned} \tag{1.85}$$

This can be written more compactly as

$$\begin{aligned} f(x) \;=\; & f(\bar{x}) + \left.\left(\tilde{x}_1\frac{\partial}{\partial x_1} + \tilde{x}_2\frac{\partial}{\partial x_2}\right) f\right|_{\bar{x}} + \frac{1}{2!}\left.\left(\tilde{x}_1\frac{\partial}{\partial x_1} + \tilde{x}_2\frac{\partial}{\partial x_2}\right)^2 f\right|_{\bar{x}} + \\ & \frac{1}{3!}\left.\left(\tilde{x}_1\frac{\partial}{\partial x_1} + \tilde{x}_2\frac{\partial}{\partial x_2}\right)^3 f\right|_{\bar{x}} + \cdots \end{aligned} \tag{1.86}$$

Extending this to the general case in which x is an $n \times 1$ vector, we see that any continuous vector-valued function $f(x)$ can be expanded in a Taylor series as

$$\begin{aligned} f(x) \;=\; & f(\bar{x}) + \left.\left(\tilde{x}_1\frac{\partial}{\partial x_1} + \cdots + \tilde{x}_2\frac{\partial}{\partial x_n}\right) f\right|_{\bar{x}} + \\ & \frac{1}{2!}\left.\left(\tilde{x}_1\frac{\partial}{\partial x_1} + \cdots + \tilde{x}_n\frac{\partial}{\partial x_n}\right)^2 f\right|_{\bar{x}} + \\ & \frac{1}{3!}\left.\left(\tilde{x}_1\frac{\partial}{\partial x_1} + \cdots + \tilde{x}_n\frac{\partial}{\partial x_n}\right)^3 f\right|_{\bar{x}} + \cdots \end{aligned} \tag{1.87}$$

Now we define the operation $D_{\tilde{x}}^k f$ as

$$D_{\tilde{x}}^k f = \left.\left(\sum_{i=1}^{n} \tilde{x}_i \frac{\partial}{\partial x_i}\right)^k f(x)\right|_{\bar{x}} \tag{1.88}$$

Using this definition we write the Taylor series expansion of $f(x)$ as

$$f(x) = f(\bar{x}) + D_{\tilde{x}} f + \frac{1}{2!} D_{\tilde{x}}^2 f + \frac{1}{3!} D_{\tilde{x}}^3 f + \cdots \tag{1.89}$$

If the nonlinear function $f(x)$ is "sufficiently smooth," then high-order derivatives of $f(x)$ should be "somewhat small." Also, if $f(x)$ is expanded around a point such

that x is "close" to $\bar{x}$, then $\tilde{x}$ will be "small" and the higher powers of $\tilde{x}$ in Equation (1.89) will be "small." Finally, the higher-order derivatives in the Taylor series expansion of Equation (1.89) are divided by increasingly large factorials, which further diminishes the magnitude of the higher-order terms in Equation (1.89). This justifies the approximation

$$\begin{aligned} f(x) &\approx f(\bar{x}) + D_{\tilde{x}} f \\ &\approx f(\bar{x}) + \left.\frac{\partial f}{\partial x}\right|_{\bar{x}} \tilde{x} \\ &\approx f(\bar{x}) + A\tilde{x} \end{aligned} \tag{1.90}$$

where A is the matrix defined by the above equation.

Returning to our nonlinear system equations in Equation (1.83), we can expand the nonlinear system equation $f(x, u, w)$ around the nominal operating point $(\bar{x}, \bar{u}, \bar{w})$. We then obtain a linear system approximation as follows.

$$\begin{aligned} \dot{x} &= f(x, u, w) \\ &\approx f(\bar{x}, \bar{u}, \bar{w}) + \left.\frac{\partial f}{\partial x}\right|_0 (x - \bar{x}) + \left.\frac{\partial f}{\partial u}\right|_0 (u - \bar{u}) + \left.\frac{\partial f}{\partial w}\right|_0 (w - \bar{w}) \\ &= \dot{\bar{x}} + A\tilde{x} + B\tilde{u} + L\tilde{w} \end{aligned} \tag{1.91}$$

where the 0 subscript means that the function is evaluated at the nominal point $(\bar{x}, \bar{u}, \bar{w})$, and A, B, and L are defined by the above equations. Subtracting $\dot{\bar{x}}$ from both sides of Equation (1.91) gives

$$\dot{\tilde{x}} = A\tilde{x} + B\tilde{u} + L\tilde{w} \tag{1.92}$$

Since w is noise, we will set $\bar{w} = 0$ so that $\tilde{w} = w$ and we obtain

$$\dot{\tilde{x}} = A\tilde{x} + B\tilde{u} + Lw \tag{1.93}$$

We see that we have a linear equation for $\dot{\tilde{x}}$ in terms of $\tilde{x}$, $\tilde{u}$, and w. We have a linear equation for the deviations of the state and control from their nominal values. As long as the deviations remain small, the linearization will be accurate and the linear equation will accurately describe deviations of x from its nominal value $\bar{x}$.

In a similar manner we can expand the nonlinear measurement equation given by Equation (1.83) around a nominal operating point $x = \bar{x}$ and $v = \bar{v} = 0$. This results in the linearized measurement equation

$$\begin{aligned} \tilde{y} &= \left.\frac{\partial h}{\partial x}\right|_0 \tilde{x} + \left.\frac{\partial h}{\partial v}\right|_0 \tilde{v} \\ &= C\tilde{x} + Dv \end{aligned} \tag{1.94}$$

where C and D are defined by the above equation. Equations (1.93) and (1.94) comprise a linear system that describes the deviations of the state and output from their nominal values. Recall that the tilde quantities in Equations (1.93) and (1.94) are defined as

$$\begin{aligned} \tilde{x} &= x - \bar{x} \\ \tilde{u} &= u - \bar{u} \\ \tilde{y} &= y - \bar{y} \end{aligned} \tag{1.95}$$

■ EXAMPLE 1.4

Consider the following model for a two-phase permanent magnet synchronous motor:

$$\begin{aligned}
\dot{i}_a &= \frac{-R}{L} i_a + \frac{\omega\lambda}{L}\sin\theta + \frac{u_a}{L} \\
\dot{i}_b &= \frac{-R}{L} i_b - \frac{\omega\lambda}{L}\cos\theta + \frac{u_b}{L} \\
\dot{\omega} &= \frac{-3\lambda}{2J} i_a \sin\theta + \frac{3\lambda}{2J} i_b \cos\theta - \frac{F\omega}{J} - \frac{T_l}{J} \\
\dot{\theta} &= \omega
\end{aligned} \tag{1.96}$$

where i_a and i_b are the currents through the two windings, R and L are the resistance and inductance of the windings, θ and ω are the angular position and velocity of the rotor, λ is the flux constant of the motor, u_a and u_b are the voltages applied across the two windings, J is the moment of inertia of the rotor and its load, F is the viscous friction of the rotor, and T_l is the load torque. The time variable does not explicitly appear on the right side of the above equation, so this is a time-invariant system. However, the system is highly nonlinear and we therefore cannot directly use any linear systems tools for control or estimation. However, if we linearize the system around a nominal (possibly time-varying) operating point then we can use linear system tools for control and estimation. We start by defining a state vector as $x = \begin{bmatrix} i_a & i_b & \omega & \theta \end{bmatrix}^T$. With this definition we write

$$\begin{aligned}
\dot{x} &= \begin{bmatrix} \dot{x}_1 & \dot{x}_2 & \dot{x}_3 & \dot{x}_4 \end{bmatrix}^T \\
&= f(x,u) \\
&= \begin{bmatrix} \frac{-R}{L}x_1 + \frac{x_3\lambda}{L}\sin x_4 + \frac{u_a}{L} \\ \frac{-R}{L}x_2 - \frac{x_3\lambda}{L}\cos x_4 + \frac{u_b}{L} \\ \frac{-3\lambda}{2J}x_1 \sin x_4 + \frac{3\lambda}{2J}x_2\cos x_4 - \frac{Fx_3}{J} - \frac{T_l}{J} \\ x_3 \end{bmatrix}
\end{aligned} \tag{1.97}$$

We linearize the system equation by taking the partial derivative of $f(x,u)$ with respect to x and u to obtain

$$\begin{aligned}
A &= \frac{\partial f}{\partial x} \\
&= \begin{bmatrix} -R/L & 0 & \lambda s_4/L & x_3\lambda c_4/L \\ 0 & -R/L & -\lambda c_4/L & x_3\lambda s_4/L \\ -3\lambda s_4/2/J & 3\lambda c_4/2/J & -F/J & -3\lambda(x_1c_4 + x_2s_4)/2/J \\ 0 & 0 & 1 & 0 \end{bmatrix} \\
B &= \frac{\partial f}{\partial u} \\
&= \begin{bmatrix} 1/L & 0 \\ 0 & 1/L \\ 0 & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned} \tag{1.98}$$

where $s_4 = \sin x_4$ and $c_4 = \cos x_4$. The linear system

$$\dot{\tilde{x}} = A\tilde{x} + B\tilde{u} \tag{1.99}$$

approximately describes the deviation of x from its nominal value $\bar{x}$. The nonlinear system was simulated with the nominal control values $\bar{u}_a(t) = \sin 2\pi t$ and $\bar{u}_b(t) = \cos 2\pi t$. This resulted in a nominal state trajectory $\bar{x}(t)$. The linear and nonlinear systems were then simulated with nonnominal control values. Figure 1.1 shows the results of the linear and nonlinear simulations when the control magnitude deviation from nominal is a small positive number. It can be seen that the simulations result in similar state-space trajectories, although they do not match exactly. If the deviation is zero, then the linear and nonlinear simulations will match exactly. As the deviation from nominal increases, the difference between the linear and nonlinear simulations will increase.

Figure 1.1 Example 1.4 comparison of nonlinear and linearized motor simulations.

∇∇∇

1.4 DISCRETIZATION

Most systems in the real world are described with continuous-time dynamics of the type shown in Equations (1.67) or (1.83). However, state estimation and control algorithms are almost always implemented in digital electronics. This often requires a transformation of continuous-time dynamics to discrete-time dynamics. This section discusses how a continuous-time linear system can be transformed into a discrete-time linear system.

Recall from Equation (1.68) that the solution of a continuous-time linear system is given by

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)\,d\tau \tag{1.100}$$

Let $t = t_k$ (some discrete time point) and let the initial time $t_0 = t_{k-1}$ (the previous discrete time point). Assume that $A(\tau)$, $B(\tau)$, and $u(\tau)$ are approximately constant in the interval of integration. We then obtain

$$x(t_k) = e^{A(t_k - t_{k-1})} x(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{A(t_k - \tau)}\, d\tau\, Bu(t_{k-1}) \tag{1.101}$$

Now define $\Delta t = t_k - t_{k-1}$, define $\alpha = \tau - t_{k-1}$, and substitute for τ in the above equation to obtain

$$\begin{aligned} x(t_k) &= e^{A\Delta t} x(t_{k-1}) + \int_0^{\Delta t} e^{A(\Delta t - \alpha)}\, d\alpha\, Bu(t_{k-1}) \\ &= e^{A\Delta t} x(t_{k-1}) + e^{A\Delta t} \int_0^{\Delta t} e^{-A\alpha}\, d\alpha\, Bu(t_{k-1}) \\ x_k &= F_{k-1} x_{k-1} + G_{k-1} u_{k-1} \end{aligned} \tag{1.102}$$

where x_k, F_k, G_k, and u_k are defined by the above equation. This is a linear discrete-time approximation to the continuous-time dynamics given in Equation (1.67). Note that this discrete-time system defines x_k only at the discrete time points $\{t_k\}$; it does not say anything about what happens to the continuous-time signal $x(t)$ in between the discrete time points.

The difficulty with the above discrete-time system is the computation of the integral of the matrix exponential, which is necessary in order to compute the G matrix. This computation can be simplified if A is invertible:

$$\begin{aligned} \int_0^{\Delta t} e^{-A\tau} d\tau &= \int_0^{\Delta t} \sum_{j=0}^{\infty} \frac{(-A\tau)^j}{j!}\, d\tau \\ &= \int_0^{\Delta t} \left[I - A\tau + A^2\tau^2/2! - \cdots \right] d\tau \\ &= \left[I\tau - A\tau^2/2! + A^2\tau^3/3! - \cdots \right]_0^{\Delta t} \\ &= \left[I\Delta t - A(\Delta t)^2/2! + A^2(\Delta t)^3/3! - \cdots \right] \\ &= \left[A\Delta t - (A\Delta t)^2/2! + (A\Delta t)^3/3! - \cdots \right] A^{-1} \\ &= \left[I - e^{-A\Delta t} \right] A^{-1} \end{aligned} \tag{1.103}$$

The conversion from continuous-time system matrices A and B to discrete-time system matrices F and G can be summarized as follows:

$$\begin{aligned} F &= e^{A\Delta t} \\ G &= F \int_0^{\Delta t} e^{-A\tau}\, d\tau\, B \\ &= F \left[I - e^{-A\Delta t} \right] A^{-1} B \end{aligned} \tag{1.104}$$

where Δt is the discretization step size.

1.5 SIMULATION

In this section, we discuss how to simulate continuous-time systems (either linear or nonlinear) on a digital computer. We consider the following form of the general

system equation from Equation (1.83):

$$\dot{x} = f(x, u, t) \tag{1.105}$$

where $u(t)$ is a known control input. In order to simulate this system on a computer, we need to program a computer to solve for $x(t_f)$ at some user-specified value of t_f. In other words, we want to compute

$$x(t_f) = x(t_0) + \int_{t_0}^{t_f} f[x(t), u(t), t]\, dt \tag{1.106}$$

Often, the initial time is taken as $t_0 = 0$, in which case we have the slightly simpler looking equation

$$x(t_f) = x(0) + \int_{0}^{t_f} f[x(t), u(t), t]\, dt \tag{1.107}$$

We see that in order to find the solution $x(t_f)$ to the differential equation $\dot{x} = f(x, u, t)$, we need to compute an integral. The problem of finding the solution $x(t_f)$ is therefore commonly referred to as an integration problem.

Now suppose that we divide the time interval $[0, t_f]$ into L equally spaced intervals so that $t_k = kT$ for $k = 0, \cdots, L$, and the time interval $T = t_f/L$. From this we note that $t_f = t_L$. With this division of the time interval, we can write the solution of Equation (1.107) as

$$\begin{aligned} x(t_f) &= x(t_L) \\ &= x(0) + \sum_{k=0}^{L} \int_{t_k}^{t_{k+1}} f[x(t), u(t), t]\, dt \end{aligned} \tag{1.108}$$

More generally, for some $n \in [0, L-1]$, we can write $x(t_n)$ and $x(t_{n+1})$ as

$$\begin{aligned} x(t_n) &= x(0) + \sum_{k=0}^{n} \int_{t_k}^{t_{k+1}} f[x(t), u(t), t]\, dt \\ x(t_{n+1}) &= x(0) + \sum_{k=0}^{n+1} \int_{t_k}^{t_{k+1}} f[x(t), u(t), t]\, dt \end{aligned} \tag{1.109}$$

which means that

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f[x(t), u(t), t]\, dt \tag{1.110}$$

If we can find a way to approximate the integral on the right side of the above equation, we can repeatedly propagate our $x(t)$ approximation from time t_n to time t_{n+1}, thus obtaining an approximation for $x(t)$ at any desired time t. The algorithm could look something like the following.

Differential equation solution

Assume that $x(0)$ is given
for t $= 0 : T : t_f - T$
  Find an approximation $I(t) \approx \int_t^{t+T} f[x(t), u(t), t]\, dt$
  $x(t+T) = x(t) + TI(t)$
end

In the following sections, we present three different ways to approximate this integral. The approximations, in order of increasing computational effort and increasing accuracy, are rectangular integration, trapezoidal integration, and fourth-order Runge–Kutta integration.

1.5.1 Rectangular integration

If the time interval $(t_{n+1} - t_n)$ is small, then $f[x(t), u(t), t]$ is approximately constant in this interval. Equation (1.110) can therefore be approximated as

$$\begin{aligned} x(t_{n+1}) &\approx x(t_n) + \int_{t_n}^{t_{n+1}} f[x(t_n), u(t_n), t_n]\, dt \\ &\approx x(t_n) + f[x(t_n), u(t_n), t_n]T \end{aligned} \tag{1.111}$$

Equation (1.109) can therefore be approximated as

$$\begin{aligned} x(t_n) &\approx x(0) + \sum_{k=0}^{n} \int_{t_k}^{t_{k+1}} f[x(t_k), u(t_k), t_k]\, dt \\ &= x(0) + \sum_{k=0}^{n} f[x(t_k), u(t_k), t_k]T \end{aligned} \tag{1.112}$$

This is called Euler integration, or rectangular integration, and is illustrated in Figure 1.2. As long as T is sufficiently small, this gives a good approximation for $x(t_n)$.

This gives the following algorithm for integrating continuous-time dynamics using rectangular integration. The time loop in the algorithm is executed for $t = 0, T, 2T, \cdots, t_f - T$.

Rectangular integration

Assume that $x(0)$ is given
for t $= 0 : T : t_f - T$
  Compute $f[x(t), u(t), t]$
  $x(t+T) = x(t) + f[x(t), u(t), t]T$
end

1.5.2 Trapezoidal integration

An inspection of Figure 1.2 suggests an idea for improving the approximation for $x(t)$. Instead of approximating each area as a rectangle, what if we approximate each area as a trapezoid? Figure 1.3 shows how an improved integration algorithm can be implemented. This is called modified Euler integration, or trapezoidal integration. A comparison of Figures 1.2 and 1.3 shows that trapezoidal integration

Figure 1.2 An illustration of rectangular integration. We have $\dot{x} = f(x)$, so $x(t)$ is the area under the $f(x)$ curve. This area can be approximated as the sum of the rectangular areas A_i. That is, $x(0.5) \approx A_1$, $x(1) \approx A_1 + A_2$, $\cdots$.

Figure 1.3 An illustration of trapezoidal integration. We have $\dot{x} = f(x)$, so $x(t)$ is the area under the $f(x)$ curve. This area can be approximated as the sum of trapezoidal areas A_i. That is, $x(1) \approx A_1$, $x(2) \approx A_1 + A_2$, and $x(3) \approx A_1 + A_2 + A_3$.

appears to give a better approximation than rectangular integration, even though the time axis is only divided into half as many intervals in trapezoidal integration.

With rectangular integration we approximated $f[x(t), u(t), t]$ as a constant in the interval $t \in [t_n, t_{n+1}]$. With trapezoidal integration, we instead approximate $f[x(t), u(t), t]$ as a linear function in the interval $t \in [t_n, t_{n+1}]$. That is,

$$
\begin{aligned}
f[x(t)] \approx\ & f[x(t_n), u(t_n), t_n] + \\
& \left(\frac{f[x(t_{n+1}), u(t_{n+1}), t_{n+1}] - f[x(t_n), u(t_n), t_n]}{T}\right)(t - t_n) \\
& \text{for } t \in [t_n, t_{n+1}]
\end{aligned} \tag{1.113}
$$

Equation (1.110) can therefore be approximated as

$$
\begin{aligned}
x(t_{n+1}) \approx\ & x(t_n) + \int_{t_n}^{t_{n+1}} \Bigg\{ f[x(t_n), u(t_n), t_n] + \\
& \left(\frac{f[x(t_{n+1}), u(t_{n+1}), t_{n+1}] - f[x(t_n), u(t_n), t_n]}{T}\right)(t - t_n) \Bigg\}\, dt \\
=\ & x(t_n) + \left(\frac{f[x(t_n), u(t_n), t_n] + f[x(t_{n+1}), u(t_{n+1}), t_{n+1}]}{2}\right) T \\
=\ & x(t_n) + \frac{1}{2}\left(f[x(t_n), u(t_n), t_n]T + f[x(t_{n+1}), u(t_{n+1}), t_{n+1}]T\right)
\end{aligned} \tag{1.114}
$$

This equation to approximate $x(t_{n+1})$, however, has $x(t_{n+1})$ on the right side of the equation. How can we plug $x(t_{n+1})$ into the right side of the equation if we do not yet know $x(t_{n+1})$? The answer is that we can use the rectangular integration approximation from the previous section for $x(t_{n+1})$ on the right side of the equation. The above equation can therefore be written as

$$
\begin{aligned}
\Delta x_1 &= f[x(t_n), u(t_n), t_n]T \\
\Delta x_2 &= f[x(t_{n+1}), u(t_{n+1}), t_{n+1}]T \\
&\approx f[x(t_n) + \Delta x_1, u(t_{n+1}), t_{n+1}]T \\
x(t_{n+1}) &\approx x(t_n) + \frac{1}{2}(\Delta x_1 + \Delta x_2)
\end{aligned} \tag{1.115}
$$

This gives the following algorithm for integrating continuous-time dynamics using trapezoidal integration. The time loop in the algorithm is executed for $t = 0, T, 2T, \cdots, t_f - T$.

Trapezoidal integration

```
Assume that x(0) is given
for t = 0 : T : t_f - T
  Δx_1 = f[x(t), u(t), t]T
  Δx_2 = f[x(t) + Δx_1, u(t + T), t + T]T
  x(t + T) = x(t) + (Δx_1 + Δx_2)/2
end
```

1.5.3 Runge–Kutta integration

From the previous sections, we see that rectangular integration involves the calculation of one function value at each time step, and trapezoidal integration involves the calculation of two function values at each time step. In order to further improve the integral approximation, we can perform additional function calculations at each time step. nth-order Runge–Kutta integration is the approximation of an integral

by performing n function calculations at each time step. Rectangular integration is therefore equivalent to first-order Runge–Kutta integration, and trapezoidal integration is equivalent to second-order Runge–Kutta integration.

The most commonly used integration scheme of this type is fourth-order Runge–Kutta integration. We present the fourth-order Runge–Kutta integration algorithm (without derivation) as follows:

$$\begin{aligned}
\Delta x_1 &= f[x(t_k), u(t_k), t_k]T \\
\Delta x_2 &= f[x(t_k) + \Delta x_1/2, u(t_{k+1/2}), t_{k+1/2}]T \\
\Delta x_3 &= f[x(t_k) + \Delta x_2/2, u(t_{k+1/2}), t_{k+1/2}]T \\
\Delta x_4 &= f[x(t_k) + \Delta x_3, u(t_{k+1}), t_{k+1}]T \\
x(t_{k+1}) &\approx x(t_k) + (\Delta x_1 + 2\Delta x_2 + 2\Delta x_3 + \Delta x_4)/6
\end{aligned} \tag{1.116}$$

where $t_{k+1/2} = t_k + T/2$. Fourth-order Runge–Kutta integration is more computationally demanding than rectangular or trapezoidal integration, but it also provides far greater accuracy. This gives the following algorithm for integrating continuous-time dynamics using fourth-order Runge–Kutta integration. The time loop in the algorithm is executed for $t = 0, T, 2T, \cdots, t_f - T$.

Fourth-order Runge–Kutta integration

Assume that $x(0)$ is given
for t = $0 : T : t_f - T$
  $t_1 = t + T/2$
  $\Delta x_1 = f[x(t), u(t), t]T$
  $\Delta x_2 = f[x(t) + \Delta x_1/2, u(t_1), t_1]T$
  $\Delta x_3 = f[x(t) + \Delta x_2/2, u(t_1), t_1]T$
  $\Delta x_4 = f[x(t) + \Delta x_3, u(t+T), t+T]T$
  $x(t+T) = x(t) + (\Delta x_1 + 2\Delta x_2 + 2\Delta x_3 + \Delta x_4)/6$
end

Runge–Kutta integration was invented by Carl Runge, a German mathematician and physicist, in 1895. It was independently invented and generalized by Wilhelm Kutta, a German mathematician and aerodynamicist, in 1901. More accurate integration algorithms have also been derived and are sometimes used, but fourth-order Runge–Kutta integration is generally considered a good trade-off between accuracy and computational effort. Further information and derivations of numerical integration algorithms can be found in many numerical analysis texts, including [Atk89].

■ EXAMPLE 1.5

Suppose we want to numerically compute $x(t)$ at $t = 1$ based on the differential equation

$$\dot{x} = \cos t \tag{1.117}$$

with the initial condition $x(0) = 0$. We can analytically integrate the equation to find out that $x(1) = \sin 1 \approx 0.8415$. If we use a numerical integration scheme, we have to choose the step size T. Table 1.1 shows the error of the rectangular, trapezoidal, and fourth-order Runge–Kutta integration methods for this example for various values of T. As expected, Runge–Kutta is more accurate than trapezoidal, and trapezoidal is more accurate than rectangular.

Also as expected, the error for given method decreases as T decreases. However, perhaps the most noteworthy feature of Table 1.1 is *how* the integration error decreases with T. We can see that with rectangular integration, when T is halved, the integration error is also halved. With trapezoidal integration, when T is halved, the integration error decreases by a factor of four. With Runge–Kutta integration, when T is halved, the integration error decreases by a factor of 16. We conclude that (in general) the error of rectangular integration is proportional to T, the error of trapezoidal integration is proportional to T^2, and the error of Runge–Kutta integration is proportional to T^4.

Table 1.1 Example 1.5 results. Percent errors when numerically integrating $\dot{x} = \cos t$ from $t = 0$ to $t = 1$, for various integration algorithms, and for various time step sizes T.

	$T = 0.1$	$T = 0.05$	$T = 0.025$
Rectangular	2.6	1.3	0.68
Trapezoidal	0.083	0.021	0.0052
Fourth-order Runge–Kutta	3.5×10^{-6}	2.2×10^{-7}	1.4×10^{-8}

$\nabla\nabla\nabla$

1.6 STABILITY

In this section, we review the concept of stability for linear time-invariant systems. We first deal with continuous-time systems in Section 1.6.1, and then discrete-time systems in Section 1.6.2. We state the important results here without proof. The interested reader can refer to standard books on linear systems for more details and additional results [Kai80, Bay99, Che99].

1.6.1 Continuous-time systems

Consider the zero-input, linear, continuous-time system

$$\begin{aligned} \dot{x} &= Ax \\ y &= Cx \end{aligned} \tag{1.118}$$

The definitions of marginal stability and asymptotic stability are as follows.

Definition 1 *A linear continuous-time, time-invariant system is marginally stable if the state $x(t)$ is bounded for all t and for all bounded initial states $x(0)$.*

Marginal stability is also called Lyapunov stability.

Definition 2 *A linear continuous-time, time-invariant system is asymptotically stable if, for all bounded initial states $x(0)$,*

$$\lim_{t \to \infty} x(t) = 0 \tag{1.119}$$

The above two definitions show that a system is marginally stable if it is asymptotically stable. That is, asymptotic stability is a subset of marginal stability. Marginal stability and asymptotic stability are types of internal stability. This is because they deal with only the state of the system (i.e., the internal condition of the system) and do not consider the output of the system. More specific categories of internal stability (e.g., uniform stability and exponential stability) are given in some books on linear systems.

Since the solution of Equation (1.118) is given as

$$x(t) = \exp(At)x(0) \tag{1.120}$$

we can state the following theorem.

Theorem 1 *A linear continuous-time, time-invariant system is marginally stable if and only if*

$$\lim_{t\to\infty} \exp(At) \leq M < \infty \tag{1.121}$$

for some constant matrix M. This is just a way of saying that the matrix exponential does not increase without bound.

The "less than or equal to" relation in the above theorem raises some questions, because the quantities on either side of this mathematical symbol are matrices. What does it mean for a matrix to be less than another matrix? It can be interpreted several ways. For example, to say that $A < B$ is usually interpreted to mean that $(B - A)$ is positive definite.[5] In the above theorem we can use any reasonable definition for the matrix inequality and the theorem still holds.

A similar theorem can be stated by combining Definition (2) with Equation (1.120).

Theorem 2 *A linear continuous-time, time-invariant system is asymptotically stable if and only if*

$$\lim_{t\to\infty} \exp(At) = 0 \tag{1.122}$$

Now recall that $\exp(At) = Q\exp(\hat{A}t)Q^{-1}$, where Q is a constant matrix containing the eigenvectors of A, and $\hat{A}$ is the Jordan form of A. The exponential $\exp(\hat{A}t)$ therefore contains terms like $\exp(\lambda_i t)$, $t\exp(\lambda_i t)$, $t^2\exp(\lambda_i t)$, and so on, where λ_i is an eigenvalue of A. The boundedness of $\exp(At)$ is therefore related to the eigenvalues of A as stated by the following theorems.

Theorem 3 *A linear continuous-time, time-invariant system is marginally stable if and only if one of the following conditions holds.*

1. *All of the eigenvalues of A have negative real parts.*
2. *All of the eigenvalues of A have negative or zero real parts, and those with real parts equal to zero have a geometric multiplicity equal to their algebraic multiplicity. That is, the Jordan blocks that are associated with the eigenvalues that have real parts equal to zero are first order.*

Theorem 4 *A linear continuous-time, time-invariant system is asymptotically stable if and only if all of the eigenvalues of A have negative real parts.*

[5] Sometimes the statement $A < B$ means that every element of A is less than the corresponding element of B. However, we will not use that definition in this book.

■ **EXAMPLE 1.6**

Consider the system

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} x \tag{1.123}$$

Since the A matrix is upper triangular, we know that its eigenvalues are on the diagonal; that is, the eigenvalues of A are equal to 0, 0, and -1. We see that the system is asymptotically unstable since some of the eigenvalues are nonnegative. We also note that the A matrix is already in Jordan form, and we see that the Jordan block corresponding to the 0 eigenvalue is second order. Therefore, the system is also marginally unstable. The solution of this system is

$$\begin{aligned} x(t) &= \exp(At)x(0) \\ &= \begin{bmatrix} 1 & t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-t} \end{bmatrix} x(0) \end{aligned} \tag{1.124}$$

The element in the first row and second column of $\exp(At)$ increases without bound as t increases, so there are some initial states $x(0)$ that will result in unbounded $x(t)$. However, there are also some initial states $x(0)$ that will result in bounded $x(t)$. For example, if $x(0) = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}^T$, then

$$\begin{aligned} x(t) &= \begin{bmatrix} 1 & t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-t} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 0 \\ e^{-t} \end{bmatrix} \end{aligned} \tag{1.125}$$

and $x(t)$ will be bounded for all t. However, this does not say anything about the stability of the system; it only says that there exists some $x(0)$ that results in a bounded $x(t)$. If we instead choose $x(0) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$, then

$$\begin{aligned} x(t) &= \begin{bmatrix} 1 & t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-t} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} t \\ 1 \\ 0 \end{bmatrix} \end{aligned} \tag{1.126}$$

and $x(t)$ increases without bound. This proves that the system is asymptotically unstable and marginally unstable.

∇∇∇

■ EXAMPLE 1.7

Consider the system

$$\dot{x} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} x \tag{1.127}$$

The eigenvalues of A are equal to 0, 0, and -1. We see that the system is asymptotically unstable since some of the eigenvalues are nonnegative. In order to see if the system is marginally stable, we need to compute the geometric multiplicity of the 0 eigenvalue. (This can be done by noticing that A is already in Jordan form, but we will go through the exercise more completely for the sake of illustration.) Solving the equation

$$(\lambda I - A)v = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{1.128}$$

(where $\lambda = 0$) for nonzero vectors v, we see that there are two linearly independent solutions given as

$$v = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{1.129}$$

This shows that the geometric multiplicity of the 0 eigenvalue is equal to 2, which means that the system is marginally stable. The solution of this system is

$$\begin{aligned} x(t) &= \exp(At)x(0) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-t} \end{bmatrix} x(0) \end{aligned} \tag{1.130}$$

Regardless of $x(0)$, we see that $x(t)$ will always be bounded, which means that the system is marginally stable. Note that $x(t)$ may approach 0 as t increases, depending on the value of $x(0)$. For example, if $x(0) = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$, then

$$x(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-t} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -e^{-t} \end{bmatrix} \tag{1.131}$$

and $x(t)$ approaches 0 as t increases. However, this does not say anything about the asymptotic stability of the system; it only says that there exists some $x(0)$ that results in state $x(t)$ that asymptotically approaches 0. If we instead choose $x(0) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$, then

$$x(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-t} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{1.132}$$

and $x(t)$ does not approach 0. This proves that the system is asymptotically unstable.

∇∇∇

1.6.2 Discrete-time systems

Consider the zero-input, linear, discrete-time, time-invariant system

$$\begin{aligned} x_{k+1} &= Fx_k \\ y_k &= Hx_k \end{aligned} \tag{1.133}$$

The definitions of marginal stability (also called Lyapunov stability) and asymptotic stability are analogous to the definitions for continuous-time systems that were given in Section 1.6.1.

Definition 3 *A linear discrete-time, time-invariant system is marginally stable if the state x_k is bounded for all k and for all bounded initial states x_0.*

Definition 4 *A linear discrete-time, time-invariant system is asymptotically stable if*

$$\lim_{k\to\infty} x_k = 0 \tag{1.134}$$

for all bounded initial states x_0.

Marginal stability and asymptotic stability are types of internal stability. This is because they deal with only the state of the system (i.e., the internal condition of the system) and do not consider the output of the system. More specific categories of internal stability (e.g., uniform stability and exponential stability) are given in some books on linear systems.

Since the solution of Equation (1.133) is given as

$$x_k = A^k x_0 \tag{1.135}$$

we can state the following theorems.

Theorem 5 *A linear discrete-time, time-invariant system is marginally stable if and only if*

$$\lim_{k\to\infty} A^k \leq M < \infty \tag{1.136}$$

for some constant matrix M. This is just a way of saying that the powers of A do not increase without bound.

Theorem 6 *A linear discrete-time, time-invariant system is asymptotically stable if and only if*

$$\lim_{k\to\infty} A^k = 0 \tag{1.137}$$

Now recall that $A^k = Q\hat{A}^k Q^{-1}$, where Q is a constant matrix containing the eigenvectors of A, and $\hat{A}$ is the Jordan form of A. The matrix $\hat{A}^k$ therefore contains terms like λ_i^k, $k\lambda_i^k$, $k^2\lambda_i^k$, and so on, where λ_i is an eigenvalue of A. The boundedness of A^k is therefore related to the eigenvalues of A as stated by the following theorems.

Theorem 7 *A linear discrete-time, time-invariant system is marginally stable if and only if one of the following conditions holds.*

1. *All of the eigenvalues of A have magnitude less than one.*

2. *All of the eigenvalues of A have magnitude less than or equal to one, and those with magnitude equal to one have a geometric multiplicity equal to their algebraic multiplicity. That is, the Jordan blocks that are associated with the eigenvalues that have magnitude equal to one are first order.*

Theorem 8 *A linear discrete-time, time-invariant system is asymptotically stable if and only if all of the eigenvalues of A have magnitude less than one.*

1.7 CONTROLLABILITY AND OBSERVABILITY

The concepts of controllability and observability are fundamental to modern control theory. These concepts define how well we can control a system (i.e., drive the state to a desired value) and how well we can observe a system (i.e., determine the initial conditions after measuring the outputs). These concepts are also important to some of the theoretical results related to optimal state estimation that we will encounter later in this book.

1.7.1 Controllability

The following definitions and theorems give rigorous definitions for controllability for linear systems in the both the continuous-time and discrete-time cases.

Definition 5 *A continuous-time system is controllable if for any initial state $x(0)$ and any final time $t > 0$ there exists a control that transfers the state to any desired value at time t.*

Definition 6 *A discrete-time system is controllable if for any initial state x_0 and some final time k there exists a control that transfers the state to any desired value at time k.*

Note the controllability definition in the continuous-time case is much more demanding than the definition in the discrete-time case. In the continuous-time case, the existence of a control is required for *any* final time. In the discrete-time case, the existence of a control is required for *some* final time. In both cases, controllability is independent of the output equation.

There are several tests for controllability. The following equivalent theorems can be used to test for the controllability of continuous linear time-invariant systems.

Theorem 9 *The n-state[6] continuous linear time-invariant system $\dot{x} = Ax + Bu$ has the controllability matrix P defined by*

$$P = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} \tag{1.138}$$

The system is controllable if and only if $\rho(P) = n$.

Theorem 10 *The n-state continuous linear time-invariant system $\dot{x} = Ax + Bu$ is controllable if and only if the controllability grammian defined by*

$$\int_0^t e^{A\tau} B B^T e^{A^T \tau} \, d\tau \tag{1.139}$$

[6]The notation *n-state system* indicates a system that has n elements in its state variable x.

is positive definite for some $t \in (0, \infty)$.

Theorem 11 *The n-state continuous linear time-invariant system* $\dot{x} = Ax + Bu$ *is controllable if and only if the differential Lyapunov equation*

$$\begin{aligned} W(0) &= 0 \\ \dot{W} &= WA^T + AW + BB^T \end{aligned} \tag{1.140}$$

has a positive definite solution $W(t)$ *for some* $t \in (0, \infty)$. *This is also called a Sylvester equation.*

Similar to the continuous-time case, the following equivalent theorems can be used to test for the controllability of discrete linear time-invariant systems.

Theorem 12 *The n-state discrete linear time-invariant system* $x_k = Fx_{k-1} + Gu_{k-1}$ *has the controllability matrix* P *defined by*

$$P = \begin{bmatrix} G & FG & \cdots & F^{n-1}G \end{bmatrix} \tag{1.141}$$

The system is controllable if and only if $\rho(P) = n$.

Theorem 13 *The n-state discrete linear time-invariant system* $x_k = Fx_{k-1} + Gu_{k-1}$ *is controllable if and only if the controllability grammian defined by*

$$\sum_{i=0}^{k} A^{k-i} BB^T (A^T)^{k-i} \tag{1.142}$$

is positive definite for some $k \in (0, \infty)$.

Theorem 14 *The n-state discrete linear time-invariant system* $x_k = Fx_{k-1} + Gu_{k-1}$ *is controllable if and only if the difference Lyapunov equation*

$$\begin{aligned} W_0 &= 0 \\ W_{i+1} &= FW_iF^T + GG^T \end{aligned} \tag{1.143}$$

has a positive definite solution W_k *for some* $k \in (0, \infty)$. *This is also called a Stein equation.*

Note that Theorems 9 and 12 give identical tests for controllability for both continuous-time and discrete-time systems. In general, these are the simplest controllability tests. Controllability tests for time-varying linear systems can be obtained by generalizing the above theorems. Controllability for nonlinear systems is much more difficult to formalize.

■ EXAMPLE 1.8

The RLC circuit of Figure 1.4 has the system description

$$\begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} -2/RC & 1/C \\ -1/L & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 1/RC \\ 1/L \end{bmatrix} u \tag{1.144}$$

where v_C is the voltage across the capacitor, i_L is the current through the inductor, and u is the applied voltage. We will use Theorem 9 to determine the conditions under which this system is controllable. The controllability matrix is computed as

$$\begin{aligned} P &= \begin{bmatrix} B & AB \end{bmatrix} \\ &= \begin{bmatrix} 1/RC & 1/LC - 2/R^2C^2 \\ 1/L & -1/RLC \end{bmatrix} \end{aligned} \tag{1.145}$$

From this we can compute the determinant of P as

$$|P| = 1/R^2LC^2 - 1/L^2C \tag{1.146}$$

The determinant of P is 0 only if $R = \sqrt{L/C}$. So the system is controllable unless $R = \sqrt{L/C}$. It would be very difficult to obtain this result from Theorems 10 and 11.

Figure 1.4 RLC circuit for Example 1.8.

∇∇∇

1.7.2 Observability

The following definitions and theorems give rigorous definitions for observability for linear systems in both the continuous-time and discrete-time cases.

Definition 7 *A continuous-time system is observable if for any initial state $x(0)$ and any final time $t > 0$ the initial state $x(0)$ can be uniquely determined by knowledge of the input $u(\tau)$ and output $y(\tau)$ for all $\tau \in [0, t]$.*

Definition 8 *A discrete-time system is observable if for any initial state x_0 and some final time k the initial state x_0 can be uniquely determined by knowledge of the input u_i and output y_i for all $i \in [0, k]$.*

Note the observability definition in the continuous-time case is much more demanding than the definition in the discrete-time case. In the continuous-time case, the initial state must be able to be determined at *any* final time. In the discrete-time case, the initial state must be able to be determined at *some* final time. If a system is observable then the initial state can be determined, and if the initial state can be determined then all states between the initial and final times can be determined.

There are several tests for controllability. The following equivalent theorems can be used to test for the controllability of continuous linear time-invariant systems.

Theorem 15 *The n-state continuous linear time-invariant system*

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{1.147}$$

has the observability matrix Q defined by

$$Q = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{1.148}$$

The system is observable if and only if $\rho(Q) = n$.

Theorem 16 *The n-state continuous linear time-invariant system*

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{1.149}$$

is observable if and only if the observability grammian defined by

$$\int_0^t e^{A^T\tau} C^T C e^{A\tau}\, d\tau \tag{1.150}$$

is positive definite for some $t \in (0, \infty)$.

Theorem 17 *The n-state continuous linear time-invariant system*

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{1.151}$$

is observable if and only if the differential Lyapunov equation

$$\begin{aligned} W(t) &= 0 \\ -\dot{W} &= WA + A^T W + C^T C \end{aligned} \tag{1.152}$$

has a positive definite solution $W(\tau)$ *for some* $\tau \in (0, t)$. *This is also called a Sylvester equation.*

Similar to the continuous-time case, the following equivalent theorems can be used to test for the observability of discrete linear time-invariant systems.

Theorem 18 *The n-state discrete linear time-invariant system*

$$\begin{aligned} x_k &= Fx_{k-1} + Gu_{k-1} \\ y_k &= Hx_k \end{aligned} \tag{1.153}$$

has the observability matrix Q defined by

$$Q = \begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n-1} \end{bmatrix} \tag{1.154}$$

The system is observable if and only if $\rho(Q) = n$.

Theorem 19 *The n-state discrete linear time-invariant system*

$$\begin{aligned} x_k &= Fx_{k-1} + Gu_{k-1} \\ y_k &= Hx_k \end{aligned} \tag{1.155}$$

is observable if and only if the observability grammian defined by

$$\sum_{i=0}^{k} (F^T)^i H^T H F^i \tag{1.156}$$

is positive definite for some $k \in (0, \infty)$.

Theorem 20 *The n-state discrete linear time-invariant system*

$$\begin{aligned} x_k &= Fx_{k-1} + Gu_{k-1} \\ y_k &= Hx_k \end{aligned} \tag{1.157}$$

is observable if and only if the difference Lyapunov equation

$$\begin{aligned} W_k &= 0 \\ W_i &= F^T W_{i+1} F + H^T H \end{aligned} \tag{1.158}$$

has a positive definite solution W_0 for some $k \in (0, \infty)$. This is also called a Stein equation.

Note that Theorems 15 and 18 give identical tests for observability for both continuous-time and discrete-time systems. In general, these are the simplest observability tests. Observability tests for time-varying linear systems can be obtained by generalizing the above theorems. Observability for nonlinear systems is much more difficult to formalize.

■ **EXAMPLE 1.9**

The RLC circuit of Example 1.8 has the system description

$$\begin{aligned} \begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix} &= \begin{bmatrix} -2/RC & 1/C \\ -1/L & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 1/RC \\ 1/L \end{bmatrix} u \\ y &= \begin{bmatrix} -1 & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} \end{aligned} \tag{1.159}$$

where v_C is the voltage across the capacitor, i_L is the current through the inductor, and u is the applied voltage. We will use Theorem 15 to determine

the conditions under which this system is observable. The observability matrix is computed as

$$Q = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 2/RC & -1/C \end{bmatrix} \tag{1.160}$$

The determinant of the observability matrix can be computed as

$$|Q| = 1/C \tag{1.161}$$

The determinant of Q is nonzero, so the system is observable. On the other hand, suppose that $R = L = C = 1$ and the output equation is

$$y = \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} \tag{1.162}$$

Then the observability matrix can be computed as

$$\begin{aligned} Q &= \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \\ |Q| &= 0 \end{aligned} \tag{1.163}$$

So the system is unobservable. It would be very difficult to obtain this result from Theorems 16 and 17.

∇∇∇

1.7.3 Stabilizability and detectability

The concepts of stabilizability and detectability are closely related to controllability and observability, respectively. These concepts are also related to the modes of a system. The modes of a system are all of the decoupled states after the system is transformed into Jordan form. A system can be transformed into Jordan form as follows. Consider the system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{1.164}$$

First find the eigendata of the system matrix A. Suppose the eigenvectors are denoted as $v_1, \cdots, v_n$. Create an $n \times n$ matrix M by augmenting the eigenvectors as follows.

$$M = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix} \tag{1.165}$$

Define a new system as

$$\begin{aligned} \dot{\bar{x}} &= M^{-1}AM\bar{x} + M^{-1}B \\ &= \bar{A}\bar{x} + \bar{B}u \\ y &= CM\bar{x} + Du \\ &= \bar{C}\bar{x} + Du \end{aligned} \tag{1.166}$$

The new system is called the Jordan form representation of the original system. Note that the matrix M will always be invertible because the eigenvectors of a matrix can always be chosen to be linearly independent. The two systems of Equations (1.164) and (1.166) are called algebraically equivalent systems. This is because they have the same input and the same output (and therefore they have the same transfer function) but they have different states.

■ **EXAMPLE 1.10**

Consider the system

$$\begin{aligned}
\dot{x} &= Ax + Bu \\
&= \begin{bmatrix} 1 & 1 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & -2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} u \\
y &= Cx + Du \\
&= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} + 2u
\end{aligned} \tag{1.167}$$

This system has the same transfer function as

$$\begin{aligned}
\dot{\bar{x}} &= \bar{A}\bar{x} + \bar{B}u \\
&= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} u \\
y &= \bar{C}\bar{x} + Du \\
&= \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \bar{x} + 2u
\end{aligned} \tag{1.168}$$

The eigenvector matrix of A is

$$\begin{aligned}
M &= \begin{bmatrix} v_1 & v_2 & v_n \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & -3 \end{bmatrix}
\end{aligned} \tag{1.169}$$

Note the equivalences

$$\begin{aligned}
\bar{A} &= M^{-1}AM \\
\bar{B} &= M^{-1}B \\
\bar{C} &= CM
\end{aligned} \tag{1.170}$$

The Jordan form system has two decoupled modes. The first mode is

$$\begin{aligned}
\dot{\bar{x}}_1 &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \bar{x}_1 + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u \\
y_1 &= \begin{bmatrix} 1 & 0 \end{bmatrix} \bar{x}_1
\end{aligned} \tag{1.171}$$

The second mode is

$$\begin{aligned}
\dot{\bar{x}}_2 &= -2\bar{x}_2 + 0\,u \\
y_2 &= \bar{x}_2
\end{aligned} \tag{1.172}$$

∇∇∇

Definition 9 *If a system is controllable or stable, then it is also stabilizable. If a system is uncontrollable or unstable, then it is stabilizable if its uncontrollable modes are stable.*

In Example 1.10, the first mode is unstable (both eigenvalues at $+1$) but controllable. The second mode is stable (eigenvalue at -2) but uncontrollable. Therefore, the system is stabilizable.

Definition 10 *If a system is observable or stable, then it is also detectable. If a system is unobservable or unstable, then it is detectable if its unobservable modes are stable.*

In Example 1.10, the first mode is unstable but observable. The second mode is both stable and observable. Therefore, the system is detectable.

Controllability and observability were introduced by Rudolph Kalman at a conference in 1959 whose proceedings were published in an obscure Mexican journal in 1960 [Kal60b]. The material was also presented at an IFAC conference in 1960 [Kal60c], and finally published in a more widely accessible format in 1963 [Kal63].

1.8 SUMMARY

In this chapter we have reviewed some of the basic concepts of linear systems theory that are fundamental to many approaches to optimal state estimation. We began with a review of matrix algebra and matrix calculus, which proves to be indispensable in much of the theory of state estimation techniques. For additional information on matrix theory, the reader can refer to several excellent texts [Hor85, Gol89, Ber05]. We continued in this chapter with a review of linear and nonlinear systems, in both continuous time and discrete time. We regard time as continuous for physical systems, but our simulations and estimation algorithms operate in discrete time because of the popularity of digital computing. We discussed the discretization of continuous-time systems, which is a way of obtaining a discrete-time mathematical representation of a continuous-time system. The concept of stability can be used to tell us if a system's states will always remain bounded. Controllability tells us if it is possible to find a control input to force system states to our desired values, and observability tells us if it is possible to determine the initial state of a system on the basis of output measurements. State-space theory in general, and linear systems theory in particular, is a wide-ranging discipline that is typically covered in a one-semester graduate course, but there is easily enough material to fill up a two-semester course. Many excellent textbooks have been written on the subject, including [Bay99, Che99, Kai00] and others. A solid understanding of linear systems will provide a firm foundation for further studies in areas such as control theory, estimation theory, and signal processing.

PROBLEMS

Written exercises

1.1 Find the rank of the matrix $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$.

1.2 Find two 2×2 matrices A and B such that $A \neq B$, neither A nor B are diagonal, $A \neq cB$ for any scalar c, and $AB = BA$. Find the eigenvectors of A and

B. Note that they share an eigenvector. Interestingly, every pair of commuting matrices shares at least one eigenvector [Hor85, p. 51].

1.3 Prove the three identities of Equation (1.26).

1.4 Find the partial derivative of the trace of AB with respect to A.

1.5 Consider the matrix

$$A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

Recall that the eigenvalues of A are found by find the roots of the polynomial $P(\lambda) = |\lambda I - A|$. Show that $P(A) = 0$. (This is an illustration of the Cayley–Hamilton theorem [Bay99, Che99, Kai00].)

1.6 Suppose that A is invertible and

$$\begin{bmatrix} A & A \\ B & A \end{bmatrix} \begin{bmatrix} A \\ C \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

Find B and C in terms of A [Lie67].

1.7 Show that AB may not be symmetric even though both A and B are symmetric.

1.8 Consider the matrix

$$A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

where a, b, and c are real, and a and c are nonnegative.

a) Compute the solutions of the characteristic polynomial of A to prove that the eigenvalues of A are real.
b) For what values of b is A positive semidefinite?

1.9 Derive the properties of the state transition matrix given in Equation (1.72).

1.10 Suppose that the matrix A has eigenvalues λ_i and eigenvectors v_i ($i = 1, \cdots, n$). What are the eigenvalues and eigenvectors of $-A$?

1.11 Show that $|e^{At}| = e^{|A|t}$ for any square matrix A.

1.12 Show that if $\dot{A} = BA$, then

$$\frac{d|A|}{dt} = \mathrm{Tr}(B)|A|$$

1.13 The linear position p of an object under constant acceleration is

$$p = p_0 + \dot{p}t + \frac{1}{2}\ddot{p}t^2$$

where p_0 is the initial position of the object.

a) Define a state vector as $x = \begin{bmatrix} p & \dot{p} & \ddot{p} \end{bmatrix}^T$ and write the state space equation $\dot{x} = Ax$ for this system.
b) Use all three expressions in Equation (1.71) to find the state transition matrix e^{At} for the system.
c) Prove for the state transition matrix found above that $e^{A0} = I$.

1.14 Consider the following system matrix.

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Show that the matrix

$$S(t) = \begin{bmatrix} e^t & 0 \\ 0 & 2e^{-t} \end{bmatrix}$$

satisfies the relation $\dot{S}(t) = AS(t)$, but $S(t)$ is not the state transition matrix of the system.

1.15 Give an example of a discrete-time system that is marginally stable but not asymptotically stable.

1.16 Show (H, F) is an observable matrix pair if and only if (H, F^{-1}) is observable (assuming that F is nonsingular).

Computer exercises

1.17 The dynamics of a DC motor can be described as

$$J\ddot{\theta} + F\dot{\theta} = T$$

where θ is the angular position of the motor, J is the moment of inertia, F is the coefficient of viscous friction, and T is the torque applied to the motor.

a) Generate a two-state linear system equation for this motor in the form

$$\dot{x} = Ax + Bu$$

b) Simulate the system for 5 s and plot the angular position and velocity. Use $J = 10$ kg m^2, $F = 100$ kg m^2/s, $x(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$, and $T = 10$ N m. Use rectangular integration with a step size of 0.05 s. Do the output plots look correct? What happens when you change the step size Δt to 0.2 s? What happens when you change the step size to 0.5 s? What are the eigenvalues of the A matrix, and how can you relate their magnitudes to the step size that is required for a correct simulation?

1.18 The dynamic equations for a series RLC circuit can be written as

$$\begin{aligned} u &= IR + L\dot{I} + V_c \\ I &= C\dot{V}_c \end{aligned}$$

where u is the applied voltage, I is the current through the circuit, and V_c is the voltage across the capacitor.

a) Write a state-space equation in matrix form for this system with x_1 as the capacitor voltage and x_2 as the current.

b) Suppose that $R = 3$, $L = 1$, and $C = 0.5$. Find an analytical expression for the capacitor voltage for $t \geq 0$, assuming that the initial state is zero, and the input voltage is $u(t) = e^{-2t}$.

c) Simulate the system using rectangular, trapezoidal, and fourth-order Runge–Kutta integration to obtain a numerical solution for the capacitor voltage. Simulate from $t = 0$ to $t = 5$ using step sizes of 0.1 and 0.2. Tabulate the RMS value of the error between the numerical and analytical solutions for the capacitor voltage for each of your six simulations.

1.19 The vertical dimension of a hovering rocket can be modeled as

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{Ku - gx_2}{x_3} - \frac{GM}{(R + x_1)^2} \\
\dot{x}_3 &= -u
\end{aligned}$$

where x_1 is the vertical position of the rocket, x_2 is the vertical velocity, x_3 is the mass of the rocket, u is the control input (the flow rate of rocket propulsion), $K = 1000$ is the thrust constant of proportionality, $g = 50$ is the drag constant, $G = 6.673E - 11$ m^3/kg/s^2 is the universal gravitational constant, $M = 5.98E24$ kg is the mass of the earth, and $R = 6.37E6$ m is the radius of the earth radius.

a) Find $u(t) = u_0(t)$ such that the system is in equilibrium at $x_1(t) = 0$ and $x_2(t) = 0$.

b) Find $x_3(t)$ when $x_1(t) = 0$ and $x_2(t) = 0$.

c) Linearize the system around the state trajectory found above.

d) Simulate the nonlinear system for five seconds and the linearized system for five seconds with $u(t) = u_0(t) + \Delta u \cos(t)$. Plot the altitude of the rocket for the nonlinear simulation and the linear simulation (on the same plot) when $\Delta u = 10$. Repeat for $\Delta u = 100$ and $\Delta u = 300$. Hand in your source code and your three plots. What do you conclude about the accuracy of your linearization?

CHAPTER 2

Probability theory

The *most* we can know is in terms of probabilities.

—Richard Feynman [Fey63, p. 6-11]

While writing my book [*Stochastic Processes*, first published in 1953] I had an argument with Feller. He asserted that everyone said "random variable" and I asserted that everyone said "chance variable." We obviously had to use the same name in our books, so we decided the issue by a stochastic procedure. That is, we tossed for it and he won.

—Joseph Doob [Sne97, p. 307]

Probabilities do not exist.

—Bruno de Finetti [deF74]

In our attempt to filter a signal, we will be trying to extract meaningful information from a noisy signal. In order to accomplish this, we need to know something about what noise is, some of its characteristics, and how it works. This chapter reviews probability theory. We begin by discussing the basic concept of probability in Section 2.1, and then move on to random variables (RVs) in Section 2.2. The chapter then continues with the following topics:

- An RV is a general case of the normal scalars that we are familiar with, and so just as we can apply a functional mapping to a number, we can also apply

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

a functional mapping to an RV. We discuss functions (transformations) of random variables in Section 2.3.

- Just as we can have vectors of numbers, we can also have vectors of RVs, and so we discuss groups of random variables and random vectors in Section 2.4.
- Just as we can have scalar functions of time, we can also have RVs that are functions of time, and so we discuss RVs that change with time (stochastic processes) in Section 2.5.
- Stochastic processes can be divided into two categories: white noise and colored noise, and we discuss these concepts in Section 2.6.

We conclude in Section 2.7 with a high-level discussion of how to write a computer simulation of a noise process.

This chapter is only a brief introduction and review of probability and stochastic processes, and more detail can be found in many other books on the subject, such as [Pap02, Pee01].

2.1 PROBABILITY

How shall we define the concept of probability? Suppose we run an experiment a certain number of times. Sometimes event A occurs and sometimes it does not. For instance, our experiment may be rolling a six-sided die. Event A may be defined as the number 4 showing up on the top surface of the die after we roll the die. Common sense tells us that the probability of event A occuring is $1/6$. Likewise, we would expect that if we run our experiment many times, then we would see the number 1 appearing about $1/6$ of the time. This intuitive explanation forms the basis for our formal description of the concept of probability. We define the probability of event A as

$$P(A) = \frac{\text{Number of times } A \text{ occurs}}{\text{Total number of outcomes}} \tag{2.1}$$

This commonsense understanding of probability is called the relative frequency definition. A more formal and mathematically rigorous definition of probability can be obtained using set theory [Bil95, Nel87], which was pioneered by Andrey Kolomogorov in the 1930s. But for our purposes, the relative frequency definition is adequate.

In general, we know that there are n-choose-k different ways of selecting k objects from a total of n objects (assuming that the order of the objects does not matter), where n-choose-k is denoted and defined as

$$\binom{n}{k} = \frac{n!}{(n-k)!k!} \tag{2.2}$$

For instance, suppose we have a penny (P), nickel (N), dime (D), and quarter (Q). How many distinct subsets of three coins can we pick from that set? We can pick PND, PNQ, PDQ, or NDQ, for a total of four possible subsets. This is equal to 4-choose-3.

■ EXAMPLE 2.1

What is the probability of being dealt four of a kind[1] in poker? The total number of possible poker hands can be computed as the total number of subsets of size five that can be picked from a deck of 52 cards. The total number of possible hands is 52-choose-5 = 2,598,960. Out of all those hands, there are 48 possible hands containing four aces, 48 possible hands containing four kings, and so on. So there are a total of 13×48 hands containing four of a kind. Therefore the probability of being dealt four of a kind is

$$\begin{aligned} P(A) &= \frac{(13)(48)}{2,598,960} \\ &= 1/4165 \\ &\approx 0.024\% \end{aligned} \tag{2.3}$$

∇∇∇

The conditional probability of event A given event B can be defined if the probability of B is nonzero. The conditional probability of A given B is defined as

$$P(A|B) = \frac{P(A,B)}{P(B)} \tag{2.4}$$

$P(A|B)$ is the conditional probability of A given B, that is, the probability that A occurs given the fact that B occurred. $P(A,B)$ is the joint probability of A and B, that is, the probability that events A and B both occur. The probability of a single event [for instance, $P(A)$ or $P(B)$] is called an *a priori* probability because it applies to the probability of an event apart from any previously known information. A conditional probability [for instance, $P(A|B)$] is called an *a posteriori* probability because it applies to a probability given the fact that some information about a possibly related event is already known.

For example, suppose that A is the appearance of a 4 on a die, and B is the appearance of an even number on a die. $P(A) = 1/6$. But if we know that the die has an even number on it, then $\mathrm{P}(A) = 1/3$ (since the even number could be either a 2, 4, or 6). This example is intuitive, but we can also obtain the answer using Equation (2.4). $P(A,B)$ is the probability that both A occurs (we roll a 4) and B occurs (we roll an even number), so $P(A,B) = 1/6$. So Equation (2.4) gives

$$\begin{aligned} P(A|B) &= \frac{1/6}{1/2} \\ &= 1/3 \end{aligned} \tag{2.5}$$

The *a priori* probability of A is $1/6$. But the *a posteriori* probability of A given B is $1/3$.

■ EXAMPLE 2.2

Cnsider the eight shapes in Figure 2.1. We have three circles and five squares, so $P(\text{circle}) = 3/8$. Only one of the shapes is a gray circle, so $P(\text{gray, circle})$

[1]Once I was dealt four sevens while playing poker with some friends (unfortunately, I was not playing for money at the time). I don't expect to see it again in my lifetime.

$= 1/8$. Of the three circles, only one is gray, so $P(\text{gray} \mid \text{circle}) = 1/3$. This last probability can be computed using Equation (2.4) as

$$\begin{aligned} P(\text{gray}|\text{circle}) &= \frac{P(\text{gray}, \text{circle})}{P(\text{circle})} \\ &= \frac{1/8}{3/8} \\ &= 1/3 \end{aligned} \tag{2.6}$$

Figure 2.1 Some shapes for illustrating probability and Bayes' Rule.

$\nabla\nabla\nabla$

Note that we can use Equation (2.4) to write $P(B|A) = P(A,B)/P(A)$. We can solve both this equation and Equation (2.4) for $P(A,B)$ and equate the two expressions for $P(A,B)$ to obtain Bayes' Rule.

$$P(A|B)P(B) = P(B|A)P(A) \tag{2.7}$$

Bayes' Rule is often written by rearranging the above equation to obtain

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.8}$$

As an example, consider Figure 2.1. The probability of picking a gray shape given the fact that the shape is a circle can be computed from Bayes' Rule as

$$\begin{aligned} P(\text{gray}|\text{circle}) &= \frac{P(\text{circle}|\text{gray})P(\text{gray})}{P(\text{circle})} \\ &= \frac{(1/5)(5/8)}{3/8} \\ &= 1/3 \end{aligned} \tag{2.9}$$

We say that two events are independent if the occurrence of one event has no effect on the probability of the occurrence of the other event. For example, if A is the appearance of a 4 after rolling a die, and B is the appearance of a 3 after rolling *another* die, then A and B are independent. Mathematically, independence of A and B can be expressed several different ways. For example, we can write

$$\begin{aligned} P(A,B) &= P(A)P(B) \\ P(A|B) &= P(A) \\ P(B|A) &= P(B) \end{aligned} \tag{2.10}$$

if A and B are independent. As an example, recall from Equation (2.5) that if A is the appearance of a 4 on a die, and B is the appearance of an even number on a die, then $\mathrm{P}(A) = 1/6$ and $\mathrm{P}(A|B) = 1/3$. Since $P(A|B) \neq P(A)$ we see that A and B are dependent events.

2.2 RANDOM VARIABLES

We define a random variable (RV) as a functional mapping from a set of experimental outcomes (the domain) to a set of real numbers (the range). For example, the roll of a die can be viewed as a RV if we map the appearance of one dot on the die to the output one, the appearance of two dots on the die to the output two, and so on.

Of course, *after* we throw the die, the value of the die is no longer a random variable – it becomes certain. The outcome of a particular experiment is not an RV. If we define X as an RV that represents the roll of a die, then the probability that X will be a four is equal to $1/6$. If we then roll a four, the four is a realization of the RV X. If we then roll the die again and get a three, the three is another realization of the RV X. However, the RV X exists independently of any of its realizations. This distinction between an RV and its realizations is important for understanding the concept of probability. Realizations of an RV are not equal to the RV itself. When we say that the probability of $X = 4$ is equal to $1/6$, that means that there is a 1 out of 6 chance that each realization of X will be equal to 4. However, the RV X will always be random and will never be equal to a specific value.

An RV can be either continuous or discrete. The throw of a die is a discrete random variable because its realizations belong to a discrete set of values. The high temperature tomorrow is a continuous random variable because its realizations belong to a continuous set of values.

The most fundamental property of an RV X is its probability distribution function (PDF) $F_X(x)$, defined as

$$F_X(x) = P(X \leq x) \tag{2.11}$$

In the above equation, $F_X(x)$ is the PDF of the RV X, and x is a nonrandom independent variable or constant. Some properties of the PDF that can be obtained from its definition are

$$\begin{aligned} F_X(x) &\in [0,1] \\ F_X(-\infty) &= 0 \\ F_X(\infty) &= 1 \\ F_X(a) &\leq F_X(b) \quad \text{if } a \leq b \\ P(a < X \leq b) &= F_X(b) - F_X(a) \end{aligned} \tag{2.12}$$

The probability density function (pdf) $f_X(x)$ is defined as the derivative of the PDF.

$$f_X(x) = \frac{dF_X(x)}{dx} \tag{2.13}$$

Some properties of the pdf that can be obtained from this definition are

$$\begin{aligned} F_X(x) &= \int_{-\infty}^{x} f_X(z)\,dz \\ f_X(x) &\geq 0 \\ \int_{-\infty}^{\infty} f_X(x)\,dx &= 1 \\ P(a < x \leq b) &= \int_{a}^{b} f_X(x)\,dx \end{aligned} \tag{2.14}$$

The Q-function of an RV is defined as one minus the PDF. This is equal to the probability that the RV is greater than the argument of the function:

$$\begin{aligned} Q(x) &= 1 - F_X(x) \\ &= P(X > x) \end{aligned} \tag{2.15}$$

Just as we spoke about conditional probabilities in Equation (2.4), we can also speak about the conditional PDF and the conditional pdf. The conditional distribution and density of the RV X given the fact that event A occurred are defined as

$$\begin{aligned} F_X(x|A) &= P(X \leq x|A) \\ &= \frac{P(X \leq x, A)}{P(A)} \\ f_X(x|A) &= \frac{dF_X(x|A)}{dx} \end{aligned} \tag{2.16}$$

Bayes' Rule, discussed in Section 2.1, can be generalized to conditional densities. Suppose we have random variables X_1 and X_2. The conditional pdf of the RV X_1 given the fact that RV X_2 is equal to the realization x_2 is defined as

$$\begin{aligned} f_{X_1|X_2}(x_1|x_2) &= P[(X_1 \leq x_1)|(X_2 = x_2)] \\ &= \frac{f_{X_1,X_2}(x_1, x_2)}{f_{X_2}(x_2)} \end{aligned} \tag{2.17}$$

Although this is not entirely intuitive, it can be derived without too much difficulty [Pap02, Pee01]. Now consider the following product of two conditional pdf's:

$$\begin{aligned} f[x_1|(x_2, x_3, x_4)]f[(x_2, x_3)|x_4] &= \frac{f(x_1, x_2, x_3, x_4)}{f(x_2, x_3, x_4)} \frac{f(x_2, x_3, x_4)}{f(x_4)} \\ &= \frac{f(x_1, x_2, x_3, x_4)}{f(x_4)} \\ &= f[(x_1, x_2, x_3)|x_4] \end{aligned} \tag{2.18}$$

Note that in the above equation we have dropped the subscripts on the $f(\cdot)$ functions for ease of notation. This is commonly done if the random variable associated with the pdf is clear from the context. This is called the Chapman–Kolmogorov equation [Pap02]. It can be extended to any number of RVs and is fundamental to the Bayesian approach to state estimation (Chapter 15).

The expected value of an RV X is defined as its average value over a large number of experiments. This can also be called the expectation, the mean, or the average of

the RV. Suppose we run the experiment N times and observe a total of m different outcomes. We observe that outcome A_1 occurs n_1 times, A_2 occurs n_2 times, $\cdots$, and A_m occurs n_m times. Then the expected value of X is computed as

$$E(X) = \frac{1}{N}\sum_{i=1}^{m} A_i n_i \tag{2.19}$$

$E(X)$ is also often written as $E(x)$, $\bar{X}$, or $\bar{x}$.

At this point, we will begin to use lowercase x instead of uppercase X when the meaning is clear. We have been using uppercase X to refer to an RV, and lowercase x to refer to a realization of the RV, which is a constant or independent variable. However, it should be clear that, for example, $E(x)$ is the expected value of the RV X, and so we will interchange x and X in order to simplify notation.

As an example of the expected value of an RV, suppose that we roll a die an infinite number of times. We would expect to see each possible number (one through six) 1/6 of the time each. We can compute the expected value of the roll of the die as

$$\begin{aligned} E(X) &= \lim_{N\to\infty} \frac{1}{N}\left[(1)(N/6) + \cdots + (6)(N/6)\right] \\ &= 3.5 \end{aligned} \tag{2.20}$$

Note that the expected value of an RV is not necessarily what we would expect to see when we run a particular experiment. For example, even though the above expected value of X is 3.5, we will never see a 3.5 when we roll a die.

We can also talk about a function of an RV, just as we can talk about a function of any scalar. (We will discuss this in more detail in Section 2.3.) If a function, say $g(X)$, acts upon an RV, then the output of the function is also an RV. For example, if X is the roll of a die, then $P(X = 4) = 1/6$. If $g(X) = X^2$, then $P[g(X) = 16] = 1/6$. We can compute the expected value of any function $g(X)$ as

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x)\, dx \tag{2.21}$$

where $f_X(x)$ is the pdf of X. If $g(X) = X$, then we can compute the expected value of X as

$$E(X) = \int_{-\infty}^{\infty} x f_X(x)\, dx \tag{2.22}$$

The variance of an RV is a measure of how much we expect the RV to vary from its mean. The variance is a measure of how much variability there is in an RV. In the extreme case, if the RV X always is equal to one value (for example, the die is loaded and we always get a 4 when we roll the die), then the variance of X is equal to 0. On the other extreme, if X can take on any value between $\pm\infty$ with equal probability, then the variance of X is equal to ∞. The variance of an RV is formally defined as

$$\begin{aligned} \sigma_X^2 &= E[(X - \bar{x})^2] \\ &= \int_{-\infty}^{\infty} (x - \bar{x})^2 f_X(x)\, dx \end{aligned} \tag{2.23}$$

The standard deviation of an RV is σ, which is the square root of the variance. Sometimes we denote the standard deviation as σ_X if we need to be explicit about the RV whose standard deviation we are discussing. Note that the variance can be written as

$$\begin{aligned} \sigma^2 &= E[X^2 - 2X\bar{x} + \bar{x}^2] \\ &= E(X^2) - 2\bar{x}^2 + \bar{x}^2 \\ &= E(X^2) - \bar{x}^2 \end{aligned} \tag{2.24}$$

We use the notation

$$X \sim (\bar{x}, \sigma^2) \tag{2.25}$$

to indicate that X is an RV with a mean of $\bar{x}$ and a variance of σ^2.

The skew of an RV is a measure of the asymmetry of the pdf around its mean. Skew is defined as

$$\text{skew} = E[(X - \bar{x})^3] \tag{2.26}$$

The skewness, also called the coefficient of skewness, is the skew normalized by the cube of the standard deviation:

$$\text{skewness} = \text{skew}/\sigma^3 \tag{2.27}$$

In general, the ith moment of a random variable X is the expected value of the ith power of X. The ith central moment of a random variable X is the expected value of the ith power of X minus its mean:

$$\begin{aligned} i\text{th moment of } X &= E(X^i) \\ i\text{th central moment of } X &= E[(X - \bar{x})^i] \end{aligned} \tag{2.28}$$

For example, the first moment of a random variable is equal to its mean. The first central moment of a random variable is always equal to 0. The second central moment of a random variable is equal to its variance.

An RV is called uniform if its pdf is a constant value between two limits. This indicates that the RV has an equally likely probability of obtaining any value between its limits, but a zero probability of obtaining a value outside of its limits:

$$f_X(x) = \begin{cases} \frac{1}{b-a} & x \in [a, b] \\ 0 & \text{otherwise} \end{cases} \tag{2.29}$$

Figure 2.2 shows the pdf of an RV that is uniformly distributed between ± 1. Note that the area of this curve is one (as is the area of all pdf's).

■ EXAMPLE 2.3

In this example we will find the mean and variance of an RV that is uniformly distributed between 1 and 3. The pdf of the RV is given as

$$f_X(x) = \begin{cases} 1/2 & x \in [1, 3] \\ 0 & \text{otherwise} \end{cases} \tag{2.30}$$

Figure 2.2 Probability density function of an RV uniformly distributed between ± 1.

The mean is computed as follows:

$$\begin{aligned} \bar{x} &= \int_{-\infty}^{\infty} x f_X(x)\, dx \\ &= \int_1^3 \frac{1}{2} x\, dx \\ &= 2 \end{aligned} \tag{2.31}$$

The variance is computed as follows:

$$\begin{aligned} \sigma_X^2 &= \int_{-\infty}^{\infty} \frac{1}{2}(x-\bar{x})^2 f(x)\, dx \\ &= \int_1^3 \frac{1}{2}(x-2)^2\, dx \\ &= \frac{1}{3} \end{aligned} \tag{2.32}$$

$\nabla\nabla\nabla$

An RV is called Gaussian or normal if its pdf is given by

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(x-\bar{x})^2}{2\sigma^2}\right] \tag{2.33}$$

This is called the Laplace distribution in France, but it had many other discoverers, including Robert Adrain. Note that $\bar{x}$ and σ in the above pdf are the mean and standard deviation of the Gaussian RV. We use the notation

$$X \sim N(\bar{x}, \sigma^2) \tag{2.34}$$

to indicate that X is a Gaussian RV with a mean of $\bar{x}$ and a variance of σ^2. Figure 2.3 shows the pdf of a Gaussian RV with a mean of zero and a variance

of one. If the mean changes, the pdf will shift to the left or right. If the variance increases, the pdf will spread out. If the variance decreases, the pdf will be squeezed in. The PDF of a Gaussian RV is given by

$$F_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp[-(z-\bar{x})^2/2\sigma^2]\, dz \tag{2.35}$$

This integral does not have a closed-form solution, and so it must be evaluated numerically. However, its evaluation can be simplified by considering the normalized Gaussian PDF of an RV with zero mean and unity variance:

$$F_{X0}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-z^2/2)\, dz \tag{2.36}$$

It can be shown that any Gaussian PDF can be expressed in terms of this normalized PDF as

$$F_X(x) = F_{X0}\left(\frac{x-\bar{x}}{\sigma}\right) \tag{2.37}$$

In addition, a Gaussian PDF can be approximated as the following closed-form expression [Bor79]:

$$\begin{aligned} F_X(x) &\approx 1 - \left[\frac{1}{(1-a)x + a\sqrt{x^2+b}}\right] \frac{\exp(-x^2/2)}{\sqrt{2\pi}} \qquad x \geq 0 \\ a &= 0.339 \\ b &= 5.510 \end{aligned} \tag{2.38}$$

Figure 2.3 Probability density function of a Gaussian RV with a mean of zero and a variance of one.

Suppose we have a random variable X with a mean of zero and a symmetric pdf [i.e., $f_X(x) = f_X(-x)$]. This is the case, for example, for the pdf's shown in

Figures 2.2 and 2.3. In this case, the ith moment of X can be written as

$$\begin{aligned} m_i &= E(X^i) \\ &= \int_{-\infty}^{\infty} x^i f_X(x)\,dx \\ &= \int_{-\infty}^{0} x^i f_X(x)\,dx + \int_{0}^{\infty} x^i f_X(x)\,dx \end{aligned} \tag{2.39}$$

If i is odd then $x^i = -(-x)^i$. Combined with the fact that $f_X(x) = f_X(-x)$, we see that

$$\begin{aligned} \int_{-\infty}^{0} x^i f_X(x)\,dx &= \int_{0}^{\infty} (-x)^i f_X(-x)\,dx \\ &= -\int_{0}^{\infty} x^i f_X(x)\,dx \end{aligned} \tag{2.40}$$

So for odd i, the ith moment in Equation (2.39) is zero. We see that all of the odd moments of a zero-mean random variable with a symmetric pdf are equal to 0.

2.3 TRANSFORMATIONS OF RANDOM VARIABLES

In this section, we will look at what happens to the pdf of an RV when we pass the RV through some function. Suppose that we have two RVs, X and Y, related to one another by the monotonic[2] functions $g(\cdot)$ and $h(\cdot)$:

$$\begin{aligned} Y &= g(X) \\ X &= g^{-1}(Y) = h(Y) \end{aligned} \tag{2.41}$$

If we know the pdf of X $[f_X(x)]$, then we can compute the pdf of Y $[f_Y(y)]$ as follows:

$$\begin{aligned} P(X \in [x, x+dx]) &= P(Y \in [y, y+dy]) \quad (dx > 0) \\ \int_{x}^{x+dx} f_X(z)dz &= \begin{cases} \int_{y}^{y+dy} f_Y(z)dz & \text{if } dy > 0 \\ -\int_{y}^{y+dy} f_Y(z)dz & \text{if } dy < 0 \end{cases} \\ f_X(x)dx &= f_Y(y)|dy| \\ f_Y(y) &= \left|\frac{dx}{dy}\right| f_X[h(y)] \\ &= |h'(y)| f_X[h(y)] \end{aligned} \tag{2.42}$$

where we have used the assumption of small dx and dy in the above calculation.

[2] A monotonic function is a function whose slope is either always nonnegative or always nonpositive. If the slope is always nonnegative, then the function is monotonically nondecreasing. If the slope is always positive, then the function is monotonically increasing. If the slope is always nonpositive, then the function is monotonically nonincreasing. If the slope is always negative, then the function is monotonically decreasing.

■ **EXAMPLE 2.4**

In this example, we will find the pdf of a linear function of a Gaussian RV. Suppose that $X \sim N(\bar{x}, \sigma_x^2)$ and $Y = g(X) = aX + b$, where $a \neq 0$ and b are any real constants. Then

$$\begin{aligned} X &= h(Y) \\ &= (Y - b)/a \\ h'(y) &= 1/a \\ f_Y(y) &= |h'(y)| f_X[h(y)] \\ &= \left|\frac{1}{a}\right| \frac{1}{\sigma_X\sqrt{2\pi}} \exp\left\{\frac{-[(y-b)/a - \bar{x}]^2}{2\sigma_X^2}\right\} \\ &= \frac{1}{a\sigma_X\sqrt{2\pi}} \exp\left\{\frac{-[y - (a\bar{x} + b)]^2}{2a^2\sigma_X^2}\right\} \end{aligned} \tag{2.43}$$

In other words, the RV Y is Gaussian with a mean and variance given by

$$\begin{aligned} \bar{y} &= a\bar{x} + b \\ \sigma_Y^2 &= a^2\sigma_X^2 \end{aligned} \tag{2.44}$$

This important example shows that a linear transformation of a Gaussian RV results in a new Gaussian RV.

∇∇∇

■ **EXAMPLE 2.5**

Suppose that we pass a Gaussian RV $X \sim N(0, \sigma_x^2)$ through the nonlinear function $Y = g(X) = X^3$:

$$\begin{aligned} X &= h(Y) \\ &= Y^{1/3} \\ h'(y) &= \frac{y^{-2/3}}{3} \\ f_Y(y) &= |h'(y)| f_X[h(y)] \\ &= \frac{y^{-2/3}}{3} \frac{1}{\sigma_x\sqrt{2\pi}} \exp[-x^2/(2\sigma_x^2)] \\ &= \frac{y^{-2/3}}{3} \frac{1}{\sigma_x\sqrt{2\pi}} \exp[-y^{2/3}/(2\sigma_x^2)] \end{aligned} \tag{2.45}$$

We see that the nonlinear transformation $Y = X^3$ converts a Gaussian RV to a non-Gaussian RV. It can be seen that $f_Y(y)$ approaches ∞ as $y \to 0$. Nevertheless, the area under the $f_Y(y)$ curve is equal to 1 since it is a pdf.

∇∇∇

In the more general case of RVs related by the function $Y = g(X)$, where $g(\cdot)$ is a nonmonotonic function, the pdf of Y (evaluated at y) can be computed from the pdf of X as

$$f_Y(y) = \sum_i f_X(x_i)/|g'(x_i)| \tag{2.46}$$

where the x_i values are the solutions of the equation $y = g(x)$.

2.4 MULTIPLE RANDOM VARIABLES

We have already defined the probability distribution function of an RV. For example, if X and Y are RVs, then their distribution functions are defined as

$$\begin{aligned} F_X(x) &= P(X \leq x) \\ F_Y(y) &= P(Y \leq y) \end{aligned} \tag{2.47}$$

Now we define the probability that both $X \leq x$ and $Y \leq y$ as the joint probability distribution function of X and Y:

$$F_{XY}(x, y) = P(X \leq x, Y \leq y) \tag{2.48}$$

If the meaning is clear from the context, we often use the shorthand notation $F(x, y)$ to represent the distribution function $F_{XY}(x, y)$. Some properties of the joint distribution function are

$$\begin{aligned} F(x, y) &\in [0, 1] \\ F(x, -\infty) = F(-\infty, y) &= 0 \\ F(\infty, \infty) &= 1 \\ F(a, c) &\leq F(b, d) \quad \text{if } a \leq b \text{ and } c \leq d \\ P(a < x \leq b, c < y \leq d) &= F(b, d) + F(a, c) - F(a, d) - F(b, c) \\ F(x, \infty) &= F(x) \\ F(\infty, y) &= F(y) \end{aligned} \tag{2.49}$$

Note from the last two properties that the distribution function of one RV can be obtained from the joint distribution function. When the distribution function for a single RV is obtained this way it is called the marginal distribution function.

The joint probability density function is defined as the following derivative of the joint PDF:

$$f_{XY}(x, y) = \frac{\partial^2 F_{XY}(x, y)}{\partial x \partial y} \tag{2.50}$$

As before, we often use the shorthand notation $f(x, y)$ to represent the density function $f_{XY}(x, y)$. Some properties of the joint pdf that can be obtained from this definition are

$$\begin{aligned} F(x, y) &= \int_{-\infty}^{x} \int_{-\infty}^{y} f(z_1, z_1)\, dz_1\, dz_2 \\ f(x, y) &\geq 0 \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\, dx\, dy &= 1 \\ P(a < x \leq b, c < y \leq d) &= \int_{c}^{d} \int_{a}^{b} f(x, y)\, dx\, dy \\ f(x) &= \int_{-\infty}^{\infty} f(x, y)\, dy \\ f(y) &= \int_{-\infty}^{\infty} f(x, y)\, dx \end{aligned} \tag{2.51}$$

Note from the last two properties that the density function of one RV can be obtained from the joint density function. When the density function for a single RV is obtained this way it is called the marginal density function. Computing the expected value of a function $g(\cdot,\cdot)$ of two RVs is similar to computing the expected value of a function of a single RV:

$$E[g(x,y)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)f(x,y)\,dx\,dy \tag{2.52}$$

2.4.1 Statistical independence

Recall from Section 2.1 that two events are independent if the occurrence of one event has no effect on the probability of the occurrence of the other event. We extend this to say that RVs X and Y are independent if they satisfy the following relation:

$$P(X \le x, Y \le y) = P(X \le x)P(Y \le y) \qquad \text{for all } x,\, y \tag{2.53}$$

From our definition of joint distribution and density functions, we see that this implies

$$\begin{aligned} F_{XY}(x,y) &= F_X(x)F_Y(y) \\ f_{XY}(x,y) &= f_X(x)f_Y(y) \end{aligned} \tag{2.54}$$

The central limit theorem says that the sum of independent RVs tends toward a Gaussian RV, regardless of the pdf of the individual RVs that contribute to the sum. This is why so many RVs in nature seem to have a Gaussian distribution. Many RVs in nature are actually the sum of many individual and independent RVs. For example, the high temperature on any given day in any given location tends to follow a Gaussian distribution. This is because the high temperature is affected by clouds, precipitation, wind, air pressure, humidity, and other factors. Each of these factors is in turn determined by other random factors. The combination of many independent random variables determines the high temperature, which has a Gaussian pdf.

We define the covariance of two scalar RVs X and Y as

$$\begin{aligned} C_{XY} &= E[(X-\bar{X})(Y-\bar{Y}] \\ &= E(XY) - \bar{X}\bar{Y} \end{aligned} \tag{2.55}$$

We define the correlation coefficient of two scalar RVs X and Y as

$$\rho = \frac{C_{XY}}{\sigma_x \sigma_y} \tag{2.56}$$

The correlation coefficient is a normalized measurement of the independence between two RVs X and Y. If X and Y are independent, then $\rho = 0$ (although the converse is not necessarily true). If Y is a linear function of X then $\rho = \pm 1$ (see Problem 2.9).

We define the correlation of two scalar RVs X and Y as

$$R_{XY} = E(XY) \tag{2.57}$$

Two RVs are said to be uncorrelated if $R_{XY} = E(X)E(Y)$.

From the definition of independence, we see that if two RVs are independent then they are also uncorrelated. Independence implies uncorrelatedness, but uncorrelatedness does not necessarily imply independence. However, in the special case in which two RVs are both Gaussian and uncorrelated, then it follows that they are also independent.

Two RVs are said to be orthogonal if $R_{XY} = 0$. If two RVs are uncorrelated, then they are orthogonal only if at least one of them is zero-mean. If two RVs are orthogonal, then they may or may not be uncorrelated.

■ EXAMPLE 2.6

Two rolls of the dice are represented by the RVs X and Y. The two RVs are independent because one roll of the die does not have any effect on a second roll of the die. Each roll of the die has an equally likely probability (1/6) of being a 1, 2, 3, 4, 5, or 6. Therefore,

$$\begin{aligned} E(X) = E(Y) &= \frac{1+2+3+4+5+6}{6} \\ &= 3.5 \end{aligned} \tag{2.58}$$

There are 36 possible combinations of the two rolls of the die. We could get the combination (1,1), (1,2), and so on. Each of these 36 combinations have an equally likely probability (1/36). Therefore, the correlation between X and Y is

$$\begin{aligned} R_{XY} = E(XY) &= \frac{1}{36}\sum_{i=1}^{6}\sum_{j=1}^{6} ij \\ &= 12.25 \\ &= E(X)E(Y) \end{aligned} \tag{2.59}$$

Since $E(XY) = E(X)E(Y)$, we see that X and Y are uncorrelated. However, $R_{XY} \neq 0$, so X and Y are not orthogonal.

∇∇∇

■ EXAMPLE 2.7

A slot machine is rigged so you get 1 or -1 with equal probability the first spin X, and the opposite number the second spin Y. We have equal probabilities of obtaining (X, Y) outcomes of $(1, -1)$ and $(-1, 1)$. The two RVs are dependent because the realization of Y depends on the realization of X. We also see that

$$\begin{aligned} E(X) &= 0 \\ E(Y) &= 0 \\ E(XY) &= \frac{(1)(-1)+(-1)(1)}{2} \\ &= -1 \end{aligned} \tag{2.60}$$

We see that X and Y are correlated because $E(XY) \neq E(X)E(Y)$. We also see that X and Y are not orthogonal because $E(XY) \neq 0$.

∇∇∇

■ **EXAMPLE 2.8**

A slot machine is rigged so you get -1, 0, or $+1$ with equal probability the first spin X. On the second spin Y you get 1 if $X = 0$, and 0 if $X \neq 0$. The two RVs are dependent because the realization of Y depends on the realization of X. We also see that

$$\begin{aligned} E(X) &= \frac{-1+0+1}{3} \\ &= 0 \\ E(Y) &= \frac{0+1+0}{3} \\ &= 1/3 \\ E(XY) &= \frac{(-1)(0)+(0)(1)+(1)(0)}{3} \\ &= 0 \end{aligned} \tag{2.61}$$

We see that X and Y are uncorrelated because $E(XY) = E(X)E(Y)$. We also see that X and Y are orthogonal because $E(XY) = 0$. This example illustrates the fact that uncorrelatedness does not necessarily imply independence.

∇∇∇

■ **EXAMPLE 2.9**

Suppose that x and y are independent RVs, and the RV z is computed as $z = g(x) + h(y)$. In this example, we will calculate the mean of z:

$$\begin{aligned} E(z) &= E[g(x) + h(y)] \\ &= \int\int [g(x) + h(y)] f(x,y)\, dx\, dy \\ &= \int\int g(x)f(x)f(y)\, dx\, dy + \int\int h(y)f(x)f(y)\, dx\, dy \\ &= \int g(x)f(x)\, dx \int f(y)\, dy + \int h(y)f(y)\, dy \int f(x)\, dx \\ &= E[g(x)](1) + E[h(y)](1) \\ &= E[g(x)] + E[h(y)] \end{aligned} \tag{2.62}$$

As a special case of this example, we see that the mean of the sum of two independent RVs is equal to the sum of their means. That is,

$$E(x+y) = E(x) + E(y) \quad \text{if } x \text{ and } y \text{ are independent} \tag{2.63}$$

∇∇∇

■ **EXAMPLE 2.10**

Suppose we roll a die twice. What is the expected value of the sum of the two outcomes? We use X and Y to refer to the two rolls of the die, and we use

Z to refer to the sum of the two outcomes. Therefore, $Z = X + Y$. Since X and Y are independent, we have

$$\begin{aligned} E(Z) &= E(X) + E(Y) \\ &= 3.5 + 3.5 \\ &= 7 \end{aligned} \tag{2.64}$$

∇∇∇

■ **EXAMPLE 2.11**

Consider the circuit of Figure 2.4. The input voltage V is uniformly distributed on $[-1, 1]$. Voltage V has units of volts, and the two currents have units of amps.

$$\begin{aligned} I_1 &= \begin{cases} 0 & \text{if } V > 0 \\ V & \text{if } V \leq 0 \end{cases} \\ I_2 &= \begin{cases} V & \text{if } V \geq 0 \\ 0 & \text{if } V < 0 \end{cases} \end{aligned} \tag{2.65}$$

We see that I_1 is uniformly distributed on $[-1, 0]$ and I_2 is uniformly distributed on $[0, 1]$. The RVs V, I_1, and I_2 have expected values

$$\begin{aligned} E(V) &= 0 \\ E(I_1) &= -1/2 \\ E(I_2) &= 1/2 \end{aligned} \tag{2.66}$$

The RVs I_1 and I_2 are not independent because they are related to each other; if $I_2 \neq 0$ then $I_1 = 0$, and if $I_1 \neq 0$ then $I_2 = 0$. Since either I_1 or I_2 is equal to 0 at every time instant, $I_1 I_2 = 0$ and $E(I_1 I_2) = 0$. Therefore I_1 and I_2 are orthogonal. Since $E(I_1)E(I_2) = -1/4$, we see that $E(I_1 I_2) \neq E(I_1)E(I_2)$, and I_1 and I_2 are correlated.

Figure 2.4 Circuit for Example 2.11.

∇∇∇

2.4.2 Multivariate statistics

The discussion in the previous subsection can be generalized for RVs that are vectors. In this case, the quantities defined earlier become vectors and matrices. Given

an n-element RV X and an m-element RV Y (assuming that both X and Y are column vectors), their correlation is defined as

$$\begin{aligned} R_{XY} &= E(XY^T) \\ &= \begin{bmatrix} E(X_1Y_1) & \cdots & E(X_1Y_m) \\ \vdots & & \vdots \\ E(X_nY_1) & \cdots & E(X_nY_m) \end{bmatrix} \end{aligned} \tag{2.67}$$

Their covariance is defined as

$$\begin{aligned} C_{XY} &= E[(X-\bar{X})(Y-\bar{Y})^T] \\ &= E(XY^T) - \bar{X}\bar{Y}^T \end{aligned} \tag{2.68}$$

The autocorrelation of the n-element RV X is defined as

$$\begin{aligned} R_X &= E[XX^T] \\ &= \begin{bmatrix} E[X_1^2] & \cdots & E[X_1X_n] \\ \vdots & & \vdots \\ E[X_nX_1] & \cdots & E[X_n^2] \end{bmatrix} \end{aligned} \tag{2.69}$$

Note that $E(X_iX_j) = E(X_jX_i)$ so $R_X = R_X^T$. An autocorrelation matrix is always symmetric. Also note that for any n-element column vector z we have

$$\begin{aligned} z^TR_Xz &= z^TE[XX^T]z \\ &= E[z^TXX^Tz] \\ &= E[(z^TX)^2] \\ &\geq 0 \end{aligned} \tag{2.70}$$

So an autocorrelation matrix is always positive semidefinite.

The autocovariance of the n-element RV X is defined as

$$\begin{aligned} C_X &= E[(X-\bar{X})(X-\bar{X})^T] \\ &= \begin{bmatrix} E[(X_1-\bar{X}_1)^2] & \cdots & E[(X_1-\bar{X}_1)(X_n-\bar{X}_n)] \\ \vdots & & \vdots \\ E[(X_n-\bar{X}_n)(X_1-\bar{X}_1)] & \cdots & E[(X_n-\bar{X}_n)^2] \end{bmatrix} \\ &= \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{1n} \\ \vdots & & \vdots \\ \sigma_{n1} & \cdots & \sigma_n^2 \end{bmatrix} \end{aligned} \tag{2.71}$$

Note that $\sigma_{ij} = \sigma_{ji}$ so $C_X = C_X^T$. An autocovariance matrix is always symmetric. Also note that for any n-element column vector z we have

$$\begin{aligned} z^TC_Xz &= z^TE[(X-\bar{X})(X-\bar{X})^T]z \\ &= E[z^T(X-\bar{X})(X-\bar{X})^Tz] \\ &= E[(z^T(X-\bar{X}))^2] \\ &\geq 0 \end{aligned} \tag{2.72}$$

So an autocovariance matrix is always positive semidefinite.

An n-element RV X is Gaussian (normal)[3] if

$$\text{pdf}(X) = \frac{1}{(2\pi)^{n/2}|C_X|^{1/2}} \exp\left[\frac{-1}{2}(X-\bar{X})^T C_X^{-1}(X-\bar{X})\right] \tag{2.73}$$

Now consider a Gaussian RV X that undergoes a linear transformation:

$$\begin{aligned} Y &= g(X) \\ &= AX + b \end{aligned} \tag{2.74}$$

where A is a constant $n \times n$ matrix, and b is a constant n-element vector. If A is invertible, then

$$\begin{aligned} X &= h(Y) \\ &= A^{-1}Y - A^{-1}b \end{aligned} \tag{2.75}$$

From Equation (2.42) we obtain

$$\begin{aligned} f_Y(y) &= |h'(y)| f_X[h(y)] \\ &= |A^{-1}| \frac{1}{(2\pi)^{n/2}|C_X|^{1/2}} \exp\left[\frac{-1}{2}(A^{-1}y - A^{-1}b - \bar{x})^T C_X^{-1}(\cdots)\right] \\ &= |A^{-1}| \frac{1}{(2\pi)^{n/2}|C_X|^{1/2}} \times \\ &\quad \exp\left[\frac{-1}{2}(A^{-1}y - A^{-1}b - A^{-1}A\bar{x})^T C_X^{-1}(\cdots)\right] \\ &= \frac{1}{(2\pi)^{n/2}|A||C_X|^{1/2}} \times \\ &\quad \exp\left[\frac{-1}{2}(A^{-1}y - A^{-1}\bar{y})^T C_X^{-1}(A^{-1}y - A^{-1}\bar{y})\right] \\ &= \frac{1}{(2\pi)^{n/2}|A|^{1/2}|C_X|^{1/2}|A^T|^{1/2}} \times \\ &\quad \exp\left[\frac{-1}{2}(y - \bar{y})^T A^{-T} C_X^{-1} A^{-1}(y - \bar{y})\right] \\ &= \frac{1}{(2\pi)^{n/2}|AC_XA^T|^{1/2}} \exp\left[\frac{-1}{2}(y - \bar{y})^T (AC_XA^T)^{-1}(y - \bar{y})\right] \\ y &\sim N(A\bar{x} + b, AC_XA^T) \end{aligned} \tag{2.76}$$

This shows that normality is preserved in linear transformations of random vectors (just as it is preserved in linear transformations of random scalars, as seen in Example 2.4).

[3]Francis Edgeworth (1845-1926), an Irish economist and mathematician, first provided a general description and study of the multivariate Gaussian probability distribution in 1892 [Sor80].

2.5 STOCHASTIC PROCESSES

A stochastic process, also called a random process, is a very simple generalization of the concept of an RV. A stochastic process $X(t)$ is an RV X that changes with time.[4] A stochastic process can be one of four types.

- If the RV at each time is continuous and time is continuous, then $X(t)$ is a continuous random process. For example, the temperature at each moment of the day is a continuous random process because both temperature and time are continuous.
- If the RV at each time is discrete and time is continuous, then $X(t)$ is a discrete random process. For example, the number of people in a given building at each moment of the day is a discrete random process because the number of people is a discrete variable and time is continuous.
- If the RV at each time is continuous and time is discrete, then $X(t)$ is a continuous random sequence. For example, the high temperature each day is a continuous random sequence because temperature is continuous but time is discrete (day one, day two, etc.).
- If the RV at each time is discrete and time is discrete, then $X(t)$ is a discrete random sequence. For example, the highest number of people in a given building each day is a discrete random sequence because the number of people is a discrete variable and time is also discrete.

Since a stochastic process is an RV that changes with time, it has a distribution and density function that are functions of time. The PDF of $X(t)$ is

$$F_X(x,t) = P(X(t) \leq x) \tag{2.77}$$

If $X(t)$ is a random vector, then the inequality above is an element-by-element inequality. For example, if $X(t)$ has n elements, then

$$F_X(x,t) = P\left[X_1(t) \leq x_1 \text{ and } \cdots X_n(t) \leq x_n(t)\right] \tag{2.78}$$

The pdf of $X(t)$ is

$$f_X(x,t) = \frac{dF_X(x,t)}{dx} \tag{2.79}$$

If $X(t)$ is a random vector, then the derivative above is taken once with respect to each element of x. For example, if $X(t)$ has n elements, then

$$f_X(x,t) = \frac{d^n F_X(x,t)}{dx_1 \cdots dx_n} \tag{2.80}$$

The mean and covariance of $X(t)$ are also functions of time:

$$\bar{x}(t) = \int_{-\infty}^{\infty} x f(x,t)\, dx$$

[4]Actually, the independent variable does not have to be time; for example, it could be spatial location or something else. But typically the independent variable is time, and in this book it will always be time.

$$\begin{aligned} C_X(t) &= E\left\{[X(t)-\bar{x}(t)]\,[X(t)-\bar{x}(t)]^T\right\} \\ &= \int_{-\infty}^{\infty} [x-\bar{x}(t)]\,[x-\bar{x}(t)]^T f(x,t)\,dx \end{aligned} \tag{2.81}$$

Note that $X(t)$ at two different times (t_1 and t_2) comprise two different random variables [$X(t_1)$ and $X(t_2)$]. Therefore, we can talk about the joint distribution and joint density functions of $X(t_1)$ and $X(t_2)$. These are called the second-order distribution function and the second-order density function:

$$\begin{aligned} F(x_1,x_2,t_1,t_2) &= P(X(t_1)\le x_1, X(t_2)\le x_2) \\ f(x_1,x_2,t_1,t_2) &= \frac{\partial^2 F(x_1,x_2,t_1,t_2)}{\partial x_1 \partial x_2} \end{aligned} \tag{2.82}$$

As discussed earlier, if $X(t)$ is an n-element random vector, then the inequality that defines $F(x_1,x_2,t_1,t_2)$ actually consists of $2n$ inequalities, and the derivative that defines $f(x_1,x_2,t_1,t_2)$ actually consists of $2n$ derivatives.

The correlation between the two RVs $X(t_1)$ and $X(t_2)$ is called the autocorrelation of the stochastic process $X(t)$:

$$R_X(t_1,t_2) = E\left[X(t_1)X^T(t_2)\right] \tag{2.83}$$

The autocovariance of a stochastic process is defined as

$$C_X(t_1,t_2) = E\left\{\left[X(t_1)-\bar{X}(t_1)\right]\left[X(t_2)-\bar{X}(t_2)\right]^T\right\} \tag{2.84}$$

For some stochastic processes, the pdf does not change with time. For example, if we flip a coin ten times then we can view that process as a stochastic process with the statistics of the process being the same at each of the ten time instances. In this case, the stochastic process is called strict-sense stationary (SSS), or just stationary for short. In this case, the mean of the stochastic process is constant with respect to time, and the autocorrelation is a function of the time difference $t_2 - t_1$ (not a function of the absolute times):

$$\begin{aligned} E[X(t)] &= \bar{x} \\ E[X(t_1)X^T(t_2)] &= R_X(t_2-t_1) \end{aligned} \tag{2.85}$$

For some stochastic processes, these two conditions are true even though the pdf does change with time. Stochastic processes for which these two conditions are true are called wide-sense stationary (WSS). A stationary process is wide-sense stationary, but a wide-sense stationary process may or may not be stationary. From the definition of autocorrelation, it can be shown that for a wide-sense stationary process the following properties hold:

$$\begin{aligned} R_X(0) &= E[X(t)X^T(t)] \\ R_X(-\tau) &= R_X(\tau) \end{aligned} \tag{2.86}$$

For scalar stochastic processes, it can be shown that

$$|R_X(\tau)| \le R_X(0) \tag{2.87}$$

■ EXAMPLE 2.12

1. The high temperature each day can be considered a stochastic process. However, this process is not stationary. The high temperature on a day in July might be an RV with a mean of 100 degrees Fahrenheit, but the high temperature on a day in December might have a mean of 30 degrees. This is a stochastic process whose statistics change with time, so the process is not stationary.

2. Electrical noise in a voltmeter might have a mean of zero and a variance of one millivolt. If we come back the next day and measure the noise again, the mean and variance may be the same as before. If the statistics of the noise are the same every day, then the electrical noise is a stationary process. Note that in reality the noise statistics will eventually change. For example, after a few decades the instrument will begin degrading and the electrical noise mean and variance will change. In this sense, there is no such thing as a stationary random process. Eventually, the universe will freeze and all signals will change. But for practical purposes, if the statistics of a random process do not change over the time interval of interest, then we consider the process to be stationary.

3. Tomorrow's closing price of the Dow Jones Industrial Average might be an RV with a certain mean and variance. However, 100 years ago the closing price had a mean that was much lower. The closing price of the stock market is an RV whose mean generally increases with time. Therefore, the stock market price is a nonstationary stochastic process.

∇∇∇

Suppose we have a stochastic process $X(t)$. Further suppose that the process has a realization $x(t)$. The time average of $X(t)$ is denoted as $A[X(t)]$, and the time autocorrelation of $X(t)$ is denoted as $R[X(t)]$. These quantities are defined for continuous-time random processes as

$$\begin{aligned} A[X(t)] &= \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)\,dt \\ R[X(t),\tau] &= A[X(t)X^T(t+\tau)] \end{aligned} \tag{2.88}$$

The definitions for discrete-time random processes are straightforward extensions of the continuous-time definitions.

An ergodic process is a stationary random process for which

$$\begin{aligned} A[X(t)] &= E(X) \\ R[X(t),\tau] &= R_X(\tau) \end{aligned} \tag{2.89}$$

In the real world, we are often limited to only a few realizations of a stochastic process. For example, if we measure the fluctuation of a voltmeter reading, we are actually only measuring only one realization of a stochastic process. We can compute the time average, time autocorrelation, and other time-based statistics of the realization. If the random process is ergodic, then we can use those time averages to estimate the statistics of the stochastic process.

■ **EXAMPLE 2.13**

1. Suppose each unit of an electrical instrument is manufactured with a small random bias. Is the noise of the instrumentation ergodic? If we measure the noise of one instrument then we measure its bias, which is equal to its mean. However, if we measure the noise of another instrument it might have a different mean because it has a different bias. In other words, we cannot obtain the mean of the stochastic process by simply investigating one instrument (i.e., one realization of the stochastic process). Therefore, the stochastic process is not ergodic.

2. Suppose each unit of an electrical instrument is manufactured identically, each with zero-mean stationary Gaussian noise. Is the noise ergodic? In this case we could measure the mean of the process by measuring the noise of many separate instruments at one instant of time, or by measuring the noise of one instrument over an extended period of time. Either experiment would correctly inform us that the mean of the stochastic process is zero. We could find the statistics of the stochastic process using all the instruments at a single time, or using a single instrument at many different times. Therefore, the stochastic process is ergodic.

$\nabla\nabla\nabla$

The definitions of correlation and covariance can be extended to two stochastic processes $X(t)$ and $Y(t)$. The cross correlation of $X(t)$ and $Y(t)$ is defined as

$$R_{XY}(t_1, t_2) = E[X(t_1)Y^T(t_2)] \tag{2.90}$$

Two random processes $X(t)$ and $Y(t)$ are said to be uncorrelated if $R_{XY}(t_1, t_2) = E[X(t_1)]E[Y^T(t_2)]$ for all t_1 and t_2. The cross covariance of $X(t)$ and $Y(t)$ is defined as

$$C_{XY}(t_1, t_2) = E\left\{[X(t_1) - \bar{X}(t_1)][Y(t_2) - \bar{Y}(t_2)]^T\right\} \tag{2.91}$$

2.6 WHITE NOISE AND COLORED NOISE

If the RV $X(t_1)$ is independent from the RV $X(t_2)$ for all $t_1 \neq t_2$ then $X(t)$ is called white noise. Otherwise, $X(t)$ is called colored noise.

The whiteness or color content of a stochastic process can be characterized by its power spectrum. The power spectrum $S_X(\omega)$ of a wide-sense stationary stochastic process $X(t)$ is defined as the Fourier transform of the autocorrelation. The autocorrelation is the inverse Fourier transform of the power spectrum.

$$\begin{aligned} S_X(\omega) &= \int_{-\infty}^{\infty} R_X(\tau)e^{-j\omega\tau}\,d\tau \\ R_X(\tau) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} S_X(\omega)e^{j\omega\tau}\,d\omega \end{aligned} \tag{2.92}$$

These equations are called the Wiener–Khintchine relations after Norbert Wiener and Aleksandr Khinchin. Note that some authors put the term $1/2\pi$ on the right side of the $S_X(\omega)$ definition, in which case the $1/2\pi$ term on the right side of the

$R_X(\tau)$ definition disappears. The power spectrum is sometimes referred to as the power density spectrum, the power spectral density, or the power density. The power of a wide-sense stationary stochastic process is defined as

$$P_X = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_X(\omega)\,d\omega \tag{2.93}$$

The cross power spectrum of two wide-sense stationary stochastic processes $X(t)$ and $Y(t)$ is the Fourier transform of the cross correlation:

$$\begin{aligned} S_{XY}(\omega) &= \int_{-\infty}^{\infty} R_{XY}(\tau)e^{-j\omega\tau}\,d\tau \\ R_{XY}(\tau) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} S_{XY}(\omega)e^{j\omega\tau}\,d\omega \end{aligned} \tag{2.94}$$

Similar definitions hold for discrete-time random processes. The power spectrum of a discrete-time random process is defined as

$$\begin{aligned} S_X(\omega) &= \sum_{k=-\infty}^{\infty} R_X(k)e^{-j\omega k} \qquad \omega \in [-\pi, \pi] \\ R_X(k) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} S_X(\omega)e^{jk\omega}\,d\omega \end{aligned} \tag{2.95}$$

A discrete-time stochastic process $X(t)$ is called white noise if

$$\begin{aligned} R_X(k) &= \begin{cases} \sigma^2 & \text{if } k = 0 \\ 0 & \text{if } k \neq 0 \end{cases} \\ &= \sigma^2\delta_k \end{aligned} \tag{2.96}$$

where δ_k is the Kronecker delta function, defined as

$$\delta_k = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{if } k \neq 0 \end{cases} \tag{2.97}$$

The definition of discrete-time white noise shows that it does not have any correlation with itself except at the present time. If $X(k)$ is a discrete-time white noise process, then the RV $X(n)$ is uncorrelated with $X(m)$ unless $n = m$. This shows that the power of a discrete-time white noise process is equal at all frequencies:

$$S_X(\omega) = R_X(0) \qquad \text{for all } \omega \in [-\pi, \pi] \tag{2.98}$$

For a continuous-time random process, white noise is defined similarly. White noise has equal power at all frequencies (like white light):

$$S_X(\omega) = R_X(0) \qquad \text{for all } \omega \tag{2.99}$$

Substituting this expression for $S_X(\omega)$ into Equation (2.92), we see that for continuous-time white noise

$$R_X(\tau) = R_X(0)\delta(\tau) \tag{2.100}$$

where $\delta(\tau)$ is the continuous-time impulse function. That is, $\delta(\tau)$ is a function that is zero everywhere except at $\tau = 0$; it has a width of 0, a height of ∞, and an area of 1. Continuous-time white noise is not something that occurs in the real world because it has infinite power, as seen by comparing Equations (2.93) and (2.99). Nevertheless, many continuous-time processes approximate white noise and are useful in mathematical analyses of signals and systems.

■ **EXAMPLE 2.14**

Suppose that a zero-mean stationary stochastic process has the autocorrelation function

$$R_X(\tau) = \sigma^2 e^{-\beta|\tau|} \tag{2.101}$$

where β is a positive real number. The power spectrum is computed from Equation (2.92) as

$$\begin{aligned} S_X(\omega) &= \int_{-\infty}^{\infty} \sigma^2 e^{-\beta|\tau|} e^{-j\omega\tau}\, d\tau \\ &= \int_{-\infty}^{0} \sigma^2 e^{(\beta - j\omega)\tau}\, d\tau + \int_{0}^{\infty} \sigma^2 e^{-(\beta + j\omega)\tau}\, d\tau \\ &= \frac{\sigma^2}{\beta - j\omega} + \frac{\sigma^2}{\beta + j\omega} \\ &= \frac{2\sigma^2\beta}{\omega^2 + \beta^2} \end{aligned} \tag{2.102}$$

The variance of the stochastic process is computed as

$$\begin{aligned} E[X^2(t)] &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{2\sigma^2\beta}{\omega^2 + \beta^2}\, d\omega \\ &= \frac{\sigma^2\beta}{\pi} \left[\frac{1}{\beta} \tan^{-1} \frac{\omega}{\beta} \right]_{-\infty}^{\infty} \\ &= \sigma^2 \\ &= R_X(0) \end{aligned} \tag{2.103}$$

∇∇∇

2.7 SIMULATING CORRELATED NOISE

In optimal filtering research and experiments, we often have to simulate correlated white noise. That is, we need to create random vectors whose elements are correlated with each other according to some predefined covariance matrix. In this section, we will present one way of accomplishing this.

Suppose we want to generate an n-element random vector w that has zero mean and covariance Q:

$$Q = \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{1n} \\ \vdots & & \vdots \\ \sigma_{1n} & \cdots & \sigma_n^2 \end{bmatrix} \tag{2.104}$$

Since Q is a covariance matrix, we know that all of its eigenvalues are real and nonnegative. We can therefore denote its eigenvalues as μ_k^2:

$$\lambda(Q) = \mu_k^2 \qquad (k = 1, \ldots, n) \tag{2.105}$$

Suppose the eigenvectors of Q are found to be $d_1, \cdots, d_n$. Augment the d_i vectors together to obtain an $n \times n$ matrix D. Since Q is symmetric, we can always choose

the eigenvectors such that D is orthogonal, that is, $D^{-1} = D^T$. We therefore obtain the Jordan form decomposition of Q as

$$Q = D\hat{Q}D^T \tag{2.106}$$

where $\hat{Q}$ is the diagonal matrix of the eigenvalues of Q. That is,

$$\hat{Q} = \text{diag}(\mu_1^2, \cdots, \mu_n^2) \tag{2.107}$$

Now we define the random vector v as $v = D^{-1}w$, so that $w = Dv$. Therefore,

$$\begin{aligned} E(vv^T) &= E(D^T ww^T D) \\ &= D^T Q D \\ &= \hat{Q} \\ &= \text{diag}(\mu_1^2, \cdots, \mu_n^2) \end{aligned} \tag{2.108}$$

This shows how we can generate an n-element random vector w with a covariance matrix of Q. The algorithm is given as follows.

Correlated noise simulation

1. Find the eigenvalues of Q, and denote them as $\mu_1^2, \cdots, \mu_n^2$
2. Find the eigenvectors of Q, and denote them as $d_1, \cdots, d_n$, such that

$$\begin{aligned} D &= \begin{bmatrix} d_1 & \cdots & d_n \end{bmatrix} \\ D^{-1} &= D^T \end{aligned} \tag{2.109}$$

3. For $i = 1, \cdots, n$ compute the random variable $v_i = \mu_i r_i$, where each r_i is an independent random number with a variance of 1 (unity variance).
4. Set $w = Dv$.

2.8 SUMMARY

In this chapter, we have reviewed the basic concepts of probability, random variables, and stochastic processes. The probability of some event occurring is simply and intuitively defined as the number of times the event occurs divided by the number of chances the event has to occur. A random variable (RV) is a variable whose value is not certain, but is governed by the laws of probability. For example, your score on the test for this chapter is not deterministic, but is a random variable. Your *actual* score, after you take the test, will be a specific, deterministic number. But *before* you take the test, you do not know what you will get on the test. You may suppose that you will probably get between 80% and 90% if you have a decent understanding of the material, but your actual score will be determined by random events such as your health, how well you sleep the night before, what topics the instructor decides to cover on the test versus what topics you study, what the traffic was like on the way to school, the mood of the instructor when she grades the test, and so on. A stochastic process is a random variable that changes with time,

such as your performance on all of the quizzes and homework assignments for this course. The expected value of your test grades may be constant throughout the duration of the course if you are a consistent person, or it may increase if you tend to study harder as the course progresses, or it may decrease if you tend to study less as the course progresses. Probability, random variables, stochastic processes, and related topics form a huge area of study that we have only touched on in this chapter. Additional information on these topics can be found in many textbooks, including [Pap02, Pee01]. A study of these topics will allow a student to delve into many practical engineering subjects, including control and estimation theory, signal processing, and communications theory.

PROBLEMS

Written exercises

2.1 What is the 0th moment of an RV? What is the 0th central moment of an RV?

2.2 Suppose a deck of 52 cards is randomly divided into four piles of 13 cards each. Find the probability that each pile contains exactly one ace [Gre01].

2.3 Determine the value of a in the function

$$f_X(x) = \begin{cases} ax(1-x) & x \in [0,1] \\ 0 & \text{otherwise} \end{cases}$$

so that $f_X(x)$ is a valid probability density function [Lie67].

2.4 Determine the value of a in the function

$$f_X(x) = \frac{a}{e^x + e^{-x}}$$

so that $f_X(x)$ is a valid probability density function. What is the probability that $|X| \leq 1$?

2.5 The probability density function of an exponentially distributed random variable is defined as follows.

$$f_X(x) = \begin{cases} ae^{-ax} & x > 0 \\ 0 & x \leq 0 \end{cases}$$

where $a \geq 0$.

a) Find the probability distribution function of an exponentially distributed random variable.
b) Find the mean of an exponentially distributed random variable.
c) Find the second moment of an exponentially distributed random variable.
d) Find the variance of an exponentially distributed random variable.
e) What is the probability that an exponentially distributed random variable takes on a value within one standard deviation of its mean?

2.6 Derive an expression for the skew of a random variable as a function of its first, second, and third moments.

2.7 Consider the following probability density function:

$$f_X(x) = \frac{ab}{b^2 + x^2}, \qquad b > 0$$

a) Determine the value of a in the so that $f_X(x)$ is a valid probability density function. (The correct value of a makes $f_X(x)$ a Cauchy pdf.)
b) Find the mean of a Cauchy random variable.

2.8 Consider two zero-mean uncorrelated random variables W and V with standard deviations σ_w and σ_v, respectively. What is the standard deviation of the random variable $X = W + V$?

2.9 Consider two scalar RVs X and Y.
a) Prove that if X and Y are independent, then their correlation coefficient $\rho = 0$.
b) Find an example of two RVs that are not independent but that have a correlation coefficient of zero.
c) Prove that if Y is a linear function of X then $\rho = \pm 1$.

2.10 Consider the following function [Lie67].

$$f_{XY}(x, y) = \begin{cases} ae^{-2x}e^{-3y} & x > 0, y > 0 \\ 0 & \text{otherwise} \end{cases}$$

a) Find the value of a so that $f_{XY}(x, y)$ is a valid joint probability density function.
b) Calculate $\bar{x}$ and $\bar{y}$.
c) Calculate $E(X^2)$, $E(Y^2)$, and $E(XY)$.
d) Calculate the autocorrelation matrix of the random vector $\begin{bmatrix} X & Y \end{bmatrix}^T$.
e) Calculate the variance σ_x^2, the variance σ_y^2, and the covariance C_{XY}.
f) Calculate the autocovariance matrix of the random vector $\begin{bmatrix} X & Y \end{bmatrix}^T$.
g) Calculate the correlation coefficient between X and Y.

2.11 A stochastic process has the autocorrelation $R_X(\tau) = Ae^{-k|\tau|}$, where A and k are positive constants.
a) What is the power spectrum of the stochastic process?
b) What is the total power of the stochastic process?
c) What value of k results in half of the total power residing in frequencies less than 1 Hz?

2.12 Suppose X is a random variable, and $Y(t) = X \cos t$ is a stochastic process.
a) Find the expected value of $Y(t)$.
b) Find $A[Y(t)]$, the time average of $Y(t)$.
c) Under what condition is $\bar{y}(t) = A[Y(t)]$?

2.13 Consider the equation $Z = X + V$. The pdf's of X and B are given in Figure 2.5.
a) Plot the pdf of $(Z|X)$ as a function of X for $Z = 0.5$.

b) Given $Z = 0.5$, what is conditional expectation of X? What is the most probable value of X? What is the median value of X?

Figure 2.5 pdf's for Problem 2.13 [Sch73].

2.14 The temperature at noon in London is a stochastic process. Is it ergodic?

Computer exercises

2.15 Generate $N = 50$ independent random numbers, each uniformly distributed between 0 and 1. Plot a histogram of the random numbers using 10 bins. What is the sample mean and standard deviation of the numbers that you generated? What would you expect to see for the mean and standard deviation (i.e., what are the theoretical mean and standard deviation)? Repeat for $N = 500$ and $N = 5,000$ random numbers. What changes in the histogram do you see as N increases?

2.16 Generate 10,000 samples of $(x_1 + x_2)/2$, where each x_i is a random number uniformly distributed on $[-1/2, +1/2]$. Plot the 50-bin histogram. Repeat for $(x_1 + x_2 + x_3 + x_4)/4$. Describe the difference between the two histograms.

CHAPTER 3

Least squares estimation

> The most probable value of the unknown quantities will be that in which the sum of the squares of the differences between the actually observed and the computed values multiplied by numbers that measure the degree of precision is a minimum.
>
> —Karl Friedrich Gauss [Gau04]

In this chapter, we will discuss least squares estimation, which is the basic idea of Karl Gauss's quote above.[1] The material in this chapter relies on the theory of the previous two chapters, and will enable us to derive optimal state estimators later in this book.

Section 3.1 discusses the estimation of a constant vector on the basis of several linear but noisy measurements of that vector. Section 3.2 extends the results of Section 3.1 to the case in which some measurements are more noisy than others; that is, we have less confidence in some measurements than in others. Sections 3.1 and 3.2 use matrices and vectors whose dimensions grow larger as more measurements are obtained. This makes the problem cumbersome if many measurements are available. This leads us to Section 3.3, which presents a recursive way of estimating a constant on the basis of noisy measurements. Recursive estimation in this chapter is a method of estimating a constant without increasing the computa-

[1] Gauss published his book in 1809, although he claimed to have worked out his theory as early as 1795 (when he was 18 years old).

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

tional effort of the algorithm, regardless of how many measurements are available. Finally, Section 3.4 presents the Wiener filter, which is a method of estimating a time-varying signal that is corrupted by noise, on the basis of noisy measurements. Until 1960, Wiener filtering was the state of the art in signal estimation. The paradigm of signal estimation was shattered with the publication of Rudolph Kalman's work and related papers in the early 1960s, but it is still worthwhile understanding Wiener filtering because of its historical place in the history of signal estimation. Furthermore, Wiener filtering is still very useful in signal processing and communication theory.

3.1 ESTIMATION OF A CONSTANT

In this section, we will determine how to estimate a constant on the basis of several noisy measurements of that constant. For example, suppose we have a resistor but we do not know its resistance. We take several measurements of its resistance using a multimeter, but the measurements are noisy because we have a cheap multimeter. We want to estimate the resistance on the basis of our noisy measurements. In this case, we want to estimate a constant scalar but, in general, we may want to estimate a constant vector.

To put the problem in mathematical terms, suppose x is a constant but unknown n-element vector, and y is a k-element noisy measurement vector. How can we find the "best" estimate $\hat{x}$ of x? Let us assume that each element of the measurement vector y is a linear combination of the elements of x, with the addition of some measurement noise:

$$\begin{aligned} y_1 &= H_{11}x_1 + \cdots + H_{1n}x_n + v_1 \\ &\vdots \\ y_k &= H_{k1}x_1 + \cdots + H_{kn}x_n + v_k \end{aligned} \tag{3.1}$$

This set of equations can be put into matrix form as

$$y = Hx + v \tag{3.2}$$

Now define ϵ_y as the difference between the noisy measurements and the vector $H\hat{x}$:

$$\epsilon_y = y - H\hat{x} \tag{3.3}$$

ϵ_y is called the measurement residual. As Karl Gauss wrote [Gau04], the most probable value of the vector x is the vector $\hat{x}$ that minimizes the sum of squares between the observed values y and the vector $H\hat{x}$. So we will try to compute the $\hat{x}$ that minimizes the cost function J, where J is given as

$$\begin{aligned} J &= \epsilon_{y1}^2 + \cdots + \epsilon_{yk}^2 \\ &= \epsilon_y^T \epsilon_y \end{aligned} \tag{3.4}$$

J is often referred to in control and estimation books and papers as a cost function, objective function, or return function. We can substitute for ϵ_y in the above equation to rewrite J as

$$\begin{aligned} J &= (y - H\hat{x})^T (y - H\hat{x}) \\ &= y^T y - \hat{x}^T H^T y - y^T H\hat{x} + \hat{x}^T H^T H\hat{x} \end{aligned} \tag{3.5}$$

In order to minimize J with respect to $\hat{x}$, we compute its partial derivative and set it equal to zero:

$$\begin{aligned} \frac{\partial J}{\partial \hat{x}} &= -y^T H - y^T H + 2\hat{x}^T H^T H \\ &= 0 \end{aligned} \tag{3.6}$$

Solving this equation for $\hat{x}$ results in

$$\begin{aligned} H^T y &= H^T H \hat{x} \\ \hat{x} &= (H^T H)^{-1} H^T y \\ &= H^L y \end{aligned} \tag{3.7}$$

where H^L, the left pseudo inverse of H, exists if $k \geq n$ and H is full rank. This means that the number of measurements k is greater than the number of variables n that we are trying to estimate, and the measurements are linearly independent. In order to prove that we have found a minimum rather than some other type of stationary point[2] of J, we need to prove that the second derivative of J is positive semidefinite (see Problem 3.1).

■ EXAMPLE 3.1

Let us go back to our original problem of trying to estimate the resistance x of an unmarked resistor on the basis of k noisy measurements from a multimeter. In this case, x is a scalar so our k noisy measurements are given as

$$\begin{aligned} y_1 &= x + v_1 \\ &\vdots \\ y_k &= x + v_k \end{aligned} \tag{3.8}$$

These k equations can be combined into a single matrix equation as

$$\begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} x + \begin{bmatrix} v_1 \\ \vdots \\ v_k \end{bmatrix} \tag{3.9}$$

Equation (3.7) shows that the optimal estimate of the resistance x is given as

$$\begin{aligned} \hat{x} &= (H^T H)^{-1} H^T y \\ &= \left(\begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} \\ &= \frac{1}{k}(y_1 + \cdots + y_k) \end{aligned} \tag{3.10}$$

In this simple example, we see that least squares estimation agrees with our intuition to simply compute the average of the measurements.

∇∇∇

[2] A stationary point of a function is any point at which its derivative is equal to zero. A stationary point of a scalar function could be a maximum, a minimum, or an inflection point. A stationary point of a vector function could be a maximum, a minimum, or a saddle point.

3.2 WEIGHTED LEAST SQUARES ESTIMATION

In the previous section, we assumed that we had an equal amount of confidence in all of our measurements. Now suppose we have more confidence in some measurements than others. In this case, we need to generalize the results of the previous section to obtain weighted least squares estimation. For example, suppose we have several measurements of the resistance of an unmarked resistor. Some of the measurements were taken with an expensive multimeter with low noise, but other measurements were taken with a cheap multimeter by a tired student late at night. We have more confidence in the first set of measurements, so we should somehow place more emphasis on those measurements than on the others. However, even though the second set of measurements is less reliable, it seems that we could get at least *some* information from them. This section shows that we can indeed get some information from less reliable measurements. We should never throw away measurements, no matter how unreliable they may be.

To put the problem in mathematical terms, suppose x is a constant but unknown n-element vector, and y is a k-element noisy measurement vector. We assume that each element of y is a linear combination of the elements of x, with the addition of some measurement noise, and the variance of the measurement noise may be different for each element of y:

$$\begin{aligned} \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} &= \begin{bmatrix} H_{11} & \cdots & H_{1n} \\ \vdots & \ddots & \vdots \\ H_{k1} & \cdots & H_{kn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} v_1 \\ \vdots \\ v_k \end{bmatrix} \\ E(v_i^2) &= \sigma_i^2 \quad (i = 1, \ldots, k) \end{aligned} \tag{3.11}$$

We assume that the noise for each measurement is zero-mean and independent. The measurement covariance matrix is

$$\begin{aligned} R &= E(vv^T) \\ &= \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \sigma_k^2 \end{bmatrix} \end{aligned} \tag{3.12}$$

Now we will minimize the following quantity with respect to $\hat{x}$.

$$J = \epsilon_{y1}^2/\sigma_1^2 + \cdots + \epsilon_{yk}^2/\sigma_k^2 \tag{3.13}$$

Note that instead of minimizing the sum of squares of the ϵ_y elements as we did in Equation (3.4), we will minimize the *weighted* sum of squares. If y_1 is a relatively noisy measurement, for example, then we do not care as much about minimizing the difference between y_1 and the first element of $H\hat{x}$ because we do not have much confidence in y_1 in the first place. The cost function J can be written as

$$\begin{aligned} J &= \epsilon_y^T R^{-1} \epsilon_y \\ &= (y - H\hat{x})^T R^{-1} (y - H\hat{x}) \\ &= y^T R^{-1} y - \hat{x}^T H^T R^{-1} y - y^T R^{-1} H\hat{x} + \hat{x}^T H^T R^{-1} H\hat{x} \end{aligned} \tag{3.14}$$

Now we take the partial derivative of J with respect to $\hat{x}$ and set it equal to zero to compute the best estimate $\hat{x}$:

$$\begin{aligned}
\frac{\partial J}{\partial \hat{x}} &= -y^T R^{-1} H + \hat{x}^T H^T R^{-1} H \\
&= 0 \\
H^T R^{-1} y &= H^T R^{-1} H \hat{x} \\
\hat{x} &= (H^T R^{-1} H)^{-1} H^T R^{-1} y
\end{aligned} \tag{3.15}$$

Note that this method requires that the measurement noise matrix R be nonsingular. In other words, each of the measurements y_i must be corrupted by at least *some* noise for this method to work.

■ EXAMPLE 3.2

We return to our original problem of trying to estimate the resistance x of an unmarked resistor on the basis of k noisy measurements from a multimeter. In this case, x is a scalar so our k noisy measurements are given as

$$\begin{aligned}
y_i &= x + v_i \\
E(v_i^2) &= \sigma_i^2 \quad (i = 1, \ldots, k)
\end{aligned} \tag{3.16}$$

The k measurement equation can be combined into a single matrix equation as

$$\begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} x + \begin{bmatrix} v_1 \\ \vdots \\ v_k \end{bmatrix} \tag{3.17}$$

and the measurement noise covariance is given as

$$R = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_k^2) \tag{3.18}$$

Equation (3.15) shows that the optimal estimate of the resistance x is given as

$$\begin{aligned}
\hat{x} &= (H^T R^{-1} H)^{-1} H^T R^{-1} y \\
&= \left(\begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k^2 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right)^{-1} \times \\
&\quad \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k^2 \end{bmatrix}^{-1} \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} \\
&= \left(\sum 1/\sigma_i^2 \right)^{-1} \left(y_1/\sigma_1^2 + \cdots + y_k/\sigma_k^2 \right)
\end{aligned} \tag{3.19}$$

We see that the optimal estimate $\hat{x}$ is a weighted sum of the measurements, where each measurement is weighted by the inverse of its uncertainty. In other words, we put more emphasis on certain measurements, in agreement

with our intuition. Note that if all of the σ_i constants are equal, this estimate reduces to the simpler form given in Equation (3.10).

$\nabla\nabla\nabla$

3.3 RECURSIVE LEAST SQUARES ESTIMATION

Equation (3.15) gives us a way to compute the optimal estimate of a constant, but there is a problem. Note that the H matrix in (3.15) is a $k \times n$ matrix. If we obtain measurements sequentially and want to update our estimate of x with each new measurement, we need to augment the H matrix and completely recompute the estimate $\hat{x}$. If the number of measurements becomes large, then the computational effort could become prohibitive. For example, suppose we obtain a measurement of a satellite's altitude once per second. After one hour has passed, the number of measurements is 3600 and growing. The computational effort of least squares estimation can rapidly outgrow our resources.

In this section, we show how to *recursively* compute the weighted least squares estimate of a constant. That is, suppose we have $\hat{x}$ after $(k-1)$ measurements, and we obtain a new measurement y_k. How can we update our estimate without completely reworking Equation (3.15)?

A linear recursive estimator can be written in the form

$$\begin{aligned} y_k &= H_k x + v_k \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \end{aligned} \tag{3.20}$$

That is, we compute $\hat{x}_k$ on the basis of the previous estimate $\hat{x}_{k-1}$ and the new measurement y_k. K_k is a matrix to be determined called the estimator gain matrix. The quantity $(y_k - H_k\hat{x}_{k-1})$ is called the correction term. Note that if the correction term is zero, or if the gain matrix is zero, then the estimate does not change from time step $(k-1)$ to k.

Before we compute the optimal gain matrix K_k, let us think about the mean of the estimation error of the linear recursive estimator. The estimation error mean can be computed as

$$\begin{aligned} E(\epsilon_{x,k}) &= E(x - \hat{x}_k) \\ &= E[x - \hat{x}_{k-1} - K_k(y_k - H_k\hat{x}_{k-1})] \\ &= E[\epsilon_{x,k-1} - K_k(H_k x + v_k - H_k\hat{x}_{k-1})] \\ &= E[\epsilon_{x,k-1} - K_k H_k(x - \hat{x}_{k-1}) - K_k v_k] \\ &= (I - K_k H_k)E(\epsilon_{x,k-1}) - K_k E(v_k) \end{aligned} \tag{3.21}$$

So if $E(v_k) = 0$ and $E(\epsilon_{x,k-1}) = 0$, then $E(\epsilon_{x,k}) = 0$. In other words, if the measurement noise v_k is zero-mean for all k, and the initial estimate of x is set equal to the expected value of x [i.e., $\hat{x}_0 = E(x)$], then the expected value of $\hat{x}_k$ will be equal to x_k for all k. Because of this, the estimator of Equation (3.20) is called an unbiased estimator. Note that this property holds regardless of the value of the gain matrix K_k. This is a desirable property of an estimator because it says that, *on average*, the estimate $\hat{x}$ will be equal to the true value x.

Next we turn our attention to the determination of the optimal value of K_k. Since the estimator is unbiased regardless of what value of K_k we use, we must

choose some other optimality criterion in order to determine K_k. The optimality criterion that we choose to minimize is the sum of the variances of the estimation errors at time k:

$$\begin{aligned} J_k &= E[(x_1 - \hat{x}_1)^2)] + \cdots + E[(x_n - \hat{x}_n)^2)] \\ &= E\left(\epsilon_{x1,k}^2 + \cdots + \epsilon_{xn,k}^2\right) \\ &= E\left(\epsilon_{x,k}^T \epsilon_{x,k}\right) \\ &= E\left[\mathrm{Tr}(\epsilon_{x,k}\epsilon_{x,k}^T)\right] \\ &= \mathrm{Tr}P_k \end{aligned} \tag{3.22}$$

where P_k, the estimation-error covariance, is defined by the above equation. We can use a process similar to that followed in Equation (3.21) to obtain a recursive formula for the calculation of P_k:

$$\begin{aligned} P_k &= E(\epsilon_{x,k}\epsilon_{x,k}^T) \\ &= E\left\{[(I - K_kH_k)\epsilon_{x,k-1} - K_kv_k][\cdots]^T\right\} \\ &= (I - K_kH_k)E(\epsilon_{x,k-1}\epsilon_{x,k-1}^T)(I - K_kH_k)^T - \\ &\quad K_kE(v_k\epsilon_{x,k-1}^T)(I - K_kH_k)^T - (I - K_kH_k)E(\epsilon_{x,k-1}v_k^T)K_k^T + \\ &\quad K_kE(v_kv_k^T)K_k^T \end{aligned} \tag{3.23}$$

Now note that $\epsilon_{x,k-1}$ [the estimation error at time $(k-1)$] is independent of v_k (the measurement noise at time k). Therefore,

$$\begin{aligned} E(v_k\epsilon_{x,k-1}^T) &= E(v_k)E(\epsilon_{x,k-1}) \\ &= 0 \end{aligned} \tag{3.24}$$

since both expected values are zero. Therefore, Equation (3.23) becomes

$$P_k = (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T \tag{3.25}$$

where R_k is the covariance of v_k. This is the recursive formula for the covariance of the least squares estimation error. This is consistent with intuition in the sense that as the measurement noise increases (i.e., R_k increases) the uncertainty in our estimate also increases (i.e., P_k increases). Note that P_k should be positive definite since it is a covariance matrix, and the form of Equation (3.25) guarantees that P_k will be positive definite, assuming that P_{k-1} and R_k are positive definite.

Now we need to find the value of K_k that makes the cost function in Equation (3.22) as small as possible. The mean of the estimation error is zero for any value of K_k. So if we choose K_k to make the cost function (i.e., the trace of P_k) small then the estimation error will not only be zero-mean, but it will also be consistently close to zero. In order to find the best value of K_k, first we need to recall from Equation (1.66) that $\frac{\partial \mathrm{Tr}(ABA^T)}{\partial A} = 2AB$ if B is symmetric. With this in mind we can use Equations (3.22), (3.25), and the chain rule to obtain

$$\frac{\partial J_k}{\partial K_k} = 2(I - K_kH_k)P_{k-1}(-H_k^T) + 2K_kR_k \tag{3.26}$$

In order to find the value of K_k that minimizes J_k, we set the above derivative equal to zero and then solve for K_k as follows:

$$\begin{aligned} K_k R_k &= (I - K_k H_k) P_{k-1} H_k^T \\ K_k(R_k + H_k P_{k-1} H_k^T) &= P_{k-1} H_k^T \\ K_k &= P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \end{aligned} \tag{3.27}$$

Equations (3.20), (3.25), and (3.27) form the recursive least squares estimator. The recursive least squares estimator can be summarized as follows.

Recursive least squares estimation

1. Initialize the estimator as follows:

$$\begin{aligned} \hat{x}_0 &= E(x) \\ P_0 &= E[(x - \hat{x}_0)(x - \hat{x}_0)^T] \end{aligned} \tag{3.28}$$

   If no knowledge about x is available before measurements are taken, then $P_0 = \infty I$. If perfect knowledge about x is available before measurements are taken, then $P_0 = 0$.

2. For $k = 1, 2, \cdots$, perform the following.

   (a) Obtain the measurement y_k, assuming that y_k is given by the equation

$$y_k = H_k x + v_k \tag{3.29}$$

   where v_k is a zero-mean random vector with covariance R_k. Further assume that the measurement noise at each time step k is independent, that is, $E(v_i v_k) = R_k \delta_{k-i}$. This implies that the measurement noise is white.

   (b) Update the estimate of x and the estimation-error covariance P as follows:

$$\begin{aligned} K_k &= P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k \hat{x}_{k-1}) \\ P_k &= (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T \end{aligned} \tag{3.30}$$

3.3.1 Alternate estimator forms

Sometimes it is useful to write the equations for P_k and K_k in alternate forms. Although these alternate forms are mathematically identical, they can be beneficial from a computational point of view. They can also lead to new results, which we will discover in later chapters.

First we will find an alternate form for the expression for the estimation-error covariance. Substituting for K_k from Equation (3.27) into Equation (3.25) we obtain

$$P_k = \left[I - P_{k-1} H_k^T S_k^{-1} H_k\right] P_{k-1} [\cdots]^T + K_k R_k K_k^T \tag{3.31}$$

where we have introduced the auxiliary variable $S_k = (H_k P_{k-1} H_k^T + R_k)$. We again substitute for K_k at the end of this equation, and expand terms to obtain

$$\begin{aligned} P_k &= P_{k-1} - P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} - P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} + \\ &\quad P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} H_k^T S_k^{-1} H_k P_{k-1} + P_{k-1}H_k^T S_k^{-1} R_k S_k^{-1} H_k P_{k-1} \end{aligned} \tag{3.32}$$

Combining the last two terms in this equation gives

$$\begin{aligned} P_k &= P_{k-1} - 2P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} + P_{k-1}H_k^T S_k^{-1} S_k S_k^{-1} H_k P_{k-1} \\ &= P_{k-1} - 2P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} + P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} \\ &= P_{k-1} - P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} \end{aligned} \tag{3.33}$$

Now notice from the expression for K_k in Equation (3.27) that K_k appears implicitly in the above equation. We can therefore rewrite this equation as

$$\begin{aligned} P_k &= P_{k-1} - K_k H_k P_{k-1} \\ &= (I - K_k H_k) P_{k-1} \end{aligned} \tag{3.34}$$

This is a simpler equation for P_k [compared with Equation (3.25)] but numerical computing problems (i.e., scaling issues) may cause this expression for P_k to not be positive definite, even when P_{k-1} and R_k are positive definite.

We can also use the matrix inversion lemma from Section 1.1.2 to rewrite the measurement update equation for P_k. Starting with Equation (3.33) we obtain

$$P_k = P_{k-1} - P_{k-1}H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} H_k P_{k-1} \tag{3.35}$$

Taking the inverse of both sides of this equation gives

$$P_k^{-1} = [P_{k-1} - P_{k-1}H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} H_k P_{k-1}]^{-1} \tag{3.36}$$

Applying the matrix inversion lemma to this equation gives

$$\begin{aligned} P_k^{-1} &= P_{k-1}^{-1} + P_{k-1}^{-1} P_{k-1} H_k^T \left[(H_k P_{k-1} H_k^T + R_k) - \right. \\ &\quad \left. H_k P_{k-1} P_{k-1}^{-1} (P_{k-1} H_k^T) \right]^{-1} H_k P_{k-1} P_{k-1}^{-1} \\ &= P_{k-1}^{-1} + H_k^T R_k^{-1} H_k \end{aligned} \tag{3.37}$$

Inverting both sides of this equation gives

$$P_k = \left[P_{k-1}^{-1} + H_k^T R_k^{-1} H_k \right]^{-1} \tag{3.38}$$

This equation for P_k is more complicated in that it requires three matrix inversions, but it may be computationally advantageous in some situations, as will be discussed in Section 6.2.

We can use Equation (3.38) to derive an equivalent equation for the estimator gain K_k. Starting with Equation (3.27) we have

$$K_k = P_{k-1}H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \tag{3.39}$$

Premultiplying the right side by $P_k P_k^{-1}$, which is equal to the identity matrix, gives

$$K_k = P_k P_k^{-1} P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \tag{3.40}$$

Substituting for P_k^{-1} from Equation (3.38) gives

$$K_k = P_k(P_{k-1}^{-1} + H_k^T R_k^{-1} H_k) P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \tag{3.41}$$

Note the $P_{k-1}H_k^T$ factor that is on the right of the first term in parentheses. We can multiply this factor inside the first term in parentheses to obtain

$$K_k = P_k(H_k^T + H_k^T R_k^{-1} H_k P_{k-1} H_k^T)(H_k P_{k-1} H_k^T + R_k)^{-1} \tag{3.42}$$

Now bring H_k^T out to the left side of the parentheses to obtain

$$K_k = P_k H_k^T (I + R_k^{-1} H_k P_{k-1} H_k^T)(H_k P_{k-1} H_k^T + R_k)^{-1} \tag{3.43}$$

Now premultiply the first parenthetical expression by R_k^{-1}, and multiply on the inside of the parenthetical expression by R_k, to obtain

$$\begin{aligned} K_k &= P_k H_k^T R_k^{-1} (R_k + H_k P_{k-1} H_k^T)(H_k P_{k-1} H_k^T + R_k)^{-1} \\ &= P_k H_k^T R_k^{-1} \end{aligned} \tag{3.44}$$

General recursive least squares estimation

The recursive least squares algorithm can be summarized with the following equations. The measurement equations are given as

$$\begin{aligned} y_k &= H_k x + v_k \\ x &= \text{constant} \\ E(v_k) &= 0 \\ E(v_k v_i^T) &= R_k \delta_{k-i} \end{aligned} \tag{3.45}$$

The initial estimate of the constant vector x, along with the uncertainty in that estimate, is given as

$$\begin{aligned} \hat{x}_0 &= E(x) \\ P_0 &= E[(x - \hat{x}_0)(x - \hat{x}_0)^T] \end{aligned} \tag{3.46}$$

The recursive least squares algorithm is given as follows.

For $k = 1, 2, \cdots$,

$$\begin{aligned} K_k &= P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \\ &= P_k H_k^T R_k^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k \hat{x}_{k-1}) \\ P_k &= (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T \\ &= (P_{k-1}^{-1} + H_k^T R_k^{-1} H_k)^{-1} \\ &= (I - K_k H_k) P_{k-1} \end{aligned} \tag{3.47}$$

■ EXAMPLE 3.3

Once again we revisit the problem of trying to estimate the resistance x of an unmarked resistor on the basis of noisy measurements from a multimeter. However, we do not want to wait until we have all the measurements in order to have an estimate. We want to recursively modify our estimate of x each time we obtain a new measurement. At sample time k our measurement is

$$\begin{aligned} y_k &= H_k x + v_k \\ H_k &= 1 \\ R_k &= E(v_k^2) \end{aligned} \tag{3.48}$$

For this scalar problem, the measurement matrix H_k is a scalar, and the measurement noise covariance R_k is also a scalar. We will suppose that each measurement has the same covariance so the measurement covariance R_k is not a function of k, and can be written as R. Initially, before we have any measurements, we have some idea about the value of the resistance x, and this forms our initial estimate. We also have some uncertainty about our initial estimate, and this forms our initial covariance:

$$\begin{aligned} \hat{x}_0 &= E(x) \\ P_0 &= E[(x-\hat{x}_0)(x-\hat{x}_0)^T] \\ &= E[(x-\hat{x}_0)^2] \end{aligned} \tag{3.49}$$

If we have absolutely no idea about the resistance value, then $P(0) = \infty$. If we are 100% certain about the resistance value before taking any measurements, then $P(0) = 0$ (but then, of course, there would not be any need to take measurements). Equation (3.47) tells us how to obtain the estimator gain, the estimate of x, and the estimation covariance, after the first measurement ($k = 1$):

$$\begin{aligned} K_k &= P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1} \\ K_1 &= P_0(P_0+R)^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \\ \hat{x}_1 &= \hat{x}_0 + \frac{P_0}{P_0+R}(y_1 - \hat{x}_0) \\ P_k &= (I-K_kH_k)P_{k-1}(I-K_kH_k)^T + K_kR_kK_k^T \\ P_1 &= \frac{P_0R}{P_0+R} \end{aligned} \tag{3.50}$$

Repeating these calculations to find these quantities after the second measurement ($k = 2$) gives

$$\begin{aligned} K_2 &= \frac{P_1}{P_1+R} = \frac{P_0}{2P_0+R} \\ P_2 &= \frac{P_1R}{P_1+R} = \frac{P_0R}{2P_0+R} \\ \hat{x}_2 &= \hat{x}_1 + \frac{P_1}{P_1+R}(y_2 - \hat{x}_1) \\ &= \frac{P_0+R}{2P_0+R}\hat{x}_1 + \frac{P_0}{2P_0+R}y_2 \end{aligned} \tag{3.51}$$

By induction, we can find general expressions for P_{k-1}, K_k, and $\hat{x}_k$ as follows:

$$\begin{aligned} P_{k-1} &= \frac{P_0 R}{(k-1)P_0 + R} \\ K_k &= \frac{P_0}{kP_0 + R} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - \hat{x}_{k-1}) \\ &= (1 - K_k)\hat{x}_{k-1} + K_k y_k \\ &= \frac{(k-1)P_0 + R}{kP_0 + R}\hat{x}_{k-1} + \frac{P_0}{kP_0 + R} y_k \end{aligned} \tag{3.52}$$

Note that if x is known perfectly *a priori* (i.e., before any measurements are obtained) then $P_0 = 0$, and the above equations show that $K_k = 0$ and $\hat{x}_k = \hat{x}_0$. That is, the optimal estimate of x is independent of any measurements that are obtained. On the other hand, if x is completely unknown *a priori*, then $P_0 \to \infty$, and the above equations show that

$$\begin{aligned} \hat{x}_k &= \frac{(k-1)P_0}{kP_0}\hat{x}_{k-1} + \frac{P_0}{kP_0} y_k \\ &= \frac{(k-1)}{k}\hat{x}_{k-1} + \frac{1}{k} y_k \\ &= \frac{1}{k}[(k-1)\hat{x}_{k-1} + y_k] \end{aligned} \tag{3.53}$$

In other words, the optimal estimate of x is equal to the running average of the measurements y_k, which can be written as

$$\begin{aligned} \bar{y}_k &= \frac{1}{k}\sum_{j=1}^{k} y_j \\ &= \frac{1}{k}\left(\sum_{j=1}^{k-1} y_j + y_k\right) \\ &= \frac{1}{k}\left[(k-1)\left(\frac{1}{k-1}\sum_{j=1}^{k-1} y_j\right) + y_k\right] \\ &= \frac{1}{k}[(k-1)\bar{y}_{k-1} + y_k] \end{aligned} \tag{3.54}$$

$\nabla\nabla\nabla$

■ EXAMPLE 3.4

In this example, we illustrate the computational advantages of the first form of the covariance update in Equation (3.47) compared with the third form. Suppose we have a scalar parameter x and a perfect measurement of it. That is, $H_1 = 1$ and $R_1 = 0$. Further suppose that our initial estimation covariance $P_0 = 6$, and our computer provides precision of three digits to the right of the decimal point for each quantity that it computes. The estimator gain K_1 is

computed as

$$\begin{aligned} K_1 &= P_0(P_0 + R_1)^{-1} \\ &= (6)\left(\frac{1}{6}\right) \\ &= (6)(0.167) \\ &= 1.002 \end{aligned} \tag{3.55}$$

If we use the third form of the covariance update in Equation (3.47) we obtain

$$\begin{aligned} P_1 &= (1 - K_1)P_0 \\ &= (-0.002)(6) \\ &= -0.012 \end{aligned} \tag{3.56}$$

The covariance after the first measurement is negative, which is physically impossible. However, if we use the first form of the covariance update in Equation (3.47) we obtain

$$\begin{aligned} P_1 &= (1 - K_1)P_0(1 - K_1) + K_1 R_1 K_1 \\ &= (1 - K_1)^2 P_0 + K_1^2 R_1 \\ &= 0 \end{aligned} \tag{3.57}$$

The reason we get zero is because $(1 - K_1)^2 = 0.000004$, but our computer retains only three digits to the right of the decimal point. Zero is the theoretically correct value of P_1. The form of the above expression for P_1 guarantees that it will never be negative, regardless of any numerical errors in P_0, R_1, and K_1.

$\nabla\nabla\nabla$

■ EXAMPLE 3.5

Suppose that a tank contains a concentration x_1 of chemical 1, and a concentration x_2 of chemical 2. You have some instrumentation that can detect the combined concentration $(x_1 + x_2)$ of the two chemicals, but your instrumentation cannot distinguish between the two chemicals. Chemical 2 is removed from the tank through a leaching process so that its concentration decreases by 1% from one measurement time to the next. The measurement equation is therefore given as

$$\begin{aligned} y_k &= x_1 + 0.99^{k-1}x_2 + v_k \\ &= \begin{bmatrix} 1 & 0.99^{k-1} \end{bmatrix} x + v_k \end{aligned} \tag{3.58}$$

where v_k is the measurement noise, which is a zero-mean random variable with a variance of $R = 0.01$. Suppose that $x_1 = 10$ and $x_2 = 5$. Further suppose that your initial estimates are $\hat{x}_1 = 8$ and $\hat{x}_2 = 7$, with an initial estimation-error variance P_0 that is equal to the identity matrix. A recursive least squares algorithm can be implemented as shown in Equation (3.47) to estimate the two concentrations. Figure 3.1 shows the estimate of x_1 and x_2 as

measurements are obtained, along with the variance of the estimation errors. It can be seen that after a couple dozen measurements the estimates become quite close to their true values of 10 and 5. The variances of the estimation errors asymptotically approach zero, which means that we have increasingly more confidence in our estimates as we obtain more measurements.

Figure 3.1 Parameter estimates and estimation variances for Example 3.5.

∇∇∇

3.3.2 Curve fitting

In this section, we will apply recursive least squares theory to the curve fitting problem. In the recursive curve fitting problem, we measure data one sample at a time (y_1, y_2, $\cdots$) and want to find the best fit of a curve to the data. The curve that we want to fit to the data could be constrained to be linear, or quadratic, or sinusoid, or some other shape, depending on the underlying problem.

■ **EXAMPLE 3.6**

Suppose that we want to fit a straight line to a set of data points. The linear data fitting problem can be written as

$$\begin{aligned} y_k &= x_1 + x_2 t_k + v_k \\ E(v_k^2) &= R_k \end{aligned} \tag{3.59}$$

t_k is the independent variable (perhaps time), y_k is the noisy data, and we want to find the linear relationship between y_k and t_k. In other words, we want to estimate the constants x_1 and x_2. The measurement matrix can be written as

$$H_k = \begin{bmatrix} 1 & t_k \end{bmatrix} \tag{3.60}$$

so that Equation (3.59) can be written as

$$y_k = H_k x + v_k \tag{3.61}$$

Our recursive estimator is initialized as

$$\begin{aligned}
\hat{x}_0 &= E(x) \\
\begin{bmatrix} \hat{x}_{1,0} \\ \hat{x}_{2,0} \end{bmatrix} &= \begin{bmatrix} E(x_1) \\ E(x_2) \end{bmatrix} \\
P_0 &= E[(x - \hat{x}_0)(x - \hat{x}_0)^T] \\
&= \begin{bmatrix} E[x_1 - \hat{x}_{1,0})^2] & E[(x_1 - \hat{x}_{1,0})(x_2 - \hat{x}_{2,0})] \\ E[(x_1 - \hat{x}_{1,0})(x_2 - \hat{x}_{2,0})] & E[x_2 - \hat{x}_{2,0})^2] \end{bmatrix}
\end{aligned} \tag{3.62}$$

The recursive estimate of the two-element vector x is then obtained from Equation (3.47) as follows:

For $k = 1, 2, \cdots$,

$$\begin{aligned}
K_k &= P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \\
\hat{x}_k &= \hat{x}_{k-1} + K_k (y_k - H_k \hat{x}_{k-1}) \\
P_k &= (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T
\end{aligned} \tag{3.63}$$

$\nabla\nabla\nabla$

■ EXAMPLE 3.7

Suppose that we know *a priori* that the underlying data is a quadratic function of time. In this case, we have a quadratic data fitting problem. For example, suppose we are measuring the altitude of a free-falling object. We know from our understanding of physics that altitude r is a function of the acceleration due to gravity, the initial altitude and velocity of the object r_0 and v_0, and time t, as given by the equation $r = r_0 + v_0 t + (a/2)t^2$. So if we measure r at various time instants and fit a quadratic to the resulting r versus t curve, then we have an estimate of the parameters r_0, v_0, and $a/2$. In general, the quadratic data fitting problem can be written as

$$\begin{aligned}
y_k &= x_1 + x_2 t_k + x_3 t_k^2 + v_k \\
E(v_k^2) &= R_k
\end{aligned} \tag{3.64}$$

t_k is the independent variable, y_k is the noisy measurement, and we want to find the quadratic relationship between y_k and t_k. In other words, we want to estimate the constants x_1, x_2, and x_3. The measurement matrix can be written as

$$H_k = \begin{bmatrix} 1 & t_k & t_k^2 \end{bmatrix} \tag{3.65}$$

so that Equation (3.64) can be written as

$$y_k = H_k x + v_k \tag{3.66}$$

Our recursive estimator is initialized as

$$\begin{aligned}
\hat{x}_0 &= E(x) \\
P_0 &= E[(x - \hat{x}_0)(x - \hat{x}_0)^T]
\end{aligned} \tag{3.67}$$

where P_0 is a 3×3 matrix. The recursive estimate of the three-element vector x is then obtained from Equation (3.47) as follows:

For $k = 1, 2, \cdots$,

$$\begin{aligned} K_k &= P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \\ P_k &= (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T \end{aligned} \tag{3.68}$$

$\nabla\nabla\nabla$

3.4 WIENER FILTERING

In this section, we will give a brief review of Wiener filtering. The rest of this book does not assume any knowledge on the reader's part of Wiener filtering. However, Wiener filtering is important from a historical perspective, and it still has a lot of applications in signal processing and communication theory. But since it is not used much for state estimation anymore, the reader can safely skip this section if desired.

Wiener filtering addresses the problem of designing a linear, time-invariant filter to extract a signal from noise, approaching the problem from the frequency domain perspective. Norbert Wiener invented his filter as part of the World War II effort for the United States. He published his work on the problem in 1942, but it was not available to the public until 1949 [Wie64]. His book was known as the "yellow peril" because of its mathematical difficulty and its yellow cover [Deu65, page 176]. Andrey Kolmogorov actually solved a more general problem earlier (1941), and Mark Krein also worked on the same problem (1945). Kolmogorov's and Krein's work was independent of Wiener's work, and Wiener acknowledges that Kolmogorov's work predated his own work [Wie56]. However, Kolmogorov's and Krein's work did not become well known in the Western world until later, since it was published in Russian [Kol41]. A nontechnical account of Wiener's work is given in his autobiography [Wie56].

To set up the presentation of the Wiener filter, we first need to ask the following question: How does the power spectrum of a stochastic process $x(t)$ change when it goes through an LTI system with impulse response $g(t)$? The output $y(t)$ of the system is given by the convolution of the impulse response with the input:

$$y(t) = g(t) * x(t) \tag{3.69}$$

Since the system is time-invariant, a time shift in the input results in an equal time shift in the output:

$$y(t + \alpha) = g(t) * x(t + \alpha) \tag{3.70}$$

Multiplying the above two equations and writing out the convolutions as integrals gives

$$y(t)y(t + \alpha) = \int g(\tau)x(t - \tau)\, d\tau \int g(\gamma)x(t + \alpha - \gamma)\, d\gamma \tag{3.71}$$

Taking the expected value of both sides of the above equation gives the autocorrelation of $y(t)$ as a function of the autocorrelation of $x(t)$:

$$E[y(t)y(t + \alpha)] = \int\int g(\tau)g(\gamma)E[x(t - \tau)x(t + \alpha - \gamma)]\, d\tau\, d\gamma \tag{3.72}$$

which we will write in shorthand notation as

$$R_y(\alpha) = \int\int g(\tau)g(\gamma)R_x(\alpha+\tau-\gamma)\,d\tau\,d\gamma \tag{3.73}$$

Now we take the Fourier transform of the above equation to obtain

$$\int R_y(\alpha)e^{-j\omega\alpha}\,d\alpha = \int\int\int g(\tau)g(\gamma)R_x(\alpha+\tau-\gamma)e^{-j\omega\alpha}\,d\tau\,d\gamma\,d\alpha \tag{3.74}$$

Now we define a new variable of integration $\beta = \alpha+\tau-\gamma$ and replace α in the above equation to obtain

$$\begin{aligned} S_y(\omega) &= \int\int\int g(\tau)g(\gamma)R_x(\beta)e^{-j\omega\beta}e^{-j\omega\gamma}e^{j\omega\tau}\,d\tau\,d\gamma\,d\beta \\ &= G(-\omega)G(\omega)S_x(\omega) \end{aligned} \tag{3.75}$$

In other words, the power spectrum of the output $y(t)$ is a function of the Fourier transform of the impulse response of the system, $G(\omega)$, and the power spectrum of the input $x(t)$.

Now we can state our problem as follows: Design a stable LTI filter to extract a signal from noise. The quantities of interest in this problem are given as

$$\begin{aligned} x(t) &= \text{noise free signal} \\ v(t) &= \text{additive noise} \\ g(t) &= \text{filter impulse response (to be designed)} \\ \hat{x}(t) &= \text{output of filter [estimate of } x(t)] \\ e(t) &= \text{estimation error} \\ &= x(t)-\hat{x}(t) \end{aligned} \tag{3.76}$$

Figure 3.2 Wiener filter representation.

These quantities are represented in Figure 3.2, from which we see that

$$\begin{aligned} \hat{x}(t) &= g(t) * [x(t)+v(t)] \\ \hat{X}(\omega) &= G(\omega)[X(\omega)+V(\omega)] \\ E(\omega) &= X(\omega)-\hat{X}(\omega) \\ &= X(\omega)-G(\omega)[X(\omega)+V(\omega)] \\ &= [1-G(\omega)]X(\omega)-G(\omega)V(\omega) \end{aligned} \tag{3.77}$$

We see that the error signal $e(t)$ is the superposition of the system $[1-G(\omega)]$ acting on the signal $x(t)$, and the system $G(\omega)$ acting on the signal $v(t)$. Therefore, from Equation (3.75), we obtain

$$S_e(\omega) = [1-G(\omega)][1-G(-\omega)]S_x(\omega) - G(\omega)G(-\omega)S_v(\omega) \tag{3.78}$$

The variance of the estimation error is obtained from Equation (2.92) as

$$E[e^2(t)] = \frac{1}{2\pi}\int S_e(\omega)\,d\omega \tag{3.79}$$

To find the optimal filter $G(\omega)$ we need to minimize $E[e^2(t)]$, which means that we need to know $S_x(\omega)$ and $S_v(\omega)$, the statistical properties of the signal $x(t)$ and the noise $v(t)$.

3.4.1 Parametric filter optimization

In order to simplify the problem of the determination of the optimal filter $G(\omega)$, we can assume that the optimal filter is a first-order, low-pass filter (stable and causal[3]) with a bandwidth $1/T$ to be determined by parametric optimization.

$$G(\omega) = \frac{1}{1+Tj\omega} \tag{3.80}$$

This may not be a valid assumption, but it reduces the problem to a parametric optimization problem. In order to simplify the problem further, suppose that $S_x(\omega)$ and $S_v(\omega)$ are in the following forms.

$$\begin{aligned} S_x(\omega) &= \frac{2\sigma^2\beta}{\omega^2+\beta^2} \\ S_v(\omega) &= A \end{aligned} \tag{3.81}$$

In other words, the noise $v(t)$ is white. From Equation (3.78) we obtain

$$\begin{aligned} S_e(\omega) &= \left(\frac{Tj\omega}{1+Tj\omega}\right)\left(\frac{-Tj\omega}{1-Tj\omega}\right)\left(\frac{2\sigma^2\beta}{\omega^2+\beta^2}\right) - \\ &\quad \left(\frac{1}{1+Tj\omega}\right)\left(\frac{1}{1-Tj\omega}\right)A \end{aligned} \tag{3.82}$$

Now we can substitute $S_e(\omega)$ in Equation (3.79) and differentiate with respect to T to find

$$T_{\text{opt}} = \frac{\sqrt{A}}{\sigma\sqrt{2\beta} - \beta\sqrt{A}} \tag{3.83}$$

■ **EXAMPLE 3.8**

If $A = \sigma = \beta = 1$ then the optimal time constant of the filter is computed as

$$\begin{aligned} T &= \frac{1}{\sqrt{2}-1} \\ &\approx 2.4 \end{aligned} \tag{3.84}$$

and the optimal filter is given as

$$G(\omega) = \frac{1}{1+j\omega T}$$

[3]A causal system is one whose output depends only on present and future inputs. Real-world systems are always causal, but a filter that is used for postprocessing may be noncausal.

$$
\begin{aligned}
&= \frac{1/T}{1/T + j\omega} \\
g(t) &= \frac{1}{T} e^{-t/T} \quad t \geq 0
\end{aligned}
\tag{3.85}
$$

Converting this filter to the time domain results in

$$
\dot{\hat{x}} = \frac{1}{T}(-\hat{x} + y) \tag{3.86}
$$

$\nabla\nabla\nabla$

3.4.2 General filter optimization

Now we take a more general approach to find the optimal filter. The expected value of the estimation error can be computed as

$$
\begin{aligned}
e(t) &= x(t) - \hat{x}(t) \\
e^2(t) &= x^2(t) - 2x(t)\hat{x}(t) + \hat{x}^2(t) \\
&= x^2(t) - 2x(t) \int g(u)[x(t-u) + v(t-u)]\, du + \\
&\quad \int\int g(u)g(\gamma)[x(t-u) + v(t-u)] \times \\
&\quad [x(t-v) + v(t-v)]\, du\, d\gamma \\
E[e^2(t)] &= E[x^2(t)] - 2\int g(u)R_x(u)\, du + \\
&\quad \int\int g(u)g(\gamma)[R_x(u-v) + R_v(u-v)]\, du\, d\gamma
\end{aligned}
\tag{3.87}
$$

Now we can use a calculus of variations approach [Fom00, Wei74] to find the filter $g(t)$ that minimizes $E[e^2(t)]$. Replace $g(t)$ in the above equation with $g(t) + \epsilon\eta(t)$, where ϵ is some small number, and $\eta(t)$ is an arbitrary perturbation in $g(t)$. The calculus of variations says that we can minimize $E(e^2(t))$ by setting

$$
\left.\frac{\partial E(e^2(t))}{\partial \epsilon}\right|_{\epsilon=0} = 0 \tag{3.88}
$$

and thus solve for the optimal $g(t)$. From Equation (3.87) we can write

$$
\begin{aligned}
R_e(0) &= R_x(0) - 2\int [g(u) + \epsilon\eta(u)]R_x(u)\, du + \\
&\quad \int\int [g(u) + \epsilon\eta(u)][g(\gamma) + \epsilon\eta(\gamma)][R_x(u-\gamma) + R_v(u-\gamma)]\, du\, d\gamma
\end{aligned}
\tag{3.89}
$$

Taking the partial derivative with respect to ϵ gives

$$
\begin{aligned}
\frac{\partial R_e(0)}{\partial \epsilon} &= -2\int \eta(u)R_x(u)\,du + \\
&\quad \int\int [\eta(u)g(\gamma) + \eta(\gamma)g(u) + 2\epsilon\eta(u)\eta(\gamma)] \times \\
&\quad [R_x(u-v) + R_v(u-\gamma)]\,du\,d\gamma \\
\left.\frac{\partial R_e(0)}{\partial \epsilon}\right|_{\epsilon=0} &= -2\int \eta(\tau)R_x(\tau)\,d\tau + \\
&\quad \int\int \eta(\tau)g(\gamma)[R_x(\tau-\gamma) + R_v(\tau-\gamma)]\,d\tau\,d\gamma + \\
&\quad \int\int \eta(\tau)g(u)[R_x(u-\tau) + R_w(u-\tau)]\,d\tau\,du \qquad (3.90)
\end{aligned}
$$

Now recall from Equation (2.87) that $R_x(\tau - u) = R_x(u - \tau)$ [i.e., $R_x(\tau)$ is even] if $x(t)$ is stationary. In this case, the above equation can be written as

$$
\begin{aligned}
0 &= -2\int \eta(\tau)R_x(\tau)\,d\tau + \\
&\quad 2\int\int \eta(\tau)g(u)[R_x(u-\tau) + R_v(u-\tau)]\,d\tau\,du \qquad (3.91)
\end{aligned}
$$

This gives the necessary condition for the optimality of the filter $g(t)$ as follows:

$$
\int \eta(\tau)\left[-R_x(\tau) + \int g(u)[R_x(u-\tau) + R_v(u-\tau)]\,du\right]d\tau = 0 \qquad (3.92)
$$

We need to solve this for $g(t)$ to find the optimal filter.

3.4.3 Noncausal filter optimization

If we do not have any restrictions on causality of our filter, then $g(t)$ can be nonzero for $t < 0$, which means that our perturbation $\eta(t)$ can also be nonzero for $t < 0$. This means that the quantity inside the square brackets in Equation (3.92) must be zero. This results in

$$
\begin{aligned}
R_x(\tau) &= \int g(u)[R_x(u-\tau) + R_v(u-\tau)]\,du \\
&= g(\tau) * [R_x(\tau) + R_v(\tau)] \\
S_x(\omega) &= G(\omega)[S_x(\omega) + S_v(\omega)] \\
G(\omega) &= \frac{S_x(\omega)}{S_x(\omega) + S_v(\omega)} \qquad (3.93)
\end{aligned}
$$

The transfer function of the optimal filter is the ratio of the power spectrum of the signal $x(t)$ to the sum of the power spectrums of $x(t)$ and the noise $v(t)$.

■ **EXAMPLE 3.9**

Consider the system discussed in Example 3.8 with $A = \beta = \sigma = 1$. The signal and noise power spectra are given as

$$\begin{aligned} S_x(\omega) &= \frac{2}{\omega^2+1} \\ S_v(\omega) &= 1 \end{aligned} \tag{3.94}$$

From this we obtain the optimal noncausal filter from Equation (3.93) as

$$\begin{aligned} G(\omega) &= \frac{2}{\omega^2+3} \\ &= \frac{1}{\sqrt{3}}\left(\frac{2\sqrt{3}}{\omega^2+3}\right) \\ g(t) &= \frac{1}{\sqrt{3}}e^{-\sqrt{3}|t|} \\ &\approx 0.58e^{-0.58|t|}, \qquad t \in [-\infty, \infty] \end{aligned} \tag{3.95}$$

In order to find a time domain representation of the filter, we perform a partial fraction expansion of $G(\omega)$ to find the causal part and the anticausal[4] part of the filter[5]:

$$G(\omega) = \underbrace{\frac{1}{\sqrt{3}(j\omega+\sqrt{3})}}_{\text{causal filter}} + \underbrace{\frac{1}{\sqrt{3}(-j\omega+\sqrt{3})}}_{\text{anticausal filter}} \tag{3.96}$$

From this we see that

$$\begin{aligned} \hat{X}(\omega) &= \frac{1}{\sqrt{3}(j\omega+\sqrt{3})}Y(s) - \frac{1}{\sqrt{3}(j\omega-\sqrt{3})}Y(s) \\ &= \hat{X}_c(\omega) + \hat{X}_a(\omega) \end{aligned} \tag{3.97}$$

$\hat{X}_c(\omega)$ and $\hat{X}_a(\omega)$ (defined by the above equation) are the causal and anticausal part of $\hat{X}(\omega)$, respectively. In the time domain, this can be written as

$$\begin{aligned} \hat{x}(t) &= \hat{x}_c(t) + \hat{x}_a(t) \\ \dot{\hat{x}}_c &= -\sqrt{3}\hat{x}_c + y/\sqrt{3} \\ \dot{\hat{x}}_a &= \sqrt{3}\hat{x}_a - y/\sqrt{3} \end{aligned} \tag{3.98}$$

The $\dot{\hat{x}}_c$ equation runs forward in time and is therefore causal and stable. The $\dot{\hat{x}}_a$ equation runs backward in time and is therefore anticausal and stable. (If it ran forward in time, it would be unstable.)

∇∇∇

[4]An anticausal system is one whose output depends only on present and future inputs.
[5]The MATLAB function RESIDUE performs partial fraction expansions.

3.4.4 Causal filter optimization

If we require a causal filter for signal estimation, then $g(t) = 0$ for $t < 0$, and the perturbation $\eta(t)$ must be equal to 0 for $t < 0$. In this case, Equation (3.92) gives

$$R_x(\tau) - \int g(u)[R_x(u-\tau) + R_v(u-\tau)]\,du = 0, \qquad t \geq 0 \tag{3.99}$$

The initial application of this equation was in the field of astrophysics in 1894 [Sob63]. Explicit solutions were thought to be impossible, but Norbert Wiener and Eberhard Hopf became instantly famous when they solved this equation in 1931. Their solution was so impressive that the equation became known as the Wiener–Hopf equation.

To solve Equation (3.99), postulate some function $a(t)$ that is arbitrary for $t < 0$, but is equal to 0 for $t \geq 0$. Then we obtain

$$\begin{aligned} R_x(\tau) - \int g(u)[R_x(u-\tau) + R_v(u-\tau)]\,du &= a(\tau) \\ S_x(\omega) - G(\omega)[S_x(\omega) + S_v(\omega)] &= A(\omega) \end{aligned} \tag{3.100}$$

For ease of notation, make the following definition:

$$S_{xv}(\omega) = S_x(\omega) + S_v(\omega) \tag{3.101}$$

Then Equation (3.100) becomes

$$S_x(\omega) - G(\omega)S_{xv}^{+}(\omega)S_{xv}^{-}(\omega) = A(\omega) \tag{3.102}$$

where $S_{xv}^{+}(\omega)$ is the part of $S_{xv}(\omega)$ that has all its poles and zeros in the LHP (and hence corresponds to a causal time function), and $S_{xv}^{-}(\omega)$ is the part of $S_{xv}(\omega)$ that has all its poles and zeros in the RHP (and hence corresponds to an anticausal time function). Equation (3.102) can be written as

$$G(\omega)S_{xv}^{+}(\omega) = \frac{S_x(\omega)}{S_{xv}^{-}(\omega)} - \frac{A(\omega)}{S_{xv}^{-}(\omega)} \tag{3.103}$$

The term on the left side corresponds to a causal time function [assuming that $g(t)$ is stable]. The last term on the right side corresponds to an anticausal time function. Therefore,

$$\begin{aligned} G(\omega)S_{xv}^{+}(\omega) &= \text{causal part of } \frac{S_x(\omega)}{S_{xv}^{-}(\omega)} \\ G(\omega) &= \frac{1}{S_{xv}^{+}(\omega)}\left[\text{causal part of } \frac{S_x(\omega)}{S_{xv}^{-}(\omega)}\right] \end{aligned} \tag{3.104}$$

This gives the TF of the optimal causal filter.

■ **EXAMPLE 3.10**

Consider the system discussed in Section 3.4.1 with $A = \beta = \sigma = 1$. This was also discussed in Example 3.9. For this example we have

$$S_x(\omega) = \frac{2}{\omega^2 + 1}$$

$$
\begin{aligned}
S_v(\omega) &= 1 \\
S_{xv}(\omega) &= \frac{\omega^2+3}{\omega^2+1}
\end{aligned}
\tag{3.105}
$$

Splitting this up into its causal and anticausal factors gives

$$
\begin{aligned}
S_{xv}(\omega) &= \underbrace{\left(\frac{j\omega+\sqrt{3}}{j\omega+1}\right)}_{S_{xv}^{+}(\omega)}\underbrace{\left(\frac{-j\omega+\sqrt{3}}{-j\omega+1}\right)}_{S_{xv}^{-}(\omega)} \\
\frac{S_x(\omega)}{S_{xv}^{-}(\omega)} &= \frac{2(-j\omega+1)}{(\omega^2+1)(-j\omega+\sqrt{3})} \\
&= \frac{2}{(-j\omega+\sqrt{3})(j\omega+1)} \\
&= \underbrace{\frac{\sqrt{3}-1}{j\omega+1}}_{\text{causal part}} + \underbrace{\frac{\sqrt{3}-1}{-j\omega+\sqrt{3}}}_{\text{anticausal part}}
\end{aligned}
\tag{3.106}
$$

Equation (3.104) gives

$$
\begin{aligned}
G(\omega) &= \left(\frac{j\omega+1}{j\omega+\sqrt{3}}\right)\left(\frac{\sqrt{3}-1}{j\omega+1}\right) \\
&= \frac{\sqrt{3}-1}{j\omega+\sqrt{3}} \\
g(t) &= (\sqrt{3}-1)e^{-\sqrt{3}t}, \qquad t \geq 0
\end{aligned}
\tag{3.107}
$$

This gives the TF and impulse response of the optimal filter when causality is required.

$\nabla\nabla\nabla$

3.4.5 Comparison

Comparing the three examples of optimal filter design presented in this section (Examples 3.8, 3.9, and 3.10), it can be shown that the mean square errors of the filter are as follows [Bro96]:

- Parameter optimization method: $E[e^2(t)] = 0.914$
- Causal Wiener filter: $E[e^2(t)] = 0.732$
- Noncausal Wiener filter: $E[e^2(t)] = 0.577$

As expected, the estimation error decreases when we have fewer constraints on the filter. However, the removal of constraints makes the filter design problem more difficult. The Wiener filter is not very amenable to state estimation because of difficulty in extension to MIMO problems with state variable descriptions, and difficulty in application to signals with time-varying statistical properties.

3.5 SUMMARY

In this chapter we discussed least squares estimation in a couple of different contexts. First we derived a method for estimating a constant vector on the basis of several noisy measurements of that vector. In fact, the measurements do not have to be direct measurements of the constant vector, but they can be measurements of some linear combination of the elements of the constant vector. In addition, the noise associated with each measurement does not have to be the same. The least squares estimation technique that we derived assumed that we the measurement noise is zero-mean and white (uncorrelated with itself from one time step to the next), and that we know the variance of the measurement noise. We then extended our least squares estimator to a recursive formulation, wherein the computational effort remains the same at each time step regardless of the total number of measurements that we have processed. Least squares estimation of a constant vector forms a large part of the foundation for the Kalman filter, which we will derive later in this book.

In Section 3.4, we took a brief segue into Wiener filtering, which is a method of estimating a time-varying signal that is corrupted by noise. The Wiener filter is based on frequency domain analyses, whereas the Kalman filter that we derive later is based on time domain analyses. Nevertheless, both filters are optimal under their own assumptions. Some problems are solvable by both the Wiener and Kalman filter methods, in which case both methods give the same result.

PROBLEMS

Written exercises

3.1 In Equation (3.6) we computed the partial derivative of our cost function with respect to our estimate and set the result equal to 0 to solve for the optimal estimate. However, the solution minimizes the cost function only if the second derivative of the cost function with respect to the estimate is positive semidefinite. Find the second derivative of the cost function and show that it is positive semidefinite.

3.2 Prove that the matrix P_k that is computed from Equation (3.25) will always be positive definite if P_{k-1} and R_k are positive definite.

3.3 Consider the recursive least squares estimator of Equations (3.28)-(3.30). If zero information about the initial state is available, then $P_0 = \infty I$. Suppose that you have a system like this with $H_k = 1$. What will be the values of K_1 and P_1?

3.4 Consider a battery with a completely unknown voltage ($P_0 = \infty$). Two independent measurements of the voltage are taken to estimate the voltage, the first with a variance of 1, and the second with a variance of 4.

a) Write the weighted least squares voltage estimate in terms of the two measurements y_1 and y_2.

b) If weighted least squares is used to estimate the voltage, what is the variance of voltage estimate after the first measurement? What is the variance of the voltage estimate after the second measurement?

c) If the voltage is estimated as $(y_1 + y_2)/2$, an unweighted average of the measurements, what is the variance of the voltage estimate?

3.5 Consider a battery whose voltage is a random variable with a variance of 1. Two independent measurements of the voltage are taken to estimate the voltage, the first with a variance of 1, and the second with a variance of 4.

a) Write the weighted least squares voltage estimate in terms of the initial estimate $\hat{x}_0$ and the two measurements y_1 and y_2.

b) If weighted least squares is used to estimate the voltage, what is the variance of voltage estimate after the first measurement? What is the variance of the voltage estimate after the second measurement?

3.6 Suppose that $\{x_1, x_2, \cdots, x_n\}$ is a set of random variables, each with mean $\bar{x}$ and variance σ^2. Further suppose that $E[(x_i - \bar{x})(x_j - \bar{x})] = 0$ for $i \neq j$. We estimate $\bar{x}$ and σ^2 as follows.

$$\begin{aligned} \hat{\bar{x}} &= \frac{1}{n}\sum_{i=1}^{n} x_i \\ \hat{\sigma}^2 &= \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\bar{x}})^2 \end{aligned}$$

a) Is $\hat{\bar{x}}$ an unbiased estimate of $\bar{x}$? That is, is $E(\hat{\bar{x}}) = \bar{x}$?

b) Find $E(x_i x_j)$ in terms of $\bar{x}$ and σ^2 for both $i = j$ and $i \neq j$.

c) Is $\hat{\sigma}^2$ an unbiased estimate of σ^2? That is, is $E(\hat{\sigma}^2) = \sigma^2$? If not, how should we change $\hat{\sigma}^2$ to make it an unbiased estimate of σ^2?

3.7 Suppose a scalar signal has the values 1, 2, and 3. Consider three different estimates of this time-varying signal. The first estimate is 3, 4, 1. The second estimate is 1, 2, 6. The third estimate is 5, 6, 7. Create a table showing the RMS value, average absolute error, and standard deviation of the error of each estimate. Which estimate results in the error with the smallest RMS value? Which estimate results in the error with the smallest infinity-norm? Which estimate gives the error with the smallest standard deviation? Which estimate do you think is best from an intuitive point of view? Which estimate do you think is worst from an intuitive point of view?

3.8 Suppose a random variable x has the pdf $f(x)$ given in Figure 3.3.

a) x can be estimated by taking the median of its pdf. That is, $\hat{x}$ is the solution to the equation

$$\int_{-\infty}^{\hat{x}} f(x)\,dx = \int_{\hat{x}}^{\infty} f(x)\,dx$$

Find the median estimate of x.

b) x can be estimated by taking the mode of its pdf. That is,

$$\hat{x} = \arg\max f(x)$$

Find the mode estimate of x.

c) x can be estimated by computing its mean. That is,

$$\hat{x} = \int_{-\infty}^{\infty} x f(x)\,dx$$

Find the mean of x.

d) x can be estimated by computing the minimax value. That is,

$$\hat{x} = \min \max_{x} |x - \hat{x}|$$

Find the minimax estimate of x.

Figure 3.3 pdf for Problem 3.8.

3.9 Suppose you are responsible for increasing the tracking accuracy of a radar system. You presently have a radar that has a measurement variance of 10. For equal cost you could either: (a) optimally combine the present radar system with a new radar system that has a measurement variance of 6; or, (b) optimally combine the present radar system with two new radar systems that both have the same performance as the original system [May79]. Which would you propose to do? Why?

3.10 Consider the differential equation

$$\dot{x} + 3x = u$$

If the input $u(t)$ is an impulse, there are two solutions $x(t)$ that satisfy the differential equation. One solution is causal and stable, the other solution is anticausal and unstable. Find the two solutions.

3.11 Suppose a signal $x(t)$ with power spectral density

$$S_x(s) = \frac{1 - s^2}{s^4 - 5s^2 + 4}$$

is corrupted with additive white noise $v(t)$ with a power spectral density $S_v(s) = 1$.

a) Find the optimal noncausal Wiener filter to extract the signal from the noise corrupted signal.

b) Find the optimal causal Wiener filter to extract the signal from the noise corrupted signal.

3.12 A system has the transfer function

$$G(s) = \frac{1}{s-3}$$

If the input is an impulse, there are two solutions for the output $x(t)$ that satisfy the transfer function. One solution is causal and unstable, the other solution is anticausal and stable. Find the two solutions.

Computer exercises

3.13 The production of steel in the United States between 1946 and 1956 was 66.6, 84.9, 88.6, 78.0, 96.8, 105.2, 93.2, 111.6, 88.3, 117.0, and 115.2 million tons [Sor80]. Find the least squares fit to these data using (a) linear curve fit; (b) quadratic curve fit; (c) cubic curve fit; (d) quartic curve fit. For each case give the following: (1) a plot of the original data along with the least squares curve; (2) the RMS error of the least squares curve; (3) the prediction of steel production in 1957.

3.14 Implement the Wiener filters for the three examples given in Section 3.4 and verify the results shown in Section 3.4.5. Hint: Example 8.6 shows that if $\dot{x} = -x + w$ where $w(t)$ is white noise with a variance of $Q_c = 2$, then

$$S_x(\omega) = \frac{2}{\omega^2 + 1}$$

From Sections 1.4 and 8.1 we see that this system can be simulated as

$$\begin{aligned} x(t+\Delta t) &= e^{-\Delta t}x(t) + w(t)\sqrt{Q_c \Delta t} \\ y(t) &= x(t) + v(t)\sqrt{R_c/\Delta t} \end{aligned}$$

where $w(t)$ and $v(t)$ are independent zero-mean, unity variance random variables.

CHAPTER 4

Propagation of states and covariances

In this chapter, we will begin with our mathematical description of a dynamic system, and then derive the equations that govern the propagation of the state mean and covariance. The material presented in this chapter is fundamental to the state estimation algorithm (the Kalman filter) that we will derive in Chapter 5.

Section 4.1 covers discrete-time systems. Section 4.2 covers sampled-data systems, which are the most common types of systems found in the real world. In this type of system, the system dynamics are described by continuous-time differential equations, but the control and measurement signals are discrete time (e.g., control based on a digital computer and measurements obtained at discrete times). Section 4.3 covers continuous-time systems.

4.1 DISCRETE-TIME SYSTEMS

Suppose we have the following linear discrete-time system:

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \tag{4.1}$$

where u_k is a known input and w_k is Gaussian zero-mean white noise with covariance Q_k. How does the mean of the state x_k change with time? If we take the expected value of both sides of Equation (4.1) we obtain

$$\begin{aligned} \bar{x}_k &= E(x_k) \\ &= F_{k-1}\bar{x}_{k-1} + G_{k-1}u_{k-1} \end{aligned} \tag{4.2}$$

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

How does the covariance of x_k change with time? We can use Equations (4.1) and (4.2) to obtain

$$\begin{aligned}(x_k - \bar{x}_k)(\cdots)^T &= (F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} - \bar{x}_k)(\cdots)^T \\ &= [F_{k-1}(x_{k-1} - \bar{x}_{k-1}) + w_{k-1}][\cdots]^T \\ &= F_{k-1}(x_{k-1} - \bar{x}_{k-1})(x_{k-1} - \bar{x}_{k-1})^T F_{k-1}^T + w_{k-1}w_{k-1}^T + \\ & F_{k-1}(x_{k-1} - \bar{x}_{k-1})w_{k-1}^T + w_{k-1}(x_{k-1} - \bar{x}_{k-1})^T F_{k-1}^T \end{aligned} \tag{4.3}$$

We therefore obtain the covariance of x_k as the expected value of the above expression. Since $(x_{k-1} - \bar{x}_{k-1})$ is uncorrelated with w_{k-1}, we obtain

$$\begin{aligned} P_k &= E\left[(x_k - \bar{x}_k)(\cdots)^T\right] \\ &= F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1} \end{aligned} \tag{4.4}$$

This is called a discrete-time Lyapunov equation, or a Stein equation [Ste52]. We will see in the next chapter that Equations (4.2) and (4.4) are fundamental in the derivation of the Kalman filter.

It is interesting to consider the conditions under which the discrete-time Lyapunov equation has a steady-state solution. That is, suppose that $F_k = F$ is a constant, and $Q_k = Q$ is a constant. Then we have the following theorem, whose proof can be found in [Kai00, Appendix D].

Theorem 21 *Consider the equation $P = FPF^T + Q$ where F and Q are real matrices. Denote by $\lambda_i(F)$ the eigenvalues of the F matrix.*

1. *A unique solution P exists if and only if $\lambda_i(F)\lambda_j(F) \neq 1$ for all i, j. This unique solution is symmetric.*

2. *Note that the above condition includes the case of stable F, because if F is stable then all of its eigenvalues are less than one in magnitude, so $\lambda_i(F)\lambda_j(F) \neq 1$ for all i, j. Therefore, we see that if F is stable then the discrete-time Lyapunov equation has a solution P that is unique and symmetric. In this case, the solution can be written as*

$$P = \sum_{i=0}^{\infty} F^i Q (F^T)^i \tag{4.5}$$

3. *If F is stable and Q is positive (semi)definite, then the unique solution P is symmetric and positive (semi)definite.*

4. *If F is stable, Q is positive semidefinite, and $(F, Q^{1/2})$ is controllable, then P is unique, symmetric, and positive definite. Note that $Q^{1/2}$, the square root of Q, is defined here as any matrix such that $Q^{1/2}(Q^{1/2})^T = Q$.*

Now let us look at the solution of the linear system of Equation (4.1):

$$x_k = F_{k,0}x_0 + \sum_{i=0}^{k-1} (F_{k,i+1}w_i + F_{k,i+1}G_i u_i) \tag{4.6}$$

The matrix $F_{k,i}$ is the state transition matrix of the system and is defined as

$$F_{k,i} = \begin{cases} F_{k-1}F_{k-2}\cdots F_i & k > i \\ I & k = i \\ 0 & k < i \end{cases} \tag{4.7}$$

Notice from Equation (4.6) that x_k is a linear combination of x_0, $\{w_i\}$, and $\{u_i\}$. If the input sequence $\{u_i\}$ is known, then it is a constant and can be considered to be a sequence of Gaussian random variables with zero covariance. If x_0 and $\{w_i\}$ are unknown but are Gaussian random variables, then x_k in Equation (4.6) is a linear combination of Gaussian random variables. Therefore, x_k is itself a Gaussian random variable (see Example 2.4). But we computed the mean and covariance of x_k in Equations (4.2) and (4.4). Therefore

$$x_k \sim N(\bar{x}_k, P_k) \tag{4.8}$$

This completely characterizes x_k in a statistical sense since a Gaussian random variable is completely characterized by its mean and covariance.

■ **EXAMPLE 4.1**

A linear system describing the population of a predator $x(1)$ and that of its prey $x(2)$ can be written as

$$\begin{aligned} x_{k+1}(1) &= x_k(1) - 0.8x_k(1) + 0.4x_k(2) + w_k(1) \\ x_{k+1}(2) &= x_k(2) - 0.4x_k(1) + u_k + w_k(2) \end{aligned} \tag{4.9}$$

In the first equation, we see that the predator population causes itself to decrease because of overcrowding, but the prey population causes the predator population to increase. In the second equation, we see that the prey population decreases due to the predator population and increases due to an external food supply u_k. The populations are also subject to random disturbances (with respective variances 1 and 2) due to environmental factors. This system can be written in state-space form as

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 0.2 & 0.4 \\ -0.4 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k + w_k \\ w_k &\sim (0, Q) \qquad Q = \mathrm{diag}(1, 2) \end{aligned} \tag{4.10}$$

Equations (4.2) and (4.4) describe how the mean and covariance of the populations change with time. Figure 4.1 depicts the two means and the two diagonal elements of the covariance matrix for the first few time steps when $u_k = 1$ and the initial conditions are set as $\bar{x}_0 = \begin{bmatrix} 10 & 20 \end{bmatrix}^T$ and $P_0 = \mathrm{diag}(40, 40)$. It is seen that the mean and covariance eventually reach steady-state values given by

$$\begin{aligned} \bar{x} &= (I - F)^{-1}Gu \\ &= \begin{bmatrix} 2.5 & 5 \end{bmatrix}^T \\ P &\approx \begin{bmatrix} 2.88 & 3.08 \\ 3.08 & 7.96 \end{bmatrix} \end{aligned} \tag{4.11}$$

The steady-state value of P can also be found directly (i.e., without simulation) using control system software.[1] Note that since F for this example is stable and Q is positive definite, Theorem 21 guarantees that P has a unique positive definite steady-state solution.

Figure 4.1 State means and variances for Example 4.1.

∇∇∇

In Equation (4.1), we showed the process noise directly entering the system dynamics. This is the convention that we use in this book. However, many times process noise is first multiplied by some matrix before it enters the system dynamics. That is,

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + L_{k-1}\tilde{w}_{k-1}, \qquad \tilde{w}_k \sim (0, \tilde{Q}_k) \tag{4.12}$$

How can we put this into the conventional form of Equation (4.1)? Notice that the rightmost term of Equation (4.12) has a covariance given by

$$\begin{aligned} E\left[(L_{k-1}\tilde{w}_{k-1})(L_{k-1}\tilde{w}_{k-1})^T\right] &= L_{k-1}E(\tilde{w}_{k-1}\tilde{w}_{k-1}^T)L_{k-1}^T \\ &= L_{k-1}\tilde{Q}_{k-1}L_{k-1}^T \end{aligned} \tag{4.13}$$

Therefore, Equation (4.12) is equivalent to the equation

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1}, \qquad w_k \sim (0, L_kQ_kL_k^T) \tag{4.14}$$

This idea is illustrated in Sections 7.3.1 and 7.3.2. The same type of transformation can be made with noisy measurement equations. That is, the measurement equation

$$y_k = H_kx_k + L_k\tilde{v}_k, \qquad \tilde{v}_k \sim (0, \tilde{R}_k) \tag{4.15}$$

is equivalent to the measurement equation

$$y_k = H_kx_k + v_k, \qquad v_k \sim (0, L_k\tilde{R}_kL_k^T) \tag{4.16}$$

[1]For example, we can use the MATLAB Control System Toolbox function DLYAP(F, Q).

4.2 SAMPLED-DATA SYSTEMS

Now we move on to sampled-data systems, which are the most frequently encountered systems in practice. A sampled-data system is a system whose dynamics are described by a continuous-time differential equation, but the input only changes at discrete time instants, because (for example) the input is generated by a digital computer. In addition, we are interested in estimating the state only at discrete time instants. We are interested in obtaining the mean and covariance of the state only at discrete time instants. The continuous-time dynamics are described as

$$\dot{x} = Ax + Bu + w \tag{4.17}$$

From Chapter 1 we know that the solution of $x(t)$ at some arbitrary time, say t_k, is given as

$$x(t_k) = e^{A(t_k - t_{k-1})}x(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}[B(\tau)u(\tau) + w(\tau)]\,d\tau \tag{4.18}$$

Now assume that $u(t) = u_k$ for $t \in [t_k, t_{k+1}]$; that is, the control $u(t)$ is piecewise constant.[2] If we make the definitions

$$\begin{aligned} \Delta t &= t_k - t_{k-1} \\ x_k &= x(t_k) \\ u_k &= u(t_k) \end{aligned} \tag{4.19}$$

then Equation (4.18) becomes

$$x_k = e^{A\Delta t}x_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}B(\tau)\,d\tau\, u_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}w(\tau)\,d\tau \tag{4.20}$$

Now if we define F_k and G_k as

$$\begin{aligned} F_k &= e^{A\Delta t} \\ G_k &= \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}B(\tau)\,d\tau \end{aligned} \tag{4.21}$$

then Equation (4.20) becomes

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}w(\tau)\,d\tau \tag{4.22}$$

$e^{A(t_k-\tau)}$ is the state transition matrix of the system from time τ to time t_k. Now take the mean of the above equation, remembering that $w(t)$ is zero-mean, to obtain

$$\begin{aligned} \bar{x}_k &= E(x_k) \\ &= F_{k-1}\bar{x}_{k-1} + G_{k-1}u_{k-1} \end{aligned} \tag{4.23}$$

[2]This assumes that a first-order hold is used for the control inputs. Other types of holds can be used in sampled data systems, but in this book we assume that first-order holds are used.

We can use the previous equations to obtain the covariance of the state as

$$\begin{aligned}
P_k &= E[(x_k - \bar{x}_k)(x_k - \bar{x}_k)^T] \\
&= E\left[\left(F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}w(\tau)d\tau - \bar{x}_k\right)\left(\cdots\right)^T\right] \\
&= F_{k-1}P_{k-1}F_{k-1}^T + E\left[\left(\int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}w(\tau)\,d\tau\right)\left(\cdots\right)^T\right] \qquad (4.24) \\
&= F_{k-1}P_{k-1}F_{k-1}^T + \int\int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}E\left[w(\tau)w^T(\alpha)\right]e^{A^T(t_k-\alpha)}\,d\tau\,d\alpha
\end{aligned}$$

Now, if we assume that $w(t)$ is continuous-time white noise with a covariance of $Q_c(t)$, we see that

$$E\left[w(\tau)w^T(\alpha)\right] = Q_c(\tau)\delta(\tau - \alpha) \qquad (4.25)$$

This means that we can use the sifting property of the impulse function (see Problem 4.10) to write Equation (4.24) as

$$\begin{aligned}
P_k &= F_{k-1}P_{k-1}F_{k-1}^T + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}Q_c(\tau)e^{A^T(t_k-\tau)}\,d\tau \\
&= F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1} \qquad (4.26)
\end{aligned}$$

where Q_{k-1} is defined by the above equation; that is,

$$Q_{k-1} = \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}Q_c(\tau)e^{A^T(t_k-\tau)}\,d\tau \qquad (4.27)$$

In general, it is difficult to calculate Q_{k-1}, but for small values of $(t_k - t_{k-1})$ we obtain

$$\begin{aligned}
e^{A(t_k-\tau)} &\approx I \text{ for } \tau \in [t_{k-1}, t_k] \\
Q_{k-1} &\approx Q_c(t_k)\Delta t \qquad (4.28)
\end{aligned}$$

■ EXAMPLE 4.2

Suppose we have a first-order, continuous-time dynamic system given by the equation

$$\begin{aligned}
\dot{x} &= fx + w \\
E[w(t)w(t+\tau)] &= q_c\delta(\tau) \qquad (4.29)
\end{aligned}$$

First-order equations can be used to describe many simple physical processes. For example, this equation describes the behavior of the current through a series RL circuit that is driven by a random voltage $w(t)$, where $f = -R/L$. Suppose we are interested in obtaining the mean and covariance of the state $x(t)$ every Δt time units; that is, $t_k - t_{k-1} = \Delta t$. For this simple scalar

example, we can explicitly calculate Q_{k-1} in Equation (4.27) as

$$\begin{aligned}
Q_{k-1} &= \int_{t_{k-1}}^{t_k} \exp[f(t_k-\tau)]q_c \exp[f(t_k-\tau)]\,d\tau \\
&= \exp(2ft_k)q_c \int_{t_{k-1}}^{t_k} \exp(-2f\tau)\,d\tau \\
&= \exp(2ft_k)q_c \left[\frac{\exp(-2ft_{k-1})-\exp(-2ft_k)}{2f}\right] \\
&= \frac{q_c}{2f}\left[\exp(2f(t_k-t_{k-1}))-1\right] \\
&= \frac{q_c}{2f}\left[\exp(2f\Delta t)-1\right] \qquad (4.30)
\end{aligned}$$

For small values of Δt, we can expand the above equation in a Taylor series around $\Delta t = 0$ to obtain

$$\begin{aligned}
Q_{k-1} &= \frac{q_c}{2f}\left[\exp(2f\Delta t)-1\right] \\
&= \frac{q_c}{2f}\left[\left(1+2f\Delta t+\frac{(2f\Delta t)^2}{2!}+\cdots\right)-1\right] \\
&\approx \frac{q_c}{2f}\left[1+2f\Delta t-1\right] \\
&= q_c\Delta t \qquad (4.31)
\end{aligned}$$

This matches Equation (4.28), which says that for small Δt we have $Q_{k-1} \approx q_c\Delta t$. The sampled mean of the state is computed from Equation (4.23) [noting that the control input in Equation (4.29) is zero] as

$$\begin{aligned}
\bar{x}_k &= F_{k-1}\bar{x}_{k-1} + G_{k-1}u_{k-1} \\
&= \exp\left[f(t_k-t_{k-1})\right]\bar{x}_{k-1} + 0 \\
&= \exp(f\Delta t)\bar{x}_{k-1} \\
&= \exp(kf\Delta t)\bar{x}_0 \qquad (4.32)
\end{aligned}$$

We see that if $f > 0$ (i.e., the system is unstable) then the mean $\bar{x}_k$ will increase without bound (unless $\bar{x}_0 = 0$). However, if $f < 0$ (i.e., the system is stable) then the mean $\bar{x}_k$ will decay to zero regardless of the value of $\bar{x}_0$. The sampled covariance of the state is computed from Equation (4.26) as

$$\begin{aligned}
P_k &= F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1} \\
&\approx (1+2f\Delta t)P_{k-1} + q_c\Delta t \\
P_k - P_{k-1} &= (2fP_{k-1}+q_c)\Delta t \qquad (4.33)
\end{aligned}$$

From the above equation, we can see that P_k reaches steady state (i.e., $P_k - P_{k-1} = 0$) when $P_{k-1} = -q_c/2f$, assuming that $f < 0$. On the other hand, if $f \geq 0$ then $P_k - P_{k-1}$ will always be greater than 0, which means that $\lim_{k\to\infty} P_k = \infty$.

$\nabla\nabla\nabla$

4.3 CONTINUOUS-TIME SYSTEMS

In this section, we will look at how the mean and covariance of the state of a continuous-time linear system propagate. Consider the continuous-time system

$$\dot{x} = Ax + Bu + w \tag{4.34}$$

where $u(t)$ is a known control input and $w(t)$ is zero-mean white noise with a covariance of

$$E[w(t)w^T(\tau)] = Q_c\delta(t-\tau) \tag{4.35}$$

By taking the mean of Equation (4.34), we can obtain the following equation for the derivative of the mean of the state:

$$\dot{\bar{x}} = A\bar{x} + Bu \tag{4.36}$$

This equation shows how the mean of the state propagates with time. The linear equation that describes the propagation of the mean looks very much like the original state equation, Equation (4.34). We can also obtain Equation (4.36) by using the equation that describes the mean of a sampled-data system and taking the limit as $\Delta t = t_k - t_{k-1}$ goes to zero. Taking the mean of Equation (4.18) gives

$$\bar{x}_k = e^{A\Delta t}\bar{x}_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}B(\tau)u(\tau)\,d\tau \tag{4.37}$$

The state transition matrix can be written as

$$\begin{aligned} F &= e^{A\Delta t} \\ &= I + A\Delta t + \frac{(A\Delta t)^2}{2!} + \cdots \end{aligned} \tag{4.38}$$

For small values of Δt, this can be approximated as

$$F \approx I + A\Delta t \tag{4.39}$$

With this substitution Equation (4.37) becomes

$$\bar{x}_k = (I + A\Delta t)\bar{x}_{k-1} + \int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}B(\tau)u(\tau)\,d\tau \tag{4.40}$$

Subtracting $\bar{x}_{k-1}$ from both sides and dividing by Δt gives

$$\frac{\bar{x}_k - \bar{x}_{k-1}}{\Delta t} = A\bar{x}_{k-1} + \frac{1}{\Delta t}\int_{t_{k-1}}^{t_k} e^{A(t_k-\tau)}B(\tau)u(\tau)\,d\tau \tag{4.41}$$

Taking some limits as Δt goes to zero gives the following:

$$\begin{aligned} \lim_{\Delta t\to 0}\frac{\bar{x}_k - \bar{x}_{k-1}}{\Delta t} &= \dot{\bar{x}} \\ \lim_{\Delta t\to 0} e^{A(t_k-\tau)} &= I \text{ for } \tau \in [t_{k-1}, t_k] \end{aligned} \tag{4.42}$$

Making these substitutions in (4.41) gives

$$\dot{\bar{x}} = A\bar{x} + Bu \tag{4.43}$$

which is the same equation as the one we derived earlier in Equation (4.36) by a more direct method. Although the limiting argument that we used here was not necessary because we already had the mean equation in Equation (4.36), this method shows us how we can use limiting arguments (in general) to obtain continuous-time formulas.

Next we will use a limiting argument to derive the covariance of the state of a continuous-time system. Recall the equation for the covariance of a sampled data system from Equation (4.26):

$$P_k = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1} \tag{4.44}$$

For small Δt we again approximate F_{k-1} as shown in Equation (4.39) and substitute into the above equation to obtain

$$\begin{aligned} P_k &\approx (I + A\Delta t)P_{k-1}(I + A\Delta t)^T + Q_{k-1} \\ &= P_{k-1} + AP_{k-1}\Delta t + P_{k-1}A^T\Delta t + AP_{k-1}A^T(\Delta t)^2 + Q_{k-1} \end{aligned} \tag{4.45}$$

Subtracting P_{k-1} from both sides and dividing by Δt gives

$$\frac{P_k - P_{k-1}}{\Delta t} = AP_{k-1} + P_{k-1}A^T + AP_{k-1}A^T\Delta t + \frac{Q_{k-1}}{\Delta t} \tag{4.46}$$

Recall from Equation (4.28) that for small Δt

$$Q_{k-1} \approx Q_c(t_k)\Delta t \tag{4.47}$$

This can be written as

$$\frac{Q_{k-1}}{\Delta t} \approx Q_c(t_k) \tag{4.48}$$

Therefore, taking the limit of Equation (4.46) as Δt goes to zero gives

$$\dot{P} = AP + PA^T + Q_c \tag{4.49}$$

This continuous-time Lyapunov equation, also sometimes called a Sylvester equation, gives us the equation for how the covariance of the state of a continuous-time system propagates with time.

It is interesting to consider the conditions under which the continuous-time Lyapunov equation has a steady-state solution. That is, suppose that $A(t) = A$ is a constant, and $Q_c(t) = Q_c$ is a constant. Then we have the following theorem, whose proof can be found in [Kai00, Appendix D].

Theorem 22 *Consider the equation $AP + PA^T + Q_c = 0$ where A and Q_c are real matrices. Denote by $\lambda_i(A)$ the eigenvalues of the A matrix.*

1. *A unique solution P exists if and only if $\lambda_i(A) + \lambda_j(A) \neq 0$ for all i, j. This unique solution is symmetric.*

2. *Note that the above condition includes the case of stable A, because if A is stable then all of its eigenvalues have real parts less than 0, so $\lambda_i(A)+\lambda_j(A) \neq 0$ for all i, j. Therefore, we see that if A is stable then the continuous-time Lyapunov equation has a solution P that is unique and symmetric. In this case, the solution can be written as*

$$P = \int_0^\infty e^{A^T\tau} Q_c e^{A\tau}\, d\tau \tag{4.50}$$

3. *If A is stable and Q_c is positive (semi)definite, then the unique solution P is symmetric and positive (semi)definite.*

4. *If A is stable, Q_c is positive semidefinite, and $\left[A, (Q_c^{1/2})^T\right]$ is controllable, then P is unique, symmetric, and positive definite. Note that $Q_c^{1/2}$, the square root of Q_c, is defined here as any matrix such that $Q_c^{1/2}(Q_c^{1/2})^T = Q_c$.*

■ EXAMPLE 4.3

Suppose we have the first-order, continuous-time dynamic system given by Equation (4.29):

$$\begin{aligned} \dot{x} &= fx + w \\ E[w(t)w(t+\tau)] &= q_c\delta(\tau) \end{aligned} \tag{4.51}$$

where $w(t)$ is zero-mean noise. The equation for the continuous-time propagation of the mean of the state is obtained from Equation (4.36):

$$\dot{\bar{x}} = f\bar{x} \tag{4.52}$$

Solving this equation for $\bar{x}(t)$ gives

$$\bar{x}(t) = \exp(ft)\bar{x}(0) \tag{4.53}$$

We see that the mean will increase without bound if $f > 0$ (i.e., if the system is unstable), but the mean will asymptotically tend to zero if $f < 0$ (i.e., if the system is stable). The equation for the continuous-time propagation of the covariance of the state is obtained from Equation (4.49):

$$\dot{P} = 2fP + q_c \tag{4.54}$$

Solving this equation for $P(t)$ gives

$$P(t) = \left(P(0) + \frac{q_c}{2f}\right)\exp(2ft) - \frac{q_c}{2f} \tag{4.55}$$

We see that the covariance will increase without bound if $f > 0$ (i.e., if the system is unstable), but the covariance will asymptotically tend to $-q_c/2f$ if $f < 0$ (i.e., if the system is stable). Compare these results with Example 4.2.

The steady-state value of P can also be computed using Equation (4.50). If we substitute f for A and q_c for Q_c in Equation (4.50), we obtain

$$\begin{aligned} P &= \int_0^\infty e^{2f\tau} q_c \, d\tau \\ &= \frac{q_c}{2f} e^{2f\tau}\Big|_0^\infty \end{aligned} \tag{4.56}$$

The integral converges for $f < 0$ (i.e., if the system is stable), in which case $P = -q_c/2f$.

∇∇∇

4.4 SUMMARY

In this chapter, we have derived equations for the propagation of the mean and covariance of the state of linear systems. For discrete-time systems, the mean and covariance are described by difference equations. Sampled-data systems are systems with continuous-time dynamics but control inputs that are constant between sample times. If the dynamics of a sampled-data system does not change between sample times, then the mean and covariance are described by difference equations, although the factors of the difference equations are more complicated than they are for discrete-time systems. For continuous-time systems, the mean and covariance are described by differential equations. These results will form part of the foundation for our Kalman filter derivation in Chapter 5.

The covariance equations that we studied in this chapter are named after Aleksandr Lyapunov, James Sylvester, and Philip Stein. Lyapunov was a Russian mathematician who lived from 1857 to 1918. He made important contributions in the areas of differential equations, system stability, and probability. Sylvester was an English mathematician and lawyer who lived from 1814 to 1897. He worked for a time in the United States as a professor at the University of Virginia and Johns Hopkins University. While at Johns Hopkins, he founded the *American Journal of Mathematics*, which was the first mathematical journal in the United States.

PROBLEMS

Written exercises

4.1 Prove that

$$\frac{d}{dt}\left(E[x]\right) = E\left[\frac{dx}{dt}\right]$$

4.2 Suppose that a dynamic scalar system is given as $x_{k+1} = fx_k + w_k$, where w_k is zero-mean white noise with variance q. Show that if the variance of x_k is σ^2 for all k, then it must be true that $f^2 = (\sigma^2 - q)/\sigma^2$.

4.3 Consider the system

$$\begin{aligned} x_k &= \begin{bmatrix} 1 & 1 \\ 0 & 1/2 \end{bmatrix} x_{k-1} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_{k-1} \\ w_k &\sim (0,1) \end{aligned}$$

where w_k is white noise.

a) Find all possible steady-state values of the mean of x_k.

b) Find all possible steady-state values of the covariance of x_k.

4.4 Consider the system of Example 1.2.

a) Discretize the system to find the single step state transition matrix F_k, the discrete-time input matrix G_k, and the multiple-step state transition matrix $F_{k,i}$.

b) Suppose the covariance of the initial state is $P_0 = \text{diag}(1, 0)$, and zero-mean discrete-time white noise with a covariance of $Q = \text{diag}(1, 0)$ is input to the discrete-time system. Find a closed-form solution for P_k.

4.5 Two chemical mixtures are poured into a tank. One has concentration c_1 and is poured at rate F_1, and the other has concentration c_2 and is poured at rate F_2. The tank has volume V, and its outflow is at concentration c and rate F. This is typical of many process control systems [Kwa72]. The linearized equation for this system can be written as

$$\dot{x} = \begin{bmatrix} -\frac{F_0}{2V_0} & 0 \\ 0 & -\frac{F_0}{V_0} \end{bmatrix} x + \begin{bmatrix} 1 & 1 \\ \frac{c_1 - c_0}{V_0} & \frac{c_2 - c_0}{V_0} \end{bmatrix} w$$

where F_0, V_0, and c_0 are the linearization points of F, V, and c. The state x consists of deviations from the steady-state values of V and c, and the noise input w consists of the deviations from the steady-state values of F_1 and F_2. Suppose that $F_0 = 2V_0$, $c_1 - c_0 = V_0$, and $c_2 - c_0 = 2V_0$. Suppose the noise input w has an identity covariance matrix.

a) Use Equation (4.27) to calculate Q_{k-1}.
b) Use Equation (4.28) to approximate Q_{k-1}.
c) Evaluate your answer to part (a) for small $(t_k - t_{k-1})$ to verify that it matches your answer to part (b).

4.6 Suppose that a certain sampled data system has the following state-transition matrix and approximate Q_{k-1} matrix [as calculated by Equation (4.28)]:

$$\begin{aligned} F_{k-1} &= \begin{bmatrix} e^{-T} & 0 \\ 0 & e^{-2T} \end{bmatrix} \\ Q_{k-1} &= \begin{bmatrix} 2T & 3T \\ 3T & 5T \end{bmatrix} \end{aligned}$$

where $T = t_k - t_{k-1}$ is the discretization step size. Use Equation (4.26) to compute the steady-state covariance of the state as a function of T.

4.7 Consider the tank system described in Problem 4.5. Find closed-form solutions for the elements of the state covariance as functions of time.

4.8 Consider the system

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} x_k + w_k \\ w_k &\sim (0, Q) \\ Q &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Use Equation (4.5) to find the steady-state covariance of the state vector.

4.9 The third condition of Theorem 21 gives a sufficient condition for the discrete-time Lyapunov equation to have a unique, symmetric, positive semidefinite solution. Since the condition is sufficient but not necessary, there may be cases that do not meet the criteria of the third condition that still have a unique, symmetric, positive semidefinite solution. Give an example of one such case with a nonzero solution.

4.10 Prove the sifting property of the continuous-time impulse function $\delta(t)$, which can be stated as

$$\int_{-\infty}^{\infty} f(t)\delta(t-\alpha)\,dt = f(\alpha)$$

Computer exercises

4.11 Write code for the propagation of the mean and variance of the state of Example 4.2. Use $m_0 = 1$, $P_0 = 2$, $f = -0.5$ and $q_c = 1$. Plot the mean and variance of x for 5 seconds. Repeat for $P_0 = 0$. Based on the plots, what does the steady-state value of the variance appear to be? What is the analytically determined steady-state value of the variance?

4.12 Consider the RLC circuit of Example 1.8 with $R = L = C = 1$. Suppose the applied voltage is continuous-time zero-mean white noise with a variance of 1. The initial capacitor voltage is a random variable with a mean of 1 and a variance of 1. The initial inductor current is a random variable (independent of the initial capacitor voltage) with a mean of 2 and a variance of 2. Write a program to propagate the mean and covariance of the state for five seconds. Plot the two elements of the mean of the state, and the three unique elements of the covariance. Based on the plots, what does the steady-state value of the covariance appear to be? What is the analytically determined steady-state value of the covariance? (Hint: The MATLAB function LYAP can be used to solve for the continuous-time algebraic Lyapunov equation.)

4.13 Consider the RLC circuit of Problem 1.18 with $R = 3$, $L = 1$, and $C = 0.5$. Suppose the applied voltage is continuous-time zero-mean white noise with a variance of 1. We can find the steady-state covariance of the state a couple of different ways.

- Use Equation (4.49).
- Discretize the system and use Equation (4.4) along with the MATLAB function DLYAP. In this case, the discrete-time white noise covariance Q is related to the continuous-time white noise covariance Q_c by the equation $Q = TQ_c$, where T is the discretization step size (see Section 8.1.1).

a) Analytically compute the continuous-time, steady-state covariance of the state.

b) Analytically compute the discretized steady-state covariance of the state in the limit as $T \to \infty$.

c) One way of measuring the distance between two matrices is by using the MATLAB function NORM to take the Frobenius norm of the difference between the matrices. Generate a plot showing the Frobenius norm of the difference between the continuous-time, steady-state covariance of the state, and the discretized steady-state covariance of the state for T between 0.01 and 1.

PART II

THE KALMAN FILTER

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

CHAPTER 5

The discrete-time Kalman filter

The Kalman filter in its various forms is clearly established as a fundamental tool for analyzing and solving a broad class of estimation problems.

—Leonard McGee and Stanley Schmidt [McG85]

This chapter forms the heart of this book. The earlier chapters were written only to provide the foundation for this chapter, and the later chapters are written only to expand and generalize the results given in this chapter.

As we will see in this chapter, the Kalman filter operates by propagating the mean and covariance of the state through time. Our approach to deriving the Kalman filter will involve the following steps.

1. We start with a mathematical description of a dynamic system whose states we want to estimate.

2. We implement equations that describe how the mean of the state and the covariance of the state propagate with time. These equations, derived in Chapter 4, themselves form a dynamic system.

3. We take the dynamic system that describes the propagation of the state mean and covariance, and implement the equations on a computer. These equations form the basis for the derivation of the Kalman filter because:

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

(a) The mean of the state is the Kalman filter estimate of the state.

(b) The covariance of the state is the covariance of the Kalman filter state estimate.

4. Every time that we get a measurement, we update the mean and covariance of the state. This is similar to what we did in Chapter 3 where we used measurements to recursively update our estimate of a constant.

In Section 5.1, we derive the equations of the discrete-time Kalman filter. This includes several different-looking, but mathematically equivalent forms. Various books and papers that deal with Kalman filters present the filter equations in ways that appear very different from one another. It is not always obvious, but these different formulations are actually mathematically equivalent, and we will see this in Section 5.1. (Sections 9.1, 10.5.1, and 11.1 also derive alternate but equivalent formulations of the Kalman filter equations.) In Section 5.2, we will examine some of the theoretical properties of the Kalman filter. One remarkable aspect of the Kalman filter is that it is optimal in several different senses, as we will see in Section 5.2. In Section 5.3, we will see how the Kalman filter equations can be written with a single time update equation. Section 5.4 presents a way to obtain a closed-form equation for the time-varying Kalman filter for a scalar time-invariant system, and a way to quickly compute the steady-state Kalman filter. Section 5.5 looks at some situations in which the Kalman filter is unstable or gives state estimates that are not close to the true state. We will also look at some ways that instability and divergence can be corrected in the Kalman filter.

5.1 DERIVATION OF THE DISCRETE-TIME KALMAN FILTER

Suppose we have a linear discrete-time system given as follows:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \end{aligned} \tag{5.1}$$

The noise processes $\{w_k\}$ and $\{v_k\}$ are white, zero-mean, uncorrelated, and have known covariance matrices Q_k and R_k, respectively:

$$\begin{aligned} w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \\ E[w_k w_j^T] &= Q_k \delta_{k-j} \\ E[v_k v_j^T] &= R_k \delta_{k-j} \\ E[v_k w_j^T] &= 0 \end{aligned} \tag{5.2}$$

where δ_{k-j} is the Kronecker delta function; that is, $\delta_{k-j} = 1$ if $k = j$, and $\delta_{k-j} = 0$ if $k \neq j$. Our goal is to estimate the state x_k based on our knowledge of the system dynamics and the availability of the noisy measurements $\{y_k\}$. The amount of information that is available to us for our state estimate varies depending on the particular problem that we are trying to solve. If we have all of the measurements up to and including time k available for use in our estimate of x_k, then we can form an *a posteriori* estimate, which we denote as $\hat{x}_k^+$. The "+" superscript denotes that

the estimate is *a posteriori*. One way to form the *a posteriori* state estimate is to compute the expected value of x_k conditioned on all of the measurements up to and including time k:

$$\hat{x}_k^+ = E[x_k|y_1, y_2, \cdots, y_k] = \text{ a posteriori estimate} \tag{5.3}$$

If we have all of the measurements before (but not including) time k available for use in our estimate of x_k, then we can form an *a priori* estimate, which we denote as $\hat{x}_k^-$. The "–" superscript denotes that the estimate is *a priori*. One way to form the *a priori* state estimate is to compute the expected value of x_k conditioned on all of the measurements before (but not including) time k:

$$\hat{x}_k^- = E[x_k|y_1, y_2, \cdots, y_{k-1}] = \text{ a priori estimate} \tag{5.4}$$

It is important to note that $\hat{x}_k^-$ and $\hat{x}_k^+$ are both estimates of the same quantity; they are both estimates of x_k. However, $\hat{x}_k^-$ is our estimate of x_k *before* the measurement y_k is taken into account, and $\hat{x}_k^+$ is our estimate of x_k *after* the measurement y_k is taken into account. We naturally expect $\hat{x}_k^+$ to be a better estimate than $\hat{x}_k^-$, because we use more information to compute $\hat{x}_k^+$:

$$\begin{aligned} \hat{x}_k^- &= \text{ estimate of } x_k \text{ before we process the measurement at time } k \\ \hat{x}_k^+ &= \text{ estimate of } x_k \text{ after we process the measurement at time } k \end{aligned} \tag{5.5}$$

If we have measurements after time k available for use in our estimate of x_k, then we can form a *smoothed* estimate. One way to form the smoothed state estimate is to compute the expected value of x_k conditioned on all of the measurements that are available:

$$\hat{x}_{k|k+N} = E[x_k|y_1, y_2, \cdots, y_k, \cdots, y_{k+N}] = \text{ smoothed estimate} \tag{5.6}$$

where N is some positive integer whose value depends on the specific problem that is being solved. If we want to find the best prediction of x_k more than one time step ahead of the available measurements, then we can form a *predicted* estimate. One way to form the predicted state estimate is to compute the expected value of x_k conditioned on all of the measurements that are available:

$$\hat{x}_{k|k-M} = E[x_k|y_1, y_2, \cdots, y_{k-M}] = \text{ predicted estimate} \tag{5.7}$$

where M is some positive integer whose value depends on the specific problem that is being solved. The relationship between the *a posteriori*, *a priori*, smoothed, and predicted state estimates is depicted in Figure 5.1.

In the notation that follows, we use $\hat{x}_0^+$ to denote our initial estimate of x_0 before any measurements are available. The first measurement is taken at time $k = 1$. Since we do not have any measurements available to estimate x_0, it is reasonable to form $\hat{x}_0^+$ as the expected value of the initial state x_0:

$$\hat{x}_0^+ = E(x_0) \tag{5.8}$$

We use the term P_k to denote the covariance of the estimation error. P_k^- denotes the covariance of the estimation error of $\hat{x}_k^-$, and P_k^+ denotes the covariance of the estimation error of $\hat{x}_k^+$:

$$\begin{aligned} P_k^- &= E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T] \\ P_k^+ &= E[(x_k - \hat{x}_k^+)(x_k - \hat{x}_k^+)^T] \end{aligned} \tag{5.9}$$

Figure 5.1 Time line showing the relationship between the *a posteriori*, *a priori*, smoothed, and predicted state estimates. In this figure, we suppose that we have received measurements at times up to and including $k = 5$. An estimate of the state at $k < 5$ is called a smoothed estimate. An estimate of the state at $k = 5$ is called the *a posteriori* estimate. An estimate of the state at $k = 6$ is called the *a priori* estimate. An estimate of the state at $k > 6$ is called the prediction.

These relationships are depicted in Figure 5.2. The figure shows that after we process the measurement at time $(k-1)$, we have an estimate of x_{k-1} (denoted $\hat{x}_{k-1}^+$) and the covariance of that estimate (denoted P_{k-1}^+). When time k arrives, before we process the measurement at time k we compute an estimate of x_k (denoted $\hat{x}_k^-$) and the covariance of that estimate (denoted P_k^-). Then we process the measurement at time k to refine our estimate of x_k. The resulting estimate of x_k is denoted $\hat{x}_k^+$, and its covariance is denoted P_k^+.

Figure 5.2 Timeline showing *a priori* and *a posteriori* state estimates and estimation-error covariances.

We begin the estimation process with $\hat{x}_0^+$, our best estimate of the initial state x_0. Given $\hat{x}_0^+$, how should we compute $\hat{x}_1^-$? We want to set $\hat{x}_1^- = E(x_1)$. But note that $\hat{x}_0^+ = E(x_0)$, and recall from Equation (4.2) how the mean of x propagates with time: $\bar{x}_k = F_{k-1}\bar{x}_{k-1} + G_{k-1}u_{k-1}$. We therefore obtain

$$\hat{x}_1^- = F_0\hat{x}_0^+ + G_0u_0 \tag{5.10}$$

This is a specific equation that shows how to obtain $\hat{x}_1^-$ from $\hat{x}_0^+$. However, the reasoning can be extended to obtain the following more general equation:

$$\hat{x}_k^- = F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1} \tag{5.11}$$

This is called the time update equation for $\hat{x}$. From time $(k-1)^+$ to time k^-, the state estimate propagates the same way that the mean of the state propagates. This makes sense intuitively. We do not have any additional measurements available to

help us update our state estimate between time $(k-1)^+$ and time k^-, so we should just update the state estimate based on our knowledge of the system dynamics.

Next we need to compute the time update equation for P, the covariance of the state estimation error. We begin with P_0^+, which is the covariance of our initial estimate of x_0. If we know the initial state perfectly, then $P_0^+ = 0$. If we have absolutely no idea of the value of x_0, then $P_0^+ = \infty I$. In general, P_0^+ represents the uncertainty in our initial estimate of x_0:

$$\begin{aligned} P_0^+ &= E[(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T] \\ &= E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T] \end{aligned} \tag{5.12}$$

Given P_0^+, how can we compute P_1^-? Recall from Equation (4.4) how the covariance of the state of a linear discrete-time system propagates with time: $P_k = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$. We therefore obtain

$$P_1^- = F_0 P_0^+ F_0^T + Q_0 \tag{5.13}$$

This is a specific equation that shows how to obtain P_1^- from P_0^+. However, the reasoning can be extended to obtain the following more general equation:

$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \tag{5.14}$$

This is called the time-update equation for P.

We have derived the time-update equations for $\hat{x}$ and P. Now we need to derive the measurement-update equations for $\hat{x}$ and P. Given $\hat{x}_k^-$, how should we compute $\hat{x}_k^+$? The quantity $\hat{x}_k^-$ is an estimate of x_k, and the quantity $\hat{x}_k^+$ is also an estimate of x_k. The only difference between $\hat{x}_k^-$ and $\hat{x}_k^+$ is that $\hat{x}_k^+$ takes the measurement y_k into account. Recall from the recursive least squares development in Section 3.3 that the availability of the measurement y_k changes the estimate of a constant x as follows:

$$\begin{aligned} K_k &= P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1} \\ &= P_kH_k^TR_k^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \\ P_k &= (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T \\ &= (P_{k-1}^{-1} + H_k^TR_k^{-1}H_k)^{-1} \\ &= (I - K_kH_k)P_{k-1} \end{aligned} \tag{5.15}$$

where $\hat{x}_{k-1}$ and P_{k-1} are the estimate and its covariance *before* the measurement y_k is processed, and $\hat{x}_k$ and P_k are the estimate and its covariance *after* the measurement y_k is processed. In this chapter, $\hat{x}_k^-$ and P_k^- are the estimate and its covariance before the measurement y_k is processed, and $\hat{x}_k^+$ and P_k^+ are the estimate and its covariance after the measurement y_k is processed. These relationships are shown in Table 5.1.[1]

We can now generalize from the formulas for the estimation of a constant in Section 3.3, to the measurement update equations required in this section. In

[1]We need to use minus and plus superscripts on $\hat{x}_k$ and P_k in order to distinguish between quantities before y_k is taken into account, and quantities after y_k is taken into account. In Chapter 3, we did not need superscripts because x was a constant.

Table 5.1 Relationships between estimates and covariances in Sections 3.3 and 5.1

Section 3.3 Least squares estimation		Section 5.1 Kalman filtering
$\hat{x}_{k-1}$ = estimate before y_k is processed	$\Longrightarrow$	$\hat{x}_k^-$ = *a priori* estimate
P_{k-1} = covariance before y_k is processed	$\Longrightarrow$	P_k^- = *a priori* covariance
$\hat{x}_k$ = estimate after y_k is processed	$\Longrightarrow$	$\hat{x}_k^+$ = *a posteriori* estimate
P_k = covariance after y_k is processed	$\Longrightarrow$	P_k^+ = *a posteriori* covariance

Equation (5.15), we replace $\hat{x}_{k-1}$ with $\hat{x}_k^-$, we replace P_{k-1} with P_k^-, we replace $\hat{x}_k$ with $\hat{x}_k^+$, and we replace P_k with P_k^+. This results in

$$\begin{aligned}
K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\
&= P_k^+ H_k^T R_k^{-1} \\
\hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k \hat{x}_k^-) \\
P_k^+ &= (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \\
&= \left[(P_k^-)^{-1} + H_k^T R_k^{-1} H_k\right]^{-1} \\
&= (I - K_k H_k) P_k^-
\end{aligned} \tag{5.16}$$

These are the measurement-update equations for $\hat{x}_k$ and P_k. The matrix K_k in the above equations is called the Kalman filter gain.

The discrete-time Kalman filter

Here we summarize the discrete-time Kalman filter by combining the above equations into a single algorithm.

1. The dynamic system is given by the following equations:

$$\begin{aligned}
x_k &= F_{k-1} x_{k-1} + G_{k-1} u_{k-1} + w_{k-1} \\
y_k &= H_k x_k + v_k \\
E(w_k w_j^T) &= Q_k \delta_{k-j} \\
E(v_k v_j^T) &= R_k \delta_{k-j} \\
E(w_k v_j^T) &= 0
\end{aligned} \tag{5.17}$$

2. The Kalman filter is initialized as follows:

$$\begin{aligned}
\hat{x}_0^+ &= E(x_0) \\
P_0^+ &= E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]
\end{aligned} \tag{5.18}$$

3. The Kalman filter is given by the following equations, which are computed for each time step $k = 1, 2, \cdots$:

$$\begin{aligned}
P_k^- &= F_{k-1} P_{k-1}^+ F_{k-1}^T + Q_{k-1} \\
K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}
\end{aligned}$$

$$
\begin{aligned}
&= P_k^+ H_k^T R_k^{-1} \\
\hat{x}_k^- &= F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1} = \textit{ a priori } \text{state estimate} \\
\hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) = \textit{ a posteriori } \text{state estimate} \\
P_k^+ &= (I - K_kH_k)P_k^-(I - K_kH_k)^T + K_kR_kK_k^T \\
&= \left[(P_k^-)^{-1} + H_k^T R_k^{-1} H_k\right]^{-1} \\
&= (I - K_kH_k)P_k^- \qquad (5.19)
\end{aligned}
$$

The first expression for P_k^+ above is called the Joseph stabilized version of the covariance measurement update equation. It was formulated by Peter Joseph in the 1960s and can be shown to be more stable and robust than the third expression for P_k^+ [Buc68, Cra04] (see Problem 5.2). The first expression for P_k^+ guarantees that P_k^+ will always be symmetric positive definite, as long as P_k^- is symmetric positive definite. The third expression for P_k^+ is computationally simpler than the first expression, but its form does not guarantee symmetry or positive definiteness for P_k^+. The second form for P_k^+ is rarely implemented as written above but will be useful in our derivation of the information filter in Section 6.2.

If the second expression for K_k is used, then the second expression for P_k^+ must be used. This is because the second expression for K_k depends on P_k^+, so we need to use an expression for P_k^+ that does not depend on K_k.

Note that if x_k is a constant, then $F_k = I$, $Q_k = 0$, and $u_k = 0$. In this case, the Kalman filter of Equation (5.19) reduces to the recursive least squares algorithm for the estimation of a constant vector as given in Equation (3.47).

Finally we mention one more important practical aspect of the Kalman filter. We see from Equation (5.19) that the calculation of P_k^-, K_k, and P_k^+ does not depend on the measurements y_k, but depends only on the system parameters F_k, H_k, Q_k, and R_k. That means that the Kalman gain K_k can be calculated offline before the system operates and saved in memory. Then when it comes time to operate the system in real time, only the $\hat{x}_k$ equations need to be implemented in real time. The computational effort of calculating K_k can be saved during real-time operation by precomputing it. If the Kalman filter is implemented in an embedded system with strict computational requirements, this can make the difference between whether or not the system can be implemented in real time. Furthermore, the performance of the filter can be investigated and evaluated before the filter is actually run. This is because P_k indicates the estimation accuracy of the filter, and it can be computed offline since it does not depend on the measurements. In contrast, as we will see in Chapter 13, the filter gain and covariance for nonlinear systems cannot (in general) be computed offline because they depend on the measurements.

5.2 KALMAN FILTER PROPERTIES

In this section, we summarize some of the interesting and important properties of the Kalman filter. Suppose we are given the linear system of Equation (5.17) and we want to find a causal filter that results in a state estimate $\hat{x}_k$. The error between the true state and the estimated state is denoted as $\tilde{x}_k$:

$$\tilde{x}_k = x_k - \hat{x}_k \qquad (5.20)$$

Since the state is partly determined by the stochastic process $\{w_k\}$, x_k is a random variable. Since the state estimate is determined by the measurement sequence $\{y_k\}$, which in turn is partly determined by the stochastic process $\{v_k\}$, $\hat{x}_k$ is a random variable. Therefore, $\tilde{x}_k$ is also a random variable.

Suppose we want to find the estimator that minimizes (at each time step) a weighted two-norm of the expected value of the estimation error $\tilde{x}_k$:

$$\min E\left[\tilde{x}_k^T S_k \tilde{x}_k\right] \tag{5.21}$$

where S_k is a positive definite user-defined weighting matrix. If S_k is diagonal with elements $S_k(1), \cdots, S_k(n)$, then the weighted sum is equal to $S_k(1)E[\tilde{x}_k^2(1)] + \cdots + S_k(n)E[\tilde{x}_k^2(n)]$.

- If $\{w_k\}$ and $\{v_k\}$ are Gaussian, zero-mean, uncorrelated, and white, then the Kalman filter is the solution to the above problem.

- If $\{w_k\}$ and $\{v_k\}$ are zero-mean, uncorrelated, and white, then the Kalman filter is the best linear solution to the above problem. That is, the Kalman filter is the best filter that is a linear combination of the measurements. There may be a nonlinear filter that gives a better solution, but the Kalman filter is the best linear filter. It is often asserted in books and papers that the Kalman filter is not optimal unless the noise is Gaussian. However, as our derivation in this chapter has shown, that is simply untrue. Such statements arise from erroneous interpretations of Kalman filter derivations. Even if the noise is not Gaussian, the Kalman filter is still the optimal *linear* filter.

- If $\{w_k\}$ and $\{v_k\}$ are correlated or colored, then the Kalman filter can be modified to solve the above problem. This will be shown in Chapter 7.

- For nonlinear systems, various formulations of nonlinear Kalman filters approximate the solution to the above problem. This will be discussed further in Chapters 13–15.

Recall the measurement update equation from Equation (5.19):

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \tag{5.22}$$

The quantity $(y_k - H_k\hat{x}_k^-)$ is called the innovations. This is the part of the measurement that contains new information about the state. In Section 10.1, we will prove that the innovations is zero-mean and white with covariance $(H_k P_k^- H_k^T + R_k)$. In fact, the Kalman filter can actually be derived as a filter that whitens the measurement and hence extracts the maximum possible amount of information from the measurement. This was first proposed in [Kai68]. When a Kalman filter is used for state estimation, the innovations can be measured and its mean and covariance can be approximated using statistical methods. If the mean and covariance of the innovations are not as expected, that means something is wrong with the filter. Perhaps the assumed system model is incorrect, or the assumed noise statistics are incorrect. This can be used in real time to verify Kalman filter performance and parameters, and even to adjust Kalman filter parameters in order to improve performance. An application of this idea will be explored in Section 10.2.

5.3 ONE-STEP KALMAN FILTER EQUATIONS

In this section, we will see how the *a priori* and *a posteriori* Kalman filter equations can be combined into a single equation. This may simplify computer implementation of the equations. We start with the *a priori* state estimate expression from Equation (5.19), with the time index increased by one:

$$\hat{x}_{k+1}^- = F_k \hat{x}_k^+ + G_k u_k \tag{5.23}$$

Now take the *a posteriori* expression for $\hat{x}_k^+$ from Equation (5.19), and substitute it into the above equation to obtain

$$\begin{aligned} \hat{x}_{k+1}^- &= F_k \left[\hat{x}_k^- + K_k(y_k - H_k \hat{x}_k^-)\right] + G_k u_k \\ &= F_k(I - K_k H_k)\hat{x}_k^- + F_k K_k y_k + G_k u_k \end{aligned} \tag{5.24}$$

This shows that the *a priori* state estimate can be computed directly from its value at the previous time step, without computing the *a posteriori* state estimate in between. A similar procedure can be followed in order to obtain a one-step expression for the *a priori* covariance. We start with the *a priori* covariance expression from Equation (5.19), with the time index increased by one:

$$P_{k+1}^- = F_k P_k^+ F_k^T + Q_k \tag{5.25}$$

Now take the expression for P_k^+ from Equation (5.19), and substitute it into the above equation to obtain

$$\begin{aligned} P_{k+1}^- &= F_k(P_k^- - K_k H_k P_k^-)F_k^T + Q_k \\ &= F_k P_k^- F_k^T - F_k K_k H_k P_k^- F_k^T + Q_k \\ &= F_k P_k^- F_k^T - F_k P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- F_k^T + Q_k \end{aligned} \tag{5.26}$$

This equation, called a discrete Riccati equation, shows how P_{k+1}^- can be computed on the basis of P_k^- without an intermediate calculation of P_k^+.

Similar manipulations can be performed to obtain one-step expressions for the *a posteriori* state estimate and covariance. This results in

$$\begin{aligned} \hat{x}_k^+ &= (I - K_k H_k)(F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1}) + K_k y_k \\ P_k^+ &= (I - K_k H_k)(F_{k-1}P_{k-1}^+ F_{k-1}^T + Q_{k-1}) \end{aligned} \tag{5.27}$$

One could imagine many different ways of combining the two expressions for K_k and the three expressions for P_k^+ in Equation (5.19). This would result in a number of different expressions for one-step updates for the *a priori* and *a posteriori* covariance.

■ EXAMPLE 5.1

Suppose we have a noise-free Newtonian system[2] with position r, velocity v, and constant acceleration a. The system can be described as

$$\begin{aligned} \begin{bmatrix} \dot{r} \\ \dot{v} \\ \dot{a} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ v \\ a \end{bmatrix} \\ \dot{x} &= Ax \end{aligned} \tag{5.28}$$

The discretized version of this system (with a sample time of T) can be written as

$$x_{k+1} = Fx_k \tag{5.29}$$

where F is given as

$$\begin{aligned} F &= \exp(AT) \\ &= I + AT + \frac{(AT)^2}{2!} + \cdots \\ &= \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{5.30}$$

The Kalman filter for this system is

$$\begin{aligned} \hat{x}_k^- &= F\hat{x}_{k-1}^+ \\ P_k^- &= FP_{k-1}^+F^T + \underbrace{Q_{k-1}}_{0} \\ &= FP_{k-1}^+F^T \end{aligned} \tag{5.31}$$

We see that the covariance of the estimation error increases between time $(k-1)^+$ [that is, time $(k-1)$ after the measurement at that time is processed], and time k^- (i.e., time k before the measurement at that time is processed). Since we do not obtain any measurements between time $(k-1)^+$ and time k^-, it makes sense that our estimation uncertainty increases. Now suppose that we measure position with a variance of σ^2:

$$\begin{aligned} y_k &= H_k x_k + v_k \\ &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x_k + v_k \\ v_k &\sim (0, R_k) \\ R_k &= \sigma^2 \end{aligned} \tag{5.32}$$

The Kalman gain can be obtained from Equation (5.19) as

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{5.33}$$

If we write out the 3×3 matrix P_k^- in terms of its individual elements, and substitute for H_k and R_k in the above equation, we obtain

[2]The system described in this example is called Newtonian because it has its roots in the mathematical work of Isaac Newton. That is, velocity is the derivative of position, and acceleration is the derivative of velocity.

$$K_k = \begin{bmatrix} P^-_{k,11} \\ P^-_{k,12} \\ P^-_{k,13} \end{bmatrix} \frac{1}{P^-_{k,11} + \sigma^2} \tag{5.34}$$

The *a posteriori* covariance can be obtained from Equation (5.19) as

$$P^+_k = P^-_k - K_k H_k P^-_k \tag{5.35}$$

If we write out the 3×3 matrix P^-_k in terms of its individual elements, and substitute for H_k and K_k in the above equation, we obtain

$$\begin{aligned} P^+_k &= P^-_k - \frac{1}{P^-_{k,11} + \sigma^2} \begin{bmatrix} P^-_{k,11} & 0 & 0 \\ P^-_{k,12} & 0 & 0 \\ P^-_{k,13} & 0 & 0 \end{bmatrix} P^-_k \\ &= P^-_k - \frac{1}{P^-_{k,11} + \sigma^2} \begin{bmatrix} (P^-_{k,11})^2 & P^-_{k,11}P^-_{k,21} & P^-_{k,11}P^-_{k,31} \\ P^-_{k,12}P^-_{k,11} & (P^-_{k,12})^2 & P^-_{k,12}P^-_{k,31} \\ P^-_{k,13}P^-_{k,11} & P^-_{k,13}P^-_{k,12} & (P^-_{k,13})^2 \end{bmatrix} \end{aligned} \tag{5.36}$$

We will use this expression to show that from time k^- to time k^+ the trace of the estimation-error covariance decreases. To see this first note that the trace of P^-_k is given as

$$\mathrm{Tr}(P^-_k) = P^-_{k,11} + P^-_{k,22} + P^-_{k,33} \tag{5.37}$$

From Equation (5.36) we see that the trace of P^+_k is given as

$$\begin{aligned} \mathrm{Tr}(P^+_k) &= P^+_{k,11} + P^+_{k,22} + P^+_{k,33} \\ &= \left(P^-_{k,11} - \frac{(P^-_{k,11})^2}{P^-_{k,11} + \sigma^2} \right) + \left(P^-_{k,22} - \frac{(P^-_{k,12})^2}{P^-_{k,11} + \sigma^2} \right) + \\ &\quad \left(P^-_{k,33} - \frac{(P^-_{k,13})^2}{P^-_{k,11} + \sigma^2} \right) \\ &= \mathrm{Tr}(P^-_k) - \frac{(P^-_{k,11})^2 + (P^-_{k,12})^2 + (P^-_{k,13})^2}{P^-_{k,11} + \sigma^2} \end{aligned} \tag{5.38}$$

When we get a new measurement, we expect our state estimate to improve. That is, we expect the covariance to decrease, and the above equation shows that it does indeed decrease. That is, the trace of P^+_k is less than the trace of P^-_k.

This system was simulated with five time units between discretization steps ($T = 5$), and a position-measurement standard deviation of 30 units. Figure 5.3 shows the variance of the position estimate ($P^-_{k,11}$ and $P^+_{k,11}$) for the first five time steps of the Kalman filter. It can be seen that the variance (uncertainty) increases from one time step to the next, but then decreases at each time step as the measurement is processed.

Figure 5.4 shows the variance of the position estimate ($P^-_{k,11}$ and $P^+_{k,11}$) for the first 60 time steps of the Kalman filter. This shows that the variance increases between time steps, and then decreases at each time step. But it

Figure 5.3 The first five time steps of the *a priori* and *a posteriori* position-estimation-error variances for Example 5.1.

Figure 5.4 The first 60 time steps of the *a priori* and *a posteriori* position-estimation-error variances for Example 5.1.

can also be seen from this figure that the variance converges to a steady-state value.

Figure 5.5 shows the position-measurement error (with a standard deviation of 30) and the error of the *a posteriori* position estimate. The estimation error starts out with a standard deviation close to 30, but by the end of the simulation the standard deviation is about 11.

∇∇∇

Figure 5.5 The position-measurement error and position estimation error for Example 5.1.

5.4 ALTERNATE PROPAGATION OF COVARIANCE

In this section, we derive an alternate equation for the propagation of the estimation-error covariance P. This alternate equation, based on [Gre01], can be used to find a closed-form equation for a scalar Kalman filter.[3] It can also be used to find a fast solution to the steady-state estimation-error covariance.

5.4.1 Multiple state systems

Recall from Equation (5.19) the update equations for the estimation-error covariance:

$$\begin{aligned} P_k^- &= F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \\ P_k^+ &= P_k^- - P_k^-H_k^T(H_kP_k^-H_k^T + R_k)^{-1}H_kP_k^- \end{aligned} \tag{5.39}$$

If the $n \times n$ matrix P_k^- can be factored as

$$P_k^- = A_kB_k^{-1} \tag{5.40}$$

where A_k and B_k are $n \times n$ matrices to be determined, then P_{k+1}^- satisfies

$$P_{k+1}^- = A_{k+1}B_{k+1}^{-1} \tag{5.41}$$

where A and B are propagated as follows:

$$\begin{bmatrix} A_{k+1} \\ B_{k+1} \end{bmatrix} = \begin{bmatrix} (F_k + Q_kF_k^{-T}H_k^TR_k^{-1}H_k) & Q_kF_k^{-T} \\ F_k^{-T}H_k^TR_k^{-1}H_k & F_k^{-T} \end{bmatrix} \begin{bmatrix} A_k \\ B_k \end{bmatrix} \tag{5.42}$$

[3]The equations given in [Gre01] have some typographical errors that have been corrected in this section.

This can be seen by noting from Equation (5.42) that

$$\begin{aligned} B_{k+1}^{-1} &= [F_k^{-T} H_k^T R_k^{-1} H_k A_k + F_k^{-T} B_k]^{-1} \\ &= \left[F_k^{-T}(H_k^T R_k^{-1} H_k A_k B_k^{-1} + I) B_k\right]^{-1} \\ &= B_k^{-1}[H_k^T R_k^{-1} H_k A_k B_k^{-1} + I]^{-1} F_k^T \end{aligned} \tag{5.43}$$

From Equation (5.42) we see that

$$A_{k+1} B_{k+1}^{-1} = [(F_k + Q_k F_k^{-T} H_k^T R_k^{-1} H_k) A_k + Q_k F_k^{-T} B_k] B_{k+1}^{-1} \tag{5.44}$$

Substituting the expression for B_{k+1}^{-1} into this equation gives

$$\begin{aligned} A_{k+1} B_{k+1}^{-1} &= [(F_k + Q_k F_k^{-T} H_k^T R_k^{-1} H_k) A_k + Q_k F_k^{-T} B_k] \times \\ &\quad B_k^{-1}[H_k^T R_k^{-1} H_k A_k B_k^{-1} + I]^{-1} F_k^T \\ &= [(F_k + Q_k F_k^{-T} H_k^T R_k^{-1} H_k) A_k B_k^{-1} + Q_k F_k^{-T}] \times \\ &\quad [H_k^T R_k^{-1} H_k A_k B_k^{-1} + I]^{-1} F_k^T \end{aligned} \tag{5.45}$$

Substituting P_k^- for $A_k B_k^{-1}$ in the above equation gives

$$\begin{aligned} A_{k+1} B_{k+1}^{-1} &= [(F_k + Q_k F_k^{-T} H_k^T R_k^{-1} H_k) P_k^- + Q_k F_k^{-T}] \times \\ &\quad [H_k^T R_k^{-1} H_k P_k^- + I]^{-1} F_k^T \\ &= [F_k P_k^- + Q_k F_k^{-T}(H_k^T R_k^{-1} H_k P_k^- + I)] \times \\ &\quad [H_k^T R_k^{-1} H_k P_k^- + I]^{-1} F_k^T \\ &= F_k P_k^- [H_k^T R_k^{-1} H_k P_k^- + I]^{-1} F_k^T + Q_k F_k^{-T} F_k^T \end{aligned} \tag{5.46}$$

Applying the matrix inversion lemma to the term in brackets gives

$$\begin{aligned} A_{k+1} B_{k+1}^{-1} &= F_k P_k^- [I - H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^-] F_k^T + Q_k \\ &= F_k [P_k^- - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^-] F_k^T + Q_k \\ &= F_k P_k^+ F_k^T + Q_k \\ &= P_{k+1}^- \end{aligned} \tag{5.47}$$

So we see that $A_{k+1} B_{k+1}^{-1} = P_{k+1}^-$.

Equation (5.42) can be used to obtain a quick solution to the steady-state covariance for multidimensional systems (although not a closed-form solution). Suppose that F, Q, H, and R are constant matrices. From Equation (5.42) we obtain

$$\begin{aligned} \begin{bmatrix} A_{k+1} \\ B_{k+1} \end{bmatrix} &= \begin{bmatrix} (F + QF^{-T} H^T R^{-1} H) & QF^{-T} \\ F^{-T} H^T R^{-1} H & F^{-T} \end{bmatrix} \begin{bmatrix} A_k \\ B_k \end{bmatrix} \\ &= \Psi \begin{bmatrix} A_k \\ B_k \end{bmatrix} \\ \begin{bmatrix} A_k \\ B_k \end{bmatrix} &= \Psi^{k-1} \begin{bmatrix} P_1^- \\ I \end{bmatrix} \end{aligned} \tag{5.48}$$

where we used the fact that $A_1 = P_1^-$ and $B_1 = I$ satisfies the original factoring of Equation (5.40). Now we can successively square Ψ a total of p times to obtain Ψ^2, Ψ^4, Ψ^8, and so on, until Ψ^{2^p} converges to a steady-state value:

$$\begin{bmatrix} A_\infty \\ B_\infty \end{bmatrix} \approx \Psi^{2^p} \begin{bmatrix} P_1^- \\ I \end{bmatrix} \quad \text{for large } p \tag{5.49}$$

The steady-state covariance is $P_\infty^- = A_\infty B_\infty^{-1}$. We can also find the steady-state Kalman gain by simply iterating the filter equations from Equation (5.19), but the method in this section could be a much quicker way to find the steady-state gain. Once we find P_∞^- as shown above, we compute $K_\infty = P_\infty^- H^T (H P_\infty^- H^T + R)^{-1}$ as the steady-state Kalman filter gain. More discussion of steady-state Kalman filtering is given in Section 7.3.

5.4.2 Scalar systems

Equation (5.42) can be used to obtain a closed-form solution for the scalar Kalman filter for time-invariant systems. Suppose that F, Q, H, and R are constant scalars. Then from Equation (5.42) we obtain

$$\begin{aligned} \begin{bmatrix} A_{k+1} \\ B_{k+1} \end{bmatrix} &= \begin{bmatrix} F + \frac{H^2 Q}{FR} & \frac{Q}{F} \\ \frac{H^2}{FR} & \frac{1}{F} \end{bmatrix} \begin{bmatrix} A_k \\ B_k \end{bmatrix} \\ &= \Psi \begin{bmatrix} A_k \\ B_k \end{bmatrix} \end{aligned} \tag{5.50}$$

where Ψ is defined by the above equation. Now find the eigendata of Ψ. Suppose that the eigenvalues of Ψ are λ_1 and λ_2, and the eigenvectors of Ψ are combined to create the 2×2 matrix M. Then

$$\Psi = M \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} M^{-1} \tag{5.51}$$

and we obtain

$$\begin{aligned} \begin{bmatrix} A_k \\ B_k \end{bmatrix} &= \Psi^{k-1} \begin{bmatrix} A_1 \\ B_1 \end{bmatrix} \\ &= M \begin{bmatrix} \lambda_1^{k-1} & 0 \\ 0 & \lambda_2^{k-1} \end{bmatrix} M^{-1} \begin{bmatrix} P_1^- \\ 1 \end{bmatrix} \end{aligned} \tag{5.52}$$

where we used the fact that $A_1 = P_1^-$ and $B_1 = 1$ satisfies the original factoring of Equation (5.40). Working through the math to obtain λ_1, λ_2, and M gives the following.

$$\begin{aligned} P_k^- &= \frac{\tau_1 \mu_1^{k-1}(2RH^2P_1^- - \tau_2) - \tau_2 \mu_2^{k-1}(2H^2P_1^- - \tau_1)}{2H^2\mu_1^{k-1}(2RH^2P_1^- - \tau_2) - 2H^2\mu_2^{k-1}(2H^2P_1^- - \tau_1)} \\ \lambda_1 &= \frac{H^2Q + R(F^2+1) + \sigma}{2FR} \\ \lambda_2 &= \frac{H^2Q + R(F^2+1) - \sigma}{2FR} \\ \sigma &= \sqrt{H^2Q + R(F+1)^2}\sqrt{H^2Q + R(F-1)^2} \\ \tau_1 &= H^2Q + R(F^2-1) + \sigma \\ \tau_2 &= H^2Q + R(F^2-1) - \sigma \\ \mu_1 &= H^2Q + R(F^2+1) + \sigma \\ \mu_2 &= H^2Q + R(F^2+1) - \sigma \end{aligned}$$

$$
\begin{aligned}
M &= \begin{bmatrix} \frac{\tau_1}{2H^2} & \frac{\tau_2}{2H^2} \\ 1 & 1 \end{bmatrix} \\
M^{-1} &= \frac{1}{\tau_1(R-1)+2\sigma} \begin{bmatrix} 2RH^2 & -\tau_1 \\ -2RH^2 & R\tau_1 \end{bmatrix}
\end{aligned} \tag{5.53}
$$

This is a closed-form equation for the time-varying Kalman filter for a scalar time-invariant system. This can easily be used to obtain the steady-state value of P_k^-. Note that $\mu_2 < \mu_1$. As k increases, μ_2^k gets smaller and smaller relative to μ_1^k. Therefore

$$
\begin{aligned}
\lim_{k\to\infty} P_k^- &= \lim_{k\to\infty} \frac{\tau_1\mu_1^{k-1}(2RH^2P_1^- - \tau_2) - \tau_2\mu_2^{k-1}(2H^2P_1^- - \tau_1)}{2H^2\mu_1^{k-1}(2RH^2P_1^- - \tau_2) - 2H^2\mu_2^{k-1}(2H^2P_1^- - \tau_1)} \\
&= \lim_{k\to\infty} \frac{\tau_1\mu_1^{k-1}(2RH^2P_1^- - \tau_2)}{2H^2\mu_1^{k-1}(2RH^2P_1^- - \tau_2)} \\
&= \frac{\tau_1}{2H^2}
\end{aligned} \tag{5.54}
$$

This gives the steady-state covariance for a scalar system.

■ EXAMPLE 5.2

In this example, we will show how a scalar covariance can be propagated. Consider the following scalar system:

$$
\begin{aligned}
x_{k+1} &= x_k + w_k \\
y_k &= x_k + v_k \\
w_k &\sim (0,1) \\
v_k &\sim (0,1)
\end{aligned} \tag{5.55}
$$

This is a very simple system but one that arises in many applications. For example, it may represent some slowly varying parameter x_k that we measure directly. The process noise term w_k accounts for the variations in x_k, and the measurement noise term v_k accounts for measurement errors. In this system, we have $F = H = Q = R = 1$. Substituting these values in Equation (5.53) gives

$$
\begin{aligned}
\tau_1 &= 1+\sqrt{5} \\
\tau_2 &= 1-\sqrt{5} \\
\mu_1 &= 3+\sqrt{5} \\
\mu_2 &= 3-\sqrt{5} \\
P_k^- &= \frac{\tau_1\mu_1^{k-1}(2P_1^- - \tau_2) - \tau_2\mu_2^{k-1}(2P_1^- - \tau_1)}{2\mu_1^{k-1}(2P_1^- - \tau_2) - 2\mu_2^{k-1}(2P_1^- - \tau_1)}
\end{aligned} \tag{5.56}
$$

Taking the limit as $k \to \infty$ gives the steady-state value of P_k^-:

$$
\begin{aligned}
P_\infty^- &= \frac{\tau_1}{2} \\
&= \frac{1+\sqrt{5}}{2} \\
&\approx 1.62
\end{aligned} \tag{5.57}
$$

Now we can use Equation (5.19) to find the steady-state value of K_k:

$$\begin{aligned} K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ &= \frac{P_k^-}{P_k^- + 1} \\ K_\infty &= \frac{1+\sqrt{5}}{3+\sqrt{5}} \\ &\approx 0.62 \end{aligned} \tag{5.58}$$

Figure 5.6 shows the *a priori* estimation covariance and the Kalman gain as a function of time, and illustrates their convergence to steady-state values. From the equation for the *a posteriori* estimation covariance, we know that $P_k^+ = (I - K_k H_k) P_k^-$. For this example we therefore see that the steady-state value of P_k^+ is given as

$$\begin{aligned} P_\infty^+ &= \left(1 - \frac{1+\sqrt{5}}{3+\sqrt{5}}\right) \frac{1+\sqrt{5}}{2} \\ &= \frac{1+\sqrt{5}}{3+\sqrt{5}} \end{aligned} \tag{5.59}$$

Figure 5.6 Estimation covariance and Kalman gain as a function of time for Example 5.2. The covariance and gain converge to steady-state values.

∇∇∇

5.5 DIVERGENCE ISSUES

The theory presented in this chapter makes the Kalman filter an attractive choice for state estimation. But when a Kalman filter is implemented on a real system it

may not work, even though the theory is correct. Two of the primary causes for the failure of Kalman filtering are finite precision arithmetic and modeling errors [Fit71].

The theory presented in this chapter assumes that the Kalman filter arithmetic is infinite precision. In digital microprocessors the arithmetic is finite precision – only a certain number of bits are used to represent the numbers in the Kalman filter equations. This may cause divergence or even instability in the implementation of the Kalman filter.

The theory presented also assumes that the system model is precisely known. It is assumed that the F, Q, H, and R matrices are exactly known, and it is assumed that the noise sequences $\{w_k\}$ and $\{v_k\}$ are pure white, zero-mean, and completely uncorrelated. If any of these assumptions are violated, as they always are in real implementations, then the Kalman filter assumptions are violated and the theory may not work.

In order to improve filter performance in the face of these realities, the designer can use several strategies:

1. Increase arithmetic precision
2. Use some form of square root filtering
3. Symmetrize P at each time step: $P = (P + P^T)/2$
4. Initialize P appropriately to avoid large changes in P
5. Use a fading-memory filter
6. Use fictitious process noise (especially for estimating "constants")

These strategies are often problem dependent and need to be explored via simulation or experimentation in order to obtain good results. Some of these strategies may be more attractive than others, depending on the specific problem.

Item 1 above, increasing arithmetic precision, simply forces the digital implementation of the filter to more closely match the analog theory. In a PC-based implementation, it may require only a trivial effort to increase the arithmetic precision – change all the variables to double precision. This trivial change may make the difference between divergence and convergence. However, in a microcontroller implementation it may not be feasible to increase the arithmetic precision.

Item 2 above, square root filtering, is a way of reformulating the filter equations. Even though the physical precision of the implementation does not change, square root filtering effectively increases arithmetic precision. This will be discussed further in Sections 6.3, 6.4, and 8.3. But square root filtering requires more computational effort, which may or may not be a major consideration for a given application. Square root filtering also adds a lot of complication to the filter equations, which invites software bugs.

Items 3 and 4 above involve forcing P to be symmetric and initializing P appropriately. These are easy solutions, but they usually do not result in major improvements to the convergence properties of the filter. However, these steps should always be implemented since they are straightforward and easy, and since they may prevent numerical problems. Note from Equation (5.19) that the P_k^- expression is already symmetric, and so there is no point to forcing symmetry for P_k^-. However, depending on which equation is used, P_k^+ may or may not be symmetric. The

expressions for P_k^+ in Equation (5.19) are mathematically equivalent, but they are not numerically equivalent. One of them has a built-in symmetry, but the others do not. If an equation for P_k^+ is used that does not have a built-in symmetry, then it is very easy and may pay large dividends to force symmetry. This has been done several different ways in the literature. One way is as described in Item 3 above; that is, after P is calculated, set $P = (P + P^T)/2$. Other ways involve forcing the terms below the diagonal to be equal to the terms above the diagonal, or forcing the eigenvalues of P to be positive.

Item 5 above is a simple way of forcing the filter to "forget" measurements in the distant past and place more emphasis on recent measurements. This causes the filter to be more responsive to measurements. It theoretically results in the loss of optimality of the Kalman filter, but it may restore convergence and stability. It is better to have a theoretically suboptimal filter that works rather than a theoretically optimal filter that does not work due to modeling errors. The greater responsiveness of the fading-memory filter to recent measurements makes the filter less sensitive to modeling errors, and hence more robust. This approach will be discussed further in Section 7.4.

Item 6 above, the use of fictitious process noise, is also easy to implement. In fact, it can be implemented in a way that is mathematically equivalent to the fading-memory filter of Item 5. Adding fictitious process noise is a way of telling the filter that you have less confidence in your system model. This causes the filter to place more emphasis on the measurements, and less emphasis on the process model (which may be incorrect) [Jaz69].

■ EXAMPLE 5.3

Let us illustrate the use of fictitious process noise with an example. Suppose we are trying to estimate a state that we think is a constant, but in reality is a ramp. In other words, we have a modeling error. Our assumed (but incorrect) model, upon which we base the Kalman filter, is given as follows:

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + v_k \\ w_k &\sim (0,0) \\ v_k &\sim (0,1) \end{aligned} \tag{5.60}$$

The assumed process noise is zero, which means that we are modeling x_k as a constant. From Equation (5.19) we derive the Kalman filter equations for this system as

$$\begin{aligned} P_k^- &= F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \\ &= P_{k-1}^+ \\ K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ &= \frac{P_k^-}{P_k^- + 1} \\ \hat{x}_k^- &= F_{k-1}\hat{x}_{k-1}^+ \\ &= \hat{x}_{k-1}^+ \end{aligned}$$

$$
\begin{aligned}
\hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \\
&= \hat{x}_k^- + K_k(y_k - \hat{x}_k^-) \\
P_k^+ &= (I - K_kH_k)P_k^-(I - K_kH_k)^T + K_kR_kK_k^T \\
&= (1 - K_k)^2P_k^- + K_k^2
\end{aligned} \tag{5.61}
$$

Suppose that the true system, although unknown to the Kalman filter designer, is given as the following two-state model:

$$
\begin{aligned}
x_{1,k+1} &= x_{1,k} + x_{2,k} \\
x_{2,k+1} &= x_{2,k} \\
y_k &= x_{1,k} + v_k \\
v_k &\sim (0,1)
\end{aligned} \tag{5.62}
$$

The first state is a ramp, which we assumed incorrectly in our system model to be a constant. Figure 5.7 shows the true state $x_{1,k}$ and the estimated state $\hat{x}_{1,k}$. It can be seen that the estimate is diverging from the true state, and the estimation error is growing without bound.

Figure 5.7 Kalman filter divergence due to mismodeling.

However, if we add fictitious process noise to the Kalman filter, then the filter will place more emphasis on the measurements, which will improve the filter performance. Figure 5.8 shows the true state and the estimated state when various values of Q are used in the Kalman filter. As the fictitious process noise gets larger, the estimation error becomes smaller. Of course, this is at the price of poorer performance in case the assumed system model is actually correct. The designer needs to add an appropriate amount of fictitious process noise to balance performance under nominal conditions with performance under mismodel conditions.

Figure 5.9 shows the time history of the Kalman gain K_k for this example for various values of Q. As expected, the gain K_k converges to a larger steady-state value when Q is larger, making the filter more responsive to

Figure 5.8 Kalman filter improvement due to fictitious process noise.

measurements [see the $\hat{x}_k^+$ expression in Equation (5.61)]. This compensates for modeling errors. As shown later in Section 7.4, the fading-memory filter accomplishes the same thing in a different way. Also note from Figure 5.9 that the steady-state Kalman gain is approximately 0.62 when $Q = 1$. This matches the results of Example 5.2.

Figure 5.9 Kalman gain for various values of process noise.

This example illustrates the general principle that model noise is good, but only to a certain extent. If a system model has too much noise then it is difficult to estimate its state. But if a system model has too little noise then

our state estimator might be overly susceptible to modeling errors.[4] When designing a model for a Kalman filter, we need to balance our confidence in our model (low noise resulting in close model tracking; i.e., low bandwidth) with a healthy self-doubt (high noise resulting in filter responsiveness; i.e., high bandwidth).

$\nabla\nabla\nabla$

Examination of the filter equations shows why adding fictitious process noise compensates for modeling errors. Recall the Kalman filter equations from Equation (5.19), some of which we repeat here:

$$\begin{aligned} P_k^- &= F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \\ K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \end{aligned} \tag{5.63}$$

If Q_k is small then the covariance may not increase very much between time samples. In Example 5.3 we had $F_k = 1$, so $P_k^- = P_{k-1}^+$ when $Q_k = 0$. But the covariance will decrease from P_k^- down to P_k^+ every time a measurement is obtained due to the measurement-update equation for the covariance. Eventually P_k^- will converge to zero. This can be seen by looking at Equation (5.26), which shows the one-step equation for P_k^-:

$$P_{k+1}^- = F_k P_k^- F_k^T - F_k K_k H_k P_k^- F_k^T + Q_k \tag{5.64}$$

If $Q_k = 0$ then this equation has a steady solution of zero. A zero value for P_k^- will result in $K_k = 0$, as seen from Equation (5.63). A zero value for K_k means that the measurement-update equation (5.63) for $\hat{x}$ will not take any account of the measurement – that is, the measurement y_k will be completely ignored in the computation of $\hat{x}^+$. This is because the measurement noise covariance R_k (assuming it is greater than zero) will be infinitely times larger than the process noise $Q_k = 0$. The filter will become sluggish in the sense that it will not respond to measurements.

On the other hand, if Q_k is larger, then the covariance will always increase between time samples – that is, P_k^- will always be larger than P_{k-1}^+. When P_k^- converges, it will converge to a larger value. This will make K_k converge to a larger value. A larger K_k means that the measurement update for $\hat{x}$ in Equation (5.63) will include a larger emphasis on the measurement – that is, the filter will pay more attention to the measurements.

5.6 SUMMARY

In this chapter, we have presented the essence of the discrete-time Kalman filter. Over the past few decades, this estimation algorithm has found applications in virtually every area of engineering. We have seen that the Kalman filter equations can be written in several different ways, each of which may appear quite different than the others, although they are all mathematically equivalent. We have seen that

[4]Noise, like most things in life, is beneficial in moderate amounts. We also see this in human psychological responses to noise. Too much noise will drive humans insane, but too little noise might also result in a loss of sanity. Noise is especially beneficial for controls engineers, who would not only lose their sanity but would also lose their research funding if not for noise [Bar01, p. 179].

the Kalman filter is optimal even when the noise is not Gaussian. The Kalman filter is the optimal estimator when the noise is Gaussian, and it is the optimal *linear* estimator when the noise is not Gaussian. We have seen that the Kalman filter may not perform well if the underlying assumptions do not hold, and we briefly mentioned some ways to compensate for violated assumptions. The later chapters of this book will expand and generalize the results presented in this chapter.

PROBLEMS

Written exercises

5.1 A radioactive mass has a half-life of τ seconds. At each time step the number of emitted particles x is half of what it was one time step ago, but there is some error w_k (zero-mean with variance Q) in the number of emitted particles due to background radiation. At each time step, the number of emitted particles is counted. The instrument used to count the number of emitted particles has a random error at time k of v_k, which is zero-mean with a variance of R. Assume that w_k and v_k are uncorrelated.

a) Write the linear system equations for this system.

b) Suppose we want to use a Kalman filter to find the optimal estimate of the number of emitted particles at each time step. Write the one-step *a posteriori* Kalman filter equations for this system.

c) Find the steady-state *a posteriori* estimation-error variance for the Kalman filter.

d) What is the steady-state Kalman gain when $Q = R$? What is the steady-state Kalman gain when $Q = 2R$? Give an intuitive explanation for why the steady-state gain changes the way it does when the ratio of Q to R changes.

5.2 This problem illustrates the robustness that is achieved by the use of the Joseph form of the covariance measurement update equation. Suppose you have a discrete-time Kalman filter for a scalar system.

a) Find $\partial P_k^+ / \partial K_k$ for the third form of the covariance measurement update in Equation (5.19).

b) Find $\partial P_k^+ / \partial K_k$ for the Joseph form (the first form) of the covariance measurement update in Equation (5.19). After you get your answer, substitute for K_k from the Kalman gain expression.

c) Use the above results to explain why the Joseph form of the covariance measurement-update equation is stable and robust.

5.3 Prove that $E[\hat{x}_k^+(\tilde{x}_k^+)^T] = 0$. Hint: Since $\hat{x}_0^+ = E[x_0]$ is a constant and $\tilde{x}_0^+ = x_0 - \hat{x}_0^+$ is zero-mean, we know that $E[\hat{x}_0^+(\tilde{x}_0^+)^T] = 0$. Given this information, prove that $E[\hat{x}_1^+(\tilde{x}_1^+)^T] = 0$. From this point, use induction to complete the proof.

5.4 Suppose that you have a fish tank with x_p piranhas and x_g guppies [Bay99]. Once per week, you put guppy food into the tank (which the piranhas do not eat). Each week the piranhas eat some of the guppies. The birth rate of the piranhas is proportional to the guppy population, and the death rate of the piranhas is

proportional to their own population (due to overcrowding). Therefore $x_p(k+1) = x_p(k) + k_1 x_g(k) - k_2 x_p(k) + w_p(k)$, where k_1 and k_2 are proportionality constants and $w_p(k)$ is white noise with a variance of one that accounts for mismodeling. The birth rate of the guppies is proportional to the food supply u, and the death rate of the guppies is proportional to the piranha population. Therefore, $x_g(k+1) = x_g(k) + u(k) - k_3 x_p(k) + w_g(k)$, where k_3 is a proportionality constant and $w_g(k)$ is white noise with a variance of one that accounts for mismodeling. The step size for this model is one week. Every week, you count the piranhas and guppies. You can count the piranhas accurately because they are so large, but your guppy count has zero-mean noise with a variance of one. Assume that $k_1 = 1$ and $k_2 = k_3 = 1/2$.

a) Generate a linear state-space model for this system.

b) Suppose that at the initial time you have a perfect count for x_p and x_g. Using a Kalman filter to estimate the guppy population, what is the variance of your guppy population estimate after one week? What is the variance after two weeks?

c) What is the ratio of the piranha population to the guppy population when they reach steady state? Assume that the process noise is zero for this part of the problem.

5.5 The measured output of a simple moving average process is $y_k = z_k + z_{k-1}$, where $\{z_j\}$ is zero-mean white noise with a variance of one.

a) Generate a state-space description for this system with the first element of x_k equal to z_{k-1} and second element equal to z_k.

b) Suppose that the initial estimation-error covariance is equal to the identity matrix. Show that the *a posteriori* estimation-error covariance is given by

$$P_k^+ = \frac{1}{k+1}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

c) Find $E\left[||x_k - \hat{x}_k^+||_2^2\right]$ as a function of k.

5.6 In this problem, we use the auxiliary variable $S_k = H_k P_k^- H_k^T + R_k$. Note that

$$\begin{bmatrix} I & 0 \\ -P_k^- H_k^T S_k^{-1} & I \end{bmatrix}\begin{bmatrix} S_k & H_k P_k^- \\ P_k^- H_k^T & P_k^- \end{bmatrix} = \begin{bmatrix} S_k & H_k P_k^- \\ 0 & P_k^+ \end{bmatrix}$$

Use the product rule for determinants to show that

$$|P_k^+| = \frac{|P_k^-||R_k|}{|S_k|}$$

5.7 In Section 4.1, we saw that Σ_k, the covariance of the state of a discrete-time system, is given as $\Sigma_{k+1} = F_k \Sigma_k F_k^T + Q_k$. Use this along with the one-step expression for the *a priori* estimation-error covariance of the Kalman filter to show that $\Sigma_k - P_k^- \geq 0$ for all k. Give an intuitive explanation for this expression [And79].

5.8 Consider the system of Problem 5.1.

a) Use the method of Section 5.4 to find a closed-form solution for P_k^-, assuming that $Q = 1$, $R = 5$, and $P_0 = 0$.

b) Use your result from above to find the steady-state value of P_k^-.

5.9 Suppose that a Kalman filter is designed for the system

$$\begin{aligned} x_{k+1} &= x_k \\ y_k &= x_k + v_k \\ v_k &\sim (0, R) \end{aligned}$$

a) Suppose that $E(x_0^2) = 1$. Design a Kalman filter for the system and find a closed-form expression for P_k^-. What is the limit of P_k^- as $k \to \infty$?

b) Now suppose that the true process equation is actually $x_{k+1} = x_k + w_k$, where $w_k \sim (0, Q)$. Find a difference equation for the variance of the *a priori* estimation error if the Kalman filter that you designed in part (a) is used to estimate the state. What is the limit of the estimation-error variance as $k \to \infty$?

5.10 Suppose that a Kalman filter is designed for a discrete LTI system with an assumed measurement noise covariance of R, but the actual measurement noise covariance is $(R + \Delta R)$. The output of the Kalman filter will indicate that the *a priori* estimation-error covariance is P_k^-, but the actual *a priori* estimation-error covariance will be Σ_k^-. Find a difference equation for $\Delta_k = (\Sigma_k^- - P_k^-)$. Will Δ_k always be positive definite?

Computer exercises

5.11 Let p_k denote the wombat population at time k, and f_k denote the size of the wombat's food supply at time k. From one time step to the next, half of the existing wombat population dies, but the number of new wombats is added to the population is equal to twice the food supply. The food supply is constant except for zero-mean random fluctuations with a variance of 10. At each time step the wombat population is counted with an error that has zero mean and a variance of 10. The initial state is

$$\begin{aligned} p_0 &= 650 \\ f_0 &= 250 \end{aligned}$$

The initial state estimate and uncertainty is

$$\begin{aligned} \hat{p}_0 &= 600 \\ E[(\hat{p}_0 - p_0)^2] &= 500 \\ \hat{f}_0 &= 200 \\ E[(\hat{f}_0 - f_0)^2] &= 200 \end{aligned}$$

Design a Kalman filter to estimate the population and food supply.

a) Simulate the system and the Kalman filter for 10 time steps. Hand in the following.

- Source code listing.
- A plot showing the true population and the estimated population as a function of time.

- A plot showing the true food supply and the estimated food supply as a function of time.
- A plot showing the standard deviation of the population and food supply estimation error as a function of time.
- A plot showing the elements of the Kalman gain matrix as a function of time.

b) Compare the standard deviation of the estimation error of your simulation with the steady-state theoretical standard deviation based on P_k^+. Why is there such a discrepancy?

c) Run the simulation again for 1000 time steps and compare the experimental estimation error standard deviation with the theoretical standard deviation.

5.12 Consider the RLC circuit described in Problem 1.18 with $R = 3$, $L = 1$, and $C = 0.5$. The input voltage is zero-mean, unity variance white noise. Suppose that the capacitor voltage is measured at 10 Hz with zero-mean, unity variance white noise. Design a Kalman filter to estimate the inductor current, with an initial covariance $P_0^+ = 0$. Generate a plot showing the *a priori* and *a posteriori* variances of the inductor current estimate for 20 time steps. Based on the plot, what is the steady-state value of P_k^-? Use the development of Section 5.4.1 to approximate the steady-state value of P_k^- using 1, 2, 3, and 4 successive squares of the Ψ matrix.

CHAPTER 6

Alternate Kalman filter formulations

> Our experiences with estimation and control applications engineers, however, indicates that they generally prefer the seemingly simpler Kalman filter algorithms for computer implementation and they dismiss reported instances of numerical failure.
>
> —Gerald Bierman and Catherine Thornton [Bie77a]

In this chapter, we will look at some alternate ways of writing the Kalman filter equations. There are a number of mathematically equivalent ways of writing the Kalman filter equations. This can be confusing. You might read two different papers or books that present the Kalman filter equations, and the equations might look completely different. You may not know if one of the equations has a typographical error, or if they are mathematically equivalent. So you try to prove the equivalence of the two sets of equations only to arrive at a mathematical dead end, because it is not always easy to prove the equivalence of two sets of equations. This chapter derives some Kalman filter formulations that are different than (but mathematically equivalent to) the equations we derived in Chapter 5. This chapter also illustrates their advantages and disadvantages.

The first alternate formulation that we discuss is called the sequential Kalman filter, derived in Section 6.1. Sequential Kalman filtering allows for the implementation of the Kalman filter without matrix inversion. This can be a great benefit, especially in an embedded system that does not have matrix libraries, but it only

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

makes sense if certain conditions are satisfied. The second formulation that we discuss is called information filtering, derived in Section 6.2. Information filtering propagates the inverse of the covariance matrix (i.e., P^{-1}) instead of P, and is computationally cheaper than Kalman filtering under certain conditions. The third formulation that we discuss is called square root filtering, derived in Section 6.3. Square root filtering effectively increases the precision of the Kalman filter, which can help prevent divergence and instability. However, this is at the cost of increased computational effort. The final formulation that we discuss is called U-D filtering, derived in Section 6.4. This is another way to implement square root filtering, which helps to prevent numerical difficulties in the implementation of the Kalman filter.

6.1 SEQUENTIAL KALMAN FILTERING

In this section, we derive the sequential Kalman filter. This is a way of implementing the Kalman filter without matrix inversion. This can be a great advantage, especially in an embedded system that may not have matrix routines. However, the use of sequential Kalman filtering only makes sense if certain conditions are satisfied, which we will discuss in this section.

Recall the Kalman filter measurement update formulas from Equation (5.16):

$$\begin{aligned} y_k &= H_k x_k + v_k \\ K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k \hat{x}_k^-) \\ P_k^+ &= (I - K_k H_k) P_k^- \end{aligned} \tag{6.1}$$

The computation of K_k requires the inversion of an $r \times r$ matrix, where r is the number of measurements. This is depicted in Figure 6.1.

Figure 6.1 The measurement-update equation of the standard Kalman filter requires an $r \times r$ matrix inversion, where r is the number of measurements.

Suppose that instead of measuring y_k at time k, we obtain r separate measurements at time k. That is, we first measure $y_k(1)$, then $y_k(2)$, $\cdots$, and finally $y_k(r)$. We will use the shorthand notation y_{ik} for the ith element of the measurement vector y_k. Assume for now that R_k (the covariance of measurement y_k) is diagonal;

that is, R_k is given as

$$R_k = \begin{bmatrix} R_{1k} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{rk} \end{bmatrix} \tag{6.2}$$

We will also use the notation that H_{ik} is the ith row of H_k, and v_{ik} is the ith element of v_k. Then we obtain

$$\begin{aligned} y_{ik} &= H_{ik}x_k + v_{ik} \\ v_{ik} &\sim (0, R_{ik}) \end{aligned} \tag{6.3}$$

So instead of processing the measurements at time k as a vector, we will implement the Kalman filter measurement-update equation one measurement at a time. We use the notation that K_{ik} is the Kalman gain that is used to process the ith measurement at time k, $\hat{x}_{ik}^+$ is the optimal estimate after the ith measurement has been processed at time k, and P_{ik}^+ is the estimation-error covariance after the ith measurement at time k has been processed. We can see from these definitions that

$$\begin{aligned} \hat{x}_{0k}^+ &= \hat{x}_k^- \\ P_{0k}^+ &= P_k^- \end{aligned} \tag{6.4}$$

That is, $\hat{x}_{0k}^+$ is the estimate after zero measurements have been processed, so it is equal to the *a priori* estimate. Similarly, P_{0k}^+ is the estimation-error covariance after zero measurements have been processed, so it is equal to the *a priori* estimation-error covariance. The gain K_{ik} and covariance P_{ik}^+ are obtained from the normal Kalman filter measurement-update equations, with the understanding that they apply to the scalar measurement y_{ik}. For $i = 1, \cdots, r$ we have

$$\begin{aligned} K_{ik} &= P_{i-1,k}^+ H_{ik}^T (H_{ik} P_{i-1,k}^+ H_{ik}^T + R_{ik})^{-1} \\ \hat{x}_{ik}^+ &= \hat{x}_{i-1,k}^+ + K_{ik}(y_{ik} - H_{ik}\hat{x}_{i-1,k}^+) \\ P_{ik}^+ &= (I - K_{ik}H_{ik})P_{i-1,k}^+ \end{aligned} \tag{6.5}$$

After all r measurements are processed, we set $\hat{x}_k^+ = \hat{x}_{rk}^+$, and $P_k^+ = P_{rk}^+$, and we have our *a posteriori* estimate and error covariance at time k. The sequential Kalman filter does not require any matrix inversions because all of the expressions in Equation (6.5) are scalar operations. This process is depicted in Figure 6.2. The sequential Kalman filter can be summarized as follows.

The sequential Kalman filter

1. The system and measurement equations are given as

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \end{aligned} \tag{6.6}$$

where w_k and v_k are uncorrelated white noise sequences. The measurement covariance R_k is a diagonal matrix given as

$$R_k = \mathrm{diag}(R_{1k}, \cdots, R_{rk}) \tag{6.7}$$

Figure 6.2 The measurement update equation of the sequential Kalman filter requires r scalar divisions (where r is the number of measurements) because the measurements at each time step are processed sequentially. This is in contrast to the standard Kalman filter processing that is depicted in Figure 6.1.

2. The filter is initialized as

$$\begin{aligned} \hat{x}^+_0 &= E(x_0) \\ P^+_0 &= E[(x_0 - \hat{x}^+_0)(x_0 - \hat{x}^+_0)^T] \end{aligned} \tag{6.8}$$

3. At each time step k, the time-update equations are given as

$$\begin{aligned} P^-_k &= F_{k-1}P^+_{k-1}F^T_{k-1} + Q_{k-1} \\ \hat{x}^-_k &= F_{k-1}\hat{x}^+_{k-1} + G_{k-1}u_{k-1} \end{aligned} \tag{6.9}$$

This is the same as the standard Kalman filter.

4. At each time step k, the measurement-update equations are given as follows.

   (a) Initialize the *a posteriori* estimate and covariance as

$$\begin{aligned} \hat{x}^+_{0k} &= \hat{x}^-_k \\ P^+_{0k} &= P^-_k \end{aligned} \tag{6.10}$$

   These are the *a posteriori* estimate and covariance at time k after zero measurements have been processed; that is, they are equal to the *a priori* estimate and covariance.

   (b) For $i = 1, \cdots, r$ (where r is the number of measurements), perform the following:

$$\begin{aligned} K_{ik} &= \frac{P^+_{i-1,k}H^T_{ik}}{H_{ik}P^+_{i-1,k}H^T_{ik} + R_{ik}} \\ &= \frac{P^+_{ik}H^T_{ik}}{R_{ik}} \\ \hat{x}^+_{ik} &= \hat{x}^+_{i-1,k} + K_{ik}(y_{ik} - H_{ik}\hat{x}^+_{i-1,k}) \\ P^+_{ik} &= (I - K_{ik}H_{ik})P^+_{i-1,k}(I - K_{ik}H_{ik})^T + K_{ik}R_{ik}K^T_{ik} \end{aligned}$$

$$
\begin{aligned}
&= \left[(P^+_{i-1,k})^{-1} + H^T_{ik} H_{ik}/R_{ik}\right]^{-1} \\
&= (I - K_{ik} H_{ik}) P^+_{i-1,k}
\end{aligned} \tag{6.11}
$$

(c) Assign the *a posteriori* estimate and covariance as

$$
\begin{aligned}
\hat{x}^+_k &= \hat{x}^+_{rk} \\
P^+_k &= P^+_{rk}
\end{aligned} \tag{6.12}
$$

The development above assumes that the measurement-noise covariance R_k is diagonal. What if R_k is not diagonal? Suppose that $R_k = R$ is not diagonal, but it is a constant matrix. We perform a Jordan form decomposition of R by finding a matrix S such that

$$
R = S\hat{R}S^{-1} \tag{6.13}
$$

$\hat{R}$ is a diagonal matrix containing the eigenvalues of R, and S is an orthogonal matrix (i.e., $S^{-1} = S^T$) containing the eigenvectors of R. This decomposition is always possible if R is symmetric positive definite, as discussed in most linear systems books [Bay99, Che99, Kai00]. Now define a new measurement $\tilde{y}_k$ as

$$
\begin{aligned}
\tilde{y}_k &= S^{-1} y_k \\
&= S^{-1}(H_k x_k + v_k) \\
&= \tilde{H}_k x_k + \tilde{v}_k
\end{aligned} \tag{6.14}
$$

where $\tilde{H}_k$ and $\tilde{v}_k$ are defined by the above equation. The covariance of $\tilde{v}_k$ can be obtained as

$$
\begin{aligned}
E(\tilde{v}_k \tilde{v}_k^T) &= E(S^{-1} v_k v_k^T S^{-T}) \\
&= E(S^{-1} v_k v_k^T S) \\
&= S^{-1} E(v_k v_k^T) S \\
&= S^{-1} R S \\
&= \hat{R}
\end{aligned} \tag{6.15}
$$

So we have introduced a normalized measurement $\tilde{y}_k$ that has a diagonal noise covariance. Now we can implement the sequential Kalman filter equations, except that we use the measurement $\tilde{y}_k$ instead of y_k, the measurement matrix $\tilde{H}_k$ instead of H_k, and the measurement noise covariance $\hat{R}$.

Note that this procedure would not make sense if R were time-varying, because in that case we would have to perform a Jordan form decomposition at each step of the Kalman filter. That would be a lot of computational effort in order to avoid a matrix inversion. However, if R is constant and it is known before the implementation of the Kalman filter, then we can perform the Jordan form decomposition offline and use the sequential Kalman filter to our advantage.

In summary, it only makes sense to use the sequential Kalman filter if one of the following two conditions holds:

1. The measurement noise covariance R_k is diagonal
2. The measurement noise covariance R is a constant.

Finally, note that the term sequential filtering is sometimes used synonymously with the Kalman filter. That is, sequential is often used as a synonym for recursive [Buc68, Chapter 13], [Bro96]. This can cause some confusion in terminology. However, sequential filtering is usually used in the literature as we use it in this section; that is, sequential filtering is a filtering method that processes measurements one at a time (rather than processing the measurements as a whole vector). Sometimes, the standard Kalman filter is called the batch Kalman filter to distinguish it from the sequential Kalman filter.

■ EXAMPLE 6.1

The change x_k from one week to the next of an American football team's ranking is related to the team's performance against that week's opponent. The expected relationships between various normalized game measures y_{ik} and the team's ranking change at the kth week are given as

$$\begin{aligned} y_{1k} &= x_k + v_{1k} = \text{ point differential} \\ y_{2k} &= \frac{1}{5}x_k + v_{2k} = \text{ turnover differential} \\ y_{3k} &= \frac{1}{50}x_k + v_{3k} = \text{ yardage differential} \end{aligned} \tag{6.16}$$

where $v_{1k} \sim (0,2)$, $v_{2k} \sim (0,1)$, and $v_{3k} \sim (0,50)$. Before the first game of the season is played, it is expected that the team ranking will increase by one due to certain players having returned from injuries. The variance of this *a priori* estimate is 4. Uncertainty in ownership conditions is expected to decrease the team's ranking by 5% each week, with a variance of 2. The system can therefore be modeled as

$$\begin{aligned} x_{k+1} &= 0.95x_k + w_k \\ y_k &= \begin{bmatrix} 1 & 1/5 & 1/50 \end{bmatrix}^T x_k + v_k \\ w_k &\sim (0,Q) \qquad Q = 2 \\ v_k &\sim (0,R) \qquad R = \text{diag}(2,1,50) \\ \hat{x}_0^+ &= 1 \\ P_0^+ &= 4 \end{aligned} \tag{6.17}$$

Suppose that the team plays its first game and wins by six points, gains three more turnovers than its opponent, and is outgained by 100 yards. That is, $y_1 = \begin{bmatrix} 6 & 3 & -100 \end{bmatrix}^T$. The standard Kalman filter adjusts the team's ranking as follows:

$$\begin{aligned} P_1^- &= FP_0^+F^T + Q \\ &= 5.61 \\ \hat{x}_1^- &= 0.95\hat{x}_0^+ \\ &= 0.95 \\ K_1 &= P_1^-H^T(HP_1^-H^T + R)^{-1} \\ &= \begin{bmatrix} 0.6961 & 0.2785 & 0.0006 \end{bmatrix} \\ \hat{x}_1^+ &= \hat{x}_1^- + K_1(y_1 - H\hat{x}_1^-) \end{aligned}$$

$$
\begin{aligned}
&= 5.1922 \\
P_1^+ &= (I - K_1 H) P_1^- \\
&= 1.3923
\end{aligned} \tag{6.18}
$$

The K_1 calculation requires the inversion of a 3×3 matrix. On the other hand, the sequential Kalman filter could be used to update the estimated team ranking as follows:

$$
\begin{aligned}
P_1^- &= F P_0^+ F^T + Q \\
&= 5.61 \\
\hat{x}_1^- &= 0.95 \hat{x}_0^+ \\
&= 0.95 \\
\hat{x}_{01}^+ &= \hat{x}_1^- \\
P_{01}^+ &= P_1^-
\end{aligned} \tag{6.19}
$$

The first measurement is processed as follows:

$$
\begin{aligned}
K_{11} &= P_{01}^+ H_1^T (H_1 P_{01}^+ H_1^T + R_{11})^{-1} \\
&= 0.7372 \\
\hat{x}_{11}^+ &= \hat{x}_{01}^+ + K_{11}(y_{11} - H_1 \hat{x}_{01}^+) \\
&= 4.6728 \\
P_{11}^+ &= (I - K_{11} H_1) P_{01}^+ \\
&= 1.4744
\end{aligned} \tag{6.20}
$$

The second measurement is processed as follows:

$$
\begin{aligned}
K_{21} &= P_{11}^+ H_2^T (H_2 P_{11}^+ H_2^T + R_{22})^{-1} \\
&= 0.2785 \\
\hat{x}_{21}^+ &= \hat{x}_{11}^+ + K_{21}(y_{21} - H_2 \hat{x}_{11}^+) \\
&= 5.2479 \\
P_{21}^+ &= (I - K_{21} H_2) P_{11}^+ \\
&= 1.3923
\end{aligned} \tag{6.21}
$$

The third measurement is processed as follows:

$$
\begin{aligned}
K_{31} &= P_{21}^+ H_3^T (H_3 P_{21}^+ H_3^T + R_{33})^{-1} \\
&= 0.0006 \\
\hat{x}_{31}^+ &= \hat{x}_{21}^+ + K_{31}(y_{33} - H_3 \hat{x}_{21}^+) \\
&= 5.1922 \\
P_{31}^+ &= (I - K_{31} H_3) P_{21}^+ \\
&= 1.3923
\end{aligned} \tag{6.22}
$$

The sequential Kalman filter requires three loops through the measurement update equations, but no matrix inversions are required.

$\nabla\nabla\nabla$

6.2 INFORMATION FILTERING

In this section, we discuss information filtering. This is an implementation of the Kalman filter that propagates the inverse of P rather than propagating P; that is, information filtering propagates the information matrix of the system. Recall that

$$P = E[(x - \hat{x})(x - \hat{x})^T] \tag{6.23}$$

That is, P represents the uncertainty in the state estimate. If P is "large" then we have a lot of uncertainty in our state estimate. In the limit as $P \to 0$ we have perfect knowledge of x, and as $P \to \infty$ we have zero knowledge of x. The information matrix is defined as

$$\mathcal{I} = P^{-1} \tag{6.24}$$

That is, $\mathcal{I}$ represents the certainty in the state estimate. If $\mathcal{I}$ is "large" then we have a lot of confidence in our state estimate. In the limit as $\mathcal{I} \to 0$ we have zero knowledge of x, and as $\mathcal{I} \to \infty$ we have perfect knowledge of x.

Recall from Equation (5.19) that the measurement update equation for P can be written as

$$(P_k^+)^{-1} = (P_k^-)^{-1} + H_k^T R_k^{-1} H_k \tag{6.25}$$

Substituting the definition of $\mathcal{I}$ into this equation gives

$$\mathcal{I}_k^+ = \mathcal{I}_k^- + H_k^T R_k^{-1} H_k \tag{6.26}$$

This gives the measurement-update equation for the information matrix. Recall from Equation (5.19) the time-update equation for P:

$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q_{k-1} \tag{6.27}$$

This implies that

$$\mathcal{I}_k^- = [F_{k-1}(\mathcal{I}_{k-1}^+)^{-1} F_{k-1}^T + Q_{k-1}]^{-1} \tag{6.28}$$

Now we can use the matrix inversion lemma from Section 1.1.2, which we restate here:

$$(A + BD^{-1}C)^{-1} = A^{-1} - A^{-1}B(D + CA^{-1}B)^{-1}CA^{-1} \tag{6.29}$$

If we make the identifications $A = Q_{k-1}$, $B = F_{k-1}$, $C = F_{k-1}^T$, and $D = \mathcal{I}_{k-1}^+$, then we can apply the matrix inversion lemma to Equation (6.28) to obtain

$$\mathcal{I}_k^- = Q_{k-1}^{-1} - Q_{k-1}^{-1} F_{k-1} (\mathcal{I}_{k-1}^+ + F_{k-1}^T Q_{k-1}^{-1} F_{k-1})^{-1} F_{k-1}^T Q_{k-1}^{-1} \tag{6.30}$$

This gives the time-update equation for the information matrix. The information filter can be summarized as follows.

The information filter

1. The dynamic system is given by the following equations:

$$\begin{aligned} x_k &= F_{k-1} x_{k-1} + G_{k-1} u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \\ w_k &\sim (0, Q_k) \end{aligned}$$

$$\begin{aligned} v_k &\sim (0, R_k) \\ E(w_k w_j^T) &= Q_k \delta_{k-j} \\ E(v_k v_j^T) &= R_k \delta_{k-j} \\ E(w_k v_k^T) &= 0 \end{aligned} \tag{6.31}$$

2. The Kalman filter is initialized as follows:

$$\begin{aligned} \hat{x}_0^+ &= E(x_0) \\ \mathcal{I}_0^+ &= \left\{E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]\right\}^{-1} \end{aligned} \tag{6.32}$$

3. The information filter is given by the following equations, which are computed for each time step $k = 1, 2, \cdots$:

$$\begin{aligned} \mathcal{I}_k^- &= Q_{k-1}^{-1} - Q_{k-1}^{-1} F_{k-1} (\mathcal{I}_{k-1}^+ + F_{k-1}^T Q_{k-1}^{-1} F_{k-1})^{-1} F_{k-1}^T Q_{k-1}^{-1} \\ \mathcal{I}_k^+ &= \mathcal{I}_k^- + H_k^T R_k^{-1} H_k \\ K_k &= (\mathcal{I}_k^+)^{-1} H_k^T R_k^{-1} \\ \hat{x}_k^- &= F_{k-1} \hat{x}_{k-1}^+ + G_{k-1} u_{k-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k (y_k - H_k \hat{x}_k^-) \end{aligned} \tag{6.33}$$

The standard Kalman filter equations require the inversion of an $r \times r$ matrix, where r is the number of measurements. The information filter equations require at least a couple of $n \times n$ matrix inversions, where n is the number of states. Therefore, if $r \gg n$ (i.e., we have significantly more measurements than states) it may be computationally more efficient to use the information filter. It could be argued that since the Kalman gain is given as

$$K_k = P_k^+ H_k^T R_k^{-1} \tag{6.34}$$

we have to perform and $r \times r$ matrix inversion on R_k anyway, whether we use the standard Kalman filter or the information filter. But if R_k is constant, then we could invert it as part of the initialization process, so the Kalman gain equation may not require this $r \times r$ matrix inversion after all. The same thinking also applies to the inversion of Q_{k-1}.

If the initial uncertainty is infinite, we cannot numerically set $P_0^+ = \infty$, but we can numerically set $\mathcal{I}_0^+ = 0$. This makes the information filter more mathematically precise for the zero initial certainty case. However, if the initial uncertainty is zero (i.e., we have perfect knowledge of x_0), we can numerically set $P_0^+ = 0$, but we cannot numerically set $\mathcal{I}_0^+ = \infty$. This makes the standard Kalman filter more mathematically precise for the zero initial uncertainty case

■ EXAMPLE 6.2

The information filter can be used to solve the American football team ranking problem of Example 6.1. The information filter equations are given as

$$\begin{aligned} \mathcal{I}_1^- &= Q^{-1} - Q^{-1} F (\mathcal{I}_0^+ + F^T Q^{-1} F)^{-1} F^T Q^{-1} \\ &= 0.1783 \end{aligned}$$

$$
\begin{aligned}
\mathcal{I}_1^+ &= \mathcal{I}_1^- + H^T R^{-1} H \\
&= 0.7183 \\
K_1 &= (\mathcal{I}_1^+)^{-1} H^T R^{-1} \\
&= \begin{bmatrix} 0.6961 & 0.2785 & 0.0006 \end{bmatrix} \\
\hat{x}_1^- &= F\hat{x}_0^+ \\
&= 0.95 \\
\hat{x}_1^+ &= \hat{x}_1^- + K_1(y_1 - H\hat{x}_1^-) \\
&= 5.1922
\end{aligned} \tag{6.35}
$$

The information filter requires the inversion of Q and R, but in many applications these matrices are constant and can therefore be inverted offline. The only other matrix inversions are in the $\mathcal{I}_k^-$ and K_k equations. These inversions are scalar in this example because there is only one state in this example.

$\nabla\nabla\nabla$

6.3 SQUARE ROOT FILTERING

The early days of Kalman filtering in the 1960s saw a lot of promise and successful applications in the aerospace industry and in NASA's space program, but sometimes problems arose in implementation. Many of the problems that were encountered were due to numerical difficulties. The Riccati equation solution P_k should theoretically always be a symmetric positive semidefinite matrix, but numerical problems in computer implementations sometimes led to P_k matrices that became indefinite or nonsymmetric. This was often because of the short word lengths in the computers of the 1960s [Sch81]. Numerical problems may arise in cases in which some elements of the state-vector x are estimated to much greater precision than other elements of x. This could be because of discrepancies in the units of the state-vector elements. For example, one state might be in units of miles and can be estimated to within 0.01 miles, whereas a second state might be in units of cm/s and can be estimated to within 10 cm/s. The covariance for the first state would be on the order of 10^{-4}, whereas the covariance for the second state would be on the order of 10^2. This led to a lot of research during the 1960s that was related to numerical implementations.

Square root filtering is a way to mathematically increase the precision of the Kalman filter when hardware precision is not available. Perhaps the first square root algorithm was developed by James Potter for NASA's Apollo space program [Bat64]. Although Potter's algorithm was limited to zero process noise and scalar measurements, its success led to a lot of additional square root research in the following years. Potter's algorithm was extended to handle process noise in [And68, Dye69], and was generalized in two different ways to handle vector measurements in [Bel67, And68]. Paul Kaminski gives a good review of square root filtering developments during the first decade of the Kalman filter [Kam71].

Now that computers have become so much more capable, we do not have to worry about numerical problems as often. Nevertheless, numerical issues still arise in finite-word-length implementations of algorithms, especially in embedded systems. In this section, we will discuss the square root filter, which was developed in order to

effectively increase the numerical precision of the Kalman filter and hence mitigate numerical difficulties in implementations. However, this improved performance is at the cost of greater computational effort. First, we will review the concept of the condition number of a matrix, then we will derive the square root version of the time update equation, and finally we will derive the square root version of the measurement update equations. Section 8.3.3 contains a discussion of square root filtering for the continuous-time Kalman filter.

6.3.1 Condition number

Recall the definition of the singular values of a matrix. An $n \times n$ matrix P has n singular values σ, given as

$$\begin{aligned} \sigma^2(P) &= \lambda(P^T P) \\ &= \lambda(P P^T) \end{aligned} \tag{6.36}$$

The matrix $P^T P$ is symmetric, and the eigenvalues of a symmetric matrix are always real and nonnegative, so the singular values of a matrix are always real and nonnegative. The matrix P is nonsingular (invertible) if and only if all of its singular values are positive. The condition number of a matrix is defined as

$$\begin{aligned} \kappa(P) &= \frac{\sigma_{\max}(P)}{\sigma_{\min}(P)} \\ &\geq 1 \end{aligned} \tag{6.37}$$

Note that some authors use alternate definitions for condition number; for example, some authors define the condition number of a matrix as the square of the above definition.[1] As $\kappa(P) \to \infty$, the matrix P is said to be poorly conditioned or ill conditioned, and P approaches a singular matrix. In the implementation of a Kalman filter, the error covariance matrix P should always be positive definite because $P = E[(x - \hat{x})(x - \hat{x})^T]$. We use the standard notation

$$P > 0 \tag{6.38}$$

to indicate that P is positive definite. This is equivalent to saying that P is invertible, which is equivalent to saying that all of the eigenvalues of P are greater than zero. But suppose in our Kalman filter that some elements of x are estimated to much greater precision than other elements of x. For example, suppose that

$$P = \begin{bmatrix} 10^6 & 0 \\ 0 & 10^{-6} \end{bmatrix} \tag{6.39}$$

This means that our estimate of x_1 has a standard deviation of 10^3 and our estimate of x_2 has a standard deviation of 10^{-3}. This could be due to drastically different units in x_1 and x_2, or it could be simply that x_1 is much more observable that x_2. The singular values of a diagonal matrix are the magnitudes of the diagonal elements, which are 10^6 and 10^{-6}. In other words,

$$\kappa(P) = 10^{12} \tag{6.40}$$

[1]In MATLAB the COND function can be used to find the condition number of a matrix.

This is a pretty large condition number, which means that the P matrix might look like a singular matrix to a digital computer. For example, if we have a fixed-point computer with 10 decimal digits of precision and the 10^6 term is represented correctly in the computer, then the 10^{-6} term will be represented as a zero in the computer. Mathematically, P is nonsingular, but computationally P is singular.

The square root filter is based on the idea of finding an S matrix such that $P = SS^T$. The S matrix is then called a square root of P. Note that the definition of the square root of P is *not* that $P = S^2$, but that $P = SS^T$. Also note that this definition of the matrix square root is not standard. Some books and papers defined the matrix square root as $P = S^2$, others define it as $P = S^T S$, and others define it as $P = SS^T$. This latter definition is the one that we will use in this book. If P is symmetric positive definite then it always has a square root [Gol89, Moo00]. The square root of a matrix may not be unique; that is, there may be more than one solution for S in the equation $P = SS^T$. (This is analogous to the scalar square root, which is usually not unique. For example, the number 1 has two square roots; $+1$ and -1.) Also note that SS^T will always be symmetric positive semidefinite no matter what the value of the S matrix. Whereas numerical difficulties might cause P to become nonsymmetric or indefinite in the Kalman filter equations, numerical difficulties can never cause SS^T to become nonsymmetric or indefinite.

Matrix square root algorithms were first given by the French military officer Andre Cholesky (1875-1918) and the Polish astronomer Tadeusz Banachiewicz (1882-1954) [Fad59]. An interesting biography of Cholesky is given in the appendix of [Mai84].

The following algorithm computes an S matrix such that $P = SS^T$ for an $n \times n$ matrix P.

The Cholesky Matrix Square Root Algorithm {

For $i = 1, \cdots, n$

{

$S_{ii} = \sqrt{P_{ii} - \sum_{j=1}^{i-1} S_{ij}^2}$

For $j = 1, \cdots, n$

{

$S_{ji} = 0 \quad j < i$

$S_{ji} = \frac{1}{S_{ii}} \left(P_{ji} - \sum_{k=1}^{i-1} S_{jk} S_{ik} \right) \quad j > i$

}

}

}

This is called Cholesky factorization and results in a matrix S such that $P = SS^T$. The matrix S is referred to as the Cholesky triangle because it is a lower triangular matrix. However, the algorithm only works if P is symmetric positive definite. If P is not symmetric positive definite, then it may or may not have a square root.[2]

In the following example we illustrate the application of Cholesky factorization.

[2]The MATLAB function CHOL outputs the transpose of the Cholesky triangle that is computed above.

EXAMPLE 6.3

This example is taken from [Kam71]. Suppose we have a P matrix given as

$$P = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 8 & 2 \\ 3 & 2 & 14 \end{bmatrix} \tag{6.41}$$

The Cholesky factorization algorithm tells us that, for $i = 1$,

$$\begin{aligned} S_{11} &= \sqrt{P_{11}} \\ &= 1 \\ S_{21} &= \frac{1}{S_{11}}(P_{21}) \\ &= 2 \\ S_{31} &= \frac{1}{S_{11}}(P_{31}) \\ &= 3 \end{aligned} \tag{6.42}$$

For $i = 2$, the algorithm tells us that

$$\begin{aligned} S_{22} &= \sqrt{P_{22} - \sum_{j=1}^{1} S_{2j}^2} \\ &= 2 \\ S_{12} &= 0 \\ S_{32} &= \frac{1}{S_{22}}\left(P_{32} - \sum_{k=1}^{1} S_{3k}S_{2k}\right) \\ &= -2 \end{aligned} \tag{6.43}$$

For $i = 3$, the algorithm tells us that

$$\begin{aligned} S_{33} &= \sqrt{P_{33} - \sum_{j=1}^{2} S_{3j}^2} \\ &= 1 \\ S_{13} &= 0 \\ S_{23} &= 0 \end{aligned} \tag{6.44}$$

So we obtain

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 2 & 0 \\ 3 & -2 & 1 \end{bmatrix} \tag{6.45}$$

and it can be verified that $P = SS^T$.

$\nabla\nabla\nabla$

After defining S as the square root of P in the Kalman filter, we will propagate S instead of P. This requires more computational effort but it doubles the precision

of the filter and helps prevent numerical problems. The singular values σ of P are given as

$$\begin{aligned} \sigma^2(P) &= \lambda(P^T P) \\ &= \lambda(SS^T SS^T) \end{aligned} \tag{6.46}$$

The singular values of S are given as

$$\sigma^2(S) = \lambda(SS^T) \tag{6.47}$$

Recall that for a general matrix A we have $\lambda(A^2) = \lambda^2(A)$. Therefore, we see from the above equations that

$$\begin{aligned} \sigma^2(P) &= \left[\sigma^2(S)\right]^2 \\ \frac{\sigma_{\max}(P)}{\sigma_{\min}(P)} &= \frac{\sigma^2_{\max}(S)}{\sigma^2_{\min}(S)} \\ \kappa(P) &= \kappa^2(S) \end{aligned} \tag{6.48}$$

That is, the condition number of P is the square of the condition number of S. For example, consider the P matrix given earlier in this section:

$$\begin{aligned} P &= \begin{bmatrix} 10^6 & 0 \\ 0 & 10^{-6} \end{bmatrix} \\ \kappa(P) &= 10^{12} \end{aligned} \tag{6.49}$$

The square root of this matrix and its condition number are

$$\begin{aligned} S &= \begin{bmatrix} 10^3 & 0 \\ 0 & 10^{-3} \end{bmatrix} \\ \kappa(S) &= 10^6 \end{aligned} \tag{6.50}$$

The condition number of P is 10^{12}, but the condition number of the square root of P is only 10^6. Square root filtering uses this idea to provide twice the precision of the standard Kalman filter. Instead of propagating P, we propagate the square root of P.

6.3.2 The square root time-update equation

Suppose we have an n-state discrete LTI system given as

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ E(w_k w_k^T) &= Q_k \end{aligned} \tag{6.51}$$

The *a priori* error covariance matrix of the Kalman filter is P_k^-, and its square root is S_k^-. The *a posteriori* error covariance matrix is P_k^+, and its square root is S_k^+. Suppose that we can find an orthogonal $2n \times 2n$ matrix T such that

$$\begin{aligned} \begin{bmatrix} (S_k^-)^T \\ 0 \end{bmatrix} &= T \begin{bmatrix} (S_{k-1}^+)^T F_{k-1}^T \\ Q_{k-1}^{T/2} \end{bmatrix} \\ &= \begin{bmatrix} T_1 & T_2 \end{bmatrix} \begin{bmatrix} (S_{k-1}^+)^T F_{k-1}^T \\ Q_{k-1}^{T/2} \end{bmatrix} \end{aligned} \tag{6.52}$$

Since T is orthogonal we see that

$$\begin{aligned} T^T T &= \begin{bmatrix} T_1^T \\ T_2^T \end{bmatrix} \begin{bmatrix} T_1 & T_2 \end{bmatrix} \\ &= \begin{bmatrix} T_1^T T_1 & T_1^T T_2 \\ T_2^T T_1 & T_2^T T_2 \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \end{aligned} \tag{6.53}$$

where T_1 and T_2 are both $n \times n$ matrices. We see from the above that

$$\begin{aligned} T_1^T T_2 = T_2^T T_1 &= 0 \\ T_1^T T_1 = T_2^T T_2 &= I \end{aligned} \tag{6.54}$$

Now note that we can use Equation (6.52) to write

$$\begin{bmatrix} S_k^- & 0 \end{bmatrix} \begin{bmatrix} (S_k^-)^T \\ 0 \end{bmatrix} = \left[T_1 (S_{k-1}^+)^T F_{k-1}^T + T_2 Q_{k-1}^{T/2} \right]^T \left[\cdots \right] \tag{6.55}$$

We can use this equation, along with Equation (6.54), to write

$$\begin{aligned} S_k^- (S_k^-)^T &= F_{k-1} S_{k-1}^+ T_1^T T_1 (S_{k-1}^+)^T F_{k-1}^T + Q_{k-1}^{1/2} T_2^T T_2 Q_{k-1}^{T/2} \\ &= F_{k-1} S_{k-1}^+ (S_{k-1}^+)^T F_{k-1}^T + Q_{k-1}^{1/2} Q_{k-1}^{T/2} \end{aligned} \tag{6.56}$$

If S_{k-1}^+ is the square root of P_{k-1}^+, this implies that

$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q_{k-1} \tag{6.57}$$

which is exactly the time-update equation for P_k that is required in the Kalman filter, as shown in Equation (5.19). So if we can find an orthogonal $2n \times 2n$ matrix T such that

$$T \begin{bmatrix} (S_{k-1}^+)^T F_{k-1}^T \\ Q_{k-1}^{T/2} \end{bmatrix} = \begin{bmatrix} n \times n \text{ matrix} \\ 0 \end{bmatrix} \tag{6.58}$$

then the $n \times n$ matrix in the upper half of the matrix on the right side is equal to $(S_k^-)^T$. This assumes that $(S_{k-1}^+)^T$ is available from a square root measurement update equation, which we will discuss in the following two subsections. The square root time update equation above is mathematically equivalent to the original Kalman filter time update equation for P, but the update equation is used to update S instead of P.

As we noted above, the square root of P_k^- is not unique, so different algorithms for solving Equation (6.58) will result in different T and $(S_k^-)^T$ matrices. We can use various methods from numerical linear algebra to find the orthogonal $2n \times 2n$ matrix T and the resulting square root matrix S_k^- (e.g., Householder, Gram–Schmidt, modified Gram–Schmidt, or Givens transformations) [Hor85, Gol89, Str90, Moo00]. A couple of these algorithms are discussed in Section 6.3.5.

■ EXAMPLE 6.4

Suppose that at time $(k-1)$ our Kalman filter has a system matrix, process noise covariance, and *a posteriori* estimation covariance square root equal to

$$\begin{aligned} F_{k-1} &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \\ Q_{k-1} &= \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} \\ S_{k-1}^{+} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{6.59}$$

It can be verified that the square root of Q_{k-1} (so that $Q_{k-1}^{1/2} Q_{k-1}^{T/2} = Q_{k-1}$) is given by

$$Q_{k-1}^{1/2} = \begin{bmatrix} 0 & 0 \\ -1 & -1 \end{bmatrix} \tag{6.60}$$

Equation (6.58) can be solved as

$$T \begin{bmatrix} (S_{k-1}^{+})^T F_{k-1}^T \\ Q_{k-1}^{T/2} \end{bmatrix} = \begin{bmatrix} (S_k^{-})^T \\ 0 \end{bmatrix}$$

$$\frac{1}{\sqrt{10}} \begin{bmatrix} \sqrt{5} & \sqrt{5} & 0 & 0 \\ 1 & -1 & 2 & 2 \\ -2 & -2 & 1 & 1 \\ 0 & 0 & -\sqrt{5} & \sqrt{5} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & -1 \\ 0 & -1 \end{bmatrix} = \frac{1}{\sqrt{10}} \begin{bmatrix} \sqrt{20} & \sqrt{5} \\ 0 & -5 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{6.61}$$

As mentioned earlier, algorithms for performing this computation will be discussed in Section 6.3.5. The upper-right square matrix on the right side of the above equation is equal to $(S_k^{-})^T$, so this shows that the square root of the *a priori* estimation covariance at time k is given as

$$S_k^{-} = \frac{1}{\sqrt{10}} \begin{bmatrix} \sqrt{20} & 0 \\ \sqrt{5} & -5 \end{bmatrix} \tag{6.62}$$

From this it can be inferred that the *a priori* estimation covariance at time k is given as

$$\begin{aligned} P_k^{-} &= S_k^{-} (S_k^{-})^T \\ &= \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \end{aligned} \tag{6.63}$$

Indeed, a straightforward implementation of the time-update equation for the estimation-error covariance gives

$$\begin{aligned} P_k^{-} &= F_{k-1} P_{k-1}^{+} F_{k-1}^T + Q_{k-1} \\ &= \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \end{aligned} \tag{6.64}$$

which confirms our square root results. However, the square root time update has essentially twice the precision of the standard time-update equation.

∇∇∇

6.3.3 Potter's square root measurement-update equation

The square root measurement-update equation discussed here is based on James Potter's algorithm, which was developed for NASA's Apollo space program [Bat64, Kam71] and modified by Angus Andrews to handle vector measurements [And68]. Recall from Equation (5.19) that the measurement update equation for the estimation covariance is given as

$$P_k^+ = (I - K_k H_k) P_k^- \tag{6.65}$$

We can process the measurements one at a time using the sequential Kalman filter of Section 6.1. That is, first we initialize $P_{0k}^+ = P_k^-$. Then, for $i = 1, \cdots, r$ (where r is the number of measurements), we compute

$$\begin{aligned} K_{ik} &= \frac{P_{i-1,k}^+ H_{ik}^T}{H_{ik} P_{i-1,k}^+ H_{ik}^T + R_{ik}} \\ P_{ik}^+ &= (I - K_{ik} H_{ik}) P_{i-1,k}^+ \end{aligned} \tag{6.66}$$

where H_{ik} is the ith row of H_k and R_{ik} is the variance of the ith measurement. (We are assuming here, as in Section 6.1, that R_k is diagonal.) Suppose we have the square root of $P_{i-1,k}^+$ so that $P_{i-1,k}^+ = S_{i-1,k}^+ S_{i-1,k}^{+T}$. Then K_{ik} can be written as

$$K_{ik} = \frac{S_{i-1,k}^+ S_{i-1,k}^{+T} H_{ik}^T}{H_{ik} S_{i-1,k}^+ S_{i-1,k}^{+T} H_{ik}^T + R_{ik}} \tag{6.67}$$

and P_{ik}^+ can be written as

$$\begin{aligned} P_{ik}^+ &= \left(I - \frac{S_{i-1,k}^+ S_{i-1,k}^{+T} H_{ik}^T H_{ik}}{H_{ik} S_{i-1,k}^+ S_{i-1,k}^{+T} H_{ik}^T + R_{ik}} \right) S_{i-1,k}^+ S_{i-1,k}^{+T} \\ &= S_{i-1,k}^+ (I - a\phi\phi^T) S_{i-1,k}^{+T} \end{aligned} \tag{6.68}$$

where ϕ and a are defined as

$$\begin{aligned} \phi &= S_{i-1,k}^{+T} H_{ik}^T \\ a &= \frac{1}{\phi^T \phi + R_{ik}} \end{aligned} \tag{6.69}$$

It can be shown (see Problem 6.9) that

$$I - a\phi\phi^T = (I - a\gamma\phi\phi^T)^2 \tag{6.70}$$

where γ is given as

$$\gamma = \frac{1}{1 \pm \sqrt{aR_{ik}}} \tag{6.71}$$

Either the plus or minus sign can be used in the computation of γ. Comparing Equations (6.68) and (6.70) shows that

$$S_{ik}^+ = S_{i-1,k}^+ (I - a\gamma\phi\phi^T) \tag{6.72}$$

This results in a square root measurement-update algorithm that can be summarized as follows.

Potter's square root measurement-update algorithm

1. After the *a priori* covariance square root S_k^- and the *a priori* state estimate $\hat{x}_k^-$ have been computed, initialize

$$\begin{aligned} \hat{x}_{0k}^+ &= \hat{x}_k^- \\ S_{0k}^+ &= S_k^- \end{aligned} \tag{6.73}$$

2. For $i = 1, \cdots, r$ (where r is the number of measurements), perform the following.

   (a) Define H_{ik} as the ith row of H_k, y_{ik} as the ith element of y_k, and R_{ik} as the variance of the ith measurement (assuming that R_k is diagonal).

   (b) Perform the following to find the square root of the covariance after the ith measurement has been processed:

$$\begin{aligned} \phi_i &= S_{i-1,k}^{+T} H_{ik}^T \\ a_i &= \frac{1}{\phi_i^T \phi_i + R_{ik}} \\ \gamma_i &= \frac{1}{1 \pm \sqrt{a_i R_{ik}}} \\ S_{ik}^+ &= S_{i-1,k}^+ (I - a_i \gamma_i \phi_i \phi_i^T) \end{aligned} \tag{6.74}$$

   (c) Compute the Kalman gain for the ith measurement as

$$K_{ik} = a_i S_{ik}^+ \phi_i \tag{6.75}$$

   (d) Compute the state estimate update due to the ith measurement as

$$\hat{x}_{ik}^+ = \hat{x}_{i-1,k}^+ + K_{ik}(y_{ik} - H_{ik}\hat{x}_{i-1,k}^+) \tag{6.76}$$

3. Set the *a posteriori* covariance square root and the *a posteriori* state estimate as

$$\begin{aligned} S_k^+ &= S_{rk}^+ \\ \hat{x}_k^+ &= \hat{x}_{rk}^+ \end{aligned} \tag{6.77}$$

Although square root filtering improves the numerical characteristics of the Kalman filter, it also increases computational requirements. Efforts to make square root filtering more efficient are reported in [Car73, Tho77, Tap80].

■ EXAMPLE 6.5

This example is based on [Kam71]. Suppose that we have an LTI system with

$$\begin{aligned} P_k^- &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ H &= \begin{bmatrix} 1 & 0 \end{bmatrix} \\ F &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ Q &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned} \tag{6.78}$$

If we had an infinite-precision computer, the exact Kalman gain and *a posteriori* covariance at time k would be given by

$$\begin{aligned} K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\ &= \begin{bmatrix} \frac{1}{1+R} \\ 0 \end{bmatrix} \\ P_k^+ &= (I - K_k H) P_k^- \\ &= \begin{bmatrix} \frac{R}{1+R} & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{6.79}$$

The *a priori* covariance and Kalman gain at the next time step $(k+1)$ would be given by

$$\begin{aligned} P_{k+1}^- &= F P_k^+ F^T + Q \\ &= \begin{bmatrix} \frac{R}{1+R} & 0 \\ 0 & 1 \end{bmatrix} \\ K_{k+1} &= P_{k+1}^- H^T (H P_{k+1}^- H^T + R)^{-1} \\ &= \begin{bmatrix} \frac{1}{2+R} \\ 0 \end{bmatrix} \end{aligned} \tag{6.80}$$

Now consider implementation in a finite precision digital computer. Suppose that the measurement covariance $R \ll 1$. The covariance R is such a tiny number that because of rounding in the computer, $1+R = 1$, but $1+\sqrt{R} > 1$. The rounded values of the Kalman gain and *a posteriori* covariance at time k would be given by

$$\begin{aligned} K_k &= \begin{bmatrix} \frac{1}{1+R} \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ P_k^+ &= (I - K_k H) P_k^- \\ &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{6.81}$$

Note that P_k^+ has become singular because of the numerical limitations of the computer. The rounded values of the *a priori* covariance and Kalman gain at the next time step $(k+1)$ would be given by

$$\begin{aligned} P_{k+1}^- &= F P_k^+ F^T + Q \\ &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\ K_{k+1} &= P_{k+1}^- H^T (H P_{k+1}^- H^T + R)^{-1} \\ &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{aligned} \tag{6.82}$$

The numerical limitations of the computer have resulted in a zero Kalman gain, whereas the infinite-precision Kalman gain as given in Equation (6.80) is about $\begin{bmatrix} 1/2 & 0 \end{bmatrix}^T$.

Now suppose we implement the measurement-update equation using Potter's algorithm. We start out with

$$S_k^- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.83}$$

We only have to iterate through Equation (6.74) one time since we only have one measurement. The rounded values of the parameters given in Equation (6.74) are

$$\begin{aligned}
\phi &= (S_k^-)^T H^T \\
&= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
a &= \frac{1}{\phi^T \phi + R} \\
&= \frac{1}{1+R} \\
&= 1 \\
\gamma &= \frac{1}{1+\sqrt{aR}} \\
&= \frac{1}{1+\sqrt{R}} \\
S_k^+ &= S_k^-(I - a\gamma\phi\phi^T) \\
&= \begin{bmatrix} \frac{\sqrt{R}}{1+\sqrt{R}} & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned} \tag{6.84}$$

Note that $S_k^+ S_k^{+T}$ is nonsingular. The rounded values of the square root of the *a priori* covariance, the parameters of Equation (6.74), and the Kalman gain at the next time step $(k+1)$ would be given by

$$\begin{aligned}
S_{k+1}^- &= S_k^+ \\
\phi &= (S_{k+1}^-)^T H^T \\
&= \begin{bmatrix} \frac{\sqrt{R}}{1+\sqrt{R}} \\ 0 \end{bmatrix} \\
a &= \frac{1}{\phi^T \phi + R} \\
&= \frac{1 + R + 2\sqrt{R}}{R^2 + 2R + 2R\sqrt{R}} \\
&= \frac{1 + 2\sqrt{R}}{2R + 2R\sqrt{R}} \\
K_{k+1} &= a S_{k+1}^- \phi \\
&= \frac{1 + 2\sqrt{R}}{2R(1+\sqrt{R})} \begin{bmatrix} \frac{R}{1+R+2\sqrt{R}} \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{2(1+\sqrt{R})} \\ 0 \end{bmatrix}
\end{aligned} \tag{6.85}$$

Note that the rounded Kalman gain is almost identical to the exact Kalman gain given by Equation (6.80). This shows the benefit that can be gained by using the square root filter.

$\nabla\nabla\nabla$

6.3.4 Square root measurement update via triangularization

The previous section derived a measurement update based on Potter's algorithm that could be performed on the square root of the Kalman filter estimation covariance. This section derives an alternative method for performing the measurement update. Suppose that we want to design a Kalman filter for a system with n states and r measurements. Suppose that we can find an orthogonal matrix $(n+r)\times(n+r)$ matrix $\tilde{T}$ such that

$$\begin{bmatrix} (R_k + H_k P_k^- H_k^T)^{T/2} & \tilde{K}_k^T \\ 0 & (S_k^+)^T \end{bmatrix} = \tilde{T} \begin{bmatrix} R_k^{T/2} & 0 \\ (S_k^-)^T H_k^T & (S_k^-)^T \end{bmatrix} \tag{6.86}$$

S_k^- and S_k^+ are the square roots of the *a priori* and *a posteriori* covariances, and $\tilde{K}_k$ is defined as

$$\tilde{K}_k = K_k (R_k + H_k P_k^- H_k^T)^{T/2} \tag{6.87}$$

where K_k is the normal Kalman gain matrix. Note that S_k^+ in Equation (6.86) is not known until after an orthogonal $\tilde{T}$ is found that forces the left side of Equation (6.86) into the specified form. That is, we need to find a $\tilde{T}$ so that the upper-left $r\times r$ block of the left side of Equation (6.86) is equal to $(R_k + H_k P_k^- H_k^T)^{T/2}$, the upper-right $r \times n$ block is equal to $\tilde{K}_k^T$, and the lower-left $n \times r$ block is equal to 0. After such a $\tilde{T}$ is found, whatever the lower right $n \times n$ block turns out to be is, by definition, equal to $(S_k^+)^T$, which is the transpose of the square root of P_k^+. Now write the $(n+r) \times (n+r)$ matrix $\tilde{T}$ as

$$\tilde{T} = \begin{bmatrix} \tilde{T}_{11} & \tilde{T}_{12} \\ \tilde{T}_{21} & \tilde{T}_{22} \end{bmatrix} \tag{6.88}$$

where $\tilde{T}_{11}$ is an $r \times r$ matrix, $\tilde{T}_{12}$ is an $r \times n$ matrix, $\tilde{T}_{21}$ is an $n \times r$ matrix, and $\tilde{T}_{22}$ is an $n \times n$ matrix. Since $\tilde{T}$ is orthogonal we can write

$$\begin{aligned} \tilde{T}^T\tilde{T} &= \begin{bmatrix} \tilde{T}_{11}^T & \tilde{T}_{21}^T \\ \tilde{T}_{12}^T & \tilde{T}_{22}^T \end{bmatrix} \begin{bmatrix} \tilde{T}_{11} & \tilde{T}_{12} \\ \tilde{T}_{21} & \tilde{T}_{22} \end{bmatrix} \\ &= \begin{bmatrix} \tilde{T}_{11}^T\tilde{T}_{11} + \tilde{T}_{21}^T\tilde{T}_{21} & \tilde{T}_{11}^T\tilde{T}_{12} + \tilde{T}_{21}^T\tilde{T}_{22} \\ \tilde{T}_{12}^T\tilde{T}_{11} + \tilde{T}_{22}^T\tilde{T}_{21} & \tilde{T}_{12}^T\tilde{T}_{12} + \tilde{T}_{22}^T\tilde{T}_{22} \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \end{aligned} \tag{6.89}$$

Now we expand Equation (6.86) as

$$\begin{bmatrix} (R_k + H_k P_k^- H_k^T)^{T/2} & \tilde{K}_k^T \\ 0 & (S_k^+)^T \end{bmatrix} = \begin{bmatrix} \tilde{T}_{11} R_k^{T/2} + \tilde{T}_{12}(S_k^-)^T H_k^T & \tilde{T}_{12}(S_k^-)^T \\ \tilde{T}_{21} R_k^{T/2} + \tilde{T}_{22}(S_k^-)^T H_k^T & \tilde{T}_{22}(S_k^-)^T \end{bmatrix} \tag{6.90}$$

We will equate the four matrix partitions of this equation to write four separate equalities. We will then take each equality and premultiply each side by its transpose to obtain four new equalities. The first two equalities obtained this way are

$$\begin{aligned}
&(R_k + H_k P_k^- H_k^T)^{1/2}(\cdots)^{T/2} = \\
&\quad R_k^{1/2}\tilde{T}_{11}^T\tilde{T}_{11}R_k^{T/2} + H_k S_k^- \tilde{T}_{12}^T\tilde{T}_{11}R_k^{T/2} + \\
&\quad R_k^{1/2}\tilde{T}_{11}^T\tilde{T}_{12}(S_k^-)^T H_k^T + H_k S_k^- \tilde{T}_{12}^T\tilde{T}_{12}(S_k^-)^T H_k^T \\
0 \;=\;& R_k^{1/2}\tilde{T}_{21}^T\tilde{T}_{21}R_k^{T/2} + R_k^{1/2}\tilde{T}_{21}^T\tilde{T}_{22}(S_k^-)^T H_k^T + \\
&\quad H_k S_k^- \tilde{T}_{22}^T\tilde{T}_{21}R_k^{T/2} + H_k S_k^- \tilde{T}_{22}^T\tilde{T}_{22}(S_k^-)^T H_k^T
\end{aligned} \tag{6.91}$$

Adding these two equations and using Equations (6.87) and (6.89) to simplify the result gives

$$R_k + H_k P_k^- H_k^T = R_k + H_k S_k^- (S_k^-)^T H_k^T \tag{6.92}$$

This shows that the proposed measurement update of Equation (6.86) is consistent with S_k^- being the square root of P_k^-.

The second two equalities that can be written from Equation (6.90) are

$$\begin{aligned}
\tilde{K}_k\tilde{K}_k^T &= S_k^- \tilde{T}_{12}^T\tilde{T}_{12}(S_k^-)^T \\
S_k^+(S_k^+)^T &= S_k^- \tilde{T}_{22}^T\tilde{T}_{22}(S_k^-)^T
\end{aligned} \tag{6.93}$$

Adding these two equations and using Equation (6.89) to simplify the result gives

$$S_k^+(S_k^+)^T + K_k(R_k + H_k P_k^- H_k^T)K_k^T = S_k^-(S_k^-)^T \tag{6.94}$$

Substituting the standard Kalman gain equation $K_k = P_k^- H_k^T(R_k + H_k P_k^- H_k^T)^{-1}$ into this equation gives

$$\begin{aligned}
S_k^+(S_k^+)^T + P_k^- H_k^T K_k^T &= P_k^- \\
S_k^+(S_k^+)^T &= P_k^- - P_k^- H_k^T K_k^T
\end{aligned} \tag{6.95}$$

Since the left side of the above equation is symmetric and the first term on the right side is symmetric, the last term on the right side must also be symmetric, which means that we can transpose it in the above equation to obtain

$$S_k^+(S_k^+)^T = P_k^- - K_k H_k P_k^- \tag{6.96}$$

The right side of this equation is the Kalman filter measurement-update equation for P, which means that the left side of the equation must be P_k^+, which means that S_k^+ must be the square root of P_k^+. So if we can find an orthogonal $(n+r)\times(n+r)$ matrix $\tilde{T}$ such that

$$\begin{bmatrix} (R_k + H_k P_k^- H_k^T)^{T/2} & \tilde{K}_k^T \\ 0 & (n \times n \text{ matrix}) \end{bmatrix} = \tilde{T}\begin{bmatrix} R_k^{T/2} & 0 \\ (S_k^-)^T H_k^T & (S_k^-)^T \end{bmatrix} \tag{6.97}$$

then the lower-right $n \times n$ matrix on the left side of the equation is equal to the transpose of the square root of P_k^+, and this equation is mathematically equivalent to the original Kalman filter measurement-update equation for P_k. This measurement-update method results in numerical precision that is effectively twice as much as the standard Kalman filter, which helps to avoid numerical problems. However, the computation of $\tilde{T}$ adds a lot of computational effort to the Kalman filter. In addition, the form of the transformation given in Equation (6.97) makes it of questionable practicality (see Problem 6.10).

6.3.5 Algorithms for orthogonal transformations

Several numerical algorithms are available for performing the orthogonal transformations that are required to solve for the T and S_k^- matrices in Equation (6.58). Some algorithms that can be used are the Householder method, the Givens method, the Gram–Schmidt method, and the modified Gram–Schmidt method. In this section we will present (without derivation) the Householder algorithm and the modified Gram–Schmidt algorithm. Derivations and presentations of the other algorithms can be found in many texts on numerical linear algebra, such as [Hor85, Gol89, Moo00]. A comparison of Gram–Schmidt, modified Gram–Schmidt, and Householder transformations can be found in [Jor68], where it is stated that the modified Gram–Schmidt procedure is best (from a numerical point of view), with the Householder method offering competitive performance.

6.3.5.1 The Householder algorithm The algorithm presented here was developed by Alston Householder [Hou64, Chapter 5], applied to least squares estimation by Gene Golub [Gol65], and summarized for Kalman filtering by Paul Kaminski [Kam71].

1. Suppose that we have a $2n \times n$ matrix $A^{(1)}$, and we want to find an $n \times n$ matrix W such that
$$TA^{(1)} = \begin{bmatrix} W \\ 0 \end{bmatrix} \tag{6.98}$$
where T is an orthogonal $2n \times 2n$ matrix, and 0 is the $n \times n$ matrix consisting of all zeros. Note that this problem statement is in the same form as Equation (6.58). Also note that we do not necessarily need to find T; our goal is to find W.

2. For $k = 1, \cdots, n$ perform the following:

   (a) Compute the scalar σ_k as
$$\sigma_k = \operatorname{sgn}\left(A_{kk}^{(k)}\right)\sqrt{\sum_{i=k}^{2n}\left(A_{ik}^{(k)}\right)^2} \tag{6.99}$$
   where $A_{ik}^{(k)}$ is the element in the ith row and kth column of $A^{(k)}$. The $\operatorname{sgn}(\cdot)$ function is defined to be equal to $+1$ if its argument is greater than or equal to zero, and -1 if its argument is less than zero.

   (b) Compute the scalar β_k as
$$\beta_k = \frac{1}{\sigma_k\left(\sigma_k + A_{kk}^{(k)}\right)} \tag{6.100}$$

   (c) For $i = 1, \cdots, 2n$ perform the following:
$$u_i^{(k)} = \begin{cases} 0 & i < k \\ \sigma_k + A_{kk}^{(k)} & i = k \\ A_{ik}^{(k)} & i > k \end{cases} \tag{6.101}$$

This gives a $2n$-element column vector $u^{(k)}$.

(d) For $i = 1, \cdots, n$ perform the following:

$$y_i^{(k)} = \begin{cases} 0 & i < k \\ 1 & i = k \\ \beta_k u^{(k)T} A_i^{(k)} & i > k \end{cases} \tag{6.102}$$

where $A_i^{(k)}$ is the ith column of $A^{(k)}$. This gives an n-element column vector $y^{(k)}$.

(e) Compute the $2n \times n$ matrix $A^{(k+1)}$ as

$$A^{(k+1)} = A^{(k)} - u^{(k)} y^{(k)T} \tag{6.103}$$

3. After the above steps have been executed, $A^{(n+1)}$ has the form

$$A^{(n+1)} = \begin{bmatrix} W \\ 0 \end{bmatrix} \tag{6.104}$$

where W is the $n \times n$ matrix that we are trying to solve for. Note that if $\sigma_k = 0$ at any stage of the algorithm, that means $A^{(1)}$ is rank deficient and the algorithm will fail. Also note that the above algorithm does not compute the T matrix. However, we can find the T matrix as

$$\begin{aligned} T &= T^{(n)} T^{(n-1)} \cdots T^{(1)} \\ T^{(k)} &= I - \beta_k u^{(k)} u^{(k)T} \quad i = 1, \cdots, n \end{aligned} \tag{6.105}$$

6.3.5.2 The modified Gram–Schmidt algorithm The modified Gram–Schmidt algorithm for orthonormalization that is presented here is discussed in most linear systems books [Kai80, Bay99, Che99]. It was first given in [Bjo67] and was summarized for Kalman filtering in [Kam71].

1. Suppose that we have a $2n \times n$ matrix $A^{(1)}$, and we want to find an $n \times n$ matrix W such that

$$TA^{(1)} = \begin{bmatrix} W \\ 0 \end{bmatrix} \tag{6.106}$$

where T is an orthogonal $2n \times 2n$ matrix, and 0 is the $n \times n$ matrix consisting of all zeros. Note that this problem statement is in the same form as Equation (6.58).

2. For $k = 1, \cdots, n$ perform the following.

(a) Compute the scalar σ_k as

$$\sigma_k = \sqrt{A_k^{(k)T} A_k^{(k)}} \tag{6.107}$$

where $A_i^{(k)}$ is the ith column of $A^{(k)}$.

(b) Compute the kth row of W as

$$W_{kj} = \begin{cases} 0 & j = 1, \cdots, k-1 \\ \sigma_k & j = k \\ A_k^{(k)T} A_j^{(k)} / \sigma_k & j = k+1, \cdots, n \end{cases} \tag{6.108}$$

(c) Compute the kth row of T as

$$T_k = A_k^{(k)T}/\sigma_k \tag{6.109}$$

(d) If $(k < n)$, compute the last $(n-k)$ columns of $A^{(k+1)}$ as

$$A_j^{(k+1)} = A_j^{(k)} - W_{kj}A_k^{(k)}/\sigma_k \qquad j = k+1, \cdots, n \tag{6.110}$$

Note that the first k columns of $A^{(k+1)}$ are not computed in this algorithm.

As with the Householder algorithm, if $\sigma_k = 0$ at any stage of the algorithm, that means $A^{(1)}$ is rank deficient and the algorithm fails. After this algorithm completes, we have the first n rows of T, and T is an $n \times 2n$ matrix. If we want to know the last n rows of T, we can compute them using a regular Gram–Schmidt algorithm as follows [Hor85, Gol89, Moo00].

1. Fill out the T matrix that was begun above by appending a $2n \times 2n$ identity matrix to the bottom of it. This ensures that the rows of T span the entire $2n$-dimensional vector space:

$$T = \begin{bmatrix} T \\ I \end{bmatrix} \tag{6.111}$$

Note that this T is a $3n \times 2n$ matrix.

2. Now we perform a standard Gram–Schmidt orthonormalization procedure on the last $2n$ rows of T (with respect to the already obtained first n rows of T). For $k = n+1, \cdots, 3n$, compute the kth row of T as

$$\begin{aligned} T_k &= T_k - \sum_{i=1}^{k-1}(T_k T_i^T)T_i \\ T_k &= \frac{T_k}{||T_k||_2} \end{aligned} \tag{6.112}$$

If T_k is zero then that means that it is a linear combination of the previous rows of T. In that case, the division in the above equation will be a divide by zero, so instead T_k should be discarded. This discard will actually occur exactly n times so that this procedure will compute n additional rows of T and we will end up with an orthogonal $2n \times 2n$ matrix T.

The Gram–Schmidt algorithms are named after the Danish mathematician Jorgen Gram (1850-1916) and the German mathematician Erhard Schmidt (1876-1959). Schmidt received his doctorate in 1905 under David Hilbert's supervision, and in 1929 he was on the doctoral committee of Eberhard Hopf (see Section 3.4.4). However, the Gram–Schmidt algorithm was actually invented by Pierre Laplace (1749-1827).

6.4 U-D FILTERING

U-D filtering was introduced in [Bie76, Bie77a] as another way to increase the numerical precision of the Kalman filter. It is sometimes considered as a type of square root filtering, and sometimes it is considered distinct from square root filtering (depending on the author). It increases the computational cost of the filter but not so severely as the square root filter of the previous section.

The idea of U-D filtering is to factor the $n \times n$ matrix P as UDU^T, where U is an $n \times n$ upper triangular matrix with ones along the diagonal, and D is an $n \times n$ diagonal matrix. This can always be accomplished for a symmetric positive definite matrix P [Gol89, Chapter 4], so it can always be implemented on a Kalman filter. A U-D factorization routine can be implemented without too much difficulty. For example, suppose that we want to compute the U-D factorization of a 3×3 matrix. We can then write

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{12} & p_{22} & p_{23} \\ p_{13} & p_{23} & p_{33} \end{bmatrix} = \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ u_{12} & 1 & 0 \\ u_{13} & u_{23} & 1 \end{bmatrix}$$
$$= \begin{bmatrix} d_{11} + d_{22}u_{12}^2 + d_{33}u_{13}^2 & d_{22}u_{12} + d_{33}u_{13}u_{23} & d_{33}u_{13} \\ d_{22}u_{12} + d_{33}u_{13}u_{23} & d_{22} + d_{33}u_{23}^2 & d_{33}u_{23} \\ d_{33}u_{13} & d_{33}u_{23} & d_{33} \end{bmatrix} \tag{6.113}$$

We need to solve for the u_{ij} and d_{ii} elements. We can begin at the lower-right element of the matrix equality to see that $d_{33} = p_{33}$. Next we can look at the other elements in the third column to see that

$$\begin{aligned} u_{13} &= p_{13}/d_{33} \\ u_{23} &= p_{23}/d_{33} \end{aligned} \tag{6.114}$$

Now look at the $(2,2)$ and $(1,2)$ elements of the equality to see that

$$\begin{aligned} d_{22} &= p_{22} - d_{33}u_{23}^2 \\ u_{12} &= (p_{12} - d_{33}u_{13}u_{23})/d_{22} \end{aligned} \tag{6.115}$$

Finally look at the $(1,1)$ element of the equality to see that

$$d_{11} = p_{11} - d_{22}u_{12}^2 - d_{33}u_{13}^2 \tag{6.116}$$

This gives us the U-D factorization for a 3×3 symmetric matrix, and provides the outline for a general U-D factorization algorithm.

6.4.1 U-D filtering: The measurement-update equation

Recall from Equation (5.19) the measurement update equation for the covariance of the Kalman filter:

$$P^+ = P^- - P^- H^T (HP^- H^T + R)^{-1} HP^- \tag{6.117}$$

We have omitted the time subscripts for ease of notation. Now suppose that we process the measurements sequentially as discussed in Section 6.1. This gives the equation

$$P_i = P_{i-1} - P_{i-1} H_i^T (H_i P_{i-1} H_i^T + R_i)^{-1} H_i P_{i-1} \tag{6.118}$$

where H_i is the ith row of H, R_i is the ith diagonal entry of R, and P_i is the estimation covariance after i measurements have been processed. Now define the scalar $\alpha_i \equiv H_i P_{i-1} H_i^T + R_i$. Suppose that $P_{i-1} = U_{i-1} D_{i-1} U_{i-1}^T$, and $P_i = U_i D_i U_i^T$. With these factorizations we can write the measurement update of Equation (6.118) as

$$\begin{aligned} U_i D_i U_i^T &= U_{i-1} D_{i-1} U_{i-1}^T - \frac{1}{\alpha_i} U_{i-1} D_{i-1} U_{i-1}^T H_i^T H_i U_{i-1} D_{i-1} U_{i-1}^T \\ &= U_{i-1} \left[D_{i-1} - \frac{1}{\alpha_i} (D_{i-1} U_{i-1}^T H_i^T)(D_{i-1} U_{i-1}^T H_i^T)^T \right] U_{i-1}^T \end{aligned} \tag{6.119}$$

The term in brackets in the above equation is symmetric positive definite so it has a U-D factorization that can be written as

$$\bar{U} \bar{D} \bar{U}^T = \left[D_{i-1} - \frac{1}{\alpha_i} (D_{i-1} U_{i-1}^T H_i^T)(D_{i-1} U_{i-1}^T H_i^T)^T \right] \tag{6.120}$$

Combining this with Equation (6.119) gives

$$\begin{aligned} U_i D_i U_i^T &= U_{i-1} \bar{U} \bar{D} \bar{U}^T U_{i-1}^T \\ &= (U_{i-1} \bar{U}) \bar{D} (U_{i-1} \bar{U})^T \end{aligned} \tag{6.121}$$

Note that $U_{i-1}\bar{U}$ is upper triangular with diagonal elements equal to 1, and $\bar{D}$ is diagonal. Therefore the above equation means that $U_i = U_{i-1}\bar{U}$, and $D_i = \bar{D}$:

$$\begin{aligned} U_i &= U_{i-1} \bar{U} \\ D_i &= \bar{D} \end{aligned} \tag{6.122}$$

This gives us a way of performing the measurement update of P in terms of its U-D factors. The algorithm can be summarized as follows.

The U-D measurement update

1. We start with the *a priori* estimation covariance P^- at time k. Define $P_0 = P^-$.
2. For $i = 1, \cdots, r$ (where r is the number of measurements), perform the following:
   (a) Define H_i as the ith row of H, R_i as the ith diagonal entry of R, and $\alpha_i = H_i P_{i-1} H_i^T + R_i$.
   (b) Perform a U-D factorization of P_{i-1} to obtain U_{i-1} and D_{i-1}, and then form the matrix on the right side of Equation (6.120).
   (c) Find the U-D factorization of the matrix on the right side of Equation (6.120) and call the factors $\bar{U}$ and $\bar{D}$.
   (d) Compute U_i and D_i from Equation (6.122).
3. The *a posteriori* estimation covariance is given as $P^+ = U_r D_r U_r^T$.

Since the U-D measurement-update equation relies on sequential filtering, the conditions discussed at the end of Section 6.1 apply to U-D filtering. That is, it probably does not make sense to implement U-D filtering unless one of the following two conditions is true.

1. The measurement noise covariance R_k is diagonal
2. The measurement noise covariance R is a constant.

6.4.2 U-D filtering: The time-update equation

Recall from Equation (5.19) the time-update equation for the covariance of the Kalman filter:

$$P^- = FP^+F^T + Q \tag{6.123}$$

We have omitted the time subscripts for ease of notation. If the Kalman filter is being used to estimate the state of an n-state system, then the P matrices will be $n \times n$ matrices. Suppose that P^+ is factored as $U^+D^+U^{+T}$ (from the measurement update equation discussed previously). We need to find the U-D factors of P^- such that $P^- = U^-D^-U^{-T} = FP^+F^T + Q$. Note that U^{-T} in this notation is *not* the transpose of the inverse of U; it is rather the transpose of U^-. The time update of Equation (6.123) can be written as

$$\begin{aligned} P^- &= FP^+F^T + Q \\ &= \begin{bmatrix} FU^+ & I \end{bmatrix} \begin{bmatrix} D^+ & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} U^{+T}F^T \\ I \end{bmatrix} \\ &= W\hat{D}W^T \end{aligned} \tag{6.124}$$

where W and $\hat{D}$ are defined by the above equation. Note that W is an $n \times 2n$ matrix, and $\hat{D}$ is a $2n \times 2n$ matrix. From the above equation we see that the U-D factors of P^- need to satisfy

$$U^-D^-U^{-T} = W\hat{D}W^T \tag{6.125}$$

The transpose of W can be written as

$$W^T = \begin{bmatrix} w_1^T & \cdots & w_n^T \end{bmatrix} \tag{6.126}$$

That is, w_i (a $2n$-element row vector) is the ith row of W. Now we find n vectors v_i such that

$$v_k\hat{D}v_j^T = 0 \quad k \neq j \tag{6.127}$$

The v_i vectors ($2n$-element row vectors) can be found with the following Gram–Schmidt orthogonalization procedure [Hor85, Gol89, Moo00]:

$$\begin{aligned} v_n &= w_n \\ v_k &= w_k - \sum_{j=k+1}^{n} \frac{w_k\hat{D}v_j^T}{v_j\hat{D}v_j^T}v_j \qquad k = n-1, \cdots, 1 \end{aligned} \tag{6.128}$$

If we define $u(k,j)$ as

$$u(k,j) = \frac{w_k\hat{D}v_j^T}{v_j\hat{D}v_j^T} \qquad j,k = 1, \cdots, n \tag{6.129}$$

then from Equation (6.128) we see that w_k can be expressed as

$$w_k = v_k + \sum_{j=k+1}^{n} u(k,j)v_j \qquad k = 1, \cdots, n \tag{6.130}$$

or equivalently

$$w_k^T = v_k^T + \sum_{j=k+1}^{n} u(k,j)v_j^T \qquad k = 1, \cdots, n \tag{6.131}$$

These n equations can be written as

$$\begin{aligned} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} &= \begin{bmatrix} 1 & u(1,2) & \cdots & u(1,n) \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & u(n-1,n) \\ 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \\ W &= U^- V \end{aligned} \tag{6.132}$$

The $n \times 2n$ matrix W, the $n \times n$ matrix U^-, and the $n \times 2n$ matrix V are defined by the above equation. Note that U^- is a unit upper triangular matrix. The matrix product $W\hat{D}W^T$ can then be written as

$$\begin{aligned} W\hat{D}W^T &= (U^- V)\hat{D}(U^- V)^T \\ &= U^-(V\hat{D}V^T)U^{-T} \\ &= U^- D^- U^{-T} \end{aligned} \tag{6.133}$$

where the D^- matrix is defined by the above equation. From Equation (6.127), we see that the v_i vectors are orthogonal with respect to the $\hat{D}$ inner product. We therefore know that

$$\begin{aligned} D^- &= V\hat{D}V^T = \mathrm{diag}(d_1, \cdots, d_n) \\ d_k &= v_k \hat{D} v_k^T \end{aligned} \tag{6.134}$$

That is, D^- is a diagonal matrix. From Equations (6.124), (6.125), and (6.133) we see that U^- and D^- satisfy the conditions of being the U-D factors of P^-. This gives us a way to perform the Kalman filter time-update equation in U-D factorization form. The algorithm can be summarized as follows.

The U-D time update

1. Begin with $P^+ = U^+ D^+ U^{+T}$ (from the measurement update equation).
2. Define the following matrices.

$$\begin{aligned} W &= \begin{bmatrix} FU^+ & I \end{bmatrix} \\ \hat{D} &= \begin{bmatrix} D^+ & 0 \\ 0 & Q \end{bmatrix} \end{aligned} \tag{6.135}$$

3. Use the rows of W along with the Gram–Schmidt orthogonalization procedure to generate v_i vectors that are orthogonal with respect to the $\hat{D}$ inner product. The algorithm for generating the v_i vectors is given in Equation (6.128).

4. Form the V matrix using the v_i vectors as rows; see Equation (6.132).
5. Use $\hat{D}$ inner products to form the unit upper triangular matrix U^-; see Equations (6.129) and (6.132).
6. Define D^- as $D^- = V\hat{D}V^T$.

The U-D filter results in twice as much precision as the standard Kalman filter, just like the square root filter, but it requires less computation than the square root filter. If some of the states are missing from the measurement vector, a more efficient U-D algorithm can be derived [Bar83].

6.5 SUMMARY

In this chapter, we discussed the sequential Kalman filter, which is mathematically identical to the Kalman filter, but which avoids matrix inversion. This is an attractive formulation for embedded systems in which computational time and memory are at a premium. However, sequential filtering can only be used if the noise covariance is diagonal, or if the noise covariance is constant. Information filtering is also equivalent to the Kalman filter, but it propagates the inverse of the covariance. This can be computationally beneficial in cases in which the number of measurements is much larger than the number of states. Square root filtering and U-D filtering effectively increase the precision of the Kalman filter. Although these approaches require additional computational effort, they can help prevent divergence and instability. Gerald Bierman's book provides an excellent and comprehensive overview of square root and U-D filtering [Bie77b].

We see that we have a number of different choices when implementing a Kalman filter.

- Covariance filtering or information filtering
- Standard filtering, square root filtering, or U-D filtering
- Batch filtering or sequential filtering

Any of these choices can be made independently of the other choices. For instance, we can choose to combine information filtering with square root filtering [Kam71] in much the same way as we combined covariance filtering with square root filtering in this chapter. The choices in the list above gives us a total of 12 different Kalman filter formulations (two choices in the first item, three choices in the second item, and two choices in the third item). There are also other choices that are not listed above, especially other types of square root filtering. A numerical comparison of various Kalman filter formulations (including the standard filter, the square root covariance filter, the square root information filter, and the Chandrasekhar algorithm) is given in [Ver86]. Numerical and computational comparisons of various Kalman filtering approaches are given in [Bie73, Bie77a]. Continuous-time square root filtering is discussed in [Mor78] and in Section 8.3.3 of this book.

PROBLEMS

Written exercises

6.1 In this chapter, we discussed alternatives to the standard Kalman filter formulation. Some of these alternatives include the sequential Kalman filter, the information filter, and the square root filter.

a) What is the advantage of the sequential Kalman filter over the batch Kalman filter? What is the advantage of the batch Kalman filter over the sequential Kalman filter?

b) What is the advantage of the information filter over the standard Kalman filter? What is the advantage of the standard Kalman filter over the information filter?

c) What is the advantage of the square root filter over the standard Kalman filter? What is an advantage of the standard Kalman filter over the square root Kalman filter?

6.2 Suppose that you have a system with the following measurement and measurement noise covariance matrices:

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

$$R = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

You want to use a sequential Kalman filter to estimate the state of the system. Derive the normalized measurement, measurement matrix, and measurement noise covariance matrix that could be used in a sequential Kalman filter.

6.3 Consider the two alternative forms for the information matrix time-update equation. What advantages does Equation (6.28) have? What advantages does Equation (6.30) have?

6.4 A radioactive mass has a half-life of τ seconds. At each time step k the number of emitted particles x is half of what it was one time step ago, but there is some error w_k (zero-mean with variance Q_k) in the number of emitted particles due to background radiation. At each time step the number of emitted particles is counted with two separate and independent instruments. The instruments used to count the number of emitted particles both have a random error at each time step that is zero-mean with a unity variance. The initial uncertainty in the number of radioactive particles is a random variable with zero mean and unity variance.

a) The discrete-time equations that model this system have a one-dimensional state and a two-dimensional measurement. Use the information filter to compute the *a priori* and *a posteriori* information matrix at $k = 1$ and $k = 2$. Assume that $Q_0 = 1$ and $Q_1 = 5/4$.

b) Another way to solve this problem is to realize that the two measurements can be averaged to form a single measurement with a smaller variance than the two independent measurements. What is the variance of the averaged measurement at each time step? Use the standard Kalman filter equations

to compute the *a priori* and *a posteriori* covariance matrix at $k = 1$ and $k = 2$, and verify that it is the inverse of the information matrix that you computed in part (a).

6.5 Prove that the singular values of a diagonal matrix are the magnitudes of the diagonal elements.

6.6 Prove that SS^T is symmetric positive semidefinite for any S matrix.

6.7 Find an upper triangular matrix S (using only paper and pencil) such that

$$SS^T = \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix}$$

Is your solution unique?

6.8 Find an upper triangular matrix S (using only paper and pencil) such that

$$SS^T = \begin{bmatrix} 5 & 2 & -2 \\ 2 & 2 & -1 \\ -2 & -1 & 1 \end{bmatrix}$$

How many solutions exist to this problem?

6.9 Verify Equation (6.70). Hint: Equate the two sides of the equation, take the trace, and solve for γ. Make sure to explain why taking the trace is valid.

6.10 Suppose that an orthogonal matrix $\tilde{T}$ is desired to satisfy Equation (6.97), where Cholesky factorization is used to compute the matrix square roots on the left side of the equation. This equation can then be written as $U = \tilde{T}A$, where U is an upper triangular matrix. Show that such a transformation cannot be found unless the two-norm of the first column of A happens to be equal to $|U_{11}|$. [Note that this does not necessarily prevent the possibility of the transformation of Equation (6.97), because U could be nontriangular if nontriangular square root matrices are used to form the U matrix.]

6.11 Use the Householder method (using only paper and pencil) to find an orthogonal T such that $TA = \begin{bmatrix} W \\ 0 \end{bmatrix}$ where W is a 2×2 matrix and

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 0 & 1 \\ 2 & 2 \end{bmatrix}$$

6.12 Use the modified Gram–Schmidt method (using only paper and pencil) to solve Problem 6.11.

6.13 Compute the U-D factorization (using only paper and pencil) for the matrix

$$P = \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix}$$

Computer exercises

6.14 Consider the RLC circuit of Example 1.8 with $R = 100$ and $L = C = 1$. Suppose the applied voltage is continuous-time, zero-mean white noise with a standard deviation of 3. The initial capacitor voltage and inductor current are both zero. Discretize the system with a time step of 0.1. The discrete-time measurements consist of the capacitor voltage and the inductor current, both measurements containing zero-mean unity variance noise. Implement a sequential Kalman filter for the system. Simulate the system for 2 seconds. Let the initial state estimate be equal to the initial state, and the initial estimation covariance be equal to $0.1I$. Hint: Set the discrete-time process noise covariance $Q = Q_c\Delta t$, where Q_c is the covariance of the continuous-time process noise, and Δt is the discretization step size. Q will be nondiagonal, which means you need to use the algorithm in Section 2.7 to simulate the process noise.

a) Generate a plot showing the *a priori* variance of the capacitor voltage estimation error, and the two *a posteriori* variances of the capacitor voltage estimation error.

b) Generate a plot showing a typical trace of the true, *a posteriori* estimated, and measured capacitor voltage. What is the standard deviation of the capacitor voltage measurement error? What is the standard deviation of the capacitor voltage estimation error?

6.15 The pitch motion of an aircraft flying at constant speed can be approximately described by the following equations [Ste94]:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -0.5680 & 17.9800 \\ 1.0000 & -1.2370 \end{bmatrix} x + \begin{bmatrix} 0.1750 & 0.1750 \\ -0.0010 & -0.0010 \end{bmatrix} u + \begin{bmatrix} 17.9800 \\ -1.2370 \end{bmatrix} w \\ y(t_k) &= x(t_k) + v_k \end{aligned}$$

where x_1 is the pitch rate, x_2 is the angle of attack, u consists of the elevator and flap angles, and w is disturbance due to wind. Suppose that the variance of the wind disturbance is 0.001, and the measurement variances are 0.3. Discretize the system with a step size of 0.01 and simulate the system and a square root Kalman filter for 100 time steps. Use an initial state of zero, an initial state estimate of zero, an initial estimation-error covariance of $0.01I$, and a control input of zero. Hint: Set the discrete-time process noise covariance $Q = Q_c\Delta t$, where Q_c is the covariance of the continuous-time process noise, and Δt is the discretization step size. Q will be nondiagonal, which means you need to use the algorithm in Section 2.7 to simulate the process noise.

a) Generate a plot showing the *a posteriori* variance of the estimation errors of the two states.

b) Generate a plot showing a typical trace of the true, *a posteriori* estimated, and measured pitch rate. What is the standard deviation of the pitch rate measurement error? What is the standard deviation of the pitch rate estimation error?

c) Generate a plot showing a typical trace of the true, *a posteriori* estimated, and measured angle of attack. What is the standard deviation of the angle of attack measurement error? What is the standard deviation of the angle of attack estimation error?

CHAPTER 7

Kalman filter generalizations

> Many practical systems exist in which the correlation times of the random measurement errors are *not* short compared to times of interest in the system; for brevity such errors are called "colored" noise.
>
> —Arthur Bryson and Donald Johansen [Bry65]

In the last two chapters, we derived the discrete-time Kalman filter and presented some alternate but mathematically equivalent formulations. In this chapter we will discuss some generalizations of the Kalman filter that will make it more flexible and effective for a broader class of problems. For example, in our derivation of the Kalman filter in Chapter 5 we assumed that the process noise and measurement noise were uncorrelated. In Section 7.1, we will show how correlated process and measurement noise changes the Kalman filter equations. Our derivation in Chapter 5 also assumed that the process noise and measurement noise were white. We modify the Kalman filter to deal with colored process noise and measurement noise in Section 7.2.

Many Kalman filter implementations are coded in embedded systems (rather than desktop computers) where memory and computational effort is still a primary consideration. For this reason, we can replace the time-varying Kalman filter of Chapter 5 with a steady-state Kalman filter that often performs nearly as well. This means that we do not have to compute the estimation-error covariance or Kalman

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

gain in real time. This is discussed in Section 7.3, which includes a presentation of α-β and α-β-γ filtering.

When the dynamics of the system are not perfectly known, then the Kalman filter may not provide acceptable state estimates. This can be addressed by giving more weight to recent measurements when updating the state estimate, and discounting measurements that arrived a long time ago. This is called the fading-memory filter and is discussed in Section 7.4. Finally, there may be other information about the states other than the system model. For example, there may be state constraints that we know must be satisfied. Section 7.5 discusses several ways to incorporate state equality constraints and state inequality constraints into the formulation of the Kalman filter.

7.1 CORRELATED PROCESS AND MEASUREMENT NOISE

Our derivation of the Kalman filter in Chapter 5 assumed that the process noise and measurement noise were uncorrelated. In this section, we will show how correlated process and measurement noise changes the Kalman filter equations. Suppose that we have a system given by

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \\ E[w_k w_j^T] &= Q_k \delta_{k-j} \\ E[v_k v_j^T] &= R_k \delta_{k-j} \\ E[w_k v_j^T] &= M_k \delta_{k-j+1} \end{aligned} \tag{7.1}$$

We see that the process noise in the system equation is correlated with the measurement noise, with the cross covariance given by $M_k\delta_{k-j+1}$. Our derivation in Chapter 5 assumed that M_k was zero, but in this section we will relax that assumption. For example, suppose that our system is an airplane and winds are buffeting the plane. We are using an anemometer to measure wind speed as an input to our Kalman filter. So the random gusts of wind affect both the process (i.e., the airplane dynamics) and the measurement (i.e., the sensed wind speed). We see that there is a correlation between the process noise and the measurement noise. From the above equation, we see that the process noise at time k is correlated with the measurement noise at time $(k+1)$; that is, w_k is correlated with v_{k+1}. This is because w_k affects the state at time $(k+1)$, just as v_{k+1} affects the measurement at time $(k+1)$.

In order to find the Kalman filter equations for the correlated noise system, we will define the estimation errors as

$$\begin{aligned} \epsilon_k^- &= x_k - \hat{x}_k^- \\ \epsilon_k^+ &= x_k - \hat{x}_k^+ \end{aligned} \tag{7.2}$$

As in our original Kalman filter derivation of Chapter 5, we still assume that our update equations for the state estimate are given as follows:

$$\begin{aligned}\hat{x}_k^- &= F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-)\end{aligned} \tag{7.3}$$

The gain matrix K_k will not be the same as we derived in Chapter 5, but the form of the measurement update equation is still the same. Equation (7.2) can be expanded using the above equations as

$$\begin{aligned}\epsilon_k^- &= x_k - \hat{x}_k^- \\ &= (F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1}) - (F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1}) \\ &= F_{k-1}\epsilon_{k-1}^+ + w_{k-1} \\ \epsilon_k^+ &= x_k - \left[\hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-)\right] \\ &= \epsilon_k^- - K_k(H_kx_k + v_k - H_k\hat{x}_k^-) \\ &= \epsilon_k^- - K_k(H_k\epsilon_k^- + v_k)\end{aligned} \tag{7.4}$$

The *a priori* and *a posteriori* estimation-error covariances can be written as

$$\begin{aligned}P_k^- &= E[\epsilon_k^-(\epsilon_k^-)^T] \\ &= F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \\ P_k^+ &= E[\epsilon_k^+(\epsilon_k^+)^T] \\ &= E\left\{[\epsilon_k^- - K_k(H_k\epsilon_k^- + v_k)][\epsilon_k^- - K_k(H_k\epsilon_k^- + v_k)]^T\right\} \\ &= P_k^- - K_kH_kP_k^- - K_kE[v_k(\epsilon_k^-)^T] - P_k^-H_k^TK_k^T + \\ &\quad K_kH_kP_k^-H_k^TK_k^T + K_kE[v_k(\epsilon_k^-)^T]H_k^TK_k^T - \\ &\quad E(\epsilon_k^-v_k^T)K_k^T + K_kH_kE(\epsilon_k^-v_k^T)K_k^T + K_kE(v_kv_k^T)K_k^T\end{aligned} \tag{7.5}$$

In order to simplify this expression for P_k^+, we need to find an expression for $E(\epsilon_k^-v_k^T)$. This can be computed as

$$\begin{aligned}E(\epsilon_k^-v_k^T) &= E[(x_k - \hat{x}_k^-)v_k^T] \\ &= E(x_kv_k^T - \hat{x}_k^-v_k^T) \\ &= E[(F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1})v_k^T] - E[\hat{x}_k^-v_k^T] \\ &= 0 + 0 + M_k - 0\end{aligned} \tag{7.6}$$

In the above equation, the first term is 0 because x_{k-1} is independent of v_k, and v_k is zero-mean. The second term is 0 because u_{k-1} is independent of v_k. The last term is 0 because the *a priori* state estimate at time k is independent of v_k. Substituting this expression for $E(\epsilon_k^-v_k^T)$ into Equation (7.5) gives

$$\begin{aligned}P_k^+ &= P_k^- - K_kH_kP_k^- - K_kM_k^T - P_k^-H_k^TK_k^T + K_kH_kP_k^-H_k^TK_k^T + \\ &\quad K_kM_k^TH_k^TK_k^T - M_kK_k^T + K_kH_kM_kK_k^T + K_kR_kK_k^T \\ &= (I - K_kH_k)P_k^-(I - K_kH_k)^T + K_kR_kK_k^T + \\ &\quad K_k(H_kM_k + M_k^TH_k^T)K_k^T - M_kK_k^T - K_kM_k^T\end{aligned} \tag{7.7}$$

Now we need to find the gain matrix K_k that minimizes $\mathrm{Tr}(P_k^+)$. Recall from Equation (1.66) that

$$\frac{\partial \mathrm{Tr}(ABA^T)}{\partial A} = 2AB \text{ if } B \text{ is symmetric} \tag{7.8}$$

We can use this fact to derive

$$\begin{aligned}\frac{\partial \mathrm{Tr}(P_k^+)}{\partial K_k} &= -2(I - K_k H_k)P_k^- H_k^T + 2K_k R_k + \\ &\quad 2K_k(H_k M_k + M_k^T H_k^T) - M_k - M_k \\ &= 2\left[K_k(H_k P_k^- H_k^T + H_k M_k + M_k^T H_k^T + R_k) - \right. \\ &\quad \left. P_k^- H_k^T - M_k\right] \end{aligned} \tag{7.9}$$

In order to make this partial derivative zero, we need to set the gain K_k as follows:

$$K_k = (P_k^- H_k^T + M_k)(H_k P_k^- H_k^T + H_k M_k + M_k^T H_k^T + R_k)^{-1} \tag{7.10}$$

This gives the optimal Kalman gain matrix for the system with correlated process and measurement noise. The estimation-error covariance is then obtained from Equation (7.7) as

$$\begin{aligned} P_k^+ &= (I - K_k H_k)P_k^-(I - K_k H_k)^T + \\ &\quad K_k(H_k M_k + M_k^T H_k^T + R_k)K_k^T - M_k K_k^T - K_k M_k^T \\ &= P_k^- - K_k H_k P_k^- - P_k^- H_k^T K_k^T + \\ &\quad K_k(H_k P_k^- H_k^T + H_k M_k + M_k^T H_k^T + R_k)K_k^T - \\ &\quad M_k K_k^T - K_k M_k^T \\ &= P_k^- - K_k(H_k P_k^- + M_k^T) - (P_k^- H_k^T + M_k)K_k^T + \\ &\quad (P_k^- H_k^T + M_k)(H_k P_k^- H_k^T + H_k M_k + M_k^T H_k^T + R_k)^{-1}(H_k P_k^- + M_k^T) \\ &= P_k^- - K_k(H_k P_k^- + M_k^T) - (P_k^- H_k^T + M_k)K_k^T + (P_k^- H_k^T + M_k)K_k^T \\ &= P_k^- - K_k(H_k P_k^- + M_k^T) \end{aligned} \tag{7.11}$$

This gives the measurement-update equation for the estimation-error covariance for the Kalman filter with correlated process and measurement noise. The measurement-update equation for the state estimate is the same as for the standard Kalman filter and is given in Equation (7.3). The time-update equations for the state estimate and the estimation-error covariance are also the same as before. The Kalman filter for the system with correlated process and measurement noise can be summarized as follows.

The general discrete-time Kalman filter

1. The system and measurement equations are given as

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \\ E[w_k w_j^T] &= Q_k \delta_{k-j} \\ E[v_k v_j^T] &= R_k \delta_{k-j} \\ E[w_k v_j^T] &= M_k \delta_{k-j+1} \end{aligned} \tag{7.12}$$

2. The Kalman filter is initialized as

$$\begin{aligned} \hat{x}_0^+ &= E(x_0) \\ P_0^+ &= E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T] \end{aligned} \tag{7.13}$$

3. For each time step $k = 1, 2, \cdots$, the Kalman filter equations are given as

$$\begin{aligned} P_k^- &= F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \\ K_k &= (P_k^- H_k^T + M_k)(H_k P_k^- H_k^T + H_k M_k + M_k^T H_k^T + R_k)^{-1} \\ &= P_k^+ \left(H_k^T + (P_k^-)^{-1}M_k\right)\left(R_k - M_k^T(P_k^-)^{-1}M_k\right)^{-1} \\ \hat{x}_k^- &= F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \\ P_k^+ &= (I - K_kH_k)P_k^-(I - K_kH_k)^T + \\ &\quad K_k(H_kM_k + M_k^TH_k^T + R_k)K_k^T - M_kK_k^T - K_kM_k^T \\ &= \Big[(P_k^-)^{-1} + \left(H_k^T + (P_k^-)^{-1}M_k\right)\left(R_k - M_k^T(P_k^-)^{-1}M_k\right)^{-1} \times \\ &\quad \left(H_k + M_k^T(P_k^-)^{-1}\right)\Big]^{-1} \\ &= P_k^- - K_k(H_kP_k^- + M_k^T) \end{aligned} \tag{7.14}$$

The second form for P_k^+ and the second form for K_k can be derived by following a procedure similar to that shown in Section 3.3.1. Note that this is a generalization of the Kalman filter that was presented in Equation (5.19). If $M_k = 0$, then the above equations reduce to Equation (5.19).

■ EXAMPLE 7.1

Consider the following scalar system:

$$\begin{aligned} x_k &= 0.8x_{k-1} + w_{k-1} \\ y_k &= x_k + v_k \\ E[w_kw_j^T] &= 1\delta_{k-j} \\ E[v_kv_j^T] &= 0.1\delta_{k-j} \\ E[w_kv_j^T] &= M\delta_{k-j+1} \end{aligned} \tag{7.15}$$

We can use the method discussed in Section 2.7 to simulate correlated noise. The Kalman filter equations given above can then be run to obtain an estimate of the state. Table 7.1 shows (for several values of M) the variance of the estimation error for the standard Kalman filter (when $M = 0$ is assumed) and for the correlated noise Kalman filter (when the correct value of M is used). When $M = 0$, the estimation-error variances are the same for the two filters, as expected. However, when $M \neq 0$, the filter that uses the correct value of M performs noticeably better than the filter that incorrectly assumes that $M = 0$.

∇∇∇

Table 7.1 Experimental estimation-error variance (50 time steps) for Example 7.1 when there is a cross covariance M between the process noise and the measurement noise. The standard filter assumes that $M = 0$, and the correlated filter uses the correct value of M

Correlation M	Standard Filter ($M = 0$ assumed)	Correlated Filter (correct M used)
0	0.076	0.076
0.25	0.030	0.019
−0.25	0.117	0.052

7.2 COLORED PROCESS AND MEASUREMENT NOISE

Our derivation of the Kalman filter in Chapter 5 assumed that the process noise and measurement noise were both white. In this section, we will show how to deal with colored process noise, and we will present two methods for dealing with colored measurement noise.

7.2.1 Colored process noise

If the process noise is colored, then it is straightforward to modify the system equations and obtain an equivalent but higher-order system with white process noise [Buc68]. Then the standard Kalman filter equations can be applied. For example, suppose that we have an LTI system given as

$$x_k = Fx_{k-1} + w_{k-1} \tag{7.16}$$

where the covariance of w_k is equal to Q_k. Further suppose that the process noise is the output of a dynamic system:

$$w_k = \psi w_{k-1} + \zeta_{k-1} \tag{7.17}$$

where ζ_{k-1} is zero-mean white noise that is uncorrelated with w_{k-1}. In this case, we can see that the covariance between w_k and w_{k-1} is equal to

$$\begin{aligned} E(w_k w_{k-1}^T) &= E(\psi w_{k-1} w_{k-1}^T + \zeta_{k-1} w_{k-1}^T) \\ &= \psi Q_{k-1} + 0 \end{aligned} \tag{7.18}$$

The 0 arises because w_{k-1} is independent from ζ_{k-1}, and ζ_{k-1} is zero-mean. We see that w_k is colored process noise (because it is correlated with itself at other time steps). We can combine Equations (7.16) and (7.17) to obtain

$$\begin{aligned} \begin{bmatrix} x_k \\ w_k \end{bmatrix} &= \begin{bmatrix} F & I \\ 0 & \psi \end{bmatrix} \begin{bmatrix} x_{k-1} \\ w_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ \zeta_{k-1} \end{bmatrix} \\ x'_k &= F' x'_{k-1} + w'_{k-1} \end{aligned} \tag{7.19}$$

This is an augmented system with a new state x', a new system matrix F', and a new process noise vector w' whose covariance is given as follows:

$$
\begin{aligned}
E(w'_k w'^T_k) &= \begin{bmatrix} 0 & 0 \\ 0 & E(\zeta_k \zeta_k^T) \end{bmatrix} \\
&= Q'_k
\end{aligned} \tag{7.20}
$$

Now the standard Kalman filter can be run on this augmented system that has white process noise, as long as we know $E(\zeta_k \zeta_k^T)$. Computational effort increases because the state vector dimension has doubled, but conceptually this is a straightforward approach to dealing with colored process noise.

7.2.2 Colored measurement noise: State augmentation

Now suppose that we have colored measurement noise. Our system and measurement equations are given as

$$
\begin{aligned}
x_k &= F_{k-1} x_{k-1} + w_{k-1} \\
y_k &= H_k x_k + v_k \\
v_k &= \psi_{k-1} v_{k-1} + \zeta_{k-1} \\
w_k &\sim N(0, Q_k) \\
\zeta_k &\sim N(0, Q_{\zeta k}) \\
E[w_k w_j^T] &= Q_k \delta_{k-j} \\
E[\zeta_k \zeta_j^T] &= Q_{\zeta k} \delta_{k-j} \\
E[w_k \zeta_j^T] &= 0
\end{aligned} \tag{7.21}
$$

The measurement noise is itself the output of a linear system. The covariance of the measurement noise is given as

$$
\begin{aligned}
E[v_k v_{k-1}^T] &= E[(\psi_{k-1} v_{k-1} + \zeta_{k-1}) v_{k-1}^T] \\
&= \psi_{k-1} E[v_{k-1} v_{k-1}^T]
\end{aligned} \tag{7.22}
$$

There are a couple of ways to solve the colored measurement-noise problem. It was solved by Richard Bucy for continuous-time problems in [Buc68]. Here we will solve the discrete-time problem by augmenting the state. This was originally proposed in [Bry65] in the context of continuous-time systems. We augment the original system model as follows:

$$
\begin{aligned}
\begin{bmatrix} x_k \\ v_k \end{bmatrix} &= \begin{bmatrix} F_{k-1} & 0 \\ 0 & \psi_{k-1} \end{bmatrix} \begin{bmatrix} x_{k-1} \\ v_{k-1} \end{bmatrix} + \begin{bmatrix} w_{k-1} \\ \zeta_{k-1} \end{bmatrix} \\
y_k &= \begin{bmatrix} H_k & I \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix} + 0
\end{aligned} \tag{7.23}
$$

This can be written as

$$
\begin{aligned}
x'_k &= F'_{k-1} x'_{k-1} + w'_{k-1} \\
y_k &= H'_k x'_k + v'_k
\end{aligned} \tag{7.24}
$$

This system is equivalent to the original system but has a modified state x', state transition matrix F', process noise w', measurement matrix H', and measurement

noise v'. The covariance of the process noise and the covariance of the measurement noise are computed as

$$\begin{aligned} E[w'_k w'^T_k] &= E\left[\begin{pmatrix} w_k \\ \zeta_k \end{pmatrix} \begin{pmatrix} w_k^T & \zeta_k^T \end{pmatrix}\right] \\ &= \begin{bmatrix} Q_k & 0 \\ 0 & Q_{\zeta k} \end{bmatrix} \\ E[v'_k v'^T_k] &= 0 \end{aligned} \tag{7.25}$$

We see that there is no measurement noise, which is equivalent to saying that the measurement noise is white with a mean of zero and a covariance of zero. Theoretically, it is fine to have zero measurement noise in the Kalman filter. In fact, Kalman's original paper [Kal60] was written without any restrictions on the singularity of the measurement-noise covariance. But practically speaking, a singular measurement-noise covariance often results in numerical problems [May79, p. 249], [Ste94, p. 365]. For that reason we will present another approach to dealing with colored measurement noise in the next section.

7.2.3 Colored measurement noise: Measurement differencing

In this section we present a method for dealing with colored measurement noise that does not rely on augmenting the state vector. This approach is due to [Bry68]. As in the previous section, our system is given as

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \\ v_k &= \psi_{k-1}v_{k-1} + \zeta_{k-1} \\ w_k &\sim (0, Q_k) \\ \zeta_k &\sim (0, Q_{\zeta k}) \\ E[w_k w_j^T] &= Q_k \delta_{k-j} \\ E[\zeta_k \zeta_j^T] &= Q_{\zeta k}\delta_{k-j} \\ E[w_k \zeta_j^T] &= 0 \end{aligned} \tag{7.26}$$

Now we define an auxiliary signal y'_k as follows:

$$y'_{k-1} = y_k - \psi_{k-1}y_{k-1} \tag{7.27}$$

Substitute for y_k and y_{k-1} in the above definition of y'_{k-1} to obtain

$$\begin{aligned} y'_{k-1} &= (H_k x_k + v_k) - \psi_{k-1}(H_{k-1}x_{k-1} + v_{k-1}) \\ &= H_k(F_{k-1}x_{k-1} + w_{k-1}) + v_k - \psi_{k-1}(H_{k-1}x_{k-1} + v_{k-1}) \\ &= (H_k F_{k-1} - \psi_{k-1}H_{k-1})x_{k-1} + H_k w_{k-1} + v_k - \psi_{k-1}v_{k-1} \\ &= (H_k F_{k-1} - \psi_{k-1}H_{k-1})x_{k-1} + (H_k w_{k-1} + \zeta_{k-1}) \\ &= H'_{k-1}x_{k-1} + v'_{k-1} \end{aligned} \tag{7.28}$$

H'_{k-1} and v'_{k-1} are defined by the above equation. We see that we have a new measurement equation for the measurement y'_{k-1} that has a measurement matrix

H'_{k-1} and measurement noise v'_{k-1}. Our new but equivalent system can therefore be written as

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + w_{k-1} \\ y'_k &= H'_k x_k + v'_k \end{aligned} \tag{7.29}$$

The covariance of the new measurement noise v', and the cross covariance between the process noise w and the new measurement noise v', can be obtained as

$$\begin{aligned} E[v'_k {v'_k}^T] &= E[(H_{k+1}w_k + \zeta_k)(w_k^T H_{k+1}^T + \zeta_k^T)] \\ &= H_{k+1}Q_k H_{k+1}^T + Q_{\zeta k} \\ E[w_k {v'_k}^T] &= E[w_k(w_k^T H_{k+1}^T + \zeta_k^T)] \\ &= Q_k H_{k+1}^T \end{aligned} \tag{7.30}$$

where we have used the fact that w_k and ζ_k are independent and zero-mean.

Now we will define the *a priori* and *a posteriori* state estimates for the system of Equation (7.29) slightly differently than we have up to this point. The state estimate $\hat{x}_k^-$ at time k is defined as the expected value of the state x_k conditioned on measurements up to and including time k.

$$\hat{x}_k^- = E[x_k | y_1, \cdots, y_k] \tag{7.31}$$

The state estimate at time $\hat{x}_k^+$ at time k is defined as the expected value of the state x_k conditioned on measurements up to and including time $(k+1)$. We assume that it is given by a standard linear predictor/corrector combination:

$$\begin{aligned} \hat{x}_k^+ &= E[x_k | y_1, \cdots, y_{k+1}] \\ &= \hat{x}_k^- + K_k(y'_k - H'_k \hat{x}_k^-) \end{aligned} \tag{7.32}$$

Note that these definitions of $\hat{x}_k^-$ and $\hat{x}_k^+$ are slightly different than the definitions used elsewhere in this book. Usually, $\hat{x}_k^-$ is based on measurements up to and including time $k-1$, and $\hat{x}_k^+$ is based on measurements up to and including time k. In this section, these two estimates are both based on one additional measurement. As in our previous derivations, we choose the gain K_k to minimize the trace of the covariance of the estimation error. In equation form this is written as

$$K_k = \text{argmin Tr } E\left[(x_k - \hat{x}_k^+)(x_k - \hat{x}_k^+)^T\right] \tag{7.33}$$

We will not work through the details here, but in [Bry68] it is shown that this minimization leads to the following estimator equations.

The discrete-time Kalman filter with colored measurement noise

1. Our system and measurement equations are given by Equation (7.26).

2. y'_k and H'_k are defined by Equations (7.27) and (7.28).

3. At each time step, execute the following equations to update the state estimate:

$$
\begin{aligned}
\hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k' - H_k'\hat{x}_k^-) \\
\hat{x}_{k+1}^- &= F_k\hat{x}_k^+ + C_k(y_k' - H_k'\hat{x}_k^+) \\
K_k &= P_k^- {H_k'}^T (H_k' P_k^- {H_k'}^T + R_k)^{-1} \\
M_k &= Q_k H_{k+1}^T \\
C_k &= M_k (H_k' P_k^- {H_k'}^T + R_k)^{-1} \\
P_k^+ &= (I - K_k H_k') P_k^- (I - K_k H_k')^T + K_k R_k K_k^T \\
P_{k+1}^- &= F_k P_k^+ F_k^T + Q_k - C_k M_k^T - F_k K_k M_k - M_k^T K_k^T F_k^T \qquad (7.34)
\end{aligned}
$$

A similar approach to the continuous-time filter with colored measurement noise is given in [Ste68].

■ EXAMPLE 7.2

Consider the following linear system with colored measurement noise:

$$
\begin{aligned}
x_k &= \begin{bmatrix} 0.70 & -0.15 \\ 0.03 & 0.79 \end{bmatrix} x_{k-1} + \begin{bmatrix} 0.15 \\ 0.21 \end{bmatrix} w_{k-1} \\
y_k &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + v_k \\
v_k &= \psi v_{k-1} + \zeta_{k-1} \\
E[w_k w_j^T] &= 1\delta_{k-j} \\
E[\zeta_k \zeta_j^T] &= \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix} \delta_{k-j} \\
E[w_k \zeta_j^T] &= 0 \qquad (7.35)
\end{aligned}
$$

The scalar ψ indicates the correlation of the measurement noise. If $\psi = 0$ then the measurement noise is white. As ψ increases, the color of the measurement noise increases (i.e., it contains more low-frequency components and less high-frequency components). In this example, we simulate the Kalman filter for this system in three different ways. First, we simulate the standard Kalman filter while simply ignoring the colored nature of the measurement noise. Second, we augment the state vector as described in Section 7.2.2, which will take the colored nature of the measurement noise into account, and then simulate the Kalman filter. Third, we implement the measurement-differencing approach that is described in this section, which again takes the colored nature of the measurement noise into account, and then simulate the filter. Table 7.2 shows the experimental values of the trace of the covariance of the estimation error for the three filters. We can see that if $\psi = 0$ then the three filters perform essentially identically. (There is some difference in performance between the filters because the performance measures in Table 7.2 are experimentally determined statistical values.) However, as ψ increases (i.e., the color of the measurement noise increases) we see that the filters that take this into account provide increasingly better performance compared to the standard Kalman filter. This example shows the improvement in perfor-

mance that is possible with the colored measurement-noise filters described in this section.

Table 7.2 Experimental values of the trace of the covariance of the estimation error (500 time steps) for Example 7.2. As the color content of the measurement noise increases (i.e., as ψ increases) the colored measurement-noise filters provide increasingly better performance than the standard Kalman filter

Color ψ	Standard Filter	Augmented Filter	Measurement Differencing
0.0	0.245	0.245	0.247
0.2	0.260	0.258	0.259
0.5	0.308	0.294	0.295
0.9	0.631	0.407	0.406

$\nabla\nabla\nabla$

7.3 STEADY-STATE FILTERING

Many Kalman filter implementations are coded in embedded systems (rather than desktop computers) in which memory and computational effort is still a primary consideration. If the underlying system is time-invariant, and the process- and measurement-noise covariances are time-invariant, then we can replace the time-varying Kalman filter of Chapter 5 with a steady-state Kalman filter. The steady-state filter often performs nearly as well as the time-varying filter. Using a steady-state filter has the advantage that we do not have to compute the estimation-error covariance or Kalman gain in real time. Note that a steady-state Kalman filter is still a dynamic system. The term "steady-state" Kalman filtering means that the Kalman filter is time-invariant; it is the Kalman gain that is in steady state.

As an example, recall the scalar system discussed in Example 5.2:

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + v_k \\ w_k &\sim (0,1) \\ v_k &\sim (0,1) \end{aligned} \tag{7.36}$$

We saw from Example 5.2 that the Kalman gain converged to a steady-state value after a few time steps:

$$\begin{aligned} \lim_{k\to\infty} K_k &= K_\infty \\ &= \frac{1+\sqrt{5}}{3+\sqrt{5}} \end{aligned} \tag{7.37}$$

So instead of performing the measurement-update equation for P_k, the time-update equation for P_k, and the Kalman gain computation for K_k at each time step, we

can simply use the constant K_∞ as our Kalman gain at each time step. For a system with many states, this can save a lot of computational effort, especially considering the fact that this will allow us to avoid real-time matrix inversions. The steady-state Kalman filter for this example is simply given as

$$\begin{aligned}
\hat{x}_k^- &= F\hat{x}_{k-1}^+ \\
\hat{x}_k^+ &= \hat{x}_k^- + K_\infty(y_k - H\hat{x}_k^-) \\
&= F\hat{x}_{k-1}^+ + K_\infty(y_k - HF\hat{x}_{k-1}^+) \\
&= (I - K_\infty H)F\hat{x}_{k-1}^+ + K_\infty y_k
\end{aligned} \tag{7.38}$$

The steady-state Kalman filter is not optimal because we are not using the optimal Kalman gain at each time step (although it approaches optimality in the limit as $k \to \infty$). We are instead using the steady-state Kalman gain. However, for many problems of practical interest, the performance of the steady-state filter is nearly indistinguishable from that of the time-varying filter. For any particular problem, the difference between the time-varying and steady-state filters needs to be assessed by simulation or experimental results.

One way to determine the steady-state Kalman gain is by numerical simulation. We can simply write a computer program to propagate the Kalman gain as a function of time, and then observe the value toward which the gain is converging.

Another way to determine the steady-state Kalman gain is to manipulate the Kalman filter equations from Equation (7.14). Recall the covariance time-update equation for a time-invariant system:

$$P_{k+1}^- = FP_k^+F^T + Q \tag{7.39}$$

Now substitute the expression for P_k^+ from Equation (7.14) into this equation to obtain

$$P_{k+1}^- = FP_k^-F^T - FK_kHP_k^-F^T - FK_kM^TF^T + Q \tag{7.40}$$

Now substitute the expression for K_k from Equation (7.14) into this equation to obtain

$$\begin{aligned}
P_{k+1}^- &= FP_k^-F^T - \\
&\quad F(P_k^-H^T + M)(HP_k^-H^T + HM + M^TH^T + R)^{-1}HP_k^-F^T - \\
&\quad F(P_k^-H^T + M)(HP_k^-H^T + HM + M^TH^T + R)^{-1}M^TF^T + Q \\
&= FP_k^-F^T - F(P_k^-H^T + M)(HP_k^-H^T + HM + M^TH^T + R)^{-1} \times \\
&\quad (HP_k^- + M^T)F^T + Q
\end{aligned} \tag{7.41}$$

If P_k^- converges to a steady-state value, then $P_k^- = P_{k+1}^-$ for large k. We will denote this steady-state value as P_∞, which means that we can write

$$\begin{aligned}
P_\infty &= FP_\infty F^T - \\
&\quad F(P_\infty H^T + M)(HP_\infty H^T + HM + M^TH^T + R)^{-1} \times \\
&\quad (HP_\infty + M^T)F^T + Q
\end{aligned} \tag{7.42}$$

This is called an algebraic Riccati equation (ARE), or more specifically a discrete ARE (DARE).[1] Once we have P_∞, we can substitute it for P_k^- in the Kalman gain

[1] In MATLAB's Control System Toolbox, we can solve this equation by invoking the command DARE($F^T, H^T, Q, HM + M^TH^T + R, FM$).

formula of Equation (7.14) to obtain the steady-state Kalman gain:

$$K_\infty = (P_\infty H^T + M)(HP_\infty H^T + HM + M^T H^T + R)^{-1} \tag{7.43}$$

There are systems for which the Riccati equation (and hence the Kalman gain) does *not* converge to a steady-state value. Furthermore, it may converge to different steady-state values depending on the initial condition P_0. Finally, even when it does converge to a steady-state value, it may result in an unstable Kalman filter. These issues comprise a rich field of study that has been reported widely in many books and papers [McG74, And79, Kai81, Goo84, Chu87]. We will summarize the most important Riccati equation convergence results below, but first we need to define what it means for a system to be controllable on the unit circle.

Definition 11 *The matrix pair (F, G) is controllable on the unit circle if there exists some matrix K such that $(F - GK)$ does not have any eigenvalues with magnitude 1.*

We illustrate this definition with some simple examples.

■ **EXAMPLE 7.3**

Consider the scalar system

$$x_{k+1} = x_k \tag{7.44}$$

In this example, $F = 1$ and $G = 0$. The system dynamics are independent of any control signal, and the system has an eigenvalue with a magnitude of 1. The system is not controllable on the unit circle because its eigenvalue has a magnitude of 1 regardless of the feedback control input.

∇∇∇

■ **EXAMPLE 7.4**

Consider the scalar system

$$x_{k+1} = 2x_k \tag{7.45}$$

In this example, $F = 2$ and $G = 0$. As in the previous example, the system dynamics are independent of any control signal. However, the system eigenvalue has a magnitude of 2. The system is controllable on the unit circle because there exists a feedback control gain K such that $(F - GK)$ does not have any eigenvalues with a magnitude of 1. In fact, regardless of the feedback control gain, the system eigenvalues will never have a magnitude of 1.

∇∇∇

■ **EXAMPLE 7.5**

Consider the system

$$x_{k+1} = \begin{bmatrix} F_1 & 0 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k \tag{7.46}$$

When the feedback control $u_k = -Kx_k$ is implemented, where $K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$, the closed-loop system becomes

$$x_{k+1} = \begin{bmatrix} F_1 & 0 \\ -K_1 & 1 - K_2 \end{bmatrix} x_k \tag{7.47}$$

The closed-loop system has eigenvalues at F_1 and $(1 - K_2)$. We see that if $F_1 = \pm 1$ then there is no feedback control gain K that results in all closed-loop eigenvalues having a nonunity magnitude, and the system is therefore not controllable on the unit circle. However, if $F_1 \neq \pm 1$, then we can find a feedback control gain K that does result in all closed-loop eigenvalues having a nonunity magnitude, and the system is therefore controllable on the unit circle.

$\nabla\nabla\nabla$

Next we summarize the most important Riccati equation convergence results from [Bit85, Pou86, Kai00], where proofs are given. Recall that the DARE is given as

$$\begin{aligned} P_\infty &= FP_\infty F^T - \\ &\quad F(P_\infty H^T + M)(HP_\infty H^T + HM + M^T H^T + R)^{-1} \times \\ &\quad (HP_\infty + M^T)F^T + Q \end{aligned} \tag{7.48}$$

We assume that $Q \geq 0$ and $R > 0$. We define G as any matrix such that $GG^T = Q - MR^{-1}M^T$. The corresponding steady-state Kalman gain K_∞ is given as

$$K_\infty = (P_\infty H^T + M)(HP_\infty H^T + HM + M^T H^T + R)^{-1} \tag{7.49}$$

The steady-state Kalman filter is given as

$$\hat{x}_k^+ = (I - K_\infty H)F\hat{x}_{k-1}^+ + K_\infty y_k \tag{7.50}$$

We say that the DARE solution P_∞ is stabilizing if it results in a stable steady-state filter. That is, P_∞ is defined as a stabilizing DARE solution if all of the eigenvalues of $(I - K_\infty H)F$ are less than one in magnitude.

Theorem 23 *The DARE has a unique positive semidefinite solution P_∞ if and only if both of the following conditions hold.*

1. *(F, H) is detectable.*
2. *$(F - MR^{-1}H, G)$ is stabilizable.*

Furthermore, the corresponding steady-state Kalman filter is stable. That is, the eigenvalues of $(I - K_\infty H)F$ have magnitude less than 1.

Theorem 23 does not preclude the existence of DARE solutions that are negative definite or indefinite. If such solutions exist, then they would result in an unstable Kalman filter. If we weaken the stabilizability condition in Theorem 23, we obtain the following.

Theorem 24 *The DARE has at least one positive semidefinite solution P_∞ if and only if both of the following conditions hold.*

1. *(F, H) is detectable.*

2. *$(F - MR^{-1}H, G)$ is controllable on the unit circle.*

Furthermore, exactly one of the positive semidefinite DARE solutions results in a stable steady-state Kalman filter.

Since controllability on the unit circle is a subset of stabilizability, we see that Theorem 24 is a subset of Theorem 23. Theorem 24 states conditions for the existence of exactly one stabilizing positive semidefinite DARE solution. However, there may be additional DARE solutions (positive semidefinite or otherwise) that result in unstable Kalman filters. If a time-varying Kalman filter is run in this situation, then the Kalman filter equations may converge to either a stable or an unstable filter, depending on the initial condition P_0^+. If we strengthen the controllability condition of Theorem 24, we obtain the following.

Theorem 25 *The DARE has at least one positive definite solution P_∞ if and only if both of the following conditions hold.*

1. *(F, H) is detectable.*

2. *$(F - MR^{-1}H, G)$ is controllable on and inside the unit circle.*

Furthermore, exactly one of the positive definite DARE solutions results in a stable steady-state Kalman filter.

If we drop the controllability condition in the above two theorems, we obtain the following.

Theorem 26 *The DARE has at least one positive semidefinite solution P_∞ if (F, H) is detectable. Furthermore, at least one such solution results in a marginally stable steady-state Kalman filter.*

Note that the resulting filter is only marginally stable, so it may have eigenvalues on the unit circle. Also note that this theorem poses a sufficient (not necessary) condition. That is, there may be a stable steady-state Kalman filter even if the conditions of the above theorem do not hold. Furthermore, even if the conditions of the theorem do hold, there may be DARE solutions that result in unstable Kalman filters.

■ **EXAMPLE 7.6**

Consider again the scalar system of Equation (7.36). We see that $F = 1$, $H = 1$, $Q = 1$, $R = 1$, and $M = 0$. Note that (F, H) is observable, and (F, G) is controllable for all G such that $GG^T = Q$ (recall that $M = 0$ for this example). We therefore know from Theorem 23 that the DARE has a unique positive semidefinite solution. We know from Theorem 25 that the DARE solution is not only positive semidefinite, but it is also positive definite. We also know from these two theorems that the corresponding steady-state Kalman filter is stable. The DARE for this system is given by

$$\begin{aligned} P &= FPF^T - FPH^T(HPH^T + R)^{-1}HPF^T + Q \\ &= P - P(P+1)^{-1}P + 1 \end{aligned} \tag{7.51}$$

This can be solved to obtain

$$P = \frac{1 \pm \sqrt{5}}{2} \tag{7.52}$$

So the DARE has two solutions, one of which is negative and one of which is positive. If we use the negative DARE solution in the steady-state Kalman filter we obtain

$$\begin{aligned} K &= PH^T(HPH^T + R)^{-1} \\ &= \frac{1-\sqrt{5}}{3-\sqrt{5}} \\ \hat{x}_k^+ &= (I - KH)F\hat{x}_{k-1}^+ + Ky_k \\ &= \frac{2}{3-\sqrt{5}}\hat{x}_{k-1}^+ + Ky_k \\ &\approx 2.62\hat{x}_{k-1}^+ + Ky_k \end{aligned} \tag{7.53}$$

We see that the resulting Kalman filter is unstable. However, if we use the positive DARE solution in the steady-state Kalman filter we obtain

$$\begin{aligned} K &= \frac{1+\sqrt{5}}{3+\sqrt{5}} \\ \hat{x}_k^+ &= \frac{2}{3+\sqrt{5}}\hat{x}_{k-1}^+ + Ky_k \\ &\approx 0.38\hat{x}_{k-1}^+ + Ky_k \end{aligned} \tag{7.54}$$

We see that the resulting Kalman filter is stable.

∇∇∇

■ **EXAMPLE 7.7**

Consider a scalar system with $F = 1$, $H = 1$, $Q = 0$, $R = 1$, and $M = 0$. Note (F, H) is detectable. However, it is not true that (F, G) is controllable on the unit circle for all G such that $GG^T = Q$. We therefore know from Theorem 24 that the DARE does not have a positive semidefinite solution that results in a stable Kalman filter. However, we know from Theorem 26 that the DARE has a positive semidefinite solution that results in a marginally stable Kalman filter. The DARE for this system is given by

$$\begin{aligned} P &= FPF^T - FPH^T(HPH^T + R)^{-1}HPF^T + Q \\ &= P - P(P+1)^{-1}P \end{aligned} \tag{7.55}$$

This has two solutions for P, both of which are 0 (i.e., positive semidefinite). If we use this solution in the steady-state Kalman filter we obtain

$$\begin{aligned} K &= 0 \\ \hat{x}_k^+ &= \hat{x}_{k-1}^+ \end{aligned} \tag{7.56}$$

We see that the resulting Kalman filter is marginally stable (the eigenvalue is 1).

∇∇∇

■ **EXAMPLE 7.8**

Consider a scalar system with $F = 2$, $H = 1$, $Q = 0$, $R = 1$, and $M = 0$. Note (F, H) is detectable. Also (F, G) is controllable on and inside the unit circle for all G such that $GG^T = Q$. We therefore know from Theorem 24 that the DARE has exactly one positive semidefinite solution that results in a stable Kalman filter.

However, we know from Theorem 26 that the DARE has exactly one positive semidefinite solution that results in a marginally stable Kalman filter is stable. We also know from Theorem 25 that this DARE solution is positive definite. The DARE for this system is given by

$$\begin{aligned} P &= FPF^T - FPH^T(HPH^T + R)^{-1}HPF^T + Q \\ &= 4P - 4P(P+1)^{-1}P \end{aligned} \tag{7.57}$$

This has two solutions for P, one of which is 0 (i.e., positive semidefinite), and one of which is 3 (i.e., positive definite). If we use $P = 0$ in the steady-state Kalman filter we obtain

$$\begin{aligned} K &= 0 \\ \hat{x}_k^+ &= 2\hat{x}_{k-1}^+ \end{aligned} \tag{7.58}$$

We see that the resulting Kalman filter is unstable (the eigenvalue is 2). If we use $P = 3$ in the steady-state Kalman filter we obtain

$$\begin{aligned} K &= \frac{3}{4} \\ \hat{x}_k^+ &= \frac{1}{2}\hat{x}_{k-1}^+ \end{aligned} \tag{7.59}$$

We see that the resulting Kalman filter is stable (the eigenvalue is 1/2). In this example, we have multiple positive semidefinite solutions to the DARE, but only one results in a stable Kalman filter.

∇∇∇

7.3.1 α-β filtering

In this section, we derive the α-β filter [Bar01], also sometimes referred to as the f-g filter or the g-h filter [Bro98]. The α-β filter is a steady-state Kalman filter that is applied to a two-state Newtonian system with a position measurement. This is the type of estimation problem that commonly arises in tracking problems, and so it is well known and has been widely studied since before the invention of the Kalman filter.

Suppose we have a Newtonian dynamic system with only two states (position and velocity) and a noisy acceleration input, and we measure position plus noise. The system and measurement equations are then given as

$$\begin{aligned} x_k &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} w'_{k-1} \\ y_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k \end{aligned}$$

$$\begin{aligned} w'_k &\sim (0, \sigma_w^2) \\ v_k &\sim (0, R) \end{aligned} \tag{7.60}$$

where T is the sample time, and w'_k and v_k are uncorrelated white noise processes. The process equation can be written as

$$\begin{aligned} x_k &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_{k-1} + w_{k-1} \\ w_k &\sim (0, Q) \\ Q &= \begin{bmatrix} T^2/2 \\ T \end{bmatrix} E[w'_k w'^T_k] \begin{bmatrix} T^2/2 & T \end{bmatrix} \\ &= \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix} \sigma_w^2 \end{aligned} \tag{7.61}$$

A steady-state Kalman filter can be designed for this system from Equation (5.19), which is repeated here using steady-state notation:

$$\begin{aligned} P^- &= FP^+F^T + Q \\ K &= P^-H^T(HP^-H^T + R)^{-1} \\ \hat{x}_k^- &= F\hat{x}_{k-1}^+ \\ \hat{x}_k^+ &= \hat{x}_k^- + K(y_k - H_k\hat{x}_k^-) \\ P^+ &= (I - KH)P^- \end{aligned} \tag{7.62}$$

For this two-state, one-measurement problem, we see that K is a 2×1 matrix, and P^- and P^+ are 2×2 matrices. We will denote their steady-state values as

$$\begin{aligned} K &= \begin{bmatrix} K_1 & K_2 \end{bmatrix}^T \\ &= \begin{bmatrix} \alpha & \beta/T \end{bmatrix}^T \\ P^- &= \begin{bmatrix} P_{11}^- & P_{12}^- \\ P_{12}^- & P_{22}^- \end{bmatrix} \\ P^+ &= \begin{bmatrix} P_{11}^+ & P_{12}^+ \\ P_{12}^+ & P_{22}^+ \end{bmatrix} \end{aligned} \tag{7.63}$$

The parameters of the Kalman gain matrix K define the α and β parameters of the α-β filter. We can use Equation (7.62) to write

$$\begin{aligned} K &= \begin{bmatrix} P_{11}^- & P_{12}^- \\ P_{12}^- & P_{22}^- \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P_{11}^- & P_{12}^- \\ P_{12}^- & P_{22}^- \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + R \right)^{-1} \\ &= \frac{1}{P_{11}^- + R} \begin{bmatrix} P_{11}^- & P_{12}^- \end{bmatrix}^T \end{aligned} \tag{7.64}$$

The P^+ expression in Equation (7.62) can be written as

$$\begin{aligned} \begin{bmatrix} P_{11}^+ & P_{12}^+ \\ P_{12}^+ & P_{22}^+ \end{bmatrix} &= \left(I - \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \right) \begin{bmatrix} P_{11}^- & P_{12}^- \\ P_{12}^- & P_{22}^- \end{bmatrix} \\ &= \begin{bmatrix} (1-K_1)P_{11}^- & (1-K_1)P_{12}^- \\ (1-K_1)P_{12}^- & P_{22}^- - K_2P_{12}^- \end{bmatrix} \end{aligned} \tag{7.65}$$

The P^- expression in Equation (7.62) can be rewritten in terms of P^+ as follows:

$$\begin{aligned} P^+ &= F^{-1}(P^- - Q)F^{-T} \\ \begin{bmatrix} P_{11}^+ & P_{12}^+ \\ P_{12}^+ & P_{22}^+ \end{bmatrix} &= \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix} \left(\begin{bmatrix} P_{11}^- & P_{12}^- \\ P_{12}^- & P_{22}^- \end{bmatrix} - \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix} \sigma_w^2 \right) \times \\ & \begin{bmatrix} 1 & 0 \\ -T & 1 \end{bmatrix} \end{aligned} \tag{7.66}$$

Carrying out the multiplication gives the elements of P^+ as

$$\begin{aligned} P_{12}^+ &= P_{12}^- + \sigma_w^2 T^3/2 - P_{22}^- T \\ P_{11}^+ &= P_{11}^- + \sigma_w^2 T^4/4 - P_{12}^- T - P_{12}^+ T \\ P_{22}^+ &= P_{22}^- - \sigma_w^2 T^2 \end{aligned} \tag{7.67}$$

Equating the P_{ij}^+ elements in Equations (7.65) and (7.67) and performing a little algebra gives

$$\begin{aligned} K_1 P_{11}^- &= 2TP_{12}^- - T^2 P_{22}^- + T^4 \sigma_w^2/4 \\ K_1 P_{12}^- &= TP_{22}^- - T^3 \sigma_w^2/2 \\ K_2 P_{12}^- &= T^2 \sigma_w^2 \end{aligned} \tag{7.68}$$

These three equations, along with the expressions for K_1 and K_2 in the last line of Equation (7.64), can be solved for the five unknowns K_1, K_2, P_{11}^-, P_{12}^-, and P_{22}^-. After some algebra, this gives

$$\begin{aligned} K_1 &= -\frac{1}{8}\left(\lambda^2 + 8\lambda - (\lambda+4)\sqrt{\lambda^2+8\lambda}\right) \\ K_2 &= \frac{1}{4T}\left(\lambda^2 + 4\lambda - \lambda\sqrt{\lambda^2+8\lambda}\right) \\ P_{11}^- &= \frac{K_1 \sigma_w^2}{1-K_1} \\ P_{12}^- &= \frac{K_2 \sigma_w^2}{1-K_1} \\ P_{22}^- &= \left(\frac{K_1}{T} + \frac{K_2}{2}\right) P_{12}^- \end{aligned} \tag{7.69}$$

where λ is called the target maneuvering index or target tracking index [Kal84] and is defined as

$$\lambda = \frac{\sigma_w^2 T^2}{R} \tag{7.70}$$

Note that λ gives the ratio of the motion uncertainty to the measurement uncertainty. From these expressions and Equation (7.65) it can be shown that the elements of the steady-state *a posteriori* estimation-error covariance are given as

$$\begin{aligned} P_{11}^+ &= K_1 R \\ P_{12}^+ &= K_2 R \\ P_{22}^+ &= \left(\frac{K_1}{T} - \frac{K_2}{2}\right) P_{12}^- \end{aligned} \tag{7.71}$$

7.3.2 α-β-γ filtering

In this section, we present (without derivation) the α-β-γ filter [Bar01], also sometimes referred to as the *f-g-h* filter or the *g-h-k* filter [Bro98]. The α-β-γ filter is a steady-state Kalman filter that is applied to a three-state Newtonian system with a position measurement. This is very similar to the α-β filter presented in the previous section, except that the dynamic system model is one order higher in the α-β-γ filter.

Consider the three-state system given in Example 5.1. The states consist of position, velocity, and acceleration, the input consists of noisy acceleration, and the measurement consists of position plus noise. The system and measurement equations are given as

$$\begin{aligned} x_k &= \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} T^2/2 \\ T \\ 1 \end{bmatrix} w'_{k-1} \\ y_k &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x_k + v_k \\ w'_k &\sim (0, \sigma_w^2) \\ v_k &\sim (0, R) \end{aligned} \tag{7.72}$$

where T is the sample time, and w'_k and v_k are uncorrelated white noise processes. The process equation can be written as

$$\begin{aligned} x_k &= \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} x_{k-1} + w_{k-1} \\ w_k &\sim (0, Q) \\ Q &= \begin{bmatrix} T^2/2 \\ T \\ 1 \end{bmatrix} E[w'_k w'^T_k] \begin{bmatrix} T^2/2 & T & 1 \end{bmatrix} \\ &= \begin{bmatrix} T^4/4 & T^3/2 & T^2/2 \\ T^3/2 & T^2 & T \\ T^2/2 & T & 1 \end{bmatrix} \sigma_w^2 \end{aligned} \tag{7.73}$$

A steady-state Kalman filter can be designed for this system from Equation (5.19), in a similar way that the α-β filter was designed in the previous section. The steady-state values of the Kalman gain and *a posteriori* estimation-error covariance are denoted as

$$\begin{aligned} K &= \begin{bmatrix} K_1 & K_2 & K_3 \end{bmatrix}^T \\ &= \begin{bmatrix} \alpha & \beta/T & \gamma/2T^2 \end{bmatrix}^T \\ P^+ &= \begin{bmatrix} P_{11}^+ & P_{12}^+ & P_{13}^+ \\ P_{12}^+ & P_{22}^+ & P_{23}^+ \\ P_{13}^+ & P_{23}^+ & P_{33}^+ \end{bmatrix} \end{aligned} \tag{7.74}$$

The parameters of the Kalman gain matrix K define the α, β, and γ parameters of the α-β-γ filter. The solution can be computed as follows [Gra93]:

$$\begin{aligned}
\alpha &= 1 - s^2 \\
\beta &= 2(1-s)^2 \\
\gamma &= 2\lambda s
\end{aligned} \tag{7.75}$$

where λ is the target maneuvering index defined in Equation (7.70), and s is an auxiliary variable. The variable s is defined via auxiliary variables b, c, p, q, and z as follows.

$$\begin{aligned}
b &= \frac{\lambda}{2} - 3 \\
c &= \frac{\lambda}{2} + 3 \\
p &= c - \frac{b^2}{3} \\
q &= \frac{2b^3}{27} - \frac{bc}{3} - 1 \\
z &= \left[\frac{-q + \sqrt{q^2 + 4p^3/27}}{2}\right]^{1/3} \\
s &= z - \frac{p}{3z} - \frac{b}{3}
\end{aligned} \tag{7.76}$$

The steady-state *a posteriori* error covariance can be computed as

$$\begin{aligned}
P_{11}^{+} &= \alpha R \\
P_{12}^{+} &= \beta R/T \\
P_{13}^{+} &= \gamma R/2T^2 \\
P_{22}^{+} &= \frac{8\alpha\beta + \gamma(\beta - 2\alpha - 4)}{8T^2(1-\alpha)} R \\
P_{23}^{+} &= \frac{\beta(2\beta - \gamma)R}{4T^3(1-\alpha)} \\
P_{33}^{+} &= \frac{\gamma(2\beta - \gamma)R}{4T^4(1-\alpha)}
\end{aligned} \tag{7.77}$$

The general idea of the α-β and α-β-γ filters date back to the 1940s [Mec49, Skl57, Ben62], before the advent of Kalman filtering, although, of course, the optimal α-β-γ values were not known at that time. Further discussion of these filters and related issues can be found in [Bro98, Bar01]. A steady-state Kalman filter that is applied to a one-state Newtonian system with a position measurement is called an α filter [Sio96].

7.3.3 A Hamiltonian approach to steady-state filtering

In this section, we present an alternative method for obtaining the steady-state Kalman filter. We will assume in this section that the correlation M between the process noise and measurement noise is zero so that we can simplify notation. The

a priori Riccati equation of Equation (7.41) can then be written as

$$P_{k+1} = FP_kF^T - FP_kH^T(HP_kH^T + R)^{-1}HP_kF^T + Q \tag{7.78}$$

where we have dropped the minus superscript for ease of notation. We can use the matrix inversion lemma of Equation (1.39) to write

$$(HP_kH^T + R)^{-1} = R^{-1} - R^{-1}H(H^TR^{-1}H + P_k^{-1})^{-1}H^TR^{-1} \tag{7.79}$$

Substituting this into Equation (7.78) gives

$$\begin{aligned} P_{k+1} &= FP_kF^T - FP_kH^TR^{-1}HP_kF^T + \\ &\quad FP_kH^TR^{-1}H(H^TR^{-1}H + P_k^{-1})^{-1}H^TR^{-1}HP_kF^T + Q \end{aligned} \tag{7.80}$$

Factoring out F and F^T from the beginning and end of the first three terms on the right side gives

$$\begin{aligned} P_{k+1} &= F\left\{P_k - P_kH^TR^{-1}HP_k + \right. \\ &\quad \left. P_kH^TR^{-1}H(H^TR^{-1}H + P_k^{-1})^{-1}H^TR^{-1}HP_k\right\}F^T + Q \\ &= F\left\{P_k - P_kH^TR^{-1}H\left[P_k - (H^TR^{-1}H + P_k^{-1})^{-1}H^TR^{-1}HP_k\right]\right\}F^T + Q \\ &= F\left\{P_k - P_kH^TR^{-1}H(H^TR^{-1}H + P_k^{-1})^{-1} \times \right. \\ &\quad \left.\left[(H^TR^{-1}H + P_k^{-1})P_k - H^TR^{-1}HP_k\right]\right\}F^T + Q \\ &= F\left\{P_k - P_kH^TR^{-1}H(H^TR^{-1}H + P_k^{-1})^{-1}\right\}F^T + Q \\ &= FP_k\left[(H^TR^{-1}H + P_k^{-1}) - H^TR^{-1}H\right](H^TR^{-1}H + P_k^{-1})^{-1}F^T + Q \\ &= F(H^TR^{-1}H + P_k^{-1})^{-1}F^T + Q \\ &= FP_k(H^TR^{-1}HP_k + I)^{-1}F^T + QF^{-T}F^T \\ &= \left[FP_k + QF^{-T}(H^TR^{-1}HP_k + I)\right](H^TR^{-1}HP_k + I)^{-1}F^T \\ &= \left[(F + QF^{-T}H^TR^{-1}H)P_k + QF^{-T}\right](H^TR^{-1}HP_k + I)^{-1}F^T \end{aligned} \tag{7.81}$$

Now suppose that P_k can be factored as

$$P_k = S_kZ_k^{-1} \tag{7.82}$$

where S_k and Z_k both have the same dimensions as P_k. Making this substitution in Equation (7.81) gives

$$\begin{aligned} P_{k+1} &= \left[(F + QF^{-T}H^TR^{-1}H)S_k + QF^{-T}Z_k\right]Z_k^{-1}(H^TR^{-1}HS_kZ_k^{-1} + I)^{-1}F^T \\ &= \left[(F + QF^{-T}H^TR^{-1}H)S_k + QF^{-T}Z_k\right](H^TR^{-1}HS_k + Z_k)^{-1}F^T \\ &= \left[(F + QF^{-T}H^TR^{-1}H)S_k + QF^{-T}Z_k\right](F^{-T}H^TR^{-1}HS_k + F^{-T}Z_k)^{-1} \\ &= S_{k+1}Z_{k+1}^{-1} \end{aligned} \tag{7.83}$$

This shows that

$$\begin{aligned} S_{k+1} &= (F + QF^{-T}H^TR^{-1}H)S_k + QF^{-T}Z_k \\ Z_{k+1} &= F^{-T}H^TR^{-1}HS_k + F^{-T}Z_k \end{aligned} \tag{7.84}$$

These equations for S_{k+1} and Z_{k+1} can be written as the following single equation:

$$\begin{aligned}\begin{bmatrix} Z_{k+1} \\ S_{k+1} \end{bmatrix} &= \begin{bmatrix} F^{-T} & F^{-T}H^TR^{-1}H \\ QF^{-T} & F+QF^{-T}H^TR^{-1}H \end{bmatrix} \begin{bmatrix} Z_k \\ S_k \end{bmatrix} \\ &= \mathcal{H} \begin{bmatrix} Z_k \\ S_k \end{bmatrix} \end{aligned} \tag{7.85}$$

If the covariance matrix P is an $n \times n$ matrix, then $\mathcal{H}$ will be a $2n \times 2n$ matrix. The matrix $\mathcal{H}$ on the right side of the above equation is called a Hamiltonian matrix and has some interesting properties. It is a symplectic matrix; that is, it satisfies the equation

$$J^{-1}\mathcal{H}^T J = \mathcal{H}^{-1} \qquad \text{where } J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \tag{7.86}$$

Symplectic matrices have the following properties (see Problem 7.7).

- None of the eigenvalues of a symplectic matrix are equal to 0.
- If λ is an eigenvalue of a symplectic matrix, then so is $1/\lambda$.
- The determinant of a symplectic matrix is equal to ± 1.

If a symplectic matrix does not have any eigenvalues with magnitude equal to one, then half of its eigenvalues will be outside the unit circle, and the other half will be inside the unit circle. Let us define Λ as the diagonal matrix that contains all of the eigenvalues of $\mathcal{H}$ that are outside the unit circle (assuming that none of the eigenvalues are on the unit circle). Then the Jordan form of $\mathcal{H}$ can be written as

$$\begin{aligned}\mathcal{H} &= \Psi \begin{bmatrix} \Lambda^{-1} & 0 \\ 0 & \Lambda \end{bmatrix} \Psi^{-1} \\ &= \Psi D \Psi^{-1} \end{aligned} \tag{7.87}$$

where the D matrix is the diagonal matrix of eigenvalues, and is defined by the above equation. The Ψ matrix can be partitioned into four $n \times n$ blocks as

$$\Psi = \begin{bmatrix} \Psi_{11} & \Psi_{12} \\ \Psi_{21} & \Psi_{22} \end{bmatrix} \tag{7.88}$$

Note that the $2n \times n$ matrix $\begin{bmatrix} \Psi_{11} \\ \Psi_{21} \end{bmatrix}$ contains the eigenvectors of $\mathcal{H}$ that correspond to the stable eigenvalues of $\mathcal{H}$ (i.e., the eigenvalues that are inside the unit circle). The $2n \times n$ matrix $\begin{bmatrix} \Psi_{12} \\ \Psi_{22} \end{bmatrix}$ contain the eigenvectors of $\mathcal{H}$ that correspond to the unstable eigenvalues of $\mathcal{H}$ (i.e., the eigenvalues that are outside the unit circle). Equation (7.85) can be written as

$$\begin{aligned}\begin{bmatrix} Z_{k+1} \\ S_{k+1} \end{bmatrix} &= \Psi D \Psi^{-1} \begin{bmatrix} Z_k \\ S_k \end{bmatrix} \\ \Psi^{-1} \begin{bmatrix} Z_{k+1} \\ S_{k+1} \end{bmatrix} &= D \Psi^{-1} \begin{bmatrix} Z_k \\ S_k \end{bmatrix} \end{aligned} \tag{7.89}$$

Now define the $n \times n$ matrices Y_{1k} and Y_{2k}, and the $2n \times n$ matrix Y_k, as follows:

$$\begin{aligned} \begin{bmatrix} Y_{1k} \\ Y_{2k} \end{bmatrix} &= \Psi^{-1} \begin{bmatrix} Z_k \\ S_k \end{bmatrix} \\ &= \begin{bmatrix} (\Psi^{-1})_{11} & (\Psi^{-1})_{12} \\ (\Psi^{-1})_{21} & (\Psi^{-1})_{22} \end{bmatrix} \begin{bmatrix} Z_k \\ S_k \end{bmatrix} \\ &= Y_k \end{aligned} \tag{7.90}$$

Note in the above equation that $(\Psi^{-1})_{11}$ is *not* the inverse of the upper left $n \times n$ partition of Ψ; the matrix $(\Psi^{-1})_{11}$ is rather the upper left $n \times n$ partition of Ψ^{-1}. (Similar statements hold for the other partitions.) With these definitions we can write Equation (7.89) as

$$\begin{aligned} Y_{k+1} &= DY_k \\ \begin{bmatrix} Y_{1,k+1} \\ Y_{2,k+1} \end{bmatrix} &= \begin{bmatrix} \Lambda^{-1} & 0 \\ 0 & \Lambda \end{bmatrix} \begin{bmatrix} Y_{1k} \\ Y_{2k} \end{bmatrix} \end{aligned} \tag{7.91}$$

From this equation we see that

$$\begin{aligned} Y_{2,k+1} &= \Lambda Y_{2k} \\ Y_{2k} &= \Lambda^k Y_{2,0} \end{aligned} \tag{7.92}$$

Similarly we see that

$$Y_{1k} = \Lambda^{-k} Y_{1,0} \tag{7.93}$$

Now note that Equation (7.90) can be written as

$$\begin{aligned} \begin{bmatrix} Z_k \\ S_k \end{bmatrix} &= \begin{bmatrix} \Psi_{11} & \Psi_{12} \\ \Psi_{21} & \Psi_{22} \end{bmatrix} \begin{bmatrix} Y_{1k} \\ Y_{2k} \end{bmatrix} \\ &= \begin{bmatrix} \Psi_{11} & \Psi_{12} \\ \Psi_{21} & \Psi_{22} \end{bmatrix} \begin{bmatrix} \Lambda^{-k} Y_{1,0} \\ \Lambda^k Y_{2,0} \end{bmatrix} \end{aligned} \tag{7.94}$$

As k increases, the Λ^{-k} matrix approaches zero (because it is a diagonal matrix whose elements are all less than one in magnitude). Therefore, for large k we obtain

$$\begin{aligned} \begin{bmatrix} Z_k \\ S_k \end{bmatrix} &= \begin{bmatrix} \Psi_{11} & \Psi_{12} \\ \Psi_{21} & \Psi_{22} \end{bmatrix} \begin{bmatrix} 0 \\ \Lambda^k Y_{2,0} \end{bmatrix} \\ Z_k &= \Psi_{12} Y_{2k} \\ S_k &= \Psi_{22} Y_{2k} \end{aligned} \tag{7.95}$$

Solving for S_k for large values of k gives

$$S_k = \Psi_{22} \Psi_{12}^{-1} Z_k \tag{7.96}$$

But we also know from Equation (7.82) that

$$S_k = P_k Z_k \tag{7.97}$$

Combining the two previous equations shows that

$$\lim_{k \to \infty} P_k = \Psi_{22} \Psi_{12}^{-1} \tag{7.98}$$

This gives us a way to determine the steady-state solution of the Riccati equation solution. However, this analysis assumed that Λ was a diagonal matrix with all elements outside the unit circle. In other words, if the Hamiltonian matrix has any eigenvalues with magnitude equal to one, then this analysis falls apart. This gives the following algorithm for computing the steady-state, discrete-time Riccati equation solution.

The Hamiltonian approach to steady-state Kalman filtering

1. Form the Hamiltonian matrix
$$\mathcal{H} = \begin{bmatrix} F^{-T} & F^{-T}H^TR^{-1}H \\ QF^{-T} & F + QF^{-T}H^TR^{-1}H \end{bmatrix} \tag{7.99}$$
For an n-state Kalman filtering problem, the Hamiltonian matrix will be a $2n \times 2n$ matrix.

2. Compute the eigenvalues of $\mathcal{H}$. If any of them are on the unit circle, then we cannot go any further with this procedure; the Riccati equation does not have a steady-state solution.

3. Collect the n eigenvectors that correspond to the n eigenvalues that are outside the unit circle. Put these n eigenvectors in a matrix partitioned as
$$\begin{bmatrix} \Psi_{12} \\ \Psi_{22} \end{bmatrix} \tag{7.100}$$
The first column of this matrix is the first eigenvector, the second column is the second eigenvector, etc. Ψ_{12} and Ψ_{22} are both $n \times n$ matrices.

4. Compute the steady-state Riccati equation solution as
$$P_\infty^- = \Psi_{22}\Psi_{12}^{-1} \tag{7.101}$$
Note that Ψ_{12} must be invertible for this method to work.

The Hamiltonian approach to steady-state filtering is due to [Vau70], which also derives time-varying DARE solutions using Hamiltonian matrices.

■ **EXAMPLE 7.9**

Consider the scalar system of Equation (7.36):

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + v_k \\ w_k &\sim N(0,1) \\ v_k &\sim N(0,1) \end{aligned} \tag{7.102}$$

We see that $F = H = Q = R = 1$. Substituting these values into the expression for the Hamiltonian matrix gives

$$\begin{aligned} \mathcal{H} &= \begin{bmatrix} F^{-T} & F^{-T}H^TR^{-1}H \\ QF^{-T} & F + QF^{-T}H^TR^{-1}H \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \end{aligned} \tag{7.103}$$

The eigenvalues of $\mathcal{H}$ are 0.38 and 2.62. None of the eigenvalues has a magnitude of one so we are able to continue with the procedure. The eigenvector of $\mathcal{H}$ that corresponds to the eigenvalue outside the unit circle is $\begin{bmatrix} 0.5257 & 0.8507 \end{bmatrix}^T$. We form the corresponding eigenvector matrix as

$$\begin{bmatrix} \Psi_{12} \\ \Psi_{22} \end{bmatrix} = \begin{bmatrix} 0.5257 \\ 0.8507 \end{bmatrix} \tag{7.104}$$

Note that Ψ_{12} is invertible so we are able to continue with the problem. The steady-state Riccati equation solution is

$$\begin{aligned} P &= \Psi_{22}\Psi_{12}^{-1} \\ &= \frac{0.8507}{0.5257} \\ &= 1.62 \end{aligned} \tag{7.105}$$

The steady-state Kalman gain is therefore computed from Equation (7.14) as

$$\begin{aligned} K &= PH^T(HPH^T + R)^{-1} \\ &= \frac{(1.62)(1)}{(1)(1.62)(1) + 1} \\ &= 0.62 \end{aligned} \tag{7.106}$$

which is in agreement with Equation (7.37).

$\nabla\nabla\nabla$

7.4 KALMAN FILTERING WITH FADING MEMORY

In Section 5.5, we discussed the problem of filter divergence due to mismodeling. That is, if our system model does not match reality, then the Kalman filter estimate may diverge from the true state. Example 5.3 showed how the addition of fictitious process noise can compensate for mismodeling. In this section, we show how to accomplish the same thing with the fading-memory filter. Recall our linear discrete-time system model:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \\ E[w_k w_j^T] &= Q_k \delta_{k-j} \\ E[v_k v_j^T] &= R_k \delta_{k-j} \\ E[w_k v_j^T] &= 0 \end{aligned} \tag{7.107}$$

The Kalman filter finds the sequence of estimates $\{\hat{x}_1^-, \cdots, \hat{x}_N^-\}$ that minimizes $E(J_N)$, where J_N is given as

$$J_N = \sum_{k=1}^{N} \left[(y_k - H_k\hat{x}_k^-)^T R_k^{-1} (y_k - H_k\hat{x}_k^-) + \hat{w}_k^T Q_k^{-1} \hat{w}_k \right] \tag{7.108}$$

Note that $\hat{x}$ determines $\hat{w}$ through the system equation, and vice versa. This expression for J_N shows how we could give greater emphasis to more recent data. Instead of finding the filter that minimizes $E(J_N)$, we can find the filter that minimizes $E(\tilde{J}_N)$, where $\tilde{J}_N$ is given as

$$\tilde{J}_N = \sum_{k=1}^{N} \left[(y_k - H_k\hat{x}_k^-)^T \alpha^{2k} R_k^{-1} (y_k - H_k\hat{x}_k^-) + \hat{w}_k^T \alpha^{2k+2} Q_k^{-1} \hat{w}_k\right] \tag{7.109}$$

where $\alpha \geq 1$. The α term in the first part of the cost function means that we are more interested in minimizing the weighted covariance of the residual at recent times (large values of k) than at times in the distant past (small values of k). This will force the filter to converge to state estimates that discount old measurements and give greater emphasis to more recent measurements. The α term in the second part of the cost function is added for mathematical tractability, as we will see in the subsequent development. The second part of the cost function is constant as far as our minimization problem is concerned.

The solution to the minimization of $E(\tilde{J}_N)$ is equivalent to the minimization of $E(J_N)$ (which is the Kalman filter), except that R_k is replaced with $\alpha^{-2k}R_k$ and Q_k is replaced with $\alpha^{-2k-2}Q_k$. The modified Kalman gain can therefore be written as

$$\begin{aligned} K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + \alpha^{-2k} R_k)^{-1} \\ &= \alpha^{2k} P_k^- H_k^T (H_k \alpha^{2k} P_k^- H_k^T + R_k)^{-1} \end{aligned} \tag{7.110}$$

The time update for the estimation-error covariance can be written as

$$\begin{aligned} P_k^- &= F_{k-1} P_{k-1}^+ F_{k-1}^T + \alpha^{-2k+2} Q_{k-1}/\alpha^2 \\ \alpha^{2k} P_k^- &= F_{k-1} \alpha^{2k} P_{k-1}^+ F_{k-1}^T + Q_{k-1} \\ &= \alpha^2 F_{k-1} \alpha^{2(k-1)} P_{k-1}^+ F_{k-1}^T + Q_{k-1} \end{aligned} \tag{7.111}$$

The measurement update for the estimation-error covariance can be written as

$$\begin{aligned} P_k^+ &= P_k^- - K_k H_k P_k^- \\ \alpha^{2k} P_k^+ &= \alpha^{2k} P_k^- - K_k H_k \alpha^{2k} P_k^- \end{aligned} \tag{7.112}$$

Now we define $\tilde{P}_k^+$ and $\tilde{P}_k^-$ as

$$\begin{aligned} \tilde{P}_k^+ &= \alpha^{2k} P_k^+ \\ \tilde{P}_k^- &= \alpha^{2k} P_k^- \end{aligned} \tag{7.113}$$

We can then write Equations (7.110), (7.111), and (7.112) as

$$\begin{aligned} K_k &= \tilde{P}_k^- H_k^T (H_k \tilde{P}_k^- H_k^T + R_k)^{-1} \\ \tilde{P}_k^- &= \alpha^2 F_{k-1} \tilde{P}_{k-1}^+ F_{k-1}^T + Q_{k-1} \\ \tilde{P}_k^+ &= \tilde{P}_k^- - K_k H_k \tilde{P}_k^- \end{aligned} \tag{7.114}$$

These are the new Kalman gain equation and covariance-update equations. The state-update equations remain as before:

$$\begin{aligned} \hat{x}_k^- &= F_{k-1} \hat{x}_{k-1}^+ + G_{k-1} u_{k-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k (y_k - H_k \hat{x}_k^-) \end{aligned} \tag{7.115}$$

We see that the fading-memory filter is identical to the standard Kalman filter, with the exception that the time-update equation for the computation of the *a priori* estimation-error covariance has an α^2 factor in its first term. This serves to increase the uncertainty in the state estimate, which results in the filter giving more credence to the measurement. This is equivalent to increasing the process noise, which also results in the filter giving relatively more credence to the measurement. This strategy, along with other solutions to the filter divergence problem, was suggested early in the history of the Kalman filter [Sch67, Sor71a]. The fading-memory filter can be summarized as follows.

The fading-memory filter

1. The dynamic system is given by the following equations:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \\ E(w_k w_j^T) &= Q_k \delta_{k-j} \\ E(v_k v_j^T) &= R_k \delta_{k-j} \\ E(w_k v_j^T) &= 0 \end{aligned} \tag{7.116}$$

2. The Kalman filter is initialized as follows:

$$\begin{aligned} \hat{x}_0^+ &= E(x_0) \\ \tilde{P}_0^+ &= E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T] \end{aligned} \tag{7.117}$$

3. Choose $\alpha \geq 1$ based on how much you want the filter to forget past measurements. If $\alpha = 1$ then the fading-memory filter is equivalent to the standard Kalman filter. In most applications, α is only slightly greater than 1 (for example, $\alpha \approx 1.01$).

4. The fading-memory filter is given by the following equations, which are computed for each time step $k = 1, 2, \cdots$:

$$\begin{aligned} \tilde{P}_k^- &= \alpha^2 F_{k-1}\tilde{P}_{k-1}^+ F_{k-1}^T + Q_{k-1} \\ K_k &= \tilde{P}_k^- H_k^T (H_k \tilde{P}_k^- H_k^T + R_k)^{-1} \\ &= \tilde{P}_k^+ H_k^T R_k^{-1} \\ \hat{x}_k^- &= F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \\ \tilde{P}_k^+ &= (I - K_k H_k)\tilde{P}_k^-(I - K_k H_k)^T + K_k R_k K_k^T \\ &= \left[(\tilde{P}_k^-)^{-1} + H_k^T R_k^{-1} H_k\right]^{-1} \\ &= \tilde{P}_k^- - K_k H_k \tilde{P}_k^- \end{aligned} \tag{7.118}$$

Note that $\tilde{P}$ is *not* equal to the covariance of the estimation error. However, the fading-memory filter is more robust to modeling errors than the standard Kalman filter.

■ **EXAMPLE 7.10**

In this example, we will show how the fading-memory filter makes the Kalman filter more responsive to measurements when the process noise is zero. Consider the following scalar system:

$$\begin{aligned} x_k &= x_{k-1} \\ y_k &= x_k + v_k \\ v_k &\sim (0, R) \end{aligned} \tag{7.119}$$

In other words, we are trying to estimate a constant on the basis of noisy measurements of that constant. Applying the fading-memory filter equations given in Equation (7.118) to this problem, we see that

$$\begin{aligned} P_k^- &= \alpha^2 P_{k-1}^+ \\ K_k &= \frac{P_k^-}{P_k^- + R} \\ &= \frac{\alpha^2 P_{k-1}^+}{\alpha^2 P_{k-1}^+ + R} \\ P_k^+ &= P_k^- - K_k H_k P_k^- \\ &= \alpha^2 P_{k-1}^+ - \left(\frac{\alpha^2 P_{k-1}^+}{\alpha^2 P_{k-1}^+ + R} \right) \alpha^2 P_{k-1}^+ \end{aligned} \tag{7.120}$$

As the filter approaches steady state, P_k^+ approaches a steady-state value that can be obtained from the above equation as

$$P_\infty^+ = \alpha^2 P_\infty^+ - \left(\frac{\alpha^2 P_\infty^+}{\alpha^2 P_\infty^+ + R} \right) \alpha^2 P_\infty^+ \tag{7.121}$$

This can be solved for P_∞^+ as

$$P_\infty^+ = \frac{(\alpha^2 - 1)R}{\alpha^2} \tag{7.122}$$

The steady-state Kalman gain K_∞ can then be solved as

$$\begin{aligned} K_\infty &= \frac{\alpha^2 P_\infty^+}{\alpha^2 P_\infty^+ + R} \\ &= \frac{\alpha^2 - 1}{\alpha^2} \end{aligned} \tag{7.123}$$

We see that if $\alpha = 1$ (i.e., if we use the standard Kalman filter) then $P_\infty^+ = K_\infty = 0$. However, if $\alpha > 1$ (i.e., if we use the fading-memory Kalman filter) then P_∞^+ and K_∞ will both be greater than zero. The measurement update equation for the state is given as

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - \hat{x}_k^-) \tag{7.124}$$

For the standard Kalman filter, $\lim_{k\to\infty} K_k = 0$, which means that new measurements will be ignored and will not be used to update the state estimate.

The Kalman filter may have a false confidence in the certainty of its state estimate. However, for the fading-memory filter, $K_k > 0$ for all k, and the filter will always be responsive to new measurements. A larger value of α will make the filter more responsive to new measurements. In the limit as $\alpha \to \infty$, we see from Equation (7.123) that $K_\infty = 1$. This will result in a measurement update from Equation (7.124) of

$$\begin{aligned} \hat{x}_k^+ &= \hat{x}_k^- + (1)(y_k - \hat{x}_k^-) \\ &= y_k \end{aligned} \tag{7.125}$$

In other words, the fading-memory filter, when carried to an extreme, ignores the system model and estimates the state solely on the basis of the measurements. This is the same thing that will happen if the process noise is extremely large. The Kalman filter will ignore the system model because we are telling it that we do not have any confidence in the system model.

$\nabla\nabla\nabla$

7.5 CONSTRAINED KALMAN FILTERING

In the application of state estimators, there is often known information that does not fit into the Kalman filter equations in an obvious way. For example, suppose that we know (on the basis of physical considerations) that the states satisfy some equality constraint $Dx = d$, or some inequality constraint $Dx \leq d$, where D is a known matrix and d is a known vector. This section discusses some ways of incorporating those constraints into the Kalman filter equations.

Some researchers have treated state equality constraints by reducing the system model parameterization [Wen92], and this will be discussed in Section 7.5.1. Others have handled state equality constraints by treating them as perfect measurements [Por88, Hay98], and this will be discussed in Section 7.5.2. A third approach is to incorporate the state constraints into the derivation of the Kalman filter [Chi85, Sim02], and this will presented in Section 7.5.3. A final approach is to incorporate the constraints by discarding that portion of the pdf of the state estimate that violates the constraints [Shi98, Sim06b], and this will be discussed in Section 7.5.4.

7.5.1 Model reduction

Some researchers have treated state equality constraints by reducing the system model parameterization [Wen92]. This is straightforward but there are some disadvantages with this approach. First, it may be desirable to maintain the form and structure of the state equations due to the physical meaning associated with each state. The reduction of the state equations makes their interpretation less natural and more difficult. Second, equality constraints that are formulated this way cannot be extended to inequality constraints. On the other hand, the model reduction approach is conceptually straightforward and usually can be easily implemented.

As an example of the model reduction approach, consider the system

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 4 & -2 & 2 \end{bmatrix} x_k + \begin{bmatrix} w_{k1} \\ w_{k2} \\ w_{k3} \end{bmatrix} \\ y_k &= \begin{bmatrix} 2 & 4 & 5 \end{bmatrix} x_k + v_k \end{aligned} \tag{7.126}$$

Now suppose that we also know, on the basis of our understanding of the physics underlying the problem, that the following constraint is always satisfied between the states:

$$\begin{bmatrix} 1 & 0 & 1 \end{bmatrix} x_k = 0 \tag{7.127}$$

This means that $x_k(3) = -x_k(1)$. If we make this substitution for $x_k(3)$ in the original state and measurement equations, we obtain

$$\begin{aligned} x_{k+1}(1) &= x_k(1) + 2x_k(2) - 3x_k(1) \\ &= -2x_k(1) + 2x_k(2) \\ x_{k+1}(2) &= 3x_k(1) + 2x_k(2) - x_k(1) \\ &= 2x_k(1) + 2x_k(2) \\ y_k &= 2x_k(1) + 4x_k(2) - 5x_k(1) + v_k \\ &= -3x_k(1) + 4x_k(2) + v_k \end{aligned} \tag{7.128}$$

These equations can be written in matrix form as

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} -2 & 2 \\ 2 & 2 \end{bmatrix} x_k + \begin{bmatrix} w_{k1} \\ w_{k2} \end{bmatrix} \\ y_k &= \begin{bmatrix} -3 & 4 \end{bmatrix} x_k + v_k \end{aligned} \tag{7.129}$$

We have reduced the filtering problem with equality constraints to an equivalent but unconstrained filtering problem. An advantage of this approach is that the dimension of the problem has been reduced, and so the computational effort of the problem is less. One disadvantage of this approach is that the physical meaning of the state variables has been lost. Also, this approach can only be used for equality constraints (i.e., constraints of the form $Dx = d$) and cannot be used for inequality constraints (i.e., constraints of the form $Dx \leq d$).

7.5.2 Perfect measurements

Some researchers treat state constraints as perfect measurements (i.e., no measurement noise) [Por88, Hay98]. Suppose that our constraints are given as $Dx_k = d$, where D is a known $s \times n$ matrix ($s < n$), and d is a known vector. We can solve the constrained Kalman filtering problem by augmenting the measurement equation with s perfect measurements of the state:

$$\begin{aligned} x_{k+1} &= F_k x_k + w_k \\ \begin{bmatrix} y_k \\ d \end{bmatrix} &= \begin{bmatrix} H_k \\ D \end{bmatrix} x_k + \begin{bmatrix} v_k \\ 0 \end{bmatrix} \end{aligned} \tag{7.130}$$

The state equation is the same as usual, but the measurement equation has been augmented. The fact that the last s elements of the measurement equation are

noise free means that the Kalman filter estimate of the state will always be consistent with these s measurements; that is, the Kalman filter estimate will always be consistent with the constraint $D\hat{x}_k^+ = d$. Note that the new measurement noise covariance will be singular – the last s rows and the last s columns of the measurement noise covariance will be zero. A singular covariance matrix does not present any theoretical problems [Gee97]. In fact, Kalman's original paper [Kal60] presents an example that uses perfect measurements. However, in practice a singular covariance increases the possibility of numerical problems [May79, p. 249], [Ste94, p. 365]. In addition, the use of perfect measurements is directly applicable only to equality constraints. It can be extended to inequality constraints by adding small nonzero measurement noise to the "perfect" measurements, but then the constraints will be soft [Mah04a] and it will be difficult to control how close the state estimate gets to the constraint boundary.

7.5.3 Projection approaches

Another approach to constrained filtering is to incorporate the state constraints into the derivation of the Kalman filter [Chi85, Sim02]. We can incorporate the constraints into a maximum probability derivation of the filter, or a mean square derivation of the Kalman filter. Also, we can simply project the standard Kalman filter estimate onto the constraint surface.

7.5.3.1 Maximum probability approach Assuming that x_0, w_k, and v_k are Gaussian, the Kalman filter solves the problem

$$\hat{x}_k = \text{argmax}_{x_k} \text{pdf}(x_k|Y_k) \tag{7.131}$$

That is, $\hat{x}_k$ is the value of x_k that maximizes $\text{pdf}(x_k|Y_k)$. In the above equation, Y_k is the vector of measurements up to and including time k; that is, $Y_k = \begin{bmatrix} y_1^T & \cdots & y_k^T \end{bmatrix}^T$. This interpretation of the Kalman filter looks at x_k as a random variable with a pdf that is conditioned on the measurements up to and including time k. The Kalman filter estimate is that value of x_k that maximizes its conditional pdf. If the noise processes are Gaussian, then

$$\text{pdf}(x_k|Y_k) = \frac{\exp[-(x_k - \bar{\bar{x}}_k)^T P_k^{-1}(x_k - \bar{\bar{x}}_k)/2]}{(2\pi)^{n/2}|P_k|^{1/2}} \tag{7.132}$$

where n is the dimension of the state, P_k is the covariance of the state estimate, and $\bar{\bar{x}}_k$ is defined as the mean of x_k conditioned on the measurements Y_k:

$$\bar{\bar{x}}_k = E(x_k|Y_k) \tag{7.133}$$

To maximize $\text{pdf}(x_k|Y_k)$, we can maximize $\ln \text{pdf}(x_k|Y_k)$, which means minimizing $(x_k - \bar{\bar{x}}_k)^T P_k^{-1}(x_k - \bar{\bar{x}}_k)$. Now suppose that we have the additional constraint that $Dx_k = d$. The solution of this constrained minimization problem is the constrained state estimate $\tilde{x}$. That is,

$$\tilde{x}_k = \text{argmin}_{\tilde{x}_k}(\tilde{x}_k - \bar{\bar{x}}_k)^T P_k^{-1}(\tilde{x}_k - \bar{\bar{x}}_k) \text{ such that } D\tilde{x}_k = d \tag{7.134}$$

Constrained optimization problems can be solved using the Lagrange multiplier method discussed in Section 11.2 [Ste94, Moo00]. We form the Lagrangian L and

find the necessary conditions for a minimum as follows:

$$\begin{aligned} L &= (\tilde{x}_k - \bar{\tilde{x}}_k)^T P_k^{-1}(\tilde{x}_k - \bar{\tilde{x}}_k) + 2\lambda^T(D\tilde{x}_k - d) \\ \frac{\partial L}{\partial \tilde{x}} &= P_k^{-1}(\tilde{x}_k - \bar{\tilde{x}}_k) + D^T\lambda = 0 \\ \frac{\partial L}{\partial \lambda} &= D\tilde{x}_k - d = 0 \end{aligned} \tag{7.135}$$

where λ is the n-element Lagrange multiplier. Solving these equations gives

$$\begin{aligned} \lambda &= (DP_kD^T)^{-1}(D\bar{\tilde{x}}_k - d) \\ &= (DP_kD^T)^{-1}(D\hat{x}_k - d) \\ \tilde{x}_k &= \bar{\tilde{x}}_k - P_kD^T\lambda \\ &= \hat{x}_k - P_kD^T(DP_kD^T)^{-1}(D\hat{x}_k - d) \end{aligned} \tag{7.136}$$

We see that the constrained state estimate $\tilde{x}$ is equal to the unconstrained state estimate $\hat{x}$, minus a correction term.

7.5.3.2 Least squares approach Another way to solve the constrained Kalman filtering problem is to approach the problem from a least squares point of view. In this approach, we find the constrained state estimate $\tilde{x}$ as

$$\tilde{x} = \text{argmin}_{\tilde{x}} E(||x - \tilde{x}||^2|Y) \text{ such that } D\tilde{x} = d \tag{7.137}$$

where we have dropped the time subscripts for ease of notation. This interpretation of the Kalman filter looks at x as a random variable. The quantity $(x - \tilde{x})$ (for any constant $\tilde{x}$) is also a random variable. The conditional expected value can be written as

$$\begin{aligned} E(||x - \tilde{x}||^2|Y) &= \int (x - \tilde{x})^T(x - \tilde{x})\,\text{pdf}(x|Y)\,dx \\ &= \int x^Tx\,\text{pdf}(x|Y)\,dx - 2\tilde{x}\int x\,\text{pdf}(x|Y)\,dx + \tilde{x}^T\tilde{x} \end{aligned} \tag{7.138}$$

We form the Lagrangian for the constrained optimization problem as

$$\begin{aligned} L &= E(||x - \tilde{x}||^2|Y) + 2\lambda^T(D\tilde{x} - d) \\ &= \int x^Tx\,\text{pdf}(x|Y)\,dx - 2\tilde{x}\int x\,\text{pdf}(x|Y)\,dx + \tilde{x}^T\tilde{x} + \\ &\quad 2\lambda^T(D\tilde{x} - d) \end{aligned} \tag{7.139}$$

Assuming that x_0, w_k, and v_k are Gaussian, the standard Kalman filter estimate $\hat{x}$ is given by

$$\begin{aligned} \hat{x} &= E(x|Y) \\ &= \int x\,\text{pdf}(x|Y)\,dx \end{aligned} \tag{7.140}$$

Solving the constrained minimization problem involves setting the partial derivatives of the Lagrangian of Equation (7.139) equal to zero. This gives the equations

$$\begin{aligned} \frac{\partial L}{\partial \tilde{x}} &= -2\hat{x} + 2\tilde{x} + 2D^T\lambda = 0 \\ \frac{\partial L}{\partial \lambda} &= D\tilde{x} - d = 0 \end{aligned} \tag{7.141}$$

Solving these equations for λ and $\tilde{x}$ gives

$$\begin{aligned} \lambda &= (DD^T)^{-1}(D\hat{x} - d) \\ \tilde{x} &= \hat{x} - D^T(DD^T)^{-1}(D\hat{x} - d) \end{aligned} \tag{7.142}$$

We see that the constrained state estimate $\tilde{x}$ is equal to the unconstrained state estimate $\hat{x}$, minus a correction term. This is similar to the constrained estimate that was obtained by the maximum probability approach in Equation (7.136).

7.5.3.3 General projection approach A third way to derive the constrained state estimate is to begin with the standard unconstrained estimate $\hat{x}$ and project it onto the constraint surface $Dx = d$. This can be written as

$$\tilde{x} = \text{argmin}_{\tilde{x}}(\tilde{x} - \hat{x})^T W(\tilde{x} - \hat{x}) \text{ such that } D\tilde{x} = d \tag{7.143}$$

where W is any positive definite weighting matrix. [W is chosen to weight various elements of the difference $(\tilde{x} - \hat{x})$. This is generally based on the designer's relative confidence in the elements of the unconstrained state estimate.] The solution to the above problem is

$$\tilde{x} = \hat{x} - W^{-1}D^T(DW^{-1}D^T)^{-1}(D\hat{x} - d) \tag{7.144}$$

This is the most general approach to the problem. Note that the maximum probability estimate of Equation (7.136) is equal to this if we set $W = P^{-1}$. The mean square estimate of Equation (7.142) is equal to this if we set $W = I$.

It is shown in [Chi85, Sim02] that the constrained state estimate of Equation (7.144) has several interesting properties.

1. The constrained estimate is unbiased. That is, $E(\tilde{x}) = E(x)$.
2. Setting $W = P^{-1}$ results in the minimum variance filter. That is, if $W = P^{-1}$ then $\text{Cov}(x - \tilde{x}) \leq \text{Cov}(x - \hat{x})$ for all $\hat{x}$.
3. Setting $W = I$ results in a constrained estimate that is always (i.e., at each time step) closer to the true state than the unconstrained estimate. That is, if $W = I$ then $||x_k - \tilde{x}_k||_2 \leq ||x_k - \hat{x}_k||_2$ for all k.

The projection approach to constrained filtering has the advantage that it can be easily extended to inequality constraints. That is, if we have the constraints $Dx \leq d$, then the constrained estimate can be obtained by modifying Equation (7.143) and solving the problem

$$\tilde{x} = \text{argmin}_{\tilde{x}}(\tilde{x} - \hat{x})^T W(\tilde{x} - \hat{x}) \text{ such that } D\tilde{x} \leq d \tag{7.145}$$

The problem defined above is known as a quadratic programming problem [Fle81, Gil81]. There are several algorithms for solving quadratic programming problems, most of which fall in the category known as active set methods. An active set method uses the fact that it is only those constraints that are active at the solution of the problem that are significant in the optimality conditions. Assume that we have s inequality constraints (i.e., D has s rows), and q of the s inequality constraints are active at the solution of Equation (7.145). Denote by $\hat{D}$ and $\hat{d}$ the q rows of D and q elements of d corresponding to the active constraints. If the correct

set of active constraints was known *a priori* then the solution of Equation (7.145) would also be a solution of the equality constrained problem

$$\tilde{x} = \text{argmin}_{\tilde{x}}(\tilde{x} - \hat{x})^T W(\tilde{x} - \hat{x}) \text{ such that } \hat{D}\tilde{x} = \hat{d} \tag{7.146}$$

This shows that the inequality constrained problem of Equation (7.145) is equivalent to the equality constrained problem of Equation (7.146). Therefore, all of the properties of the equality constrained state estimate enumerated above also apply to the inequality constrained state estimate. Standard quadratic programming routines[2] can be used to solve inequality constrained problems that are in the form of Equation (7.145).

■ EXAMPLE 7.11

Suppose that we have an unconstrained estimate and covariance given as

$$\begin{aligned} \hat{x} &= \begin{bmatrix} 3 \\ 3 \end{bmatrix} \\ P &= \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{7.147}$$

That is, we are twice as certain of our x_2 estimate as we are of our x_1 estimate. We also know (from our understanding of the underlying system) that the state must satisfy the constraint

$$\begin{aligned} Dx &= d \\ \begin{bmatrix} 1 & 1 \end{bmatrix} x &= 1 \end{aligned} \tag{7.148}$$

Clearly, the unconstrained estimate does not satisfy this constraint. The least squares approach to constrained estimation uses Equation (7.142) to compute the constrained estimate as

$$\begin{aligned} \tilde{x}_{LS} &= \hat{x} - D^T(DD^T)^{-1}(D\hat{x} - d) \\ &= \begin{bmatrix} 3 \\ 3 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \left(\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right)^{-1} \left(\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 3 \end{bmatrix} - 1 \right) \\ &= \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} \end{aligned} \tag{7.149}$$

We see that the estimates for x_1 and x_2 both changed by the same amount (from the unconstrained values of 3, to the constrained values of 1/2). The maximum probability approach to constrained estimation uses Equation (7.136) to compute the constrained estimate as

$$\begin{aligned} \tilde{x}_{MP} &= \hat{x} - PD^T(DPD^T)^{-1}(D\hat{x} - d) \\ &= \begin{bmatrix} -1/3 \\ 4/3 \end{bmatrix} \end{aligned} \tag{7.150}$$

The estimate for x_1 changed by 10/3 (from the unconstrained value of 3, to the constrained value of $-1/3$). The estimate for x_2 changed by 5/3. We see

[2]For example, the QP function in MATLAB's Optimization Toolbox.

that the estimate for x_1 changed twice as much as the estimate for x_2, because the certainty of the unconstrained x_2 estimate was twice the certainty of the unconstrained x_1 estimate. This is illustrated in Figure 7.1.

Figure 7.1 In Example 7.11, the unconstrained estimate violates the equality constraint. The least squares approach to constrained estimation projects the estimate in the direction perpendicular to the constraint surface. The maximum probability approach projects the estimate in the direction P^{-1} relative to the constraint surface.

$\nabla\nabla\nabla$

7.5.4 A pdf truncation approach

In the projection approach to constrained estimation discussed in the previous section, the state estimates are projected onto the constraint surface. In the pdf truncation approach, we take the probability density function that is computed by the Kalman filter (assuming that it is Gaussian) and truncate it at the constraint edges. The constrained state estimate then becomes equal to the mean of the truncated pdf [Shi98, Sim06b]. This approach is designed for inequality constraints on the state, although it can also be applied to equality constraints.

Suppose that at time k we have the s scalar state constraints

$$a_{ki} \leq \phi_{ki}^T x_k \leq b_{ki} \qquad i = 1, \ldots, s \tag{7.151}$$

where $a_{ki} < b_{ki}$. This is a two-sided constraint on the linear function of the state $\phi_{ki}^T x_k$. If we have a one-sided constraint, then we set $a_{ki} = -\infty$ or $b_{ki} = +\infty$. Now suppose at time k that we have a standard Kalman filter estimate $\hat{x}_k$ with covariance P_k. The problem is to truncate the Gaussian pdf $N(\hat{x}_k, P_k)$ at the s constraints given in Equation (7.151), and then find the mean $\tilde{x}_k$ and covariance $\tilde{P}_k$ of the truncated pdf. These new quantities, $\tilde{x}_k$ and $\tilde{P}_k$, become the constrained state estimate and its covariance.

In order to make the problem tractable, we will define $\tilde{x}_{ki}$ as the state estimate after the first i constraints of (7.151) have been enforced, and $\tilde{P}_{ki}$ as the covariance

of $\tilde{x}_{ki}$. We therefore initialize

$$\begin{aligned} i &= 0 \\ \tilde{x}_{ki} &= \hat{x}_k \\ \tilde{P}_{ki} &= P_k \end{aligned} \tag{7.152}$$

Now perform the following transformation:

$$z_{ki} = \rho W^{-1/2} T^T (x_k - \tilde{x}_{ki}) \tag{7.153}$$

ρ is an orthogonal $n \times n$ matrix that will be determined later, and T and W are obtained from the Jordan canonical decomposition of $\tilde{P}_{ki}$. This transformation will allow us to solve the pdf truncation problem that we have posed, and find the mean of the pdf as the estimated state after i constraints have been enforced. From the description of T and W we know that

$$TWT^T = \tilde{P}_{ki} \tag{7.154}$$

T is orthogonal and W is diagonal (therefore, its square root is very easy to compute). Next we use Gram–Schmidt orthogonalization [Moo00] to find the orthogonal ρ matrix that satisfies

$$\rho W^{1/2} T^T \phi_{ki} = \begin{bmatrix} (\phi_{ki}^T \tilde{P}_{ki} \phi_{ki})^{1/2} & 0 & \cdots & 0 \end{bmatrix}^T \tag{7.155}$$

The Gram–Schmidt orthogonalization procedure for computing ρ is given as follows.

1. Suppose that ρ is an $n \times n$ matrix with rows ρ_i $(i = 1, \cdots, n)$:

$$\rho = \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_n \end{bmatrix} \tag{7.156}$$

   The first row of ρ is computed as

$$\rho_1 = \frac{\phi_{ki}^T T W^{1/2}}{(\phi_{ki}^T \tilde{P}_{ki} \phi_{ki})^{1/2}} \tag{7.157}$$

2. For $k = 2, \cdots, n$, perform the following.

   (a) Compute the kth row of ρ as follows:

$$\rho_k = e_k - \sum_{i=1}^{k-1} (e_k^T \rho_i) \rho_i \tag{7.158}$$

   where e_k is the unit vector; that is, e_k is an n-element column vector comprised entirely of zeros, except that its kth element is a 1.

$$e_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow k\text{th element} \tag{7.159}$$

(b) If the ρ_k computed above is zero, then replace it with the following:

$$\rho_k = e_1 - \sum_{i=1}^{k-1} (e_1^T \rho_i)\rho_i \tag{7.160}$$

(c) Normalize ρ_k:

$$\rho_k = \frac{\rho_k}{||\rho_k||_2} \tag{7.161}$$

It can be shown from Equations (7.153)–(7.155) that z_{ki} has a mean of 0 and covariance matrix of identity. With these definitions we see that the upper bound of Equation (7.151) is transformed as follows:

$$\begin{aligned}
\phi_{ki}^T x_k &\leq b_{ki} \\
\phi_{ki}^T T W^{1/2} \rho^T z_{ki} + \phi_{ki}^T \tilde{x}_{ki} &\leq b_{ki} \\
\frac{(\phi_{ki}^T T W^{1/2} \rho^T) z_{ki}}{(\phi_{ki}^T \tilde{P}_{ki} \phi_{ki})^{1/2}} &\leq \frac{b_{ki} - \phi_{ki}^T \tilde{x}_{ki}}{(\phi_{ki}^T \tilde{P}_{ki} \phi_{ki})^{1/2}} \\
\begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} z_{ki} &\leq \frac{b_{ki} - \phi_{ki}^T \tilde{x}_{ki}}{(\phi_{ki}^T \tilde{P}_{ki} \phi_{ki})^{1/2}} \\
&\leq d_{ki}
\end{aligned} \tag{7.162}$$

where d_{ki} is defined by the above equation. Similarly we can see that

$$\begin{aligned}
\begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} z_{ki} &\geq \frac{a_{ki} - \phi_{ki}^T \tilde{x}_{ki}}{(\phi_{ki}^T \tilde{P}_{ki} \phi_{ki})^{1/2}} \\
&\geq c_{ki}
\end{aligned} \tag{7.163}$$

where c_{ki} is defined by the above equation. We therefore have the normalized scalar constraint

$$c_{ki} \leq \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} z_{ki} \leq d_{ki} \tag{7.164}$$

Since z_{ki} has a covariance of identity, its elements are statistically independent of each other. Only the first element of z_{ki} is constrained, so the pdf truncation reduces to a one dimensional pdf truncation. The first element of z_{ki} is distributed as $N(0,1)$ (before constraint enforcement), but the constraint says that z_{ki} must lie between c_{ki} and d_{ki}. We therefore remove that part of the Gaussian pdf that is outside of the constraints and compute the area of the remaining portion of the pdf as

$$\int_{c_{ki}}^{d_{ki}} \frac{1}{\sqrt{2\pi}} \exp(-\zeta^2/2)\, d\zeta = \frac{1}{2}\left[\mathrm{erf}(d_i(k)/\sqrt{2}) - \mathrm{erf}(c_{ki}/\sqrt{2})\right] \tag{7.165}$$

where $\mathrm{erf}(\cdot)$ is the error function, defined as

$$\mathrm{erf}(t) = \frac{2}{\sqrt{\pi}} \int_0^t \exp(-\gamma^2)\, d\gamma \tag{7.166}$$

(Note that the error function is sometimes defined without the $2/\sqrt{\pi}$ factor, which can lead to confusion. However, the above definition is the most commonly used

one.) We normalize the truncated pdf so that it has an area of one, and we find that the truncated pdf (i.e., the constrained pdf of the first element of z_{ki}) is given by

$$\begin{aligned} \text{pdf}(\zeta) &= \begin{cases} \alpha \exp(-\zeta^2/2) & \zeta \in [c_{ki}, d_{ki}] \\ 0 & \text{otherwise} \end{cases} \\ \alpha &= \frac{\sqrt{2}}{\sqrt{\pi}\left[\text{erf}(d_{ki}/\sqrt{2}) - \text{erf}(c_{ki}/\sqrt{2})\right]} \end{aligned} \tag{7.167}$$

We define $z_{k,i+1}$ as the random variable that has the same pdf as z_{ki} except that the pdf is truncated and normalized, so that its pdf lies entirely between the limits c_{ki} and d_{ki}:

$$\text{pdf}(z_{k,i+1}) = \text{truncated pdf}(z_{k,i}) \tag{7.168}$$

We can compute the mean and variance of $z_{k,i+1}$ as follows:

$$\begin{aligned} \mu &= E[z_{k,i+1}] \\ &= \alpha \int_{c_{ki}}^{d_{ki}} \zeta \exp(-\zeta^2/2)\, d\zeta \\ &= \alpha \left[\exp(-c_{ki}^2/2) - \exp(-d_{ki}^2/2)\right] \\ \sigma^2 &= E\left[(z_{k,i+1} - \mu)^2\right] \\ &= \alpha \int_{c_{ki}}^{d_{ki}} (\zeta - \mu)^2 \exp(-\zeta^2/2)\, d\zeta \\ &= \alpha \left[\exp(-c_{ki}^2/2)(c_{ki} - 2\mu) - \exp(-d_{ki}^2/2)(d_{ki} - 2\mu)\right] + \mu^2 + 1 \end{aligned} \tag{7.169}$$

The mean and variance of the transformed state estimate, after enforcement of the first constraint, are therefore given as

$$\begin{aligned} \tilde{z}_{k,i+1} &= \begin{bmatrix} \mu & 0 & \cdots & 0 \end{bmatrix}^T \\ \text{Cov}(\tilde{z}_{k,i+1}) &= \text{diag}(\sigma^2, 1, \cdots, 1) \end{aligned} \tag{7.170}$$

We then take the inverse of the transformation of Equation (7.153) to find the mean and variance of the state estimate after enforcement of the first constraint.

$$\begin{aligned} \tilde{x}_{k,i+1} &= TW^{1/2}\rho^T \tilde{z}_{k,i+1} + \tilde{x}_{ki} \\ \tilde{P}_{k,i+1} &= TW^{1/2}\rho^T \text{Cov}(\tilde{z}_{k,i+1})\rho W^{1/2} T^T \end{aligned} \tag{7.171}$$

We then increment i by one and repeat the process of Equations (7.153)–(7.171) to obtain the state estimate after enforcement of the next constraint. Note that $\tilde{x}_{k0}$ is the unconstrained state estimate at time k, $\tilde{x}_{k1}$ is the state estimate at time k after the enforcement of the first constraint, $\tilde{x}_{k2}$ is the state estimate at time k after the enforcement of the first two constraints, and so on. After going through this process s times (once for each constraint), we have the final constrained state estimate and covariance at time k:

$$\begin{aligned} \tilde{x}_k &= \tilde{x}_{ks} \\ \tilde{P}_k &= \tilde{P}_{ks} \end{aligned} \tag{7.172}$$

Figure 7.2 shows an example of a one-dimensional state estimate before and after truncation. Before truncation, the state estimate is outside of the state constraints. After truncation, the state estimate is set equal to the mean of the truncated pdf. Figure 7.3 shows another example. In this case, the unconstrained state estimate is inside the state constraints. However, truncation changes the pdf and so the constrained state estimate changes to the mean of the truncated pdf.

Figure 7.2 The unconstrained estimate at $x = 0$ violates the constraints. The constrained estimate, which is at $x \approx -1.38$, is the centroid of the truncated pdf.

Figure 7.3 The unconstrained estimate at $x = 0$ satisfies the constraints. Nevertheless, the truncation approach to constrained estimation shifts the estimate to the centroid of the truncated pdf, which is at $x \approx -0.23$.

■ **EXAMPLE 7.12**

In this example, we consider a vehicle navigation problem. The first two state elements are the north and east positions of a land vehicle, and the last two elements are the north and east velocities. The velocity of the vehicle is in the direction of θ, an angle measured clockwise from due east. A position-measuring device provides a noisy measurement of the vehicle's north and east positions. The process and measurement equations for this system can be written as

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 0 \\ T\sin\theta \\ T\cos\theta \end{bmatrix} u_k + w_k \\ y_k &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_k + v_k \end{aligned} \tag{7.173}$$

where T is the discretization step size. We can implement a Kalman filter to estimate the position and velocity of the vehicle based on our noisy position measurements. If we know that the vehicle is on a road with a heading of θ, then we know that

$$\begin{aligned} \tan\theta &= x(1)/x(2) \\ &= x(3)/x(4) \end{aligned} \tag{7.174}$$

These constraints can be written as

$$\begin{bmatrix} 1 & -\tan\theta & 0 & 0 \\ 0 & 0 & 1 & -\tan\theta \end{bmatrix} x_k = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{7.175}$$

The constrained filter can be implemented using any of the four approaches discussed in this section (model reduction, perfect measurements, projection, or pdf truncation). Figure 7.4 shows the magnitude of the north position estimation error of the unconstrained and constrained filters (projection approach using $W = I$) for a typical simulation. In this example, significant estimation improvement can be obtained when constraint information is incorporated into the filter, although the improvement will be problem dependent.

$\nabla\nabla\nabla$

It is clear from this section that there are a variety of ways to enforce equality or inequality constraints on state estimation problems. The "best" way is not clear-cut, and probably depends on the application. Other approaches to constrained estimation and some discussion of the mathematical meaning of state constraints can be found in [Hel94, Rao03, Dew04, Goo05a, Goo05b, Ko06].

7.6 SUMMARY

In this chapter, we discussed a variety of Kalman filter generalizations that make the filter more widely applicable to a broader class of problems. Correlated and colored process and measurement noise were studied early in the history of the

Figure 7.4 North position estimation error magnitude of the unconstrained and constrained Kalman filters for Example 7.12.

Kalman filter. We showed in this chapter that filter modifications taking correlation and color into account can improve estimation performance. However, whether or not these approaches are worth the extra complexity and computational effort is problem dependent. One of the most practical extensions of the Kalman filter is the steady-state Kalman filter. The steady-state Kalman filter often performs nearly identically to the more theoretically rigorous time-varying filter. However, the steady-state filter requires only a fraction of the computational cost. The α-β and α-β-γ filters are special cases of the steady-state Kalman filter. We also discussed the fading-memory filter, which is a way of making the Kalman filter more robust to modeling errors. The fading-memory filter is a simple modification to the Kalman filter that can noticeably improve filter performance. Further discussion of filter robustness is found in Section 10.4 and Chapter 11. Finally, we discussed several ways to incorporate state constraints in the Kalman filter to improve estimation accuracy when information other than the state model is available. Other Kalman filter generalizations are discussed in later chapters of this book.

- Kalman filters with fewer states than the system (Section 10.3)
- Kalman filtering when the system model or noise statistics are not known (Section 10.4)
- Kalman filtering when the measurements arrive at the filter in the wrong order (Section 10.5)
- Kalman filters for nonlinear systems (Chapter 13)

Further generalizations undoubtedly await future development by the efforts of enterprising students and researchers.

PROBLEMS

Written exercises

7.1 Consider the scalar system

$$\begin{aligned} x_k &= \frac{1}{2}x_{k-1} + w_{k-1} \\ y_k &= x_k + v_k \\ v_k &= \frac{1}{2}v_{k-1} + \zeta_{k-1} \end{aligned}$$

where $w_k \sim (0, Q)$ and $\zeta_k \sim (0, Q_\zeta)$. Let $Q = Q_\zeta = 1$.

a) Design a Kalman filter in which the dynamics of the measurement noise v_k are ignored and it is assumed that v_k is white noise with a variance of Q_ζ. Based on the incorrect Kalman filter equations, what does the Kalman filter think that the steady-state *a posteriori* estimation covariance is?

b) Based on the incorrect Kalman filter equations, what is the true steady-state *a posteriori* estimation covariance $E(e_k^2)$? Hint: Find a recursive equation for $E(e_k^2)$ in terms of $E(e_{k-1}^2)$, $E(w_{k-1}^2)$, $E(v_k^2)$, and $E(e_{k-1}v_k)$, then solve for the steady-state value of $E(e_k^2)$.

c) Design a Kalman filter using the state augmentation approach in which the dynamics of the measurement noise are correctly taken into account. What is the steady-state estimation covariance? Hint: You may need to use MATLAB's DARE function to solve the steady-state Riccati equation that is associated with this question.

7.2 Show that the Kalman filter for an LTI system with a noise-free scalar measurement that satisfies the equation $(HQH^T)Q = QH^THQ$ has a steady-state *a posteriori* covariance of zero.

7.3 Consider the scalar system

$$\begin{aligned} x_k &= x_{k-1} + w_{k-1} \\ y_k &= x_k + v_k \end{aligned}$$

where $w_k \sim (0, Q)$ and $v_k \sim (0, R)$ are white noise processes with $Q = R = 1$. Suppose that $E(w_k v_{k+1}) = M = 1$.

a) Design a Kalman filter in which the correlation between w_k and v_{k+1} is ignored. Based on the incorrect Kalman filter equations, what does it appear that the steady-state *a posteriori* estimation covariance is?

b) For the Kalman filter designed above, write a recursive equation for the *a posteriori* estimation error $e_k = x_k - \hat{x}_k^+$. Use this equation to find the steady-state solution to $E(e_k^2)$.

c) Design a Kalman filter in which the correlation between w_k and v_{k+1} is correctly taken into account. Show that the steady-state *a posteriori* estimation covariance is zero. Explain why the estimation covariance goes to zero in spite of the existence of process noise and measurement noise. (Hint: Use the correlation between w_k and v_{k+1} to write an equivalent two-state system, and then use the results of Problem 7.2.)

7.4 Consider the system

$$\begin{aligned} x_k &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix} x_{k-1} + w_{k-1} \\ y_k &= \begin{bmatrix} 1 & 1 \end{bmatrix} x_k + v_k \end{aligned}$$

where $w_k \sim (0, Q)$ and $Q = I$.

a) Find one matrix square root of Q.
b) Is (F, H) observable?
c) Is (F, H) detectable?
d) Is (F, G) controllable for all G such that $GG^T = Q$?
e) Is (F, G) stabilizable for all G such that $GG^T = Q$?
f) Use the above results to specify how many positive definite solutions exist to the DARE that is associated with the Kalman filter for this problem.
g) Use the above results to specify whether or not the steady-state Kalman filter for this system is stable.

7.5 Prove that the matrix $\mathcal{H}$ in Equation (7.85) is symplectic.

7.6 In this problem, we will use the shorthand notation $P = P^+$ and $M = P^-$. Use the following procedure to find α as a function of β for the α-β filter [Bar01].

a) Use the time-update equation for M to solve for the three unique elements of P as a function of the three unique elements of M.
b) Use the measurement-update equation for P to solve for the three unique elements of P as a function of the three unique elements of M.
c) Equate the sets of equations from the two steps above to get expressions for $M_{11}K_1$, $M_{12}K_1$, and $M_{12}K_2$, that do not have any P_{ij} terms.
d) Use Equation (7.64) to solve for M_{11} and M_{12}.
e) Combine the five equations from the two previous steps to get a single equation with K_1 and K_2 that does not have any M_{ij} terms.
f) Replace K_1 and K_2 in the previously obtained equation with α and β from Equation (7.63), then solve for α as a function of β.

7.7 Prove the properties of symplectic matrices that are listed immediately following Equation (7.86).

7.8 Recall that the steady-state, zero-input, one-step formulation of the *a posteriori* Kalman filter can be written as

$$\begin{aligned} \hat{x}_k^+ &= (I - KH)F\hat{x}_{k-1}^+ + Ky_k \\ \hat{y}_k &= H\hat{x}_k^+ \end{aligned}$$

Prove that if (F, H) is observable and $(I - HK)$ is full rank, then the Kalman filter in the above equation is an observable system. Hint: $H(I - KH) = (I - HK)H$.

7.9 Suppose you have a two-state Newtonian system of the type described in Section 7.3.1. The sample time is 1 and the variance of the acceleration noise is 1. A requirement is given to estimate the position with an *a posteriori* steady-state variance of 1 or less. What is the largest measurement variance that will meet the requirement?

Computer exercises

7.10 Consider the system described in Problem 7.1. Implement the Kalman filter that assumes white noise and the Kalman filter that assumes colored noise. Numerically calculate the RMS *a posteriori* estimation-error variance and verify that it matches the analytically calculated values from your answer to Problem 7.1.

7.11 Plot the α and β parameters of the α-β filter as a function of λ. Use a log scale for λ with a range of 10^{-3} to 10^3. What are the limiting values of α and β as $\lambda \to 0$? Does this make intuitive sense? What are the limiting values of α and β as $\lambda \to \infty$?

7.12 Plot the α, β, and γ parameters of the α-β-γ filter as a function of λ. Use a log scale for λ with a range of 10^{-3} to 10^3. What are the limiting values of α, β, and γ as $\lambda \to 0$? Does this make intuitive sense? What are the limiting values of α, β, and γ as $\lambda \to \infty$?

7.13 A simple model of the ingestion and metabolism of a drug is given as

$$\begin{aligned} \dot{x}_1 &= -k_1 x_1 + u \\ \dot{x}_2 &= k_1 x_1 - k_2 x_2 \\ y(t_k) &= x_2(t_k) + v(t_k) \end{aligned}$$

where the units of time are days, x_1 is the mass of the drug in the gastrointestinal tract, x_2 is the mass of the drug in the bloodstream, and u is the ingestion rate of the drug. Suppose that $k_1 = k_2 = 1$. The measurement noise $v(t_k)$ is zero-mean and unity variance. The initial state, estimate, and covariance are

$$\begin{aligned} x(0) &= \begin{bmatrix} 0.8 \\ 0 \end{bmatrix} \\ \hat{x}(0) &= x(0) \\ P(0) &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

It is known from physical constraints that $x_1 \in [0.8, 1]$.

a) Discretize the system with a step size of 1 hour.

b) Implement the discrete-time Kalman filter, the projection-based constrained Kalman filter with $W = I$, and the pdf truncation constrained filter. Run simulations of these filters for a three-day period. Plot the magnitude of the x_1 estimation error for the three filters. Which filter appears to perform best? Which filter appears to perform worst?

CHAPTER 8

The continuous-time Kalman filter

Our philosophy here will be to model phenomena with differential equations and then to form estimates of the physical quantities which also satisfy differential equations.
—Richard Bucy [Buc68, Chapter 1]

James Follin, A. G. Carlton, James Hanson, and Richard Bucy developed the continuous-time Kalman filter in unpublished work for the Johns Hopkins Applied Physics Lab in the late 1950s. Rudolph Kalman independently developed the discrete-time Kalman filter in 1960. In April 1960 Kalman and Bucy became aware of each other's work and collaborated on the publication of the continuous-time Kalman filter in [Kal61]. This filter is sometimes referred to as the Kalman–Bucy filter. Further historical notes are given in Appendix A.

The vast majority of Kalman filter applications are implemented in digital computers, so it may seem superfluous to discuss Kalman filtering for continuous-time measurements. However, there are still opportunities to implement Kalman filters in continuous time (i.e., in analog circuits) [Hug88]. Furthermore, the derivation of the continuous-time filter is instructive from a pedagogical point of view. Finally, steady-state continuous-time estimators can be analyzed using conventional frequency-domain concepts, which provides an advantage over discrete-time estimators [Bal87, Ste94]. In light of these factors, this chapter presents the continuous-time Kalman filter.

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

Our derivation of the continuous-time filter starts with the previously developed discrete-time filter from Chapter 5, and then takes the limit as the time step decreases to zero. Section 8.1 shows the relationship between continuous-time white noise and discrete-time white noise, which is the foundation for the derivation of the continuous-time Kalman filter. Section 8.2 derives the Kalman filter for the case of continuous-time system dynamics and continuous-time measurements. Section 8.3 shows some creative methods to solve the continuous-time Riccati equation, which is a key component of the continuous-time Kalman filter. Section 8.4 discusses the continuous-time Kalman filter for the cases of correlated process and measurement noise, and for colored measurement noise. Section 8.5 discusses the steady-state continuous-time Kalman filter, its relationship to the Wiener filter of Section 3.4, and its relationship to linear quadratic optimal control.

8.1 DISCRETE-TIME AND CONTINUOUS-TIME WHITE NOISE

In this section, we will show the relationship between discrete-time white noise and continuous-time white noise. We need to understand this relationship because in the next section we will derive the continuous-time Kalman filter as the limiting case of the discrete-time Kalman filter as the sample time decreases to zero. First we will discuss the relationship between discrete-time and continuous-time process noise, and then we will discuss the relationship between discrete-time and continuous-time measurement noise.

8.1.1 Process noise

Consider the following discrete-time system with an identity state transition matrix and a sample time of T:

$$\begin{aligned} x_k &= x_{k-1} + w_{k-1} \\ w_k &\sim (0, Q) \\ x_0 &= 0 \end{aligned} \tag{8.1}$$

where $\{w_k\}$ is a discrete-time white noise process. Let us see what effect the white noise has on the covariance of the state. We can solve this discrete-time system for the state as follows:

$$x_k = w_0 + w_1 + \cdots + w_{k-1} \tag{8.2}$$

The covariance of the state is therefore given as

$$\begin{aligned} E[x_k x_k^T] &= E[(w_0 + w_1 + \cdots + w_{k-1})(w_0 + w_1 + \cdots + w_{k-1})^T] \\ &= E[w_0 w_0^T] + E[w_1 w_1^T] + \cdots + E[w_{k-1} w_{k-1}^T] \\ &= kQ \end{aligned} \tag{8.3}$$

The value of the continuous-time parameter t is equal to the number of discrete-time steps k times the sample time T. That is, $t = kT$. We therefore see that

$$\begin{aligned} E[x(t)x^T(t)] &= E[x_k x_k^T] \\ &= kQ \end{aligned} \tag{8.4}$$

The covariance of the state increases linearly with time for a given sample time T. Now consider the continuous-time system with an identity state transition matrix:

$$\dot{x}(t) = w(t) \tag{8.5}$$

where $w(t)$ is continuous-time white noise. We propose (in hindsight) the following definition for continuous-time white noise:

$$E[w(t)w^T(\tau)] = \frac{Q}{T}\delta(t-\tau) \tag{8.6}$$

where Q and T are the same as they are in the discrete-time system of Equation (8.1). $\delta(t-\tau)$ is the continuous-time impulse response; it is a function with a value of ∞ at $t=\tau$, a value of 0 everywhere else, and an area of 1. Let us compute the covariance of $x(t)$ in Equation (8.5):

$$\begin{aligned} E[x(t)x^T(t)] &= E\left[\int_0^t w(\alpha)\,d\alpha \int_0^t w^T(\beta)\,d\beta\right] \\ &= \int_0^t \int_0^t E\left[w(\alpha)w^T(\beta)\right]\,d\alpha\,d\beta \end{aligned} \tag{8.7}$$

Substituting Equation (8.6) into the above equation gives

$$\begin{aligned} E[x(t)x^T(t)] &= \int_0^t \int_0^t \frac{Q}{T}\delta(\alpha-\beta)\,d\alpha\,d\beta \\ &= \int_0^t \frac{Q}{T}\,d\beta \\ &= \frac{Qt}{T} \end{aligned} \tag{8.8}$$

where we have used the sifting property of the continuous-time impulse function (see Problem 4.10). Recalling that $t = kT$, we can write the above equation as

$$E[x(t)x^T(t)] = kQ \tag{8.9}$$

Comparing this with Equation (8.4), we see that the covariance of the state of the continuous-time system increases with time in exactly the same way as the covariance of the state of the discrete-time system. In other words, discrete-time white noise with covariance Q in a system with a sample period of T, is equivalent to continuous-time white noise with covariance $Q_c\delta(t)$, where $Q_c = Q/T$. Zero-mean continuous-time white noise is denoted as

$$w(t) \sim (0, Q_c) \tag{8.10}$$

which is equivalent to saying that

$$E[w(t)w^T(\tau)] = Q_c\delta(t-\tau) \tag{8.11}$$

Continuous-time white noise is counterintuitive because $w(t)$ is infinitely correlated with $w(\tau)$ at $t=\tau$, but it has zero correlation with itself when $t \neq \tau$. Nevertheless, it can be approximately descriptive of real processes. Also, continuous-time white noise is mathematically well defined and is a useful device that we will use in this chapter. Additional discussion about the relationship between discrete-time and continuous-time white noise can be found in [Kai81, Smi78].

8,1.2 Measurement noise

Now let us think about measurement noise. Suppose we have a discrete-time measurement of a constant x every T seconds. The measurement times are $t_k = kT$ $(k = 1, 2, \cdots)$:

$$\begin{aligned} x_k &= x_{k-1} \\ y_k &= x_k + v_k \\ v_k &\sim (0, R) \end{aligned} \tag{8.12}$$

From the Kalman filter equations in Section 5.1 we find that the *a posteriori* estimation-error covariance is given by

$$P_{k+1}^+ = \frac{P_k^+ R}{P_k^+ + R} \tag{8.13}$$

From this it can be shown that

$$\begin{aligned} P_k^+ &= \frac{P_0 R}{k P_0 + R} \\ \lim_{P_0 \to \infty} P_k^+ &= \frac{R}{k} \\ &= \frac{RT}{t_k} \end{aligned} \tag{8.14}$$

The error covariance at time t_k is independent of the sample time T if

$$R = \frac{R_c}{T} \tag{8.15}$$

where R_c is some constant. This implies that

$$\lim_{T \to 0} R = R_c \delta(t) \tag{8.16}$$

where $\delta(t)$ is the continuous-time impulse function. This establishes the equivalence between white measurement noise in discrete time and continuous time. The effects of white measurement noise in discrete time and continuous time are the same if

$$\begin{aligned} v_k &\sim (0, R) \\ v(t) &\sim (0, R_c) \end{aligned} \tag{8.17}$$

Equation (8.15) specifies the relationship between R and R_c, and the second equation above is a shorthand way of saying

$$E[v(t)v(\tau)] = R_c \delta(t - \tau) \tag{8.18}$$

8.1.3 Discretized simulation of noisy continuous-time systems

The results of the above sections can be combined with the results of Section 1.4 to obtain a discretized simulation of a noisy continuous-time system for the purpose

of implementing a discrete-time state estimator. Suppose that we have a system given as

$$\begin{aligned} \dot{x} &= Ax + Bu + w \\ y &= Cx + v \\ w &\sim (0, Q_c) \\ v &\sim (0, R_c) \end{aligned} \tag{8.19}$$

Both $w(t)$ and $v(t)$ are continuous-time noise, and $u(t)$ is a known input. This system is approximately equivalent to the following discrete-time system:

$$\begin{aligned} x_k &= e^{A\Delta t} x_{k-1} + e^{A\Delta t} \int_0^{\Delta t} e^{-A\tau}\, d\tau\, B u_{k-1} + w_k \\ &= e^{A\Delta t} x_{k-1} + e^{A\Delta t} \left[I - e^{-A\Delta t} \right] A^{-1} B u_{k-1} + w_k \\ y_k &= C x_k + v_k \\ w_k &\sim (0, Q_c \Delta t) \\ v_k &\sim (0, R_c / \Delta t) \end{aligned} \tag{8.20}$$

where Δt is the discretization step size. The second expression for x_k above is valid if A^{-1} exists. If we use these discretized equations to simulate a continuous-time system, then we can simulate a continuous-time state estimator using the resulting measurements with one of the integration methods discussed in Section 1.5. The remainder of this chapter discusses continuous-time state estimation.

8.2 DERIVATION OF THE CONTINUOUS-TIME KALMAN FILTER

We will now use the results of the previous section to derive the continuous-time Kalman filter. Suppose that we have a continuous-time system given as

$$\begin{aligned} \dot{x} &= Ax + Bu + w \\ y &= Cx + v \\ w &\sim (0, Q_c) \\ v &\sim (0, R_c) \end{aligned} \tag{8.21}$$

When we write $w \sim (0, Q_c)$ we mean exactly what is written in Equation (8.11). When we write $v \sim (0, R_c)$ we mean exactly what is written in Equation (8.18). Now suppose that we discretize this system with a sample time of T (see Section 1.4). We obtain

$$\begin{aligned} x_k &= F x_{k-1} + G u_{k-1} + \Lambda w_{k-1} \\ y_k &= H x_k + v_k \end{aligned} \tag{8.22}$$

The matrices in this discrete-time system are computed as follows:

$$
\begin{aligned}
F &= \exp(AT) \\
&\approx (I + AT) \text{ for small } T \\
G &= (\exp(AT) - I)A^{-1}B \\
&\approx BT \text{ for small } T \\
\Lambda &= (\exp(AT) - I)A^{-1} \\
&\approx IT \text{ for small } T \\
H &= C \\
w_k &\sim (0, Q), \qquad Q = Q_c T \\
v_k &\sim N(0, R), \qquad R = R_c/T
\end{aligned} \tag{8.23}
$$

The discrete-time Kalman filter gain for this system was derived in Section 5.1 as

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \tag{8.24}$$

From this we can derive

$$
\begin{aligned}
K_k &= P_k^- C^T (C P_k^- C^T + R_c/T)^{-1} \\
\frac{K_k}{T} &= P_k^- C^T (C P_k^- C^T T + R_c)^{-1} \\
\lim_{T \to 0} \frac{K_k}{T} &= P_k^- C^T R_c^{-1}
\end{aligned} \tag{8.25}
$$

The estimation-error covariances were derived in Section 5.1 as

$$
\begin{aligned}
P_k^+ &= (I - K_k H) P_k^- \\
P_{k+1}^- &= F P_k^+ F^T + Q
\end{aligned} \tag{8.26}
$$

For small values of T, this can be written as

$$
\begin{aligned}
P_{k+1}^- &= (I + AT) P_k^+ (I + AT)^T + Q_c T \\
&= P_k^+ + (A P_k^+ + P_k^+ A^T + Q_c) T + A P_k^+ A^T T^2
\end{aligned} \tag{8.27}
$$

Substituting for P_k^+ gives

$$
\begin{aligned}
P_{k+1}^- &= (I - K_k C) P_k^- + A P_k^+ A^T T^2 + \\
&\quad [A(I - K_k C) P_k^- + (I - K_k C) P_k^- A^T + Q_c] T
\end{aligned} \tag{8.28}
$$

Subtracting P_k^- from both sides and then dividing by T gives

$$
\begin{aligned}
\frac{P_{k+1}^- - P_k^-}{T} &= \frac{-K_k C P_k^-}{T} + A P_k^+ A^T T + \\
&\quad (A P_k^- + A K_k C P_k^- + P_k^- A^T - K_k C P_k^- A^T + Q_c)
\end{aligned} \tag{8.29}
$$

Taking the limit as $T \to 0$ and using Equation (8.25) gives

$$
\begin{aligned}
\dot{P} &= \lim_{T \to 0} \frac{P_{k+1}^- - P_k^-}{T} \\
&= -P C^T R_c^{-1} C P + A P + P A^T + Q_c
\end{aligned} \tag{8.30}
$$

This equation for P is called a differential Riccati equation and can be used to compute the estimation-error covariance for the continuous-time Kalman filter. This requires n^2 integrations because P is an $n \times n$ matrix. But P is symmetric, so in practice we only need to integrate $n(n+1)/2$ equations in order to solve for P.

In Section 5.1 we derived the Kalman filter equations for $\hat{x}$ as

$$\begin{aligned} \hat{x}_k^- &= F\hat{x}_{k-1}^+ + Gu_{k-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H\hat{x}_k^-) \end{aligned} \tag{8.31}$$

If we assume that T is small we can use Equation (8.23) to write the measurement update equation as

$$\begin{aligned} \hat{x}_k^+ &= F\hat{x}_{k-1}^+ + Gu_{k-1} + K_k(y_k - HF\hat{x}_{k-1}^+ - HGu_{k-1}) \\ &\approx (I + AT)\hat{x}_{k-1}^+ + BTu_{k-1} + \\ &\quad K_k(y_k - C(I + AT)\hat{x}_{k-1}^+ - CBTu_{k-1}) \end{aligned} \tag{8.32}$$

Now substitute for K_k from Equation (8.25) to obtain

$$\begin{aligned} \hat{x}_k^+ &= \hat{x}_{k-1}^+ + AT\hat{x}_{k-1}^+ + BTu_{k-1} + \\ &\quad PC^T R_c^{-1} T(y_k - C\hat{x}_{k-1}^+ - CAT\hat{x}_{k-1}^+ - CBTu_{k-1}) \end{aligned} \tag{8.33}$$

Subtracting x_{k-1}^+ from both sides, dividing by T, and taking the limit as $T \to 0$, gives

$$\begin{aligned} \dot{\hat{x}} &= \lim_{T \to 0} \frac{\hat{x}_k^+ - \hat{x}_{k-1}^+}{T} \\ &= A\hat{x} + Bu + PC^T R_c^{-1}(y - C\hat{x}) \end{aligned} \tag{8.34}$$

This can be written as

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + K(y - C\hat{x}) \\ K &= PC^T R_c^{-1} \end{aligned} \tag{8.35}$$

This gives the differential equation that can be used to integrate the state estimate in the continuous-time Kalman filter.

The continuous-time Kalman filter

The continuous-time Kalman filter can be summarized as follows.

1. The continuous-time system dynamics and measurement equations are given as

$$\begin{aligned} \dot{x} &= Ax + Bu + w \\ y &= Cx + v \\ w &\sim (0, Q_c) \\ v &\sim (0, R_c) \end{aligned} \tag{8.36}$$

   Note that $w(t)$ and $v(t)$ are continuous-time white noise processes.

2. The continuous-time Kalman filter equations are given as

$$\begin{aligned}
\hat{x}(0) &= E[x(0)] \\
P(0) &= E[(x(0)-\hat{x}(0))(x(0)-\hat{x}(0))^T] \\
K &= PC^T R_c^{-1} \\
\dot{\hat{x}} &= A\hat{x} + Bu + K(y - C\hat{x}) \\
\dot{P} &= -PC^T R_c^{-1} CP + AP + PA^T + Q_c
\end{aligned} \tag{8.37}$$

Other methods of deriving the continuous-time Kalman filter also exist. For example, George Johnson presented a derivation that is based on finding the gain that minimizes the derivative of the estimation covariance [Joh69].

■ EXAMPLE 8.1

In this example we will use the continuous-time Kalman filter to estimate a constant given continuous-time noisy measurements:

$$\begin{aligned}
\dot{x} &= 0 \\
y &= x + v \\
v &\sim (0, R)
\end{aligned} \tag{8.38}$$

We see that $A = 0$, $Q = 0$, and $C = 1$. Equation (8.37) gives the differential equation for the covariance as

$$\begin{aligned}
\dot{P} &= -PC^T R^{-1} CP + AP + PA^T + Q \\
&= -P^2/R
\end{aligned} \tag{8.39}$$

with the initial condition $P(0) = P_0$. From this we can derive

$$\begin{aligned}
\frac{dP}{P^2} &= \frac{-d\tau}{R} \\
\int_{P(0)}^{P(t)} \frac{1}{P^2}\, dP &= -\int_0^t \frac{1}{R}\, d\tau \\
-(P^{-1} - P_0^{-1}) &= -t/R \\
P^{-1} &= P_0^{-1} + t/R \\
P &= (P_0^{-1} + t/R)^{-1} \\
&= \frac{P_0}{1 + P_0 t/R} \\
\lim_{t\to\infty} P &= 0
\end{aligned} \tag{8.40}$$

Equation (8.37) gives the Kalman gain as

$$\begin{aligned}
K &= PC^T R^{-1} \\
&= \frac{P_0/R}{1 + P_0 t/R} \\
\lim_{t\to\infty} K &= 0
\end{aligned} \tag{8.41}$$

Equation (8.37) gives the state-update equation as

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - C\hat{x}) \tag{8.42}$$

from which we can derive

$$\begin{aligned} \dot{\hat{x}} &= K(y - \hat{x}) \\ \lim_{t\to\infty} \dot{\hat{x}} &= 0 \end{aligned} \tag{8.43}$$

This shows that as time goes to infinity, $\hat{x}$ reaches a steady-state value. This is intuitive because as we obtain an infinite number of measurements of a constant, our estimate of that constant becomes perfect and additional measurements cannot improve our estimate. Furthermore, the Kalman gain goes to zero as time goes to infinity, which again says that we ignore additional measurements (since our estimate becomes perfect). Finally, the covariance P goes to zero as time goes to infinity, which says that the uncertainty in our estimate goes to zero, meaning that our estimate is perfect. Compare this example with the equivalent discrete-time system discussed in Example 7.10.

$\nabla\nabla\nabla$

■ EXAMPLE 8.2

In this example we are able to obtain measurements of the velocity of an object that is moving in one dimension. The object is subject to random accelerations. We want to estimate the velocity x from noisy velocity measurements. The system and measurement equations are given as

$$\begin{aligned} \dot{x} &= w \\ y &= x + v \\ w &\sim (0, Q) \\ v &\sim (0, R) \end{aligned} \tag{8.44}$$

We see that $A = 0$ and $C = 1$. From the covariance update of Equation (8.37) we obtain

$$\begin{aligned} \dot{P} &= -PC^T R^{-1} CP + AP + PA^T + Q \\ &= -P^2/R + Q \end{aligned} \tag{8.45}$$

with the initial condition $P(0) = P_0$. From this we can derive

$$\begin{aligned} dP &= (Q - P^2/R)d\tau \\ \int_{P(0)}^{P(t)} \frac{dP}{Q - P^2/R} &= \int_0^t d\tau \\ \frac{1}{2\sqrt{Q}} \ln\left(\frac{\sqrt{Q} + P/\sqrt{R}}{\sqrt{Q} - P/\sqrt{R}}\right)\Bigg|_{P(0)}^{P(t)} &= t \end{aligned} \tag{8.46}$$

Solving this for P gives

$$\begin{aligned} P &= \sqrt{QR}\left[\frac{P_0 - \sqrt{QR} + (\sqrt{QR} + P_0)\exp(2t\sqrt{Q})}{\sqrt{QR} - P_0 + (\sqrt{QR} + P_0)\exp(2t\sqrt{Q})}\right] \\ \lim_{t\to\infty} P &= \sqrt{QR} \end{aligned} \tag{8.47}$$

The Kalman gain is obtained from Equation (8.37) as

$$\begin{aligned} K &= PC^T R^{-1} \\ &= P/R \\ \lim_{t\to\infty} K &= \sqrt{Q/R} \end{aligned} \tag{8.48}$$

The state estimate update expression is obtained from Equation (8.37) as

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + K(y - C\hat{x}) \\ &= K(y - \hat{x}) \end{aligned} \tag{8.49}$$

From these expressions we see that if process noise increases (i.e., Q increases) then K increases. This is intuitively agreeable, because from the $\dot{\hat{x}}$ equation we see that K defines the rate at which we change $\hat{x}$ based on the measurements. If Q is large then we have less confidence in our system model, and relatively more confidence in our measurements, so we change $\hat{x}$ more aggressively to be consistent with our measurements.

Similarly, we see that if we have large measurement noise (i.e., R is large) then K decreases. This is again intuitively agreeable. Large measurement noise means that we have less confidence in our measurements, so we change $\hat{x}$ less aggressively to be consistent with our measurements.

Finally, we see that P increases as both Q and R increase. An increase in the noise in either the system model or the measurements will degrade our confidence in our state estimate.

$\nabla\nabla\nabla$

8.3 ALTERNATE SOLUTIONS TO THE RICCATI EQUATION

The differential Riccati equation of Equation (8.37) can be computationally expensive to integrate, especially for systems with small time constants. Also, direct integration of the Riccati equation may result in a P matrix that loses its positive definiteness due to numerical problems. In this section we will look at some alternate solutions to the differential Riccati equation. This first two methods, called the transition matrix approach and the Chandrasekhar algorithm, are both intended to reduce computational effort. The third method, called square root filtering, is intended to reduce numerical difficulties.

8.3.1 The transition matrix approach

Assume that $P = \Lambda Y^{-1}$, where Λ and Y are $n \times n$ matrices to be determined. In the following we will determine what equalities must be satisfied by Λ and Y in order for this factorization to be valid. If the factorization is valid then

$$\begin{aligned} \dot{P} &= \dot{\Lambda} Y^{-1} + \Lambda \frac{d}{dt}(Y^{-1}) \\ &= \dot{\Lambda} Y^{-1} - \Lambda Y^{-1} \dot{Y} Y^{-1} \end{aligned} \tag{8.50}$$

where we have used Equation (1.51) for the time derivative of Y^{-1}. We post-multiply both sides of the above equation by Y to obtain

$$\dot{P}Y = \dot{\Lambda} - \Lambda Y^{-1}\dot{Y} \tag{8.51}$$

Recall from Equation (8.37) that the differential equation for P is given by

$$\dot{P} = AP + PA^T - PC^T R^{-1} CP + Q \tag{8.52}$$

Substitute ΛY^{-1} for P in this equation to obtain

$$\dot{P} = A\Lambda Y^{-1} + \Lambda Y^{-1} A^T - \Lambda Y^{-1} C^T R^{-1} C\Lambda Y^{-1} + Q \tag{8.53}$$

Post-multiply both sides of this equation by Y to obtain

$$\dot{P}Y = A\Lambda + \Lambda Y^{-1} A^T Y - \Lambda Y^{-1} C^T R^{-1} C\Lambda + QY \tag{8.54}$$

Now we can equate the right sides of Equations (8.51) and (8.54) to obtain

$$\begin{aligned} \dot{\Lambda} - \Lambda Y^{-1}\dot{Y} &= A\Lambda + \Lambda Y^{-1} A^T Y - \Lambda Y^{-1} C^T R^{-1} C\Lambda + QY \\ \dot{\Lambda} &= A\Lambda + QY + \Lambda Y^{-1}(\dot{Y} + A^T Y - C^T R^{-1} C\Lambda) \end{aligned} \tag{8.55}$$

This equation came from our original factorization of P, and if this equation reduces to $0 = 0$ then we know that the original factorization was valid. So if $\dot{Y} = C^T R^{-1} C\Lambda - A^T Y$, and $\dot{\Lambda} = A\Lambda + QY$, then our assumed factorization will be valid. These differential equations for Y and Λ can be combined as

$$\begin{aligned} \begin{bmatrix} \dot{\Lambda} \\ \dot{Y} \end{bmatrix} &= \begin{bmatrix} A & Q \\ C^T R^{-1} C & -A^T \end{bmatrix} \begin{bmatrix} \Lambda \\ Y \end{bmatrix} \\ &= J \begin{bmatrix} \Lambda \\ Y \end{bmatrix} \end{aligned} \tag{8.56}$$

where J is defined by the above equation. The initial conditions on Λ and Y can be chosen to be consistent with the initial condition on P as follows:

$$\begin{aligned} \Lambda(0) &= P(0) \\ Y(0) &= I \end{aligned} \tag{8.57}$$

Now suppose that A, Q, C, and R are constant (that is, we have an LTI system with constant process and measurement noise covariances). In this case J is constant and Equation (8.56) can be solved as

$$\begin{bmatrix} \Lambda(t+T) \\ Y(t+T) \end{bmatrix} = \exp(JT) \begin{bmatrix} \Lambda(t) \\ Y(t) \end{bmatrix} \tag{8.58}$$

This can be written as

$$\begin{bmatrix} \Lambda(t+T) \\ Y(t+T) \end{bmatrix} = \begin{bmatrix} \phi_{11}(T) & \phi_{12}(T) \\ \phi_{21}(T) & \phi_{22}(T) \end{bmatrix} \begin{bmatrix} \Lambda(t) \\ Y(t) \end{bmatrix} \tag{8.59}$$

where the ϕ_{ij} matrices are defined as the four $n \times n$ submatrices in $\exp(JT)$. From our original factorization assumption we have $\Lambda = PY$, so this equation can be written as

$$\begin{bmatrix} \Lambda(t+T) \\ Y(t+T) \end{bmatrix} = \begin{bmatrix} \phi_{11}(T) & \phi_{12}(T) \\ \phi_{21}(T) & \phi_{22}(T) \end{bmatrix} \begin{bmatrix} P(t)Y(t) \\ Y(t) \end{bmatrix} \tag{8.60}$$

This can be written as two separate equations:

$$\begin{aligned} \Lambda(t+T) &= \phi_{11}(T)P(t)Y(t) + \phi_{12}(T)Y(t) \\ Y(t+T) &= \phi_{21}(T)P(t)Y(t) + \phi_{22}(T)Y(t) \end{aligned} \tag{8.61}$$

Since $\Lambda(t+T) = P(t+T)Y(t+T)$, we can write the first equation as

$$P(t+T)Y(t+T) = \phi_{11}(T)P(t)Y(t) + \phi_{12}(T)Y(t) \tag{8.62}$$

Substituting for $Y(t+T)$ from Equation (8.61) in the above equation gives

$$\begin{aligned} P(t+T)\left[\phi_{21}(T)P(t)Y(t) + \phi_{22}(T)Y(t)\right] &= \phi_{11}(T)P(t)Y(t) + \phi_{12}(T)Y(t) \\ P(t+T)\left[\phi_{21}(T)P(t) + \phi_{22}(T)\right] &= \phi_{11}(T)P(t) + \phi_{12}(T) \end{aligned} \tag{8.63}$$

This equation is finally solved for $P(t+T)$ as

$$P(t+T) = \left[\phi_{11}(T)P(t) + \phi_{12}(T)\right]\left[\phi_{21}(T)P(t) + \phi_{22}(T)\right]^{-1} \tag{8.64}$$

This may be a faster way to solve for P instead of integrating the Riccati equation. Note that we do not have to worry about the integration step size with this method. This method can be used to propagate from $P(t)$ to $P(t+T)$ in a single equation, for any values t and T.

■ EXAMPLE 8.3

Suppose that we want to estimate a gyroscope drift rate ϵ (assumed to be constant) given measurements of the gyro angle θ. The system and measurement model can be written as

$$\begin{aligned} \dot{\theta} &= \epsilon \\ y &= \theta + v \\ \begin{bmatrix} \dot{\theta} \\ \dot{\epsilon} \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \epsilon \end{bmatrix} \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \epsilon \end{bmatrix} + v \\ v &\sim (0, R) \end{aligned} \tag{8.65}$$

Direct use of the differential Riccati equation from Equation (8.37) gives

$$\begin{aligned} \dot{P} &= AP + PA^T - PC^TR^{-1}CP + Q \\ \begin{bmatrix} \dot{P}_{11} & \dot{P}_{12} \\ \dot{P}_{12} & \dot{P}_{22} \end{bmatrix} &= \begin{bmatrix} 2P_{12} - P_{11}^2/R & P_{22} - P_{11}P_{12}/R \\ P_{22} - P_{11}P_{12}/R & -P_{12}^2/R \end{bmatrix} \end{aligned} \tag{8.66}$$

We can solve for P by performing three numerical integrations (recall that P is symmetric). However, it would be difficult to find a closed-form solution for $P(t)$ from these coupled differential equations. A transition matrix approach to this problem would proceed as follows, assuming that $P(0)$ is diagonal. We suppose that P is factored as $P = \Lambda Y^{-1}$, where Λ and Y are 2×2 matrices. The initial conditions on $\Lambda(t)$ and $Y(t)$ can be chosen as

$$\begin{aligned} \Lambda(0) &= P(0) \\ &= \begin{bmatrix} P_{11}(0) & 0 \\ 0 & P_{22}(0) \end{bmatrix} \\ Y(0) &= I \end{aligned} \tag{8.67}$$

The differential equation for $\Lambda(t)$ and $Y(t)$ is given as

$$\begin{bmatrix} \dot{\Lambda} \\ \dot{Y} \end{bmatrix} = J \begin{bmatrix} \Lambda \\ Y \end{bmatrix} \tag{8.68}$$

where the matrix J is computed as

$$\begin{aligned} J &= \begin{bmatrix} A & Q \\ C^T R^{-1} C & -A^T \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1/R & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \end{aligned} \tag{8.69}$$

The transition matrix for the differential equation for Λ and Y is computed as

$$\begin{aligned} \exp(Jt) &= \begin{bmatrix} 1 & t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ t/R & t^2/2R & 1 & 0 \\ -t^2/2R & -t^3/6R & -t & 1 \end{bmatrix} \\ &= \begin{bmatrix} \phi_{11}(t) & \phi_{12}(t) \\ \phi_{21}(t) & \phi_{22}(t) \end{bmatrix} \end{aligned} \tag{8.70}$$

where the $\phi_{ij}(t)$ terms are 2×2 matrix partitions. The Riccati equation solution is obtained from Equation (8.64) as

$$\begin{aligned} P(t) &= [\phi_{11}(t)P(0) + \phi_{12}(t)][\phi_{21}(t)P(0) + \phi_{22}(t)]^{-1} \\ &= \begin{bmatrix} P_{11}(0) & tP_{22}(0) \\ 0 & P_{22}(0) \end{bmatrix} \frac{1}{\Delta} \begin{bmatrix} 12R^2 - 2t^3 P_{22}(0) & -6Rt^2 P_{22}(0) \\ 12R^2 t + 6t^2 P_{11}(0) & 12R^2 + 12tP_{11}(0) \end{bmatrix} \end{aligned} \tag{8.71}$$

where Δ is given as

$$\Delta = 12R^2 + P_{11}(0)P_{22}(0)t^4 + 12P_{11}(0)tR + 4P_{22}(0)t^3 R \tag{8.72}$$

Carrying out the multiplication and some algebra gives the Riccati equation solution as

$$\begin{aligned} P(t) &= \begin{bmatrix} P_{11}(t) & P_{12}(t) \\ P_{12}(t) & P_{22}(t) \end{bmatrix} \\ P_{11}(t) &= \frac{1}{\Delta} 4R \left[P_{11}(0)P_{22}(0)t^3 + 3P_{11}(0)R + 3t^2 P_{22}(0)R \right] \\ P_{12}(t) &= \frac{1}{\Delta} 6RP_{22}(0)t \left[P_{11}(0)t + 2R \right] \\ P_{22}(t) &= \frac{1}{\Delta} 12RP_{22}(0) \left[P_{11}(0)t + R \right] \end{aligned} \tag{8.73}$$

With the transition matrix approach we have obtained a closed-form solution for $P(t)$, something that was not possible with a direct approach to the Riccati equation. In the special case that our initial uncertainty is infinite, we can

further simplify $P(t)$ as

$$\begin{aligned}
\lim_{P(0)\to\infty} \Delta &= P_{11}(0)P_{22}(0)t^4 \\
\lim_{P(0)\to\infty} P(t) &= \begin{bmatrix} 4R/t & 6R/t^2 \\ 6R/t^2 & 12R/t^3 \end{bmatrix} \\
\lim_{t\to\infty}\left[\lim_{P(0)\to\infty} P(t)\right] &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned} \tag{8.74}$$

That is, our uncertainty goes to zero as time goes to infinity. This occurs because the process noise is zero (i.e., we are estimating a constant). Since $K = PC^TR^{-1}$, we see that the Kalman gain also goes to zero as time goes to infinity. This simply means that eventually we get so many measurements that our knowledge is complete. Additional measurements cannot give us any new information, so we ignore additional measurements.

$\nabla\nabla\nabla$

8.3.2 The Chandrasekhar algorithm

Recall the differential Riccati equation for the continuous-time Kalman filter from Equation (8.37):

$$\dot{P} = AP + PA^T - PC^TR^{-1}CP + Q \tag{8.75}$$

If P were not symmetric then the numerical computation of P would require n^2 integrations. However, since $P = P^T$ the computation of P requires only $n(n+1)/2$ integrations. This can still be computationally taxing, especially for problems with small time constants. The Chandrasekhar algorithm gives computational savings in some circumstances. The algorithm is based on the work of the Nobel prize winning astrophysicist Subramanan Chandrasekhar, who used similar algorithms to solve computationally difficult astrophysics problems in the 1940s [Cha47, Cha48]. Chandrasekhar's algorithms were applied to Kalman filtering in [Kai73, Kai00]. The Chandrasekhar algorithm applies only when A, C, R, and Q are constant.

8.3.2.1 The Chandrasekhar algorithm derivation Consider the continuous-time differential equation for the state estimate, assuming that the original system is time-invariant and the Kalman gain K is a constant:

$$\begin{aligned}
\dot{\hat{x}} &= A\hat{x} + K(y - C\hat{x}) \\
&= (A - KC)\hat{x} + Ky
\end{aligned} \tag{8.76}$$

The measurement y is the output of the system, but it is the input to the filter. Consider the zero-input Kalman filter (i.e., $y = 0$).

$$\dot{\hat{x}} = (A - KC)\hat{x} \tag{8.77}$$

This equation has the solution

$$\begin{aligned}
\hat{x}(t) &= \exp[(A - KC)t]\hat{x}(0) \\
&= \phi(t)\hat{x}(0)
\end{aligned} \tag{8.78}$$

where $\phi(t)$ is the state transition matrix of the filter and is defined by the above equation. From the definition of $\phi(t)$ as a state transition matrix we know that

$$\begin{aligned} \dot{\phi} &= (A - KC)\phi \\ \phi(0) &= I \end{aligned} \tag{8.79}$$

We can differentiate both sides of Equation (8.75) to obtain

$$\begin{aligned} \ddot{P} &= A\dot{P} + \dot{P}A^T - \dot{P}C^T R^{-1} CP - PC^T R^{-1} C\dot{P} \\ &= A\dot{P} + \dot{P}A^T - \dot{P}C^T K^T - KC\dot{P} \\ &= (A - KC)\dot{P} + \dot{P}(A - KC)^T \end{aligned} \tag{8.80}$$

Now note that for a general time-varying matrix $Y(t)$, if $\dot{Y} = AY + YA^T$, where A is a constant matrix, then $Y(t) = \exp(At)Y(0)\exp(A^T t)$ (see Problem 8.2). Therefore, we can solve the above equation for $\dot{P}$ as

$$\dot{P} = \phi\dot{P}(0)\phi^T \tag{8.81}$$

where $\dot{P}(0)$ is obtained from Equation (8.75) as

$$\dot{P}(0) = AP(0) + P(0)A^T - P(0)C^T R^{-1} CP(0) + Q \tag{8.82}$$

The symmetric matrix $\dot{P}(0)$ can be factored as follows (see Section 8.3.2.2):

$$\dot{P}(0) = M_1 M_1^T - M_2 M_2^T \tag{8.83}$$

$\dot{P}(0)$ is an $n \times n$ matrix. The rank of $\dot{P}(0)$ is $\alpha \leq n$. Since $\dot{P}(0)$ is symmetric, all of its eigenvalues are real. The number of positive eigenvalues of $\dot{P}(0)$ is β, and the number of negative eigenvalues is $(\alpha - \beta)$. Matrix M_1 is an $n \times \beta$ matrix, and M_2 is an $n \times (\alpha - \beta)$ matrix. From the previous three equations we can write

$$\begin{aligned} \dot{P} &= \phi\dot{P}(0)\phi^T \\ &= \phi(M_1 M_1^T - M_2 M_2^T)\phi^T \\ &= \phi M_1 M_1^T \phi^T - \phi M_2 M_2^T \phi^T \end{aligned} \tag{8.84}$$

Now define the matrices Y_1 and Y_2 as

$$\begin{aligned} Y_1 &= \phi M_1 \\ Y_2 &= \phi M_2 \end{aligned} \tag{8.85}$$

Then the $\dot{P}$ equation can be written as

$$\dot{P} = Y_1 Y_1^T - Y_2 Y_2^T \tag{8.86}$$

Also, from the definition of Y_1 we can see that

$$\begin{aligned} Y_1(0) &= \phi(0)M_1 = M_1 \\ \dot{Y}_1 &= \dot{\phi} M_1 \\ &= (A - KC)\phi M_1 \\ &= (A - KC)Y_1 \end{aligned} \tag{8.87}$$

Similarly, we see that

$$\begin{aligned} Y_2(0) &= \phi(0)M_2 = M_2 \\ \dot{Y}_2 &= (A - KC)Y_2 \end{aligned} \tag{8.88}$$

Recall from Equation (8.37) that $K = PC^T R^{-1}$. Therefore, a differential equation and initial condition for K can be written as

$$\begin{aligned} \dot{K} &= \dot{P}C^T R^{-1} \\ &= (Y_1 Y_1^T - Y_2 Y_2^T)C^T R^{-1} \\ K(0) &= P(0)C^T R^{-1} \end{aligned} \tag{8.89}$$

To compute K from its differential equation we need to integrate three equations.

1. We need to integrate Y_1 from Equation (8.87), where Y_1 is an $n \times \beta$ matrix.
2. We need to integrate Y_2 from Equation (8.88), where Y_2 is an $n \times (\alpha - \beta)$ matrix.
3. We need to integrate K from Equation (8.89), where K is an $n \times r$ matrix (r is the number of measurements of the system).

So we need to perform a total of $n(\alpha + r)$ integrations. The direct computation of P from the differential Riccati equation requires $n(n+1)/2$ integrations. So if $2(\alpha + r) < (n+1)$ then the Chandrasekhar algorithm reduces the computational effort of solving the differential Riccati equation.

The Chandrasekhar algorithm

The Chandrasekhar algorithm can be summarized as follows.

1. Compute $\dot{P}(0)$.
2. Use the method of Section 8.3.2.2 to find M_1 and M_2 matrices that satisfy $\dot{P}(0) = M_1 M_1^T - M_2 M_2^T$.
3. Initialize $Y_1(0) = M_1$, $Y_2(0) = M_2$, and $K(0) = P(0)C^T R^{-1}$.
4. Integrate K, Y_1, and Y_2 as follows:

$$\begin{aligned} \dot{K} &= (Y_1 Y_1^T - Y_2 Y_2^T)C^T R^{-1} \\ \dot{Y}_1 &= (A - KC)Y_1 \\ \dot{Y}_2 &= (A - KC)Y_2 \end{aligned} \tag{8.90}$$

8.3.2.2 Chandrasekhar factorization The derivation of the Chandrasekhar algorithm requires the factorization of $\dot{P}(0)$ as shown in Equation (8.83):

$$\dot{P}(0) = M_1 M_1^T - M_2 M_2^T \tag{8.91}$$

$\dot{P}(0)$ is an $n \times n$ matrix with rank $\alpha \leq n$. The number of positive eigenvalues of $\dot{P}(0)$ is β, and the number of negative eigenvalues is $(\alpha - \beta)$. Matrix M_1 is an $n \times \beta$

matrix, and M_2 is an $n \times (\alpha - \beta)$ matrix. In this section, we will show one way to perform that factorization.

Since $\dot{P}(0)$ is symmetric, all of its eigenvalues are real. We can therefore write the Jordan form of $\dot{P}(0)$ as

$$\begin{aligned} \dot{P}(0) &= SDS^T \\ &= \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \begin{bmatrix} D_1 & 0 & 0 \\ 0 & -D_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} S_{11}^T & S_{21}^T & S_{31}^T \\ S_{12}^T & S_{22}^T & S_{32}^T \\ S_{13}^T & S_{23}^T & S_{33}^T \end{bmatrix} \end{aligned} \tag{8.92}$$

S is an orthogonal matrix whose columns comprise the eigenvectors of $\dot{P}(0)$. The $\beta \times \beta$ matrix D_1 is a diagonal matrix whose entries are the positive eigenvalues of $\dot{P}(0)$. The $(\alpha - \beta) \times (\alpha - \beta)$ matrix D_2 is a diagonal matrix whose entries are the magnitudes of the negative eigenvalues of $\dot{P}(0)$. Multiplying out the above equation results in

$$\dot{P}(0) = N_1 + N_2 \tag{8.93}$$

where N_1 and N_2 are given as

$$\begin{aligned} N_1 &= \begin{bmatrix} S_{11}D_1S_{11}^T & S_{11}D_1S_{21}^T & S_{11}D_1S_{31}^T \\ S_{21}D_1S_{11}^T & S_{21}D_1S_{21}^T & S_{21}D_1S_{31}^T \\ S_{31}D_1S_{11}^T & S_{31}D_1S_{21}^T & S_{31}D_1S_{31}^T \end{bmatrix} \\ &= \begin{bmatrix} S_{11} \\ S_{21} \\ S_{31} \end{bmatrix} D_1 \begin{bmatrix} S_{11}^T \\ S_{21}^T \\ S_{31}^T \end{bmatrix} \\ N_2 &= \begin{bmatrix} S_{12}D_2S_{12}^T & S_{12}D_2S_{22}^T & S_{12}D_2S_{32}^T \\ S_{22}D_2S_{12}^T & S_{22}D_2S_{22}^T & S_{22}D_2S_{32}^T \\ S_{32}D_2S_{12}^T & S_{32}D_2S_{22}^T & S_{32}D_2S_{32}^T \end{bmatrix} \\ &= \begin{bmatrix} S_{12} \\ S_{22} \\ S_{32} \end{bmatrix} D_2 \begin{bmatrix} S_{12}^T \\ S_{22}^T \\ S_{32}^T \end{bmatrix} \end{aligned} \tag{8.94}$$

Note that N_1 is the product of an $n \times \beta$ matrix, the $\beta \times \beta$ matrix D_1, and a $\beta \times n$ matrix. N_1 can therefore be written as

$$N_1 = M_1 M_1^T \tag{8.95}$$

where M_1 is the $n \times \beta$ matrix

$$M_1 = \begin{bmatrix} S_{11} \\ S_{21} \\ S_{31} \end{bmatrix} \sqrt{D_1} \tag{8.96}$$

A similar development can be followed to see that M_2 is the $n \times (\alpha - \beta)$ matrix

$$M_2 = \begin{bmatrix} S_{12} \\ S_{22} \\ S_{32} \end{bmatrix} \sqrt{D_2} \tag{8.97}$$

8.3.3 The square root filter

The early days of Kalman filtering in the 1960s saw a lot of successful applications. But there were also some problems in implementation, many due to numerical difficulties. The differential Riccati equation solution $P(t)$ should theoretically always be a symmetric positive semidefinite matrix (since it is a covariance matrix). But numerical problems in computer implementations sometimes led to P matrices that became indefinite or nonsymmetric. This was often because of the short word lengths in the computers of the 1960s [Sch81].[1] This led to a lot of research during that decade related to numerical implementations.

Now that computers have become so much more capable, we don't have to worry about numerical problems as often. Nevertheless, numerical issues still arise in finite word-length implementations of algorithms, especially in embedded systems.[2] The square root filter was developed in order to effectively increase the numerical precision of the Kalman filter and hence mitigate numerical difficulties in implementations.

The square root filter is based on the idea of finding an S matrix such that $P = SS^T$. The S matrix is then called a square root of P. Note that the definition of the square root of P is *not* that $P = S^2$, but rather $P = SS^T$. Also note that this definition of the matrix square root is not standard. Some books and papers define the matrix square root as $P = S^2$, others define it as $P = S^T S$, and others define it as $P = SS^T$. The latter definition is the one that we will use in this book. Finally, note that the square root of a matrix may not be unique; that is, there may be more than one solution for S in the equation $P = SS^T$. (This is analogous to the existence of multiple square roots for scalars. For example, the number 4 has two square roots: $+2$ and -2.) Sections 6.3 and 6.4 contain a discussion of square root filtering for the discrete-time Kalman filter.

After defining S as the square root of P, we will integrate S instead of P in our Kalman filter solution. This requires more computational effort but it doubles the precision of the filter and helps prevent numerical problems. From the differential Riccati equation of Equation (8.37), and the definition of S, we obtain

$$\begin{aligned} \dot{P} &= AP + PA^T - PC^T R^{-1} CP + Q \\ \dot{S}S^T + S\dot{S}^T &= ASS^T + SS^T A^T - SS^T C^T R^{-1} CSS^T + Q \end{aligned} \tag{8.98}$$

Now premultiply both sides by S^{-1} and postmultiply by S^{-T} to obtain

$$\begin{aligned} S^{-1}\dot{P}S^{-T} &= S^{-1}\dot{S} + \dot{S}^T S^{-T} \\ &= S^{-1}AS + S^T A^T S^{-T} - S^T C^T R^{-1} CS + S^{-1} Q S^{-T} \end{aligned} \tag{8.99}$$

Since P is symmetric positive definite, we can always find an upper triangular S such that $P = SS^T$ [Gol89, Moo00]. For example, consider the following matrices:

[1] The United States' Apollo space program of the 1960s resulted in the first man on the moon in 1969. The Apollo spacecraft guidance computer had a word length of 16 bits [Bat82], which corresponds to 4.8 decimal digits of precision.

[2] Most microcontrollers in the first decade of the 21st century have 16 bit words, and 8 bit microcontrollers still comprise a large share of the market.

$$
\begin{aligned}
P &= \begin{bmatrix} 5 & 2 \\ 2 & 1 \end{bmatrix} \\
S &= \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}
\end{aligned} \tag{8.100}
$$

P is symmetric positive definite, S is upper triangular, and $P = SS^T$. It can be shown that if S is upper triangular, then $\dot{S}$ and S^{-1} are also upper triangular (see Problem 8.4). Also, the product of upper triangular matrices is another upper triangular matrix (see Problem 8.5). Therefore, the product $S^{-1}\dot{S}$ is upper triangular. Similarly, since $\dot{S}^T$ and S^{-T} are lower triangular, the product $\dot{S}^T S^{-T}$ is lower triangular. That is,

$$
\begin{aligned}
S^{-1}\dot{S} &= M_U \\
\dot{S}^T S^{-T} &= M_L
\end{aligned} \tag{8.101}
$$

where M_U and M_L denote upper triangular and lower triangular matrices. From this we can obtain

$$
\dot{S} = SM_U \tag{8.102}
$$

Now we can use Equations (8.99) and (8.101) to find

$$
\begin{aligned}
S^{-1}\dot{P}S^{-T} &= S^{-1}\dot{S} + \dot{S}^T S^{-T} \\
&= M_U + M_L
\end{aligned} \tag{8.103}
$$

So we see that M_U is the upper triangular portion of $S^{-1}\dot{P}S^{-T}$. This gives us the square root algorithm as follows.

The continuous-time square root Kalman filter

1. The initialization step consists of computing the upper triangular $S(0)$ such that $S(0)S^T(0) = P(0)$.
2. At each time step compute $\dot{P}$ from the differential Riccati equation, and then compute M_U as the upper triangular portion of $S^{-1}\dot{P}S^{-T}$.
3. Use $\dot{S} = SM_U$ to integrate S to the next time step.
4. Use the equation $K = PC^T R^{-1} = SS^T C^T R^{-1}$ to compute the Kalman gain.

This is more computationally expensive than a straightforward integration of the differential Riccati equation, but it is also more numerically stable. The numerical benefits of square root filtering are discussed in more detail in Section 6.3.

8.4 GENERALIZATIONS OF THE CONTINUOUS-TIME FILTER

In this section, we will discuss some generalizations of the continuous-time Kalman filter, just as we did in Chapter 7 for the discrete-time Kalman filter. The continuous-time filter was derived under the assumptions that the process and measurement noise was uncorrelated, and that the process and measurement noise was white. We will consider the case in which the process and measurement noise are correlated in Section 8.4.1, and the case in which the measurement noise is colored in Section 8.4.2.

8.4.1 Correlated process and measurement noise

Consider the continuous-time system

$$\begin{aligned} \dot{x} &= Ax + w \\ w &\sim (0, Q) \\ y &= Cx + v \\ v &\sim (0, R) \\ E[w(t)v^T(\tau)] &= M\delta(t-\tau) \end{aligned} \tag{8.104}$$

Since $y - Cx - v = 0$ we can write the system dynamics as

$$\begin{aligned} \dot{x} &= Ax + w + MR^{-1}(y - Cx - v) \\ &= (A - MR^{-1}C)x + MR^{-1}y + (w - MR^{-1}v) \\ &= \tilde{A}x + \tilde{u} + \tilde{w} \end{aligned} \tag{8.105}$$

where $\tilde{A}$, $\tilde{u}$, and $\tilde{w}$ are defined by the above equation. Note that $\tilde{u}$ is a known input to the $\dot{x}$ equation, and $\tilde{w}$ is a new process noise term. The cross covariance between the new process noise $\tilde{w}$ and the measurement noise v can be found as

$$\begin{aligned} E(\tilde{w}v^T) &= E[(w - MR^{-1}v)v^T] \\ &= E(wv^T) - MR^{-1}E(vv^T) \\ &= M - M \\ &= 0 \end{aligned} \tag{8.106}$$

So $\tilde{w}$ and v are uncorrelated. The covariance of the new process noise $\tilde{w}$ can be found as

$$\begin{aligned} \tilde{Q} &= E(\tilde{w}\tilde{w}^T) \\ &= E[(w - MR^{-1}v)(w - MR^{-1}v)^T] \\ &= Q - MR^{-1}M^T - MR^{-1}M^T + MR^{-1}M^T \\ &= Q - MR^{-1}M^T \end{aligned} \tag{8.107}$$

The differential Riccati equation for Kalman filter for the system given in Equation (8.105) is given by

$$\begin{aligned} \dot{P} &= \tilde{A}P + P\tilde{A}^T - PC^TR^{-1}CP + \tilde{Q} \\ &= (A - MR^{-1}C)P + P(A - MR^{-1}C)^T - PC^TR^{-1}CP + \\ &\quad Q - MR^{-1}M^T \end{aligned} \tag{8.108}$$

If we define $\tilde{K}$ as

$$\begin{aligned} \tilde{K} &= K + MR^{-1} \\ &= PC^TR^{-1} + MR^{-1} \\ &= (PC^T + M)R^{-1} \end{aligned} \tag{8.109}$$

then the differential Riccati equation becomes

$$\dot{P} = AP + PA^T + Q - \tilde{K}R\tilde{K}^T \tag{8.110}$$

The differential equation for the state estimate can be written as

$$\begin{aligned}\dot{\hat{x}} &= \tilde{A}\hat{x} + \tilde{u} + K(y - C\hat{x}) \\ &= (A - MR^{-1}C)\hat{x} + MR^{-1}y + K(y - C\hat{x}) \\ &= A\hat{x} - MR^{-1}C\hat{x} + MR^{-1}y + (\tilde{K} - MR^{-1})(y - C\hat{x}) \\ &= A\hat{x} + \tilde{K}(y - C\hat{x}) \end{aligned} \tag{8.111}$$

We see that the introduction of correlation between the process and measurement noise has the effect of simply modifying the Kalman gain. The state-update equation and the differential Riccati equation retain the same form as for the standard Kalman filter. The Kalman filter for correlated process and measurement noise can be summarized as follows.

The continuous-time Kalman filter with correlated noise

1. The system dynamics and measurement equation are given as

$$\begin{aligned}\dot{x} &= Ax + w \\ w &\sim (0, Q) \\ y &= Cx + v \\ v &\sim (0, R) \\ E[w(t)v^T(\tau)] &= M\delta(t - \tau) \end{aligned} \tag{8.112}$$

2. The continuous-time Kalman filter is given as

$$\begin{aligned}\dot{P} &= AP + PA^T + Q - KRK^T \\ K &= (PC^T + M)R^{-1} \\ \dot{\hat{x}} &= A\hat{x} + K(y - C\hat{x}) \end{aligned} \tag{8.113}$$

Note that (as expected) this filter reduces to the standard continuous-time filter of Equation (8.37) if the process and measurement noise are uncorrelated (i.e., $M = 0$). This filter can therefore be considered as a general formulation of the continuous-time Kalman filter, with the situation $M = 0$ as a special case.

8.4.2 Colored measurement noise

In this section we will derive the Kalman filter when the measurement noise is not white. Suppose we have the system

$$\begin{aligned}\dot{x} &= Ax + w \\ w &\sim (0, Q) \\ y &= Cx + v \\ \dot{v} &= Nv + \phi \\ \phi &\sim (0, \Phi) \end{aligned} \tag{8.114}$$

We will assume that w and ϕ are uncorrelated white noise processes. We could augment v onto the state vector (as suggested in Section 7.2.2 for discrete-time systems), but then the covariance of the measurement noise of the augmented system

would be singular, which could potentially cause numerical problems in the Kalman filter implementation. Instead, we will define a new signal as

$$\begin{aligned}
\tilde{y} &= \dot{y} - Ny \\
&= \dot{C}x + C\dot{x} + \dot{v} - N(Cx + v) \\
&= \dot{C}x + C(Ax + w) + (Nv + \phi) - N(Cx + v) \\
&= (\dot{C} + CA - NC)x + (Cw + \phi) \\
&= \tilde{C}x + \tilde{v}
\end{aligned} \tag{8.115}$$

where $\tilde{C}$ and $\tilde{v}$ are defined by the above equation. Note that $\tilde{v}$ is a white noise process (since w and ϕ are uncorrelated and white). So we have defined a new measurement equation that has white noise, but this is at the expense of creating a correlation between the process noise w and the new measurement noise $\tilde{v}$. The correlation can be obtained as

$$\begin{aligned}
E[w(t)\tilde{v}^T(\tau)] &= E\left[w(t)(Cw(\tau) + \phi(\tau))^T\right] \\
&= QC^T\delta(t - \tau) + 0 \\
&= M\delta(t - \tau)
\end{aligned} \tag{8.116}$$

where the cross correlation matrix M is defined by the above equation. The covariance of the new measurement noise $\tilde{v}$ can be obtained as

$$\begin{aligned}
E(\tilde{v}\tilde{v}^T) &= E[(Cw + \phi)(Cw + \phi)^T] \\
\tilde{R} &= CQC^T + \Phi
\end{aligned} \tag{8.117}$$

So we have defined a new measurement equation with white noise. We have the correlation between the process noise and the new measurement noise in Equation (8.116), and the covariance of the new measurement noise in Equation (8.117). Now we can use the results from Section 8.4.1 which discussed Kalman filtering for systems with correlated process and measurement noise. The Kalman filter can be written from Equation (8.113) as

$$\begin{aligned}
\dot{P} &= AP + PA^T + Q - K\tilde{R}K^T \\
K &= (PC^T + M)\tilde{R}^{-1} \\
\dot{\hat{x}} &= A\hat{x} + K(\tilde{y} - \tilde{C}\hat{x}) \\
&= A\hat{x} + K(\dot{y} - Ny - \tilde{C}\hat{x})
\end{aligned} \tag{8.118}$$

However, the new measurement that we defined in Equation (8.115) could cause some problems. The original measurement y is already a noisy measurement, so the new measurement (which contains $\dot{y}$) will be even more noisy. How can we avoid the use of $\dot{y}$ in the filter? We can attack this problem by looking at the derivative of the product Ky as follows:

$$\begin{aligned}
\frac{d(Ky)}{dt} &= \dot{K}y + K\dot{y} \\
K\dot{y} &= \frac{d(Ky)}{dt} - \dot{K}y
\end{aligned} \tag{8.119}$$

The dynamic equation for the state estimate in Equation (8.118) can then be written as follows:

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + \frac{d(Ky)}{dt} - \dot{K}y - K(Ny + \tilde{C}\hat{x}) \\ \dot{\hat{x}} - \frac{d(Ky)}{dt} &= (A - K\tilde{C})\hat{x} - (\dot{K} + KN)y \end{aligned} \tag{8.120}$$

Now define a new signal z as

$$z = \hat{x} - Ky \tag{8.121}$$

Differentiating z results in the right side of Equation (8.120):

$$\dot{z} = (A - K\tilde{C})\hat{x} - (\dot{K} + KN)y \tag{8.122}$$

Here we have an equation for $\dot{z}$ that we can integrate to solve for z. We can then use our solution for z in Equation (8.121) to solve for $\hat{x}$. So the only signal we have to differentiate in the Kalman filter algorithm is the Kalman gain K, because we need $\dot{K}$ in the computation of $\dot{z}$ above. However, this differentiation should be much easier than differentiating y, because we expect the Kalman gain K to be much smoother than the noisy measurement y. The Kalman filter for the case of colored measurement noise can be summarized as follows.

The continuous-time Kalman filter with colored measurement noise

1. The system and measurement equations are given as

$$\begin{aligned} \dot{x} &= Ax + w \\ w &\sim (0, Q) \\ y &= Cx + v \\ \dot{v} &= Nv + \phi \\ \phi &\sim (0, \Phi) \end{aligned} \tag{8.123}$$

   where w and ϕ and uncorrelated white noise processes.

2. Make the following matrix definitions:

$$\begin{aligned} \tilde{C} &= \dot{C} + CA - NC \\ \tilde{R} &= CQC^T + \Phi \\ M &= QC^T \end{aligned} \tag{8.124}$$

3. Initialize the Kalman filter as

$$\begin{aligned} K(0) &= [P(0)C^T + M]\tilde{R}^{-1} \\ z(0) &= \hat{x}(0) - K(0)y(0) \end{aligned} \tag{8.125}$$

4. Integrate P, K, and z using the following equations:

$$\begin{aligned} \dot{P} &= AP + PA^T + Q - K\tilde{R}K^T \\ \dot{K} &= \frac{d}{dt}[(PC^T + M)\tilde{R}^{-1}] \\ \dot{z} &= (A - K\tilde{C})\hat{x} - (\dot{K} + KN)y \end{aligned} \tag{8.126}$$

Note that the $\dot{K}$ equation can be simplified to the following if Q, C, and Φ are constant:

$$\dot{K} = \dot{P}C^T\tilde{R}^{-1} \tag{8.127}$$

5. Compute the state estimate as

$$\hat{x} = z + Ky \tag{8.128}$$

■ **EXAMPLE 8.4**

Suppose that it is known that a continuous-time measurement $v(t)$ has a total power of 1 watt and a power spectrum that is bandlimited to frequencies below 10 Hz. In this example, we will use our knowledge of the frequency content of $v(t)$ to obtain a dynamic model for $v(t)$. The power spectrum $S_v(\omega)$ can be plotted as shown in Figure 8.1. The magnitude of the spectrum, $1/40\pi$, is obtained by realizing that the total power of the signal (1 watt) is equal to the integral from $-\infty$ to $+\infty$ of $S_v(\omega)$, and $S_v(\omega)$ is an even function of ω. The spectrum shown in Figure 8.1 can be approximated as

$$\begin{aligned} S_v(\omega) &\approx \frac{1/2}{\omega^2 + (20\pi)^2} \\ &= \left(\frac{1}{j\omega + 20\pi}\right)\left(\frac{1}{-j\omega + 20\pi}\right)\left(\frac{1}{2}\right) \\ &= G(\omega)G(-\omega)S_\phi(\omega) \end{aligned} \tag{8.129}$$

This shows that $v(t)$ is the output of a linear system with a transfer function of $G(\omega)$ and an input of $\phi(t)$, where $\phi(t)$ is white noise with a variance of $1/2$ (see Equation 3.75). This can be written in the $s$domain and then translated to the time domain as follows:

$$\begin{aligned} V(s) &= G(s)\Phi(s) \\ &= \frac{\Phi(s)}{s + 20\pi} \\ sV(s) + 20\pi V(s) &= \Phi(s) \\ sV(s) &= -20\pi V(s) + \Phi(s) \\ \dot{v} &= -20\pi v + \phi \end{aligned} \tag{8.130}$$

where $\phi(t)$ is white noise with variance $\Phi = 1/2$. Additional discussion and examples of this method can be found in [Bur99].

∇∇∇

8.5 THE STEADY-STATE CONTINUOUS-TIME KALMAN FILTER

In some situations, the Kalman filter converges to an LTI filter. If this is the case then we can often get good filtering performance by using a constant Kalman gain K in the filter. Then we do not have to worry about integrating the differential Riccati equation to solve for P and we do not have to worry about updating K in

Figure 8.1 Power spectrum of bandlimited measurement noise for Example 8.4.

real time. This can provide a large savings in filter complexity and computational effort at the cost of only a small sacrifice of performance. In this section, we discuss the conditions under which the continuous-time filter converges to an LTI filter, and the steady-state filter's relationship to Wiener filtering and optimal control.

8.5.1 The algebraic Riccati equation

Recall from Equation (8.37) that the differential Riccati equation is given as

$$\dot{P} = -PC^T R^{-1} CP + AP + PA^T + Q \tag{8.131}$$

If A, C, Q, and R are constant (i.e., the system and measurement equations form an LTI system with constant noise covariances) then P may reach a steady-state value and $\dot{P}$ may eventually reach zero. This implies that

$$-PC^T R^{-1} CP + AP + PA^T + Q = 0 \tag{8.132}$$

This is called an algebraic Riccati equation (ARE). To be more specific, it is called a continuous ARE (CARE).[3]

The ARE solution may not always exist, and even if it does exist it may not result in a stable Kalman filter. We will summarize the most important Riccati equation convergence results below, but first we need to define what it means for a system to be controllable on the imaginary axis.

Definition 12 *The matrix pair (A, B) is controllable on the imaginary axis if there exists some matrix K such that $(A - BK)$ does not have any eigenvalues on the imaginary axis.*

[3]In the MATLAB Control System Toolbox the CARE can be solved by invoking the command $P = \text{CARE}(A^T, C^T, Q, R)$. The reason that the transposes are required is that MATLAB's CARE command is designed to solve the ARE for continuous-time optimal control problems. When we use it to solve for the Kalman filtering problem we need to transpose the A and C matrices, as discussed in Section 8.5.3.

This is similar to the concept of controllability on the unit circle for discrete-time systems (see Section 7.3). Now we summarize the most important Riccati equation convergence results from [Kai00], where proofs are given. Recall that the ARE is given as

$$-PC^TR^{-1}CP + AP + PA^T + Q = 0 \tag{8.133}$$

We assume that $Q \geq 0$ and $R > 0$. We define G as any matrix such that $GG^T = Q$. The corresponding steady-state Kalman gain K is given as

$$K = PC^TR^{-1} \tag{8.134}$$

The steady-state Kalman filter is given as

$$\dot{\hat{x}} = (A - KC)\hat{x} + Ky \tag{8.135}$$

We say that the CARE solution P is stabilizing if it results in a stable steady-state filter. That is, P is defined as a stabilizing CARE solution if all of the eigenvalues of $(A - KC)$ have negative real parts.

Theorem 27 *The CARE has a unique positive semidefinite solution P if and only if both of the following conditions hold.*

1. *(A, C) is detectable.*
2. *(A, G) is stabilizable.*

Furthermore, the corresponding steady-state Kalman filter is stable. That is, the eigenvalues of $(A - KC)$ have negative real parts.

This theorem is analogous to Theorem 23 for discrete-time Kalman filters. The above theorem does not preclude the existence of CARE solutions that are negative definite or indefinite. If such solutions exist, then they would result in an unstable Kalman filter. If we weaken the stabilizability condition in the above theorem, we obtain the following.

Theorem 28 *The CARE has at least one positive semidefinite solution P if and only if both of the following conditions hold.*

1. *(A, C) is detectable.*
2. *(A, G) is controllable on the imaginary axis.*

Furthermore, exactly one of the positive semidefinite ARE solutions results in a stable steady-state Kalman filter.

This theorem is analogous to Theorem 24 for discrete-time Kalman filters. This theorem states conditions for the existence of exactly one stabilizing positive definite CARE solution. However, there may be additional CARE solutions (positive definite or otherwise) that result in unstable Kalman filters. If a time-varying Kalman filter is run in this situation, then the Kalman filter equations may converge to either a stable or an unstable filter, depending on the initial condition $P(0)$. If we strengthen the controllability condition of Theorem 28, we obtain the following.

Theorem 29 *The CARE has at least one positive definite solution P if and only if both of the following conditions hold.*

1. *(A, C) is detectable.*
2. *(A, G) is controllable in the closed left half plane.*

Furthermore, exactly one of the positive definite CARE solutions results in a stable steady-state Kalman filter.

This theorem is analogous to Theorem 25 for discrete-time Kalman filters. If we drop the controllability condition in the above two theorems, we obtain the following.

Theorem 30 *The CARE has at least one positive semidefinite solution P if (A, C) is detectable. Furthermore, at least one such solution results in a marginally stable steady-state Kalman filter.*

This theorem is analogous to Theorem 26 for discrete-time Kalman filters. Note that the resulting filter is only marginally stable, so it may have eigenvalues on the imaginary axis. Also note that this theorem poses a sufficient (not necessary) condition. That is, there may be a stable steady-state Kalman filter even if the conditions of the above theorem do not hold. Furthermore, even if the conditions of the theorem do hold, there may be CARE solutions that result in unstable Kalman filters.

Additional results related to the stability of the steady-state continuous-time filter can be found many places, including [Aok67, Buc67, Buc68, Kwa72]. Many practical Kalman filters are applied to systems that do not meet the conditions of the above theorems, but the filters still work well in practice.

■ **EXAMPLE 8.5**

In this example we consider the following two-state system that is taken from [Buc68, Chapter 5]:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix} x + w \\ y &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x + v \\ Q &= \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix} \\ R &= \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \end{aligned} \tag{8.136}$$

In the remainder of this example, we use the symbol G to denote any matrix such that $GG^T = Q$. The differential Riccati equation for the Kalman filter is given as

$$\dot{P} = -PC^T R^{-1} CP + AP + PA^T + Q \tag{8.137}$$

This can be written as the following three coupled differential equations.

$$
\begin{aligned}
\dot{p}_{11} &= 2a_1p_{11} - p_{11}^2/r_1 - p_{12}^2 + q_11 \\
\dot{p}_{12} &= (a_1 + a_2)p_{12} - p_{11}p_{12}/r_1 - p_{12}p_{22}/r_2 + q_{12} \\
\dot{p}_{22} &= 2a_2p_{22} - p_{12}^2/r_1 - p_{22}^2/r_2 + q_{22}
\end{aligned} \tag{8.138}
$$

We set these derivatives equal to zero to obtain the steady-state Riccati equation solution.

If $a_1 \neq a_2$ and $q_{12} \neq 0$, then (A, C) is detectable and (A, G) is stabilizable (see Problem 8.8). The results of Theorem 27 therefore apply to this situation. It can be shown that the unique positive semidefinite ARE solution in this case is

$$
\begin{aligned}
p_{11} &= r_1\left[a_1 + \left(\gamma_1 - \frac{p_{12}^2}{r_1r_2}\right)^{1/2}\right] \\
p_{22} &= r_2\left[a_2 + \left(\gamma_2 - \frac{p_{12}^2}{r_1r_2}\right)^{1/2}\right] \\
p_{12} &= q_{12}\left[\gamma_1 + \gamma_2 + 2\left(\gamma_1\gamma_2 - q_{12}^2/r_1r_2\right)^{1/2}\right]^{-1/2} \\
\gamma_1 &= \frac{q_{11}}{r_1} + a_1^2 \\
\gamma_2 &= \frac{q_{22}}{r_2} + a_2^2
\end{aligned} \tag{8.139}
$$

This results in a stable steady-state Kalman filter.

If $a_1 = a_2 < 0$, $q_{12} \neq 0$, and $|Q| = 0$, then (A, C) is detectable, and (A, G) is stabilizable (see Problem 8.9). The results of Theorem 27 therefore apply to this situation as well. It can be shown that the unique positive semidefinite ARE solution in this case is given as

$$
\begin{aligned}
p_{11} &= q_{11}/\gamma_3 \\
p_{22} &= q_{22}/\gamma_3 \\
p_{12} &= q_{12}/\gamma_3 \\
\gamma_3 &= -a_1 + \left(a_1^2 + q_{11}/r_1 + q_{22}/r_2\right)^{1/2}
\end{aligned} \tag{8.140}
$$

This results in a stable steady-state Kalman filter.

If $a_1 = a_2 > 0$, $q_{12} \neq 0$, and $|Q| = 0$, then (A, C) is detectable and (A, G) is controllable on the imaginary axis, but (A, G) is not stabilizable (see Problem 8.10). The results of Theorem 27 do not apply to this situation, but Theorem 28 does apply to this situation. it can be shown that Equations (8.139) and (8.140) are both positive semidefinite ARE solutions in this case. If we integrate Equation (8.138) we may come up with Equation (8.139) as the steady-state solution, or we may come up with Equation (8.140) as the steady-state solution, depending on the initial condition $P(0)$. However, only one of the solutions will result in a stable Kalman filter.[4]

To be more specific, consider the case $a_1 = a_2 = 1$, $q_{11} = q_{12} = q_{22} = 0$, and $r_1 = r_2 = 1$. For these values, we can simulate the differential Riccati

[4]If we use MATLAB's CARE function then we will get the stabilizing solution.

equations of Equation (8.138) to find the steady-state Riccati solution, the steady-state Kalman gain, and the steady-state estimator, as follows:

$$\begin{aligned} P &= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \text{ or } \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ K &= PC^T R^{-1} \\ &= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \text{ or } \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \dot{\hat{x}} &= (A - KC)\hat{x} + Ky \\ &= (-\hat{x} + Ky) \text{ or } (\hat{x} + Ky) \end{aligned} \tag{8.141}$$

The ARE solution depends on the initial condition $P(0)$. The first ARE solution results in a positive semidefinite ARE solution that gives a stable Kalman filter. The second ARE solution results in a positive semidefinite ARE solution that gives an unstable Kalman filter. This agrees with Theorem 28.

$\nabla\nabla\nabla$

8.5.2 The Wiener filter is a Kalman filter

Consider the steady-state continuous-time Kalman filter.

$$\dot{\hat{x}} = Ax + K(y - C\hat{x}) \tag{8.142}$$

Taking the Laplace transform of both sides of this equation gives

$$\begin{aligned} (sI - A + KC)\hat{X}(s) &= KY(s) \\ \hat{X}(s) &= (sI - A + KC)^{-1}KY(s) \end{aligned} \tag{8.143}$$

The transfer function from $y(t)$ to $\hat{x}(t)$ is identical to the transfer function of the Wiener filter [Buc68, Chapter 5],Sha82, [Sag71, Chapter 7]. In other words, the Wiener filter is a special case of the Kalman filter. The equivalence of discrete-time Wiener and Kalman filtering is discussed in [Men87].

■ **EXAMPLE 8.6**

Consider the scalar system given by

$$\begin{aligned} \dot{x} &= -x + w \\ y &= x + v \end{aligned} \tag{8.144}$$

where w and v are zero-mean, uncorrelated white noise processes with respective variances $Q = 2$ and $R = 1$. The steady-state Kalman filter for this system can be obtained by solving Equation (8.37) with $\dot{P} = 0$, from which we obtain

$$\dot{\hat{x}} = -\sqrt{3}\hat{x} + (\sqrt{3} - 1)y \tag{8.145}$$

Taking the Laplace transform of this estimator gives

$$(s + \sqrt{3})\hat{X}(s) = (\sqrt{3} - 1)Y(s) \tag{8.146}$$

In other words, the Kalman filter is equivalent to passing the measurement $y(t)$ through the transfer function $G(s)$, which is given as

$$G(s) = \frac{\sqrt{3}-1}{s+\sqrt{3}} \tag{8.147}$$

The impulse response of the Kalman filter is obtained by taking the inverse Laplace transform, which gives

$$g(t) = (\sqrt{3}-1)e^{-\sqrt{3}t}, \qquad t \geq 0 \tag{8.148}$$

Now we will obtain the power spectrum of the state by taking the Laplace transform of Equation (8.144). This gives

$$\begin{aligned} sX(s) &= -X(s) + W(s) \\ X(s) &= \frac{1}{s+1}W(s) \end{aligned} \tag{8.149}$$

We see that the state $x(t)$ can be obtained by passing the white noise $w(t)$ (which has a power spectrum $S_w(\omega) = Q = 2$) through the transfer function $L(s) = 1/(s+1)$. From Equation (3.75) we see how to compute the power spectrum of the output of a linear system. This gives the power spectrum of $x(t)$ as

$$\begin{aligned} S_x(\omega) &= L(-\omega)L(\omega)S_w(\omega) \\ &= \left(\frac{1}{-j\omega+1}\right)\left(\frac{1}{j\omega+1}\right)2 \\ &= \frac{2}{\omega^2+1} \end{aligned} \tag{8.150}$$

The causal Wiener filter for a signal with this power spectrum, corrupted by white measurement noise with a unity power spectrum, was obtained in Example 3.10. The Wiener filter was found to be identical to the steady-state Kalman filter of Equation (8.148). This example serves to illustrate the equivalence of Wiener filtering and steady-state Kalman filtering.

$\nabla\nabla\nabla$

8.5.3 Duality

It is interesting to note the duality between optimal estimation and optimal control. The optimal estimation problem begins with the system and measurement equations

$$\begin{aligned} \dot{x} &= Ax + w \\ w &\sim N(0, Q) \\ y &= Cx + v \\ v &\sim N(0, R) \end{aligned} \tag{8.151}$$

Recall that Q and R are symmetric matrices. The optimal estimation problem tries to find the state estimate $\hat{x}$ that minimizes the cost function

$$J_e = \int_0^{t_f} E[(x-\hat{x})^T(x-\hat{x})]\,dt \tag{8.152}$$

The optimal estimator (the Kalman filter) is given as

$$\begin{aligned} P_e(0) &= E[(x(0)-\hat{x}(0))(x(0)-\hat{x}(0))^T] \\ \dot{P}_e &= AP_e + P_eA^T - P_eC^TR^{-1}CP_e + Q \\ K_e &= P_eC^TR^{-1} \\ \dot{\hat{x}} &= A\hat{x} + K_e(y - C\hat{x}) \end{aligned} \tag{8.153}$$

The differential Riccati equation for the optimal estimator is integrated forward in time from its initial condition $P_e(0)$.

The optimal control problem begins with the system

$$\dot{x} = Ax + Cu \tag{8.154}$$

where u is the control variable. The finite-time optimal control problem tries to find the control u that minimizes the cost function

$$J_c = x^T\phi x\big|_{t_f} + \int_0^{t_f} (x^TQx + u^TRu)\,dt \tag{8.155}$$

ϕ, Q, and R (which are assumed to be symmetric positive definite matrices) provide user-specified weighting in the performance index. The optimal controller is given as

$$\begin{aligned} P_c(t_f) &= \phi(t_f) \\ \dot{P}_c &= -A^TP_c - P_cA + P_cCR^{-1}C^TP_c - Q \\ K_c &= R^{-1}C^TP_c \\ u &= -K_cx \end{aligned} \tag{8.156}$$

The differential Riccati equation for the optimal control problem is integrated backward in time from the final condition $P(t_f)$. Note the relationships between the optimal estimation solution of Equation (8.153) and the optimal control solution of Equation (8.156). The differential Riccati equations have the same form, except they are negatives of each other, and A and C are replaced by their transposes. The estimator gain K_e and the controller gain K_c have very similar forms. The Q and R covariance matrices in the estimation problems have duals in the cost function weighting matrices of the optimal control problem.

The dual relationship between the estimation and control problems was noted in the very first papers on the Kalman filter [Kal60, Kal61]. Since then, it has been used many times to extrapolate results known from one problem to obtain new results for the dual problem.

8.6 SUMMARY

In this chapter, we derived the continuous-time Kalman filter by applying a limiting argument to the discrete-time Kalman filter. However, just as there are several ways to derive the discrete-time Kalman filter, there are also several ways to derive the continuous-time Kalman filter. Kalman and Bucy's original derivation [Kal61] involved the solution of the Wiener–Hopf integral equation. Another derivation is provided in [Joh69].

We have seen that the differential and algebraic Riccati equations are key to the solution of the continuous-time Kalman filter. The scalar version of what is now known as the Riccati equation was initially studied by such mathematical luminaries as James Bernoulli and John Bernoulli in the 1600s, and Jacopo Riccati, Daniel Bernoulli, Leonard Euler, Jean-le-Rond d'Alembert, and Adrien Legendre in the 1700s. The equation was first called "Riccati's equation" by d'Alembert in 1763 [Wat22]. Jacopo Riccati originally entered the University of Padua in 1693 to study law, but he found his true calling when his astronomy professor, Stefano Angeli, inspired him to study math. Additional technical discussion of Riccati equations can be found in many places, including [Rei72, Lan95, Abo03]. An account of Riccati equations with indefinite quadratic terms is given in [Ion99]. Interesting historical background to the Riccati equation can be found in [Wat22, Bit91].

The continuous-time Kalman filter applies to systems with continuous-time white noise in the both the process and measurement equations. Continuous-time white noise is nonintuitive because it has an infinite correlation with itself at the present time, but zero correlation with itself when separated by arbitrarily small nonzero times. However, continuous-time white noise is a limiting case of discrete-time white noise, which is intuitively acceptable. Therefore, continuous-time white noise can be accepted as an approximation to reality. This corresponds to many other approximations to reality that we accept at face value (e.g., our mathematical system model is an approximation to reality, and our infinite-precision arithmetic is an approximation to reality).

The continuous-time Kalman filter applies regardless of the statistical nature of the noise, as long it is zero-mean. That is, the Kalman filter is optimal even when the noise is not Gaussian. The Kalman filter was extended in this chapter to systems with correlated process and measurement noise, and with colored measurement noise. The steady-state Kalman filter provides near-optimal estimation performance at a small fraction of the computational effort of the time-varying Kalman filter. The steady-state Kalman filter is identical to the Wiener filter of Section 3.4, and has an interesting dual relationship to linear quadratic optimal control.

PROBLEMS

Written exercises

8.1 Suppose you have two discrete-time systems with identity transition matrices driven with stationary zero-mean white noise. The first system has a sample period of T, and the second system has a sample period of T/n for some integer $n > 1$. The noise in the first system has a covariance of Q. What should the covariance of the noise in the second system be in order for both states to have the same covariance at times kT $(k = 0, 1, 2, \cdots)$?

8.2 Show that for a general time-varying matrix $Y(t)$, if $\dot{Y} = AY + YA^T$, where A is a constant matrix, then $Y(t) = \exp(At)Y(0)\exp(A^T t)$.

8.3 Suppose you have a third-order Newtonian system with

$$
\begin{aligned}
A &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \\
C &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\
Q &= \begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \\
R &= 1
\end{aligned}
$$

with $P(0) = I$.

a) What is the rank of $\dot{P}(0)$? How much computational savings in integration effort can be obtained by using the Chandrasekhar algorithm to find the Kalman gain for this system?

b) Find M_1 and M_2 such that $\dot{P}(0) = M_1 M_1^T - M_2 M_2^T$.

8.4 Show that if S is upper triangular, then $\dot{S}$ and S^{-1} are also upper triangular.

8.5 Show that the product of upper triangular matrices is another upper triangular matrix.

8.6 Find the steady-state solution of the differential Riccati equation for a scalar system. Show from your solution how the steady-state solution changes with A, C, Q, and R, and give intuitive explanations.

8.7 Consider the system of Example 8.3 except with process noise that has a covariance of $\mathrm{diag}(0, q)$. Find an analytical expression for the steady-state estimation-error covariance.

8.8 Show that if $a_1 \neq a_2$ and $q_{12} \neq 0$ in the system of Example 8.5, then (A, C) is detectable and (A, G) is stabilizable for all matrices G such that $GG^T = Q$.

8.9 Show that if $a_1 = a_2 < 0$, $q_{12} \neq 0$, and $|Q| = 0$ in the system of Example 8.5, then (A, C) is detectable and (A, G) is stabilizable for all matrices G such that $GG^T = Q$.

8.10 Show that if $a_1 = a_2 > 0$, $q_{12} \neq 0$, and $|Q| = 0$ in the system of Example 8.5, then (A, C) is detectable and (A, G) is controllable on the imaginary axis, but (A, G) is not stabilizable for all matrices G such that $GG^T = Q$.

Computer exercises

8.11 Consider the discrete-time system $x_{k+1} = x_k + w_k$ with the initial condition $x_0 = 0$. The sample time is T and the variance of the zero-mean process noise w_k is equal to $2T$. Simulate the system a few thousand times for 10 s with: (a) $T = 0.5$ s; (b) $T = 0.4$ s; (c) $T = 0.2$ s. Use the value of x_k at $t = 10$ s to obtain a statistical estimate of $P(10) = E[x^2(10)]$.

a) What is your estimate of $P(10)$ for the three sample times given?

b) What is the analytically derived value for $P(10)$?

8.12 Consider the continuous-time scalar system

$$\begin{aligned} \dot{x} &= -x + w \\ y &= x + v \end{aligned}$$

where $w(t)$ and $v(t)$ are continuous-time white noise with variances $Q_c = 2$ and $R_c = 1$ respectively. Design a continuous-time Kalman filter to estimate x.

a) What is the theoretical steady-state variance of the estimation error?

b) Simulate the system for 1000 s with discretization step sizes of 0.4, 0.2, and 0.1 s. What are the resulting experimental estimation-error variances?

8.13 Simulate the system of Problem 8.7 for 10 seconds with $q = 2$ and $R = 3$. Plot the elements of the estimation-error covariance matrix as a function of time. Compare the experimental RMS estimation errors when using a time-varying Kalman gain and a constant Kalman gain.

8.14 Repeat Problem 8.13 using the correlated noise filter when the process noise that affects the second state is equal to the measurement noise. How much do the estimation-error variances decrease due to the correlation between the two noise terms?

8.15 Consider the system of Example 8.5 with $R = I$.

a) Integrate the Riccati equation with $a_1 = 1$, $a_2 = 2$, $q_{11} = q_{12} = q_{22} = 1$, and $P(0) = I$. Plot the Riccati equation solution as a function of time and verify that its steady-state value matches the results of Equation (8.139) and MATLAB's CARE function.

b) Integrate the Riccati equation with $a_1 = a_2 = -1$, $q_{11} = 1$, $q_{12} = 2$, $q_{22} = 4$, and $P(0) = I$. Plot the Riccati equation solution as a function of time and verify that its steady-state value matches the results of Equation (8.140) and MATLAB's CARE function.

c) Integrate the Riccati equation with $a_1 = a_2 = 1$, $q_{11} = 1$, $q_{12} = 2$, $q_{22} = 4$, and $P(0) = I$. Plot the Riccati equation solution as a function of time and verify that its steady-state value matches the results of Equation (8.139) and MATLAB's CARE function.

d) Integrate the Riccati equation with $a_1 = a_2 = 1$, $q_{11} = 1$, $q_{12} = 2$, $q_{22} = 4$, and $P(0) = 0$. [Note that this is the same as part (c) except for $P(0)$.] Plot the Riccati equation solution as a function of time and verify that its steady-state value matches the results of Equation (8.140). Does it match the results of MATLAB's CARE function? Does it result in a stable steady-state Kalman filter?

CHAPTER 9

Optimal smoothing

In a *post mortem* (after the fact) analysis, it is possible to wait for more observations to accumulate. In that case, the estimate can be improved by smoothing.

—Andrew Jazwinski [Jaz70, p. 143]

In previous chapters, we discussed how to obtain the optimal *a priori* and *a posteriori* state estimates. The *a priori* state estimate at time k, $\hat{x}_k^-$, is the state estimate at time k based on all the measurements up to (but not including) time k. The *a posteriori* state estimate at time k, $\hat{x}_k^+$, is the state estimate at time k based on all the measurements up to and including time k:

$$\begin{aligned} \hat{x}_k^- &= E(x_k|y_1, \cdots, y_{k-1}) \\ \hat{x}_k^+ &= E(x_k|y_1, \cdots, y_k) \end{aligned} \tag{9.1}$$

There are often situations in which we want to obtain other types of state estimates. We will define $\hat{x}_{k,j}$ as the estimate of x_k given all measurements up to and including time j. With this notation, we see that

$$\begin{aligned} \hat{x}_{k,k-1} &= \hat{x}_k^- \\ \hat{x}_{k,k} &= \hat{x}_k^+ \end{aligned} \tag{9.2}$$

Now suppose, for example, that we have recorded measurements up to time index 54 and we want to obtain an estimate of the state at time index 33. Our theory in

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

the previous chapters tells us how to obtain $\hat{x}_{33}^-$ or $\hat{x}_{33}^+$, but those estimates only use the measurements up to and including times 32 and 33, respectively. If we have more measurements (e.g., measurements up to time 54) it stands to reason that we should be able to get an even better estimate of x_{33}. This chapter discusses some ways of obtaining better estimates.

In another scenario, it may be that we are interested in obtaining an estimate of the state at a fixed time j. As measurements keep rolling in, we want to keep updating our estimate $\hat{x}_j$. In other words, we want to obtain $\hat{x}_{j,j+1}$, $\hat{x}_{j,j+2}$, $\cdots$. This could be the case, for example, if a satellite takes a picture at time j. In order to more accurately process the photograph at time j we need an estimate of the satellite state (position and velocity) at time j. As the satellite continues to orbit, we may obtain additional range measurements of the satellite, so we can continue to update the estimate of x_j and thus improve the quality of the processed photograph. This situation is called fixed-point smoothing because the time point for which we want to obtain a state estimate (time j in this example) is fixed, but the number of measurements that are available to improve that estimate continually changes. Fixed-point smoothing is depicted in Figure 9.1 and is discussed in Section 9.2.

Figure 9.1 Fixed-point smoothing. We desire an estimate of x_4. Up until $k = 4$, the standard Kalman filter operates. At $k = 4$, we have $\hat{x}_4^- = \hat{x}_{4,4}$, which is the estimate of x_4 based on measurements up to and including y_3. As time progresses, we continue to refine our estimate of x_4 based on an increasing number of measurements. At time $k = N$, we have $\hat{x}_{4,N}$, which is the estimate of x_4 based on measurements up to and including time $N - 1$.

Another type of smoothing is fixed-lag smoothing. In this situation, we want to obtain an estimate of the state at time $(k - N)$ given measurements up to and including time k, where the time index k continually changes as we obtain new measurements, but the lag N is a constant. In other words, at each time point we have N future measurements available for our state estimate. We therefore want to obtain $\hat{x}_{k-N,k}$ for $k = N, N+1, \cdots$, where N is a fixed positive integer. This could be the case, for example, if a satellite is continually taking photographs that are to be displayed or transmitted N time steps after the photograph is taken. In this case, since the photograph is processed N time steps after it is taken, we have N additional measurements after each photograph that are available to update the estimate of the satellite state and hence improve the quality of the photograph. Fixed-lag smoothing is depicted in Figure 9.2 and is discussed in Section 9.3.

The final type of smoothing is fixed-interval smoothing. In this situation, we have a fixed interval of measurements $(y_1, y_2, \cdots, y_M)$ that are available, and we want to obtain the optimal state estimates at all the times in that interval. For each state estimate we want to use all of the measurements in the time interval. That is, we want to obtain $\hat{x}_{0,M}$, $\hat{x}_{1,M}$, $\cdots$, $\hat{x}_{M,M}$. This is the case when we have recorded

Figure 9.2 Fixed-lag smoothing. We desire an estimate of the state at each time step based on measurements two time steps ahead. After processing y_2, we form the estimate $\hat{x}_{0,2}$, which is the estimate of x_0 based on measurements up to and including y_2. Similarly, $\hat{x}_{1,3}$ is the estimate of x_1 based on measurements up to and including y_3.

some data that are available for post-processing. For example, if a manufacturing process has run over the weekend and we have recorded all of the data, and now we want to plot a time history of the best estimate of the process state, we can use all of the recorded data to estimate the states at each of the time points. Fixed-interval smoothing is depicted in Figure 9.3 and is discussed in Section 9.4.

Figure 9.3 Fixed-interval smoothing. We desire an estimate of the state at each time step based on all of the measurements in some interval. After processing all of the measurements from y_1 to y_M, we form the estimate $\hat{x}_{0,M}$, which is the estimate of x_0 based on all the measurements. Similarly, $\hat{x}_{1,M}$ is the estimate of x_1 based on all the measurements.

Our derivation of these optimal smoothers will be based on a form for the Kalman filter different than we have seen in previous chapters. Therefore, before we can discuss the optimal smoothers, we will first present an alternate Kalman filter form in Section 9.1.

9.1 AN ALTERNATE FORM FOR THE KALMAN FILTER

In order to put ourselves in position to derive optimal smoothers, we first need to derive yet another form for the Kalman filter. This is the form presented in [And79]. The equations describing the system and the Kalman filter were derived in Section 5.1 as follows:

$$
\begin{aligned}
x_k &= F_{k-1}x_{k-1} + w_{k-1} \\
y_k &= H_k x_k + v_k \\
P_{k+1}^- &= F_k P_k^+ F_k^T + Q_k \\
K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\
P_k^+ &= (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \\
\hat{x}_k^- &= F_{k-1}\hat{x}_{k-1}^+ \\
\hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-)
\end{aligned} \tag{9.3}
$$

Now if we define L_k as

$$
\begin{aligned}
L_k &= F_k K_k \\
&= F_k P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}
\end{aligned} \tag{9.4}
$$

and substitute the expression for $\hat{x}_k^+$ into the expression for $\hat{x}_{k+1}^-$, then we obtain

$$
\begin{aligned}
\hat{x}_{k+1}^- &= F_k\hat{x}_k^- + F_k K_k(y_k - H_k\hat{x}_k^-) \\
&= F_k\hat{x}_k^- + L_k(y_k - H_k\hat{x}_k^-)
\end{aligned} \tag{9.5}
$$

Expanding the expression for P_k^+ gives

$$
P_k^+ = P_k^- - K_k H_k P_k^- - P_k^- H_k^T K_k^T + K_k H_k P_k^- H_k^T K_k^T + K_k R_k K_k^T \tag{9.6}
$$

Substituting for K_k gives

$$
\begin{aligned}
P_k^+ = {} & P_k^- - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- - \\
& P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- + \\
& P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- + \\
& P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} R_k (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^-
\end{aligned} \tag{9.7}
$$

Performing some factoring and collection of like terms on this equation gives

$$
\begin{aligned}
P_k^+ &= P_k^- + P_k^- H_k^T \left[-(H_k P_k^- H_k^T + R_k)^{-1} - (H_k P_k^- H_k^T + R_k)^{-1} + \right. \\
&\quad (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} + \\
&\quad \left. (H_k P_k^- H_k^T + R_k)^{-1} R_k (H_k P_k^- H_k^T + R_k)^{-1} \right] H_k P_k^- \\
&= P_k^- + P_k^- H_k^T \left[-(H_k P_k^- H_k^T + R_k)^{-1} - (H_k P_k^- H_k^T + R_k)^{-1} + \right. \\
&\quad \left. (H_k P_k^- H_k^T + R_k)^{-1} (H_k P_k^- H_k^T + R_k)(H_k P_k^- H_k^T + R_k)^{-1} \right] H_k P_k^- \\
&= P_k^- - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^-
\end{aligned} \tag{9.8}
$$

Substituting this expression for P_k^+ into the expression for P_{k+1}^- gives

$$
\begin{aligned}
P_{k+1}^- &= F_k P_k^+ F_k^T + Q_k \\
&= F_k [P_k^- - P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^-] F_k^T + Q_k \\
&= F_k P_k^- (F_k - L_k H_k)^T + Q_k
\end{aligned} \tag{9.9}
$$

Combining Equations (9.4), (9.5), and (9.9) gives the alternate form for the one-step *a priori* Kalman filter, which can be summarized as follows:

$$\begin{aligned} L_k &= F_k P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ P_{k+1}^- &= F_k P_k^- (F_k - L_k H_k)^T + Q_k \\ \hat{x}_{k+1}^- &= F_k \hat{x}_k^- + L_k (y_k - H_k \hat{x}_k^-) \end{aligned} \tag{9.10}$$

where L_k is the redefined Kalman gain. This form of the filter obtains only *a priori* state estimates and covariances. Note that the Kalman gain, L_k, for this form of the filter is not the same as the Kalman gain, K_k, for the form of the filter that we derived in Section 5.1. However, the two forms result in identical state estimates and estimation-error covariances.

9.2 FIXED-POINT SMOOTHING

The objective in fixed-point smoothing is to obtain *a priori* state estimates of x_j at times $j+1$, $j+2$, $\cdots$, k, $k+1$, $\cdots$. We will use the notation $\hat{x}_{j,k}$ to refer to the estimate of x_j that is obtained by using all of the measurements up to and including time $(k-1)$. That is, $\hat{x}_{j,k}$ can be thought of as the *a priori* estimate of x_j at time k:

$$\hat{x}_{j,k} = E(x_j | y_1, \cdots, y_{k-1}) \qquad k \geq j \tag{9.11}$$

With this definition we see that

$$\begin{aligned} \hat{x}_{j,j} &= E(x_j | y_1, \cdots, y_{j-1}) \\ &= \hat{x}_j^- \end{aligned} \tag{9.12}$$

In other words, $\hat{x}_{j,j}$ is just the normal *a priori* state estimate at time j that we derived in Section 5.1. We also see that

$$\begin{aligned} \hat{x}_{j,j+1} &= E(x_j | y_1, \cdots, y_j) \\ &= \hat{x}_j^+ \end{aligned} \tag{9.13}$$

In other words, $\hat{x}_{j,j+1}$ is just the normal *a posteriori* state estimate at time j that we derived in Section 5.1. The question addressed by fixed-point smoothing is as follows: When we get the next measurement at time $(j+1)$, how can we incorporate that information to obtain an improved estimate (along with its covariance) for the state at time j? Furthermore, when we get additional measurements at times $(j+2)$, $(j+3)$, etc., how can we incorporate that information to obtain an improved estimate (along with its covariance) for the state at time j?

In order to derive the fixed-point smoother, we will define a new state variable x'. This new state variable will be initialized as $x'_j = x_j$, and will have the dynamics $x'_{k+1} = x'_k$ $(k = j, j+1, \cdots)$. With this definition, we see that $x'_k = x_j$ for all $k \geq j$. So if we can use the standard Kalman filter to find the *a priori* estimate of x'_k then we will, by definition, have a smoothed estimate of x_j given measurements up to and including time $(k-1)$. In other words, the *a priori* estimate of x'_k will be equal to $\hat{x}_{j,k}$. This idea is depicted in Figure 9.4.

Our original system is given as

$$\begin{aligned} x_k &= F_{k-1} x_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \end{aligned} \tag{9.14}$$

Figure 9.4 This illustrates the idea that is used to obtain the fixed-point smoother. A fictitious state variable x' is initialized as $x'_j = x_j$ and from that point on has an identity state transition matrix. The *a priori* estimate of x'_k is then equal to $\hat{x}_{j,k}$.

Augmenting the dynamics of our newly defined state x' to the original system results in the following:

$$\begin{aligned} \begin{bmatrix} x_k \\ x'_k \end{bmatrix} &= \begin{bmatrix} F_{k-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x_{k-1} \\ x'_{k-1} \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} w_{k-1} \\ y_k &= \begin{bmatrix} H_k & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x'_k \end{bmatrix} + v_k \end{aligned} \tag{9.15}$$

If we use a standard Kalman filter to obtain an *a priori* estimate of the augmented state, the covariance of the estimation error can be written as

$$E\left[\begin{pmatrix} x_k - \hat{x}_k^- \\ x_j - \hat{x}_{j,k} \end{pmatrix} \begin{pmatrix} (x_k - \hat{x}_k^-)^T & (x_j - \hat{x}_{j,k})^T \end{pmatrix}\right] = \begin{bmatrix} P_k & \Sigma_k^T \\ \Sigma_k & \Pi_k \end{bmatrix} \tag{9.16}$$

The covariance P_k above is the normal *a priori* covariance of the estimate of x_k. We have dropped the minus superscript for ease of notation, and we will also feel free to drop the minus superscript on all other quantities in this section with the understanding that all estimates and covariances are *a priori*. The Σ_k and Π_k matrices are defined by the above equation. Note that at time $k = j$, Σ_k and Π_k are given as

$$\begin{aligned} \Sigma_j &= E\left[(x_j - \hat{x}_{j,j})(x_j - \hat{x}_j^-)^T\right] \\ &= E\left[(x_j - \hat{x}_j^-)(x_j - \hat{x}_j^-)^T\right] \\ &= P_j \\ \Pi_j &= E\left[(x_j - \hat{x}_{j,j})(x_j - \hat{x}_{j,j})^T\right] \\ &= E\left[(x_j - \hat{x}_j^-)(x_j - \hat{x}_j^-)^T\right] \\ &= P_j \end{aligned} \tag{9.17}$$

The Kalman filter summarized in Equation (9.10) can be written for the augmented system as follows:

$$\begin{bmatrix} \hat{x}_{k+1}^- \\ \hat{x}_{j,k+1}^- \end{bmatrix} = \begin{bmatrix} F_{k-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}_k^- \\ \hat{x}_{j,k}^- \end{bmatrix} + \begin{bmatrix} L_k \\ \lambda_k \end{bmatrix} \left(y_k - \begin{bmatrix} H_k & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_k^- \\ \hat{x}_{j,k}^- \end{bmatrix} \right) \tag{9.18}$$

where L_k is the normal Kalman filter gain given in Equation (9.10), and λ_k is the additional part of the Kalman gain, which will be determined later in this section. Writing Equation (9.18) as two separate equations gives

$$\begin{aligned} \hat{x}_{k+1}^- &= F_{k-1}\hat{x}_k^- + L_k\left(y_k - H_k\hat{x}_k^-\right) \\ \hat{x}_{j,k+1}^- &= \hat{x}_{j,k}^- + \lambda_k\left(y_k - H_k\hat{x}_k^-\right) \end{aligned} \tag{9.19}$$

The Kalman gain can be written from Equation (9.10) as follows:

$$\begin{aligned}\begin{bmatrix} L_k \\ \lambda_k \end{bmatrix} &= \begin{bmatrix} F_{k-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} P_k & \Sigma_k^T \\ \Sigma_k & \Pi_k \end{bmatrix} \begin{bmatrix} H_k^T \\ 0 \end{bmatrix} \times \\ &\quad \left(\begin{bmatrix} H_k & 0 \end{bmatrix} \begin{bmatrix} P_k & \Sigma_k^T \\ \Sigma_k & \Pi_k \end{bmatrix} \begin{bmatrix} H_k^T \\ 0 \end{bmatrix} + R_k \right)^{-1} \\ &= \begin{bmatrix} F_{k-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} P_k & \Sigma_k^T \\ \Sigma_k & \Pi_k \end{bmatrix} \begin{bmatrix} H_k^T \\ 0 \end{bmatrix} \left(H_k P_k H_k^T + R_k\right)^{-1}\end{aligned} \tag{9.20}$$

Writing this equation as two separate equations gives

$$\begin{aligned} L_k &= F_k P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \\ \lambda_k &= \Sigma_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \end{aligned} \tag{9.21}$$

The Kalman filter estimation-error covariance-update equation can be written from Equation (9.10) as follows:

$$\begin{aligned}\begin{bmatrix} P_{k+1} & \Sigma_{k+1}^T \\ \Sigma_{k+1} & \Pi_{k+1} \end{bmatrix} &= \begin{bmatrix} F_k & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} P_k & \Sigma_k^T \\ \Sigma_k & \Pi_k \end{bmatrix} \times \\ &\quad \left(\begin{bmatrix} F_k^T & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} H_k^T \\ 0 \end{bmatrix} \begin{bmatrix} L_k^T & \lambda_k^T \end{bmatrix} \right) + \begin{bmatrix} Q_k & 0 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} F_k P_k F_k^T - F_k P_k H_k^T L_k^T & -F_k P_k H_k^T \lambda_k^T + F_k \Sigma_k^T \\ \Sigma_k F_k^T - \Sigma_k H_k^T L_k^T & -\Sigma_k H_k^T \lambda_k^T + \Pi_k \end{bmatrix} + \\ &\quad \begin{bmatrix} Q_k & 0 \\ 0 & 0 \end{bmatrix}\end{aligned} \tag{9.22}$$

Writing this equation as three separate equations gives

$$\begin{aligned} P_{k+1} &= F_k P_k (F_k - L_k H_k)^T + Q_k \\ \Pi_{k+1} &= \Pi_k - \Sigma_k H_k^T \lambda_k^T \\ \Sigma_{k+1}^T &= -F_k P_k H_k^T \lambda_k^T + F_k \Sigma_k^T \\ \Sigma_{k+1} &= \Sigma_k (F_k - L_k H_k)^T \end{aligned} \tag{9.23}$$

It is not immediately apparent from the above expressions that Σ_{k+1}^T is really the transpose of Σ_{k+1}, but the equality can be established by substituting for P_k and L_k.

Equations (9.19) – (9.23) completely define the fixed-point smoother. The fixed-point smoother, which is used for obtaining $\hat{x}_{j,k} = E(x_j | y_1, \cdots, y_{k-1})$ for $k \geq j$, can be summarized as follows.

The fixed-point smoother

1. Run the standard Kalman filter up until time j, at which point we have $\hat{x}_j^-$ and P_j^-. In the algorithm below, we omit the minus superscript on P_j^- for ease of notation.

2. Initialize the filter as follows:

$$\begin{aligned} \Sigma_j &= P_j \\ \Pi_j &= P_j \\ \hat{x}_{j,j} &= \hat{x}_j^- \end{aligned} \tag{9.24}$$

3. For $k = j, j+1, \cdots$, perform the following:

$$\begin{aligned}
L_k &= F_k P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \\
\lambda_k &= \Sigma_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \\
\hat{x}_{j,k+1} &= \hat{x}_{j,k} + \lambda_k (y_k - H_k \hat{x}_k^-) \\
\hat{x}_{k+1}^- &= F_k \hat{x}_k^- + L_k (y_k - H_k \hat{x}_k^-) \\
P_{k+1} &= F_k P_k (F_k - L_k H_k)^T + Q_k \\
\Pi_{k+1} &= \Pi_k - \Sigma_k H_k^T \lambda_k^T \\
\Sigma_{k+1} &= \Sigma_k (F_k - L_k H_k)^T
\end{aligned} \tag{9.25}$$

As we recall from Equation (9.16), P_k is the *a priori* covariance of the standard Kalman filter estimate, Π_k is the covariance of the smoothed estimate of x_j at time k, and Σ_k is the cross covariance between the two.

9.2.1 Estimation improvement due to smoothing

Now we will look at the improvement in the estimate of x_j due to smoothing. The estimate $\hat{x}_j^-$ is the standard *a priori* Kalman filter estimate of x_j, and the estimate $\hat{x}_{j,k+1}$ is the smoothed estimate after measurements up to and including time k have been processed. In other words, $\hat{x}_{j,k}$ uses $(k+1-j)$ more measurements to obtain the estimate of x_j than $\hat{x}_j^-$ uses. How much more accurate can we expect our estimate to be with the use of these additional $(k+1-j)$ measurements? The estimation accuracy can be measured by the covariance. The improvement in estimation accuracy due to smoothing is equal to the standard estimation covariance P_j minus the smoothed estimation covariance Π_{k+1}. We can use Equations (9.24) and (9.25) to write this improvement as

$$\begin{aligned}
P_j - \Pi_{k+1} &= \Pi_j - \left(\Pi_j - \sum_{i=j}^{k} \Sigma_i H_i^T \lambda_i^T \right) \\
&= \sum_{i=j}^{k} \Sigma_i H_i^T \lambda_i^T
\end{aligned} \tag{9.26}$$

Now assume for purposes of additional analysis that the system is time-invariant and the covariance of the standard filter has reached steady state at time j. Then we have

$$\lim_{k \to \infty} P_k^- = P \tag{9.27}$$

From Equation (9.25) we see that

$$\Sigma_{k+1} = \Sigma_k (F - LH)^T \tag{9.28}$$

where Σ is initialized as $\Sigma_j = P$. Combining this expression for Σ_{k+1} with its initial value, we see that

$$\begin{aligned}
\Sigma_{k+1} &= P \left[(F - LH)^T \right]^{k+1-j} \\
&= P \left(\tilde{F}^T \right)^{k+1-j}
\end{aligned} \tag{9.29}$$

where $\tilde{F}$ is defined by the above equation. Now substitute this expression, and the expression for λ from Equation (9.25), into Equation (9.26) to obtain

$$\begin{aligned} P_j - \Pi_{k+1} &= \sum_{i=j}^{k} \Sigma_i H^T \lambda^T \\ &= P \left[\sum_{i=j}^{k} \left(\tilde{F}^T\right)^{i-j} H^T (HPH^T + R)^{-1} H \tilde{F}^{i-j} \right] P \end{aligned} \tag{9.30}$$

The quantity on the right side of this equation is positive definite, which shows that the smoothed estimate of x_j is always better than the standard Kalman filter estimate. In other words, $(P_j - \Pi_{k+1}) > 0$, which implies that $\Pi_{k+1} < P_j$. Furthermore, the quantity on the right side is a sum of positive definite matrices, which shows that the larger the value of k (i.e., the more measurements that we use to obtain our smoothed estimate), the greater the improvement in the estimation accuracy. Also note from the above that the quantity $(HPH^T + R)$ inside the summation is inverted. This shows that as R increases, the quantity on the right side decreases. In the limit we see from Equation (9.30) that

$$\lim_{R \to \infty} (P_j - \Pi_{k+1}) = 0 \tag{9.31}$$

This illustrates the general principle that the larger the measurement noise, the smaller the improvement in estimation accuracy that we can obtain by smoothing. This is intuitive because large measurement noise means that additional measurements will not provide much improvement to our estimation accuracy.

■ EXAMPLE 9.1

In this example, we will see the improvement due to smoothing that can be obtained for a vehicle navigation problem. This is a second-order Newtonian system where $x(1)$ is position and $x(2)$ is velocity. The input is comprised of a commanded acceleration u plus acceleration noise $\tilde{u}$. The measurement y is a noisy measurement of position. After discretizing with a step size of T, the system equations can be written as

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} (u_k + \tilde{u}_k) \\ &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} u_k + w_k \\ y_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k \end{aligned} \tag{9.32}$$

Note that the process noise w_k is given as

$$w_k = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \tilde{u}_k \tag{9.33}$$

Now suppose the acceleration noise $\tilde{u}_k$ has a standard deviation of a. We obtain the process noise covariance as follows:

$$
\begin{aligned}
Q_k &= E(w_k w_k^T) \\
&= \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix} E(\tilde{u}_k^2) \\
&= a^2 \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix}
\end{aligned} \tag{9.34}
$$

The percent improvement due to smoothing can be defined as

$$
\text{Percent Improvement} = \frac{100\ \mathrm{Tr}(P_j - \Pi_{k+1})}{\mathrm{Tr}(P_j)} \tag{9.35}
$$

where j is the point which is being smoothed, and k is the number of measurements that are processed by the smoother. We can run the fixed-point smoother given by Equation (9.25) in order to smooth the position and velocity estimate at any desired time. Suppose we use the smoother equations to smooth the estimate at the second time step ($k = 1$). If we use measurements at times up to and including 10 seconds to estimate x_1, then our estimate is denoted as $\hat{x}_{1,101}$. In this case, Table 9.1 shows the percent improvement due to smoothing after 10 seconds when the time step $T = 0.1$ and the acceleration noise standard deviation $a = 0.2$. As expected from the results of the previous subsection, we see that the improvement due to smoothing is more dramatic for small measurement noise.

Table 9.1 Improvement due to smoothing the state at the first time step after 10 seconds for Example 9.1. The improvement due to smoothing is more noticeable when the measurement noise is small.

Measurement noise standard deviation	Percent Improvement
0.1	99.7
1	96.6
10	59.3
100	13.7
1000	0.2

Figure 9.5 shows the trace of Π_k, which is the covariance of the estimation error of the state at the first time step. As time progresses, our estimate of the state at the first time step improves. After 10 seconds of additional measurements, the estimate of the state at the first time step has improved by 96.6% relative to the standard Kalman filter estimate. Figure 9.6 shows the smoothed estimation error of the position and velocity of the first time step. We see that processing more measurements decreases the estimation-error covariance.

In general, the smoothed estimation errors shown in Figure 9.6 will converge to nonzero values. The estimation errors are zero-mean, but not for

Figure 9.5 This shows the trace of the estimation-error covariance of the smoothed estimate of the state at the first time step for Example 9.1. As time progresses and we process more measurements, the covariance decreases, eventually reaching steady state.

Figure 9.6 This shows typical estimation errors of the smoothed estimate of the state at the first time step for Example 9.1. As time progresses and we process more measurements, the estimation error decreases, and its standard deviation eventually reaches steady state.

any particular simulation. The estimation errors are zero-mean when averaged over many simulations. The system discussed here was simulated 1000 times and the variance of the estimation errors $(x_1 - \hat{x}_{1,101})$ were computed numerically to be equal to 0.054 and 0.012 for the two states. The diagonal elements of Π_{101} were equal to 0.057 and 0.012.

$\nabla\nabla\nabla$

9.2.2 Smoothing constant states

Now we will think about the improvement (due to smoothing) in the estimation accuracy of constant states. If the system states are constant then $F_k = I$ and $Q = 0$. Equation (9.25) shows that

$$\begin{aligned} P_{k+1} &= F_k P_k (F_k - L_k H_k)^T + Q_k \\ &= P_k (I - L_k H_k)^T \\ \Sigma_{k+1} &= \Sigma_k (F_k - L_k H_k)^T \\ &= \Sigma_k (I - L_k H_k)^T \end{aligned} \tag{9.36}$$

Comparing these expressions for P_{k+1} and Σ_{k+1}, and realizing from Equation (9.24) that the initial value of $\Sigma_j = P_j$, we see that $\Sigma_k = P_k$ for $k \geq j$. This means that the expression for L_k from Equation (9.25) can be written as

$$\begin{aligned} L_k &= F_k P_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \\ &= \Sigma_k H_k^T (H_k P_k H_k^T + R_k)^{-1} \\ &= \lambda_k \end{aligned} \tag{9.37}$$

Substituting these results into the expression for Π_{k+1} from Equation (9.25) we see that

$$\begin{aligned} \Pi_{k+1} &= \Pi_k - \Sigma_k H_k^T \lambda_k^T \\ &= \Pi_k - P_k H_k^T L_k^T \end{aligned} \tag{9.38}$$

Realizing that the initial value of $\Pi_j = P_j$, and comparing this expression for Π_{k+1} with Equation (9.36) for P_{k+1}, we see that $\Pi_k = P_k$ for $k \geq j$. Recall that P_k is the covariance of the estimate of x_k from the standard Kalman filter, and Π_k is the covariance of the estimate of x_j given measurements up to and including time $(k-1)$.

This result shows that constant states are not smoothable. Additional measurements are still helpful for refining an estimate of a constant state. However, there is no point to using smoothing for estimation of a constant state. If we want to estimate a constant state at time j using measurements up to time $k > j$, then we may as well simply run the standard Kalman filter up to time k. Implementing the smoothing equations will not gain any improvement in estimation accuracy.

9.3 FIXED-LAG SMOOTHING

In fixed-lag smoothing we want to obtain an estimate of the state at time $(k-N)$ given measurements up to and including time k, where the time index k continually changes as we obtain new measurements, but the lag N is a constant. In other words, at each time point we have N future measurements available for our state estimate. We therefore want to obtain $\hat{x}_{k-N,k}$ for $k = N, N+1, \cdots$, where N is a fixed positive integer. This could be the case, for example, if a satellite is continually taking photographs that are to be displayed or transmitted N time steps after the photograph is taken. In this case, since the photograph is processed N time steps after it is taken, we have N additional measurements after each photograph that

are available to update the estimate of the satellite state and hence improve the quality of the photograph. In this section we use the notation

$$\begin{aligned} \hat{x}_{k-N,k} &= E(x_{k-N}|y_1,\cdots,y_k) \\ \Pi_{k-N} &= E\left[(x_{k-N}-\hat{x}_{k-N,k})(x_{k-N}-\hat{x}_{k-N,k})^T\right] \end{aligned} \tag{9.39}$$

Note that the notation has changed slightly from the previous section. In the previous section we used the notation $\hat{x}_{k,m}$ to refer to the estimate of x_k given measurements up to and including time $(m-1)$. In this section (and in the remainder of this chapter) we use $\hat{x}_{k,m}$ to refer to the estimate of x_k given measurements up to and including time m.

Let us define $x_{k,m}$ as the state x_{k-m} propagated with an identity transition matrix and zero process noise to time k. With this definition we see that

$$\begin{aligned} x_{k+1,1} &= x_k \\ x_{k+1,2} &= x_{k-1} \\ &= x_{k,1} \\ x_{k+1,3} &= x_{k-2} \\ &= x_{k,2} \\ &\text{etc.} \end{aligned} \tag{9.40}$$

We can therefore define the augmented system

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ x_{k+1,1} \\ \vdots \\ x_{k+1,N+1} \end{bmatrix} &= \begin{bmatrix} F_k & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k,1} \\ \vdots \\ x_{k,N+1} \end{bmatrix} + \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} w_k \\ y_k &= \begin{bmatrix} H_k & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k,1} \\ \vdots \\ x_{k,N+1} \end{bmatrix} + v_k \end{aligned} \tag{9.41}$$

The Kalman filter estimates of the components of this augmented state vector are given as

$$\begin{aligned} E(x_{k+1}|y_1\cdots y_k) &= \hat{x}_{k+1}^- \\ &= \hat{x}_{k+1,k} \\ E(x_{k+1,1}|y_1\cdots y_k) &= E(x_k|y_1\cdots y_k) \\ &= \hat{x}_k^+ \\ &= \hat{x}_{k,k} \\ E(x_{k+1,2}|y_1\cdots y_k) &= E(x_{k-1}|y_1\cdots y_k) \\ &= \hat{x}_{k-1,k} \\ &\vdots \\ E(x_{k+1,N+1}|y_1\cdots y_k) &= \hat{x}_{k-N,k} \end{aligned} \tag{9.42}$$

We see that if we can use a Kalman filter to estimate the states of the augmented system (using measurements up to and including time k), then the estimate of the

last element of the augmented state vector, $x_{k+1,N+1}$, will be equal to the estimate of x_{k-N} given measurements up to and including time k. This is the estimate that we are looking for in fixed-lag smoothing. This idea is illustrated in Figure 9.7.

Figure 9.7 This illustrates the idea that is used to obtain the fixed-lag smoother. A fictitious state variable $x_{k,m}$ is initialized as $x_{k,m} = x_{k-m}$ and from that point on has an identity state transition matrix. The *a posteriori* estimate of $x_{k+m,m}$ is then equal to $\hat{x}_{k-m,k}$.

From Equation (9.10) we can write the Kalman filter for the augmented system of Equation (9.41) as follows:

$$\begin{bmatrix} \hat{x}_{k+1}^- \\ \hat{x}_{k,k} \\ \vdots \\ \hat{x}_{k-N,k} \end{bmatrix} = \begin{bmatrix} F_k & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_k^- \\ \hat{x}_{k-1,k-1} \\ \vdots \\ \hat{x}_{k-(N+1),k-1} \end{bmatrix} + \begin{bmatrix} L_{k,0} \\ L_{k,1} \\ \vdots \\ L_{k,N+1} \end{bmatrix} \left(y_k - \begin{bmatrix} H_k & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_k^- \\ \hat{x}_{k-1,k-1} \\ \vdots \\ x_{k-(N+1),k-1} \end{bmatrix} \right) \tag{9.43}$$

where the $L_{k,i}$ matrices are components of the smoother gain that will be determined in this section. Note that $L_{k,0}$ is the standard Kalman gain. The smoother gain L_k is defined as

$$L_k = \begin{bmatrix} L_{k,0} \\ L_{k,1} \\ \vdots \\ L_{k,N+1} \end{bmatrix} \tag{9.44}$$

From Equation (9.10) we see that the L_k gain matrix is given by

$$L_k = \begin{bmatrix} F_k & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} P_k^{0,0} & \cdots & (P_k^{0,N+1})^T \\ \vdots & \ddots & \vdots \\ P_k^{0,N+1} & \cdots & P_k^{N+1,N+1} \end{bmatrix} \begin{bmatrix} H_k^T \\ 0 \\ \vdots \\ 0 \end{bmatrix} \times \left(\begin{bmatrix} H_k & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} P_k^{0,0} & \cdots & (P_k^{0,N+1})^T \\ \vdots & \ddots & \vdots \\ P_k^{0,N+1} & \cdots & P_k^{N+1,N+1} \end{bmatrix} \begin{bmatrix} H_k^T \\ 0 \\ \vdots \\ 0 \end{bmatrix} + R_k \right)^{-1} \tag{9.45}$$

where the $P_k^{i,j}$ covariance matrices are defined as

$$P_k^{i,j} = E\left[(x_{k-j} - \hat{x}_{k-j,k-1})(x_{k-i} - \hat{x}_{k-i,k-1})^T\right] \tag{9.46}$$

The L_k expression above can be simplified to

$$L_k = \begin{bmatrix} F_k P_k^{0,0} H_k^T \\ P_k^{0,0} H_k^T \\ \vdots \\ P_k^{0,N} H_k^T \end{bmatrix} (H_k P_k^{0,0} H_k^T + R_k)^{-1} \tag{9.47}$$

From Equation (9.10) we see that the covariance-update equation for the Kalman filter for our augmented system can be written as

$$\begin{bmatrix} P_{k+1}^{0,0} & \cdots & (P_{k+1}^{0,N+1})^T \\ \vdots & \ddots & \vdots \\ P_{k+1}^{0,N+1} & \cdots & P_{k+1}^{N+1,N+1} \end{bmatrix} = \begin{bmatrix} F_k & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} P_k^{0,0} & \cdots & (P_k^{0,N+1})^T \\ \vdots & \ddots & \vdots \\ P_k^{0,N+1} & \cdots & P_k^{N+1,N+1} \end{bmatrix} \times$$
$$\left(\begin{bmatrix} F_k^T & I & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & I \\ 0 & \cdots & \cdots & 0 \end{bmatrix} - \begin{bmatrix} H_k^T \\ 0 \\ \vdots \\ 0 \end{bmatrix} L_k^T \right) + \begin{bmatrix} Q_k & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix} \tag{9.48}$$

Substituting for L_k from Equation (9.47) and multiplying out gives

$$\begin{bmatrix} P_{k+1}^{0,0} & \cdots & (P_{k+1}^{0,N+1})^T \\ \vdots & \ddots & \vdots \\ P_{k+1}^{0,N+1} & \cdots & P_{k+1}^{N+1,N+1} \end{bmatrix} = \begin{bmatrix} F_k P_k^{0,0} & F_k (P_k^{0,1})^T & \cdots & F_k (P_k^{0,N+1})^T \\ P_k^{0,0} & (P_k^{0,1})^T & \cdots & (P_k^{0,N+1})^T \\ \vdots & \ddots & \ddots & \vdots \\ P_k^{0,N} & P_k^{1,N} & \cdots & (P_k^{N,N+1})^T \end{bmatrix} \times$$
$$\left(\begin{bmatrix} F_k^T & I & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & I \\ 0 & \cdots & \cdots & 0 \end{bmatrix} - H_k^T (H_k P_k^{0,0} H_k^T + R_k)^{-1} H_k \times \right.$$
$$\left. \begin{bmatrix} P_k^{0,0} F_k^T & P_k^{0,0} & \cdots & P_k^{0,N} \\ 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \right) + \begin{bmatrix} Q_k & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix} \tag{9.49}$$

This gives us the update equations for the P matrices. The equations for the first column of the P matrix are as follows:

$$\begin{aligned} P_{k+1}^{0,0} &= F_k P_k^{0,0} \left[F_k^T - H_k^T (H_k P_k^{0,0} H_k^T + R_k)^{-1} H_k P_k^{0,0} F_k^T \right] + Q_k \\ &= F_k P_k^{0,0} (F_k - L_{k,0} H_k)^T + Q_k \\ P_{k+1}^{0,1} &= P_k^{0,0} (F_k - L_{k,0} H_k)^T \\ &\vdots \\ P_{k+1}^{0,N+1} &= P_k^{0,N} (F_k - L_{k,0} H_k)^T \end{aligned} \tag{9.50}$$

The equations for the diagonal elements of the P matrix are as follows:

$$\begin{aligned}
P_{k+1}^{1,1} &= P_k^{0,0}\left[I - H_k^T(H_k P_k^{0,0} H_k^T + R_k)^{-1} H_k P_k^{0,0} F_k^T\right] \\
&= P_k^{0,0} - P_k^{0,0} H_k^T L_{k,1}^T F_k^T \\
P_{k+1}^{2,2} &= P_k^{0,1}[-H_k^T L_{k,2}^T F_k^T] + P_k^{1,1} \\
&= P_k^{1,1} - P_k^{0,1} H_k^T L_{k,2}^T F_k^T \\
&\vdots \\
P_{k+1}^{i,i} &= P_k^{i-1,i-1} - P_k^{0,i-1} H_k^T L_{k,i}^T F_k^T
\end{aligned} \tag{9.51}$$

These equations give us the formulas that we can use for fixed-lag smoothing. This gives us the estimate $E(x_{k-N}|y_1, \cdots, y_k)$ for a fixed N as k continually increments. The fixed-lag smoother is summarized as follows.

The fixed-lag smoother

1. Run the standard Kalman filter of Equation (9.10) to obtain $\hat{x}_{k+1}^-$, L_k, and P_k^-.

2. Initialize the fixed-lag smoother as follows:

$$\begin{aligned}
\hat{x}_{k+1,k} &= \hat{x}_{k+1}^- \\
L_{k,0} &= L_k \\
P_k^{0,0} &= P_k^-
\end{aligned} \tag{9.52}$$

3. For $i = 1, \cdots, N+1$, perform the following:

$$\begin{aligned}
L_{k,i} &= P_k^{0,i-1} H_k^T (H_k P_k^{0,0} H_k^T + R_k)^{-1} \\
P_{k+1}^{i,i} &= P_k^{i-1,i-1} - P_k^{0,i-1} H_k^T L_{k,i}^T F_k^T \\
P_{k+1}^{0,i} &= P_k^{0,i-1}(F_k - L_{k,0} H_k)^T \\
\hat{x}_{k+1-i,k} &= \hat{x}_{k+2-i,k} + L_{k,i}(y_k - H_k \hat{x}_k^-)
\end{aligned} \tag{9.53}$$

   Note that the first time through this loop is the measurement update of the standard Kalman filter. At the end of this loop we have the smoothed estimates of *each* state with delays between 0 and N, given measurements up to and including time k. These estimates are denoted $\hat{x}_{k,k}, \cdots, \hat{x}_{k-N,k}$. We also have the estimation-error covariances, denoted $P_{k+1}^{1,1}, \cdots, P_{k+1}^{N+1,N+1}$

The percent improvement due to smoothing can be computed as

$$\text{Percent Improvement} = \frac{100\ \mathrm{Tr}(P_k^{0,0} - P_k^{N+1,N+1})}{\mathrm{Tr}(P_k^{0,0})} \tag{9.54}$$

■ **EXAMPLE 9.2**

Consider the same two state system as described in Example 9.1. Suppose we are trying to estimate the state of the system with a fixed time lag. The discretization time step $T = 0.1$ and the standard deviation of the acceleration noise is 10. Figure 9.8 shows the percent improvement in state estimation that is available with fixed-lag smoothing. The figure shows percent improvement as a function of lag size, and for two different values of measurement noise. The values on the plot are based on the theoretical estimation-error covariance. As expected, the improvement in estimation accuracy is more dramatic as the measurement noise decreases. This was discussed at the end of Section 9.2.

Figure 9.8 This shows the percent improvement of the trace of the estimation-error covariance of the smoothed estimate of the state (relative to the standard Kalman filter) for Example 9.2. As the number of lag intervals increases, the estimation error of the smoother decreases and the percent improvement increases. Also, as the measurement noise decreases, the improvement due to smoothing is more dramatic.

▽▽▽

9.4 FIXED-INTERVAL SMOOTHING

Suppose we have measurements for a fixed time interval. In fixed-interval smoothing we seek an estimate of the state at some of the interior points of the time interval. During the smoothing process we do not obtain any new measurements. Section 9.4.1 discusses the forward–backward approach to smoothing, which is perhaps the most straightforward smoothing algorithm. Section 9.4.2 discusses the RTS smoother, which is conceptually more difficult but is computationally cheaper than forward–backward smoothing.

9.4.1 Forward–backward smoothing

Suppose we want to estimate the state x_m based on measurements from $k = 1$ to $k = N$, where $N > m$. The forward–backward approach to smoothing obtains two estimates of x_m. The first estimate, $\hat{x}_f$, is based on the standard Kalman filter that operates from $k = 1$ to $k = m$. The second estimate, $\hat{x}_b$, is based on a Kalman filter that runs backward in time from $k = N$ back to $k = m$. The forward–backward approach to smoothing combines the two estimates to form an optimal smoothed estimate. This approach was first suggested in [Fra69].

Suppose that we combine a forward estimate $\hat{x}_f$ of the state and a backward estimate $\hat{x}_b$ of the state to get a smoothed estimate of x as follows:

$$\hat{x} = K_f \hat{x}_f + K_b \hat{x}_b \tag{9.55}$$

where K_f and K_b are constant matrix coefficients to be determined. Note that $\hat{x}_f$ and $\hat{x}_b$ are both unbiased since they are both outputs from Kalman filters. Therefore, if $\hat{x}$ is to be unbiased, we require $K_f + K_b = I$ (see Problem 9.9). This gives

$$\hat{x} = K_f \hat{x}_f + (I - K_f)\hat{x}_b \tag{9.56}$$

The covariance of the estimate can then be found as

$$\begin{aligned} P &= E[(x - \hat{x})(x - \hat{x})^T] \\ &= E\left\{[x - K_f\hat{x}_f - (I - K_f)\hat{x}_b][\cdots]^T\right\} \\ &= E\left\{[K_f(e_f - e_b) + e_b][\cdots]^T\right\} \\ &= E\left\{K_f(e_f e_f^T + e_b e_b^T)K_f^T + e_b e_b^T - K_f e_b e_b^T - e_b e_b^T K_f^T\right\} \end{aligned} \tag{9.57}$$

where $e_f = x - x_f$, $e_b = x - x_b$, and we have used the fact that $E(e_f e_b^T) = 0$. The estimates $\hat{x}_f$ and $\hat{x}_b$ are both unbiased, and e_f and e_b are independent (since they depend on separate sets of measurements). We can minimize the trace of P with respect to K_f using results from Equation (1.66) and Problem 1.4:

$$\begin{aligned} \frac{\partial \mathrm{Tr}(P)}{\partial K_f} &= 2E\left\{K_f(e_f e_f^T + e_b e_b^T) - e_b e_b^T\right\} \\ &= 2[K_f(P_f + P_b) - P_b] \end{aligned} \tag{9.58}$$

where $P_f = E(e_f e_f^T)$ is the covariance of the forward estimate, and $P_b = E(e_b e_b^T)$ is the covariance of the backward estimate. Setting this equal to zero to find the optimal value of K_f gives

$$\begin{aligned} K_f &= P_b(P_f + P_b)^{-1} \\ K_b &= P_f(P_f + P_b)^{-1} \end{aligned} \tag{9.59}$$

The inverse of $(P_f + P_b)$ always exists since both covariance matrices are positive definite. We can substitute this result into Equation (9.57) to find the covariance of the fixed-interval smoother as follows:

$$\begin{aligned} P = {} & P_b(P_f + P_b)^{-1}(P_f + P_b)(P_f + P_b)^{-1}P_b + \\ & P_b - P_b(P_f + P_b)^{-1}P_b - P_b(P_f + P_b)^{-1}P_b \end{aligned} \tag{9.60}$$

Using the identity $(A+B)^{-1} = B^{-1}(AB^{-1}+I)^{-1}$ (see Problem 9.2), we can write the above equation as

$$\begin{aligned} P &= (P_f P_b^{-1} + I)^{-1}(P_f + P_b)(P_b^{-1}P_f + I)^{-1} + \\ &\quad P_b - (P_f P_b^{-1} + I)^{-1}P_b - (P_f P_b^{-1} + I)^{-1}P_b \end{aligned} \tag{9.61}$$

Multiplying out the first term, and again using the identity $(A+B)^{-1} = B^{-1}(AB^{-1}+I)^{-1}$ on the last two terms, results in

$$\begin{aligned} P &= \left[(P_b^{-1} + P_f^{-1})^{-1} + (P_b^{-1}P_f P_b^{-1} + P_b^{-1})^{-1}\right](P_b^{-1}P_f + I)^{-1} + \\ &\quad P_b - 2(P_b^{-1}P_f P_b^{-1} + P_b^{-1})^{-1} \end{aligned} \tag{9.62}$$

From the matrix inversion lemma of Equation (1.39) we see that

$$(P_b^{-1}P_f P_b^{-1} + P_b^{-1})^{-1} = P_b - (P_f^{-1} + P_b^{-1})^{-1} \tag{9.63}$$

Substituting this into Equation (9.62) gives

$$\begin{aligned} P &= P_b(P_b^{-1}P_f + I)^{-1} + P_b - 2P_b + 2(P_f^{-1} + P_b^{-1})^{-1} \\ &= (P_b^{-1}P_f P_b^{-1} + P_b^{-1})^{-1} - P_b + 2(P_f^{-1} + P_b^{-1})^{-1} \\ &= P_b - (P_f^{-1} + P_b^{-1})^{-1} - P_b + 2(P_f^{-1} + P_b^{-1})^{-1} \\ &= (P_f^{-1} + P_b^{-1})^{-1} \end{aligned} \tag{9.64}$$

These results form the basis for the fixed-interval smoothing problem. The system model is given as

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \end{aligned} \tag{9.65}$$

Suppose we want a smoothed estimate at time index m. First we run the forward Kalman filter normally, using measurements up to and including time m.

1. Initialize the forward filter as follows:

$$\begin{aligned} \hat{x}_{f0}^{+} &= E(x_0) \\ P_{f0}^{+} &= E\left[(x_0 - \hat{x}_{f0}^{+})(x_0 - \hat{x}_{f0}^{+})^T\right] \end{aligned} \tag{9.66}$$

2. For $k = 1, \cdots, m$, perform the following:

$$\begin{aligned} P_{fk}^{-} &= F_{k-1}P_{f,k-1}^{+}F_{k-1}^T + Q_{k-1} \\ K_{fk} &= P_{fk}^{-}H_k^T(H_k P_{fk}^{-} H_k^T + R_k)^{-1} \\ &= P_{fk}^{+}H_k^T R_k^{-1} \\ \hat{x}_{fk}^{-} &= F_{k-1}\hat{x}_{f,k-1}^{+} \\ \hat{x}_{fk}^{+} &= \hat{x}_{fk}^{-} + K_{fk}(y_k - H_k\hat{x}_{fk}^{-}) \\ P_{fk}^{+} &= (I - K_{fk}H_k)P_{fk}^{-} \end{aligned} \tag{9.67}$$

At this point we have a forward estimate for x_m, along with its covariance. These quantities are obtained using measurements up to and including time m.

The backward filter needs to run backward in time, starting at the final time index N. Since the forward and backward estimates must be independent, none of the information that was used in the forward filter is allowed to be used in the backward filter. Therefore, P_{bN}^- must be infinite:

$$P_{bN}^- = \infty \tag{9.68}$$

We are using the minus superscript on P_{bN}^- to indicate the backward covariance at time N before the measurement at time N is processed. (Recall that the filtering is performed backward in time.) So P_{bN}^- will be updated to obtain P_{bN}^+ after the measurement at time N is processed. Then it will be extrapolated backward in time to obtain $P_{b,N-1}^-$, and so on.

Now the question arises how to initialize the backward state estimate $\hat{x}_{bk}^-$ at the final time $k = N$. We can solve this problem by introducing the new variable

$$s_k = P_{bk}^{-1}\hat{x}_{bk} \tag{9.69}$$

A minus or plus superscript can be added on all the quantities in the above equation to indicate values before or after the measurement at time k is taken into account. Since $P_{bN}^- = \infty$ it follows that

$$s_N^- = 0 \tag{9.70}$$

The infinite boundary condition on P_{bk}^- means that we cannot run the standard Kalman filter backward in time because we have to begin with an infinite covariance. Instead we run the information filter from Section 6.2 backward in time. This can be done by writing the system of Equation (9.65) as

$$\begin{aligned}
x_{k-1} &= F_{k-1}^{-1}x_k + F_{k-1}^{-1}w_{k-1} \\
&= F_{k-1}^{-1}x_k + w_{b,k-1} \\
y_k &= H_k x_k + v_k \\
w_{bk} &\sim (0, F_k^{-1}Q_k F_k^{-T}) \\
v_k &\sim (0, R_k)
\end{aligned} \tag{9.71}$$

Note that F_k^{-1} should always exist if it comes from a real system, because F_k comes from a matrix exponential that is always invertible (see Sections 1.2 and 1.4). The backward information filter can be written as follows.

1. Initialize the filter with $\mathcal{I}_{bN}^- = 0$.

2. For $k = N, N-1, \cdots$, perform the following:

$$\begin{aligned}
\mathcal{I}_{bk}^+ &= \mathcal{I}_{bk}^- + H_k^T R_k^{-1} H_k \\
K_{bk} &= (\mathcal{I}_{bk}^+)^{-1} H_k^T R_k^{-1} \\
\hat{x}_{bk}^+ &= \hat{x}_{bk}^- + K_{bk}(y_k - H_k \hat{x}_{bk}^-) \\
\mathcal{I}_{b,k-1}^- &= \left[F_{k-1}^{-1}(\mathcal{I}_{bk}^+)^{-1}F_{k-1}^{-T} + F_{k-1}^{-1}Q_{k-1}F_{k-1}^{-T}\right]^{-1} \\
&= F_{k-1}^T\left[(\mathcal{I}_{bk}^+)^{-1} + Q_{k-1}\right]^{-1} F_{k-1} \\
&= F_{k-1}^T\left[Q_{k-1}^{-1} - Q_{k-1}^{-1}(\mathcal{I}_{bk}^+ + Q_{k-1}^{-1})^{-1}Q_{k-1}^{-1}\right] F_{k-1} \\
\hat{x}_{b,k-1}^- &= F_{k-1}^{-1}\hat{x}_{bk}^+
\end{aligned} \tag{9.72}$$

The first form for $\mathcal{I}_{b,k-1}^-$ above requires the inversion of $\mathcal{I}_{bk}^+$. Consider the first time step for the backward filter (i.e., at $k = N$). The information matrix $\mathcal{I}_{bN}^-$ is initialized to zero, and then the first time through the above loop we set $\mathcal{I}_{bk}^+ = \mathcal{I}_{bk}^- + H_k^T R_k^{-1} H_k$. If there are fewer measurements than states, $H_k^T R_k^{-1} H_k$ will always be singular and, therefore, $\mathcal{I}_{bk}^+$ will be singular at $k = N$. Therefore, the first form given above for $\mathcal{I}_{b,k-1}^-$ will not be computable. In practice we can get around this by initializing $\mathcal{I}_{bN}^-$ to a small nonzero matrix instead of zero.

The third form for $\mathcal{I}_{b,k-1}^-$ above has its own problems. It does not require the inversion of $\mathcal{I}_{bk}^+$, but it does require the inversion of Q_{k-1}. So the third form of $\mathcal{I}_{b,k-1}^-$ is not computable unless Q_{k-1} is nonsingular. Again, in practice we can get around this by making a small modification to Q_{k-1} so that it is numerically nonsingular.

Since we need to update $s_k = \mathcal{I}_{bk}\hat{x}_{bk}$ instead of $\hat{x}_{bk}$ (because of initialization issues) as defined in Equation (9.69), we rewrite the update equations for the state estimate as follows:

$$\begin{aligned} \hat{x}_{bk}^+ &= \hat{x}_{bk}^- + K_{bk}(y_k - H_k\hat{x}_{bk}^-) \\ s_k^+ &= \mathcal{I}_{bk}^+\hat{x}_{bk}^+ \\ &= \mathcal{I}_{bk}^+\hat{x}_{bk}^- + \mathcal{I}_{bk}^+ K_{bk}(y_k - H_k\hat{x}_{bk}^-) \end{aligned} \tag{9.73}$$

Now note from Equation (6.33) that we can write $\mathcal{I}_{bk}^+ = \mathcal{I}_{bk}^- + H_k^T R_k^{-1} H_k$, and $K_{bk} = P_{bk}^+ H_k^T R_k^{-1}$. Substituting these expressions into the above equation for s_k^+ gives

$$\begin{aligned} s_k^+ &= \mathcal{I}_{bk}^-\hat{x}_{bk}^- + H_k^T R_k^{-1} H_k \hat{x}_{bk}^- + H_k^T R_k^{-1}(y_k - H_k\hat{x}_{bk}^-) \\ &= s_k^- + H_k^T R_k^{-1} y_k \end{aligned} \tag{9.74}$$

We combine this with Equation (9.72) to write the backward information filter as follows.

1. Initialize the filter as follows:

$$\begin{aligned} s_N^- &= 0 \\ \mathcal{I}_{bN}^- &= 0 \end{aligned} \tag{9.75}$$

2. For $k = N, N-1, \cdots, m+1$, perform the following:

$$\begin{aligned} \mathcal{I}_{bk}^+ &= \mathcal{I}_{bk}^- + H_k^T R_k^{-1} H_k \\ s_k^+ &= s_k^- + H_k^T R_k^{-1} y_k \\ \mathcal{I}_{b,k-1}^- &= \left[F_{k-1}^{-1}(\mathcal{I}_{bk}^+)^{-1}F_{k-1}^{-T} + F_{k-1}^{-1}Q_{k-1}F_{k-1}^{-T}\right]^{-1} \\ &= F_{k-1}^T\left[(\mathcal{I}_{bk}^+)^{-1} + Q_{k-1}\right]^{-1} F_{k-1} \\ &= F_{k-1}^T\left[Q_{k-1}^{-1} - Q_{k-1}^{-1}(\mathcal{I}_{bk}^+ + Q_{k-1}^{-1})^{-1}Q_{k-1}^{-1}\right] F_{k-1} \\ s_{k-1}^- &= \mathcal{I}_{b,k-1}^- F_{k-1}^{-1}(\mathcal{I}_{bk}^+)^{-1} s_k^+ \end{aligned} \tag{9.76}$$

3. Perform one final time update to obtain the backward estimate of x_m:

$$
\begin{aligned}
\mathcal{I}_{bm}^{-} &= Q_m^{-1} - Q_m^{-1}F_m^{-1}(\mathcal{I}_{b,m+1}^{+} + F_m^{-T}Q_m^{-1}F_m^{-1})^{-1}F_m^{-T}Q_m^{-1} \\
P_{bm}^{-} &= (\mathcal{I}_{bm}^{-})^{-1} \\
s_m^{-} &= \mathcal{I}_{bm}^{-}F_m^{-1}(\mathcal{I}_{b,m+1}^{+})^{-1}s_{m+1}^{+} \\
\hat{x}_{bm}^{-} &= (\mathcal{I}_{bm}^{-})^{-1}s_m^{-}
\end{aligned} \tag{9.77}
$$

Now we have the backward estimate $\hat{x}_{bm}^{-}$ and its covariance P_{bm}^{-}. These quantities are obtained from measurements $m+1, m+2, \cdots, N$.

After we obtain the backward quantities as outlined above, we combine them with the forward quantities from Equation (9.67) to obtain the final state estimate and covariance:

$$
\begin{aligned}
K_f &= P_{bm}^{-}(P_{fm}^{+} + P_{bm}^{-})^{-1} \\
\hat{x}_m &= K_f\hat{x}_{fm}^{+} + (I - K_f)\hat{x}_{bm}^{-} \\
P_m &= \left[(P_{fm}^{+})^{-1} + (P_{bm}^{-})^{-1}\right]^{-1}
\end{aligned} \tag{9.78}
$$

We can obtain an alternative equation for $\hat{x}_m$ by manipulating the above equations. If we substitute for K_f in the above expression for $\hat{x}_m$ then we obtain

$$
\begin{aligned}
\hat{x}_m &= P_{bm}^{-}(P_{fm}^{+} + P_{bm}^{-})^{-1}\hat{x}_{fm}^{+} + \left[I - P_{bm}^{-}(P_{fm}^{+} + P_{bm}^{-})^{-1}\right]\hat{x}_{bm}^{-} \\
&= P_{bm}^{-}(P_{fm}^{+} + P_{bm}^{-})^{-1}\hat{x}_{fm}^{+} + \left[(P_{fm}^{+} + P_{bm}^{-}) - P_{bm}^{-}\right](P_{fm}^{+} + P_{bm}^{-})^{-1}\hat{x}_{bm}^{-} \\
&= P_{bm}^{-}(P_{fm}^{+} + P_{bm}^{-})^{-1}\hat{x}_{fm}^{+} + P_{fm}^{+}(P_{fm}^{+} + P_{bm}^{-})^{-1}\hat{x}_{bm}^{-}
\end{aligned} \tag{9.79}
$$

Using the matrix inversion lemma on the rightmost inverse in the above equation and performing some other manipulations gives

$$
\begin{aligned}
\hat{x}_m &= [(P_{bm}^{-} + P_{fm}^{+}) - P_{fm}^{+}](P_{fm}^{+} + P_{bm}^{-})^{-1}\hat{x}_{fm}^{+} + \\
&\quad P_{fm}^{+}\left[\mathcal{I}_{bm}^{-} - \mathcal{I}_{bm}^{-}(\mathcal{I}_{fm}^{+} + \mathcal{I}_{bm}^{-})^{-1}\mathcal{I}_{bm}^{-}\right]\hat{x}_{bm}^{-} \\
&= \left[I - P_{fm}^{+}(P_{fm}^{+} + P_{bm}^{-})^{-1}\right]\hat{x}_{fm}^{+} + \\
&\quad P_{fm}^{+}\left[I - \mathcal{I}_{bm}^{-}(\mathcal{I}_{fm}^{+} + \mathcal{I}_{bm}^{-})^{-1}\right]\mathcal{I}_{bm}^{-}\hat{x}_{bm}^{-} \\
&= \left[I - P_{fm}^{+}\mathcal{I}_{bm}^{-}(I + P_{fm}^{+}\mathcal{I}_{bm}^{-})^{-1}\right]\hat{x}_{fm}^{+} + \\
&\quad P_{fm}^{+}\left[I - \mathcal{I}_{bm}^{-}(I + P_{fm}^{+}\mathcal{I}_{bm}^{-})^{-1}P_{fm}^{+}\right]\mathcal{I}_{bm}^{-}\hat{x}_{bm}^{-} \\
&= P_{fm}^{+}\left[I - \mathcal{I}_{bm}^{-}(I + P_{fm}^{+}\mathcal{I}_{bm}^{-})^{-1}P_{fm}^{+}\right]\mathcal{I}_{fm}^{+}\hat{x}_{fm}^{+} + \\
&\quad P_{fm}^{+}\left[I - \mathcal{I}_{bm}^{-}(I + P_{fm}^{+}\mathcal{I}_{bm}^{-})^{-1}P_{fm}^{+}\right]\mathcal{I}_{bm}^{-}\hat{x}_{bm}^{-}
\end{aligned} \tag{9.80}
$$

where we have relied on the identity $(A+B)^{-1} = B^{-1}(AB^{-1} + I)^{-1}$ (see Problem 9.2). The coefficients of $\hat{x}_{fm}^{+}$ and $\hat{x}_{bm}^{-}$ in the above equation both have a common factor which can be written as follows:

$$
\begin{aligned}
& P_{fm}^{+}\left[I-\mathcal{I}_{bm}^{-}(I+P_{fm}^{+}\mathcal{I}_{bm}^{-})^{-1}P_{fm}^{+}\right] \\
&= P_{fm}^{+}-P_{fm}^{+}\mathcal{I}_{bm}^{-}(I+P_{fm}^{+}\mathcal{I}_{bm}^{-})^{-1}P_{fm}^{+} \\
&= P_{fm}^{+}-P_{fm}^{+}(\mathcal{I}_{fm}^{+}P_{bm}^{-}+I)^{-1} \\
&= \left[P_{fm}^{+}(\mathcal{I}_{fm}^{+}P_{bm}^{-}+I)-P_{fm}^{+}\right](\mathcal{I}_{fm}^{+}P_{bm}^{-}+I)^{-1} \\
&= P_{bm}^{-}(\mathcal{I}_{fm}^{+}P_{bm}^{-}+I)^{-1} \\
&= (\mathcal{I}_{fm}^{+}+\mathcal{I}_{bm}^{-})^{-1}
\end{aligned}
\tag{9.81}
$$

Therefore, using Equation (9.78), we can write Equation (9.80) as

$$
\begin{aligned}
\hat{x}_m &= P_m\mathcal{I}_{fm}^{+}\hat{x}_{fm}^{+}+P_m\mathcal{I}_{bm}^{-}\hat{x}_{bm}^{-} \\
&= P_m\left(\mathcal{I}_{fm}^{+}\hat{x}_{fm}^{+}+\mathcal{I}_{bm}^{-}\hat{x}_{bm}^{-}\right)
\end{aligned}
\tag{9.82}
$$

Figure 9.9 illustrates how the forward–backward smoother works.

Figure 9.9 This figure illustrates the concept of the forward–backward smoother. The forward filter is run to obtain *a posteriori* estimates and covariances up to time m. Then the backward filter is run to obtain *a priori* estimates and covariances back to time m (i.e., *a priori* from a reversed time perspective). Then the forward and backward estimates and covariances at time m are combined to obtain the final estimate $\hat{x}_m$ and covariance P_m.

■ EXAMPLE 9.3

In this example we consider the same problem given in Example 9.1. Suppose that we want to estimate the position and velocity of the vehicle at $t = 5$ seconds. We have measurements every 0.1 seconds for a total of 10 seconds. The standard deviation of the measurement noise is 10, and the standard deviation of the acceleration noise is 10. Figure 9.10 shows the trace of the covariance of the estimation of the forward filter as it runs from $t = 0$ to $t = 5$, the backward filter as it runs from $t = 10$ back to $t = 5$, and the smoothed estimate at $t = 5$. The forward and backward filters both converge to the same steady-state value, even though the forward filter was initialized to a covariance of 20 for both the position and velocity estimation errors, and the backward filter was initialized to an infinite covariance. The smoothed filter has a covariance of about 7.6, which shows the dramatic improvement that can be obtained in estimation accuracy when smoothing is used.

∇∇∇

Figure 9.10 This shows the trace of the estimation-error covariance for Example 9.3. The forward filter runs from $t = 0$ to $t = 5$, the backward filter runs from $t = 10$ to $t = 5$, and the trace of the covariance of the smoothed estimate is shown at $t = 5$.

9.4.2 RTS smoothing

Several other forms of the fixed-interval smoother have been obtained. One of the most common is the smoother that was presented by Rauch, Tung, and Striebel, usually called the RTS smoother [Rau65]. The RTS smoother is more computationally efficient than the smoother presented in the previous section because we do not need to directly compute the backward estimate or covariance in order to get the smoothed estimate and covariance. In order to obtain the RTS smoother, we will first look at the smoothed covariance given in Equation (9.78) and obtain an equivalent expression that does not use P_{bm}. Then we will look at the smoothed estimate given in Equation (9.78), which uses the gain K_f, which depends on P_{bm}, and obtain an equivalent expression that does not use P_{bm} or $\hat{x}_{bm}$.

9.4.2.1 RTS covariance update First consider the smoothed covariance given in Equation (9.78). This can be written as

$$\begin{aligned} P_m &= \left[(P_{fm}^+)^{-1} + (P_{bm}^-)^{-1}\right]^{-1} \\ &= P_{fm}^+ - P_{fm}^+(P_{fm}^+ + P_{bm}^-)^{-1}P_{fm}^+ \end{aligned} \tag{9.83}$$

where the second expression comes from an application of the matrix inversion lemma to the first expression (see Problem 9.3). From Equation (9.72) we see that

$$P_{bm}^- = F_m^{-1}\left[P_{b,m+1}^+ + Q_m\right]F_m^{-T} \tag{9.84}$$

Substituting this into the expression $(P_{fm}^+ + P_{bm}^-)^{-1}$ gives the following:

$$
\begin{aligned}
(P_{fm}^{+}+P_{bm}^{-})^{-1} &= \left[P_{fm}^{+}+F_m^{-1}(P_{b,m+1}^{+}+Q_m)F_m^{-T}\right]^{-1} \\
&= \left[F_m^{-1}F_mP_{fm}^{+}F_m^{T}F_m^{-T}+F_m^{-1}(P_{b,m+1}^{+}+Q_m)F_m^{-T}\right]^{-1} \\
&= \left[F_m^{-1}(F_mP_{fm}^{+}F_m^{T}+P_{b,m+1}^{+}+Q_m)F_m^{-T}\right]^{-1} \\
&= F_m^{T}(F_mP_{fm}^{+}F_m^{T}+P_{b,m+1}^{+}+Q_m)^{-1}F_m \\
&= F_m^{T}(P_{f,m+1}^{-}+P_{b,m+1}^{+})^{-1}F_m
\end{aligned}
\tag{9.85}
$$

From Equations (6.26) and (9.76) recall that

$$
\begin{aligned}
\mathcal{I}_{fm}^{+} &= \mathcal{I}_{fm}^{-}+H_m^{T}R_m^{-1}H_m \\
\mathcal{I}_{bm}^{+} &= \mathcal{I}_{bm}^{-}+H_m^{T}R_m^{-1}H_m
\end{aligned}
\tag{9.86}
$$

We can combine these two equations to obtain

$$
\mathcal{I}_{b,m+1}^{+}=\mathcal{I}_{b,m+1}^{-}+\mathcal{I}_{f,m+1}^{+}-\mathcal{I}_{f,m+1}^{-}
\tag{9.87}
$$

Substituting this into Equation (9.78) gives

$$
\begin{aligned}
P_{m+1} &= \left[\mathcal{I}_{f,m+1}^{+}+\mathcal{I}_{b,m+1}^{-}\right]^{-1} \\
&= \left[\mathcal{I}_{b,m+1}^{+}+\mathcal{I}_{f,m+1}^{-}\right]^{-1} \\
P_{m+1}^{-1} &= \mathcal{I}_{b,m+1}^{+}+\mathcal{I}_{f,m+1}^{-} \\
P_{b,m+1}^{+} &= \left[P_{m+1}^{-1}-\mathcal{I}_{f,m+1}^{-}\right]^{-1}
\end{aligned}
\tag{9.88}
$$

Substituting this into Equation (9.85) gives

$$
\begin{aligned}
(P_{fm}^{+}+P_{bm}^{-})^{-1} &= F_m^{T}\left[P_{f,m+1}^{-}+\left(P_{m+1}^{-1}-\mathcal{I}_{f,m+1}^{-}\right)^{-1}\right]^{-1}F_m \\
&= F_m^{T}\mathcal{I}_{f,m+1}^{-}\left[\mathcal{I}_{f,m+1}^{-}+\mathcal{I}_{f,m+1}^{-}\left(P_{m+1}^{-1}-\mathcal{I}_{f,m+1}^{-}\right)^{-1}\mathcal{I}_{f,m+1}^{-}\right]^{-1}\mathcal{I}_{f,m+1}^{-}F_m \\
&= F_m^{T}\mathcal{I}_{f,m+1}^{-}\left(P_{f,m+1}^{-}-P_{m+1}\right)\mathcal{I}_{f,m+1}^{-}F_m
\end{aligned}
\tag{9.89}
$$

where the last equality comes from an application of the matrix inversion lemma. Substituting this expression into Equation (9.83) gives

$$
P_m=P_{fm}^{+}-K_m(P_{f,m+1}^{-}-P_{m+1})K_m^{T}
\tag{9.90}
$$

where the smoother gain K_m is given as

$$
K_m=P_{fm}^{+}F_m^{T}\mathcal{I}_{f,m+1}^{-}
\tag{9.91}
$$

The covariance update equation for P_m is not a function of the backward covariance. The smoother covariance P_m can be solved by using only the forward covariance P_{fm}, which reduces the computational effort (compared to the algorithm presented in Section 9.4.1).

9.4.2.2 RTS state estimate update Next we consider the smoothed estimate $\hat{x}_m$ given in Equation (9.78). We will find an equivalent expression that does not use P_{bm} or $\hat{x}_{bm}$. In order to do this we will first need to establish a few lemmas.

Lemma 1

$$F_{k-1}^{-1}Q_{k-1}F_{k-1}^{-T} = F_{k-1}^{-1}P_{fk}^{-}F_{k-1}^{-T} - P_{f,k-1}^{+} \tag{9.92}$$

Proof: From Equation (9.67) we see that

$$P_{fk}^{-} = F_{k-1}P_{f,k-1}^{+}F_{k-1}^{T} + Q_{k-1} \tag{9.93}$$

Rearranging this equation gives

$$Q_{k-1} = P_{fk}^{-} - F_{k-1}P_{f,k-1}^{+}F_{k-1}^{T} \tag{9.94}$$

Premultiplying both sides by F_{k-1}^{-1} and postmultiplying both sides by F_{k-1}^{-T} gives the desired result.
QED

Lemma 2 *The a posteriori covariance P_{bk}^{+} of the backward filter satisfies the equation*

$$P_{bk}^{+} = (P_{fk}^{-} + P_{bk}^{+})\mathcal{I}_{fk}^{-}P_k \tag{9.95}$$

Proof: From Equation (9.78) we obtain

$$\begin{aligned} I &= (\mathcal{I}_{bk}^{+} + \mathcal{I}_{fk}^{-})P_k \\ P_{bk}^{+} &= (I + P_{bk}^{+}\mathcal{I}_{fk}^{-})P_k \\ &= P_k + P_{bk}^{+}\mathcal{I}_{fk}^{-}P_k \\ &= P_{fk}^{-}\mathcal{I}_{fk}^{-}P_k + P_{bk}^{+}\mathcal{I}_{fk}^{-}P_k \\ &= (P_{fk}^{-} + P_{bk}^{+})\mathcal{I}_{fk}^{-}P_k \end{aligned} \tag{9.96}$$

QED

Lemma 3 *The covariances of the forward and backward filters satisfy the equation*

$$P_{fk}^{-} + P_{bk}^{+} = F_{k-1}(P_{f,k-1}^{+} + P_{b,k-1}^{-})F_{k-1}^{T} \tag{9.97}$$

Proof: From Equation (9.67) and (9.72) we see that

$$\begin{aligned} P_{f,k-1}^{+} &= F_{k-1}^{-1}P_{fk}^{-}F_{k-1}^{-T} - F_{k-1}Q_{k-1}F_{k-1}^{-T} \\ P_{b,k-1}^{-} &= F_{k-1}^{-1}P_{bk}^{+}F_{k-1}^{-T} + F_{k-1}^{-1}Q_{k-1}F_{k-1}^{-T} \end{aligned} \tag{9.98}$$

Adding these two equations and rearranging gives

$$\begin{aligned} P_{f,k-1}^{+} + P_{b,k-1}^{-} &= F_{k-1}^{-1}(P_{fk}^{-} + P_{bk}^{+})F_{k-1}^{-T} \\ P_{fk}^{-} + P_{bk}^{+} &= F_{k-1}(P_{f,k-1}^{+} + P_{b,k-1}^{-})F_{k-1}^{T} \end{aligned} \tag{9.99}$$

QED

Lemma 4 *The smoothed estimate $\hat{x}_k$ can be written as*

$$\hat{x}_k = P_k\mathcal{I}_{fk}^{+}\hat{x}_{fk}^{-} - P_kH_k^{T}R_k^{-1}H_k\hat{x}_{fk}^{-} + P_ks_k^{+} \tag{9.100}$$

Proof: From Equations (9.69) and (9.82) we have

$$\begin{aligned}\hat{x}_k &= P_k\mathcal{I}_{fk}^+\hat{x}_{fk}^+ + P_k\mathcal{I}_{bk}^-\hat{x}_{bk}^- \\ &= P_k\mathcal{I}_{fk}^+\hat{x}_{fk}^+ + P_k s_k^- \end{aligned} \tag{9.101}$$

From Equation (9.76) we see that

$$s_k^- = s_k^+ - H_k^T R_k^{-1} y_k \tag{9.102}$$

Substitute this expression for s_k^-, and the expression for $\hat{x}_{fk}^+$ from Equation (9.67), into Equation (9.101) to obtain

$$\hat{x}_k = P_k\mathcal{I}_{fk}^+\hat{x}_{fk}^- + P_k\mathcal{I}_{fk}^+K_{fk}(y_k - H_k\hat{x}_{fk}^-) + P_k s_k^+ - P_k H_k^T R_k^{-1} y_k \tag{9.103}$$

Now substitute $P_{fk}^+H_k^TR_k^{-1}$ for K_{fk} [from Equation (9.67)] in the above equation to obtain

$$\begin{aligned}\hat{x}_k &= P_k\mathcal{I}_{fk}^+\hat{x}_{fk}^- + P_k\mathcal{I}_{fk}^+P_{fk}^+H_k^TR_k^{-1}(y_k - H_k\hat{x}_{fk}^-) + P_k s_k^+ - P_k H_k^T R_k^{-1} y_k \\ &= P_k\mathcal{I}_{fk}^+\hat{x}_{fk}^- + P_k H_k^T R_k^{-1}(y_k - H_k\hat{x}_{fk}^-) + P_k s_k^+ - P_k H_k^T R_k^{-1} y_k \\ &= P_k\mathcal{I}_{fk}^+\hat{x}_{fk}^- - P_k H_k^T R_k^{-1} H_k\hat{x}_{fk}^- + P_k s_k^+ \end{aligned} \tag{9.104}$$

QED

Lemma 5

$$(P_{f,k-1}^+ + P_{b,k-1}^-)^{-1} = F_{k-1}^T\mathcal{I}_{fk}^-(P_{fk}^- - P_k)\mathcal{I}_{fk}^-F_{k-1} \tag{9.105}$$

Proof: Recall from Equations (6.26) and (9.72) that

$$\begin{aligned}\mathcal{I}_{fk}^+ &= \mathcal{I}_{fk}^- + H_k^TR_k^{-1}H_k \\ \mathcal{I}_{bk}^+ &= \mathcal{I}_{bk}^- + H_k^TR_k^{-1}H_k \end{aligned} \tag{9.106}$$

Combining these two equations gives

$$\begin{aligned}\mathcal{I}_{bk}^+ &= \mathcal{I}_{bk}^- + \mathcal{I}_{fk}^+ - \mathcal{I}_{fk}^- \\ &= \left[\left(\mathcal{I}_{bk}^- + \mathcal{I}_{fk}^+\right)^{-1}\right]^{-1} - \mathcal{I}_{fk}^- \\ &= P_k^{-1} - \mathcal{I}_{fk}^- \\ P_{bk}^+ &= \left(\mathcal{I}_k - \mathcal{I}_{fk}^-\right)^{-1} \end{aligned} \tag{9.107}$$

where we have used Equation (9.78) in the above derivation. Substitute this expression for P_{bk}^+ into Equation (9.97) to obtain

$$\begin{aligned}F_{k-1}(P_{f,k-1}^+ + P_{b,k-1}^-)F_{k-1}^T &= P_{fk}^- + P_{bk}^+ \\ &= P_{fk}^- + \left(\mathcal{I}_k - \mathcal{I}_{fk}^-\right)^{-1} \end{aligned} \tag{9.108}$$

Invert both sides to obtain

$$
\begin{aligned}
F_{k-1}^{-T}(P_{f,k-1}^{+}+P_{b,k-1}^{-})^{-1}F_{k-1}^{-1} &= \left[P_{fk}^{-}+\left(\mathcal{I}_k-\mathcal{I}_{fk}^{-}\right)^{-1}\right]^{-1} \\
(P_{f,k-1}^{+}+P_{b,k-1}^{-})^{-1} &= F_{k-1}^{T}\left[P_{fk}^{-}+\left(\mathcal{I}_k-\mathcal{I}_{fk}^{-}\right)^{-1}\right]^{-1}F_{k-1} \\
&= F_{k-1}^{T}\left[P_{fk}^{-}\mathcal{I}_{fk}^{-}P_{fk}^{-}+P_{fk}^{-}\mathcal{I}_{fk}^{-}\left(\mathcal{I}_k-\mathcal{I}_{fk}^{-}\right)^{-1}\mathcal{I}_{fk}^{-}P_{fk}^{-}\right]^{-1}F_{k-1} \\
&= F_{k-1}^{T}\mathcal{I}_{fk}^{-}\left[\mathcal{I}_{fk}^{-}+\mathcal{I}_{fk}^{-}\left(\mathcal{I}_k-\mathcal{I}_{fk}^{-}\right)^{-1}\mathcal{I}_{fk}^{-}\right]^{-1}\mathcal{I}_{fk}^{-}F_{k-1} \qquad (9.109)
\end{aligned}
$$

Now apply the matrix inversion lemma to the term $(\mathcal{I}_k-\mathcal{I}_{fk}^{-})^{-1}$ *in the above equation. This results in*

$$
\begin{aligned}
&(P_{f,k-1}^{+}+P_{b,k-1}^{-})^{-1} \\
&= F_{k-1}^{T}\mathcal{I}_{fk}^{-}\left[\mathcal{I}_{fk}^{-}+\mathcal{I}_{fk}^{-}\left(-P_{fk}^{-}-P_{fk}^{-}(-P_{fk}^{-}+P_k)^{-1}P_{fk}^{-}\right)\mathcal{I}_{fk}^{-}\right]^{-1}\mathcal{I}_{fk}^{-}F_{k-1} \\
&= F_{k-1}^{T}\mathcal{I}_{fk}^{-}\left[\mathcal{I}_{fk}^{-}+\left(-I-(-P_{fk}^{-}+P_k)^{-1}P_{fk}^{-}\right)\mathcal{I}_{fk}^{-}\right]^{-1}\mathcal{I}_{fk}^{-}F_{k-1} \\
&= F_{k-1}^{T}\mathcal{I}_{fk}^{-}\left[\mathcal{I}_{fk}^{-}-\mathcal{I}_{fk}^{-}-(P_{fk}^{-}+P_k)^{-1}\right]^{-1}\mathcal{I}_{fk}^{-}F_{k-1} \\
&= F_{k-1}^{T}\mathcal{I}_{fk}^{-}(P_{fk}^{-}-P_k)\mathcal{I}_{fk}^{-}F_{k-1} \qquad (9.110)
\end{aligned}
$$

QED

With the above lemmas we now have the tools that we need to obtain an alternate expression for the smoothed estimate. Starting with the expression for s_{k-1}^{-} in Equation (9.76), and substituting the expression for $\mathcal{I}_{b,k-1}^{-}$ from Equation (9.72) gives

$$
\begin{aligned}
s_{k-1}^{-} &= \mathcal{I}_{bk}^{-}F_{k-1}^{-1}P_{bk}^{+}s_k^{+} \\
&= F_{k-1}^{T}\left[Q_{k-1}^{-1}-Q_{k-1}^{-1}\left(\mathcal{I}_{bk}^{+}+Q_{k-1}^{-1}\right)^{-1}Q_{k-1}^{-1}\right]F_{k-1}F_{k-1}^{-1}P_{bk}^{+}s_k^{+} \\
&= F_{k-1}^{T}Q_{k-1}^{-1}\left[I-\left(\mathcal{I}_{bk}^{+}+Q_{k-1}^{-1}\right)^{-1}Q_{k-1}^{-1}\right]P_{bk}^{+}s_k^{+} \\
&= F_{k-1}^{T}Q_{k-1}^{-1}\left(\mathcal{I}_{bk}^{+}+Q_{k-1}^{-1}\right)\left(\mathcal{I}_{bk}^{+}+Q_{k-1}^{-1}-Q_{k-1}^{-1}\right)P_{bk}^{+}s_k^{+} \\
&= F_{k-1}^{T}Q_{k-1}^{-1}\left(\mathcal{I}_{bk}^{+}+Q_{k-1}^{-1}\right)^{-1}s_k^{+} \\
&= F_{k-1}^{T}(I+\mathcal{I}_{bk}^{+}Q_{k-1})^{-1}s_k^{+} \qquad (9.111)
\end{aligned}
$$

Rearranging this equation gives

$$
(I+\mathcal{I}_{bk}^{+}Q_{k-1})F_{k-1}^{-T}s_{k-1}^{-}=s_k^{+} \qquad (9.112)
$$

Multiplying out this equation, and premultiplying both sides by $F_{k-1}^{-1}P_{bk}^{+}$, gives

$$
F_{k-1}^{-1}P_{bk}^{+}F_{k-1}^{-T}s_{k-1}^{-}+F_{k-1}^{-1}Q_{k-1}F_{k-1}^{-T}s_{k-1}^{-}=F_{k-1}^{-1}P_{bk}^{+}s_k^{+} \qquad (9.113)
$$

Substituting for $F_{k-1}^{-1}Q_{k-1}F_{k-1}^{-T}$ from Equation (9.92) gives

$$\begin{aligned} F_{k-1}^{-1}P_{bk}^{+}F_{k-1}^{-T}s_{k-1}^{-} + F_{k-1}^{-1}P_{fk}^{-}F_{k-1}^{-T}s_{k-1}^{-} - P_{f,k-1}^{+}s_{k-1}^{-} &= F_{k-1}^{-1}P_{bk}^{+}s_{k}^{+} \\ \left[F_{k-1}^{-1}(P_{fk}^{-}+P_{bk}^{+})F_{k-1}^{-T} - P_{f,k-1}^{+}\right]s_{k-1}^{-} &= F_{k-1}^{-1}P_{bk}^{+}s_{k}^{+} \\ \left[(P_{fk}^{-}+P_{bk}^{+})F_{k-1}^{-T} - F_{k-1}P_{f,k-1}^{+}\right]s_{k-1}^{-} &= P_{bk}^{+}s_{k}^{+} \end{aligned} \tag{9.114}$$

Substituting in this expression for P_{bk}^{+} from Equation (9.95) gives

$$\left[(P_{fk}^{-}+P_{bk}^{+})F_{k-1}^{-T} - F_{k-1}P_{f,k-1}^{+}\right]s_{k-1}^{-} = (P_{fk}^{-}+P_{bk}^{+})\mathcal{I}_{fk}^{-}P_{k}s_{k}^{+} \tag{9.115}$$

Substituting for $(P_{fk}^{-}+P_{bk}^{+})$ from Equation (9.97) on both sides of this expression gives

$$\left[F_{k-1}(P_{f,k-1}^{+}+P_{b,k-1}^{-}) - F_{k-1}P_{f,k-1}^{+}\right]s_{k-1}^{-} = F_{k-1}(P_{f,k-1}^{+}+P_{b,k-1}^{-})F_{k-1}^{T}\mathcal{I}_{fk}^{-}P_{k}s_{k}^{+} \tag{9.116}$$

Premultiplying both sides by $(P_{f,k-1}^{+}+P_{b,k-1}^{-})^{-1}F_{k-1}^{-1}$ gives

$$\left[I - (P_{f,k-1}^{+}+P_{b,k-1}^{-})^{-1}P_{f,k-1}^{+}\right]s_{k-1}^{-} = F_{k-1}^{T}\mathcal{I}_{fk}^{-}P_{k}s_{k}^{+} \tag{9.117}$$

Substituting Equation (9.105) for $(P_{f,k-1}^{+}+P_{b,k-1}^{-})^{-1}$ gives

$$s_{k-1}^{-} - F_{k-1}^{T}\mathcal{I}_{fk}^{-}(P_{fk}^{-}-P_{k})\mathcal{I}_{fk}^{-}F_{k-1}P_{f,k-1}^{+}s_{k-1}^{-} = F_{k-1}^{T}\mathcal{I}_{fk}^{-}P_{k}s_{k}^{+} \tag{9.118}$$

Now from Equation (9.105) we see that

$$-(P_{f,k-1}^{+}+P_{b,k-1}^{-})^{-1}F_{k-1}^{-1}\hat{x}_{fk}^{-} = F_{k-1}^{T}\mathcal{I}_{fk}^{-}(P_{k}-P_{fk}^{-})\mathcal{I}_{fk}^{-}\hat{x}_{fk}^{-} \tag{9.119}$$

So we can add the two sides of this equation to the two sides of Equation (9.118) to get

$$\begin{aligned} &s_{k-1}^{-} - F_{k-1}^{T}\mathcal{I}_{fk}^{-}(P_{fk}^{-}-P_{k})\mathcal{I}_{fk}^{-}F_{k-1}P_{f,k-1}^{+}s_{k-1}^{-} - (P_{f,k-1}^{+}+P_{b,k-1}^{-})^{-1}F_{k-1}^{-1}\hat{x}_{fk}^{-} \\ &= F_{k-1}^{T}\mathcal{I}_{fk}^{-}P_{k}s_{k}^{+} + F_{k-1}^{T}\mathcal{I}_{fk}^{-}(P_{k}-P_{fk}^{-})\mathcal{I}_{fk}^{-}\hat{x}_{fk}^{-} \end{aligned} \tag{9.120}$$

Now use Equation (9.100) to substitute for $P_{k}s_{k}^{+}$ in the above equation and obtain

$$\begin{aligned} &s_{k-1}^{-} - F_{k-1}^{T}\mathcal{I}_{fk}^{-}(P_{fk}^{-}-P_{k})\mathcal{I}_{fk}^{-}F_{k-1}P_{f,k-1}^{+}s_{k-1}^{-} - (P_{f,k-1}^{+}+P_{b,k-1}^{-})^{-1}F_{k-1}^{-1}\hat{x}_{fk}^{-} \\ &= F_{k-1}^{T}\mathcal{I}_{fk}^{-}\hat{x}_{k} - F_{k-1}^{T}\mathcal{I}_{fk}^{-}P_{k}\mathcal{I}_{fk}^{+}\hat{x}_{fk}^{-} + \\ &F_{k-1}^{T}\mathcal{I}_{fk}^{-}P_{k}H_{k}^{T}R_{k}^{-1}H_{k}\hat{x}_{fk}^{-} + F_{k-1}^{T}\mathcal{I}_{fk}^{-}(P_{k}-P_{fk}^{-})\mathcal{I}_{fk}^{-}\hat{x}_{fk}^{-} \end{aligned} \tag{9.121}$$

Rearrange this equation to obtain

$$\begin{aligned} &s_{k-1}^{-} - F_{k-1}^{T}\mathcal{I}_{fk}^{-}(P_{fk}^{-}-P_{k})\mathcal{I}_{fk}^{-}F_{k-1}P_{f,k-1}^{+}s_{k-1}^{-} + \\ &\Big[-(P_{f,k-1}^{+}+P_{b,k-1}^{-})^{-1}F_{k-1}^{-1} + F_{k-1}^{T}\mathcal{I}_{fk}^{-}P_{k}\mathcal{I}_{fk}^{+} - F_{k-1}^{T}\mathcal{I}_{fk}^{-}P_{k}H_{k}^{T}R_{k}^{-1}H_{k} - \\ &F_{k-1}^{T}\mathcal{I}_{fk}^{-}P_{k}\mathcal{I}_{fk}^{-}\Big]\hat{x}_{fk}^{-} = F_{k-1}^{T}\mathcal{I}_{fk}^{-}(\hat{x}_{k}-\hat{x}_{fk}^{-}) \end{aligned} \tag{9.122}$$

From Equation (9.106) we see that $\mathcal{I}_{fk}^{+} - \mathcal{I}_{fk}^{-} = H_k^T R_k^{-1} H_k$. Also note that part of the coefficient of $\hat{x}_{fk}^{-}$ on the left side of the above equation can be expressed as

$$(P_{f,k-1}^{+} + P_{b,k-1}^{-})^{-1} F_{k-1}^{-1} = \mathcal{I}_{b,k-1}^{-}(I + P_{f,k-1}^{+}\mathcal{I}_{b,k-1}^{-})^{-1} F_{k-1}^{-1} \tag{9.123}$$

From Equation (9.67) we see that $F_{k-1}^{-1}\hat{x}_{fk}^{-} = \hat{x}_{f,k-1}^{+}$. Therefore Equation (9.122) can be written as

$$\begin{aligned} & s_{k-1}^{-} - F_{k-1}^{T}\mathcal{I}_{fk}^{-}(P_{fk}^{-} - P_k)\mathcal{I}_{fk}^{-}F_{k-1}P_{f,k-1}^{+}s_{k-1}^{-} - \\ & \mathcal{I}_{b,k-1}^{-}(I + P_{f,k-1}^{+}\mathcal{I}_{b,k-1}^{-})^{-1}\hat{x}_{f,k-1}^{+} = F_{k-1}^{T}\mathcal{I}_{fk}^{-}(\hat{x}_k - \hat{x}_{fk}^{-}) \end{aligned} \tag{9.124}$$

Now substitute for P_k from Equation (9.90) and use Equation (9.91) in the above equation to obtain

$$\begin{aligned} & s_{k-1}^{-} - F_{k-1}^{T}\mathcal{I}_{fk}^{-}(P_{fk}^{-} - P_{fk}^{+} + K_k P_{f,k+1}^{-} K_k^T - K_k P_{k+1} K_k^T) K_{k-1}^T s_{k-1}^{-} - \\ & \mathcal{I}_{b,k-1}^{-}(I + P_{f,k-1}^{+}\mathcal{I}_{b,k-1}^{-})^{-1}\hat{x}_{f,k-1}^{+} = F_{k-1}^{T}\mathcal{I}_{fk}^{-}(\hat{x}_k - \hat{x}_{fk}^{-}) \end{aligned} \tag{9.125}$$

Premultiplying both sides by $P_{f,k-1}^{+}$ gives

$$\begin{aligned} & \left[P_{f,k-1}^{+} - P_{f,k-1}^{+}F_{k-1}^{T}\mathcal{I}_{fk}^{-}(P_{fk}^{-} - P_{fk}^{+} + K_k P_{f,k+1}^{-} K_k^T - K_k P_{k+1} K_k^T) K_{k-1}^T\right] s_{k-1}^{-} \\ & -P_{f,k-1}^{+}\mathcal{I}_{b,k-1}^{-}(I + P_{f,k-1}^{+}\mathcal{I}_{b,k-1}^{-})^{-1}\hat{x}_{f,k-1}^{+} = K_{k-1}(\hat{x}_k - \hat{x}_{fk}^{-}) \end{aligned} \tag{9.126}$$

Now use Equation (9.91) to notice that the coefficient of s_{k-1}^{-} on the left side of the above equation can be written as

$$P_{f,k-1}^{+} - K_{k-1}\left(P_{fk}^{-} - P_{fk}^{+} + K_k P_{f,k+1}^{-} K_k^T - K_k P_{k+1} K_k^T\right) K_{k-1}^T \tag{9.127}$$

Using Equation (9.90) to substitute for $K_k P_{k+1} K_k^T$ allows us to write the above expression as

$$\begin{aligned} & P_{f,k-1}^{+} - K_{k-1}\left(P_{fk}^{-} - P_{fk}^{+} + K_k P_{f,k+1}^{-} K_k^T - P_k + P_{fk}^{+} - K_k P_{f,k+1}^{-} K_k^T\right) K_{k-1}^T \\ & \quad = P_{f,k-1}^{+} - K_{k-1}P_{fk}^{-}K_{k-1}^T + K_{k-1}P_k K_{k-1}^T \\ & \quad = P_{f,k-1}^{+} - K_{k-1}P_{fk}^{-}K_{k-1}^T + P_{k-1} - P_{f,k-1}^{+} + K_{k-1}P_{fk}^{-}K_{k-1}^T \\ & \quad = P_{k-1} \end{aligned} \tag{9.128}$$

Since this is the coefficient of s_{k-1}^{-} in Equation (9.126), we can write that equation as

$$P_{k-1}s_{k-1}^{-} - P_{f,k-1}^{+}\mathcal{I}_{b,k-1}^{-}(I + P_{f,k-1}^{+}\mathcal{I}_{b,k-1}^{-})^{-1}\hat{x}_{f,k-1}^{+} = K_{k-1}(\hat{x}_k - \hat{x}_{fk}^{-}) \tag{9.129}$$

Now from Equations (9.78) and (9.82) we see that

$$\begin{aligned} \hat{x}_k &= (\mathcal{I}_{fk}^{+} + \mathcal{I}_{bk}^{-})^{-1}\mathcal{I}_{fk}^{+}\hat{x}_{fk}^{+} + P_k\mathcal{I}_{bk}^{-}\hat{x}_{bk}^{-} \\ &= (I + P_{fk}^{+}\mathcal{I}_{bk}^{-})^{-1}\hat{x}_{fk}^{+} + P_k s_k^{-} \end{aligned} \tag{9.130}$$

From this we see that

$$\begin{aligned} \hat{x}_k - \hat{x}_{fk}^{+} &= \left[(I + P_{fk}^{+}\mathcal{I}_{bk}^{-})^{-1} - I\right]\hat{x}_{fk}^{+} + P_k s_k^{-} \\ &= \left[I - (I + P_{fk}^{+}\mathcal{I}_{bk}^{-})\right](I + P_{fk}^{+}\mathcal{I}_{bk}^{-})^{-1}\hat{x}_{fk}^{+} + P_k s_k^{-} \\ &= -P_{fk}^{+}\mathcal{I}_{bk}^{-}(I + P_{fk}^{+}\mathcal{I}_{bk}^{-})^{-1}\hat{x}_{fk}^{+} + P_k s_k^{-} \end{aligned} \tag{9.131}$$

Rewriting the above equation with the time subscripts $(k-1)$ and then substituting for the left side of Equation (9.129) gives

$$\hat{x}_{k-1} - \hat{x}^{+}_{f,k-1} = K_{k-1}(\hat{x}_k - \hat{x}^{-}_{fk}) \tag{9.132}$$

from which we can write

$$\hat{x}_k = \hat{x}^{+}_{fk} + K_k(\hat{x}_{k+1} - \hat{x}^{-}_{f,k+1}) \tag{9.133}$$

This gives the smoothed estimate $\hat{x}_k$ without needing to explicitly calculate the backward estimate. The RTS smoother is implemented by first running the standard Kalman filter of Equation (9.67) forward in time to the final time, and then implementing Equations (9.90), (9.91), and (9.133) backward in time. The RTS smoother can be summarized as follows.

The RTS smoother

1. The system model is given as follows:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \end{aligned} \tag{9.134}$$

2. Initialize the forward filter as follows:

$$\begin{aligned} \hat{x}_{f0} &= E(x_0) \\ P^{+}_{f0} &= E\left[(x_0 - \hat{x}_{f0})(x_0 - \hat{x}_{f0})^T\right] \end{aligned} \tag{9.135}$$

3. For $k = 1, \cdots, N$ (where N is the final time), execute the standard forward Kalman filter:

$$\begin{aligned} P^{-}_{fk} &= F_{k-1}P^{+}_{f,k-1}F^T_{k-1} + Q_{k-1} \\ K_{fk} &= P^{-}_{fk}H^T_k(H_kP^{-}_{fk}H^T_k + R_k)^{-1} \\ &= P^{+}_{fk}H^T_kR^{-1}_k \\ \hat{x}^{-}_{fk} &= F_{k-1}\hat{x}^{+}_{f,k-1} + G_{k-1}u_{k-1} \\ \hat{x}^{+}_{fk} &= \hat{x}^{-}_{fk} + K_{fk}(y_k - H_k\hat{x}^{-}_{fk}) \\ P^{+}_{fk} &= (I - K_{fk}H_k)P^{-}_{fk}(I - K_{fk}H_k)^T + K_{fk}R_kK^T_{fk} \\ &= \left[(P^{-}_{fk})^{-1} + H^T_kR^{-1}_kH_k\right]^{-1} \\ &= (I - K_{fk}H_k)P^{-}_{fk} \end{aligned} \tag{9.136}$$

4. Initialize the RTS smoother as follows:

$$\begin{aligned} \hat{x}_N &= \hat{x}^{+}_{fN} \\ P_N &= P^{+}_{fN} \end{aligned} \tag{9.137}$$

5. For $k = N-1, \cdots, 1, 0$, execute the following RTS smoother equations:

$$\begin{aligned}
\mathcal{I}_{f,k+1}^{-} &= \left(P_{f,k+1}^{-}\right)^{-1} \\
K_k &= P_{fk}^{+} F_k^T \mathcal{I}_{f,k+1}^{-} \\
P_k &= P_{fk}^{+} - K_k(P_{f,k+1}^{-} - P_{k+1})K_k^T \\
\hat{x}_k &= \hat{x}_{fk}^{+} + K_k(\hat{x}_{k+1} - \hat{x}_{f,k+1}^{-})
\end{aligned} \tag{9.138}$$

9.5 SUMMARY

In this chapter we derived the optimal smoothing filters. These filters, sometimes called retrodiction filters [Bar01], include the following variants.

- $\hat{x}_{j,k} = E(x_j|y_1, \cdots, y_{k-1})$ $(k \geq j)$ is the output of the fixed-point smoother. In this filter we find the estimate of the state at the fixed time j when measurements continue to arrive at the filter at times greater than j. The time index j is fixed while k continues to increase as we obtain more measurements.
- $\hat{x}_{k-N,k} = E(x_{k-N}|y_1, \cdots, y_k)$ for a fixed N is the output of the fixed-lag smoother. In this filter we find the estimate of the state at each time k while using measurements up to and including time $(k+N)$. The time index k varies while N remains fixed.
- $\hat{x}_{k,N} = E(x_k|y_1, \cdots, y_N)$ for a fixed N is the output of the fixed-interval smoother. In this filter we find the estimate of the state at each time k while using measurements up to and including time N. The time index k varies while the total number of measurements N is fixed. The two formulas we derived for this type of smoothing included the forward–backward smoother and the RTS smoother.

Just as steady-state filters can be used for standard filtering, we can also derive steady-state smoothers to save computational effort [Gel74]. An early survey of smoothing algorithms is given in [Med73].

PROBLEMS

Written exercises

9.1 Prove or disprove the following conjecture: The trace of the inverse of a matrix is equal to the inverse of the trace of the matrix.

9.2 Show that $(A+B)^{-1} = B^{-1}(AB^{-1}+I)^{-1}$.

9.3 Derive Equation (9.83).

9.4 Consider a scalar system with $F = 1$, $H = 1$, and $R = 2Q$.

a) What is the steady-state value of the *a priori* estimation-error covariance P_k^-?

b) Suppose that after the Kalman filter has reached steady state, the fixed-point smoother begins to operate. Find a closed-form solution to the covariance of the smoothed estimate Π_k as a function of the time index k. What is the limiting value of Π_k as $k \to \infty$?

9.5 Repeat Problem 9.4 for the case $R = 12Q$. What is the percent improvement in the estimation-error covariance due to smoothing? Explain why the percent improvement due to smoothing for this case differs in the way that it does from the results of Problem 9.4.

9.6 Consider a scalar system with $F = 1$, $H = 1$, and $R = 2Q$. Suppose that the fixed-lag smoother for this system is in steady state so that $P_{k+1}^- = P_k^-$, $L_{k+1,i} = L_{k,i}$, $P_{k+1}^{i,i} = P_k^{i,i}$, and $P_{k+1}^{0,i} = P_k^{0,i}$, for $i = 1, \cdots, N+1$. Find closed-form expressions for P_k^-, $L_{k,i}$, $P_k^{i,i}$, and $P_k^{0,i}$ as functions of i. What is the limit as $i \to \infty$ of $L_{k,i}$, $P_k^{i,i}$, and $P_k^{0,i}$?

9.7 Suppose you have a fixed-lag smoother as shown in Equation (9.43) that is in steady state. How do the eigenvalues of the fixed-lag smoother relate to the eigenvalues of the standard Kalman filter? What do you conclude about the stability of the fixed-lag smoother?

9.8 Solve Equation (9.10) for $(y_k - H_k\hat{x}_k^-)$ [assuming that $\rho(L_k) = r$, where r is the number of measurements in the system]. Substitute the resulting expression for $(y_k - H_k\hat{x}_k^-)$ in the fixed-lag smoother equation for $\hat{x}_{k+1-i,k}$ to show that the smoothed state estimate can be driven by the state estimates without any input from the measurements [And79].

9.9 Suppose that $\hat{x}_f$ and $\hat{x}_b$ are unbiased estimates of x, and $\hat{x} = K_f\hat{x}_f + K_b\hat{x}_b$. Show that if $\hat{x}$ is an unbiased estimate of x, then we must have $K_f + K_b = I$.

9.10 Consider a scalar system with $F = 1$, $H = 1$, and $R = 2Q$. Use the forward–backward smoother of Section 9.4.1 to find the steady-state value of the covariance of the smoothed state estimate.

9.11 Consider a scalar system with $F = 1$, $H = 1$, and $R = 2Q$. Use the RTS smoother of Section 9.4.2 to find the steady-state value of the covariance of the smoothed state estimate.

9.12 Consider a scalar system with $F = 1$, $H = 1$, and $R = 2Q$. Suppose that the forward filter has reached steady state. Use the RTS smoother of Section 9.4.2 to find the covariance of the smoothed state estimate for $k = N, N-1, N-2, N-3$, and $N-4$. At what point does the covariance of the smoothed state estimate get within 1% of its steady-state value?

9.13 Repeat Problem 9.12 for $R = 12Q$. How do you intuitively explain the quicker convergence of P_k to steady state?

9.14 Use the RTS smoother equations to show that constant states are not smoothable. That is, if $F = I$ and $Q = 0$, then $P_k = P_{fN}^+$ for all k.

Computer exercises

9.15 Consider the second-order system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t)$$

where $\omega = 6$ rad/s is the natural frequency of the system, and $\zeta = 0.16$ is the damping ratio. The input $w(t)$ is continuous-time white noise with a variance of 0.01. Measurements of the first state are taken every 0.5 s:

$$y(t_k) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t_k) + v(t_k)$$

where $v(t_k)$ is discrete-time white noise with a variance of 10^{-4}. The initial state, estimate, and covariance are

$$\begin{aligned} x(0) &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ \hat{x}(0) &= x(0) \\ P(0) &= \begin{bmatrix} 10^{-5} & 0 \\ 0 & 10^{-2} \end{bmatrix} \end{aligned}$$

a) Discretize the system equation.

b) Implement the discrete-time Kalman filter and the RTS smoother for 10 s (20 time steps). Plot the variance of the estimation error of the first state for the forward filter and for the RTS smoother on a single plot. Do the same for the second state. Why is the second state more smoothable than the first state?

9.16 Repeat Problem 9.15 with the continuous-time process noise $w(t)$ having a variance of 1. How does this change the smoothability of the states?

9.17 Design a fixed-interval smoother for the system described in Problem 5.11 to estimate the state at each time on the basis of measurements at all 10 time steps.

a) Plot the *a posteriori* covariance of the forward state estimate and the covariance of the smoothed state estimate as a function of time for both states.

b) What are the percent improvements in the estimation-error variances due to smoothing for the two states at the initial time? Why is there so much more improvement for one state than for the other state?

c) Simulate the system and smoother a hundred times or so, each simulation with a different noise history. On the basis of your simulations, derive a numerical estimate of the smoother estimation-error variances of the two states at the initial time. How do your numerical variances compare with the theoretical variances obtained in part (b)?

CHAPTER 10

Additional topics in Kalman filtering

The use of wrong *a priori* statistics in the design of a Kalman filter can lead to large estimation errors or even to a divergence of errors.

—Raman Mehra [Meh72]

The previous chapters covered the essentials of Kalman filtering and should provide a firm foundation for further studies. This chapter discusses some additional important topics related to Kalman filtering. Section 10.1 talks about how to verify that a Kalman filter is operating reliably. When we run computer-based simulations of a Kalman filter, we can tell if the filter is working because we are in control of the simulation model and so we can compare the true state with the estimated state. However, in the real world we do not know what the true state is – after all, that is why we need a Kalman filter. In those situations, it is more difficult to verify that the Kalman filter's estimates are reliable.

Section 10.2 discusses multiple-model estimation, which is a way of estimating system states when we are not sure of which model is governing the dynamics of the system. This can be useful when the system model changes due to events of which the engineer may not be aware. Section 10.3 discusses reduced-order filtering. Many system models are of high order, which means that the corresponding Kalman filter will also be of high order. The high order of the filter may prevent the real-time implementation of the Kalman filter due to computational constraints. In these

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

cases a smaller, suboptimal filter (called a reduced-order filter) can be designed to give acceptable estimation performance at a lower computational cost. Section 10.4 discusses robust Kalman filtering, which is a way of making the filter less sensitive to variations in the assumed system model. Section 10.5 discusses the topic of delayed measurements. Sometimes the measurements do not arrive at the filter in chronological order because of processing delays. In these cases, we can modify the filter to optimally incorporate measurements that arrive at the filter in the wrong sequence.

10.1 VERIFYING KALMAN FILTER PERFORMANCE

We can verify Kalman filter performance, or adjust the gain of the Kalman filter, using our knowledge of the statistics of the innovations. The innovations is defined as $(y_k - H_k\hat{x}_k^-)$, and in this section we will show that it is a zero-mean white stochastic process with a covariance of $(H_kP_k^-H_k^T + R_k)$.

Recall our original system model, along with the one-step *a priori* update equation for the state estimate:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + w_{k-1} \\ y_k &= H_kx_k + v_k \\ \hat{x}_{k+1}^- &= F_k\hat{x}_k^- + F_kK_k(y_k - H_k\hat{x}_k^-) \end{aligned} \tag{10.1}$$

The innovations is defined as the quantity in parentheses in the update equation. The innovations can be thought of as the part of the measurement that contains new information and that is therefore used to update the state estimate (apart from our knowledge of the state transition matrix). If the innovations was zero then the state estimate would simply be updated according to the state transition matrix. A nonzero innovations allows the measurement to affect the state estimate. The innovations r_k can be written as

$$\begin{aligned} r_k &= y_k - H_k\hat{x}_k^- \\ &= (H_kx_k + v_k) - H_k\hat{x}_k^- \\ &= H_k(x_k - \hat{x}_k^-) + v_k \\ &= H_k\epsilon_k + v_k \end{aligned} \tag{10.2}$$

where ϵ_k, the *a priori* estimation error, is defined by the above equation. The covariance of the innovations is given as

$$E[r_kr_i^T] = E\left[(H_k\epsilon_k + v_k)(H_i\epsilon_i + v_i)^T\right] \tag{10.3}$$

Let us see what the covariance is when $k \neq i$. We can assume without loss of generality that $k > i$. We then obtain

$$E[r_kr_i^T] = H_kE(\epsilon_k\epsilon_i^T)H_i^T + H_kE(\epsilon_kv_i^T) \tag{10.4}$$

Note that two of the cross terms reduced to zero because of the whiteness of v_k, and the fact that the estimation error ϵ_i is independent of v_k for $k > i$. In order to evaluate this covariance, we need to evaluate $E(\epsilon_k\epsilon_i^T)$ and $E(\epsilon_kv_i^T)$. First we

will evaluate $E(\epsilon_k \epsilon_i^T)$. In order to evaluate this term, notice that the *a priori* state estimate can be written as follows:

$$\begin{aligned} \hat{x}_{k+1}^- &= F_k \hat{x}_k^- + F_k K_k (y_k - H_k \hat{x}_k^-) \\ &= F_k \hat{x}_k^- + F_k K_k (H_k x_k + v_k - H_k \hat{x}_k^-) \\ &= F_k \hat{x}_k^- + F_k K_k H_k (x_k - \hat{x}_k^-) + F_k K_k v_k \end{aligned} \tag{10.5}$$

The *a priori* estimation error can be written as

$$\begin{aligned} \epsilon_{k+1} &= x_{k+1} - \hat{x}_{k+1}^- \\ &= F_k (x_k - \hat{x}_k^-) - F_k K_k H_k (x_k - \hat{x}_k^-) + w_k - F_k K_k v_k \\ &= F_k (I - K_k H_k) \epsilon_k + (w_k - F_k K_k v_k) \\ &= \tilde{\phi}_k \epsilon_k + v_k' \end{aligned} \tag{10.6}$$

where $\tilde{\phi}_k$ and v_k' are defined by the above equation. This is a linear discrete-time system for ϵ_k with the state transition matrix

$$\tilde{\phi}_{k,i} = \begin{cases} \tilde{\phi}_{k-1} \tilde{\phi}_{k-2} \cdots \tilde{\phi}_i & k > i \\ I & k = i \end{cases} \tag{10.7}$$

ϵ_k can be solved from the initial condition ϵ_i as follows:

$$\epsilon_k = \tilde{\phi}_{k,i} \epsilon_i + \sum_{j=i}^{k-1} \tilde{\phi}_{k,j+1} v_j' \tag{10.8}$$

The covariance of $\epsilon_k \epsilon_i^T$ can be written as

$$E(\epsilon_k \epsilon_i^T) = E\left[\left(\tilde{\phi}_{k,i} \epsilon_i + \sum_{j=i}^{k-1} \tilde{\phi}_{k,j+1} v_j' \right) \epsilon_i^T \right] \tag{10.9}$$

We see that all of the $v_j' \epsilon_i^T$ terms in the above expression are zero-mean. This is because all of the v_j' noise terms occur at time i or later and so do not affect ϵ_i. [Note from Equation (10.6) that ϵ_i is affected only by the noise terms at time $(i-1)$ or earlier.] Therefore,

$$E(v_j' \epsilon_i^T) = 0 \qquad (j \geq i) \tag{10.10}$$

We therefore see that Equation (10.9) can be written as

$$\begin{aligned} E(\epsilon_k \epsilon_i^T) &= \tilde{\phi}_{k,i} E(\epsilon_i \epsilon_i^T) \\ &= \tilde{\phi}_{k,i} P_i^- \end{aligned} \tag{10.11}$$

Now that we have computed $E(\epsilon_k \epsilon_i^T)$, we need to solve for $E(\epsilon_k v_i^T)$ in order to arrive at our goal, which is the evaluation of Equation (10.4). $E(\epsilon_k v_i^T)$ can be written as

$$E(\epsilon_k v_i^T) = E\left[\left(\tilde{\phi}_{k,i} \epsilon_i + \sum_{j=i}^{k-1} \tilde{\phi}_{k,j+1} v_j' \right) v_i^T \right] \tag{10.12}$$

The $\epsilon_i v_i^T$ term in the above expression is zero-mean, and the $v_j' v_i^T$ terms are zero-mean for $j > i$. The above covariance can therefore be written as

$$\begin{aligned} E(\epsilon_k v_i^T) &= E\left(\tilde{\phi}_{k,i+1} v_i' v_i^T\right) \\ &= E\left(\tilde{\phi}_{k,i+1}(w_i - F_i K_i v_i) v_i^T\right) \\ &= -\tilde{\phi}_{k,i+1} F_i K_i R_i \end{aligned} \tag{10.13}$$

Substituting this equation, along with Equation (10.11), into Equation (10.4) gives

$$\begin{aligned} E(r_k r_i^T) &= H_k E(\epsilon_k \epsilon_i^T) H_i^T + H_k E(\epsilon_k v_i^T) \\ &= H_k \tilde{\phi}_{k,i} P_i^- H_i^T - H_k \tilde{\phi}_{k,i+1} F_i K_i R_i \\ &= H_k \tilde{\phi}_{k,i+1}(\tilde{\phi}_i P_i^- H_i^T - F_i K_i R_i) \end{aligned} \tag{10.14}$$

Now use the fact from Equation (10.6) that $\tilde{\phi}_i = F_i(I - K_i H_i)$ to obtain

$$\begin{aligned} E(r_k r_i^T) &= H_k \tilde{\phi}_{k,i+1}(F_i P_i^- H_i^T - F_i K_i H_i P_i^- H_i^T - F_i K_i R_i) \\ &= H_k \tilde{\phi}_{k,i+1}[F_i P_i^- H_i^T - F_i K_i (H_i P_i^- H_i^T + R_i)] \end{aligned} \tag{10.15}$$

Now use the fact that $K_i = P_i^- H_i^T (H_i P_i^- H_i^T + R_i)^{-1}$ (the standard Kalman filter gain equation) to obtain

$$\begin{aligned} E(r_k r_i^T) &= H_k \tilde{\phi}_{k,i+1}(F_i P_i^- H_i^T - F_i P_i^- H_i^T) \\ &= 0 \qquad \text{for } k > i \end{aligned} \tag{10.16}$$

So we see that the innovations r_k is white noise. Our next task is to determine its covariance. In order to do this we write the covariance as

$$\begin{aligned} E(r_k r_k^T) &= E[(y_k - H_k \hat{x}_k^-)(y_k - H_k \hat{x}_k^-)^T] \\ &= E\left\{[H_k(x_k - \hat{x}_k^-) + v_k][H_k(x_k - \hat{x}_k^-) + v_k]^T\right\} \\ &= H_k E(\epsilon_k \epsilon_k^T) H_k^T + E(v_k v_k^T) \\ &= H_k P_k^- H_k^T + R_k \end{aligned} \tag{10.17}$$

We therefore see that the innovations is a white noise process with zero mean and a covariance of $(H_k P_k^- H_k^T + R_k)$. While the Kalman filter is operating, we can process the innovations, compute its mean and covariance, and verify that it is white with the expected mean and covariance. If it is colored, nonzero-mean, or has the wrong covariance, then there is something wrong with the filter. The most likely reason for such a discrepancy is a modeling error. In particular, an incorrect value of F, H, Q, or R could cause the innovations to statistically deviate from its theoretically expected behavior. Statistical methods can then be used to tune F, H, Q, and R in order to force the innovations to be white zero-mean noise with a covariance of $(H_k P_k^- H_k + R_k)$ [Meh70, Meh72]. This concept is illustrated in Figure 10.1. A scalar example is presented in Problem 10.1.

Alternatively, if the engineer is uncertain of the correct values of F, H, Q, and R, then a bank of Kalman filters can be run in parallel, each Kalman filter with a value of F, H, Q, and R that the engineer thinks may be likely. Then the innovations can be inspected in each filter, and the one that matches theory is assumed to have

Figure 10.1 This figure illustrates how the performance of a Kalman filter can be used to tune the values of F, H, Q, and R in order to obtain residual statistics that agree with theory. Alternatively, the Kalman gain K could be tuned directly.

the correct F, H, Q, and R, so the state estimate that comes out of that filter is probably the most correct. See [Kob03] for an application of this idea.

The analysis of this section can also be conducted for the continuous-time Kalman filter. The continuous-time innovations, $y(t) - H(t)\hat{x}(t)$, is a zero-mean white stochastic process with a covariance $R(t)$ (see Problem 10.2).

10.2 MULTIPLE-MODEL ESTIMATION

Suppose our system model is not known, or the system model changes depending on unknown factors. We can use multiple Kalman filters (one for each possible system model) and combine the state estimates to obtain a refined state estimate. Remember Bayes' rule from Section 2.1:

$$\Pr(x|y) = \frac{\Pr(y|x)\Pr(x)}{\Pr(y)} \tag{10.18}$$

Suppose that a random variable x can take one of N mutually exclusive values $x_1, \cdots, x_N$. Then we can use Bayes' rule to write

$$\begin{aligned} \Pr(y) &= \Pr(y|x_1)\Pr(x_1) + \cdots + \Pr(y|x_N)\Pr(x_N) \\ \Pr(x|y) &= \frac{\mathrm{pdf}(y|x)\Pr(x)}{\sum_{i=1}^{N} \mathrm{pdf}(y|x_i)\Pr(x_i)} \end{aligned} \tag{10.19}$$

where we have used the fact that the probability of an event occurring is directly proportional to the value of its pdf. Now suppose that we have the time-invariant system

$$\begin{aligned} x_k &= Fx_{k-1} + Gu_{k-1} + w_{k-1} \\ y_k &= Hx_k + v_k \\ w_k &\sim N(0, Q) \\ v_k &\sim N(0, R) \end{aligned} \tag{10.20}$$

The parameter set p is defined as the set (F, G, H, Q, R). Suppose that p can take one of N possible values $p_1, \cdots, p_N$. The question that we want to answer in this section is as follows: Given the measurements y_k, what is the probability that $p = p_j$? From Equation (10.19) this probability can be written as

$$\Pr(p_j|y_k) = \frac{\mathrm{pdf}(y_k|p_j)\Pr(p_j)}{\sum_{i=1}^{N} \mathrm{pdf}(y_k|p_i)\Pr(p_i)} \tag{10.21}$$

Now think about the probability that measurement y_k is observed given the fact that $p = p_j$. If $p = p_j$ then the state will take on some value x_k that is determined by the parameter set p_j. We therefore see that

$$\begin{aligned} \Pr(y_k|p_j) &= \Pr(y_k|x_k) \\ \text{pdf}(y_k|p_j) &= \text{pdf}(y_k|x_k) \end{aligned} \tag{10.22}$$

However, if our state estimate is accurate, then we know that $x_k \approx \hat{x}_k^-$. Therefore, the above equation can be written as

$$\text{pdf}(y_k|p_j) \approx \text{pdf}(y_k|\hat{x}_k^-) \tag{10.23}$$

The right side of the equation is the pdf of the measurement y_k given the fact that the state is $\hat{x}_k^-$. But since $y_k \approx H\hat{x}_k^- + v_k$, this pdf is approximately equal to the pdf of $(y_k - H\hat{x}_k^-)$. We therefore have

$$\begin{aligned} \text{pdf}(y_k|p_j) &\approx \text{pdf}(y_k - H_k\hat{x}_k^-) \\ &= \text{pdf}(r_k) \end{aligned} \tag{10.24}$$

where r_k is the residual defined in Section 10.1. From Section 10.1 we see that if w_k, v_k, and x_0 are Gaussian, then the residual r_k is a linear combination of Gaussian random variables. Recall from Section 2.4.2 that a linear combination of Gaussian random variables is itself Gaussian. In Section 10.1 we found the mean and variance of r_k. The pdf of r_k, which is approximated by the pdf of y_k given p_j, can therefore be approximated as

$$\text{pdf}(y_k|p_j) \approx \frac{\exp(-r_k^T S_k^{-1} r_k/2)}{(2\pi)^{q/2}|S_k|^{1/2}} \tag{10.25}$$

where $r_k = y_k - H_k\hat{x}_k^-$, $S_k = H_k P_k^- H_k^T + R_k$, and q is the number of measurements.

Now from Bayes' rule we can write the following equation for the probability that $p = p_j$ given the fact that the measurement y_{k-1} is observed.

$$\Pr(p_j|y_{k-1}) = \frac{\Pr(y_{k-1}|p_j)\Pr(p_j)}{\Pr(y_{k-1})} \tag{10.26}$$

If we are presently at time k, then the measurement at time $(k-1)$ is a given. The value of the measurement at time $(k-1)$ is a certain event with a probability equal to one. Therefore, $\Pr(y_{k-1}|p_j) = \Pr(y_{k-1}) = 1$ and the above equation becomes

$$\Pr(p_j|y_{k-1}) = \Pr(p_j) \tag{10.27}$$

Now in Equation (10.21) we can substitute this equation for $\Pr(p_j)$, and we substitute Equation (10.25) for $\text{pdf}(y_k|p_j)$. This gives a time-recursive equation for evaluating the probability that $p = p_j$ given the fact that the measurement was equal to y_k. The multiple-model estimator can be summarized as follows.

The multiple-model estimator

1. For $j = 1, \cdots, N$, initialize the probabilities of each parameter set before any measurements are obtained. These probabilities are denoted as $\Pr(p_j|y_0)$ $(j = 1, \cdots, N)$.

2. At each time step k we perform the following steps.

   (a) Run N Kalman filters, one for each parameter set p_j ($j = 1, \cdots, N$). The *a priori* state estimate and covariance of the jth filter are denoted as $\hat{x}_{kj}^-$ and P_{kj}^-.

   (b) After the measurement at time k is received, for each parameter set approximate the pdf of y_k given p_j as follows:

$$\text{pdf}(y_k|p_j) \approx \frac{\exp(-r_k^T S_k^{-1} r_k/2)}{(2\pi)^{q/2}|S_k|^{1/2}} \tag{10.28}$$

   where $r_k = y_k - H_k\hat{x}_{kj}^-$, $S_k = HP_{kj}^-H^T + R_k$, and q is the number of measurements.

   (c) Estimate the probability that $p = p_j$ as follows.

$$\Pr(p_j|y_k) = \frac{\text{pdf}(y_k|p_j)\Pr(p_j|y_{k-1})}{\sum_{i=1}^N \text{pdf}(y_k|p_i)\Pr(p_i|y_{k-1})} \tag{10.29}$$

   (d) Now that each parameter set p_j has an associated probability, we can weight each $\hat{x}_{kj}^-$ and P_{kj}^- accordingly to obtain

$$\begin{aligned} \hat{x}_k^- &= \sum_{j=1}^N \Pr(p_j|y_k)\hat{x}_{kj}^- \\ P_k^- &= \sum_{j=1}^N \Pr(p_j|y_k)P_{kj}^- \end{aligned} \tag{10.30}$$

   (e) We can estimate the true parameter set in one of several ways, depending on our application. For example, we can use the parameter set with the highest conditional probability as our parameter estimate, or we can estimate the parameter set as a weighted average of the parameter sets:

$$\hat{p} = \begin{cases} \text{argmax}_{p_j} \Pr(p_j|y_k) & \text{max-probability method} \\ \sum_{j=1}^N \Pr(p_j|y_k)p_j & \text{weighted-average method} \end{cases} \tag{10.31}$$

As time progresses, some of the $\Pr(p_j|y_k)$ terms will approach zero. Those p_j possibilities can then be eliminated and the number N can be reduced.

In Equation (10.31), the function $\text{argmax}_x f(x)$ returns the value of x at which the maximum of $f(x)$ occurs. For example, $\max(1-x)^2 = 0$ because the maximum of $(1-x)^2$ is 0, but $\text{argmax}_x(1-x)^2 = 1$ because $(1-x)^2$ attains its maximum value when $x = 1$. A similar definition holds for the function argmin.

■ EXAMPLE 10.1

In this example, we consider a second-order system identification problem [Ste94]. Suppose that we have a continuous-time system with discrete-time measurements described as follows:

$$
\begin{aligned}
\dot{x} &= Ax + Bw_1 \\
&= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} x + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} w_1 \\
y_k &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + v_k \\
w_1 &\sim N(0, Q_c) \\
v_k &\sim N(0, R)
\end{aligned} \tag{10.32}
$$

The damping ratio $\zeta = 0.1$, and the process and measurement noise covariances Q_c and R are respectively equal to 1000 and $10I$. The natural frequency $\omega_n = 2$, but this is not known to the engineer. The engineer knows that ω_n^2 is either 4, 4.4, or 4.8 with the following *a priori* probabilities:

$$
\begin{aligned}
\Pr(\omega_n^2 = 4) &= 0.1 \\
\Pr(\omega_n^2 = 4.4) &= 0.6 \\
\Pr(\omega_n^2 = 4.8) &= 0.3
\end{aligned} \tag{10.33}
$$

The state equation can be written as

$$
\begin{aligned}
\dot{x} &= Ax + w \\
w &\sim N(0, BQ_cB^T)
\end{aligned} \tag{10.34}
$$

We can discretize the system using the technique given in Section 1.4. If the measurements are obtained every 0.1 seconds, then we discretize the state equation with a sample time of $T = 0.1$ to obtain

$$
\begin{aligned}
x_k &= Fx_{k-1} + \Lambda w'_{k-1} \\
y_k &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + v_k \\
F &= \exp(AT) \\
\Lambda &= (F - I)F^{-1}
\end{aligned} \tag{10.35}
$$

From Section 8.1 we know that the covariance Q' of the discrete-time noise w'_k is given as

$$
Q' \approx BQ_cB^TT \tag{10.36}
$$

This means that the discrete-time process dynamics can be written as

$$
\begin{aligned}
x_k &= Fx_{k-1} + w_{k-1} \\
w_k &\sim N(0, Q) \\
Q &= (F - I)F^{-1}(BQB^TT)F^{-T}(F^T - I)
\end{aligned} \tag{10.37}
$$

The multiple-model estimator described in this section was run on this example. Three Kalman filters running in parallel each generate an estimate of the state. As the filters run, the probability of each parameter is updated by the multiple-model Kalman filter. Figure 10.2 shows the parameter probabilities for a typical simulation run. It is seen that even though the correct parameter has the lowest initial probability, the multiple-model filter estimate converges to the correct parameter after a few seconds.

$\nabla\nabla\nabla$

Figure 10.2 Parameter probabilities for the multiple-model Kalman filter for Example 10.1. The true parameter value is 4, and the filter converges to the correct parameter after a few seconds.

10.3 REDUCED-ORDER KALMAN FILTERING

If a user wants to estimate only a subset of the state vector, then a reduced-order filter can be designed. This can be the case, for example, in a real-time application where computational effort is a main consideration. Even in off-line applications, some types of problems (e.g., weather forecasting) can involves tens of thousands of states, which naturally motivates reduced-order filtering as a means to reduce computational effort [Pha98, Bal01].

Various approaches to reduced-order filtering have been proposed over the years. For example, if the dynamic model of the underlying system can be reduced to a lower-order model that approximates the full-order model, then the reduced-order model can form the basis of a normally designed Kalman filter [Kel99]. This is the approach taken in [Gli94, Sot99] for motor state estimation, in [Sim69, Ara94] for navigation system alignment, in [Bur93, Pat98] for image processing, and in [Cha96] for audio processing. If some of the states are not observable then the Kalman filter Riccati equation reduces to a lower-order equation [Yon80]. Reduced-order filtering can be implemented by approximating the covariance with a lower-rank SVD-like decomposition [Pha98, Bal01]. If some of the measurements are noise free, or if there are known equality constraints between some of the states, then the Kalman filter is a filter with an order that is lower than the underlying system [Bry65, Hae98] as discussed in Section 7.5.1 of this book. Optimal reduced-order filters are obtained from first principles in [Ber85, Nag87]. A more heuristic approach to reduced-order filtering is to decouple portions of the matrix multiplications in the Kalman filter equations [Chu87]. In this section we will present two different approaches to reduced-order filtering.

10.3.1 Anderson's approach to reduced-order filtering

Anderson and Moore [And79] suggest a framework for reduced-order filtering that is fully developed in [Sim06a] and in this section. This approach is based on the idea that we do not always need to estimate all of the states of a system. Sometimes, with a system that has n states, we are interested only in estimating m linear combinations of the states, where $m < n$. In this case, it stands to reason that we could devise a filter with an order less than n that estimates the m linear combinations that we are interested in. Suppose our state space system is given as

$$\begin{aligned} \bar{x}_{k+1} &= \bar{F}\bar{x}_k + \bar{G}w_k \\ y_k &= \bar{H}\bar{x}_k + v_k \end{aligned} \tag{10.38}$$

We desire to estimate the following m linear combinations of the state: $T_1^T\bar{x}$, $T_2^T\bar{x}$, $\cdots$, $T_m^T\bar{x}$, where each T_i^T is a row vector. Define the $n \times n$ matrix T as

$$T = \begin{bmatrix} T_1^T \\ \vdots \\ T_m^T \\ S \end{bmatrix} \tag{10.39}$$

where S is arbitrary as long as it makes T a nonsingular $n \times n$ matrix. Now perform the state transformation

$$x = T\bar{x} \tag{10.40}$$

This means that $\bar{x} = T^{-1}x$. From these relationships we can obtain a state space description of the system in terms of the new state as follows:

$$\begin{aligned} T^{-1}x_{k+1} &= \bar{F}T^{-1}x_k + \bar{G}w_k \\ x_{k+1} &= T\bar{F}T^{-1}x_k + T\bar{G}w_k \\ &= Fx_k + Gw_k \\ y_k &= \bar{H}T^{-1}x_k + v_k \\ &= Hx_k + v_k \end{aligned} \tag{10.41}$$

where F, G, and H are defined by the above equations. Remember that our goal is to estimate the first m elements of x, which we will denote as $\tilde{x}$. We therefore partition x as follows:

$$x = \begin{bmatrix} \tilde{x} \\ \tilde{\tilde{x}} \end{bmatrix} \tag{10.42}$$

We can then write equations for $\tilde{x}_{k+1}$, $\tilde{\tilde{x}}_{k+1}$, and y_k as follows:

$$\begin{aligned} \tilde{x}_{k+1} &= F_{11}\tilde{x}_k + F_{12}\tilde{\tilde{x}}_k + G_1 w_k \\ \tilde{\tilde{x}}_{k+1} &= F_{21}\tilde{x}_k + F_{22}\tilde{\tilde{x}}_k + G_2 w_k \\ y_k &= H_1\tilde{x}_k + H_2\tilde{\tilde{x}}_k + v_k \end{aligned} \tag{10.43}$$

where the F_{ij}, G_i, and H_i matrices are appropriately dimensioned partitions of F, G, and H. Now we propose the following form for the one-step *a posteriori* estimator of $\tilde{x}$:

$$\hat{\tilde{x}}_{k+1}^+ = F_{11}\hat{\tilde{x}}_k^+ + K_k(y_{k+1} - H_1F_{11}\hat{\tilde{x}}_k^+) \tag{10.44}$$

This predictor/corrector form for the estimate of $\tilde{x}$ is very similar to the predictor/corrector form of the standard Kalman filter. The estimation error is given as follows:

$$\begin{aligned} e_{k+1} &= \tilde{x}_{k+1} - \hat{\tilde{x}}_{k+1}^{+} \\ &= F_{11}(\tilde{x}_k - \hat{\tilde{x}}_k^{+}) + F_{12}\tilde{\tilde{x}}_k + G_1 w_k - K_k(y_{k+1} - H_1 F_{11}\hat{\tilde{x}}_k^{+}) \\ &= (I - K_k H_1)F_{11}e_k + [F_{12} - K_k(H_1F_{12} - H_2F_{22})]\tilde{\tilde{x}}_k - \\ & K_k H_2 F_{21}\tilde{x}_k - K_k v_{k+1} + [G_1 - K_k(H_1G_1 + H_2G_2)]w_k \end{aligned} \tag{10.45}$$

Now we will introduce the following notation for various covariance matrices:

$$\begin{aligned} P_k &= E(e_k e_k^T) \\ \tilde{P}_k &= E(\tilde{x}_k \tilde{x}_k^T) \\ \tilde{\tilde{P}}_k &= E(\tilde{\tilde{x}}_k \tilde{\tilde{x}}_k^T) \\ \Sigma_k &= E(\tilde{x}_k \tilde{\tilde{x}}_k^T) \\ \tilde{\Pi}_k &= E(\hat{\tilde{x}}_k \tilde{x}_k^T) \\ \tilde{\tilde{\Pi}}_k &= E(\hat{\tilde{x}}_k \tilde{\tilde{x}}_k^T) \end{aligned} \tag{10.46}$$

With this notation and the equations given earlier in this section, we can obtain the following expressions for these covariances:

$$\begin{aligned} \tilde{P}_{k+1} &= F_{11}\tilde{P}_k F_{11}^T + (F_{11}\Sigma_k F_{12}^T) + (\cdots)^T + F_{12}\tilde{\tilde{P}}_k F_{12}^T + G_1 Q_k G_1^T \\ \tilde{\tilde{P}}_{k+1} &= F_{21}\tilde{P}_k F_{21}^T + (F_{21}\Sigma_k F_{22}^T) + (\cdots)^T + F_{22}\tilde{\tilde{P}}_k F_{22}^T + G_2 Q_k G_2^T \\ \Sigma_{k+1} &= F_{11}\tilde{P}_k F_{21}^T + F_{11}\Sigma_k F_{22}^T + F_{12}\tilde{\tilde{P}}_k F_{22}^T + G_1 Q_k G_2^T \\ \tilde{\Pi}_{k+1} &= (I - K_k H_1)F_{11}(\tilde{\Pi}_k F_{11}^T + \tilde{\tilde{\Pi}}_k F_{12}^T) + \\ & K_k(H_1F_{11} + H_2F_{21})(\tilde{P}_k F_{11}^T + \Sigma_k F_{12}^T) + \\ & K_k(H_1F_{12} + H_2F_{22})(\Sigma_k^T F_{11}^T + \tilde{\tilde{P}}_k F_{12}^T) + \\ & K_k(H_1G_1 + H_2G_2)Q_k G_1^T \\ \tilde{\tilde{\Pi}}_{k+1} &= (I - K_k H_1)F_{11}(\tilde{\Pi}_k F_{21}^T + \tilde{\tilde{\Pi}}_k F_{22}^T) + \\ & K_k(H_1F_{11} + H_2F_{21})(\tilde{P}_k F_{21}^T + \Sigma_k F_{22}^T) + \\ & K_k(H_1F_{12} + H_2F_{22})(\Sigma_k^T F_{21}^T + \tilde{\tilde{P}}_k F_{22}^T) + \\ & K_k(H_1G_1 + H_2G_2)Q_k G_2^T \\ P_{k+1} &= (I - K_k H_1)F_{11}P_k F_{11}^T(I - K_k H_1)^T + \\ & \left[(I - K_k H_1)F_{11}(\Sigma_k - \tilde{\tilde{\Pi}}_k)Y_{k+1}\right] + \left[\cdots\right]^T + \\ & \left[(I - K_k H_1)F_{11}(\tilde{\Pi}_k - \tilde{P}_k)F_{21}^T H_2^T K_k^T\right] + \left[\cdots\right]^T + \\ & Y_{k+1}^T \tilde{\tilde{P}}_k Y_{k+1} - \left(Y_{k+1}^T \Sigma_k^T F_{21}^T H_2^T K_k^T\right) + \left(\cdots\right)^T + \\ & K_k H_2 F_{21}\tilde{P}_k F_{21}^T H_2^T K_k^T + K_k R_{k+1} K_k^T + \\ & [G_1 - K_k(H_1G_1 + H_2G_2)]Q_k[G_1 - K_k(H_1G_1 + H_2G_2)]^T \end{aligned} \tag{10.47}$$

where Y_k is defined as

$$Y_k = [F_{12} - K_k(H_1F_{12} + H_2F_{22})]^T \tag{10.48}$$

Now we can find the optimal reduced-order gain K_k at each time step as follows:

$$\begin{aligned} K_k &= \text{argmin } \mathrm{Tr} P_{k+1} \\ \frac{\partial \mathrm{Tr} P_{k+1}}{\partial K_k} &= 0 \end{aligned} \tag{10.49}$$

In order to compute the partial derivative we have to remember from Section 1.1.3 that

$$\begin{aligned} \frac{\partial \mathrm{Tr}(ABA^T)}{\partial A} &= AB + AB^T \\ \frac{\partial \mathrm{Tr}(AB)}{\partial A} &= B^T \\ \frac{\partial \mathrm{Tr}(BA^T)}{\partial A} &= B \end{aligned} \tag{10.50}$$

Armed with these tools we can compute the partial derivative of Equation 10.49 and set it equal to zero to obtain

$$K_k = A_k^{-1} B_k \tag{10.51}$$

where A_k and B_k are given as follows:

$$\begin{aligned} A_k &= H_1 F_{11} P_k F_{11}^T H_1^T + \left[H_1 F_{11} (\Sigma_k - \tilde{\tilde{\Pi}}_k)(H_1 F_{12} + H_2 F_{22})^T \right] + \\ &\quad \left[\cdots \right]^T + \left[H_{11} F_{11} (\tilde{P}_k - \tilde{\Pi}_k) F_{21}^T H_2^T \right] + \left[\cdots \right]^T + \\ &\quad (H_1 F_{12} + H_2 F_{22}) \tilde{\tilde{P}}_k (H_1 F_{12} + H_2 F_{22})^T + \\ &\quad \left[(H_1 F_{12} + H_2 F_{22}) \Sigma_k^T F_{21}^T H_2^T \right] + \left[\cdots \right]^T + H_2 F_{21} \tilde{P}_k F_{21}^T H_2^T + \\ &\quad R_{k+1} + (H_1 G_1 + H_2 G_2) Q_k (H_2 G_1 + H_2 G_2)^T \\ B_k &= \left(F_{11} P_k + F_{12} \Sigma_k^T - F_{12} \tilde{\tilde{\Pi}}^T \right) F_{11}^T H_1^T + \\ &\quad \left(F_{11} \Sigma_k - F_{11} \tilde{\tilde{\Pi}}_k + F_{12} \tilde{\tilde{P}}_k \right) (H_1 F_{12} + H_2 F_{22})^T + \\ &\quad \left(F_{11} \tilde{P}_k - F_{11} \tilde{\Pi}_k + F_{12} \Sigma_k^T \right) F_{21}^T H_2^T + G_1 Q_k (H_1 G_1 + H_2 G_2)^T \end{aligned} \tag{10.52}$$

Equation (10.51) ends up being a long and complicated expression for the reduced-order gain. In fact, this reduced-order filter is probably more computationally expensive than the full-order filter (depending on the values of m and n). However, if the gain of the reduced-order filter converges to steady state, then it can be computed off-line to obtain savings in real-time computational cost and memory usage. However, note that the reduced-order filter may not be stable, even if the full-order Kalman filter is stable.

■ EXAMPLE 10.2

Suppose we are given the following system:

$$x_{k+1} = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.7 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0 \end{bmatrix} w_k$$

$$
\begin{aligned}
y_k &= \begin{bmatrix} 0 & 1 \end{bmatrix} x_k + v_k \\
w_k &\sim (0, 0.1) \\
v_k &\sim (0, 1)
\end{aligned} \tag{10.53}
$$

We want to find a reduced-order estimator of the first element of x. In this example the reduced-order gain of Equation (10.51) converges to a steady-state value after about 80 time steps. The estimation-error variance of the reduced-order filter converges to a value that is about 10% higher than the estimation-error variance of the full-order filter for the first state, as shown in Figure 10.3. The estimation error for the reduced-order filter and the full-order filter is shown in Figure 10.3 for a typical simulation. In this example, the standard deviation of the estimation error was 0.46 for the full-order filter and 0.50 for the reduced-order filter. The steady-state full-order estimator is given as follows:

$$
\begin{aligned}
\hat{x}_{k+1}^- &= \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.7 \end{bmatrix} \hat{x}_k^+ \\
\hat{x}_k^+ &= \hat{x}_k^- + K\left(y_k - \begin{bmatrix} 0 & 1 \end{bmatrix} \hat{x}_k^-\right) \\
K &= \begin{bmatrix} 0.1983 \\ 0.1168 \end{bmatrix}
\end{aligned} \tag{10.54}
$$

The steady-state reduced-order estimator is given as follows:

$$
\begin{aligned}
\hat{\tilde{x}}_{k+1}^+ &= 0.9\hat{\tilde{x}}_k^+ + K_r\left[y_{k+1} - (0)(0.9)\hat{\tilde{x}}_k^+\right] \\
&= 0.9\hat{\tilde{x}}_k^+ + K_r y_{k+1} \\
K_r &= 0.1420
\end{aligned} \tag{10.55}
$$

$\nabla\nabla\nabla$

10.3.2 The reduced-order Schmidt–Kalman filter

Stanley Schmidt's approach to reduced-order filtering can be used if the states are decoupled from each other in the dynamic equation [Sch66, Bro96, Gre01]. This happens, for instance, if colored measurement noise is accounted for by augmenting the state vector (see Section 7.2.2). In fact, satellite navigation with colored measurement noise was the original motivation for this approach.

Suppose we have a system in the form

$$
\begin{aligned}
\begin{bmatrix} \tilde{x}_{k+1} \\ \tilde{\tilde{x}}_{k+1} \end{bmatrix} &= \begin{bmatrix} F_1 & 0 \\ 0 & F_2 \end{bmatrix} \begin{bmatrix} \tilde{x}_k \\ \tilde{\tilde{x}}_k \end{bmatrix} + \begin{bmatrix} \tilde{w}_k \\ \tilde{\tilde{w}}_k \end{bmatrix} \\
\tilde{w}_k &\sim (0, Q_1) \\
\tilde{\tilde{w}}_k &\sim (0, Q_2) \\
y_k &= \begin{bmatrix} H_1 & H_2 \end{bmatrix} \begin{bmatrix} \tilde{x}_k \\ \tilde{\tilde{x}}_k \end{bmatrix} + v_k \\
v_k &\sim (0, R)
\end{aligned} \tag{10.56}
$$

We want to estimate $\tilde{x}_k$ but we do not care about estimating $\tilde{\tilde{x}}_k$. Suppose we use a Kalman filter to estimate the entire state vector. The estimation-error covariance

Figure 10.3 Results for Example 10.2. The top figure shows the analytical estimation-error variances for the first state for the full-order filter and the reduced-order filter. As expected, the reduced-order filter has a higher estimation-error variance, but the small degradation in performance may be worth the computational savings, depending on the application. The bottom figure shows typical error magnitudes for the estimate of the first state for the full-order filter and the reduced-order filter. The reduced-order filter has slightly larger estimation errors.

can be partitioned as follows:

$$P = \begin{bmatrix} \tilde{P} & \Sigma \\ \Sigma^T & \tilde{\tilde{P}} \end{bmatrix} \tag{10.57}$$

We are omitting the time subscripts for ease of notation. The Kalman gain is usually written as $K = P^- H^T (HP^- H^T + R)^{-1}$. With our new notation it can be written as follows:

$$\begin{aligned} K &= \begin{bmatrix} \tilde{K} \\ \tilde{\tilde{K}} \end{bmatrix} \\ &= \begin{bmatrix} \tilde{P}^- & \Sigma^- \\ (\Sigma^-)^T & \tilde{\tilde{P}}^- \end{bmatrix} \times \\ &\begin{bmatrix} H_1^T \\ H_2^T \end{bmatrix} \left[\begin{pmatrix} H_1 & H_2 \end{pmatrix} \begin{pmatrix} \tilde{P}^- & \Sigma^- \\ (\Sigma^-)^T & \tilde{\tilde{P}}^- \end{pmatrix} \begin{pmatrix} H_1^T \\ H_2^T \end{pmatrix} + R \right]^{-1} \end{aligned} \tag{10.58}$$

By multiplying out this equation we can write the formula for $\tilde{K}$ as follows.

$$\tilde{K} = (\tilde{P}^- H_1^T + \Sigma^- H_2^T)\alpha^{-1} \tag{10.59}$$

where α is defined as

$$\alpha = H_1 \tilde{P}^- H_1^T + H_1 \Sigma^- H_2^T + H_2 (\Sigma^-)^T H_1^T + H_2 \tilde{\tilde{P}}^- H_2^T + R \tag{10.60}$$

The measurement-update equation for $\hat{x}$ is normally written as $\hat{x}_k^+ = \hat{x}_k^- + K(y_k - H\hat{x}_k^-)$. With our new notation it is written as

$$\begin{bmatrix} \hat{\tilde{x}}_k^+ \\ \hat{\tilde{\tilde{x}}}_k^+ \end{bmatrix} = \begin{bmatrix} \tilde{K} \\ \tilde{\tilde{K}} \end{bmatrix} \left(y_k - H_1 \hat{\tilde{x}}_k^- - H_2 \hat{\tilde{\tilde{x}}}_k^- \right) \tag{10.61}$$

Since we are not going to estimate $\tilde{\tilde{x}}$ with the reduced-order filter, we set $\hat{\tilde{\tilde{x}}}_k^- = 0$ in the above equation to obtain the following measurement-update equation for $\hat{\tilde{x}}_k^+$:

$$\hat{\tilde{x}}_k^+ = \hat{\tilde{x}}_k^- + \tilde{K} \left(y_k - H_1 \hat{\tilde{x}}_k^- \right) \tag{10.62}$$

The measurement-update equation for P is usually written as $P^+ = (I-KH)P^-(I-KH)^T + KRK^T$. With our new notation it is written as

$$\begin{aligned} \begin{bmatrix} \tilde{P}^+ & \Sigma^+ \\ (\Sigma^+)^T & \tilde{\tilde{P}}^+ \end{bmatrix} &= \left[\begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} - \begin{pmatrix} \tilde{K} \\ \tilde{\tilde{K}} \end{pmatrix} \begin{pmatrix} H_1 & H_2 \end{pmatrix} \right] \begin{bmatrix} \tilde{P}^- & \Sigma^- \\ (\Sigma^-)^T & \tilde{\tilde{P}}^- \end{bmatrix} \times \\ & \left[\begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} - \begin{pmatrix} \tilde{K} \\ \tilde{\tilde{K}} \end{pmatrix} \begin{pmatrix} H_1 & H_2 \end{pmatrix} \right]^T + \begin{pmatrix} \tilde{K} \\ \tilde{\tilde{K}} \end{pmatrix} R \begin{pmatrix} \tilde{K}^T & \tilde{\tilde{K}}^T \end{pmatrix} \end{aligned} \tag{10.63}$$

At this point, we assume that $\tilde{\tilde{K}} = 0$. This can be justified if the measurement noise associated with the $\tilde{\tilde{x}}$ states is large, or if H_2 is small, or if the elements of $\tilde{\tilde{x}}$ are small. The $\tilde{\tilde{x}}$ elements are then referred to as consider states, nuisance states, or nuisance variables, because they are only partially used in the reduced-order state estimator, and because we are not interested in estimating them. Based on Equation (10.63), the update equation for $\tilde{P}^+$ can then be written as

$$\begin{aligned} \tilde{P}^+ &= (I - \tilde{K}H_1)\tilde{P}^-(I - \tilde{K}H_1)^T - \tilde{K}H_2(\Sigma^-)^T(I - \tilde{K}H_1)^T - \\ & \quad (I - \tilde{K}H_1)\Sigma^- H_2^T \tilde{K}^T + \tilde{K}H_2\tilde{\tilde{P}}^- H_2^T \tilde{K}^T + \tilde{K}R\tilde{K}^T \end{aligned} \tag{10.64}$$

Multiplying out the above equation and then using the definition of α from Equation (10.60) results in

$$\begin{aligned} \tilde{P}^+ &= \tilde{P}^- - \tilde{K}H_1\tilde{P}^- - \tilde{P}^- H_1^T \tilde{K}^T + \tilde{K}\alpha\tilde{K}^T - \tilde{K}H_2(\Sigma^-)^T - \Sigma^- H_2^T \tilde{K}^T \\ &= \tilde{P}^- - \tilde{K}H_1\tilde{P}^- - \tilde{P}^- H_1^T \tilde{K}^T + (\tilde{P}^- H_1^T + \Sigma^- H_2^T)\tilde{K}^T - \tilde{K}H_2(\Sigma^-)^T - \\ & \quad \Sigma^- H_2^T \tilde{K}^T \\ &= (I - \tilde{K}H_1)\tilde{P}^- - \tilde{K}H_2(\Sigma^-)^T \end{aligned} \tag{10.65}$$

This gives the measurement-update equation for $\tilde{P}^+$. We can go through similar manipulations with Equation (10.63) to obtain

$$\begin{aligned} \Sigma^+ &= (I - \tilde{K}H_1)\Sigma^- - \tilde{K}H_2\tilde{\tilde{P}}^- \\ \tilde{\tilde{P}}^+ &= \tilde{\tilde{P}}^- \end{aligned} \tag{10.66}$$

Putting it all together results in the reduced-order Schmidt–Kalman filter. We can summarize the reduced-order filter as follows.

The reduced-order Schmidt–Kalman filter

1. The system and measurement equations are given in Equation (10.56).
2. At each time step execute the following equations to obtain $\hat{\tilde{x}}_k$, the estimate of the desired part of the state:

$$
\begin{aligned}
\tilde{K}_k &= (\tilde{P}_k^- H_1^T + \Sigma_k^- H_2^T)\alpha_k^{-1} \\
\alpha_k &= H_1 \tilde{P}_k^- H_1^T + H_1 \Sigma_k^- H_2^T + H_2 (\Sigma_k^-)^T H_1^T + H_2 \tilde{\tilde{P}}_k^- H_2^T + R \\
\hat{\tilde{x}}_k^+ &= \hat{\tilde{x}}_k^- + \tilde{K}_k (y_k - H_1 \hat{\tilde{x}}_k^-) \\
\tilde{P}_k^+ &= (I - \tilde{K}_k H_1)\tilde{P}_k^- - \tilde{K}_k H_2 (\Sigma_k^-)^T \\
\Sigma_k^+ &= (I - \tilde{K}_k H_1)\Sigma_k^- - \tilde{K}_k H_2 \tilde{\tilde{P}}_k^- \\
\tilde{\tilde{P}}_k^+ &= \tilde{\tilde{P}}_k^- \\
\hat{\tilde{x}}_{k+1}^- &= F_1 \hat{\tilde{x}}_k^+ \\
\tilde{P}_{k+1}^- &= F_1 \tilde{P}_k^+ F_1^T + Q_1 \\
\Sigma_{k+1}^- &= F_1 \Sigma_k^+ F_2^T \\
\tilde{\tilde{P}}_{k+1}^- &= F_1 \tilde{\tilde{P}}_k^+ F_2^T + Q_2
\end{aligned}
\tag{10.67}
$$

■ **EXAMPLE 10.3**

Consider the following system:

$$
\begin{aligned}
x_{k+1} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + w_k \\
y_k &= \begin{bmatrix} 1 & 1 \end{bmatrix} x_k + v_k \\
w_k &\sim (0, Q) \qquad Q = \mathrm{diag}(1, 0) \\
v_k &\sim (0, R) \qquad R = 1
\end{aligned}
\tag{10.68}
$$

Figure 10.4 shows a typical example of the estimation error of the first element of the state vector for the full-order filter and the reduced-order filter. It is seen that the performances of the two estimators are virtually identical. In other words, we can save a lot of computational effort with only a marginal degradation of estimation performance by using the reduced-order filter.

∇∇∇

10.4 ROBUST KALMAN FILTERING

The Kalman filter works well, but it assumes that the system model and noise statistics are known. If any of these assumptions are violated then the filter estimates can degrade. This was noted early in the history of Kalman filtering [Soo65, Hef66, Nis66].

Daniel Pena and Irwin Guttman give an overview of several methods of robustifying the Kalman filter [Spa88, Chapter 9]. For example, although the Kalman filter

Figure 10.4 Results for Example 10.3. Typical error magnitudes for the estimate of the first state for the full-order filter and the reduced-order filter. The reduced-order filter has only slightly larger estimation errors.

is the optimal linear filter, it is not the optimal filter in general for non-Gaussian noise. Noise in nature is often approximately Gaussian but with heavier tails, and the Kalman filter can be modified to accommodate these types of density functions [Mas75, Mas77, Tsa83]. Sometimes, measurements do not contain any useful information but consist entirely of noise (probabilistically), and the Kalman filter can be modified to deal with this possibility also [Nah69, Sin73, Ath77, Bar78].

The problem of Kalman filtering with uncertainties in the system matrix F_k, the measurement matrix H_k, and the noise covariances Q_k and R_k, has been considered by several authors [Xie94, Zha95, Hsi96, The96, Xie04]. This can be called adaptive filtering or robust filtering. Comparisons of adaptive filtering methods for navigation are presented in [Hid03]. Continuous-time adaptive filtering is discussed in [Bar05, Mar05]. Methods for identifying the noise covariances Q and R are presented in [Meh70, Meh72, Als74, Mye76]. Additional material on robust Kalman filtering can be found in [Che93].

In this section we present a conceptually straightforward way of making the Kalman filter more robust to uncertainties in Q and R [Kos04]. Suppose we have the linear time-invariant system

$$\begin{aligned} x_{k+1} &= Fx_k + w_k \\ y_k &= Hx_k + v_k \\ w_k &\sim (0, Q) \\ v_k &\sim (0, R) \end{aligned} \tag{10.69}$$

Now suppose that a general steady-state gain K (not necessarily the Kalman gain) is used in a predictor/corrector type of state estimator. The state estimate update equations are then given as follows:

$$\begin{aligned}
\hat{x}_{k+1}^{-} &= F\hat{x}_k^{+} \\
\hat{x}_{k+1}^{+} &= \hat{x}_{k+1}^{-} + K(y_{k+1} - H\hat{x}_{k+1}^{-}) \\
&= F\hat{x}_k^{+} + K(Hx_{k+1} + v_{k+1} - HF\hat{x}_k^{+}) \\
&= KHx_{k+1} + (I - KH)F\hat{x}_k^{+} + Kv_{k+1} \\
&= (KHFx_k + KHw_k) + (I - KH)F\hat{x}_k^{+} + Kv_{k+1} \qquad (10.70)
\end{aligned}$$

The error in the *a posteriori* state estimate can be written as

$$\begin{aligned}
e_{k+1} &= x_{k+1} - \hat{x}_{k+1}^{+} \\
&= (Fx_k + w_k) - [(KHFx_k + KHw_k) + (I - KH)F\hat{x}_k^{+} + Kv_{k+1}] \\
&= (I - KH)Fx_k + (I - KH)w_k - (I - KH)F\hat{x}_k^{+} - Kv_{k+1} \\
&= (I - KH)Fe_k + (I - KH)w_k - Kv_{k+1} \qquad (10.71)
\end{aligned}$$

So the covariance of the estimation error can be written as

$$\begin{aligned}
P_{k+1} &= E(e_{k+1}e_{k+1}^T) \\
&= (I - KH)FP_kF^T(I - KH)^T + (I - KH)Q(I - KH)^T + \\
&\quad KRK^T \qquad (10.72)
\end{aligned}$$

The steady-state covariance P satisfies the following Riccati equation:

$$P = (I - KH)FPF^T(I - KH)^T + (I - KH)Q(I - KH)^T + KRK^T \qquad (10.73)$$

Note that we derived this without making any assumption on the optimality of the filter gain K. That is, this equation holds regardless of what filter gain K we use. Now we can consider what happens when there is no measurement noise, and what happens when there is no process noise. Define P_1 as the steady-state estimation-error covariance when $R = 0$, and P_2 as the steady-state estimation-error covariance when $Q = 0$. The above equation for P shows that

$$\begin{aligned}
P_1 &= (I - KH)FP_1F^T(I - KH)^T + (I - KH)Q(I - KH)^T \\
P_2 &= (I - KH)FP_2F^T(I - KH)^T + KRK^T \qquad (10.74)
\end{aligned}$$

Adding these two covariances together results in

$$\begin{aligned}
P_1 + P_2 &= (I - KH)FP_1F^T(I - KH)^T + (I - KH)Q(I - KH)^T + \\
&\quad (I - KH)FP_2F^T(I - KH)^T + KRK^T \\
&= (I - KH)F(P_1 + P_2)F^T(I - KH)^T + \\
&\quad (I - KH)Q(I - KH)^T + KRK^T \qquad (10.75)
\end{aligned}$$

Comparing this equation with Equation (10.73) shows that P and the sum (P_1+P_2) both satisfy the same Riccati equation. This shows that

$$P = P_1 + P_2 \qquad (10.76)$$

This shows an interesting linearity property of a general predictor/corrector type of state estimator. The estimation covariance is equal to the sum of the covariance

due to process noise only and the covariance due to measurement noise only. Recall from Chapter 5 that the Kalman filter was designed to minimize the trace of P. So the Kalman filter minimizes the trace of $(P_1 + P_2)$.

Now suppose that the true process noise and measurement noise covariances are different from those assumed by the Kalman filter. The filter is designed under the assumption that the noise covariances are Q and R, but the true noise covariances are $\tilde{Q}$ and $\tilde{R}$:

$$\begin{aligned} \tilde{Q} &= (1+\alpha)Q \\ \tilde{R} &= (1+\beta)R \end{aligned} \tag{10.77}$$

where α and β are unknown scalars. These differences between the assumed and true covariances will result in a change in the estimation-error covariance of the filter. The true estimation-error covariance $\tilde{P}$ will be equal to the assumed covariance P plus some difference ΔP. This can be written as

$$\begin{aligned} \tilde{P} &= (I-KH)F\tilde{P}F^T(I-KH)^T + (I-KH)\tilde{Q}(I-KH)^T + K\tilde{R}K^T \\ P+\Delta P &= (I-KH)F(P+\Delta P)F^T(I-KH)^T + \\ &\quad (1+\alpha)(I-KH)Q(I-KH)^T + (1+\beta)KRK^T \end{aligned} \tag{10.78}$$

Comparing this equation with Equation (10.73) shows that

$$\Delta P = (I-KH)F\Delta PF^T(I-KH)^T + \alpha(I-KH)Q(I-KH)^T + \beta KRK^T \tag{10.79}$$

Now we repeat this same line of reasoning for the computation of the true estimation-error covariance when the process noise is zero ($\tilde{P}_1 = P_1 + \Delta P_1$) and the true estimation-error covariance when the measurement noise is zero ($\tilde{P}_2 = P_2 + \Delta P_2$). Equation (10.74) shows that

$$\begin{aligned} \tilde{P}_1 &= (I-KH)F\tilde{P}_1F^T(I-KH)^T + (I-KH)\tilde{Q}(I-KH)^T \\ P_1+\Delta P_1 &= (I-KH)F(P_1+\Delta P_1)F^T(I-KH)^T + \\ &\quad (1+\alpha)(I-KH)Q(I-KH)^T \\ \tilde{P}_2 &= (I-KH)F\tilde{P}_2F^T(I-KH)^T + K\tilde{R}K^T \\ P_2+\Delta P_2 &= (I-KH)F(P_2+\Delta P_2)F^T(I-KH)^T + (1+\beta)KRK^T \end{aligned} \tag{10.80}$$

Comparing these equations with Equation (10.74) shows that

$$\begin{aligned} \Delta P_1 &= (I-KH)F\Delta P_1F^T(I-KH)^T + \alpha(I-KH)Q(I-KH)^T \\ \Delta P_2 &= (I-KH)F\Delta P_2F^T(I-KH)^T + \beta KRK^T \end{aligned} \tag{10.81}$$

Adding these two equations and comparing with Equation (10.79 shows that

$$\Delta P = \Delta P_1 + \Delta P_2 \tag{10.82}$$

Comparing Equations (10.74) and (10.81) shows that

$$\begin{aligned} \Delta P_1 &= \alpha P_1 \\ \Delta P_2 &= \beta P_2 \end{aligned} \tag{10.83}$$

Combining Equations (10.82) and (10.83) shows that

$$\Delta P = \alpha P_1 + \beta P_2 \tag{10.84}$$

Now suppose that α and β are independent zero-mean random variables with variances σ_1^2 and σ_2^2, respectively. The previous equation shows that

$$\begin{aligned} E[\mathrm{Tr}(\Delta P)] &= E(\alpha)\mathrm{Tr}(P_1) + E(\beta)\mathrm{Tr}(P_2) \\ &= 0 \\ E\left\{[\mathrm{Tr}(\Delta P)]^2\right\} &= E\left\{[\alpha\mathrm{Tr}(X_1) + \beta\mathrm{Tr}(X_2)]^2\right\} \\ &= \sigma_1^2\mathrm{Tr}^2(P_1) + \sigma_2^2\mathrm{Tr}^2(P_2) \end{aligned} \tag{10.85}$$

This gives the variance of the change in the estimation-error covariance due to changes in the process and measurement-noise covariances. A robust filter should try to minimize this variance. In other words, a robust filter should have an estimation-error covariance that is insensitive to changes in the process and measurement-noise covariances. So the performance index of a robust filter can be written as follows:

$$\begin{aligned} J &= \rho\mathrm{Tr}(P) + (1-\rho)E\left\{[\mathrm{Tr}(\Delta P)]^2\right\} \\ &= \rho[\mathrm{Tr}(P_1) + \mathrm{Tr}(P_2)] + (1-\rho)\left[\sigma_1^2\mathrm{Tr}^2(P_1) + \sigma_2^2\mathrm{Tr}^2(P_2)\right] \end{aligned} \tag{10.86}$$

where ρ is the relative importance given to filter performance under nominal conditions (i.e., when Q and R are as expected), and $(1-\rho)$ is the relative importance given to robustness. In other words, $(1-\rho)$ is the relative weight given to minimizing the variation of the estimation-error covariance due to changes in Q and R. If $\rho = 1$ then we have the standard Kalman filter. If $\rho = 0$ then we will minimize changes in the estimation-error covariance, but the nominal estimation-error covariance may be poor. So ρ should be chosen to balance nominal performance and robustness. Unfortunately, the performance index J cannot be minimized analytically, so numerical methods must be used. P_1 and P_2 are functions of the gain K and can be computed using a DARE function in control system software. The partial derivative of J with respect to K must be computed numerically, and then the value of K can be changed using a gradient-descent method to decrease J.

■ EXAMPLE 10.4

Suppose we have a discretized second-order Newtonian system that is driven by an acceleration input. $x(1)$ represents position, $x(2)$ represents velocity, u_k represents the known acceleration input, and w_k represents the noisy acceleration input. This is the same as the system described in Example 9.1. The system is described as follows:

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} u_k + w_k \\ y_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k \\ w_k &\sim (0, Q) \\ v_k &\sim (0, R) \\ Q &= q^2 \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T/2 \end{bmatrix} \end{aligned} \tag{10.87}$$

The sample time $T = 0.1$. The variance q^2 of the acceleration noise is equal to 0.2^2, and the variance R of the measurement noise is equal to 10^2. Now

suppose that Q and R have relative uncertainties of one (one standard deviation). That is, $\sigma_1^2 = \sigma_2^2 = 1$. Suppose we find the robust filter gain using equal weighting for both nominal and robust performance (i.e., $\rho = 0.5$). Table 10.1 shows the average performance of the robust filter and the standard Kalman filter when Q and R change by factors of -0.8 and 3, respectively. One question that remains is, How does the robust filter perform under nominal conditions? That is, since the Kalman filter is optimal, the robust filter will not perform as well as the Kalman filter when Q and R are equal to their nominal values. However, Table 10.2 shows that the performance degradation is marginal. In fact, the robust filter performs identically to the optimal filter (to two decimal places) under nominal conditions. During the gradient-descent optimization of Equation (10.86), the nominal part of the cost function increases from 2.02 to 2.04, the robust part of the cost function decreases from 2.54 to 2.38, and the total cost function decreases from 2.28 to 2.21.

Table 10.1 RMS estimation errors for Example 10.4 over 100 seconds when the noise covariances are not nominal ($\rho = 0.5$, $\sigma_1 = \sigma_2 = 1$, $\alpha = -0.8$, $\beta = 3$)

	Position	Velocity
Standard Filter	4.62	0.38
Robust Filter	4.47	0.32

Table 10.2 RMS estimation errors for Example 10.4 over 100 seconds when the noise covariances are nominal ($\rho = 0.5$, $\sigma_1 = \sigma_2 = 1$, $\alpha = 0$, $\beta = 0$)

	Position	Velocity
Standard Filter	1.38	0.19
Robust Filter	1.38	0.19

$\nabla\nabla\nabla$

The robust filtering approach presented here opens several possible research topics. For example, under what conditions is the robust filter stable? Is the gain of the robust filter equal to the gain of a standard Kalman filter for some other related system? What is the true estimation-error covariance of the robust filter?

10.5 DELAYED MEASUREMENTS AND SYNCHRONIZATION ERRORS

In decentralized filtering systems, observations are often collected at various physical locations, and then transmitted in bulk to a central processing computer. In this type of setup, the measurements may not arrive at the processing computer synchronously. That is, the computer may receive measurements out of sequence. This is typically the case in target-tracking systems. Various approaches have been

taken to deal with this problem [Ale91, Bar95, Kas96, Lar98, Mal01]. The case of delayed measurements with uncertainty in the measurement sampling time is discussed in [Tho94a, Tho94b]. The approach to filtering delayed measurements that is presented here is based on [Bar02].

First we will present yet another form of the Kalman filter that will provide the basis for the delayed-measurement filter. Then we will derive the optimal way to incorporate delayed measurements into the Kalman filter estimate and covariance. In this section, we will have to change our notation slightly in order to carry out the derivation of the delayed measurement Kalman filter. We will use the following notation to represent a discrete-time system:

$$\begin{aligned} x(k) &= F(k-1)x(k-1) + w(k-1) \\ y(k) &= H(k)x(k) + v(k) \end{aligned} \tag{10.88}$$

where $w(k)$ and $v(k)$ are independent zero-mean white noise process with covariances $Q(k)$ and $R(k)$, respectively.

10.5.1 A statistical derivation of the Kalman filter

Suppose that we have an *a priori* estimate $\hat{x}^-(k)$ at time k, and we want to find an optimal way to update the state estimate based on the measurement at time k. We want our update equation to be linear (for reasons of mathematical tractability) so we decide to update the state estimate at time k with the equation

$$\hat{x}^+(k) = K(k)y(k) + b(k) \tag{10.89}$$

where $K(k)$ and $b(k)$ are a matrix and vector to be determined. Our first state estimation criterion is unbiasedness. We can see by taking the mean of Equation (10.89) that

$$\overline{\hat{x}^+}(k) = K(k)\bar{y}(k) + b(k) \tag{10.90}$$

This gives us the constraint that

$$b(k) = \bar{x}(k) - K(k)\bar{y}(k) \tag{10.91}$$

This will ensure that $\hat{x}^+(k)$ is unbiased regardless of the value of the gain matrix $K(k)$. Next we find the gain matrix $K(k)$ that minimizes the trace of the estimation error. First recall that

$$\begin{aligned} P_z &= E[(z-\bar{z})(z-\bar{z})^T] \\ &= E(zz^T) - \bar{z}\bar{z}^T \end{aligned} \tag{10.92}$$

for any general random vector z. Now set $z = x(k) - \hat{x}^+(k)$. With this definition of z we see that $\bar{z} = 0$. The quantity we want to minimize is given by the trace of the following matrix:

$$\begin{aligned} P^+(k) &= E[(x(k) - \hat{x}^+(k))(x(k) - \hat{x}^+(k))^T] \\ &= P_z + \bar{z}\bar{z}^T \end{aligned} \tag{10.93}$$

P_z can be computed as follows:

$$
\begin{aligned}
P_z &= E\left\{[x(k) - \hat{x}^+(k) - E(x(k) - \hat{x}^+(k))][\cdots]^T\right\} \\
&= E\left\{[x(k) - (K(k)y(k) + b(k)) - \bar{x}(k) - (K(k)\bar{y}(k) + b(k))][\cdots]^T\right\} \\
&= E\left\{[(x(k) - \bar{x}(k)) - K(k)(y(k) - \bar{y}(k))][\cdots]^T\right\} \\
&= P^-(k) - K(k)P_{yx} - P_{xy}K^T(k) + K(k)P_yK^T(k) \qquad (10.94)
\end{aligned}
$$

We are using the symbol P_{yx} to denote the cross covariance between y_k and x_k, P_{xy} to denote the cross covariance between x_k and y_k, and P_y to denote the covariance of y_k. Recall that $P_{xy} = P_{yx}^T$. We have omitted the subscript k on P_{yx}, P_{xy}, and P_y for notational convenience. We combine the above equation with (10.93) to obtain

$$
\begin{aligned}
\mathrm{Tr}\, P^+(k) &= \mathrm{Tr}\left(P^-(k) - K(k)P_{yx} - P_{xy}K(k)^T + K(k)P_yK(k)^T\right) + \mathrm{Tr}(\bar{z}\bar{z}^T) \\
&= \mathrm{Tr}\left(P^-(k) - K(k)P_{yx} - P_{xy}K(k)^T + K(k)P_yK(k)^T\right) + \\
&\quad ||\bar{x}(k) - K(k)\bar{y}(k) - b(k)||^2 \\
&= \mathrm{Tr}\left[(K(k) - P_{xy}P_y^{-1})P_y(K(k) - P_{xy}P_y^{-1})^T\right] + \\
&\quad \mathrm{Tr}\left(P^-(k) - P_{xy}P_y^{-1}P_{xy}^T\right) + ||\bar{x}(k) - K(k)\bar{y}(k) - b(k)||^2 \qquad (10.95)
\end{aligned}
$$

where we have used the fact that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ for compatibly dimensioned matrices [see Equation (1.26)]. We want to choose $K(k)$ and $b(k)$ in order to minimize the above expression. The second term is independent of $K(k)$ and $b(k)$, and the first and third terms are always nonnegative. The first and third terms can be minimized to zero when

$$
\begin{aligned}
K(k) &= P_{xy}P_y^{-1} \\
b(k) &= \bar{x}(k) - K(k)\bar{y}(k) \qquad (10.96)
\end{aligned}
$$

Note that this is the same value for $b(k)$ that we obtained in Equation (10.91) when we enforced unbiasedness in the state estimate. With these values of $K(k)$ and $b(k)$, we see that the first and third terms in (10.95) are equal to zero, so the estimation-error covariance $P^+(k)$ can be seen to be equal to the second term. Substituting these values into Equation (10.89) we obtain

$$
\begin{aligned}
\hat{x}^+(k) &= K(k)y(k) + \bar{x}(k) - K(k)\bar{y}(k) \\
&= K(k)y(k) + \hat{x}^-(k) - K(k)H(k)\hat{x}^-(k) \\
&= \hat{x}^-(k) + K(k)(y(k) - H(k)\hat{x}^-(k)) \\
P^+(k) &= P^-(k) - P_{xy}P_y^{-1}P_{xy}^T \\
&= P^-(k) - K(k)P_yK^T(k) \qquad (10.97)
\end{aligned}
$$

Straightforward calculations (see Problem 10.8) show that P_{xy} and P_y can be computed as

$$
\begin{aligned}
P_{xy} &= P^-(k)H(k)^T \\
P_y &= H(k)P^-(k)H(k)^T + R(k) \qquad (10.98)
\end{aligned}
$$

Now consider our linear discrete-time system:

$$
\begin{aligned}
x(k) &= F(k-1)x(k-1) + w(k-1) \\
y(k) &= H(k)x(k) + v(k) \qquad (10.99)
\end{aligned}
$$

The noise processes $w(k)$ and $v(k)$ are white, zero-mean, and uncorrelated, with covariances $Q(k)$ and $R(k)$, respectively. We saw in Chapter 4 how the mean and covariance of the state propagates between measurement times. Those equations, along with the measurement-update equations derived above, provide the following Kalman filter equations:

$$\begin{aligned}
\hat{x}^-(k) &= F(k-1)\hat{x}^+(k-1) \\
P^-(k) &= F(k-1)P^+(k-1)F^T(k-1) + Q(k) \\
P_{xy} &= P^-(k)H^T(k) \\
P_y &= H(k)P^-(k)H^T(k) + R(k) \\
K(k) &= P_{xy}P_y^{-1} \\
\hat{x}^+(k) &= \hat{x}^-(k) + K(k)(y(k) - H(k)\hat{x}^-(k)) \\
P^+(k) &= P^-(k) - K(k)P_yK^T(k) \\
&= P^-(k) - P_{xy}P_y^{-1}P_{xy}^T
\end{aligned} \tag{10.100}$$

These equations appear much different than the Kalman filter equations derived earlier in this book, but actually they are mathematically identical for linear systems.

10.5.2 Kalman filtering with delayed measurements

Now we need to complicate the notation a little bit more in order to derive the Kalman filter with delayed measurements. We will write our system equations as

$$\begin{aligned}
x(k) &= F(k,k-1)x(k-1) + w(k,k-1) \\
y(k) &= H(k)x(k) + v(k)
\end{aligned} \tag{10.101}$$

$F(k,k-1)$ is the matrix that quantifies the state transition from time $(k-1)$ to time k. Similarly, $w(k,k-1)$ is the effect of the process noise on the state from time $(k-1)$ to time k. We can then generalize the state-space equation to the following:

$$x(k) = F(k,k_0)x(k_0) + w(k,k_0) \tag{10.102}$$

where k_0 is any time index less than k. The above equation can be solved for $x(k_0)$ as

$$x(k_0) = F(k_0,k)[x(k) - w(k,k_0)] \tag{10.103}$$

where $F(k_0,k) = F^{-1}(k,k_0)$. Note that $F(k,k_0)$ should always be invertible if it comes from a real system, because $F(k,k_0)$ comes from a matrix exponential that is always invertible (see Sections 1.2 and 1.4). The noise $w(k,k_0)$ is the cumulative effect of all of the process noise on the state from time k_0 to time k. Its covariance is defined as $Q(k,k_0)$:

$$w(k,k_0) \sim [0, Q(k,k_0)] \tag{10.104}$$

At time k we have the standard *a posteriori* Kalman filter estimate, which is the expected value of the state $x(k)$ conditioned on all of the measurements up to and including time k. We also have the *a posteriori* covariance of the estimate:

$$\begin{aligned}
\hat{x}(k) &= E[x(k)|y(1),\cdots,y(k)] \\
&= E[x(k)|Y(k)] \\
P(k) &= E\left\{[x(k)-\hat{x}(k)][x(k)-\hat{x}(k)]^T|Y(k)\right\}
\end{aligned} \tag{10.105}$$

where $Y(k)$ is defined by the above equation; that is, $Y(k)$ is all of the measurements up to and including time k that have been processed by the Kalman filter. (There may be some measurements before time k that have not yet been processed by the filter. These measurements are not part of $Y(k)$.)

Now suppose an out-of-sequence measurement arrives. That is, we obtain a measurement from time $k_0 < k$ that we want to incorporate into the estimate and covariance at time k. The problem is how to modify the state estimate and covariance on the basis of this new measurement. The modified state estimate and covariance are given as follows:

$$\begin{aligned} \hat{x}(k|k_0) &= E[x(k)|Y(k), y(k_0)] \\ P(k|k_0) &= E\left\{[x(k) - \hat{x}(k,k_0)][x(k) - \hat{x}(k,k_0)]^T|Y(k), y(k_0)\right\} \end{aligned} \quad (10.106)$$

The approach here is to use the new measurement at time k_0 to obtain an updated state estimate and covariance at time k_0, and then use those quantities to update the estimate and covariance at time k. We can use Equation (10.103) to obtain

$$\begin{aligned} E[x(k_0)|Y(k)] &= F(k_0,k)E[x(k) - w(k,k_0)|Y(k)] \\ &= F(k_0,k)[\hat{x}^-(k) - \hat{w}(k,k_0)] \end{aligned} \quad (10.107)$$

where $\hat{w}(k,k_0)$ is defined by the above equation; it is the expected value of the cumulative effect of the process noise from time k_0 to time k, conditioned on all of the measurements up to and including time k [but not including measurement $y(k_0)$]. Now define the vector

$$z(k) = \begin{bmatrix} x(k) \\ w(k,k_0) \end{bmatrix} \quad (10.108)$$

In general, we define the covariance of vector a conditioned on vector c, and the cross covariance of vectors a and b conditioned on vector c, as follows:

$$\begin{aligned} \mathrm{Cov}(a|c) &= E[(a-\bar{a})(a-\bar{a})^T|c] \\ \mathrm{Cov}(a,b|c) &= E[(a-\bar{a})(b-\bar{b})^T|c] \end{aligned} \quad (10.109)$$

We can then generalize Equation (10.100) to obtain

$$\begin{aligned} \hat{z}(k) &= \hat{z}^-(k) + \\ \mathrm{Cov}[z(k),y(k)|Y(k-1)]&\mathrm{Cov}^{-1}[y(k)|Y(k-1)]\left(y(k) - H(k)\hat{x}^-(k)\right) \\ \mathrm{Cov}[z(k)|Y(k)] &= \mathrm{Cov}[z(k)|Y(k-1)] - \\ \mathrm{Cov}[z(k),y(k)|Y(k-1)]&\mathrm{Cov}^{-1}[y(k)|Y(k-1)]\mathrm{Cov}[y(k),z(k)|Y(k-1)] \end{aligned} \quad (10.110)$$

The first covariance on the right side of the above $\hat{z}(k)$ equation can be written as

$$\mathrm{Cov}[z(k),y(k)|Y(k-1)] = \begin{bmatrix} \mathrm{Cov}[x(k),y(k)|Y(k-1)] \\ \mathrm{Cov}[w(k,k_0),y(k)|Y(k-1)] \end{bmatrix} \quad (10.111)$$

Now consider the first covariance in the above equation. This can be written as

$$\begin{aligned} \mathrm{Cov}[x(k),y(k)|Y(k-1)] &= \mathrm{Cov}\left\{x(k)(H(k)x(k) + v(k)]^T|Y(k-1)\right\} \\ &= \mathrm{Cov}\left\{x(k)[H(k)x(k)]^T|Y(k-1)\right\} \\ &= \mathrm{Cov}\left\{x(k)\right\}H^T(k) \\ &= P^-(k)H^T(k) \end{aligned} \quad (10.112)$$

where the covariance of $x(k)$ and $v(k)$ is zero since they are independent. Now consider the second covariance on the right side of Equation (10.111). This can be written as

$$\begin{aligned}
\mathrm{Cov}[w(k,k_0), y(k)|Y(k-1)] &= E\left\{w(k,k_0)[y(k)-\hat{y}^-(k)]^T|Y(k-1)\right\} \\
&= E\left\{w(k,k_0)[H(k)(F(k,k_0)x(k_0)+w(k,k_0))+v(k)-\hat{y}^-(k)]^T|Y(k-1)\right\} \\
&= E\left\{w(k,k_0)w^T(k,k_0)H^T(k)\right\} \\
&= Q(k,k_0)H^T(k) \qquad (10.113)
\end{aligned}$$

where the cross covariances of $w(k,k_0)$ with $x(k_0)$, $v(k)$, and $\hat{y}^-(k)$ are zero since they are independent. We are using the notation $\hat{y}^-(k)$ to denote the expected value of $y(k)$ based on measurements up to (but not including) time k. Now consider the conditional covariance of $y(k)$ in Equation (10.110). This was derived in Equation (10.17) in Section 10.1 as

$$\mathrm{Cov}[y(k)|Y(k-1)] = H(k)P^-(k)H^T(k) + R(k) \qquad (10.114)$$

We will write this expression more compactly as

$$\mathrm{Cov}[r(k)] = S(k) \qquad (10.115)$$

where the residual $r(k) = y(k) - H(k)\hat{x}^-(k)$ and its covariance $S(k)$ are defined by the two above equations. Substituting Equations (10.112) and (10.113) into Equation (10.111), and then substituting into Equation (10.110), gives

$$\begin{aligned}
\hat{z}(k) &= \begin{bmatrix} \hat{x}(k) \\ \hat{w}(k,k_0) \end{bmatrix} \\
&= \begin{bmatrix} \hat{x}^-(k) \\ \hat{w}^-(k,k_0) \end{bmatrix} + \begin{bmatrix} P^-(k)H^T(k) \\ Q(k,k_0)H^T(k) \end{bmatrix} S^{-1}(k)r(k) \qquad (10.116)
\end{aligned}$$

This shows that

$$\begin{aligned}
\hat{w}(k,k_0) &= \hat{w}^-(k,k_0) + Q(k,k_0)H^T(k)S^{-1}(k)r(k) \\
&= Q(k,k_0)H^T(k)S^{-1}(k)r(k) \qquad (10.117)
\end{aligned}$$

because $E[\hat{w}(k,k_0)|Y(k-1)] = 0$ [since $w(k,k_0)$ is independent of the measurements]. Substituting this expression into Equation (10.107) gives

$$E[x(k_0)|Y(k)] = F(k_0,k)\left[\hat{x}(k) - Q(k,k_0)H^T(k)S^{-1}(k)r(k)\right] \qquad (10.118)$$

This is called the retrodiction of the state estimate from time k back to time k_0. Whereas a prediction equation is used to predict the state at some future time, a retrodiction equation is used to predict the state at some past time. In this case, the state estimate at time k [i.e., $\hat{x}(k)$] is retrodicted back to time k_0 to obtain the state estimate at time k_0, which is denoted above as $E[x(k_0)|Y(k)]$. Note that $E[x(k_0)|Y(k)]$ is computed on the basis of all the measurements up to and including time k, but does not consider the measurement at time k_0.

Now we can write Equation (10.110) as follows:

$$
\begin{aligned}
\mathrm{Cov}[z(k)|Y(k)] &= \mathrm{Cov}\left\{\begin{bmatrix} x(k) \\ w(k,k_0) \end{bmatrix}|Y(k)\right\} \\
&= \mathrm{Cov}\left\{\begin{bmatrix} x(k) \\ w(k,k_0) \end{bmatrix}|Y(k-1)\right\} - \\
&\quad \mathrm{Cov}\left\{\begin{bmatrix} x(k) \\ w(k,k_0) \end{bmatrix}, y(k)|Y(k-1)\right\} \mathrm{Cov}^{-1}[y(k)|Y(k-1)] \times \\
&\quad \mathrm{Cov}\left\{y(k), \begin{bmatrix} x(k) \\ w(k,k_0) \end{bmatrix}^T |Y(k-1)\right\} \\
&= \begin{bmatrix} \mathrm{Cov}[x(k)|Y(k-1)] & \mathrm{Cov}[x(k), w(k,k_0)|Y(k-1)] \\ \mathrm{Cov}[w(k,k_0), x(k)|Y(k-1)] & \mathrm{Cov}[w(k,k_0)|Y(k-1)] \end{bmatrix} - \\
&\quad \begin{bmatrix} \mathrm{Cov}[x(k), y(k)|Y(k-1)] \\ \mathrm{Cov}[w(k,k_0), y(k)|Y(k-1)] \end{bmatrix} \mathrm{Cov}^{-1}[y(k)|Y(k-1)] \times \\
&\quad \begin{bmatrix} \mathrm{Cov}[x(k), y(k)|Y(k-1)] \\ \mathrm{Cov}[w(k,k_0), y(k)|Y(k-1)] \end{bmatrix}^T
\end{aligned} \tag{10.119}
$$

From Equation (10.102) we can write

$$
\begin{aligned}
\mathrm{Cov}[x(k), w(k,k_0)|Y(k-1)] &= E[x(k)w^T(k,k_0)|Y(k-1)] \\
&= E\left\{[F(k,k_0)x(k_0) + w(k,k_0)]w^T(k,k_0)|Y(k-1)\right\} \\
&= E\left\{w(k,k_0)w^T(k,k_0)|Y(k-1)\right\} \\
&= Q(k,k_0)
\end{aligned} \tag{10.120}
$$

where we have used the independence of $x(k_0)$ and $w(k,k_0)$. Now substitute this equation along with Equations (10.112), (10.113), and (10.114) into Equation (10.119) to obtain

$$
\begin{aligned}
\mathrm{Cov}\left\{\begin{bmatrix} x(k) \\ w(k,k_0) \end{bmatrix}|Y(k)\right\} &= \begin{bmatrix} P^-(k) & Q(k,k_0) \\ Q(k,k_0) & Q(k,k_0) \end{bmatrix} - \\
&\quad \begin{bmatrix} P^-(k)H^T(k) \\ Q(k,k_0)H^T(k) \end{bmatrix} S^{-1}(k) \begin{bmatrix} P^-(k)H^T(k) \\ Q(k,k_0)H^T(k) \end{bmatrix}^T
\end{aligned} \tag{10.121}
$$

From this equation we can write the conditional covariance of $w(k,k_0)$, and cross covariance of $x(k)$ and $w(k,k_0)$, as follows:

$$
\begin{aligned}
P_w(k,k_0) &= \mathrm{Cov}[w(k,k_0)|Y(k)] \\
&= Q(k,k_0) - Q(k,k_0)H^T(k)S^{-1}(k)H(k)Q(k,k_0) \\
P_{xw}(k,k_0) &= \mathrm{Cov}[x(k), w(k,k_0)|Y(k)] \\
&= Q(k,k_0) - P^-(k)H^T(k)S^{-1}(k)H(k)Q(k,k_0)
\end{aligned} \tag{10.122}
$$

Using this in Equation (10.103) gives the conditional covariance of the state retrodiction as follows:

$$\begin{aligned}
P(k_0,k) &= \mathrm{Cov}[x(k_0)|Y(k)] \\
&= F(k_0,k)\mathrm{Cov}[x(k)-w(k,k_0)|Y(k)]F^T(k_0,k) \\
&= F(k_0,k)\{\mathrm{Cov}[x(k)|Y(k)] - \mathrm{Cov}[x(k),w(k,k_0)|Y(k)] - \\
&\quad \mathrm{Cov}^T[x(k),w(k,k_0)|Y(k)] + \mathrm{Cov}[w(k,k_0)|Y(k)]\}F^T(k_0,k) \\
&= F(k_0,k)\left\{P^+(k) - P_{xw}(k,k_0) - P_{xw}^T(k,k_0) + \right. \\
&\quad \left. P_w(k,k_0)\right\}F^T(k_0,k) \qquad (10.123)
\end{aligned}$$

Using the above along with Equation (10.101) we obtain the conditional covariance of $y(k_0)$ as

$$\begin{aligned}
S(k_0) &= \mathrm{Cov}[y(k_0)|Y(k)] \\
&= E\left\{[H(k_0)x(k_0)+v(k_0)][H(k_0)x(k_0)+v(k_0)]^T|Y(k)\right\} \\
&= H(k_0)P(k_0,k)H^T(k_0) + R(k_0) \qquad (10.124)
\end{aligned}$$

We can use Equations (10.101) and (10.103) to obtain the conditional covariance between $x(k)$ and $y(k_0)$ as

$$\begin{aligned}
P_{xy}(k,k_0) &= \mathrm{Cov}[x(k),y(k_0)|Y(k)] \\
&= \mathrm{Cov}\left\{x(k),H(k_0)F(k_0,k)[x(k)-w(k,k_0)]+v(k_0)|Y(k)\right\} \\
&= [P^+(k)-P_{xw}(k,k_0)]F^T(k_0,k)H^T(k_0) \qquad (10.125)
\end{aligned}$$

We can substitute this into the top partition of the $\hat{z}(k)$ expression in Equation (10.110) to obtain the estimate of $x(k)$ which is updated on the basis of the measurement $y(k_0)$:

$$\hat{x}(k,k_0) = \hat{x}(k) + P_{xy}(k,k_0)S^{-1}(k_0)[y(k_0) - H(k_0)\hat{x}(k_0,k)] \qquad (10.126)$$

where $\hat{x}(k_0,k)$ is the retrodiction of the state estimate given in Equation (10.118). From the top partition of the $\mathrm{Cov}[z(k)|Y(k)]$ expression in Equation (10.110) we obtain

$$\begin{aligned}
\mathrm{Cov}[x(k)|Y(k),y(k_0)] &= P(k,k_0) \\
&= P(k) - P_{xy}(k,k_0)S^{-1}(k_0)P_{xy}^T(k,k_0) \qquad (10.127)
\end{aligned}$$

These equations show how the state estimate and its covariance can be updated on the basis of an out-of-sequence measurement. The delayed-measurement Kalman filter can be summarized as follows.

The delayed-measurement Kalman filter

1. The Kalman filter is run normally on the basis of measurements that arrive sequentially. If we are presently at time k in the Kalman filter, then we have $\hat{x}^-(k)$ and $P^-(k)$, the *a priori* state estimate and covariance that are based on measurements up to and including time $(k-1)$. We also have $\hat{x}(k)$ and $P(k)$, the *a posteriori* state estimate and covariance that are based on measurements up to and including time k.

2. If we receive a measurement $y(k_0)$, where $k_0 < k$, then we can update the state estimate and its covariance to $\hat{x}(k, k_0)$ and $P(k, k_0)$ as follows.

   (a) Retrodict the state estimate from k back to k_0 as shown in Equation (10.118):

$$\begin{aligned} S(k) &= H(k)P^-(k)H^T(k) + R(k) \\ \hat{x}(k_0, k) &= F(k_0, k)\left[\hat{x}(k) - Q(k, k_0)H^T(k)S^{-1}(k)r(k)\right] \end{aligned} \tag{10.128}$$

   (b) Compute the covariance of the retrodicted state using Equations (10.122) and (10.123):

$$\begin{aligned} P_w(k, k_0) &= Q(k, k_0) - Q(k, k_0)H^T(k)S^{-1}(k)H(k)Q(k, k_0) \\ P_{xw}(k, k_0) &= Q(k, k_0) - P^-(k)H^T(k)S^{-1}(k)H(k)Q(k, k_0) \\ P(k_0, k) &= F(k_0, k)\left\{P(k) - P_{xw}(k, k_0) - P_{xw}^T(k, k_0) + \right. \\ &\quad \left. P_w(k, k_0)\right\} F^T(k_0, k) \end{aligned} \tag{10.129}$$

   (c) Compute the covariance of the retrodicted measurement at time k_0 using Equation (10.124):

$$S(k_0) = H(k_0)P(k_0, k)H^T(k_0) + R(k_0) \tag{10.130}$$

   (d) Compute the covariance of the state at time k and the retrodicted measurement at time k_0 using Equation (10.125):

$$P_{xy}(k, k_0) = [P(k) - P_{xw}(k, k_0)]F^T(k_0, k)H^T(k_0) \tag{10.131}$$

   (e) Use the delayed measurement $y(k_0)$ to update the state estimate and its covariance:

$$\begin{aligned} \hat{x}(k, k_0) &= \hat{x}(k) + P_{xy}(k, k_0)S^{-1}(k_0)[y(k_0) - H(k_0)\hat{x}(k_0, k)] \\ P(k, k_0) &= P(k) - P_{xy}(k, k_0)S^{-1}(k_0)P_{xy}^T(k, k_0) \end{aligned} \tag{10.132}$$

It is possible to make some simplifying approximations to this delayed measurement filter in order to decrease computational cost with only a slight degradation in performance [Bar02].

10.6 SUMMARY

In this chapter we discussed some important topics related to Kalman filtering that extend beyond standard results. We have seen how to verify if a Kalman filter is operating reliably. This gives us a quantifiable confidence in the accuracy of our filter estimates. We also discussed multiple-model estimation, which is a way of estimating system states when we are not sure of which model is governing the dynamics of the system. This can be useful when the system model changes in unpredictable ways. We discussed reduced-order filtering, which can be used to estimate a subset of the system states while saving computational effort. We derived a robust Kalman filter, which makes the filter less sensitive to variations

in the assumed system model. Robust filtering naturally leads into the topic of H_∞ filtering, which we will discuss in Chapter 11. Finally, we derived a way to update the state estimate when a measurement arrives at the filter in the wrong chronological order because of processing delays.

There are several other important extensions to Kalman filtering that we have not had time to discuss in this chapter. One is the variable structure filter, which is a combination of the Kalman filter with variable structure control. This guarantees stability under certain conditions and often provides performance better than the Kalman filter, especially when applied to nonlinear systems [Hab03]. Another recent proposal is the proportional integral Kalman filter, which adds an integral term to the measurement state update and thereby improves stability and reduces steady-state tracking errors [Bas99]. Another interesting topic is the use of a perturbation estimator to estimate the process noise . This allows model uncertainties to be lumped with process noise so that the process-noise estimate increases the robustness of the filter [Kwo03].

PROBLEMS

Written exercises

10.1 In this problem we consider the scalar system

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + v_k \end{aligned}$$

where w_k and v_k are white and uncorrelated with respective variances Q and R, which are unknown. A suboptimal steady-state value of K is used in the state estimator since Q and R are unknown.

a) Use the expression for P_k^- along with the first expression for P_k^+ in Equation (5.19) to find the steady-state value of P_k^- as a function of the suboptimal value of K and the true values of Q and R. [Note that the first expression for P_k^+ in Equation (5.19) does not depend on the value for K_k being optimal.]

b) Now suppose that $E(r_k^2)$ and $E(r_{k+1}r_k)$ are found numerically as the filter runs. Find the true value of R and the steady-state value of P_k^- as a function of $E(r_k^2)$ and $E(r_{k+1}r_k)$.

c) Use your results from parts (a) and (b) to find the true value of Q.

10.2 Show that the innovations $r = y - C\hat{x}$ of the continuous-time Kalman filter is white with covariance R.

10.3 Consider the system described in Problem 5.1. Find the steady-state variance of the Kalman filter innovations when $Q = R$ and when $Q = 2R$.

10.4 Consider the system of Problem 10.3 with $Q = R = 1$. Suppose the Kalman filter for the system has reached steady state. At time k the innovations $r_k = y_k - \hat{x}_k^-$.

a) Find an approximate value for $\text{pdf}(y_k|p)$ (where p is the model used in the Kalman filter) if $r_k = 0$, if $r_k = 1$, and if $r_k = 2$.

b) Suppose that the use of model p_1 gives $r_k = 0$, model p_2 gives $r_k = 1$, and model p_3 gives $r_k = 2$. Further suppose that $\Pr(p_1|y_{k-1}) = 1/4$, $\Pr(p_2|y_{k-1}) = 1/4$, and $\Pr(p_3|y_{k-1}) = 1/2$. Find $\Pr(p_j|y_k)$ for $j = 1, 2, 3$.

10.5 Consider the system described in Example 4.1 where the measurement consists of the predator population. Suppose that we want to estimate $x(1) + x(2)$, the sum of the predator and prey populations. Create an equivalent system with transformed states such that our goal is to estimate the first element of the transformed state vector.

10.6 Consider the system

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w_k \\ y_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k \end{aligned}$$

where w_k and v_k are uncorrelated zero-mean white noise processes with variances q and R, respectively.

a) Use Anderson's approach to reduced-order filtering to estimate the first element of the state vector. Find steady-state values for $\tilde{P}$, $\tilde{\tilde{P}}$, Σ, $\tilde{\Pi}$, $\tilde{\tilde{\Pi}}$, and P. Find the steady-state gain K of the reduced-order filter.

b) Use the full-order filter to estimate the entire state vector. Find steady-state values for P and K.

c) Comment on the comparison between your answer for P in part (a) and part (b).

10.7 Consider the reduced-order filter of Example 10.3 with the initial condition $\tilde{\tilde{P}}_0^+ = 1$.

a) Find analytical expressions for the steady-state values of $\tilde{K}$, α, $\tilde{P}^+$, Σ^+, $\tilde{\tilde{P}}^+$, $\tilde{P}^-$, Σ^-, and $\tilde{\tilde{P}}^-$

b) What does the reduced-order filter indicate for the steady-state *a posteriori* estimation-error variance of the first state? Find an analytical expression for the true steady-state *a posteriori* estimation-error variance of the first state when the reduced-order filter is used. Your answer should be a function of $x(2)$. Solve for the true steady-state *a posteriori* estimation-error variance of the first state when $x(2) = 0$, when $x(2) = 1$, and when $x(2) = 2$.

c) What is the steady-state *a posteriori* estimation-error variance of the first state when the full-order filter is used?

10.8 Verify that the two expressions in Equation (10.98) are respectively equal to the cross-covariance of x and y, and the covariance of y.

10.9 Suppose you have the linear system $x_{k+1} = Fx_k + w_k$, where $w_k \sim (0, Q_k)$ is zero-mean white noise. Define $w(k+2, k)$ as the cumulative effect of all of the process noise on the state from time k to time $(k+2)$. What are the mean and covariance of $w(k+2, k)$?

10.10 Suppose that a Kalman filter is running with

$$F = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$
\begin{aligned}
H &= \begin{bmatrix} 1 & 0 \end{bmatrix} \\
Q &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\
R &= 1 \\
P^+(k) &= \begin{bmatrix} 1/2 & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned}
$$

An out-of-sequence measurement from time $(k-1)$ is received at the filter.

a) What was the value of $P^-(k)$?

b) Use the delayed-measurement filter to find the quantities $P_w(k, k-1)$, $P_{xw}(k, k-1)$, $P(k-1, k)$, $P_{xy}(k, k-1)$, and $P(k, k-1)$.

c) Realizing that the measurement at time $(k-1)$ was not received at time $(k-1)$, derive the value of $P^-(k-1)$. Now suppose that the measurement was received in the correct sequence at time $(k-1)$. Use the standard Kalman filter equations to compute $P^+(k-1)$, $P^-(k)$, and $P^+(k)$. How does your computed value of $P^+(k)$ compare with the value of $P(k, k-1)$ that you computed in part (b) of this problem?

10.11 Under what conditions will P_y in Equation (10.100) be invertible for all k?

Computer exercises

10.12 Consider the equations

$$
\begin{aligned}
300x + 400y &= 700 \\
100x + 133y &= 233
\end{aligned}
$$

a) What is the solution of these equations?

b) What is the solution of these equations if each constant in the second equation increases by 1?

c) What is the condition number of the original set of equations?

10.13 Repeat Problem 10.12 for the equations

$$
\begin{aligned}
300x + 400y &= 700 \\
100x + 200y &= 200
\end{aligned}
$$

Comment on the difference between this set of equations and the set given in Problem 10.12.

10.14 Tire tread is measured every τ weeks. After τ weeks, 20% of the tread has worn off, so we can model the dynamics of the tread height as $x_{k+1} = fx_k + w_k$, where $f = 0.8$, and w_k is zero-mean white noise with a variance of 0.01. We measure the tread height every τ weeks with zero-mean white measurement noise that has a variance of 0.01. The initial tread height is known to be exactly 1 cm. Write a program to simulate the system and a Kalman filter to estimate the tread height.

a) Run the program for 10 time steps per tire, and for 1000 tires. What is the mean of the 10,000 measurement residuals?

b) Suppose the Kalman filter designer incorrectly believes that 30% of the tread wears off every τ weeks. What is the mean of the 10,000 measurement residuals in this case?

c) Suppose the Kalman filter designer incorrectly believes that 10% of the tread wears off every τ weeks. What is the mean of the 10,000 measurement residuals in this case?

10.15 Consider the system described in Problem 10.14. Suppose the engineer does not know the true value of f but knows the initial probabilities $\Pr(f = 0.8) = \Pr(f = 0.85) = \Pr(f = 0.9) = 1/3$. Run the multiple-model estimator for 10 time steps on 100 tires to estimate f. The f probabilities at each time step can be taken as the mean of the 100 f probabilities that are obtained from the 100 tire simulations, and similarly for the f estimate at each time step. Plot the f probabilities and the f estimate as a function of time.

10.16 Consider a scalar system with $F = H = 1$ and nominal noise variances $Q = R = 5$. The true but unknown noise variances $\tilde{Q}$ and $\tilde{R}$ are given as

$$\begin{aligned}
\tilde{Q} &= (1+\alpha)Q \\
\tilde{R} &= (1+\beta)R \\
E(\alpha^2) &= \sigma_1^2 = 1/2 \\
E(\beta^2) &= \sigma_2^2 = 1
\end{aligned}$$

where α and β are independent zero-mean random variables. The variance of the *a posteriori* estimation error is P if $\alpha = \beta = 0$. In general, α and β are nonzero and the variance of the estimation error is $P + \Delta P$. Plot P, $E(\Delta P^2)$, and $(P + E(\Delta P^2))$ as a function of K for $K \in [0.3, 0.7]$. What are the minimizing values of K for the three plots?

PART III

THE H_∞ FILTER

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

CHAPTER 11

The H_∞ filter

> [Kalman filtering] assumes that the message generating process has a known dynamics and that the exogenous inputs have known statistical properties. Unfortunately, these assumptions limit the utility of minimum variance estimators in situations where the message model and/or the noise descriptions are unknown.
>
> —Uri Shaked and Yahali Theodor [Sha92]

As we have seen in earlier chapters, the Kalman filter is an effective tool for estimating the states of a system. The early success in the 1960s of the Kalman filter in aerospace applications led to attempts to apply it to more common industrial applications in the 1970s. However, these attempts quickly made it clear that a serious mismatch existed between the underlying assumptions of Kalman filters and industrial state estimation problems. Accurate system models are not as readily available for industrial problems. The government spent millions of dollars on the space program in the 1960s (hence the accurate system models), but industry rarely has millions of dollars to spend on engineering problems (hence the inaccurate system models). In addition, engineers rarely understand the statistical nature of the noise processes that impinge on industrial processes. After a decade or so of reappraising the nature and role of Kalman filters, engineers realized they needed a new filter that could handle modeling errors and noise uncertainty. State estimators that can tolerate such uncertainty are called robust. Although robust

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

estimators based on Kalman filter theory can be designed (as seen in Section 10.4), these approaches are somewhat ad-hoc in that they attempt to modify an already existing approach. The H_∞ filter was specifically designed for robustness.

In Section 11.1 we derive a different form of the Kalman filter and discuss the limitations of the Kalman filter. Section 11.2 discusses constrained optimization using Lagrange multipliers, which we will need later for our derivation of the H_∞ filter. In Section 11.3 we use a game theory approach to derive the discrete-time H_∞ filter, which minimizes the worst-case estimation error. This is in contrast to the Kalman filter's minimization of the expected value of the variance of the estimation error. Furthermore, the H_∞ filter does not make any assumptions about the statistics of the process and measurement noise (although this information can be used in the H_∞ filter if it is available). Section 11.4 presents the continuous-time H_∞ filter, and Section 11.5 discusses an alternative method for deriving the H_∞ filter using a transfer function approach.

11.1 INTRODUCTION

In this section we will first derive an alternate form for the Kalman filter. We do this to facilitate comparisons that we will make later in this chapter between the Kalman and H_∞ filters. After we derive an alternate Kalman filter form, we will briefly discuss the limitations of the Kalman filter.

11.1.1 An alternate form for the Kalman filter

Recall that the Kalman filter estimates the state of a linear dynamic system defined by the equations

$$\begin{aligned} x_{k+1} &= F_k x_k + w_k \\ y_k &= H_k x + v_k \end{aligned} \tag{11.1}$$

where $\{w_k\}$ and $\{v_k\}$ are stochastic processes with covariances Q_k and R_k, respectively. As derived in Section 5.1, the Kalman filter equations are given as follows:

$$\begin{aligned} \hat{x}_{k+1}^- &= F_k \hat{x}_k^- + F_k K_k (y_k - H_k \hat{x}_k^-) \\ K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ P_k^- &= F_{k-1} P_{k-1}^+ F_{k-1}^T + Q_{k-1} \\ P_k^+ &= (I - K_k H_k) P_k^- \end{aligned} \tag{11.2}$$

Using the matrix inversion lemma from Section 1.1.2 we see that

$$\begin{aligned} (H_k P_k^- H_k^T + R_k)^{-1} &= R_k^{-1} - R_k^{-1} H_k (\mathcal{I}_k^- + H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1} \\ &= R_k^{-1} - R_k^{-1} H_k (I + P_k^- H_k^T R_k^{-1} H_k)^{-1} P_k^- H_k^T R_k^{-1} \end{aligned} \tag{11.3}$$

where $\mathcal{I}_k$ is the information matrix (i.e., the inverse of the covariance matrix P_k). The Kalman gain can therefore be written as follows:

$$
\begin{aligned}
K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\
&= P_k^- H_k^T R_k^{-1} - P_k^- H_k^T R_k^{-1} H_k (I + P_k^- H_k^T R_k^{-1} H_k)^{-1} P_k^- H_k^T R_k^{-1} \\
&= \left[I - P_k^- H_k^T R_k^{-1} H_k (I + P_k^- H_k^T R_k^{-1} H_k)^{-1}\right] P_k^- H_k^T R_k^{-1} \\
&= \left[(I + P_k^- H_k^T R_k^{-1} H_k) - P_k^- H_k^T R_k^{-1} H_k\right] (I + P_k^- H_k^T R_k^{-1} H_k)^{-1} P_k^- H_k^T R_k^{-1} \\
&= (I + P_k^- H_k^T R_k^{-1} H_k)^{-1} P_k^- H_k^T R_k^{-1} \qquad (11.4)
\end{aligned}
$$

Substituting this into the expression for P_{k+1}^- in Equation (11.2) we get

$$
\begin{aligned}
P_{k+1}^- &= F_k P_k^+ F_k^T + Q_k \\
&= F_k (I - K_k H_k) P_k^- F_k^T + Q_k \\
&= F_k P_k^- F_k^T - F_k K_k H_k P_k^- F_k^T + Q_k \\
&= F_k P_k^- F_k^T - F_k (I + P_k^- H_k^T R_k^{-1} H_k)^{-1} P_k^- H_k^T R_k^{-1} H_k P_k^- F_k^T + Q_k \\
&= F_k P_k^- F_k^T - F_k (\mathcal{I}_k^- + H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1} H_k P_k^- F_k^T + Q_k \qquad (11.5)
\end{aligned}
$$

Apply the matrix inversion lemma again to the inverse on the right side of the above equation to obtain

$$
\begin{aligned}
P_{k+1}^- &= F_k P_k^- F_k^T - \\
&\quad F_k \left[P_k^- - P_k^- H_k^T (R_k + H_k P_k^- H_k^T)^{-1} H_k P_k^-\right] H_k^T R_k^{-1} H_k P_k^- F_k^T + Q_k \\
&= F_k P_k^- \left[I - H_k^T R_k^{-1} H_k P_k^- + \right. \\
&\quad \left. H_k^T (R_k + H_k P_k^- H_k^T)^{-1} H_k P_k^- H_k^T R_k^{-1} H_k P_k\right] F_k^T + Q_k \\
&= F_k P_k^- T_k F_k^T + Q_k \qquad (11.6)
\end{aligned}
$$

where T_k is defined by the above equation. Apply the matrix inversion lemma to the inverse that is in T_k to obtain

$$
\begin{aligned}
T_k &= I - H_k^T R_k^{-1} H_k P_k^- + \\
&\quad H_k^T \left[R_k^{-1} - R_k^{-1} H_k (\mathcal{I}_k^- + H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1}\right] H_k P_k^- H_k^T R_k^{-1} H_k P_k \\
&= I - H_k^T R_k^{-1} H_k P_k^- + (H_k^T R_k^{-1} H_k P_k^-)^2 - \\
&\quad H_k^T R_k^{-1} H_k (\mathcal{I}_k^- + H_k^T R_k^{-1} H_k)^{-1} (H_k^T R_k^{-1} H_k P_k^-)^2 \\
&= I - H_k^T R_k^{-1} H_k P_k^- + (H_k^T R_k^{-1} H_k P_k^-)^2 - \\
&\quad H_k^T R_k^{-1} H_k P_k^- (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} (H_k^T R_k^{-1} H_k P_k^-)^2 \\
&= I - H_k^T R_k^{-1} H_k P_k^- + (H_k^T R_k^{-1} H_k P_k^-)^2 - \\
&\quad (H_k^T R_k^{-1} H_k P_k^-)^3 (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} \\
&= \left[(I + H_k^T R_k^{-1} H_k P_k^-) - H_k^T R_k^{-1} H_k P_k^- (I + H_k^T R_k^{-1} H_k P_k^-) + \right. \\
&\quad \left. (H_k^T R_k^{-1} H_k P_k^-)^2 (I + H_k^T R_k^{-1} H_k P_k^-) - (H_k^T R_k^{-1} H_k P_k^-)^3\right] \times \\
&\quad (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} \\
&= (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} \qquad (11.7)
\end{aligned}
$$

Substituting this expression for T_k into Equation (11.6) gives

$$
P_{k+1}^- = F_k P_k^- (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} F_k^T + Q_k \qquad (11.8)
$$

From Equation (11.4) the Kalman gain can be written as

$$K_k = (I + P_k^- H_k^T R_k^{-1} H_k)^{-1} P_k^- H_k^T R_k^{-1} \tag{11.9}$$

We can premultiply outside the parentheses by P_k^-, and postmultiply each term inside the parenthesis by P_k^-, to obtain

$$K_k = P_k^- (P_k^- + P_k^- H_k^T R_k^{-1} H_k P_k^-)^{-1} P_k^- H_k^T R_k^{-1} \tag{11.10}$$

We can postmultiply outside the parentheses by the inverse of P_k^-, and premultiply each term inside the parentheses by the inverse of P_k^-, to obtain

$$K_k = P_k^- (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} H_k^T R_k^{-1} \tag{11.11}$$

Combining this expression for K_k with Equations (11.2) and (11.8) we can summarize the Kalman filter as follows:

$$\begin{aligned} \hat{x}_{k+1}^- &= F_k \hat{x}_k^- + F_k K_k (y_k - H_k \hat{x}_k^-) \\ K_k &= P_k^- (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} H_k^T R_k^{-1} \\ P_{k+1}^- &= F_k P_k^- (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} F_k^T + Q_k \end{aligned} \tag{11.12}$$

11.1.2 Kalman filter limitations

The Kalman filter works well, but only under certain conditions.

- First, we need to know the mean and correlation of the noise w_k and v_k at each time instant.
- Second, we need to know the covariances Q_k and R_k of the noise processes. The Kalman filter uses Q_k and R_k as design parameters, so if we do not know Q_k and R_k then it may be difficult to successfully use a Kalman filter.
- Third, the attractiveness of the Kalman filter lies in the fact that it is the one estimator that results in the smallest possible standard deviation of the estimation error. That is, the Kalman filter is the minimum variance estimator if the noise is Gaussian, and it is the linear minimum variance estimator if the noise is not Gaussian. If we desire to minimize a different cost function (such as the worst-case estimation error) then the Kalman filter may not accomplish our objectives.
- Finally, we need to know the system model matrices F_k and H_k.

So what do we do if one of the Kalman filter assumptions is not satisfied? What should we do if we do not have any information about the noise statistics? What should we do if we want to minimize the worst-case estimation error rather than the covariance of the estimation error?

Perhaps we could just use the Kalman filter anyway, even though its assumptions are not satisfied, and hope for the best. That is a common solution to our Kalman filter quandary and it works reasonably well in many cases. However, there is yet another option that we will explore in this chapter: the H_∞ filter, also called the minimax filter. The H_∞ filter does not make any assumptions about the noise, and it minimizes the worst-case estimation error (hence the term minimax).

11.2 CONSTRAINED OPTIMIZATION

In this section we show how constrained optimization can be performed through the use of Lagrange multipliers. This background is required for the solution of the H_∞ filtering problem that is presented in Section 11.3. In Section 11.2.1 we will investigate static problems (i.e., problems in which the independent variables are constant). In Section 11.2.2 we will take a brief segue to look at problems with inequality constraints. In Section 11.2.3 we will extend our constrained optimization method to dynamic problems (i.e., problems in which the independent variables change with time).

11.2.1 Static constrained optimization

Suppose we want to minimize some scalar function $J(x, w)$ with respect to x and w. x is an n-dimensional vector, and w is an m-dimensional vector. w is the independent variable and x is the dependent variable; that is, x is somehow determined by w. Suppose our vector-valued constraint is given as $f(x, w) = 0$. Further assume that the dimension of $f(x, w)$ is the same as the dimension of x. This problem can be written as

$$\min_{x,w} J(x, w) \text{ such that } f(x, w) = 0 \tag{11.13}$$

Suppose that the constrained minimum of $J(x, w)$ occurs at $x = x^*$ and $w = w^*$. We call this the stationary point of $J(x, w)$. Now suppose that we choose values of x and w such that x is close to x^*, w is close to w^*, and $f(x, w) = 0$. Expanding $J(x, w)$ and $f(x, w)$ in a Taylor series around x^* and w^* gives

$$\begin{aligned} J(x, w) &= J(x^*, w^*) + \left.\frac{\partial J}{\partial x}\right|_{x^*,w^*} \Delta x + \left.\frac{\partial J}{\partial w}\right|_{x^*,w^*} \Delta w \\ f(x, w) &= f(x^*, w^*) + \left.\frac{\partial f}{\partial x}\right|_{x^*,w^*} \Delta x + \left.\frac{\partial f}{\partial w}\right|_{x^*,w^*} \Delta w \end{aligned} \tag{11.14}$$

where higher-order terms have been neglected (with the assumption that x is close to x^*, and w is close to w^*), $\Delta x = x - x^*$, and $\Delta w = w - w^*$. These equations can be written as

$$\begin{aligned} \Delta J(x, w) &= J(x, w) - J(x^*, w^*) \\ &= \left.\frac{\partial J}{\partial x}\right|_{x^*,w^*} \Delta x + \left.\frac{\partial J}{\partial w}\right|_{x^*,w^*} \Delta w \\ \Delta f(x, w) &= f(x, w) - f(x^*, w^*) \\ &= \left.\frac{\partial f}{\partial x}\right|_{x^*,w^*} \Delta x + \left.\frac{\partial f}{\partial w}\right|_{x^*,w^*} \Delta w \end{aligned} \tag{11.15}$$

Now note that for values of x and w that are close to x^* and w^*, we have $\Delta J(x, w) = 0$. This is because the partial derivatives on the right side of the $\Delta J(x, w)$ equation are zero at the stationary point of $J(x, w)$. We also see that $\Delta f(x, w) = 0$ at the stationary point of $J(x, w)$. This is because $f(x^*, w^*) = 0$ at the constrained stationary point of $J(x, w)$, and we chose x and w such that $f(x, w) = 0$ also. The

above equations can therefore be written as

$$\begin{aligned}\left.\frac{\partial J}{\partial x}\right|_{x^*,w^*}\Delta x+\left.\frac{\partial J}{\partial w}\right|_{x^*,w^*}\Delta w &= 0\\ \left.\frac{\partial f}{\partial x}\right|_{x^*,w^*}\Delta x+\left.\frac{\partial f}{\partial w}\right|_{x^*,w^*}\Delta w &= 0\end{aligned} \tag{11.16}$$

These equations are true for arbitrary x and w that are close to x^* and w^* and that satisfy the constraint $f(x,w)=0$. Equation (11.16) can be solved for Δx as

$$\Delta x=-\left(\left.\frac{\partial f}{\partial x}\right|_{x^*,w^*}\right)^{-1}\left.\frac{\partial f}{\partial w}\right|_{x^*,w^*}\Delta w \tag{11.17}$$

This can be substituted into Equation (11.16) to obtain

$$\left.\frac{\partial J}{\partial w}\right|_{x^*,w^*}-\left.\frac{\partial J}{\partial x}\right|_{x^*,w^*}\left(\left.\frac{\partial f}{\partial x}\right|_{x^*,w^*}\right)^{-1}\left.\frac{\partial f}{\partial w}\right|_{x^*,w^*}=0 \tag{11.18}$$

This equation, combined with the constraint $f(x,w)=0$, gives us $(m+n)$ equations that can be solved for the vectors w and x to find the constrained stationary point of $J(x,w)$.

Now consider the augmented cost function

$$J_a=J+\lambda^T f \tag{11.19}$$

where λ is an n-element unknown constant vector called a Lagrange multiplier. Note that

$$\begin{aligned}\frac{\partial J_a}{\partial x} &= \frac{\partial J}{\partial x}+\lambda^T\frac{\partial f}{\partial x}\\ \frac{\partial J_a}{\partial w} &= \frac{\partial J}{\partial w}+\lambda^T\frac{\partial f}{\partial w}\\ \frac{\partial J_a}{\partial \lambda} &= f\end{aligned} \tag{11.20}$$

If we set all three of these equations equal to zero then we have

$$\begin{aligned}\lambda^T &= \frac{-\partial J}{\partial x}\left(\frac{\partial f}{\partial x}\right)^{-1}\\ \frac{\partial J}{\partial w}-\frac{\partial J}{\partial x}\left(\frac{\partial f}{\partial x}\right)^{-1}\frac{\partial f}{\partial w} &= 0\\ f &= 0\end{aligned} \tag{11.21}$$

The first equation gives us the value of the Lagrange multiplier, the second equation is identical to Equation (11.18), and the third equation forces the constraint to be satisfied. We therefore see that we can solve the original constrained problem by creating an augmented cost function J_a, taking the partial derivatives with respect to x, w, and λ, setting them equal to zero, and solving for x, w, and λ. The partial derivative equations give us $(2n+m)$ equations to solve for the n-element vector x, the m-element vector w, and the n-element vector λ. We have increased the dimension of the original problem by introducing a Lagrange multiplier, but we have transformed the constrained optimization problem into an unconstrained optimization problem, which can simplify the problem considerably.

■ **EXAMPLE 11.1**

Suppose we need to find the minimum of $J(x,u) = x^2/2 + xu + u^2 + u$ with respect to x and u such that $f(x,u) = x - 3 = 0$. This simple example can be solved by simply realizing that $x = 3$ in order to satisfy the constraint. Substituting $x = 3$ into $J(x,u)$ gives $J(x,u) = 9/2 + 4u + u^2$. Setting the derivative with respect to u equal to zero and solving for u gives $u = -2$.

We can also solve this problem using the Lagrange multiplier method. We create an augmented cost function as

$$\begin{aligned} J_a &= J + \lambda^T f \\ &= x^2/2 + xu + u^2 + u + \lambda(x-3) \end{aligned} \tag{11.22}$$

The Lagrange multiplier λ has the same dimension as x (scalar in this example). The three necessary conditions for a constrained stationary point of J are obtained by setting the partial derivations of Equation (11.20) equal to 0.

$$\begin{aligned} \frac{\partial J_a}{\partial x} &= x + u + \lambda = 0 \\ \frac{\partial J_a}{\partial u} &= x + 2u + 1 = 0 \\ \frac{\partial J_a}{\partial \lambda} &= x - 3 = 0 \end{aligned} \tag{11.23}$$

Solving these three equations for x, u, and λ gives $x = 3$, $u = -2$, and $\lambda = -1$. In this example the Lagrange multiplier method seems to require more effort than simply solving the problem directly. However, in more complicated constrained optimization problems the Lagrange multiplier method is essential for finding a solution.

$\nabla\nabla\nabla$

11.2.2 Inequality constraints

Suppose that we want to minimize a scalar function that is subject to an inequality constraint:

$$\min J(x) \text{ such that } f(x) \leq 0 \tag{11.24}$$

This can be reduced to two minimization problems, neither of which contain inequality constraints. The first minimization problem is unconstrained, and the second minimization problem has an equality constraint:

1. $\min J(x)$
2. $\min J(x)$ such that $f(x) = 0$

In other words, the optimal value of x is either not on the constraint boundary [i.e., $f(x) < 0$], or it is on the constraint boundary [i.e., $f(x) = 0$]. If it is not on the constraint boundary then $f(x) < 0$ and the optimal value of x is obtained by solving the problem without the constraint. If it is on the constraint boundary then $f(x) = 0$ at the constrained minimum, and the optimal value of x is obtained by solving the problem with the equality constraint $f(x) = 0$.

The procedure for solving Equation (11.24) involves solving the unconstrained problem first. Then we check to see if the unconstrained minimum satisfies the constraint. If the unconstrained minimum satisfies the constraint, then the unconstrained minimum solves the inequality-constrained minimization problem and we are done. However, if the unconstrained minimum does not satisfy the constraint, then the minimization problem with the inequality constraint is equivalent to the minimization problem with the equality constraint. So we solve the problem with the equality constraint $f(x) = 0$ to obtain the final solution. This is illustrated for the scalar case in Figure 11.1.

Figure 11.1 This illustrates the constrained minimization of x^2. If the constraint is $x - 1 \leq 0$, then the constrained minimum is equal to the unconstrained minimum and occurs at $x = 0$. If the constraint is $x + 1 \leq 0$, then the constrained minimum can be solved by enforcing the equality constraint $x + 1 = 0$ and occurs at $x = -1$.

When we extend this idea to more than one dimension, we obtain the following procedure, which is called the active-set method for optimization with inequality constraints [Fle81, Gil81].

1. The problem is to minimize $J(x)$ such that $f(x) \leq 0$, where $f(x)$ is an m-element constraint function and the inequality is taken one element at a time.

2. First solve the unconstrained minimization problem. If the unconstrained solution satisfies the constraint $f(x) \leq 0$ then the problem is solved. If not, continue to the next step.

3. For all possible combinations of constraints, solve the problem using those constraints as equality constraints. If the solution satisfies the remaining (unused) constraints, then the solution is feasible. Note that this step requires the solution of $(2^m - 1)$ constrained optimization problems.

4. Out of all the feasible solutions that were obtained in the previous step, the one with the smallest $J(x)$ is the solution to the constrained minimization problem.

Note that there are also other methods for solving optimization problems with inequality constraints, including primal–dual interior-point methods [Wri97].

11.2.3 Dynamic constrained optimization

In this section we extend the Lagrange multiplier method of constrained optimization to the optimization of dynamic systems. Suppose that we have a dynamic system given as

$$x_{k+1} = F_k x_k + w_k \qquad (k = 0, \cdots, N-1) \tag{11.25}$$

where x_k is an n-dimensional state vector. We want to minimize the scalar function

$$J = \psi(x_0) + \sum_{k=0}^{N-1} \mathcal{L}_k \tag{11.26}$$

where $\psi(x_0)$ is a known function of x_0, and $\mathcal{L}_k$ is a known function of x_k and w_k. This is a constrained dynamic optimization problem similar to the type that arises in optimal control [Lew86a, Ste94]. It is slightly different than typical optimal control problems because $\psi(x_k)$ in the above equation is evaluated at the initial time ($k = 0$) instead of the final time ($k = N$), but the methods of optimal control can be used with only slight modifications to solve our problem. The constraints are given in Equation (11.25). From the previous section we know that we can solve this problem by introducing a Lagrange multiplier λ, creating an augmented cost function J_a, and then setting the partial derivatives of J_a with respect to x_k, w_k, and λ equal to zero. Since we have N constraints in Equation (11.25) (each of dimension n), we have to introduce N Lagrange multipliers $\lambda_1, \cdots, \lambda_N$ (each of dimension n). The augmented cost function is therefore written as

$$J_a = \psi(x_0) + \sum_{k=0}^{N-1} \left[\mathcal{L}_k + \lambda_{k+1}^T (F_k x_k + w_k - x_{k+1})\right] \tag{11.27}$$

This can be written as

$$\begin{aligned} J_a &= \psi(x_0) + \sum_{k=0}^{N-1} \left[\mathcal{L}_k + \lambda_{k+1}^T (F_k x_k + w_k)\right] - \sum_{k=0}^{N-1} \lambda_{k+1}^T x_{k+1} \\ &= \psi(x_0) + \sum_{k=0}^{N-1} \left[\mathcal{L}_k + \lambda_{k+1}^T (F_k x_k + w_k)\right] - \sum_{k=0}^{N} \lambda_k^T x_k + \lambda_0^T x_0 \end{aligned} \tag{11.28}$$

where λ_0 is now an additional term in the Lagrange multiplier sequence. It is not in the original augmented cost function, but we will see in Section 11.3 that its value will be determined when we solve the constrained optimization problem. Now we define the Hamiltonian $\mathcal{H}_k$ as

$$\mathcal{H}_k = \mathcal{L}_k + \lambda_{k+1}^T (F_k x_k + w_k) \tag{11.29}$$

With this notation we can write the augmented cost function as follows.

$$
\begin{aligned}
J_a &= \psi(x_0) + \sum_{k=0}^{N-1} \mathcal{H}_k - \sum_{k=0}^{N} \lambda_k^T x_k + \lambda_0^T x_0 \\
&= \psi(x_0) + \sum_{k=0}^{N-1} \mathcal{H}_k - \sum_{k=0}^{N-1} \lambda_k^T x_k - \lambda_N^T x_N + \lambda_0^T x_0 \\
&= \psi(x_0) + \sum_{k=0}^{N-1} \left(\mathcal{H}_k - \lambda_k^T x_k\right) - \lambda_N^T x_N + \lambda_0^T x_0 \qquad (11.30)
\end{aligned}
$$

The conditions that are required for a constrained stationary point are

$$
\begin{aligned}
\frac{\partial J_a}{\partial x_k} &= 0 \quad (k = 0, \cdots, N) \\
\frac{\partial J_a}{\partial w_k} &= 0 \quad (k = 0, \cdots, N-1) \\
\frac{\partial J_a}{\partial \lambda_k} &= 0 \quad (k = 0, \cdots, N) \qquad (11.31)
\end{aligned}
$$

These conditions can also be written as

$$
\begin{aligned}
\frac{\partial J_a}{\partial x_0} &= 0 \\
\frac{\partial J_a}{\partial x_N} &= 0 \\
\frac{\partial J_a}{\partial x_k} &= 0 \quad (k = 1, \cdots, N-1) \\
\frac{\partial J_a}{\partial w_k} &= 0 \quad (k = 0, \cdots, N-1) \\
\frac{\partial J_a}{\partial \lambda_k} &= 0 \quad (k = 0, \cdots, N) \qquad (11.32)
\end{aligned}
$$

The fifth condition ensures that the constraint $x_{k+1} = F_k x_k + w_k$ is satisfied. Based on the expression for J_a in Equation (11.30), the first four conditions above can be written as

$$
\begin{aligned}
\lambda_0^T + \frac{\partial \psi_0}{\partial x_0} &= 0 \\
-\lambda_N^T &= 0 \\
\lambda_k^T &= \frac{\partial \mathcal{H}_k}{\partial x_k} \quad (k = 1, \cdots, N-1) \\
\frac{\partial \mathcal{H}_k}{\partial w_k} &= 0 \quad (k = 0, \cdots, N-1) \qquad (11.33)
\end{aligned}
$$

This gives us the necessary conditions for a constrained stationary point of our dynamic optimization problem. These are the results that we will use to solve the H_∞ estimation problem in the next section.

11.3 A GAME THEORY APPROACH TO H_∞ FILTERING

The H_∞ solution that we present in this section was originally developed by Ravi Banavar [Ban92] and is further discussed in [She95, She97]. Suppose we have the standard linear discrete-time system

$$\begin{aligned} x_{k+1} &= F_k x_k + w_k \\ y_k &= H_k x_k + v_k \end{aligned} \tag{11.34}$$

where w_k and v_k are noise terms. These noise terms may be random with possibly unknown statistics, or they may be deterministic. They may have a nonzero mean. Our goal is to estimate a linear combination of the state. That is, we want to estimate z_k, which is given by

$$z_k = L_k x_k \tag{11.35}$$

where L_k is a user-defined matrix (assumed to be full rank). If we want to directly estimate x_k (as in the Kalman filter) then we set $L_k = I$. But in general we may only be interested in certain linear combinations of the state. Our estimate of z_k is denoted $\hat{z}_k$, and our estimate of the state at time 0 is denoted $\hat{x}_0$. We want to estimate z_k based on measurements up to and including time $(N-1)$. In the game theory approach to H_∞ filtering we define the following cost function:

$$J_1 = \frac{\sum_{k=0}^{N-1} ||z_k - \hat{z}_k||^2_{S_k}}{||x_0 - \hat{x}_0||^2_{P_0^{-1}} + \sum_{k=0}^{N-1} \left(||w_k||^2_{Q_k^{-1}} + ||v_k||^2_{R_k^{-1}} \right)} \tag{11.36}$$

Our goal as engineers is to find an estimate $\hat{z}_k$ that minimizes J_1. Nature's goal as our adversary is to find disturbances w_k and v_k, and the initial state x_0, to maximize J_1. Nature's ultimate goal is to maximize the estimation error $(z_k - \hat{z}_k)$. The way that nature maximizes $(z_k - \hat{z}_k)$ is by a clever choice of w_k, v_k, and x_0. Nature could maximize $(z_k - \hat{z}_k)$ by simply using infinite magnitudes for w_k, v_k, and x_0, but this would not make the game fair. That is why we define J_1 with $(x_0 - \hat{x}_0)$, w_k, and v_k in the denominator. If nature uses large magnitudes for w_k, v_k, and x_0 then $(z_k - \hat{z}_k)$ will be large, but J_1 may not be large because of the denominator. The form of J_1 prevents nature from using brute force to maximize $(z_k - \hat{z}_k)$. Instead, nature must try to be clever in its choice of w_k, v_k, and x_0 as it tries to maximize $(z_k - \hat{z}_k)$. Likewise, we as engineers must be clever in finding an estimation strategy to minimize $(z_k - \hat{z}_k)$.

This discussion highlights a fundamental difference in the philosophy of the Kalman filter and the H_∞ filter. In Kalman filtering, nature is assumed to be indifferent. The pdf of the noise is given. We (as filter designers) know the pdf of the noise and can use that knowledge to obtain a statistically optimal state estimate. But nature cannot change the pdf to degrade our state estimate. In H_∞ filtering, nature is assumed to be perverse and actively seeks to degrade our state estimate as much as possible. Intuition and experience seem to indicate that neither of these extreme viewpoints of nature is entirely correct, but reality probably lies somewhere in the middle.[1]

[1]Nevertheless, it is advisable to remember the *principle of perversity of inanimate objects* [Bar01, p. 96] – for instance, when dropping a piece of buttered toast on the floor, the probability is significantly more than 50% that the toast will land buttered-side down.

P_0, Q_k, R_k, and S_k in Equation (11.36) are symmetric positive definite matrices chosen by the engineer based on the specific problem. For example, if the user is particularly interested in obtaining an accurate estimate of the third element of z_k, then $S_k(3,3)$ should be chosen to be large relative to the other elements of S_k. If the user knows *a priori* that the second element of the w_k disturbance is small, then $Q_k(2,2)$ should be chosen to be small relative to the other elements of Q_k. In this way, we see that P_0, Q_k, and R_k are analogous to those same quantities in the Kalman filter, if those quantities are known. That is, suppose that we know that the initial estimation error, the process noise, and the measurement noise are zero-mean. Further suppose that we know their covariances. Then we should use those quantities for P_0, Q_k, and R_k in the H_∞ estimation problem. In the Kalman filter, there is no analogy to the S_k matrix given in Equation (11.36). The Kalman filter minimizes the S_k-weighted sum of estimation-error variances for all positive definite S_k matrices (see Section 5.2). But in the H_∞ filter, we will see that the choice of S_k affects the filter gain.

The direct minimization of J_1 is not tractable, so instead we choose a performance bound and seek an estimation strategy that satisfies the threshold. That is, we will try to find an estimate $\hat{z}_k$ that results in

$$J_1 < \frac{1}{\theta} \tag{11.37}$$

where θ is our user-specified performance bound. Rearranging this equation results in

$$\begin{aligned} J &= \frac{-1}{\theta}||x_0 - \hat{x}_0||^2_{P_0^{-1}} + \sum_{k=0}^{N-1}\left[||z_k - \hat{z}_k||^2_{S_k} - \frac{1}{\theta}\left(||w_k||^2_{Q_k^{-1}} + ||v_k||^2_{R_k^{-1}}\right)\right] \\ &< 1 \end{aligned} \tag{11.38}$$

where J is defined by the above equation. The minimax problem becomes

$$J^* = \min_{\hat{z}_k} \max_{w_k, v_k, x_0} J \tag{11.39}$$

Since $z_k = L_k x_k$, we naturally choose $\hat{z}_k = L_k \hat{x}_k$ and try to find the $\hat{x}_k$ that minimizes J. This gives us the problem

$$J^* = \min_{\hat{x}_k} \max_{w_k, v_k, x_0} J \tag{11.40}$$

Nature is choosing x_0, w_k, and v_k to maximize J. But x_0, w_k, and v_k completely determine y_k, so we can replace the v_k in the minimax problem with y_k. We therefore have

$$J^* = \min_{\hat{x}_k} \max_{w_k, y_k, x_0} J \tag{11.41}$$

Since $y_k = H_k x_k + v_k$, we see that $v_k = y_k - H_k x_k$ and

$$||v_k||^2_{R_k^{-1}} = ||y_k - H_k x_k||^2_{R_k^{-1}} \tag{11.42}$$

Since $z_k = L_k x_k$ and $\hat{z}_k = L_k \hat{x}_k$, we see that

$$\begin{aligned} ||z_k - \hat{z}_k||^2_{S_k} &= (z_k - \hat{z}_k)^T S_k (z_k - \hat{z}_k) \\ &= (x_k - \hat{x}_k)^T L_k^T S_k L_k (x_k - \hat{x}_k) \\ &= ||x_k - \hat{x}_k||^2_{S_k} \end{aligned} \tag{11.43}$$

where $\bar{S}_k$ is defined as

$$\bar{S}_k = L_k^T S_k L_k \tag{11.44}$$

We substitute these results in Equation (11.38) to obtain

$$\begin{aligned} J &= \frac{-1}{\theta}||x_0 - \hat{x}_0||^2_{P_0^{-1}} + \sum_{k=0}^{N-1}\left[||x_k - \hat{x}_k||^2_{\bar{S}_k} - \frac{1}{\theta}\left(||w_k||^2_{Q_k^{-1}} + ||y_k - H_k x_k||^2_{R_k^{-1}}\right)\right] \\ &= \psi(x_0) + \sum_{k=0}^{N-1}\mathcal{L}_k \end{aligned} \tag{11.45}$$

where $\psi(x_0)$ and $\mathcal{L}_k$ are defined by the above equation. To solve the minimax problem, we will first find a stationary point of J with respect to x_0 and w_k, and then we will find a stationary point of J with respect to $\hat{x}_k$ and y_k.

11.3.1 Stationarity with respect to x_0 and w_k

The problem in this section is to maximize $J = \psi(x_0) + \sum_{k=0}^{N-1}\mathcal{L}_k$ (subject to the constraint $x_{k+1} = F_k x_k + w_k$) with respect to x_0 and w_k. This is the dynamic constrained optimization problem that we solved in Section 11.2.3. The Hamiltonian for this problem is defined as

$$\mathcal{H}_k = \mathcal{L}_k + \frac{2\lambda_{k+1}^T}{\theta}(F_k x_k + w_k) \tag{11.46}$$

where $2\lambda_{k+1}/\theta$ is the time-varying Lagrange multiplier that must be computed ($k = 0, \cdots, N-1$). Note that we have defined the Lagrange multiplier as $2\lambda_{k+1}/\theta$ instead of λ_{k+1}. This does not change the solution to the problem, it simply scales the Lagrange multiplier (in hindsight) by a constant to make the ensuing math more straightforward. From Equation (11.33) we know that the constrained stationary point of J (with respect to x_0 and w_k) is solved by the following four equations:

$$\begin{aligned} \frac{2\lambda_0^T}{\theta} + \frac{\partial\psi_0}{\partial x_0} &= 0 \\ \frac{2\lambda_N^T}{\theta} &= 0 \\ \frac{\partial\mathcal{H}_k}{\partial w_k} &= 0 \\ \frac{2\lambda_k^T}{\theta} &= \frac{\partial\mathcal{H}_k}{\partial x_k} \end{aligned} \tag{11.47}$$

From the first expression in the above equation we obtain

$$\begin{aligned} \frac{2\lambda_0}{\theta} - \frac{2}{\theta}P_0^{-1}(x_0 - \hat{x}_0) &= 0 \\ P_0\lambda_0 - x_0 + \hat{x}_0 &= 0 \\ x_0 &= \hat{x}_0 + P_0\lambda_0 \end{aligned} \tag{11.48}$$

From the second expression in Equation (11.47) we obtain

$$\lambda_N = 0 \tag{11.49}$$

From the third expression in Equation (11.47) we obtain

$$\begin{aligned} -\frac{2}{\theta}Q_k^{-1}w_k + \frac{2}{\theta}\lambda_{k+1} &= 0 \\ w_k &= Q_k\lambda_{k+1} \end{aligned} \tag{11.50}$$

This can be substituted into the process dynamics equation to obtain

$$x_{k+1} = F_k x_k + Q_k\lambda_{k+1} \tag{11.51}$$

From the fourth expression in Equation (11.47) we obtain

$$\begin{aligned} \frac{2\lambda_k}{\theta} &= 2\bar{S}_k(x_k - \hat{x}_k) + \frac{2}{\theta}H_k^T R_k^{-1}(y_k - H_k x_k) + \frac{2}{\theta}F_k^T\lambda_{k+1} \\ \lambda_k &= F_k^T\lambda_{k+1} + \theta\bar{S}_k(x_k - \hat{x}_k) + H_k^T R_k^{-1}(y_k - H_k x_k) \end{aligned} \tag{11.52}$$

At this point we have to make an assumption in order to proceed any further. From Equation (11.48) we know that $x_0 = \hat{x}_0 + P_0\lambda_0$, so we will assume that

$$x_k = \mu_k + P_k\lambda_k \tag{11.53}$$

for all k, where μ_k and P_k are some functions to be determined, with P_0 given, and the initial condition $\mu_0 = \hat{x}_0$. That is, we assume that x_k is an affine function of λ_k. This assumption may or may not turn out to be valid. We will proceed as if the assumption were true, and if our results turn out to be correct then we will know that our assumption was indeed valid. Substituting Equation (11.53) into Equation (11.51) gives

$$\mu_{k+1} + P_{k+1}\lambda_{k+1} = F_k\mu_k + F_k P_k\lambda_k + Q_k\lambda_{k+1} \tag{11.54}$$

Substituting Equation (11.53) into Equation (11.52) gives

$$\lambda_k = F_k^T\lambda_{k+1} + \theta\bar{S}_k(\mu_k + P_k\lambda_k - \hat{x}_k) + H_k^T R_k^{-1}\left[y_k - H_k(\mu_k + P_k\lambda_k)\right] \tag{11.55}$$

Rearranging this equation gives

$$\begin{aligned} &\lambda_k - \theta\bar{S}_k P_k\lambda_k + H_k^T R_k^{-1}H_k P_k\lambda_k = \\ &\quad F_k^T\lambda_{k+1} + \theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^T R_k^{-1}(y_k - H_k\mu_k) \end{aligned} \tag{11.56}$$

This can be solved for λ_k as

$$\begin{aligned} \lambda_k &= \left[I - \theta\bar{S}_k P_k + H_k^T R_k^{-1}H_k P_k\right]^{-1} \times \\ &\quad \left[F_k^T\lambda_{k+1} + \theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^T R_k^{-1}(y_k - H_k\mu_k)\right] \end{aligned} \tag{11.57}$$

Substituting this expression for λ_k into Equation (11.54) gives

$$\begin{aligned} \mu_{k+1} + P_{k+1}\lambda_{k+1} = F_k\mu_k + F_k P_k\left[I - \theta\bar{S}_k P_k + H_k^T R_k^{-1}H_k P_k\right]^{-1} \times \\ \left[F_k^T\lambda_{k+1} + \theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^T R_k^{-1}(y_k - H_k\mu_k)\right] + Q_k\lambda_{k+1} \end{aligned} \tag{11.58}$$

This equation can be rearranged as follows:

$$\begin{aligned} &\mu_{k+1} - F_k\mu_k - F_k P_k\left[I - \theta\bar{S}_k P_k + H_k^T R_k^{-1}H_k P_k\right]^{-1} \times \\ &\left[\theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^T R_k^{-1}(y_k - H_k\mu_k)\right] = \\ &\left[-P_{k+1} + F_k P_k\left[I - \theta\bar{S}_k P_k + H_k^T R_k^{-1}H_k P_k\right]^{-1}F_k^T + Q_k\right]\lambda_{k+1} \end{aligned} \tag{11.59}$$

This equation is satisfied if both sides are zero. Setting the left side of the above equation equal to zero gives

$$\begin{aligned}\mu_{k+1} &= F_k\mu_k + F_kP_k\left[I - \theta\bar{S}_kP_k + H_k^TR_k^{-1}H_kP_k\right]^{-1} \times \\ &\quad \left[\theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^TR_k^{-1}(y_k - H_k\mu_k)\right] \end{aligned} \tag{11.60}$$

with the initial condition

$$\mu_0 = \hat{x}_0 \tag{11.61}$$

Setting the right side of Equation (11.59) equal to zero gives

$$\begin{aligned}P_{k+1} &= F_kP_k\left[I - \theta\bar{S}_kP_k + H_k^TR_k^{-1}H_kP_k\right]^{-1}F_k^T + Q_k \\ &= F_k\tilde{P}_kF_k^T + Q_k \end{aligned} \tag{11.62}$$

where $\tilde{P}_k$ is defined by the above equation. That is,

$$\begin{aligned}\tilde{P}_k &= P_k\left[I - \theta\bar{S}_kP_k + H_k^TR_k^{-1}H_kP_k\right]^{-1} \\ &= \left[P_k^{-1} - \theta\bar{S}_k + H_k^TR_k^{-1}H_k\right]^{-1} \end{aligned} \tag{11.63}$$

From the above equation we see that if P_k, $\bar{S}_k$, and R_k are symmetric, then $\tilde{P}_k$ will be symmetric. We see from Equation (11.62) that if Q_k is also symmetric, then P_{k+1} will be symmetric. So if P_0, Q_k, R_k, and S_k are symmetric for all k, then $\tilde{P}_k$ and P_k will be symmetric for all k. The values of x_0 and w_k that provide a stationary point of J can be summarized as follows:

$$\begin{aligned}x_0 &= \hat{x}_0 + P_0\lambda_0 \\ w_k &= Q_k\lambda_{k+1} \\ \lambda_N &= 0 \\ \lambda_k &= \left[I - \theta\bar{S}_kP_k + H_k^TR_k^{-1}H_kP_k\right]^{-1} \times \\ &\quad \left[F_k^T\lambda_{k+1} + \theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^TR_k^{-1}(y_k - H_k\mu_k)\right] \\ P_{k+1} &= F_kP_k\left[I - \theta\bar{S}_kP_k + H_k^TR_k^{-1}H_kP_k\right]^{-1}F_k^T + Q_k \\ \mu_0 &= \hat{x}_0 \\ \mu_{k+1} &= F_k\mu_k + F_kP_k\left[I - \theta\bar{S}_kP_k + H_k^TR_k^{-1}H_kP_k\right]^{-1} \times \\ &\quad \left[\theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^TR_k^{-1}(y_k - H_k\mu_k)\right] \end{aligned} \tag{11.64}$$

The fact that we were able to find a stationary point of J shows that we were correct in our assumption that x_k was an affine function of λ_k. In the following section, given these values of x_0 and w_k, we will find the values of $\hat{x}_k$ and y_k that provide a stationary point of J.

11.3.2 Stationarity with respect to $\hat{x}$ and y

The problem in this section is to find a stationary point (with respect to $\hat{x}_k$ and y_k) of $J = \psi(x_k)|_{k=0} + \sum_{k=0}^{N-1}\mathcal{L}_k$ (subject to the constraint $x_{k+1} = F_kx_k + w_k$). This problem is solved given the fact that x_0 and w_k have already been set to their

maximizing values as described in Section 11.3.1. From Equation (11.53), and the initial condition of μ_k in Equation (11.61), we see that

$$\begin{aligned} \lambda_k &= P_k^{-1}(x_k - \mu_k) \\ \lambda_0 &= P_0^{-1}(x_0 - \hat{x}_0) \end{aligned} \tag{11.65}$$

We therefore obtain

$$\begin{aligned} ||\lambda_0||^2_{P_0} &= \lambda_0^T P_0 \lambda_0 \\ &= (x_0 - \hat{x}_0)^T P_0^{-T} P_0 P_0^{-1}(x_0 - \hat{x}_0) \\ &= (x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) \\ &= ||x_0 - \hat{x}_0||^2_{P_0^{-1}} \end{aligned} \tag{11.66}$$

Therefore, Equation (11.45) becomes

$$J = \frac{-1}{\theta}||\lambda_0||^2_{P_0} + \sum_{k=0}^{N-1}\left[||x_k - \hat{x}_k||^2_{S_k} - \frac{1}{\theta}\left(||w_k||^2_{Q_k^{-1}} + ||y_k - H_k x_k||^2_{R_k^{-1}}\right)\right] \tag{11.67}$$

Substituting for x_k from Equation (11.53) in this expression gives

$$\begin{aligned} J = &\frac{-1}{\theta}||\lambda_0||^2_{P_0} + \\ &\sum_{k=0}^{N-1}\left[||\mu_k + P_k\lambda_k - \hat{x}_k||^2_{S_k} - \frac{1}{\theta}\left(||w_k||^2_{Q_k^{-1}} + ||y_k - H_k(\mu_k + P_k\lambda_k)||^2_{R_k^{-1}}\right)\right] \end{aligned} \tag{11.68}$$

Consider the term $w_k^T Q_k^{-1} w_k$ in the above equation. Substituting for w_k from Equation (11.50) in this term gives

$$\begin{aligned} w_k^T Q_k^{-1} w_k &= \lambda_{k+1}^T Q_k^T Q_k^{-1} Q_k \lambda_{k+1} \\ &= \lambda_{k+1}^T Q_k \lambda_{k+1} \end{aligned} \tag{11.69}$$

where we have used the fact that Q_k is symmetric. Equation (11.68) can therefore be written as

$$\begin{aligned} J = &\frac{-1}{\theta}||\lambda_0||^2_{P_0} + \\ &\sum_{k=0}^{N-1}\left[||\mu_k + P_k\lambda_k - \hat{x}_k||^2_{S_k} - \frac{1}{\theta}||y_k - H_k(\mu_k + P_k\lambda_k)||^2_{R_k^{-1}}\right] - \frac{1}{\theta}\sum_{k=0}^{N-1}||\lambda_{k+1}||^2_{Q_k} \end{aligned} \tag{11.70}$$

Now we take a slight digression to notice that

$$\sum_{k=0}^{N}\lambda_k^T P_k \lambda_k - \sum_{k=0}^{N-1}\lambda_k^T P_k \lambda_k = 0 \tag{11.71}$$

The reason that this equation is correct is because from Equation (11.49) we know that $\lambda_N = 0$. Therefore, the last term in the first summation above is equal to zero

and the two summations are equal. The above equation can be written as

$$\begin{aligned} 0 &= \lambda_0^T P_0 \lambda_0 + \sum_{k=1}^{N} \lambda_k^T P_k \lambda_k - \sum_{k=0}^{N-1} \lambda_k^T P_k \lambda_k \\ &= \lambda_0^T P_0 \lambda_0 + \sum_{k=0}^{N-1} \lambda_{k+1}^T P_{k+1} \lambda_{k+1} - \sum_{k=0}^{N-1} \lambda_k^T P_k \lambda_k \\ &= \frac{-1}{\theta} ||\lambda_0||_{P_0}^2 - \frac{1}{\theta} \sum_{k=0}^{N-1} (\lambda_{k+1}^T P_{k+1} \lambda_{k+1} - \lambda_k^T P_k \lambda_k) \end{aligned} \tag{11.72}$$

We can subtract this zero term to the cost function of Equation (11.70) to obtain

$$\begin{aligned} J &= \sum_{k=0}^{N-1} \Big[||\mu_k + P_k \lambda_k - \hat{x}_k||_{\bar{S}_k}^2 - \\ &\quad \frac{1}{\theta} ||\lambda_{k+1}||_{Q_k}^2 + \frac{1}{\theta} (\lambda_{k+1}^T P_{k+1} \lambda_{k+1} - \lambda_k^T P_k \lambda_k) - \frac{1}{\theta} ||y_k - H_k(\mu_k + P_k \lambda_k)||_{R_k^{-1}}^2 \Big] \\ &= \sum_{k=0}^{N-1} \Big[(\mu_k - \hat{x}_k)^T \bar{S}_k (\mu_k - \hat{x}_k) + 2(\mu_k - \hat{x}_k)^T \bar{S}_k P_k \lambda_k + \lambda_k^T P_k \bar{S}_k P_k \lambda_k + \\ &\quad \frac{1}{\theta} \lambda_{k+1}^T (P_{k+1} - Q_k) \lambda_{k+1} - \frac{1}{\theta} \lambda_k^T P_k \lambda_k - \frac{1}{\theta} (y_k - H_k \mu_k)^T R_k^{-1} (y_k - H_k \mu_k) + \\ &\quad \frac{2}{\theta} (y_k - H_k \mu_k)^T R_k^{-1} H_k P_k \lambda_k - \frac{1}{\theta} \lambda_k^T P_k H_k^T R_k^{-1} H_k P_k \lambda_k \Big] \end{aligned} \tag{11.73}$$

Now we consider the term $\lambda_{k+1}^T (P_{k+1} - Q_k) \lambda_{k+1}$ in the above expression. Substituting for P_{k+1} from Equation (11.62) in this term gives

$$\begin{aligned} \lambda_{k+1}^T (P_{k+1} - Q_k) \lambda_{k+1} &= \lambda_{k+1}^T (Q_k + F_k \tilde{P}_k F_k^T - Q_k) \lambda_{k+1} \\ &= \lambda_{k+1}^T F_k \tilde{P}_k F_k^T \lambda_{k+1} \end{aligned} \tag{11.74}$$

But from Equation (11.55) we see that

$$F_k^T \lambda_{k+1} = \lambda_k - \theta \bar{S}_k (\mu_k + P_k \lambda_k - \hat{x}_k) - H_k^T R_k^{-1} [y_k - H_k(\mu_k + P_k \lambda_k)] \tag{11.75}$$

Substituting this expression for $F_k^T \lambda_{k+1}$ into Equation (11.74) gives

$$\begin{aligned} &\lambda_{k+1}^T (P_{k+1} - Q_k) \lambda_{k+1} \\ &= \left\{ \lambda_k - \theta \bar{S}_k (\mu_k + P_k \lambda_k - \hat{x}_k) - H_k^T R_k^{-1} [y_k - H_k(\mu_k + P_k \lambda_k)] \right\}^T \\ &\quad \tilde{P}_k \left\{ \lambda_k - \theta \bar{S}_k (\mu_k + P_k \lambda_k - \hat{x}_k) - H_k^T R_k^{-1} [y_k - H_k(\mu_k + P_k \lambda_k)] \right\} \\ &= \left\{ \lambda_k^T (I - \theta P_k \bar{S}_k + P_k H_k^T R_k^{-1} H_k) - \theta (\mu_k - \hat{x}_k)^T \bar{S}_k - \right. \\ &\quad (y_k - H_k \mu_k)^T R_k^{-1} H_k \Big\} \tilde{P}_k \left\{ \lambda_k^T (I - \theta P_k \bar{S}_k + P_k H_k^T R_k^{-1} H_k) - \right. \\ &\quad \left. \theta (\mu_k - \hat{x}_k)^T \bar{S}_k - (y_k - H_k \mu_k)^T R_k^{-1} H_k \right\}^T \end{aligned} \tag{11.76}$$

Now note from Equation (11.63) that $(I - \theta P_k \bar{S}_k + P_k H_k^T R_k^{-1} H_k) = P_k \tilde{P}_k^{-1}$. Making this substitution in the above equation gives the following.

$$
\begin{aligned}
&\lambda_{k+1}^T(P_{k+1}-Q_k)\lambda_{k+1} \\
&= \left\{\lambda_k^T P_k \tilde{P}_k^{-1} - \theta(\mu_k - \hat{x}_k)^T \bar{S}_k - (y_k - H_k\mu_k)^T R_k^{-1} H_k\right\} \\
&\quad \tilde{P}_k \left\{\lambda_k^T P_k \tilde{P}_k^{-1} - \theta(\mu_k - \hat{x}_k)^T \bar{S}_k - (y_k - H_k\mu_k)^T R_k^{-1} H_k\right\}^T \\
&= \lambda_k^T P_k \tilde{P}_k^{-1} P_k \lambda_k - \theta(\mu_k - \hat{x}_k)^T \bar{S}_k P_k \lambda_k - (y_k - H_k\mu_k)^T R_k^{-1} H_k P_k \lambda_k - \\
&\quad \theta\lambda_k P_k \bar{S}_k(\mu_k - \hat{x}_k) + \theta^2(\mu_k - \hat{x}_k)^T \bar{S}_k \tilde{P}_k \bar{S}_k(\mu_k - \hat{x}_k) + \\
&\quad \theta(y_k - H_k\mu_k)^T R_k^{-1} H_k \tilde{P}_k \bar{S}_k(\mu_k - \hat{x}_k) - \lambda_k^T P_k H_k^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad \theta(\mu_k - \hat{x}_k)^T \bar{S}_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad (y_k - H_k\mu_k)^T R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k)
\end{aligned} \tag{11.77}
$$

Notice that the above expression is a scalar. That means that each term on the right side is a scalar, which means that each term is equal to its transpose. For example, consider the second term on the right side. Since it is a scalar, we see that $\theta(\mu_k - \hat{x}_k)^T \bar{S}_k P_k \lambda_k = \theta\lambda_k^T P_k \bar{S}_k(\mu_k - \hat{x}_k)$. (We have used the fact that P_k and $\bar{S}_k$ are symmetric, and θ is a scalar.) Equation (11.77) can therefore be written as

$$
\begin{aligned}
&\lambda_{k+1}^T(P_{k+1}-Q_k)\lambda_{k+1} \\
&= \lambda_k^T P_k \tilde{P}_k^{-1} P_k \lambda_k - 2\theta(\mu_k - \hat{x}_k)^T \bar{S}_k P_k \lambda_k - \\
&\quad 2(y_k - H_k\mu_k)^T R_k^{-1} H_k P_k \lambda_k + \theta^2(\mu_k - \hat{x}_k)^T \bar{S}_k \tilde{P}_k \bar{S}_k(\mu_k - \hat{x}_k) + \\
&\quad 2\theta(\mu_k - \hat{x}_k)^T \bar{S}_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad (y_k - H_k\mu_k)^T R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k)
\end{aligned} \tag{11.78}
$$

Now note from Equation (11.63) that

$$
\begin{aligned}
\tilde{P}_k^{-1} &= \left[I - \theta\bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k\right] P_k^{-1} \\
&= P_k^{-1}\left[P_k^{-1} - \theta\bar{S}_k + H_k^T R_k^{-1} H_k\right] P_k^{-1} \\
&= P_k^{-1}\left[I - P_k\theta\bar{S}_k + P_k H_k^T R_k^{-1} H_k\right]
\end{aligned} \tag{11.79}
$$

We therefore see that

$$
\begin{aligned}
\lambda_k^T P_k \tilde{P}_k^{-1} P_k \lambda_k &= \lambda_k^T\left[I - \theta P_k \bar{S}_k + P_k H_k^T R_k^{-1} H_k\right] P_k \lambda_k \\
&= \lambda_k^T P_k \lambda_k - \theta\lambda_k^T P_k \bar{S}_k P_k \lambda_k + \lambda_k^T P_k H_k^T R_k^{-1} H_k P_k \lambda_k
\end{aligned} \tag{11.80}
$$

Substituting this into Equation (11.78) gives

$$
\begin{aligned}
&\lambda_{k+1}^T(P_{k+1}-Q_k)\lambda_{k+1} \\
&= \lambda_k^T P_k \lambda_k - \theta\lambda_k^T P_k \bar{S}_k P_k \lambda_k + \lambda_k^T P_k H_k^T R_k^{-1} H_k P_k \lambda_k - \\
&\quad 2\theta(\mu_k - \hat{x}_k)^T \bar{S}_k P_k \lambda_k - 2(y_k - H_k\mu_k)^T R_k^{-1} H_k P_k \lambda_k + \\
&\quad \theta^2(\mu_k - \hat{x}_k)^T \bar{S}_k \tilde{P}_k \bar{S}_k(\mu_k - \hat{x}_k) + 2\theta(\mu_k - \hat{x}_k)^T \bar{S}_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad (y_k - H_k\mu_k)^T R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k)
\end{aligned} \tag{11.81}
$$

Substituting this equation for $\lambda_{k+1}^T(P_{k+1} - Q_k)\lambda_{k+1}$ into Equation (11.73) gives the following.

$$
\begin{aligned}
J &= \sum_{k=0}^{N-1} \Big[(\mu_k - \hat{x}_k)^T \bar{S}_k (\mu_k - \hat{x}_k) - \frac{1}{\theta}(y_k - H_k\mu_k)^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad \theta(\mu_k - \hat{x})_k^T \bar{S}_k \tilde{P}_k \bar{S}_k (\mu_k - \hat{x}_k) + 2(\mu_k - \hat{x}_k)^T \bar{S}_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad \frac{1}{\theta}(y_k - H_k\mu_k)^T R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k) \Big] \\
&= \sum_{k=0}^{N-1} \Big[(\mu_k - \hat{x}_k)^T (\bar{S}_k + \theta \bar{S}_k \tilde{P}_k \bar{S}_k)(\mu_k - \hat{x}_k) + \\
&\quad 2(\mu_k - \hat{x}_k)^T \bar{S}_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad \frac{1}{\theta}(y_k - H_k\mu_k)^T (R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1} - R_k^{-1})(y_k - H_k\mu_k) \Big]
\end{aligned} \tag{11.82}
$$

Now recall our original objective: we are trying to find the stationary point of J with respect to $\hat{x}_k$ and y_k. If we take the partial derivative of the above expression for J with respect to $\hat{x}_k$ and y_k and set them equal to 0 we obtain

$$
\begin{aligned}
\frac{\partial J}{\partial \hat{x}_k} &= 2(\bar{S}_k + \theta \bar{S}_k \tilde{P}_k \bar{S}_k)(\hat{x}_k - \mu_k) + 2\bar{S}_k \tilde{P}_k H_k^T R_k^{-1}(H_k\mu_k - y_k) \\
&= 0 \\
\frac{\partial J}{\partial y_k} &= \frac{2}{\theta}(R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1} - R_k^{-1})(y_k - H_k\mu_k) + 2R_k^{-1} H_k \tilde{P}_k \bar{S}_k (\mu_k - \hat{x}_k) \\
&= 0
\end{aligned} \tag{11.83}
$$

These equations are clearly satisfied for the following values of $\hat{x}_k$ and y_k:

$$
\begin{aligned}
\hat{x}_k &= \mu_k \\
y_k &= H_k\mu_k
\end{aligned} \tag{11.84}
$$

These are the extremizing values of $\hat{x}_k$ and y_k. However, we still are not sure if these extremizing values give a local minimum or maximum of J. Recall that the second derivative of J tells us what kind of stationary point we have. If the second derivative is positive definite, then our stationary point is a minimum. If the second derivative is negative definite, then our stationary point is a maximum. If the second derivative has both positive and negative eigenvalues, then our stationary point is a saddle point. The second derivative of J with respect to $\hat{x}_k$ can be computed as

$$
\frac{\partial^2 J}{\partial \hat{x}_k^2} = 2(\bar{S}_k + \theta \bar{S}_k \tilde{P}_k \bar{S}_k) \tag{11.85}
$$

Our $\hat{x}_k$ will therefore be a minimizing value of J if $(\bar{S}_k + \theta \bar{S}_k \tilde{P}_k \bar{S}_k)$ is positive definite. The value of S_k chosen for use in Equation (11.36) should always be positive definite, which means that $\bar{S}_k$ defined in Equation (11.44) will be positive definite. This means that our $\hat{x}_k$ will be a minimizing value of J if $\tilde{P}_k$ is positive definite.

So, from the definition of $\tilde{P}_k$ in Equation (11.63), the condition required for $\hat{x}_k$ to minimize J is that $\left(P_k^{-1} - \theta \bar{S}_k + H_k^T R_k^{-1} H_k\right)^{-1}$ be positive definite. This is

equivalent to requiring that $(P_k^{-1} - \theta\bar{S}_k + H_k^T R_k^{-1} H_k)$ be positive definite. The individual terms in this expression are always positive definite [note in particular from Equation (11.62) that P_k will be positive definite if $\tilde{P}_k$ is positive definite]. So the condition for $\hat{x}_k$ to minimize J is that $\theta\bar{S}_k$ be "small enough" so that $(P_k^{-1} - \theta\bar{S}_k + H_k^T R_k^{-1} H_k)$ is positive definite. Requiring that $\theta\bar{S}_k$ be small can be accomplished three different ways.

1. $\theta\bar{S}_k$ will be small if θ is small. This means that the performance requirement specified in Equation (11.37) is not too stringent. As long as our performance requirement is not too stringent then the problem will have a solution. If, however, the performance requirement is too stringent (i.e., θ is large) then the problem will not have a solution.

2. $\theta\bar{S}_k$ will be small if L_k is small. This statement is based on the relationship between $\bar{S}_k$ and L_k as shown in Equation (11.44). From Equation (11.36) we see that the numerator of the cost function is given as $(x_k - \hat{x}_k)^T L_k^T S_k L_k (x_k - \hat{x}_k)$. So if L_k is small we see that the numerator of the cost function will be small, which means that it will be easier to minimize the cost function. If, however, L_k is too large, then the problem will not have a solution.

3. $\theta\bar{S}_k$ will be small if S_k is small. This statement is based on the relationship between $\bar{S}_k$ and S_k as shown in Equation (11.44). From Equation (11.36) we see that the numerator of the cost function is given as $(x_k - \hat{x}_k)^T L_k^T S_k L_k (x_k - \hat{x}_k)$. So if S_k is small we see that the numerator of the cost function will be small, which means that it will be easier to minimize the cost function. If, however, S_k is too large, then the problem will not have a solution.

Note from Equation (11.62) that the positive definiteness of $\tilde{P}_k$ implies the positive definiteness of P_{k+1}. Therefore, if P_0 is positive definite (per our original problem statement), and $\tilde{P}_k$ is positive definite for all k, then P_k will also be positive definite for all k.

It is also academically interesting (though of questionable utility) to note the conditions under which the y_k that we found in Equation (11.84) will be a maximizing value of J. (Recall that y_k is chosen by nature, our adversary, to maximize the cost function.) The second derivative of J with respect to y_k can be computed as

$$\begin{aligned} \frac{\partial^2 J}{\partial y_k^2} &= \frac{2}{\theta}(R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1} - R_k^{-1}) \\ &= \frac{2}{\theta} R_k^{-1}(H_k \tilde{P}_k H_k^T - R_k) R_k^{-1} \end{aligned} \tag{11.86}$$

R_k and R_k^{-1}, specified by the user as part of the problem statement in Equation (11.36), should always be positive definite. So the second derivative above will be negative definite (which means that y_k will be a maximizing value of J) if $(R_k - H_k \tilde{P}_k H_k^T)$ is positive definite. This requirement can be satisfied in two ways.

1. $(R_k - H_k \tilde{P}_k H_k^T)$ will be positive definite if R_k is large enough. A large value of R_k means that the denominator of the cost function of Equation (11.36) will be small, which means that the cost function will be large. A large cost function value is easier to maximize and will therefore tend to have a

maximizing value for y_k. Also note that the designer typically chooses R_k to be proportional to the magnitude of the measurement noise. If the user knows that the measurement noise is large, then R_k will be large, which again will result in a problem with a maximizing value for y_k. In other words, nature will be better able to maximize the cost function if the measurement noise is large.

2. $(R_k - H_k \tilde{P}_k H_k^T)$ will be positive definite if H_k is small enough. If H_k becomes smaller, that means that the measurement noise becomes larger relative to the size of the measurements, as seen in Equation (11.34). In other words, a small value of H_k means a smaller signal-to-noise ratio for the measurements. A small signal-to-noise ratio gives nature a better opportunity to find a maximizing value of y_k.

Of course, we are not really interested in finding a maximizing value of y_k. Our goal was to find the minimizing value of x_k. The H_∞ filter algorithm can be summarized as follows.

The discrete-time H_∞ filter

1. The system equations are given as

$$\begin{aligned} x_{k+1} &= F_k x_k + w_k \\ y_k &= H_k x_k + v_k \\ z_k &= L_k x_k \end{aligned} \tag{11.87}$$

where w_k and v_k are noise terms, and our goal is to estimate z_k.

2. The cost function is given as

$$J_1 = \frac{\sum_{k=0}^{N-1} ||z_k - \hat{z}_k||^2_{S_k}}{||x_0 - \hat{x}_0||^2_{P_0^{-1}} + \sum_{k=0}^{N-1} \left(||w_k||^2_{Q_k^{-1}} + ||v_k||^2_{R_k^{-1}} \right)} \tag{11.88}$$

where P_0, Q_k, R_k, and S_k are symmetric, positive definite matrices chosen by the engineer based on the specific problem.

3. The cost function can be made to be less than $1/\theta$ (a user-specified bound) with the following estimation strategy, which is derived from Equations (11.44), (11.60), (11.62), and (11.84):

$$\begin{aligned} \bar{S}_k &= L_k^T S_k L_k \\ K_k &= P_k \left[I - \theta \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k \right]^{-1} H_k^T R_k^{-1} \\ \hat{x}_{k+1} &= F_k \hat{x}_k + F_k K_k (y_k - H_k \hat{x}_k) \\ P_{k+1} &= F_k P_k \left[I - \theta \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k \right]^{-1} F_k^T + Q_k \end{aligned} \tag{11.89}$$

4. The following condition must hold at each time step k in order for the above estimator to be a solution to the problem:

$$P_k^{-1} - \theta \bar{S}_k + H_k^T R_k^{-1} H_k > 0 \tag{11.90}$$

11.3.3 A comparison of the Kalman and H_∞ filters

Comparing the Kalman filter in Equation (11.12) and the H_∞ filter in Equation (11.89) reveals some fascinating connections. For instance, in the H_∞ filter, Q_k, R_k, and P_0 are design parameters chosen by the user based on *a priori* knowledge of the magnitude of the process disturbance w_k, the measurement disturbance v_k, and the initial estimation error $(x_0 - \hat{x}_0)$. In the Kalman filter, w_k, v_k, and $(x_0 - \hat{x}_0)$ are zero-mean, and Q_k, R_k, and P_0 are their respective covariances.

Now suppose we use $L_k = S_k = I$ in the H_∞ filter. That is, we are interested in estimating the entire state, and we want to weight all of the estimation errors equally in the cost function. If we use $\theta = 0$ then the H_∞ filter reduces to the Kalman filter (assuming Q_k, R_k, and P_0 are chosen as above). This provides an interesting interpretation of the Kalman filter; that is, the Kalman filter is the minimax filter in the case that the performance bound in Equation (11.36) is set equal to ∞. We see that although the Kalman filter minimizes the variance of the estimation error (as discussed in Section 5.2), it does not provide any guarantee as far as limiting the worst-case estimation error. That is, it does not guarantee any bound for the cost function of Equation (11.36).

The Kalman and H_∞ filter equations have an interesting difference. If we want to estimate a linear combination of states using the Kalman filter, the estimator is the same regardless of the linear combination that we want to estimate. That is, if we want to estimate $L_k x_k$ using the Kalman filter, the answer is the same regardless of the L_k matrix that we choose. However, in the H_∞ approach, the resulting filter depends strongly on L_k and the particular linear combination of states that we want to estimate.

Note that the H_∞ filter of Equation (11.89) is identical to the Kalman filter except for subtraction of the term $\theta \bar{S}_k P_k$ in the K_k and P_{k+1} equations. Recall from Section 5.5 that the Kalman filter can be made more robust to unmodeled noise and unmodeled dynamics by artificially increasing Q_k in the Kalman filter equations. This results in a larger covariance P_k, which in turn results in a larger gain K_k. From Equation (11.89) we can see that subtracting $\theta \bar{S}_k P_k$ on the right side of the P_{k+1} equation tends to make P_{k+1} larger (since the subtraction is inside a matrix inverse operation). Similarly, subtracting $\theta \bar{S}_k P_k$ on the right side of the K_k equation tends to make K_k larger. Increasing Q_k in the Kalman filter is conceptually the same as increasing P_k and K_k. Therefore, the H_∞ filter equations make intuitive sense when compared with the Kalman filter equations. The H_∞ filter is a worst-case filter in the sense that it assumes that w_k, v_k, and x_0 will be chosen by nature to maximize the cost function. The H_∞ filter is therefore robust by design. Comparing the H_∞ filter with the Kalman filter, we can see that the H_∞ filter is simply a robust version of the Kalman filter. When we robustified the Kalman filter in Section 5.5 to add tolerance to unmodeled noise and dynamics, we did not derive an optimal way to increase Q_k. However, H_∞ filter theory shows us the optimal way to robustify the Kalman filter.

11.3.4 Steady-state H_∞ filtering

If the underlying system and the design parameters are time-invariant, then it may be possible to obtain a steady-state solution to the H_∞ filtering problem. Suppose

that our system is given as

$$\begin{aligned} x_{k+1} &= Fx_k + w_k \\ y_k &= Hx_k + v_k \\ z_k &= Lx_k \end{aligned} \tag{11.91}$$

where w_k and v_k are noise terms. Our goal is to estimate z_k such that

$$\lim_{N\to\infty} \frac{\sum_{k=0}^{N-1} ||z_k - \hat{z}_k||_S^2}{\sum_{k=0}^{N-1} \left(||w_k||_{Q^{-1}}^2 + ||v_k||_{R^{-1}}^2 \right)} < \frac{1}{\theta} \tag{11.92}$$

where Q, R, and S are symmetric positive definite matrices chosen by the engineer based on the specific problem. The steady-state filter of Equation (11.89) becomes

$$\begin{aligned} \bar{S} &= L^T S L \\ K &= P\left[I - \theta\bar{S}P + H^T R^{-1} H P\right]^{-1} H^T R^{-1} \\ \hat{x}_{k+1} &= F\hat{x}_k + FK_k(y_k - H\hat{x}_k) \\ P &= FP\left[I - \theta\bar{S}P + H^T R^{-1} H P\right]^{-1} F^T + Q \end{aligned} \tag{11.93}$$

The following condition must hold in order for the above estimator to be a solution to the problem:

$$P^{-1} - \theta\bar{S} + H^T R^{-1} H > 0 \tag{11.94}$$

If θ, L, R, or S is too large, or if H is too small, then the H_∞ estimator will not have a solution. Note that the expression for P in Equation (11.93) can be written as

$$P = F\left[P^{-1} - \theta\bar{S} + H^T R^{-1} H\right]^{-1} F^T + Q \tag{11.95}$$

Applying the matrix inversion lemma to the inverse in the above expression gives

$$\begin{aligned} P &= F\left\{P - P\left[(H^T R^{-1} H - \theta\bar{S})^{-1} + P\right]^{-1} P\right\} F^T + Q \\ &= FPF^T - FP\left[(H^T R^{-1} H - \theta\bar{S})^{-1} + P\right]^{-1} PF^T + Q \end{aligned} \tag{11.96}$$

This is a discrete-time algebraic Riccati equation that can be solved with control system software.[2] If control system software is not available, then the algebraic Riccati equation can be solved by numerically iterating the discrete-time Riccati equation of Equation (11.89) until it converges to a steady-state value. The steady-state filter is much easier to implement in a system in which real-time computational effort or code size is a serious consideration. The disadvantage of the steady-state filter is that (theoretically) it does not perform as well as the time-varying filter. However, the reduced performance that is seen in the steady-state filter is often a small fraction of the optimal performance, whereas the computational savings can be significant.

[2]For example, in MATLAB's Control System Toolbox we can use the command DARE($F^T, I, Q, (H^T R^{-1} H - \theta\bar{S})^{-1}$.

■ EXAMPLE 11.2

Suppose we are trying to estimate a randomly varying scalar on the basis of noisy measurements. We have the scalar system

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + v_k \\ z_k &= x_k \end{aligned} \tag{11.97}$$

This system could describe our attempt to estimate a noisy voltage. The voltage is essentially constant, but it is subject to random fluctuations, hence the noise term w_k in the process equation. Our measurement of the voltage is also subject to noise or instrument bias, hence the noise term v_k in the measurement equation. We see in this example that $F = H = L = 1$. Further suppose that $Q = R = S = 1$ in the cost function of Equation (11.88). Then the discrete-time Riccati equation associated with the H_∞ filter equations becomes

$$\begin{aligned} P_{k+1} &= F_k P_k \left[I - \theta \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k\right]^{-1} F_k^T + Q_k \\ &= P_k \left[1 - \theta P_k + P_k\right]^{-1} + 1 \end{aligned} \tag{11.98}$$

This can be solved numerically or analytically as a function of time for a given θ to give P_k, and then the H_∞ gain can be obtained as

$$\begin{aligned} K_k &= P_k \left[I - \theta \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k\right]^{-1} H_k^T R_k^{-1} \\ &= P_k \left[1 - \theta P_k + P_k\right]^{-1} \end{aligned} \tag{11.99}$$

We can set $P_{k+1} = P_k$ in Equation (11.98) to obtain the steady-state solution for P_k. This gives

$$\begin{aligned} P &= P(1 - \theta P + P)^{-1} + 1 \\ P(1 - \theta P + P) &= P + (1 - \theta P + P) \\ (1-\theta)P^2 + (\theta - 1)P - 1 &= 0 \\ P &= \frac{1 - \theta \pm \sqrt{(\theta-1)(\theta-5)}}{2(1-\theta)} \end{aligned} \tag{11.100}$$

As we discussed earlier, in order for this value of P to be a solution to the H_∞ estimation problem, P must be positive definite. The first solution for P is positive if $\theta < 1$, and both solutions for P are positive if $\theta \geq 5$. Another condition for the solution of the H_∞ estimation problem is that

$$\begin{aligned} P^{-1} - \theta \bar{S} + H^T R^{-1} H &> 0 \\ P^{-1} - \theta + 1 &> 0 \end{aligned} \tag{11.101}$$

If $\theta < 1$ then the first solution for P satisfies this bound. However, if $\theta \geq 5$, then neither solution for P satisfies this bound. Combining this data shows that the H_∞ estimator problem has a solution for $\theta < 1$. Every H_∞ estimator problem will have a solution for θ less than some upper bound because of the nature of the cost function.

For a general estimator gain K the estimate can be written as

$$\begin{aligned} \hat{x}_{k+1} &= F\hat{x}_k + FK(y_k - H_k\hat{x}_k) \\ &= (1-K)\hat{x}_k + Ky_k \end{aligned} \tag{11.102}$$

If we choose $\theta = 1/2$, then we obtain $P = 2$ and $K = 1$. As seen from the above equation, this results in $\hat{x}_{k+1} = y_k$. In other words, the estimator ignores the previous estimate and simply sets the estimate equal to the previous measurement. As θ increases toward 1, P increases above 2 and approaches ∞, and the estimator gain K increases greater than 1 and also approaches ∞. In this case, the estimator will actually place a negative weight on the previous estimate and compensate by placing additional weight on the measurement. If θ increases too much (gets too close to 1) then the estimator gain K will be greater than 2 and the H_∞ estimator will be unstable. It is always a good idea to check the stability of your H_∞ filter. If the filter is unstable then you should probably decrease θ to obtain a stable filter. As θ decreases below $1/2$, P decreases below 2 and the gain K decreases below 1. In this case, the estimator balances the relative weight placed on the previous estimate and the measurement.

A Kalman filter to estimate x_k is equivalent to an H_∞ filter with $\theta = 0$. In this case, we obtain the positive definite solution of the steady-state Riccati equation as $P = (1+\sqrt{5})/2$. This gives a steady-state estimator gain $K = (1+\sqrt{5})/(3+\sqrt{5}) = (\sqrt{5}-1)/2 \approx 0.62$. The Kalman filter gain is smaller than the H_∞ filter gain for $\theta > 0$, which means that the Kalman filter relies less on measurements and more on the system model. The Kalman filter gives an optimal estimate if the model and the noise statistics are known, but it may undervalue the measurements if there are errors in the system model or the assumed noise statistics.

Figure 11.2 shows the true state x_k and the estimate $\hat{x}_k$ when the steady-state Kalman and H_∞ filters are used to estimate the state. The H_∞ filter was designed with $\theta = 1/3$, which gave a filter gain $K = (3+3\sqrt{7})/(8+2\sqrt{7}) \approx 0.82$. The disturbances w_k and v_k were both normally distributed zero-mean white noise sequences with standard deviations equal to 10. The performance of the two filters is very similar. The RMS estimation error of the Kalman filter is 3.6 and the RMS estimation error of the H_∞ filter is 4.1. As expected, the Kalman filter performs better than the H_∞ filter. However, suppose that the process noise has a mean of 10. Figure 11.3 shows the performance of the filters for this situation. In this case the H_∞ filter performs better. The RMS estimation error of the Kalman filter is 15.6 and the RMS estimation error of the H_∞ filter is 12.0.

If we choose $\theta = 1/10$ then we obtain $P = 5/3$ and $K = 2/3$. As θ gets smaller, the H_∞ estimator gain gets closer and closer to the Kalman filter gain.

$\nabla\nabla\nabla$

11.3.5 The transfer function bound of the H_∞ filter

In this section, we show that the steady-state H_∞ filter derived in the previous section bounds the transfer function from the noise to the estimation error, if Q,

Figure 11.2 Example 11.2 results. Kalman and H_∞ filter peformance when the noise statistics are known. The Kalman gain is 0.62 and the H_∞ gain is 0.82. The Kalman filter performs about 12% better than the H_∞ filter.

Figure 11.3 Example 11.2 results. Kalman and H_∞ filter peformance when the process noise is biased. The Kalman gain is 0.62 and the H_∞ gain is 0.82. The H_∞ filter performs about 23% better than the Kalman filter.

R, and S are all identity matrices. Recall that the two-norm of a column vector x is defined as

$$||x||_2^2 = x^T x \tag{11.103}$$

Now suppose we have a time-varying vector $x_0, x_1, x_2, \cdots$. The signal two-norm of x is defined as

$$||x||_2^2 = \sum_{k=0}^{\infty} ||x_k||_2^2 \tag{11.104}$$

That is, the square of the signal two-norm is the sum of all of the squares of the vector two-norms that are taken at each time step.[3] Now suppose that we have a system with input u and output x, and the transfer function is $G(z)$. If the input u is comprised entirely of signals at the frequency ω and the sample time of the system is T, then we define the phase of u as $\phi = T\omega$. In this case the maximum gain from u to x is determined as

$$\sup_{u \neq 0} \frac{||x||_2}{||u||_2} = \sigma_1 \left[G\left(e^{j\phi}\right) \right] \tag{11.105}$$

where $\sigma_1(G)$ is the largest singular value of the matrix G. If u can be comprised of an arbitrary mix of frequencies, then the maximum gain from u to x is determined as follows:

$$\begin{aligned} \sup_{\phi} \frac{||x||_2}{||u||_2} &= \sup_{\phi} \sigma_1 \left[G\left(e^{j\phi}\right) \right] \\ &= ||G||_\infty \end{aligned} \tag{11.106}$$

The above equation defines $||G||_\infty$, which is the infinity-norm of the system that has the transfer function $G(z)$.[4]

Now consider Equation (11.92), the cost function that is bounded by the steady-state H_∞ filter:

$$J = \lim_{N \to \infty} \frac{\sum_{k=0}^{N-1} ||z_k - \hat{z}_k||_S^2}{\sum_{k=0}^{N-1} \left(||w_k||_{Q^{-1}}^2 + ||v_k||_{R^{-1}}^2 \right)} \tag{11.107}$$

If Q, R, and S are all equal to identity matrices, then

$$J = \lim_{N \to \infty} \frac{\sum_{k=0}^{N-1} ||z_k - \hat{z}_k||_2^2}{\sum_{k=0}^{N-1} \left(||w_k||_2^2 + ||v_k||_2^2 \right)} \tag{11.108}$$

Since the H_∞ filter makes this scalar less than $1/\theta$ for all w_k and v_k, we can write

$$\begin{aligned} ||G_{\tilde{z}e}||_\infty^2 &= \sup_{\phi} \frac{||z - \hat{z}||_2^2}{||w||_2^2 + ||v||_2^2} \\ &\leq \frac{1}{\theta} \end{aligned} \tag{11.109}$$

where we have defined $\tilde{z} = z - \hat{z}$, $e^T = \begin{bmatrix} w^T & v^T \end{bmatrix}^T$, and $G_{\tilde{z}e}$ is the system that has e as its input and $\tilde{z}$ as its output. We see that the steady-state H_∞ filter bounds the infinity-norm (i.e., the maximum gain) from the combined disturbances w and v to the estimation error $\tilde{z}$, if Q, R, and S are all identity matrices. Further information about the computation of infinity-norms and related issues can be found in [Bur99].

[3]Note that this definition means that many signals have unbounded signal two-norms. The signal two-norm can also be defined as the sum from $k = 0$ to a finite limit $k = N$.

[4]Note that the infinity-norm of a matrix has a definition that is different than the infinity-norm of a system. In general, the expression $||G||_\infty$ could refer either to the matrix infinity-norm or the system infinity-norm. The meaning needs to be inferred from the context unless it is explicitly stated.

■ EXAMPLE 11.3

Consider the system and filter discussed in Example 11.2:

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + v_k \\ \hat{x}_{k+1} &= (1-K)\hat{x}_k + Ky_k \end{aligned} \tag{11.110}$$

The estimation error can be computed as

$$\begin{aligned} \tilde{x}_{k+1} &= x_{k+1} - \hat{x}_{k+1} \\ &= (1-K)\tilde{x}_k + w_k - Kv_k \end{aligned} \tag{11.111}$$

Taking the z-transform of this equation gives

$$\begin{aligned} z\tilde{X}(z) &= (1-K)\tilde{X}(z) + W(z) - KV(z) \\ \tilde{X}(z) &= \frac{1}{z-1+K}\begin{bmatrix} 1 & -K \end{bmatrix}\begin{bmatrix} W(z) \\ V(z) \end{bmatrix} \\ &= G(z)\begin{bmatrix} W(z) \\ V(z) \end{bmatrix} \end{aligned} \tag{11.112}$$

$G(z)$, the transfer function from w_k and v_k to $\tilde{x}_k$, is a 2×1 matrix. This matrix has one singular value, which is computed as

$$\begin{aligned} \sigma^2(G) &= \lambda_{\max}\left[G(e^{j\phi})G^H(e^{j\phi})\right] \\ &= \frac{1+K^2}{(e^{j\phi}-1+K)(e^{-j\phi}-1+K)} \\ &= \frac{1+K^2}{K^2+2(K-1)(\cos\phi-1)} \end{aligned} \tag{11.113}$$

The supremum of this expression occurs at $\phi = 0$ when $K \leq 1$, so

$$\begin{aligned} ||G||_\infty^2 &= \sup_\phi \sigma^2\left[G\left(e^{j\phi}\right)\right] \\ &= \frac{1+K^2}{K^2} \end{aligned} \tag{11.114}$$

Recall from Example 11.2 that $\theta = 1/2$ resulted in $K = 1$. In this case, the above expression indicates that $||G||_\infty^2 = 2 \leq 1/\theta = 2$. In this case, the infinity-norm bound specified by θ is exact. Also recall from Example 11.2 that $\theta = 1/10$ resulted in $K = 2/3$. In this case, the above expression indicates that $||G||_\infty^2 = 13/4 \leq 1/\theta = 10$. In this case, the infinity-norm bound specified by θ is quite conservative.

Note that as K increases, the infinity-norm from the noise to the estimation error decreases. However, the estimator also is unstable for $K > 1$. So even though large K reduces the infinity-norm of the estimator, it gives poor results. In other words, just because the effect of the noise on the estimation error is small does not necessarily prove that the estimator is good. For example, we could set the estimate $\hat{x}_k = \infty$ for all k. In that case, the noise

will have zero effect on the estimation error because the estimation error will be infinite regardless of the noise value. However, the estimate will obviously be poor. This example shows the importance of balancing H_∞ performance with other performance criteria.

$\nabla\nabla\nabla$

11.4 THE CONTINUOUS-TIME H_∞ FILTER

The methods of the earlier sections can also be used to derive a continuous-time H_∞ filter, as shown in [Rhe89, Ban91, Ban92]. In this section we consider the continuous-time system

$$\begin{aligned} \dot{x} &= Ax + Bu + w \\ y &= Cx + v \\ z &= Lx \end{aligned} \tag{11.115}$$

where L is a user-defined matrix and z is the vector that we want to estimate. Our estimate of z is denoted $\hat{z}$, and our estimate of the state at time 0 is denoted $\hat{x}(0)$. The vectors w and v are disturbances with unknown statistics; they may not even be zero-mean. In the game theory approach to H_∞ filtering we define the following cost function:

$$J_1 = \frac{\int_0^T ||z - \hat{z}||_S^2 \, dt}{||x(0) - \hat{x}(0)||_{P_0^{-1}}^2 + \int_0^T \left(||w||_{Q^{-1}}^2 + ||v||_{R^{-1}}^2 \right) dt} \tag{11.116}$$

P_0, Q, R, and S are positive definite matrices chosen by the engineer based on the specific problem. Our goal is to find an estimator such that

$$J_1 < \frac{1}{\theta} \tag{11.117}$$

The estimator that solves this problem is given by

$$\begin{aligned} P(0) &= P_0 \\ \dot{P} &= AP + PA^T + Q - KCP + \theta PL^T SLP \\ K &= PC^T R^{-1} \\ \dot{\hat{x}} &= A\hat{x} + Bu + K(y - C\hat{x}) \\ \hat{z} &= L\hat{x} \end{aligned} \tag{11.118}$$

These equations are identical to the continuous-time Kalman filter equations (see Section 8.2) except for the θ term in the $\dot{P}$ equation. The inclusion of the θ term in the $\dot{P}$ equation tends to increase P, which tends to increase the gain K, which tends to make the estimator more responsive to measurements than the Kalman filter. This is a way of robustifying the filter to uncertainty in the system model. The estimator given above solves the H_∞ estimation problem if and only if $P(t)$ remains positive definite for all $t \in [0, T]$. As with the discrete-time filter, we can also obtain a steady-state continuous-time H_∞ filter. To do this we let $\dot{P} = 0$ so that the differential Riccati equation above reduces to an algebraic Riccati equation.

■ EXAMPLE 11.4

Consider the scalar continuous-time system

$$\begin{aligned} \dot{x} &= x + w \\ y &= x + v \\ z &= x \end{aligned} \tag{11.119}$$

We see that $A = C = L = 1$. Further suppose that $Q = R = S = 1$ in the cost function of Equation (11.116). Then the differential Riccati equation for the H_∞ filter is

$$\begin{aligned} \dot{P} &= AP + PA^T + Q - PC^TR^{-1}CP + \theta PL^TSLP \\ &= 2P + 1 + (\theta - 1)P^2 \end{aligned} \tag{11.120}$$

This can be solved numerically or analytically as a function of time for a given θ to give P, and then the H_∞ gain $K = PC^TR^{-1} = P$ can be obtained. We can also set $\dot{P} = 0$ in Equation (11.120) to obtain the steady-state solution for P. This gives

$$(\theta - 1)P^2 + 2P + 1 = 0 \tag{11.121}$$

As mentioned above, the solution to this quadratic equation must be positive definite in order for it to solve the H_∞ estimation problem. For this scalar equation, positive definite simply means positive. The equation has a positive solution for $\theta < 1$, in which case the steady-state solution is given by

$$P = \frac{-1 - \sqrt{2 - \theta}}{\theta - 1} \tag{11.122}$$

Suppose we choose $\theta = 7/16$. In this case, the analytic solution for the time-varying P can be obtained from Equation (11.120) as

$$\begin{aligned} P(t) &= \frac{4 + 160ce^{5t/2}}{-9 + 40ce^{5t/2}} \\ c &= \frac{9P(0) + 4}{40P(0) - 160} \end{aligned} \tag{11.123}$$

From this analytic expression for $P(t)$ we can see that

$$\lim_{t \to \infty} P(t) = 4 \tag{11.124}$$

Alternatively, we can substitute $\theta = 7/16$ in Equation (11.122) to obtain $P = 4$. Figure 11.4 shows P as a function of time when $P(0) = 1$. Note that in this example, since $C = R = 1$, the H_∞ gain K is equal to P.

Figure 11.5 shows the state estimation errors for the time-varying H_∞ filter and the steady-state H_∞ filter. In these simulations, the disturbances w and v were both normally distributed white noise sequences with standard deviations equal to 10. w had a mean of zero, and v had a mean of 10. Both simulations were run with identical disturbance time histories. It can be seen that the performance of the two filters is very similar. There are some differences between the two plots at small values of time before the

Figure 11.4 Example 11.4 H_∞ Riccati equation solution as a function of time.

time-varying Riccati solution has converged to steady state (note that the time-varying filter performs better during the initial transient). But after the Riccati solution gets close to steady state (after about $t = 1$) the performance of the two filters is nearly identical. This illustrates the possibility of saving a lot of computational effort by using a steady-state filter while giving up only an incremental amount of performance.

Figure 11.5 Example 11.4 time-varying and steady-state H_∞ filter performance when the measurement noise is zero-mean.

If we use the performance bound $\theta = 0$ in this example then we obtain the Kalman filter. The steady-state Riccati equation solution from Equation (11.120) is $(1+\sqrt{2})$ when $\theta = 0$, so the steady-state Kalman gain $K \approx 2.4$,

which is less than the steady-state H_∞ gain $K = 4$ that we obtained for $\theta = 7/16$. From Equation (11.118) we see that this will make the Kalman filter less responsive to measurements than the H_∞ filter, but the Kalman filter should provide optimal RMS error performance. Indeed, if we run the time-varying Kalman filter ($\theta = 0$) then the two-norm of the estimation error turns out to be 26.5. If we run the time-varying H_∞ filter ($\theta = 7/16$) then the two-norm of the estimation error increases to 30.0.

However, the Kalman filter assumes that the system model is known exactly, the process and measurement noises are zero-mean and uncorrelated, and the noise statistics are known exactly. If we change the simulation so the measurement noise has a mean of 10 then the H_∞ filter works better than the Kalman filter. Figure 11.6 shows the estimation error of the two filters in this case. The two-norm of the estimation error is 112.8 for the Kalman filter but only 94.2 for the H_∞ filter.

Figure 11.6 Example 11.4 time-varying Kalman and H_∞ filter performance when the measurement noise is not zero-mean.

$\nabla\nabla\nabla$

As with the discrete-time steady-state filter, if Q, R, and S are all identity matrices, the continuous-time steady-state filter bounds the maximum gain from the noise to the estimation error:

$$\begin{aligned} ||G_{\tilde{z}e}||_\infty^2 &= \sup_\omega \frac{||z - \hat{z}||_2^2}{||w||_2^2 + ||v||_2^2} \\ &\leq \frac{1}{\theta} \end{aligned} \tag{11.125}$$

where ω is the frequency of the noise, and we have defined $\tilde{z} = z - \hat{z}$, $e^T = \begin{bmatrix} w^T & v^T \end{bmatrix}^T$, and $G_{\tilde{z}e}$ is the system that has e as its input and $\tilde{z}$ as its output. The continuous-time infinity-norm of the system $G_{\tilde{z}e}$ is defined as follows:

$$
\begin{aligned}
||G_{\tilde{z}e}||_\infty &= \sup_\omega \frac{||\tilde{z}||_2}{||e||_2} \\
&= \sup_\omega \sigma_1 \left[G_{\tilde{z}e}(j\omega)\right]
\end{aligned} \tag{11.126}
$$

where $G_{\tilde{z}e}(s)$ is the transfer function from e to $\tilde{z}$.

11.5 TRANSFER FUNCTION APPROACHES

It should be emphasized that other formulations to H_∞ filtering have been proposed. For instance, Isaac Yaesh and Uri Shaked [Yae91] consider the following time-invariant system:

$$
\begin{aligned}
x_{k+1} &= Fx_k + w_k \\
x_0 &= 0 \\
y_k &= Hx_k + v_k \\
z_k &= Lx_k
\end{aligned} \tag{11.127}
$$

where w_k and v_k are uncorrelated process and measurement noise, y_k is the measurement, and z_k is the vector to be estimated. Define the estimation error as

$$
\tilde{z}_k = z_k - \hat{z}_k \tag{11.128}
$$

Define an augmented disturbance vector as

$$
e_k = \begin{bmatrix} w_k \\ v_k \end{bmatrix} \tag{11.129}
$$

The goal is to find a steady-state estimator such that the infinity-norm of the transfer function from the augmented disturbance vector e to the estimation error $\tilde{z}$ is less than some user specified bound:

$$
||G_{\tilde{z}e}||_\infty^2 < \frac{1}{\theta} \tag{11.130}
$$

The steady-state *a priori* filter that solves this problem is given as

$$
\begin{aligned}
P &= I + FPF^T - FPH^T(I + HPH^T)^{-1}HPF^T + \\
&\quad PL(I/\theta + LPL^T)^{-1}LP \\
K &= FPH^T(I + HPH^T)^{-1} \\
\hat{x}_{k+1} &= F\hat{x}_k + K(y_k - H\hat{x}_k)
\end{aligned} \tag{11.131}
$$

These equations solve the H_∞ estimation problem if and only if P is positive definite.

The steady-state *a posteriori* filter that solves this problem is given as

$$
\begin{aligned}
\Sigma^{-1} &= \tilde{P}^{-1} - \theta L^T L + H^T H \\
\tilde{P} &= F\tilde{P}(H^T H\tilde{P} - \theta L^T L\tilde{P} + I)^{-1}F^T + I \\
\tilde{K} &= (I + \theta L^T L)^{-1}\Sigma H^T \\
&= \tilde{P}(I + H^T H\tilde{P})^{-1}H^T \\
\hat{x}_{k+1} &= F\hat{x}_k + \tilde{K}(y_{k+1} - HF\hat{x}_k)
\end{aligned} \tag{11.132}
$$

Again, these equations solve the H_∞ estimation problem if and only if $\tilde{P}$ is positive definite.

Interestingly, the P matrix in the *a priori* filter of Equation (11.131) is related to the $\tilde{P}$ matrix in the *a posteriori* filter of Equation (11.132) by the following equation:

$$P^{-1} = \tilde{P}^{-1} - \theta L^T L \tag{11.133}$$

In general, the Riccati equations in these filters can be difficult to solve. However, the solution can be obtained by the eigenvector method shown in [Yae91]. (This is similar to the Hamiltonian approach to steady-state Kalman filtering described in Section 7.3.3.) Define the $2n \times 2n$ matrix

$$\mathcal{H} = \begin{bmatrix} F^T + H^T H F^{-1} & \theta F^T L^T L - H^T H F^{-1}(I - \theta L^T L) \\ -F^{-1} & F^{-1}(I - \theta L^T L) \end{bmatrix} \tag{11.134}$$

Note that F^{-1} should always exist if it comes from a real system, because F comes from a matrix exponential that is always invertible (see Sections 1.2 and 1.4). Compute the n eigenvectors of $\mathcal{H}$ that correspond to the eigenvalues outside the unit circle. Denote those eigenvectors as ξ_i $(i = 1, \ldots, n)$. Form the $2n \times n$ matrix

$$\begin{bmatrix} \xi_1 & \cdots & \xi_n \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{11.135}$$

where X_1 and X_2 are $n \times n$ matrices. The P matrix used in the *a priori* H_∞ filter can be computed as

$$P = X_2 X_1^{-1} \tag{11.136}$$

For the *a posteriori* filter, define the $2n \times 2n$ matrix

$$\tilde{\mathcal{H}} = \begin{bmatrix} F^{-T} & F^{-T}(H^T H - \theta L^T L) \\ F^{-T} & F + F^{-T}(H^T H - \theta L^T L) \end{bmatrix} \tag{11.137}$$

Compute the n eigenvectors of $\tilde{\mathcal{H}}$ that correspond to the eigenvalues outside the unit circle. Denote those eigenvectors as $\tilde{\xi}_i$ $(i = 1, \ldots, n)$. Form the $2n \times n$ matrix

$$\begin{bmatrix} \tilde{\xi}_1 & \cdots & \tilde{\xi}_n \end{bmatrix} = \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{bmatrix} \tag{11.138}$$

where $\tilde{X}_1$ and $\tilde{X}_2$ are $n \times n$ matrices. The $\tilde{P}$ matrix used in the *a posteriori* H_∞ filter can be computed as

$$\tilde{P} = \tilde{X}_2 \tilde{X}_1^{-1} \tag{11.139}$$

The eigenvector method for the Riccati equation solutions works because $\mathcal{H}$ and $\tilde{\mathcal{H}}$ are symplectic matrices (see Section 7.3.3 and Problem 11.9). This assumes that F is nonsingular and that $\mathcal{H}$ and $\tilde{\mathcal{H}}$ do not have any eigenvalues on the unit circle. If these assumptions are violated, then the problem becomes more complicated [Yae91]. A method similar to this for continuous-time systems is developed in [Nag91].

It is important to be aware that the P and $\tilde{P}$ solutions given by Equations (11.136) and (11.139) only give one solution each to Equations (11.131) and (11.132). Equations (11.136) and (11.139) may give solutions to Equations (11.131) and (11.132) that are not positive definite and therefore do not satisfy the H_∞ filtering problem. However, that does not prove that the H_∞ filtering solution does not exist (see Problem 11.13).

■ **EXAMPLE 11.5**

We will revisit Example 11.2, but assume that the initial state is 0:

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ x_0 &= 0 \\ y_k &= x_k + v_k \\ z_k &= x_k \end{aligned} \tag{11.140}$$

From Equation (11.131) we can find the *a priori* steady-state filter that bounds the infinity-norm of the transfer function from e to $\tilde{z}$ by $1/\sqrt{\theta}$. (Recall that $e_k = \begin{bmatrix} w_k & v_k \end{bmatrix}^T$.) The algebraic Riccati equation associated with this problem is given by

$$\begin{aligned} P &= 1 + P - P(1+P)^{-1}P + P(1/\theta + P)^{-1}P \\ &= 1 + P - \frac{P^2}{1+P} + \frac{P^2}{1/\theta + P} \end{aligned} \tag{11.141}$$

Solving the above for P we obtain

$$P = \frac{-\theta - 1 \pm \sqrt{\theta^2 - 6\theta + 5}}{2(2\theta - 1)} \tag{11.142}$$

In order for the solution of this equation to solve the H_∞ filtering problem, we must have $P > 0$. The only solution for which $P > 0$ is when $0 \leq \theta < 1/2$ and when we use the negative sign in the above solution.[5] If we choose $\theta = 1/10$ then $P = 2$. The gain of the *a priori* filter is then computed from Equation (11.131) as

$$\begin{aligned} K &= P(1+P)^{-1} \\ &= 2/3 \end{aligned} \tag{11.143}$$

Note that the P value that is obtained for $\theta = 1/10$ does not match Example 11.2, but K does match. The H_∞ filter equation is computed from Equation (11.131) as

$$\begin{aligned} \hat{x}_{k+1} &= \hat{x}_k + K(y_k - \hat{x}_k) \\ &= \hat{x}_k + (2/3)(y_k - \hat{x}_k) \end{aligned} \tag{11.144}$$

∇∇∇

11.6 SUMMARY

In this chapter, we have presented a couple of different approaches to H_∞ estimation, also called minimax estimation. H_∞ filtering minimizes the worst-case

[5]Note that Example 11.2 showed that this problem has a solution for $0 \leq \theta < 1$, which indicates that the game theory approach to H_∞ filtering may be more general than the transfer function approach.

estimation error and is thus more robust than Kalman filtering, which minimizes the RMS estimation error. H_∞ filtering has sometimes been criticized for being too pessimistic in its assumption about the noise processes that impinge on the system and measurement equations. After all, H_∞ estimation assumes that the noise is worst case, thus attributing a degree of perversity to the noise that intuitively seems unrealistic. This has led to mixed Kalman/H_∞ estimation techniques, which we will discuss in Chapter 12.

Research in H_∞ estimation began in the 1980s. During that decade, some work was directed toward the design of minimax state estimators for systems corrupted by random noise whose covariances were within known bounds [Poo81, Dar84, Ver84]. This was a first step toward H_∞ filtering, although it still assumed that the noise was characterized by statistical measurements. The earliest work that could pass for what we now call H_∞ filtering was probably published by Mike Grimble [Gri88]. However, unlike the presentation in this chapter, he used a frequency domain approach. He designed a state estimator such that the frequency response from the noise to the estimation error had a user-defined upper bound.

Some early tutorials on H_∞ filtering can be found in [Gri91b, Sha92]. A polynomial systems approach to H_∞ filtering is presented in [Gri90]. Nonlinear H_∞ filtering is discussed in [Rei99], where a stable state estimator with a bounded infinity-norm is derived. System identification using H_∞ methods is discussed in [Sto94, Tse94, Bai95, Did95, Pan96].

The effectiveness of the H_∞ filter can be highly sensitive to the weighting functions [e.g., S_k, P_0, Q_k, and R_k in Equation (11.36), and θ in the performance bound]. This sometimes makes H_∞ filter design more sensitive than Kalman filter design (which is ironic, considering the higher degree of robustness in H_∞ filtering). The advantages of H_∞ estimation over Kalman filtering can be summarized as follows.

1. H_∞ filtering provides a rigorous method for dealing with systems that have model uncertainty.

2. H_∞ filtering provides a natural way to limit the frequency response of the estimator.

The disadvantages of H_∞ filtering compared to Kalman filtering can be summarized as follows.

1. The filter performance is more sensitive to the design parameters.

2. The theory underlying H_∞ filtering is more abstract and complicated.

The types of applications where H_∞ filtering may be preferred over Kalman filtering could include the following.

1. Systems in which stability margins must be guaranteed, or worst-case estimation performance is a primary consideration (rather than RMS estimation performance) [Sim96].

2. Systems in which the model changes unpredictably, and identification and gain scheduling are too complex or time-consuming.

3. Systems in which the model is not well known.

Work by Babak Hassibi, Ali Sayed, and Thomas Kailath involves the solution of state estimation problems within the context of Krein spaces (as opposed to the usual Hilbert space approach). This provides a general framework for both Kalman and H_∞ filtering (along with other types of filtering), and is discussed in some of their papers [Has96a, Has96b] and books [Has99, Kai00].

PROBLEMS

Written exercises

11.1 Show that $(I + A)^{-1}A = A(I + A)^{-1}$.

11.2 Consider a scalar system with $F = H = 1$ and with process noise and measurement noise variances Q and R. Suppose a state estimator of the form

$$\hat{x}_{k+1}^- = \hat{x}_k^- + K(y_k - \hat{x}_k^-)$$

is used to estimate the state, where K is a general estimator gain.

a) Find the optimal gain K if $R = 2Q$. Call this gain K_0. What is the resulting steady-state *a priori* estimation-error variance?

b) Suppose that $R = 0$. What is the optimal steady-state *a priori* estimation-error variance? What is the (suboptimal) steady-state *a priori* estimation-error variance if K_0 is used in the estimator? Repeat for $R = Q$ and $R = 5Q$.

11.3 Consider a scalar system with $F = H = 1$ and with process noise and measurement noise variances Q and $R = 2Q$. A Kalman filter is designed to estimate the state, but (unknown to the engineer) the process noise has a mean of $\bar{w}$.

a) What is the steady-state value of the mean of the *a priori* estimation error?

b) Introduce a new state-vector element that is equal to $\bar{w}$. Augment the new state-vector element to the original system so that a Kalman filter can be used to estimate both the original state element and the new state element. Find an analytical solution to the steady-state *a priori* estimation-error covariance for the augmented system.

11.4 Suppose that a Kalman filter is designed to estimate the state of a scalar system. The assumed system is given as

$$\begin{aligned} x_{k+1} &= Fx_k + w_k \\ y_k &= Hx_k + v_k \end{aligned}$$

where $w_k \sim (0, Q)$ and $v_k \sim (0, R)$ are uncorrelated zero-mean white noise processes. The actual system matrix is $\tilde{F} = F + \Delta F$.

a) Under what conditions is the mean of the steady-state value of the *a priori* state estimation error equal to zero?

b) What is the steady-state value of the *a priori* estimation-error variance P? How much larger is P because of the modeling error ΔF?

11.5 Find the stationary point of $(x_1^2 + x_1x_2 + x_2x_3)$ subject to the constraint $(x_1 + x_2 = 4)$ [Moo00].

11.6 Maximize $(14x - x^2 + 6y - y^2 + 7)$ subject to the constraints $(x + y \leq 2)$ and $(x + 2y \leq 3)$ [Lue84].

11.7 Consider the system

$$\begin{aligned} x_k &= \frac{1}{2}x_{k-1} + w_{k-1} \\ y_k &= x_k + v_k \end{aligned}$$

Note that this is the system model for the radiation system described in Problem 5.1.

a) Find the steady-state value of P_k for the H_∞ filter, using a variable θ and $L = R = Q = S = 1$.

b) Find the bound on θ such that the steady-state H_∞ filter exists.

11.8 Suppose that you use a continuous-time H_∞ filter to estimate a constant on the basis of noisy measurements. The measurement noise is zero-mean and white with a covariance of R. Find the H_∞ estimator gain as a function of P_0, R, θ, and time. What is the limit of the estimator gain as $t \to \infty$? What is the maximum value of θ such that the H_∞ estimation problem has a solution? How does the value of θ influence the estimator gain?

11.9 Prove that $\mathcal{H}$ and $\tilde{\mathcal{H}}$ in Equations (11.134) and (11.137) are symplectic.

11.10 Prove that the solution of the *a posteriori* H_∞ Riccati equation given in Equation (11.132) with $\theta = 0$ is equivalent to the solution of the steady-state *a priori* Kalman filter Riccati equation with $R = I$ and $Q = I$.

11.11 Prove that Σ in Equation (11.132) with $\theta = 0$ is equivalent to the solution of the steady-state *a posteriori* Kalman filter Riccati equation with $R = I$ and $Q = I$.

11.12 Find the *a posteriori* steady-state H_∞ filter for Example 11.5 when $\theta = 1/10$. Verify that the *a priori* and *a posteriori* Riccati equation solutions satisfy Equation (11.133).

11.13 Find all possible solutions P to the *a priori* H_∞ filtering problem for Example 11.5 when $\theta = 0$. Next use Equation (11.139) to find the P solution. Repeat for $\theta = 1/10$. [Note that Equation (11.139) gives a negative solution for P and therefore cannot be used.]

Computer exercises

11.14 Generate the time-varying solution to P_k for Problem 11.7 with $P_0 = 1$. What is the largest value of θ for which Equation (11.90) will be satisfied for all k up to and including $k = 20$? Answer to the nearest 0.01. Repeat for $k = 10$, $k = 5$, and $k = 1$.

11.15 Consider the vehicle navigation problem described in Example 7.12. Design a Kalman filter and an H_∞ filter to estimate the states of the system. Use the

following parameters.

$$\begin{aligned} T &= 3 \\ u_k &= 1 \\ Q &= \text{diag}(4, 4, 1, 1) \\ R &= \text{diag}(900, 900) \\ \text{heading angle} &= 0.9\pi \\ x(0) &= \hat{x}(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T \end{aligned}$$

Simulate the system and the filters for 300 seconds. In the H_∞ filter use $S = L = I$ and $\theta = 0.0005$.

a) Plot the position estimation errors for the Kalman and H_∞ filters. What are the RMS position estimation errors for the two filters?

b) Now suppose that unknown to the filter designer, $u_k = 2$. Plot the position estimation errors for the Kalman and H_∞ filters. What are the RMS position estimation errors for the two filters?

c) What are the closed loop estimator eigenvalues for the Kalman and H_∞ filters? Do their relative magnitudes agree with your intuition?

d) Use MATLAB's DARE function to find the largest θ for which a steady-state solution exists to the H_∞ DARE. Answer to the nearest 0.0001. How well does the H_∞ filter work for this value of θ? What are the closed-loop eigenvalues of the H_∞ filter for this value of θ?

CHAPTER 12

Additional topics in H_∞ filtering

> Since [H_∞ filters] make no assumption about the disturbances, they have to accommodate for all conceivable disturbances, and are thus over-conservative.
>
> —Babak Hassibi and Thomas Kailath [Has95]

In this chapter we will briefly introduce some advanced topics in H_∞ filtering. H_∞ filtering was not introduced until the 1980s and is therefore considerably less mature than Kalman filtering. As such, there is more room for additional work and development in H_∞ filtering than Kalman filtering. This chapter introduces some of the current directions of research in the area of H_∞ filtering.

Section 12.1 looks at the mixed Kalman/H_∞ estimation problem. We present a filter that satisfies an H_∞ performance bound while at the same time minimizing a Kalman performance bound. Section 12.2 looks at the robust mixed Kalman/H_∞ estimation problem. This is the same as mixed Kalman/H_∞ filtering but with the added complication of uncertainties in the system matrices. Section 12.3 discusses the solution of the constrained H_∞ filter, where equality (or inequality) constraints are enforced on the state estimate.

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

12.1 MIXED KALMAN/H_∞ FILTERING

In this section we look at the problem of finding a filter that combines the best features of Kalman filtering with the best features of H_∞ filtering. This problem can be attacked a couple of different ways. Recall from Section 5.2 the cost function that is minimized by the steady-state Kalman filter:

$$J_2 = \lim_{N\to\infty} \sum_{k=0}^{N} E\left(||x_k - \hat{x}_k||_2\right) \tag{12.1}$$

Recall from Section 11.3 the cost function that is minimized by the steady-state H_∞ state estimator if S_k and L_k are identity matrices:

$$J_\infty = \lim_{N\to\infty} \max_{x_0, w_k, v_k} \frac{\sum_{k=0}^{N} ||x_k - \hat{x}_k||^2}{||x(0) - \hat{x}(0)||^2_{P_0^{-1}} + \sum_{k=0}^{N}\left(||w_k||^2_{Q_k^{-1}} + ||v_k||^2_{R_k^{-1}}\right)} \tag{12.2}$$

Loosely speaking, the Kalman filter minimizes the RMS estimation error, and the H_∞ filter minimizes the worst-case estimation error.

In [Had91] these two performance objectives are combined to form the following problem: Given the n-state observable LTI system

$$\begin{aligned} x_{k+1} &= Fx_k + w_k \\ y_k &= Hx_k + v_k \end{aligned} \tag{12.3}$$

where $\{w_k\}$ and $\{v_k\}$ are uncorrelated zero-mean, white noise processes with covariances Q and R respectively, find an estimator of the form

$$\hat{x}_{k+1} = \hat{F}x_k + Ky_k \tag{12.4}$$

that satisfies the following criteria:

1. $\hat{F}$ is a stable matrix (so the estimator is stable).
2. The H_∞ cost function is bounded by a user-specified parameter:

$$J_\infty < \frac{1}{\theta} \tag{12.5}$$

3. Among all estimators satisfying the above criteria, the filter minimizes the Kalman filter cost function J_2.

The solution to this problem provides the best RMS estimation error among all estimators that bound the worst-case estimation error. The filter that solves this problem is given as follows.

The mixed Kalman/H_∞ filter

1. Find the $n \times n$ positive semidefinite matrix P that satisfies the following Riccati equation:

$$P = FPF^T + Q + FP(I/\theta^2 - P)^{-1}PF^T - P_aV^{-1}P_a^T \tag{12.6}$$

where P_a and V are defined as

$$\begin{aligned} P_a &= FPH^T + FP(I/\theta^2 - P)^{-1}PH^T \\ V &= R + HPH^T + HP(I/\theta^2 - P)^{-1}PH^T \end{aligned} \tag{12.7}$$

2. Derive the $\hat{F}$ and K matrices in Equation (12.4) as

$$\begin{aligned} K &= P_a V^{-1} \\ \hat{F} &= F - KH \end{aligned} \tag{12.8}$$

3. The estimator of Equation (12.4) satisfies the mixed Kalman/H_∞ estimation problem if and only if $\hat{F}$ is stable. In this case, the state estimation error satisfies the bound

$$\lim_{k\to\infty} E\left(||x_k - \hat{x}_k||^2\right) \leq \mathrm{Tr}(P) \tag{12.9}$$

Note that if $\theta = 0$, then the problem statement reduces to the Kalman filter problem statement. In this case we can see that Equation (12.6) reduces to the discrete-time algebraic Riccati equation that is associated with the Kalman filter (see Problem 12.2 and Section 7.3). The continuous-time version of this theory is given in [Ber89].

■ EXAMPLE 12.1

In this example, we take another look at the scalar system that is described in Example 11.2:

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + v_k \end{aligned} \tag{12.10}$$

where $\{w_k\}$ and $\{v_k\}$ are uncorrelated zero-mean, white noise processes with covariances Q and R, respectively. Equation (12.6), the Riccati equation for the mixed Kalman/H_∞ filter, reduces to the following scalar equation:

$$\theta^2(1 - R\theta^2)P^3 + (Q\theta^2 + 1)(R\theta^2 - 1)P^2 + Q(1 - 2R\theta^2) + QR = 0 \tag{12.11}$$

Suppose that (for some value of Q, R, and θ) this equation has a solution $P \geq 0$, and $|1 - K| < 1$, where the filter gain K from Equation (12.8) is given as

$$K = \frac{P}{P + R - PR\theta^2} \tag{12.12}$$

Then J_∞ from Equation (12.2) is bounded from above by $1/\theta$, and the variance of the state estimation error is bounded from above by P. The top half of Figure 12.1 shows the Kalman filter performance bound P and the estimator gain K as a function of θ when $Q = R = 1$. Note that at $\theta = 0$ the mixed Kalman/H_∞ filter reduces to a standard Kalman filter. In this case the performance bound $P \approx 1.62$ and the estimator gain $K \approx 0.62$, as discussed in Example 11.2. However, if $\theta = 0$ then we do not have any guarantee on the worst-case performance index J_∞.

From the top half of Figure 12.1, we see that as θ increases, the performance bound P increases, which means that our Kalman performance index gets

worse. However, at the same time, the worst-case performance index J_∞ decreases as θ increases. From the bottom half of Figure 12.1, we see that as θ increases, K increases, which is consistent with better H_∞ performance and worse Kalman performance (see Example 11.2). When θ reaches about 0.91, numerical difficulties prevent a solution to the mixed filter problem.

The bottom half of Figure 12.1 shows that at $\theta = 0.5$ the estimator gain $K \approx 0.76$. Recall from Example 11.2 that the H_∞ filter had an estimator gain $K = 1$ for the same value of θ. This shows that the mixed Kalman/H_∞ filter has a smaller estimator gain (for the same θ) than the pure H_∞ filter. In other words, the mixed filter uses a lower gain in order to obtain better Kalman performance, whereas the pure H_∞ filter uses a higher gain because it does not take Kalman performance into account.

Figure 12.1 Results for Example 12.1 of the estimation-error variance bound and estimator gain as a function of θ for the mixed Kalman/H_∞ filter. As θ increases, the worst-case performance bound $1/\theta$ decreases, the error-variance bound increases, and the estimator gain increases greater than the Kalman gain ($\theta = 0$). This shows a trade-off between worst-case performance and RMS performance.

$\nabla\nabla\nabla$

Although the approach presented above is a theoretically elegant method of obtaining a mixed Kalman/H_∞ filter, the solution of the Riccati equation can be challenging for problems with a large number of states. Other more straightforward approaches can be used to combine the Kalman and H_∞ filters. For example, if the steady-state Kalman filter gain for a given problem is denoted as K_2 and the steady-state H_∞ filter gain is denoted as K_∞, then a hybrid filter gain can be constructed as

$$K = dK_2 + (1-d)K_\infty \tag{12.13}$$

where $d \in [0, 1]$. This hybrid filter gain is a convex combination of the Kalman and H_∞ filter gains, which would be expected to provide a balance between RMS and worst-case performance [Sim96]. However, this approach is not as attractive theoretically since stability must be determined numerically, and no *a priori* bounds on

the Kalman or H_∞ performance measures can be given. Analytical determination of stability and performance bounds for this type of filter is an open research issue.

12.2 ROBUST KALMAN/H_∞ FILTERING

The material in this section is based on [Hun03]. In most practical problems, an exact model of the system may not be available. The performance of the system in the presence of model uncertainties becomes an important issue. For example, suppose we have a system given as

$$\begin{aligned} x_{k+1} &= (F_k + \Delta F_k)x_k + w_k \\ y_k &= (H_k + \Delta H_k)x_k + v_k \end{aligned} \tag{12.14}$$

where $\{w_k\}$ and $\{v_k\}$ are uncorrelated zero-mean white noise processes with covariances Q_k and R_k, respectively. Matrices ΔF_k and ΔH_k represent uncertainties in the system and measurement matrices. These uncertainties are assumed to be of the form

$$\begin{bmatrix} \Delta F_k \\ \Delta H_k \end{bmatrix} = \begin{bmatrix} M_{1k} \\ M_{2k} \end{bmatrix} \Gamma_k N_k \tag{12.15}$$

where M_{1k}, M_{2k}, and N_k are known matrices, and Γ_k is an unknown matrix satisfying the bound

$$\Gamma_k^T \Gamma_k \leq I \tag{12.16}$$

[Recall that we use the general notation $A \leq B$ to denote that $(A - B)$ is a negative semidefinite matrix.] Assume that F_k is nonsingular. This assumption is not too restrictive; F_k should always be nonsingular for a real system because it comes from the matrix exponential of the system matrix of a continuous-time system, and the matrix exponential is always nonsingular (see Sections 1.2 and 1.4). The problem is to design a state estimator of the form

$$\hat{x}_{k+1} = \hat{F}_k \hat{x}_k + K_k y_k \tag{12.17}$$

with the following characteristics:

1. The estimator is stable (i.e., the eigenvalues of $\hat{F}_k$ are less than one in magnitude).

2. The estimation error $\tilde{x}_k$ satisfies the following worst-case bound:

$$\max_{w_k, v_k} \frac{||\tilde{x}_k||_2}{||w_k||_2 + ||v_k||_2 + ||\tilde{x}_0||_{S_1^{-1}} + ||x_0||_{S_2^{-1}}} < \frac{1}{\theta} \tag{12.18}$$

3. The estimation error $\tilde{x}_k$ satisfies the following RMS bound:

$$E\left(\tilde{x}_k \tilde{x}_k^T\right) < P_k \tag{12.19}$$

The solution to the problem can be found by the following procedure [Hun03].

The robust mixed Kalman/H_∞ filter

1. Choose some scalar sequence $\alpha_k > 0$, and a small scalar $\epsilon > 0$.

2. Define the following matrices:

$$\begin{aligned} R_{11k} &= Q_k + \alpha_k M_{1k} M_{1k}^T \\ R_{12k} &= \alpha_k M_{1k} M_{2k}^T \\ R_{22k} &= R_k + \alpha_k M_{2k} M_{2k}^T \end{aligned} \tag{12.20}$$

3. Initialize P_k and $\tilde{P}_k$ as follows:

$$\begin{aligned} P_0 &= S_1 \\ \tilde{P}_0 &= S_2 \end{aligned} \tag{12.21}$$

4. Find positive definite solutions P_k and $\tilde{P}_k$ satisfying the following Riccati equations:

$$\begin{aligned} P_{k+1} &= F_{1k} T_k F_{1k}^T + R_{11k} + R_{11k} R_{2k} R_{11k}^T - \\ &\quad \left[F_{1k} T_k H_{1k}^T + R_{11k} R_{2k} R_{12k} + R_{12k}\right] R_k^{-1} \left[\cdots\right]^T + \epsilon I \\ \tilde{P}_{k+1} &= F_k \tilde{P}_k F_k^T + F_k \tilde{P}_k N_k^T (\alpha_k I - N_k \tilde{P}_k N_k^T)^{-1} N_k \tilde{P}_k F_k^T + \\ &\quad R_{11k} + \epsilon I \end{aligned} \tag{12.22}$$

where the matrices R_{1k}, R_{2k}, F_{1k}, H_{1k}, and T_k are defined as

$$\begin{aligned} R_{1k} &= (\tilde{P}_k^{-1} - N_k^T N_k / \alpha_k)^{-1} F_k^T \\ R_{2k} &= R_{1k}^{-1} (\tilde{P}_k^{-1} - N_k^T N_k / \alpha_k)^{-1} R_{1k}^{-T} \\ F_{1k} &= F_k + R_{11k} R_{1k}^{-1} \\ H_{1k} &= H_k + R_{12k}^T R_{1k}^{-1} \\ T_k &= (P_k^{-1} - \theta^2 I)^{-1} \end{aligned} \tag{12.23}$$

5. If the Riccati equation solutions satisfy

$$\begin{aligned} \frac{1}{\theta^2} I &> P_k \\ \alpha_k I &> N_k \tilde{P}_k N_k^T \end{aligned} \tag{12.24}$$

then the estimator of Equation (12.17) solves the problem with

$$\begin{aligned} K_k &= \left[F_{1k} T_k H_{1k}^T + R_{11k} R_{2k} R_{12k} + R_{12k}\right] \tilde{R}_k^{-1} \\ \tilde{R}_k &= H_{1k} T_k H_{1k}^T + R_{12k}^T R_{2k} R_{12k} + R_{22k} \\ \hat{F}_k &= F_{1k} - K_k H_{1k} \end{aligned} \tag{12.25}$$

The parameter ϵ is generally chosen as a very small positive number. In the example in [Hun03] the value is $\epsilon = 10^{-8}$. The parameter α_k has to be chosen large enough so that the conditions of Equation (12.24) are satisfied. However, as α_k increases, P_k also increases, which results in a looser bound on the RMS estimation error.

A steady-state robust filter can be obtained by letting $P_{k+1} = P_k$ and $\tilde{P}_{k+1} = \tilde{P}_k$ in Equation (12.22) and removing all the time subscripts (assuming that the system is time-invariant). But the resulting coupled steady-state Riccati equations will be more difficult to solve than the discrete-time Riccati equations in Equation (12.22), which can be solved by a simple (albeit tedious) iterative process. Similar problems have been solved in [Mah04b, Xie04, Yoo04].

■ EXAMPLE 12.2

Suppose we have an angular positioning system such as a motor. The moment of inertia of the motor and its load is J and the coefficient of viscous friction is B. The torque that is applied to the motor is $cu+w$, where u is the applied voltage, c is a motor constant that relates applied voltage to generated torque, and w is unmodeled torque that can be considered as noise. The differential equation for this system is given as

$$J\ddot{\phi} + B\dot{\phi} = cu + w \tag{12.26}$$

where ϕ is the motor shaft angle. We choose the states as $x(1) = \phi$ and $x(2) = \dot{\phi}$. The dynamic system model can then be written as

$$\begin{aligned} \dot{x} &= Ax + B_u u + B_w w \\ &= \begin{bmatrix} 0 & 1 \\ 0 & -B/J \end{bmatrix} x + \begin{bmatrix} 0 \\ c/J \end{bmatrix} u + \begin{bmatrix} 0 \\ 1/J \end{bmatrix} w \end{aligned} \tag{12.27}$$

In order to discretize the system with a sample time T, we use the method of Section 1.4 to obtain

$$x_{k+1} = Fx_k + G_u u_k + G_w w_k \tag{12.28}$$

The discrete-time system matrices are given as

$$\begin{aligned} F &= \exp(AT) \\ G_u &= \int_0^T \exp(At)\, dt\, B_u \\ &= \frac{c}{B} \begin{bmatrix} T - 1/\alpha + e^{-\alpha T}/\alpha \\ 1 - e^{-\alpha T} \end{bmatrix} \\ G_w &= \frac{1}{B} \begin{bmatrix} T - 1/\alpha + e^{-\alpha T}/\alpha \\ 1 - e^{-\alpha T} \end{bmatrix} \\ \alpha &= \frac{B}{J} \end{aligned} \tag{12.29}$$

If our measurement is angular position ϕ corrupted by noise, then our measurement equation can be written as

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k \tag{12.30}$$

Suppose the system has a torque disturbance w_k with a standard deviation of 2, and a measurement noise v_k with a standard deviation of 0.2 degrees. We can run the Kalman filter and the robust mixed Kalman/H$_\infty$ filter for

this problem. Figure 12.2 shows the position and velocity estimation errors of the Kalman and robust mixed Kalman/H_∞ filters. The robust filter performs better at the beginning of the simulation, although the Kalman filter performs better in steady state. (It is not easy to see the advantage of Kalman filter during steady state in Figure 12.2 because of the scale, but over the last half of the plot the Kalman filter estimation errors have standard deviations of 0.33 deg and 1.65 deg/s, whereas the robust filter has standard deviations of 0.36 deg and 4.83 deg/s.)

Now suppose that the moment of inertia of the motor changes by a factor of 100. That is, the filter assumes that J is 100 times greater than it really is. In this, case Figure 12.3 shows the position and velocity estimation errors of the Kalman and robust filters. It is apparent that in this case the robust filter performs better not only at the beginning of the simulation, but in steady state also. After the filters reach "steady state" (which we have defined somewhat arbitrarily as the time at which the position estimation error magnitude falls below 1 degree) the Kalman filter RMS estimation errors are 0.36 degrees for position and 1.33 degrees/s for velocity, whereas the robust filter RMS estimation errors are 0.28 degrees for position and 1.29 degrees/s for velocity. The square roots of the diagonal elements of the P_k Riccati equation solution of Equation (12.22) reach steady-state values of 0.51 degrees and 1.52 degrees/s, which shows that the estimation-error variance is indeed bounded by the Riccati equation solution P_k.

Figure 12.2 Position and velocity estimation errors for Example 12.2 for the Kalman filter and the robust filter, assuming that the system model is perfectly known. The robust filter performs better at the beginning of the simulation, but the Kalman filter performs better in steady state. The steady-state Kalman filter estimation errors have standard deviations of 0.33 deg and 1.65 deg/s, whereas the robust filter has standard deviations of 0.36 deg and 4.83 deg/s.

∇∇∇

Figure 12.3 Position and velocity estimation errors for Example 12.2 for the Kalman filter and the robust filter, assuming that the system model is not well known. The robust filter performs better both at the beginning of the simulation and in steady state. The steady-state Kalman filter estimation errors have standard deviations of 0.36 deg and 1.33 deg/s, whereas the robust filter has standard deviations of 0.28 deg and 1.29 deg/s.

12.3 CONSTRAINED H_∞ FILTERING

As in Section 7.5, suppose that we know (on the basis of physical considerations) that the states satisfy some equality constraint $D_k x_k = d_k$, or some inequality constraint $D_k x_k \leq d_k$, where D_k is a known matrix and d_k is a known vector. This section discusses how those constraints can be incorporated into the H_∞ filter equations. As discussed in Section 7.5, state equality constraints can always be handled by reducing the system-model parameterization [Wen92], or by treating state equality constraints as perfect measurements [Por88, Hay98]. However, these approaches cannot be extended to inequality constraints. The approach summarized in this section is to incorporate the state constraints into the derivation of the H_∞ filter [Sim06c].

Consider the discrete LTI system given by

$$\begin{aligned} x_{k+1} &= F x_k + w_k + \delta_k \\ y_k &= H x_k + v_k \end{aligned} \tag{12.31}$$

where y_k is the measurement, $\{w_k\}$ and $\{v_k\}$ are uncorrelated white noise sequences with respective covariances Q and I, and $\{\delta_k\}$ is a noise sequence generated by an adversary (i.e., nature). Note that we are assuming that the measurement noise has a unity covariance matrix. In a real system, if the measurement noise covariance is not equal to the identity matrix, then we will have to normalize the measurement equation as shown in Example 12.3 below. In general, F, H, and Q can be time-varying matrices, but we will omit the time subscript on these matrices for ease of notation. In addition to the state equation, we know (on the basis of physical considerations or other *a priori* information) that the states satisfy the following

constraint:

$$D_k x_k = d_k \tag{12.32}$$

We assume that the D_k matrix is full rank and normalized so that $D_k D_k^T = I$. In general, D_k is an $s \times n$ matrix, where s is the number of constraints, n is the number of states, and $s < n$. If $s = n$ then Equation (12.32) completely defines x_k, which makes the estimation problem trivial (i.e., $\hat{x}_k = D_k^{-1} d_k$). For $s < n$, which is the case in this section, there are fewer constraints than states, which makes the estimation problem nontrivial. Assuming that D_k is full rank is the same as the assumption made in the constrained Kalman filtering problem of Section 7.5. For notational convenience we define the matrix V_k as

$$V_k = D_k^T D_k \tag{12.33}$$

We assume that both the noisy system and the noise-free system satisfy the state constraint. The problem is to find an estimate $\hat{x}_{k+1}$ of x_{k+1} given the measurements $\{y_1, y_2, \cdots, y_k\}$. The estimate should satisfy the state constraint. We will assume that the estimate is given by the following standard predictor/corrector form:

$$\begin{aligned} \hat{x}_0 &= 0 \\ \hat{x}_{k+1} &= F\hat{x}_k + K_k(y_k - H\hat{x}_k) \end{aligned} \tag{12.34}$$

The noise δ_k in (12.31) is introduced by an adversary that has the goal of maximizing the estimation error. We will assume that our adversary's input to the system is given as follows:

$$\delta_k = L_k[G_k(x_k - \hat{x}_k) + n_k] \tag{12.35}$$

where L_k is a gain to be determined, G_k is a given matrix, and $\{n_k\}$ is a noise sequence with variance equal to the identity matrix. We assume that $\{n_k\}$ is uncorrelated with $\{w_k\}$, $\{v_k\}$, and x_0. This form of the adversary's input is not intuitive because it is based on the state estimation error, but this form is taken because the solution of the resulting problem results in a state estimator that bounds the infinity-norm of the transfer function from the random noise terms to the state estimation error [Yae92].

G_k in Equation (12.35) is chosen by the designer as a tuning parameter or weighting matrix that can be adjusted on the basis of our *a priori* knowledge about the adversary's noise input. Suppose, for example, that we know ahead of time that the first component of the adversary's noise input to the system is twice the magnitude of the second component, the third component is zero, and so on; then that information can be reflected in the designer's choice of G_k. We do not need to make any assumptions about the form of G_k (e.g., it does not need to be positive definite or square). From Equation (12.35) we see that as G_k approaches the zero matrix, the adversary's input becomes a purely random process without any deterministic component. This causes the resulting filter to approach the Kalman filter; that is, we obtain better RMS error performance but poorer worst-case error performance. As G_k becomes large, the filter places more emphasis on minimizing the estimation error due to the deterministic component of the adversary's input. That is, the filter assumes less about the adversary's input, and we obtain better worst-case error performance but worse RMS error performance. The estimation error is defined as

$$e_k = x_k - \hat{x}_k \tag{12.36}$$

It can be shown from the preceding equations that the dynamic system describing the estimation error is given as

$$\begin{aligned} e_0 &= x_0 \\ e_{k+1} &= (F - K_k H + L_k G_k)e_k + w_k + L_k n_k - K_k v_k \end{aligned} \tag{12.37}$$

Since $D_k x_k = D_k \hat{x}_k = d_k$, we see that $D_k e_k = 0$. But it can also be shown [Sim06c] that $D_{k+1} F e_k = 0$. Therefore, we can subtract the zero term $D_{k+1}^T D_{k+1} F e_k = V_{k+1} F e_k$ from Equation (12.37) to obtain

$$\begin{aligned} e_0 &= x_0 \\ e_{k+1} &= [(I - V_{k+1})F - K_k H + L_k G_k]e_k + w_k + L_k n_k - K_k v_k \end{aligned} \tag{12.38}$$

However, this is an inappropriate term for a minimax problem because the adversary can arbitrarily increase e_k by arbitrarily increasing L_k. To prevent this, we decompose e_k as

$$e_k = e_{1,k} + e_{2,k} \tag{12.39}$$

where $e_{1,k}$ and $e_{2,k}$ evolve as follows:

$$\begin{aligned} e_{1,0} &= x_0 \\ e_{1,k} &= [(I - V_{k+1})F - K_k H + L_k Gk]e_{1,k} + w_k - K_k v_k \\ e_{2,0} &= 0 \\ e_{2,k} &= [(I - V_{k+1})F - K_k H + L_k G_k]e_{2,k} + L_k n_k \end{aligned} \tag{12.40}$$

We define the objective function for the filtering problem as

$$J(K, L) = \text{trace} \sum_{k=0}^{N} W_k E\left(e_{1,k} e_{1,k}^T - e_{2,k} e_{2,k}^T\right) \tag{12.41}$$

where W_k is any positive definite weighting matrix. The differential game is for the filter designer to find a gain sequence $\{K_k\}$ that minimizes J, and for the adversary to find a gain sequence $\{L_k\}$ that maximizes J. As such, J is considered a function of $\{K_k\}$ and $\{L_k\}$, which we denote in shorthand notation as K and L. This objective function is not intuitive, but is used here because the solution of the problem results in a state estimator that bounds the infinity-norm of the transfer function from the random noise terms to the state estimation error [Yae92]. That is, suppose we can find an estimator gain K^* that minimizes $J(K, L)$ when the matrix G_k in (12.35) is equal to θI for some positive scalar θ. Then the infinity-norm of the weighted transfer function from the noise terms w_k and v_k to the estimation error e_k is bounded by $1/\theta$. That is,

$$\sup_{w_k, v_k} \frac{\sum_{k=0}^{N} e_k^T e_k}{\sum_{k=0}^{N} \left(w_k^T Q^{-1} w_k + v_k^T v_k\right)} < \frac{1}{\theta} \tag{12.42}$$

where *sup* stands for *supremum.*[1] The filtering solution is obtained by finding optimal gain sequences $\{K_k^*\}$ and $\{L_k^*\}$ that satisfy the following saddle point:

$$J(K^*, L) \leq J(K^*, L^*) \leq J(K, L^*) \text{ for all } K, L \tag{12.43}$$

[1]The supremum of a function is its least upper bound. This is similar to the maximum of a function, but a maximum is a value that is actually attained by a function, whereas a supremum may or may not be attained. For example, the supremum of $(1 - e^{-x})$ is 1, but $(1 - e^{-x})$ never actually reaches the value 1. Similar distinctions hold for the operators minimum and infimum (usually abbreviated *inf*).

This problem is solved subject to the constraint that $D_k\hat{x}_k = d_k$ in [Sim06c], whose result is presented here. We define P_k and Σ_k as the nonsingular solutions to the following set of equations:

$$\begin{aligned} P_0 &= E(x_0 x_0^T) \\ \Sigma_k &= (P_k H^T H - P_k G_k^T G_k + I)^{-1} P_k \\ P_{k+1} &= (I - V_{k+1}) F \Sigma_k F^T (I - V_{k+1}) + Q \end{aligned} \tag{12.44}$$

Nonsingular solutions to these equations are not always guaranteed to exist, in which case a solution to the H_∞ filtering problem may not exist. However, if nonsingular solutions do exist, then the following gain matrices for our estimator and adversary satisfy the constrained H_∞ filtering problem:

$$\begin{aligned} K_k^* &= (I - V_{k+1}) F \Sigma_k H^T \\ L_k^* &= (I - V_{k+1}) F \Sigma_k G_k^T \end{aligned} \tag{12.45}$$

These matrices solve the constrained H_∞ filtering problem only if $(I - G_k P_k G_k^T) \geq 0$. Note that as G_k becomes larger, we will be less likely to satisfy this condition. From Equation (12.35) we see that a larger G_k gives the adversary more latitude in choosing a disturbance. This makes it less likely that the designer can minimize the cost function.

The mean square estimation error that results from using the optimal gain K_k^* cannot be specified because it depends on the adversary's input δ_k. However, we can state an upper bound for the mean square estimation error [Sim06c] as follows:

$$E\left[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\right] \leq P_k \tag{12.46}$$

This provides additional motivation for using the game theory approach presented in this section. The estimator not only bounds the worst-case estimation error, but also bounds the mean square estimation error.

Now consider the special case that there are no state constraints. Then in Equation (12.32) we can set the D_k matrix equal to a zero row vector and the d_k vector equal to the zero scalar. In this case $V_{k+1} = 0$ and we obtain from Equations (12.44) and (12.45) the following estimator and adversary strategies:

$$\begin{aligned} P_0 &= E(x_0 x_0^T) \\ P_k(I - H^T H \Sigma_k) &= (I - P_k G_k^T G_k)\Sigma_k \\ P_{k+1} &= F \Sigma_k F^T + Q \\ K_k^* &= F \Sigma_k C^T \\ L_k^* &= F \Sigma_k G_k^T \end{aligned} \tag{12.47}$$

This is identical to the unconstrained H_∞ estimator [Yae92]. The unconstrained H_∞ estimator for continuous-time systems is given in [Yae04].

In the case of state inequality constraints (i.e., constraints of the form $D_k x_k \leq d_k$), a standard active-set method [Fle81, Gil81] can be used to solve the H_∞ filtering problem. An active-set method uses the fact that it is only those constraints that are active at the solution of the problem that affect the optimality conditions; the inactive constraints can be ignored. Therefore, an inequality-constrained problem is equivalent to an equality-constrained problem. An active-set method determines

which constraints are active at the solution of the problem and then solves the problem using the active constraints as equality constraints. Inequality constraints will significantly increase the computational effort required for a problem solution because the active constraints need to be determined, but conceptually this poses no difficulty.

The constrained H_∞ filter

The constrained H_∞ filter can be summarized as follows.

1. We have a linear system given as

$$\begin{aligned} x_{k+1} &= F_k x_k + w_k \\ y_k &= H_k x_k + v_k \\ D_k x_k &= d_k \end{aligned} \tag{12.48}$$

   where w_k is the process noise, v_k is the measurement noise, and the last equation above specifies equality constraints on the state. We assume that the constraints are normalized so $D_k D_k^T = I$. The covariance of w_k is equal to Q_k, but w_k might have a zero mean or it might have a nonzero mean (i.e., it might contain a deterministic component). The covariance of v_k is the identity matrix.

2. Initialize the filter as follows:

$$\begin{aligned} \hat{x}_0 &= 0 \\ P_0 &= E(x_0 x_0^T) \end{aligned} \tag{12.49}$$

3. At each time step $k = 0, 1, \cdots$, do the following.

   (a) Choose the tuning parameter matrix G_k to weight the deterministic, biased component of the process noise. If $G_k = 0$ then we are assuming that the process noise is zero-mean and we get Kalman filter performance. As G_k increases we are assuming that there is more of a deterministic, biased component to the process noise. This gives us better worst-case error performance but worse RMS error performance.

   (b) Compute the next state estimate as follows:

$$\begin{aligned} V_k &= D_k^T D_k \\ \Sigma_k &= (P_k H_k^T H_k - P_k G_k^T G_k + I)^{-1} P_k \\ P_{k+1} &= (I - V_{k+1}) F_k \Sigma_k F_k^T (I - V_{k+1}) + Q_k \\ K_k &= (I - V_{k+1}) F_k \Sigma_k H_k^T \\ \hat{x}_{k+1} &= F_k \hat{x}_k + K_k (y_k - H_k \hat{x}_k) \end{aligned} \tag{12.50}$$

   (c) Verify that

$$(I - G_k P_k G_k^T) \geq 0 \tag{12.51}$$

   If not then the filter is invalid.

■ EXAMPLE 12.3

Consider a land-based vehicle that is equipped to measure its latitude and longitude (e.g., through the use of a GPS receiver). This is the same problem as that considered in Example 7.12. The vehicle dynamics and measurements can be approximated by the following equations:

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 0 \\ T\sin\alpha \\ T\cos\alpha \end{bmatrix} u_k + w_k + \delta_k \\ y'_k &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_k + v'_k \end{aligned} \tag{12.52}$$

The first two components of x_k are the latitude and longitude positions, the last two components of x_k are the latitude and longitude velocities, w_k represents zero-mean process noise due to potholes and other disturbances, δ_k is additional unknown process noise, and u_k is the commanded acceleration. T is the sample period of the system, and α is the heading angle (measured counterclockwise from due east). The measurement y'_k consists of latitude and longitude, and v'_k is the measurement noise. Suppose the standard deviations of the measurement noises are known to be σ_1 and σ_2. Then we must normalize our measurement equation to satisfy the condition that the measurement noise has a unity covariance. We therefore define the normalized measurement y_k as

$$y_k = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}^{-1} y'_k \tag{12.53}$$

In our simulation we set the covariances of the process and measurement noise as follows:

$$\begin{aligned} Q &= \mathrm{Diag}(4\ \mathrm{m}^2, 4\ \mathrm{m}^2, 1\ (\mathrm{m/s})^2, 1\ (\mathrm{m/s})^2) \\ R &= \mathrm{Diag}(\sigma_1^2, \sigma_2^2) = \mathrm{Diag}(900\ \mathrm{m}^2, 900\ \mathrm{m}^2) \end{aligned} \tag{12.54}$$

We can use an H_∞ filter to estimate the position of the vehicle. There may be times when the vehicle is traveling off-road, or on an unknown road, in which case the problem is unconstrained. At other times it may be known that the vehicle is traveling on a given road, in which case the state estimation problem is constrained. For instance, if it is known that the vehicle is traveling on a straight road with a heading of α then the matrix D_k and the vector d_k of Equation (12.32) can be given as follows:

$$\begin{aligned} D_k &= \begin{bmatrix} 1 & -\tan\alpha & 0 & 0 \\ 0 & 0 & 1 & -\tan\alpha \end{bmatrix} \\ d_k &= \begin{bmatrix} 0 & 0 \end{bmatrix}^T \end{aligned} \tag{12.55}$$

We can enforce the condition $D_k D_k^T = I$ by dividing D_k by $\sqrt{1+\tan^2\alpha}$. In our simulation we set the sample period T to 1 s and the heading angle α to a constant 60 degrees. The commanded acceleration is toggled between

± 10 m/s^2, as if the vehicle were accelerating and decelerating in traffic. The initial conditions are set to

$$x_0 = \begin{bmatrix} 0 & 0 & 173 & 100 \end{bmatrix}^T \tag{12.56}$$

We found via tuning that a G_k matrix of θI, with $\theta = 1/40$, gave good filter performance. Smaller values of θ make the H_∞ filter perform like a Kalman filter. Larger values of θ prevent the H_∞ filter from finding a solution as the positive definite conditions in Equations (12.44) and (12.45) are not satisfied.

This example could be solved by reducing the system-model parameterization [Wen92], or by introducing artificial perfect measurements into the problem [Hay98, Por88]. In fact, those methods could be used for any estimation problem with equality constraints. However, those methods cannot be extended to inequality constraints, whereas the method discussed in this section can be extended to inequality constraints, as discussed earlier.

The unconstrained and constrained H_∞ filters were simulated 100 times each, and the average RMS position and estimation error magnitudes at each time step are plotted in Figure 12.4. It can be seen that the constrained filter results in more accurate estimates. The unconstrained estimator results in position errors that average 35.3 m, whereas the constrained estimator gives position errors that average about 27.1 m. The unconstrained velocity estimation error is 12.9 m/s, whereas the constrained velocity estimation error is 10.9 m/s.

Table 12.1 shows a comparison of the unconstrained and constrained Kalman and H_∞ filters when the noise statistics are nominal. Table 12.2 shows a comparison of the unconstrained and constrained Kalman and H_∞ filters when the acceleration noise on the system has a bias of 1 m/s^2 in both the north and east directions. In both situations, the H_∞ filter estimates position more accurately, but the Kalman filter estimates velocity more accurately. In the off-nominal noise case, the advantage of the H_∞ filter over the Kalman filter for position estimation is more pronounced than when the noise is nominal.

Table 12.1 Example 12.3 estimation errors (averaged over 100 Monte Carlo simulations) of the unconstrained and constrained Kalman and H_∞ filters with nominal noise statistics. The H_∞ filters perform better for position estimation, and the Kalman filters perform better for velocity estimation. Position errors are in units of meters, and velocity errors are in units of meters/second.

	Kalman		H_∞	
	Pos.	Vel.	Pos.	Vel.
Unconstrained	40.3	12.4	35.3	12.9
Constrained	33.2	10.4	27.1	10.9

$\nabla\nabla\nabla$

Table 12.2 Example 12.3 estimation errors (averaged over 100 Monte Carlo simulations) of the unconstrained and constrained Kalman and H_∞ filters with off-nominal noise statistics. The H_∞ filters perform better for position estimation, and the Kalman filters perform better for velocity estimation. Position errors are in units of meters, and velocity errors are in units of meters/second.

	Kalman		H_∞	
	Pos.	Vel.	Pos.	Vel.
Unconstrained	60.8	19.2	45.9	20.6
Constrained	56.2	17.6	39.1	19.1

Figure 12.4 Example 12.3 unconstrained and constrained H_∞ filter estimation-error magnitudes. The plots show the average estimation-error magnitudes of 100 Monte Carlo simulations when the noise statistics are nominal.

12.4 SUMMARY

In this chapter we briefly introduced some advanced topics in the area of H_∞ filtering. We discussed an approach for minimizing a combination of the Kalman and H_∞ filter performance indices. This provides a way to balance the excessive optimism of the Kalman filter with the excessive pessimism of the H_∞ filter. We also looked at the robust mixed Kalman/H_∞ estimation problem, where we took system-model uncertainties into account. This is an important problem because (in practice) the system model is never perfectly known. Finally we discussed constrained H_∞ filtering, in which equality (or inequality) constraints are enforced on the state estimate. This can improve filter performance in cases in which we know that the state must satisfy certain constraints.

There is still a lot of room for additional work and development in H_∞ filtering. For example, reduced-order H_∞ filtering tries to obtain good minimax estimation performance with a filter whose order is less than that of the underlying system.

Reduced-order Kalman filtering was discussed in Section 10.3, and reduced-order H_∞ filtering is considered in [Bet94, Gri97, Xu02]. The approach taken in [Ko06] for constrained Kalman filtering may be applicable to constrained H_∞ filtering and may give better results than the method discussed in this chapter. The use of Krein space approaches for solving various H_∞ filtering problems is promising [Has96a, Has96b]. H_∞ smoothing is discussed in [Gri91a, The94b, Has99, Zha05a], and robust H_∞ smoothing is discussed in [The94a]. An information form for the H_∞ filter (analogous to the Kalman information filter discussed in Section 6.2) is presented in [Zha05b]. Approaches to dealing with delayed measurements and synchronization errors have been extensively explored for Kalman filters (see Section 10.5), but are notably absent in the H_∞ filter literature. There has been a lot of work on nonlinear Kalman filtering (see Chapters 13–15), but not nearly as much on nonlinear H_∞ filtering.

PROBLEMS

Written exercises

12.1 Consider the system described in Example 12.1 with $Q = R = 1$.

a) Find the steady-state *a priori* estimation-error variance P as a function of the estimator gain K.

b) Find $||G_{\tilde{x}e}||_\infty^2$, the square of the infinity-norm of the transfer function from the noise w and v to the *a priori* state estimation error $\tilde{x}$, as a function of the estimator gain K.

c) Find the estimator gain K that minimizes $(P + ||G_{\tilde{x}e}||_\infty^2)$.

12.2 Verify that if $\theta = 0$, the Riccati equation associated with the mixed Kalman/H_∞ filter in Equation (12.6) reduces to the Riccati equation associated with the Kalman filter.

12.3 Suppose that the hybrid filter gain of Equation (12.13) is used for the system of Example 12.1 with $\theta = 1/2$. For what values of d will the hybrid filter be stable?

12.4 Suppose that the robust filter of Section 12.2 is used for a system with n states and r measurements. What are the dimensions of M_1, M_2, Γ, and N?

12.5 Suppose that a system matrix is given as

$$F = \begin{bmatrix} 0.4 \pm 0.2 & 0.4 \\ -0.4 & 1 \end{bmatrix}$$

(Note that this is the system matrix of Example 4.1 in case the effect of overcrowding on the predator population is uncertain.) Give an M_1 and N matrix that satisfy Equation (12.15) for this uncertainty.

12.6 Consider an uncertain system with $F = -1$, $H = 1$, $Q = R = 1$, $M_1 = 1/5$, $M_2 = 0$, and $N = 1$. Suppose that $\epsilon = 0$ is used to design a robust mixed Kalman/H_∞ filter.

a) For what values of α will the steady-state value of $\tilde{P}$ in Equation (12.22) be real and positive?

b) For what values of α will the steady-state value of $\tilde{P}$ satisfy the second condition of Equation (12.24)?

12.7 Consider a constrained H_∞ state estimation problem with

$$\begin{aligned} F &= \begin{bmatrix} 1 & 1/2 \\ 1/2 & 0 \end{bmatrix} \\ G = H &= \begin{bmatrix} G_1 & 0 \end{bmatrix} \\ D &= \begin{bmatrix} 1 & 1 \end{bmatrix} \\ Q &= \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} \end{aligned}$$

Find the steady-state constrained Riccati solution for P from Equation (12.50). For what values of G_1 will the condition of Equation (12.51) be satisfied?

Computer exercises

12.8 Consider a two-state Newtonian system as discussed in Example 9.1 with $T = 1$, $a = 1$, and $R = 1$.

a) What is the steady-state Kalman gain?

b) What is the maximum θ for which the H_∞ estimator exists? Answer to the nearest 0.01. What is the H_∞ gain for this value of θ?

c) What is the H_∞ gain when $\theta = 0.5$? Plot the maximum estimator eigenvalue magnitude as a function of d for the hybrid filter of Equation (12.13) when $\theta = 0.5$.

12.9 Implement the time-varying Riccati equations for the robust mixed Kalman/H_∞ filter for $F = 1/2$, $H = Q = R = 1$, $M_1 = 1/4$, $M_2 = 0$, $N = 1$, $\epsilon = 0$, $\theta = 1/10$, and $S_1 = S_2 = 1$.

a) At what time do the conditions of Equation (12.24) fail to be satisfied when $\alpha = 2$? Repeat for $\alpha = 3$, 4, 5, and 6.

b) What is the steady-state theoretical bound on the estimation error when $\alpha = 10$? Repeat for $\alpha = 20$, 30, and 40.

12.10 Consider a constrained H_∞ state estimation problem with

$$\begin{aligned} F &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \\ H &= \begin{bmatrix} 1 & 0 \end{bmatrix} \\ G &= \begin{bmatrix} G_1 & 0 \end{bmatrix} \\ D &= \begin{bmatrix} 1 & 1 \end{bmatrix} \\ Q &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Implement the Σ_k and P_k expressions from Equation (12.50).

a) What is the largest value of G_1 for which P_k reaches a positive definite steady-state solution that satisfies the condition given in Equation (12.51)? Answer to the nearest 0.01. What is the resulting steady-state value of P?

b) Set G_1 equal to 1% of the maximum G_1 that you found in part (a). What is the new steady-state value of P? Give an intuitive explanation for why P gets smaller when G_1 gets smaller.

PART IV

NONLINEAR FILTERS

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

CHAPTER 13

Nonlinear Kalman filtering

> It appears that no particular approximate [nonlinear] filter is consistently better than any other, though ... any nonlinear filter is better than a strictly linear one.
>
> —Lawrence Schwartz and Edwin Stear [Sch68]

All of our discussion to this point has considered linear filters for linear systems. Unfortunately, linear systems do not exist. All systems are ultimately nonlinear. Even the simple $I = V/R$ relationship of Ohm's Law is only an approximation over a limited range. If the voltage across a resistor exceeds a certain threshold, then the linear approximation breaks down. Figure 13.1 shows a typical relationship between the current through a resistor and the voltage across the resistor. At small input voltages the relationship is approximately linear, but if the power dissipated by the resistor exceeds some threshold then the relationship becomes highly nonlinear. Even a device as simple as a resistor is only approximately linear, and even then only in a limited range of operation.

So we see that linear systems do not really exist. However, many systems are close enough to linear that linear estimation approaches give satisfactory results. But "close enough" can only be carried so far. Eventually, we run across a system that does not behave linearly even over a small range of operation, and our linear approaches for estimation no longer give good results. In this case, we need to explore nonlinear estimators.

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

Figure 13.1 Typical current/voltage relationship for a resistor. The relationship is linear for a limited range of operation, but becomes highly nonlinear beyond that range.

Nonlinear filtering can be a difficult and complex subject. It is certainly not as mature, cohesive, or well understood as linear filtering. There is still a lot of room for advances and improvement in nonlinear estimation techniques. However, some nonlinear estimation methods have become (or are becoming) widespread. These techniques include nonlinear extensions of the Kalman filter, unscented filtering, and particle filtering.

In this chapter, we will discuss some nonlinear extensions of the Kalman filter. The Kalman filter that we discussed earlier in this book directly applies only to linear systems. However, a nonlinear system can be linearized as discussed in Section 1.3, and then linear estimation techniques (such as the Kalman or H_∞ filter) can be applied. This chapter discusses those types of approaches to nonlinear Kalman filtering.

In Section 13.1, we will discuss the linearized Kalman filter. This will involve finding a linear system whose states represent the deviations from a nominal trajectory of a nonlinear system. We can then use the Kalman filter to estimate the deviations from the nominal trajectory, and hence obtain an estimate of the states of the nonlinear system. In Section 13.2, we will extend the linearized Kalman filter to directly estimate the states of a nonlinear system. This filter, called the extended Kalman filter (EKF), is undoubtedly the most widely used nonlinear state estimation technique that has been applied in the past few decades. In Section 13.3, we will discuss "higher-order" approaches to nonlinear Kalman filtering. These approaches involve more than a direct linearization of the nonlinear system, hence the expression "higher order." Such methods include second-order Kalman filtering, iterated Kalman filtering, sum-based Kalman filtering, and grid-based Kalman filtering. These filters provide ways to reduce the linearization errors that are inherent in the EKF. They typically provide estimation performance that is better than the EKF, but they do so at the price of higher complexity and computational expense.

Section 13.4 covers parameter estimation using Kalman filtering. Sometimes, an engineer wants to estimate the parameters of a system but does not care about estimating the states. This becomes a system identification problem. The system equations are generally nonlinear functions of the system parameters. System parameters are usually considered to be constant, or slowly time-varying, and a nonlinear Kalman filter (or any other nonlinear state estimator) can be adapted to estimate system parameters.

13.1 THE LINEARIZED KALMAN FILTER

In this section, we will show how to linearize a nonlinear system, and then use Kalman filtering theory to estimate the deviations of the state from a nominal state value. This will then give us an estimate of the state of the nonlinear system. We will derive the linearized Kalman filter from the continuous-time viewpoint, but the analogous derivation for discrete-time or hybrid systems are straightforward.

Consider the following general nonlinear system model:

$$\begin{aligned} \dot{x} &= f(x, u, w, t) \\ y &= h(x, v, t) \\ w &\sim (0, Q) \\ v &\sim (0, R) \end{aligned} \tag{13.1}$$

The system equation $f(\cdot)$ and the measurement equation $h(\cdot)$ are nonlinear functions. We will use Taylor series to expand these equations around a nominal control u_0, nominal state x_0, nominal output y_0, and nominal noise values w_0 and v_0. These nominal values (all of which are functions of time) are based on *a priori* guesses of what the system trajectory might look like. For example, if the system equations represent the dynamics of an airplane, then the nominal control, state, and output might be the planned flight trajectory. The *actual* flight trajectory will differ from this nominal trajectory due to mismodeling, disturbances, and other unforeseen effects. But the actual trajectory should be close to the nominal trajectory, in which case the Taylor series linearization should be approximately correct. The Taylor series linearization of Equation (13.1) gives

$$\begin{aligned} \dot{x} &\approx f(x_0, u_0, w_0, t) + \left.\frac{\partial f}{\partial x}\right|_0 (x - x_0) + \left.\frac{\partial f}{\partial u}\right|_0 (u - u_0) + \\ &\quad \left.\frac{\partial f}{\partial w}\right|_0 (w - w_0) \\ &= f(x_0, u_0, w_0, t) + A\Delta x + B\Delta u + L\Delta w \\ y &\approx h(x_0, v_0, t) + \left.\frac{\partial h}{\partial x}\right|_0 (x - x_0) + \left.\frac{\partial h}{\partial v}\right|_0 (v - v_0) \\ &= h(x_0, v_0, t) + C\Delta x + M\Delta v \end{aligned} \tag{13.2}$$

The definitions of the partial derivative matrices A, B, C, L, and M are apparent from the above equations. The 0 subscript on the partial derivatives means that they are evaluated at the nominal control, state, output, and noise values. The definitions of the deviations Δx, Δu, Δw, and Δv are also apparent from the above equations.

Let us assume that the nominal noise values $w_0(t)$ and $v_0(t)$ are both equal to 0 for all time. [If they are not equal to 0 then we should be able to write them as the sum of a known deterministic part and a zero-mean part, redefine the noise quantities, and rewrite Equation (13.1) so that the nominal noise values are equal to 0. See Problem 13.1]. Since $w_0(t)$ and $v_0(t)$ are both equal to 0, we see that $\Delta w(t) = w(t)$ and $\Delta v(t) = v(t)$. Further assume that the control $u(t)$ is perfectly known. In general, this is a reasonable assumption. After all, the control input $u(t)$ is determined by our control system, so there should not be any uncertainty in its value. This means that $u_0(t) = u(t)$ and $\Delta u(t) = 0$. However, in reality there may be uncertainties in the outputs of our control system because they are connected to actuators that have biases and noise. If this is the case then we can express the control as $u_0(t) + \Delta u(t)$, where $u_0(t)$ is known and $\Delta u(t)$ is a zero-mean random variable, rewrite the system equations with a perfectly known control signal, and include $\Delta u(t)$ as part of the process noise (see Problem 13.2). Now we define the nominal system trajectory as

$$\begin{aligned} \dot{x}_0 &= f(x_0, u_0, w_0, t) \\ y_0 &= h(x_0, v_0, t) \end{aligned} \tag{13.3}$$

We define the deviation of the true state derivative from the nominal state derivative, and the deviation of the true measurement from the nominal measurement, as follows:

$$\begin{aligned} \Delta\dot{x} &= \dot{x} - \dot{x}_0 \\ \Delta y &= y - y_0 \end{aligned} \tag{13.4}$$

With these definitions Equation (13.2) becomes

$$\begin{aligned} \Delta\dot{x} &= A\Delta x + Lw \\ &= A\Delta x + \tilde{w} \\ \tilde{w} &\sim (0, \tilde{Q}), \quad \tilde{Q} = LQL^T \\ \Delta y &= C\Delta x + Mv \\ &= C\Delta x + \tilde{v} \\ \tilde{v} &\sim (0, \tilde{R}), \quad \tilde{R} = MRM^T \end{aligned} \tag{13.5}$$

The above equation is a linear system with state Δx and measurement Δy, so we can use a Kalman filter to estimate Δx. The inputs to the filter consist of Δy, which is the difference between the actual measurement y and the nominal measurement y_0. The Δx that is output from the Kalman filter is an estimate of the difference between the actual state x and the nominal state x_0. The Kalman filter equations for the linearized Kalman filter are

$$\begin{aligned} \Delta\hat{x}(0) &= 0 \\ P(0) &= E\left[(\Delta x(0) - \Delta\hat{x}(0))(\Delta x(0) - \Delta\hat{x}(0))^T\right] \\ \Delta\dot{\hat{x}} &= A\Delta\hat{x} + K(\Delta y - C\Delta\hat{x}) \\ K &= PC^T\tilde{R}^{-1} \\ \dot{P} &= AP + PA^T + \tilde{Q} - PC^T\tilde{R}^{-1}CP \\ \hat{x} &= x_0 + \Delta\hat{x} \end{aligned} \tag{13.6}$$

For the Kalman filter, P is equal to the covariance of the estimation error. In the linearized Kalman filter this is no longer true because of errors that creep into the linearization of Equation (13.2). However, if the linearization errors are small then P should be approximately equal to the covariance of the estimation error. The linearized Kalman filter can be summarized as follows.

The continuous-time linearized Kalman filter

1. The system equations are given as

$$\begin{aligned} \dot{x} &= f(x, u, w, t) \\ y &= h(x, v, t) \\ w &\sim (0, Q) \\ v &\sim (0, R) \end{aligned} \tag{13.7}$$

   The nominal trajectory is known ahead of time:

$$\begin{aligned} \dot{x_0} &= f(x_0, u_0, 0, t) \\ y_0 &= h(x_0, 0, t) \end{aligned} \tag{13.8}$$

2. Compute the following partial derivative matrices evaluated at the nominal trajectory values:

$$\begin{aligned} A &= \left.\frac{\partial f}{\partial x}\right|_0 \\ L &= \left.\frac{\partial f}{\partial w}\right|_0 \\ C &= \left.\frac{\partial h}{\partial x}\right|_0 \\ M &= \left.\frac{\partial h}{\partial v}\right|_0 \end{aligned} \tag{13.9}$$

3. Compute the following matrices:

$$\begin{aligned} \tilde{Q} &= LQL^T \\ \tilde{R} &= MRM^T \end{aligned} \tag{13.10}$$

4. Define Δy as the difference between the actual measurement y and the nominal measurement y_0:

$$\Delta y = y - y_0 \tag{13.11}$$

5. Execute the following Kalman filter equations:

$$\begin{aligned} \Delta\hat{x}(0) &= 0 \\ P(0) &= E\left[(\Delta x(0) - \Delta\hat{x}(0))(\Delta x(0) - \Delta\hat{x}(0))^T\right] \\ \Delta\dot{\hat{x}} &= A\Delta\hat{x} + K(\Delta y - C\Delta\hat{x}) \\ K &= PC^T\tilde{R}^{-1} \\ \dot{P} &= AP + PA^T + \tilde{Q} - PC^T\tilde{R}^{-1}CP \end{aligned} \tag{13.12}$$

6. Estimate the state as follows:

$$\hat{x} = x_0 + \Delta\hat{x} \tag{13.13}$$

The hybrid linearized Kalman filter and the discrete-time linearized Kalman filter are not presented here, but if the development above is understood then their derivations should be straightforward.

13.2 THE EXTENDED KALMAN FILTER

The previous section obtained a linearized Kalman filter for estimating the states of a nonlinear system. The derivation was based on linearizing the nonlinear system around a nominal state trajectory. The question that arises is, How do we know the nominal state trajectory? In some cases it may not be straightforward to find the nominal trajectory. However, since the Kalman filter estimates the state of the system, we can use the Kalman filter estimate as the nominal state trajectory. This is sort of a bootstrap method. We linearize the nonlinear system around the Kalman filter estimate, and the Kalman filter estimate is based on the linearized system. This is the idea of the extended Kalman filter (EKF), which was originally proposed by Stanley Schmidt so that the Kalman filter could be applied to nonlinear spacecraft navigation problems [Bel67].

In Section 13.2.1, we will present the EKF for continuous-time systems with continuous-time measurements. In Section 13.2.2, we will present the hybrid EKF, which is the EKF for continuous-time systems with discrete-time measurements. In Section 13.2.3, we will present the EKF for discrete-time systems with discrete-time measurements.

13.2.1 The continuous-time extended Kalman filter

Combine the $\dot{x}_0$ expression in Equation (13.3) with the $\Delta\dot{\hat{x}}$ expression in Equation (13.6) to obtain

$$\dot{x}_0 + \Delta\dot{\hat{x}} = f(x_0, u_0, w_0, t) + A\Delta\hat{x} + K[y - y_0 - C(\hat{x} - x_0)] \tag{13.14}$$

Now choose $x_0(t) = \hat{x}(t)$ so that $\Delta\hat{x}(t) = 0$ and $\Delta\dot{\hat{x}}(t) = 0$. In other words, our linearization trajectory $x_0(t)$ is equal to our linearized Kalman filter estimate $\hat{x}$(t). Then the nominal measurement expression in Equation (13.3) becomes

$$\begin{aligned} y_0 &= h(x_0, v_0, t) \\ &= h(\hat{x}, v_0, t) \end{aligned} \tag{13.15}$$

and Equation (13.14) becomes

$$\dot{\hat{x}} = f(\hat{x}, u, w_0, t) + K\left[y - h(\hat{x}, v_0, t)\right] \tag{13.16}$$

This is equivalent to the linearized Kalman filter except that we have chosen $x_0 = \hat{x}$, and we have rearranged the equations to obtain $\hat{x}$ directly. The Kalman gain K is the same as that presented in Equation (13.6). But this formulation inputs the measurement y directly, and outputs the state estimate $\hat{x}$ directly. This is often referred to as the extended Kalman–Bucy filter because Richard Bucy collaborated with Rudolph Kalman in the first publication of the continuous-time Kalman filter [Kal61]. The continuous-time EKF can be summarized as follows.

The continuous-time extended Kalman filter

1. The system equations are given as

$$\begin{aligned} \dot{x} &= f(x, u, w, t) \\ y &= h(x, v, t) \\ w &\sim (0, Q) \\ v &\sim (0, R) \end{aligned} \tag{13.17}$$

2. Compute the following partial derivative matrices evaluated at the current state estimate:

$$\begin{aligned} A &= \left.\frac{\partial f}{\partial x}\right|_{\hat{x}} \\ L &= \left.\frac{\partial f}{\partial w}\right|_{\hat{x}} \\ C &= \left.\frac{\partial h}{\partial x}\right|_{\hat{x}} \\ M &= \left.\frac{\partial h}{\partial v}\right|_{\hat{x}} \end{aligned} \tag{13.18}$$

3. Compute the following matrices:

$$\begin{aligned} \tilde{Q} &= LQL^T \\ \tilde{R} &= MRM^T \end{aligned} \tag{13.19}$$

4. Execute the following Kalman filter equations:

$$\begin{aligned} \hat{x}(0) &= E[x(0)] \\ P(0) &= E\left[(x(0) - \hat{x}(0))(x(0) - \hat{x}(0))^T\right] \\ \dot{\hat{x}} &= f(\hat{x}, u, w_0, t) + K\left[y - h(\hat{x}, v_0, t)\right] \\ K &= PC^T\tilde{R}^{-1} \\ \dot{P} &= AP + PA^T + \tilde{Q} - PC^T\tilde{R}^{-1}CP \end{aligned} \tag{13.20}$$

where the nominal noise values are given as $w_0 = 0$ and $v_0 = 0$.

■ EXAMPLE 13.1

In this example, we will use the continuous-time EKF to estimate the state of a two-phase permanent magnet synchronous motor. The system equations are given in Example 1.4 and are repeated here:

$$\begin{aligned} i_a &= \frac{-R}{L}i_a + \frac{\omega\lambda}{L}\sin\theta + \frac{u_a + q_1}{L} \\ \dot{i_b} &= \frac{-R}{L}i_b - \frac{\omega\lambda}{L}\cos\theta + \frac{u_b + q_2}{L} \\ \dot{\omega} &= \frac{-3\lambda}{2J}i_a\sin\theta + \frac{3\lambda}{2J}i_b\cos\theta - \frac{F\omega}{J} + q_3 \\ \dot{\theta} &= \omega \end{aligned} \tag{13.21}$$

where i_a and i_b are the currents in the two windings, θ and ω are the angular position and velocity of the rotor, R and L are the winding resistance and inductance, λ is the flux constant, and F is the coefficient of viscous friction. The control inputs u_a and u_b consist of the applied voltages across the two windings, and J is the moment of inertia of the motor shaft and load. The state is defined as

$$x = \begin{bmatrix} i_a & i_b & \omega & \theta \end{bmatrix}^T \tag{13.22}$$

The q_i terms are process noise due to uncertainty in the control inputs (q_1 and q_2) and the load torque (q_3). The partial derivative A matrix is obtained as

$$\begin{aligned} A &= \frac{\partial f}{\partial x} \\ &= \begin{bmatrix} -R/L & 0 & \lambda s/L & x_3\lambda c/L \\ 0 & -R/L & -\lambda c/L & x_3\lambda s/L \\ -3\lambda s/2/J & 3\lambda c/2/J & -F/J & -3\lambda(x_1 c + x_2 s)/2/J \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{aligned} \tag{13.23}$$

where we have used the notation $s = \sin x_4$ and $c = \cos x_4$. Suppose that we can measure the winding currents with sense resistors so our measurement equations are

$$\begin{aligned} y(1) &= i_a + v(1) \\ y(2) &= i_b + v(2) \end{aligned} \tag{13.24}$$

where $v(1)$ and $v(2)$ are independent zero-mean white noise processes with standard deviations equal to 0.1 amps. The nominal control inputs are set to

$$\begin{aligned} u_a(t) &= \sin(2\pi t) \\ u_b(t) &= \cos(2\pi t) \end{aligned} \tag{13.25}$$

The actual control inputs are equal to the nominal values plus q_1 and q_2 (electrical noise terms), which are independent zero-mean white noise processes with standard deviations equal to 0.01 amps. The noise due to load torque disturbances (q_3) has a standard deviation of 0.5 rad/sec^2. Measurements are obtained continuously. Even though our measurements consist only of the winding currents and the system is nonlinear, we can use a continuous-time EKF (implemented in analog circuitry or very fast digital logic) to estimate the rotor position and velocity. The simulation results are shown in Figure 13.2. The four states are estimated quite well. In particular, the rotor position estimate is so good that the true and estimated rotor position traces are not distinguishable in Figure 13.2.

The P matrix quantifies the uncertainty in the state estimates. If the nonlinearities in the system and measurement are not too severe, then the P matrix should give us an idea of how accurate our estimates are. In this example, the standard deviations of the state estimation errors were obtained from the simulation and then compared with the diagonal elements of the steady-state P matrix that came out of the Kalman filter. Table 13.1 shows a comparison of the estimation errors that were determined by simulation and

Figure 13.2 Continuous extended Kalman filter simulation results for the two-phase permanent magnet synchronous motor of Example 13.1.

Table 13.1 Example 13.1 results showing one standard deviation state estimation errors determined from simulation results and determined from the P matrix of the EKF. These results are for the two-phase permanent magnet motor simulation. This table shows that the P matrix gives a good indication of the magnitude of the EKF state estimation errors.

	Simulation	P Matrix
Winding A Current	0.054 amps	0.094 Amps
Winding B Current	0.052 amps	0.094 Amps
Speed	0.26 rad/sec	0.44 rad/sec
Position	0.013 rad	0.025 rad

the theoretical estimation errors based on the P matrix. We see that the P matrix gives a good indication of the magnitude of the estimation errors.

$\nabla\nabla\nabla$

13.2.2 The hybrid extended Kalman filter

Many real engineering systems are governed by continuous-time dynamics whereas the measurements are obtained at discrete instants of time. In this section, we will derive the hybrid EKF, which considers systems with continuous-time dynamics and discrete-time measurements. This is the most common situation encountered in practice.

Suppose we have a continuous-time system with discrete-time measurements as follows:

$$\begin{aligned} \dot{x} &= f(x, u, w, t) \\ y_k &= h_k(x_k, v_k) \\ w(t) &\sim (0, Q) \\ v_k &\sim (0, R_k) \end{aligned} \tag{13.26}$$

The process noise $w(t)$ is continuous-time white noise with covariance Q, and the measurement noise v_k is discrete-time white noise with covariance R_k. Between measurements we propagate the state estimate according to the known nonlinear dynamics, and we propagate the covariance as derived in the continuous-time EKF of Section 13.2.1 using Equation (13.20). Recall that the $\dot{P}$ expression from Equation (13.20) is given as

$$\dot{P} = AP + PA^T + LQL^T - PC^T(MRM^T)^{-1}CP \tag{13.27}$$

In the hybrid EKF, we should not include the R term in the $\dot{P}$ equation because we are integrating P between measurement times, during which we do not have any measurements. Another way of looking at it is that in between measurement times we have measurements with infinite covariance ($R = \infty$), so the last term on the right side of the $\dot{P}$ equation goes to zero. This gives us the following for the time-update equations of the hybrid EKF:

$$\begin{aligned} \dot{\hat{x}} &= f(\hat{x}, u, w_0, t) \\ \dot{P} &= AP + PA^T + LQL^T \end{aligned} \tag{13.28}$$

where A and L are given in Equation (13.18). The above equations propagate $\hat{x}$ from $\hat{x}_{k-1}^+$ to $\hat{x}_k^-$, and P from P_{k-1}^+ to P_k^-. Note that w_0 is the nominal process noise in the above equation; that is, $w_0(t) = 0$.

At each measurement time, we update the state estimate and the covariance as derived in the discrete-time Kalman filter (Chapter 5):

$$\begin{aligned} K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k[y_k - h_k(\hat{x}_k^-, v_0, t_k)] \\ P_k^+ &= (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k M_k R_k M_k^T K_k^T \end{aligned} \tag{13.29}$$

where v_0 is the nominal measurement noise; that is, $v_0 = 0$. H_k is the partial derivative of $h_k(x_k, v_k)$ with respect to x_k, and M_k is the partial derivative of $h_k(x_k, v_k)$ with respect to v_k. H_k and M_k are evaluated at $\hat{x}_k^-$.

Note that P_k and K_k cannot be computed offline because they depend on H_k and M_k, which depend on $\hat{x}_k^-$, which in turn depends on the noisy measurements. Therefore, a steady-state solution does not (in general) exist to the extended Kalman filter. However, some efforts at obtaining steady-state approximations to the extended Kalman filter have been reported in [Saf78].

The hybrid EKF can be summarized as follows.

The hybrid extended Kalman filter

1. The system equations with continuous-time dynamics and discrete-time measurements are given as follows:

$$\begin{aligned} \dot{x} &= f(x, u, w, t) \\ y_k &= h_k(x_k, v_k) \\ w(t) &\sim (0, Q) \\ v_k &\sim (0, R_k) \end{aligned} \tag{13.30}$$

2. Initialize the filter as follows:

$$\begin{aligned} \hat{x}_0^+ &= E[x_0] \\ P_0^+ &= E\left[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T\right] \end{aligned} \tag{13.31}$$

3. For $k = 1, 2, \cdots$, perform the following.

   (a) Integrate the state estimate and its covariance from time $(k-1)^+$ to time k^- as follows:

$$\begin{aligned} \dot{\hat{x}} &= f(\hat{x}, u, 0, t) \\ \dot{P} &= AP + PA^T + LQL^T \end{aligned} \tag{13.32}$$

   where F and L are given in Equation (13.18). We begin this integration process with $\hat{x} = \hat{x}_{k-1}^+$ and $P = P_{k-1}^+$. At the end of this integration we have $\hat{x} = \hat{x}_k^-$ and $P = P_k^-$.

   (b) At time k, incorporate the measurement y_k into the state estimate and estimation covariance as follows:

$$\begin{aligned} K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - h_k(\hat{x}_k^-, 0, t_k)) \\ P_k^+ &= (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k M_k R_k M_k^T K_k^T \end{aligned} \tag{13.33}$$

   H_k and M_k are the partial derivatives of $h_k(x_k, v_k)$ with respect to x_k and v_k, and are both evaluated at $\hat{x}_k^-$. Note that other equivalent expressions can be used for K_k and P_k^+, as is apparent from Equation (5.19).

■ EXAMPLE 13.2

In this example, we will use the continuous-time EKF and the hybrid EKF to estimate the altitude x_1, velocity x_2, and constant ballistic coefficient $1/x_3$ of a body as it falls toward earth. A range-measuring device measures the altitude of the falling body. This example (or a variant thereof) is given in several places, for example [Ath68, Ste94, Jul00]. The equations for this system are

$$\begin{aligned} \dot{x}_1 &= x_2 + w_1 \\ \dot{x}_2 &= \rho_0 \exp(-x_1/k) x_2^2 / 2x_3 - g + w_2 \\ \dot{x}_3 &= w_3 \\ y &= x_1 + v \end{aligned} \tag{13.34}$$

As usual, w_i is the noise that affects the ith process equation, and v is the measurement noise. ρ_0 is the air density at sea level, k is a constant that defines the relationship between air density and altitude, and g is the acceleration due to gravity. The partial derivative matrices for this system are given as follows:

$$\begin{aligned}
A &= \frac{\partial f}{\partial x} \\
&= \begin{bmatrix} 0 & 1 & 0 \\ A_{21} & A_{22} & A_{23} \\ 0 & 0 & 0 \end{bmatrix} \\
A_{21} &= -\rho_0 \exp(-x_1/k) x_2^2 / 2k x_3 \\
A_{22} &= \rho_0 \exp(-x_1/k) x_2 / x_3 \\
A_{23} &= -\rho_0 \exp(-x_1/k) x_2^2 / 2x_3^2 \\
C = H &= \frac{\partial h}{\partial x} \\
&= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}
\end{aligned} \tag{13.35}$$

We will use the continuous-time system equations to simulate the system. For the hybrid system we suppose that we obtain range measurements every 0.5 seconds. The constants that we will use are given as

$$\begin{aligned}
\rho_0 &= 0.0034 \text{ lb-sec}^2/\text{ft}^4 \\
g &= 32.2 \text{ ft/sec}^2 \\
k &= 22000 \text{ ft} \\
E[v^2(t)] &= 100 \text{ ft}^2 \\
E[w_i^2(t)] &= 0 \quad (i = 1, 2, 3)
\end{aligned} \tag{13.36}$$

The initial conditions of the system and the estimator are given as

$$\begin{aligned}
x_0 &= \begin{bmatrix} 100,000 & -6,000 & 1/2,000 \end{bmatrix}^T \\
\hat{x}_0^+ &= \begin{bmatrix} 100,010 & -6,100 & 1/2,500 \end{bmatrix}^T \\
P_0^+ &= \begin{bmatrix} 500 & 0 & 0 \\ 0 & 20,000 & 0 \\ 0 & 0 & 1/250,000 \end{bmatrix}
\end{aligned} \tag{13.37}$$

We use rectangular integration with a step size of 0.4 msec to simulate the system, the continuous-time EKF, and the hybrid EKF (with a measurement time of 0.5 sec). Figure 13.3 shows estimation-error magnitudes averaged over 100 simulations for the altitude, velocity, and ballistic coefficient reciprocal of the falling body. We see that the continuous-time EKF appears to perform better in general than the hybrid EKF. This is to be expected since more measurements are incorporated in the continuous-time EKF. The RMS estimation errors averaged over 100 simulations was 2.8 feet for the continuous-time EKF and 5.1 feet for the hybrid EKF for altitude estimation, 1.2 feet/s for the continuous-time EKF and 2.0 feet/s for the hybrid EKF for velocity estimation, and 213 for the continuous-time EKF and 246 for the hybrid EKF

for the reciprocal of ballistic coefficient estimation. Of course, a continuous-time EKF (in analog hardware) would be more difficult to implement, tune, and modify than a hybrid EKF (in digital hardware).

Figure 13.3 Example 13.2 altitude, velocity, and ballistic coefficient reciprocal estimation-error magnitudes of a falling body averaged over 100 simulations. The continuous-time EKF generally performs better than the hybrid EKF.

∇∇∇

13.2.3 The discrete-time extended Kalman filter

In this section, we will derive the discrete-time EKF, which considers discrete-time dynamics and discrete-time measurements. This situation is often encountered in practice. Even if the underlying system dynamics are continuous time, the EKF usually needs to be implemented in a digital computer. This means that there might not be enough computational power to integrate the system dynamics as required in a continuous-time EKF or a hybrid EKF. So the dynamics are often discretized (see Section 1.4) and then a discrete-time EKF can be used.

Suppose we have the system model

$$\begin{aligned} x_k &= f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}) \\ y_k &= h_k(x_k, v_k) \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \end{aligned} \tag{13.38}$$

We perform a Taylor series expansion of the state equation around $x_{k-1} = \hat{x}_{k-1}^+$ and $w_{k-1} = 0$ to obtain the following:

$$
\begin{aligned}
x_k &= f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}, 0) + \left.\frac{\partial f_{k-1}}{\partial x}\right|_{\hat{x}_{k-1}^+} (x_{k-1} - \hat{x}_{k-1}^+) + \left.\frac{\partial f_{k-1}}{\partial w}\right|_{\hat{x}_{k-1}^+} w_{k-1} \\
&= f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}, 0) + F_{k-1}(x_{k-1} - \hat{x}_{k-1}^+) + L_{k-1} w_{k-1} \\
&= F_{k-1} x_{k-1} + \left[f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}, 0) - F_{k-1}\hat{x}_{k-1}^+\right] + L_{k-1} w_{k-1} \\
&= F_{k-1} x_{k-1} + \tilde{u}_{k-1} + \tilde{w}_{k-1} \qquad (13.39)
\end{aligned}
$$

F_{k-1} and L_{k-1} are defined by the above equation. The known signal $\tilde{u}_k$ and the noise signal $\tilde{w}_k$ are defined as follows:

$$
\begin{aligned}
\tilde{u}_k &= f_k(\hat{x}_k^+, u_k, 0) - F_k \hat{x}_k^+ \\
\tilde{w}_k &\sim (0, L_k Q_k L_k^T) \qquad (13.40)
\end{aligned}
$$

We linearize the measurement equation around $x_k = \hat{x}_k^-$ and $v_k = 0$ to obtain

$$
\begin{aligned}
y_k &= h_k(\hat{x}_k^-, 0) + \left.\frac{\partial h_k}{\partial x}\right|_{\hat{x}_k^-} (x_k - \hat{x}_k^-) + \left.\frac{\partial h_k}{\partial v}\right|_{\hat{x}_k^-} v_k \\
&= h_k(\hat{x}_k^-, 0) + H_k(x_k - \hat{x}_k^-) + M_k v_k \\
&= H_k x_k + \left[h_k(\hat{x}_k^-, 0) - H_k \hat{x}_k^-\right] + M_k v_k \\
&= H_k x_k + z_k + \tilde{v}_k \qquad (13.41)
\end{aligned}
$$

H_k and M_k are defined by the above equation. The known signal z_k and the noise signal $\tilde{v}_k$ are defined as

$$
\begin{aligned}
z_k &= h_k(\hat{x}_k^-, 0) - H_k \hat{x}_k^- \\
\tilde{v}_k &\sim (0, M_k R_k M_k^T) \qquad (13.42)
\end{aligned}
$$

We have a linear state-space system in Equation (13.39) and a linear measurement in Equation (13.41). That means we can use the standard Kalman filter equations to estimate the state. This results in the following equations for the discrete-time extended Kalman filter.

$$
\begin{aligned}
P_k^- &= F_{k-1} P_{k-1}^+ F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T \\
K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1} \\
\hat{x}_k^- &= f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}, 0) \\
z_k &= h_k(\hat{x}_k^-, 0) - H_k \hat{x}_k^- \\
\hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k \hat{x}_k^- - z_k) \\
&= \hat{x}_k^- + K_k[y_k - h_k(\hat{x}_k^-, 0)] \\
P_k^+ &= (I - K_k H_k) P_k^- \qquad (13.43)
\end{aligned}
$$

The discrete-time EKF can be summarized as follows.

The discrete-time extended Kalman filter

1. The system and measurement equations are given as follows:

$$\begin{aligned} x_k &= f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}) \\ y_k &= h_k(x_k, v_k) \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \end{aligned} \tag{13.44}$$

2. Initialize the filter as follows:

$$\begin{aligned} \hat{x}_0^+ &= E(x_0) \\ P_0^+ &= E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T] \end{aligned} \tag{13.45}$$

3. For $k = 1, 2, \cdots$, perform the following.

   (a) Compute the following partial derivative matrices:

$$\begin{aligned} F_{k-1} &= \left.\frac{\partial f_{k-1}}{\partial x}\right|_{\hat{x}_{k-1}^+} \\ L_{k-1} &= \left.\frac{\partial f_{k-1}}{\partial w}\right|_{\hat{x}_{k-1}^+} \end{aligned} \tag{13.46}$$

   (b) Perform the time update of the state estimate and estimation-error covariance as follows:

$$\begin{aligned} P_k^- &= F_{k-1}P_{k-1}^+F_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T \\ \hat{x}_k^- &= f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}, 0) \end{aligned} \tag{13.47}$$

   (c) Compute the following partial derivative matrices:

$$\begin{aligned} H_k &= \left.\frac{\partial h_k}{\partial x}\right|_{\hat{x}_k^-} \\ M_k &= \left.\frac{\partial h_k}{\partial v}\right|_{\hat{x}_k^-} \end{aligned} \tag{13.48}$$

   (d) Perform the measurement update of the state estimate and estimation-error covariance as follows:

$$\begin{aligned} K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k[y_k - h_k(\hat{x}_k^-, 0)] \\ P_k^+ &= (I - K_k H_k) P_k^- \end{aligned} \tag{13.49}$$

   Note that other equivalent expressions can be used for K_k and P_k^+, as is apparent from Equation (5.19).

13.3 HIGHER-ORDER APPROACHES

More refined linearization techniques can be used to reduce the linearization error in the EKF for highly nonlinear systems. In this section, we will derive and illustrate two such approaches: the iterated EKF, and the second-order EKF. We will also briefly discuss other approaches, including Gaussian sum filters and grid filters.

13.3.1 The iterated extended Kalman filter

In this section, we will discuss the iterated EKF. We will confine our discussion here to discrete-time filtering, although the concepts can easily be extended to continuous or hybrid filters.

When we derived the discrete-time EKF in Section 13.2.3, we approximated $h(x_k, v_k)$ by expanding it in a Taylor series around $\hat{x}_k^-$, as shown in Equation (13.41):

$$\begin{aligned} h(x_k, v_k) &= h(\hat{x}_k^-, 0) + \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_k^-} (x_k - \hat{x}_k^-) + \left.\frac{\partial h}{\partial v}\right|_{\hat{x}_k^-} v_k \\ &= h(\hat{x}_k^-, 0) + H_k(x_k - \hat{x}_k^-) + M_k v_k \end{aligned} \tag{13.50}$$

Based on this linearization, we then wrote the measurement-update equations as shown in Equation (13.43):

$$\begin{aligned} K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1} \\ P_k^+ &= (I - K_k H_k) P_k^- \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k [y_k - h_k(\hat{x}_k^-, 0)] \end{aligned} \tag{13.51}$$

The reason that we expanded $h(x_k)$ around $\hat{x}_k^-$ was because that was our best estimate of x_k before the measurement at time k is taken into account. But after we implement the discrete EKF equations to obtain the *a posteriori* estimate $\hat{x}_k^+$, we have a better estimate of x_k. So we can reduce the linearization error by reformulating the Taylor series expansion of $h(x_k)$ around our new estimate. If we then use that new Taylor series expansion of $h(x_k)$ and recalculate the measurement-update equations, we should get a better a posteriori estimate of $\hat{x}_k^+$. But then we can repeat the previous step; since we have an even better estimate of x_k, we can again reformulate the expansion of $h(x_k)$ around this even better estimate to get an even *better* estimate. This process can be repeated as many times as desired, although for most problems the majority of the possible improvement is obtained by only relinearizing one time.

We use the notation $\hat{x}_{k,i}^+$ to refer to the *a posteriori* estimate of x_k after i relinearizations have been performed. So $\hat{x}_{k,0}$ is the *a posteriori* estimate that results from the application of the standard EKF. Likewise, we use $P_{k,i}^+$ to refer to the approximate estimation-error covariance of $\hat{x}_{k,i}^+$, $K_{k,i}$ to refer to the Kalman gain that is used during the ith relinearization step, and $H_{k,i}$ to refer to the partial derivative matrix evaluated at the $x_k = \hat{x}_{k,i}^+$.

With this notation, we can describe an algorithm for the iterated EKF as follows. First, at each time step k we initialize the iterated EKF estimate to the standard EKF estimate:

$$\begin{aligned} \hat{x}_{k,0}^+ &= \hat{x}_k^+ \\ P_{k,0}^+ &= P_k^+ \end{aligned} \tag{13.52}$$

Second, for $i = 0, 1, \cdots, N$, evaluate the following equations:

$$\begin{aligned} H_{k,i} &= \left.\frac{\partial h}{\partial x}\right|_{\hat{x}^+_{k,i}} \\ K_{k,i} &= P^-_k H^T_{k,i}(H_{k,i}P^-_k H^T_{k,i} + M_k R_k M^T_k)^{-1} \\ P^+_{k,i+1} &= (I - K_{k,i}H_{k,i})P^-_k \\ \hat{x}^+_{k,i+1} &= \hat{x}^-_k + K_{k,i}[y_k - h_k(\hat{x}^-_k)] \end{aligned} \tag{13.53}$$

This is done for as many steps as desired to improve the linearization. If $N = 0$ then the iterated EKF reduces to the standard EKF.

We still have to make one more modification to the above equations to obtain the iterated Kalman filter. Recall that in the derivation of the EKF, the $\hat{x}$ measurement update equation was originally derived from the following first-order Taylor series expansion of the measurement equation:

$$\begin{aligned} y_k &= h(x_k) \\ &\approx h(\hat{x}^-_k) + H|_{\hat{x}^-_k}(x_k - \hat{x}^-_k) \end{aligned} \tag{13.54}$$

To derive the measurement-update equation for $\hat{x}$ we evaluated the right side at the *a priori* estimate $\hat{x}^-_k$ and subtracted from y_k to get our correction term (the residual):

$$\begin{aligned} r_k &= y_k - h(\hat{x}^-_k) - H|_{\hat{x}^-_k}(\hat{x}^-_k - \hat{x}^-_k) \\ &= y_k - h(\hat{x}^-_k) \end{aligned} \tag{13.55}$$

With the iterated EKF we instead want to expand the measurement equation around $\hat{x}^+_{k,i}$ as follows:

$$y_k \approx h(\hat{x}^+_{k,i}) + H|_{\hat{x}^+_{k,i}}(x_k - \hat{x}^+_{k,i}) \tag{13.56}$$

To derive the iterated EKF measurement-update equation for $\hat{x}$, we evaluate the right side of the above equation at the *a priori* estimate $\hat{x}^-_k$ and subtract from y_k to get our correction term:

$$r_k = y_k - h(\hat{x}^+_{k,i}) - H_{k,i}(\hat{x}^-_k - \hat{x}^+_{k,i}) \tag{13.57}$$

This gives the iterated EKF update equation for $\hat{x}$ as

$$\hat{x}^+_{k,i+1} = \hat{x}^-_k + K_{k,i}[y_k - h(\hat{x}^+_{k,i}) - H_{k,i}(\hat{x}^-_k - \hat{x}^+_{k,i})] \tag{13.58}$$

The iterated EKF can then be summarized as follows.

The iterated extended Kalman filter

1. The nonlinear system and measurement equations are given as follows:

$$\begin{aligned} x_k &= f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}) \\ y_k &= h_k(x_k, v_k) \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \end{aligned} \tag{13.59}$$

2. Initialize the filter as follows.

$$\begin{aligned} \hat{x}_0^+ &= E(x_0) \\ P_0^+ &= E\left[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\right] \end{aligned} \tag{13.60}$$

3. For $k = 1, 2, \cdots$, do the following.

   (a) Perform the following time-update equations:

$$\begin{aligned} P_k^- &= F_{k-1} P_{k-1}^+ F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T \\ \hat{x}_k^- &= f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}, 0) \end{aligned} \tag{13.61}$$

   where the partial derivative matrices F_{k-1} and L_{k-1} are defined as follows:

$$\begin{aligned} F_{k-1} &= \left.\frac{\partial f_{k-1}}{\partial x}\right|_{\hat{x}_{k-1}^+} \\ L_{k-1} &= \left.\frac{\partial f_{k-1}}{\partial w}\right|_{\hat{x}_{k-1}^+} \end{aligned} \tag{13.62}$$

   Up to this point the iterated EKF is the same as the standard discrete-time EKF.

   (b) Perform the measurement update by initializing the iterated EKF estimate to the standard EKF estimate:

$$\begin{aligned} \hat{x}_{k,0}^+ &= \hat{x}_k^- \\ P_{k,0}^+ &= P_k^- \end{aligned} \tag{13.63}$$

   For $i = 0, 1, \cdots, N$, evaluate the following equations (where N is the desired number of measurement-update iterations):

$$\begin{aligned} H_{k,i} &= \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_{k,i}^+} \\ M_{k,i} &= \left.\frac{\partial h}{\partial v}\right|_{\hat{x}_{k,i}^+} \\ K_{k,i} &= P_k^- H_{k,i}^T \left(H_{k,i} P_k^- H_{k,i}^T + M_{k,i} R_k M_{k,i}^T\right)^{-1} \\ P_{k,i+1}^+ &= (I - K_{k,i} H_{k,i}) P_k^- \\ \hat{x}_{k,i+1}^+ &= \hat{x}_k^- + K_{k,i}\left[y_k - h(\hat{x}_{k,i}^+) - H_{k,i}(\hat{x}_k^- - \hat{x}_{k,i}^+)\right] \end{aligned} \tag{13.64}$$

   (c) The final *a posteriori* state estimate and estimation-error covariance are given as follows:

$$\begin{aligned} \hat{x}_k^+ &= \hat{x}_{k,N+1}^+ \\ P_k^+ &= P_{k,N+1}^+ \end{aligned} \tag{13.65}$$

An illustration of the iterated EKF will be presented in Example 13.3.

13.3.2 The second-order extended Kalman filter

The second-order EKF is similar to the iterated EKF in that it attempts to reduce the linearization error of the EKF. In the iterated EKF of the previous section, we refined the point at which we performed a first-order Taylor series expansion of the measurement equation $h(\cdot)$. In the second-order EKF we instead perform a *second-order* Taylor series expansion of $f(\cdot)$ and $h(\cdot)$. The second-order EKF presented in this section is based on [Ath68, Gel74].

In this section, we will consider the hybrid system with continuous-time system dynamics and discrete-time measurements:

$$\begin{aligned} \dot{x} &= f(x, u, w, t) \\ y_k &= h(x_k, t_k) + v_k \\ w(t) &\sim (0, Q) \\ v_k &\sim (0, R_k) \end{aligned} \tag{13.66}$$

In the standard EKF, we expanded $f(x, u, w, t)$ using a first-order Taylor series. In this section, we will consider only the expansion around a nominal x, ignoring the expansion around nominal u and w values. This is done so that we can present the main ideas of the second-order EKF without getting too bogged down in notation. The development in this section can be easily extended to second-order expansions around u and w once the main idea is understood.

The first-order expansion of $f(x, u, w, t)$ around $x = \hat{x}$ is given as

$$f(x, u, w, t) = f(\hat{x}, u_0, w_0, t) + \left.\frac{\partial f}{\partial x}\right|_{\hat{x}} (x - \hat{x}) \tag{13.67}$$

In the standard EKF, we evaluated this expression at $x = \hat{x}$ to obtain our time-update equation for $\hat{x}$ as

$$\dot{\hat{x}} = f(\hat{x}, u_0, w_0, t) \tag{13.68}$$

In the second-order EKF we expand $f(x, u, w, t)$ with an additional term in the Taylor series:

$$f(x, u, w, t) = f(\hat{x}, u_0, w_0, t) + \left.\frac{\partial f}{\partial x}\right|_{\hat{x}} (x - \hat{x}) + \frac{1}{2} \sum_{i=1}^{n} \phi_i (x - \hat{x})^T \left.\frac{\partial^2 f_i}{\partial x^2}\right|_{\hat{x}} (x - \hat{x}) \tag{13.69}$$

where n is the dimension of the state vector, f_i is the ith element of $f(x, u, w, t)$, and the ϕ_i vector is defined as an $n \times 1$ vector with all zeros except for a one in the ith element. The quadratic term in the summation can be written as

$$(x - \hat{x})^T \left.\frac{\partial^2 f_i}{\partial x^2}\right|_{\hat{x}} (x - \hat{x}) = \mathrm{Tr}\left[\left.\frac{\partial^2 f_i}{\partial x^2}\right|_{\hat{x}} (x - \hat{x})(x - \hat{x})^T\right] \tag{13.70}$$

Since we do not know the value of $(x - \hat{x})(x - \hat{x})^T$ in the above equation, we replace it with its expected value, which is the covariance of the Kalman filter, to obtain

$$(x - \hat{x})^T \left.\frac{\partial^2 f_i}{\partial x^2}\right|_{\hat{x}} (x - \hat{x}) \approx \mathrm{Tr}\left[\left.\frac{\partial^2 f_i}{\partial x^2}\right|_{\hat{x}} P\right] \tag{13.71}$$

We then evaluate Equation (13.69) at $x = \hat{x}$ and substitute the above expression in the summation to obtain the time-update equation for $\hat{x}$ as

$$\dot{\hat{x}} = f(\hat{x}, u_0, w_0, t) + \frac{1}{2}\sum_{i=1}^{n} \phi_i \mathrm{Tr}\left[\left.\frac{\partial^2 f_i}{\partial x^2}\right|_{\hat{x}} P\right] \tag{13.72}$$

The time-update equation for P remains the same as in the standard hybrid EKF as shown in Equation (13.28):

$$\dot{P} = FP + PF^T + LQL^T \tag{13.73}$$

Now we will derive the measurement-update equations. Suppose that the measurement-update equation for the state estimate is given as

$$\hat{x}_k^+ = \hat{x}_k^- + K_k\left[y_k - h(\hat{x}_k^-, t_k)\right] - \pi_k \tag{13.74}$$

where K_k is the Kalman gain to be determined, and π_k is a correction term to be determined. We will choose π_k so that the estimate $\hat{x}_k^+$ is unbiased, and we will then choose K_k to minimize the trace of the covariance of the estimate.

If we define the estimation errors as

$$\begin{aligned} e_k^- &= x_k - \hat{x}_k^- \\ e_k^+ &= x_k - \hat{x}_k^+ \end{aligned} \tag{13.75}$$

we can see from Equations (13.66) and (13.74) that

$$e_k^+ = e_k^- - K_k\left[h(x_k, t_k) - h(\hat{x}_k^-, t_k)\right] - K_k v_k + \pi_k \tag{13.76}$$

Now we perform a second-order Taylor series expansion of $h(x_k, t_k)$ around the nominal point $\hat{x}_k^-$ to obtain

$$\begin{aligned} h(x_k, t_k) &= h(\hat{x}_k^-, t_k) + \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_k^-}(x_k - \hat{x}_k^-) + \\ &\quad \frac{1}{2}\sum_{i=1}^{m} \phi_i (x_k - \hat{x}_k^-)^T \left.\frac{\partial^2 h(i)}{\partial x^2}\right|_{\hat{x}_k^-}(x_k - \hat{x}_k^-) \\ &= h(\hat{x}_k^-, t_k) + H_k(x_k - \hat{x}_k^-) + \frac{1}{2}\sum_{i=1}^{m} \phi_i (x_k - \hat{x}_k^-)^T \left.\frac{\partial^2 h_i}{\partial x^2}\right|_{\hat{x}_k^-}(x_k - \hat{x}_k^-) \end{aligned} \tag{13.77}$$

where H_k is defined by the above equation, m is the dimension of the measurement vector, and h_i is the ith element of $h(x_k, t_k)$. This gives the *a posteriori* estimation error as

$$e_k^+ = e_k^- - K_k H_k e_k^- - \frac{1}{2}K_k \sum_{i=1}^{m} \phi_i (e_k^-)^T D_{k,i} e_k^- - K_k v_k + \pi_k \tag{13.78}$$

where $D_{k,i}$ is defined as

$$D_{k,i} = \left.\frac{\partial^2 h_i}{\partial x^2}\right|_{\hat{x}_k^-} \tag{13.79}$$

Taking the expected value of both sides of Equation (13.78), assuming that $E(e_k^-) = 0$, and making the same approximation as in Equation (13.71), we can see that in order to have $E(e_k^+) = 0$ we must set

$$\pi_k = \frac{1}{2} K_k \sum_{i=1}^{m} \phi_i \mathrm{Tr}\left[D_{k,i} P_k^-\right] \tag{13.80}$$

Defining P_k^+ as

$$P_k^+ = E\left[e_k^+ (e_k^+)^T\right] \tag{13.81}$$

and using the above equations, it can be shown after some involved algebraic calculations [Ath68] that

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k (R_k + \Lambda_k) K_k^T \tag{13.82}$$

where the matrix Λ_k is defined as

$$\Lambda_k = \frac{1}{4} E\left\{\left[\sum_{i=1}^{m} \phi_i \mathrm{Tr}[D_{k,i}(e_k^- (e_k^-)^T - P_k^-)]\right]\left[\cdots\right]^T\right\} \tag{13.83}$$

Now we define a cost function J_k that we want to minimize as a weighted sum of estimation errors:

$$\begin{aligned} J_k &= E[(e_k^+)^T S_k e_k^+] \\ &= \mathrm{Tr}[S_k P_k^+] \end{aligned} \tag{13.84}$$

where S_k is any positive definition weighting matrix. The K_k that minimizes this cost function can be found as

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k + \Lambda_k\right)^{-1} \tag{13.85}$$

This gives the P_k^+ matrix from Equation (13.82) as

$$P_k^+ = P_k^- - P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k + \Lambda_k\right)^{-1} H_k P_k^- \tag{13.86}$$

Now we need to figure out how to evaluate the Λ_k matrix in Equation (13.83). Note that Λ_k can be written as the double summation

$$\Lambda_k = \frac{1}{4} E\left\{\sum_{i,j=1}^{m} \phi_i \phi_j^T \mathrm{Tr}\left[D_{k,i}(e_k^- (e_k^-)^T - P_k^-)\right] \mathrm{Tr}\left[D_{k,j}(e_k^- (e_k^-)^T - P_k^-)\right]\right\} \tag{13.87}$$

The product $\phi_i \phi_j^T$ is an $m \times m$ matrix whose elements are all zero except for the element in the ith row and jth column. Therefore, the element in the ith row and jth column of Λ_k can be written as

$$\Lambda_k(i,j) = \frac{1}{4} E\left\{\mathrm{Tr}\left[D_{k,i}(e_k^- (e_k^-)^T - P_k^-)\right] \mathrm{Tr}\left[D_{k,j}(e_k^- (e_k^-)^T - P_k^-)\right]\right\} \tag{13.88}$$

This expression can be evaluated with the following lemma [Ath68].

Lemma 6 *Suppose we have the n-element random vector $x \sim N(0, P)$. Then*

$$\begin{aligned} E\left[x\,Tr(Axx^T)\right] &= 0 \\ E\left[Tr(Axx^T Bxx^T)\right] &= E\left[Tr(Axx^T)\,Tr(Bxx^T)\right] \\ &= 2\,Tr(APBP) + Tr(AP)\,Tr(BP) \end{aligned} \tag{13.89}$$

where A and B are arbitrary $n \times n$ matrices.

Using this lemma with Equation (13.88) we can see that

$$\Lambda_k(i,j) = \frac{1}{2}\mathrm{Tr}(D_{k,i}P_k^- D_{k,j}P_k^-) \tag{13.90}$$

This equation, along with Equations (13.74), (13.80), (13.82), and (13.85), specify the measurement-update equations for the second-order EKF. The second-order EKF can be summarized as follows.

The second-order hybrid extended Kalman filter

1. The system equations are given as follows:

$$\begin{aligned} \dot{x} &= f(x,u,w,t) \\ y_k &= h(x_k,t_k) + v_k \\ w(t) &\sim (0,Q) \\ v_k &\sim (0,R_k) \end{aligned} \tag{13.91}$$

2. The estimator is initialized as follows:

$$\begin{aligned} \hat{x}_0^+ &= E(x_0) \\ P_0^+ &= E\left[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T\right] \end{aligned} \tag{13.92}$$

3. The time-update equations are given as

$$\begin{aligned} \dot{\hat{x}} &= f(\hat{x},u,0,t) + \frac{1}{2}\sum_{i=1}^{n}\phi_i \mathrm{Tr}\left[\left.\frac{\partial^2 f_i}{\partial x^2}\right|_{\hat{x}} P\right] \\ \dot{P} &= FP + PF^T + LQL^T \\ \phi_i &= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i\text{th element} \\ F &= \left.\frac{\partial f}{\partial x}\right|_{\hat{x}} \\ L &= \left.\frac{\partial f}{\partial w}\right|_{\hat{x}} \end{aligned} \tag{13.93}$$

4. The measurement update equations are given as

$$
\begin{aligned}
\hat{x}_k^+ &= \hat{x}_k^- + K_k\left[y_k - h(\hat{x}_k^-)\right] - \pi_k \\
\pi_k &= \frac{1}{2}K_k\sum_{i=1}^{m}\phi_i \mathrm{Tr}\left[D_{k,i}P_k^-\right] \\
D_{k,i} &= \left.\frac{\partial^2 h_i(x_k,t_k)}{\partial x^2}\right|_{\hat{x}_k^-} \\
K_k &= P_k^- H_k^T\left(H_k P_k^- H_k^T + R_k + \Lambda_k\right)^{-1} \\
H_k &= \left.\frac{\partial h(x_k,t_k)}{\partial x}\right|_{\hat{x}_k^-} \\
\Lambda_k(i,j) &= \frac{1}{2}\mathrm{Tr}(D_{k,i}P_k^- D_{k,j}P_k^-) \\
P_k^+ &= P_k^- - P_k^- H_k^T\left(H_k P_k^- H_k^T + R_k + \Lambda_k\right)^{-1} H_k P_k^- \qquad (13.94)
\end{aligned}
$$

Note that setting the second partial derivative matrices in this algorithm to zero matrices results in the standard hybrid EKF.

■ EXAMPLE 13.3

In this example, we compare the performance of the EKF, the second-order EKF, and the iterated EKF for the falling body problem described in Example 13.2. A similar comparison was shown in [Wis69], where it was concluded that the iterated EKF had better RMS error performance, but the second-order filter had smaller bias. The system equations are the same as those shown in Example 13.2:

$$
\begin{aligned}
\dot{x}_1 &= x_2 + w_1 \\
\dot{x}_2 &= \rho_0 \exp(-x_1/k)x_2^2 x_3/2 - g + w_2 \\
\dot{x}_3 &= w_3 \qquad (13.95)
\end{aligned}
$$

In this example, we change the measurement system so that it does not measure the altitude of the falling body, but instead measures the range to the measuring device. The measuring device is located at an altitude a and at a horizontal distance M from the body's vertical line of fall. The measurement equation is therefore given by

$$
\begin{aligned}
y_k &= \sqrt{M^2 + (x_1(t_k) - a)^2} + v_k \\
&= h(x_k) + v_k \qquad (13.96)
\end{aligned}
$$

This makes the problem more nonlinear and hence more difficult to estimate (i.e., in Example 13.2 we had a nonlinear system but a linear measurement, whereas in this example we have nonlinearities in both the system and the measurement equations). The partial derivative F matrix for the EKFs are given in Example 13.2. The other partial derivative matrices used in the second-order EKF are given as follows:

$$
\begin{aligned}
H &= \frac{\partial h}{\partial x} \\
&= \begin{bmatrix} (x_1 - a)(M^2 + (x_1 - a)^2)^{-1/2} & 0 & 0 \end{bmatrix} \\
L &= \frac{\partial f}{\partial w} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
D_1 &= \frac{\partial^2 h_1}{\partial x^2} \\
&= \begin{bmatrix} h^{-1}(1 - (x_1 - a)^2 h^{-2}) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
\frac{\partial^2 f_1}{\partial x^2} = \frac{\partial^2 f_3}{\partial x^2} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
\frac{\partial^2 f_2}{\partial x^2} &= \rho_0 \exp(-x_1/k) \begin{bmatrix} x_2^2 x_3/2k^2 & -x_2 x_3/k & -x_2^2/2k \\ -x_2 x_3/k & x_3 & x_2 \\ -x_2^2/2k & x_2 & 0 \end{bmatrix}
\end{aligned}
\tag{13.97}
$$

Table 13.2 shows the performances of the EKFs (averaged over 20 simulation runs). It is seen that second-order EKF provides significant improvement over the first-order EKF for altitude and velocity estimation, but for some reason it actually provides worse performance for ballistic coefficient estimation. Also note that the iterated EKF provides only slight improvement over the first-order EKF, and (as expected) the iterated EKF performs better when more iterations are executed for the linearization refinement.

Table 13.2 Example 13.3 results. A comparison of the estimation errors of different EKF approaches for tracking a falling body.

Filter	Altitude	Velocity	Ballistic Coefficient
First-order EKF	758 feet	518 feet/sec	0.091 feet^3/lb/sec^2
Second-order EKF	356	483	0.129
Iterated EKF ($N = 2$)	755	517	0.091
Iterated EKF ($N = 3$)	745	516	0.091
Iterated EKF ($N = 4$)	738	509	0.091
Iterated EKF ($N = 5$)	733	506	0.091
Iterated EKF ($N = 6$)	723	506	0.091

We conclude from this that the second-order filter has better estimation performance. However, the implementation is much more difficult and requires the computation of second-order derivatives. In this example, the second-order derivatives could be taken analytically because we have explicit

analytical system and measurement equations. In many applications second-order derivatives will not be available analytically, and approximations will inevitably be subject to error.

These results are different than reported in [Wis69], where it was shown that the iterated EKF performed better than the second-order EKF. The different conclusions between this book and [Wis69] show that comparisons between different algorithms are often subjective. Perhaps the discrepancies are due to differences in implementations of the filtering algorithms, differences in implementations of the system dynamics or random noise generation, differences in the way that the estimation errors were measured, or even differences in the computing platforms that were used.

$\nabla\nabla\nabla$

The second-order filter was initially developed by Bass [Bas66] and Jazwinski [Jaz66]. A Gaussian second-order filter was developed by Athans [Ath68] and Jazwinski [Jaz70], in which fourth-order terms in Taylor series approximations are retained and approximated by assuming that the underlying probabilities are Gaussian. A small correction in the original derivations of the second-order EKF was reported by Rolf Henriksen [Hen82]. Although the second-order filter often provides improved performance over the extended Kalman filter, nothing definitive can be said about its performance, as evidenced by an example of an unstable second-order filter reported in [Kus67]. Additional comparison and analysis of some nonlinear Kalman filters can be found in [Sch68, Wis69, Wis70, Net78]. A simplified version of Henriksen's discrete-time second-order filter can be summarized as follows.

The second-order discrete-time extended Kalman filter

1. The system equations are given as follows:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, k) + w_k \\ y_k &= h(x_k, k) + v_k \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \end{aligned} \tag{13.98}$$

2. The estimator is initialized as follows:

$$\begin{aligned} \hat{x}_0^+ &= E(x_0) \\ P_0^+ &= E\left[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T\right] \end{aligned} \tag{13.99}$$

3. The time update equations are given as follows:

$$\begin{aligned} \hat{x}_{k+1}^- &= f(\hat{x}_k^+, u_k, k) + \frac{1}{2}\sum_{i=1}^{n} \phi_i \mathrm{Tr}\left[\left.\frac{\partial^2 f_i}{\partial x}\right|_{\hat{x}_k^+} P_k^+\right] \\ P_{k+1}^- &= F P_k^+ F^T + Q_k \end{aligned}$$

$$
\begin{aligned}
\phi_i &= \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i\text{th element} \\
F &= \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_k^+}
\end{aligned}
\tag{13.100}
$$

4. The measurement update equations are given as follows:

$$
\begin{aligned}
\hat{x}_k^+ &= \hat{x}_k^- + K_k[y_k - h(\hat{x}_k^-, k)] - \pi_k \\
\pi_k &= \frac{1}{2} K_k \sum_{i=1}^{m} \phi_i \mathrm{Tr}\left[D_{k,i} P_k^-\right] \\
D_{k,i} &= \left.\frac{\partial^2 h_i(x_k, k)}{\partial x^2}\right|_{\hat{x}_k^-} \\
K_k &= P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1} \\
H_k &= \left.\frac{\partial h(x_k, k)}{\partial x}\right|_{\hat{x}_k^-} \\
P_k^+ &= (I - K_k H_k) P_k^-
\end{aligned}
\tag{13.101}
$$

A more general version of the above algorithm can be found in [Hen82]. Similar to the hybrid second-order EKF presented earlier in this section, we note that setting the second-order partial derivative matrices in this algorithm to zero matrices results in the standard discrete-time EKF.

13.3.3 Other approaches

We have considered a couple of higher-order approaches to reducing the linearization error of the EKF. We looked at the iterated EKF and the second-order EKF, but other approaches are also available. For example, Gaussian sum filters are based on the idea that a non-Gaussian pdf can be approximated by a sum of Gaussian pdfs. This is similar to the idea that any curve can be approximated by a piecewise constant function. Since the true pdf of the process noise and measurement noise can be approximated by a sum of M Gaussian pdfs, we can run M Kalman filters in parallel on M Gaussian filtering problems, each of them optimal filters, and then combine them to obtain an approximately optimal estimate. The number of filters M is a trade-off between approximation accuracy (and hence optimality) and computational effort. This idea was first mentioned in [Aok65] and was explored in [Cam68, Sor71b, Als74, Kit89]. The Gaussian sum filter algorithm presented in [Als72] can be summarized as follows.

The Gaussian sum filter

1. The discrete-time n-state system and measurement equations are given as follows:

$$\begin{aligned} x_k &= f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}) \\ y_k &= h_k(x_k, v_k) \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \end{aligned} \tag{13.102}$$

2. Initialize the filter by approximating the pdf of the initial state as follows:

$$\text{pdf}(\hat{x}_0^+) = \sum_{i=1}^{M} a_{0i} N(\hat{x}_{0i}^+, P_{0i}^+) \tag{13.103}$$

The a_{0i} coefficients (which are positive and add up to 1), the $\hat{x}_{0i}^+$ means, and the P_{0i}^+ covariances, are chosen by the user to provide a good approximation to the pdf of the initial state.

3. For $k = 1, 2, \cdots$, do the following.

   (a) The *a priori* state estimate is obtained by first executing the following time-update equations for $i = 1, \cdots, M$:

$$\begin{aligned} \hat{x}_{ki}^- &= f_{k-1}(\hat{x}_{k-1,i}^+, u_{k-1}, 0) \\ F_{k-1,i} &= \left.\frac{\partial f_{k-1}}{\partial x_{k-1}}\right|_{\hat{x}_{k-1,i}^+} \\ P_{ki}^- &= F_{k-1,i} P_{k-1,i}^+ F_{k-1,i}^T + Q_{k-1} \\ a_{ki} &= a_{k-1,i} \end{aligned} \tag{13.104}$$

   The pdf of the *a priori* state estimate is obtained by the following sum:

$$\text{pdf}(\hat{x}_k^-) = \sum_{i=1}^{M} a_{ki} N(\hat{x}_{ki}^-, P_{ki}^-) \tag{13.105}$$

   (b) The *a posteriori* state estimate is obtained by first executing the following measurement update equations for $i = 1, \cdots, M$:

$$\begin{aligned} H_{ki} &= \left.\frac{\partial h_k}{\partial x_k}\right|_{\hat{x}_{ki}^-} \\ K_{ki} &= P_{ki}^- H_{ki}^T (H_{ki} P_{ki}^- H_{ki}^T + R_k)^{-1} \\ P_{ki}^+ &= P_{ki}^- - K_{ki} H_{ki} P_{ki}^- \\ \hat{x}_{ki}^+ &= \hat{x}_{ki}^- + K_{ki} \left[y_k - h_k(\hat{x}_{ki}^-, 0) \right] \end{aligned} \tag{13.106}$$

   The weighting coefficients a_{ki} for the individual estimates are obtained as follows:

$$
\begin{aligned}
r_{ki} &= y_k - h_k(\hat{x}_{ki}^-, 0) \\
S_{ki} &= H_{ki} P_{ki}^- H_{ki}^T + R_k \\
\beta_{ki} &= \frac{\exp\left[-r_{ki}^T S_{ki}^{-1} r_{ki}/2\right]}{(2\pi)^{n/2}|S_{ki}|^{1/2}} \\
a_{ki} &= \frac{a_{k-1,i}\beta_{ki}}{\sum_{j=1}^{M} a_{k-1,j}\beta_{kj}}
\end{aligned}
\tag{13.107}
$$

Note that the weighting coefficient a_{ki} is computed by using the measurement y_k to obtain the relative confidence β_{ki} of the estimate $\hat{x}_{ki}^-$. The pdf of the *a posteriori* state estimate is obtained by the following sum:

$$
\mathrm{pdf}(\hat{x}_k^+) = \sum_{i=1}^{M} a_{ki} N(\hat{x}_{ki}^+, P_{ki}^+) \tag{13.108}
$$

This approach can also be extended to smoothing [Kit94]. Similar approaches can be taken to expand the pdf using non-Gaussian functions [Aok67, Sor68, Sri70, deF71, Hec71, Hec73, Mcr75, Wil81, Kit87, Kra88]. A related filter has been derived for the case where either the process noise or the measurement noise is strictly Gaussian, but the other noise is Gaussian with heavy tails [Mas75, Tsa83]. This is motivated by the observation that many instances of noise in nature have pdfs that are approximately Gaussian but with heavier tails [Mas77].

Another approach to nonlinear filtering is called grid-based filtering. In grid-based filtering, the value of the pdf of the state is approximated, stored, propagated, and updated at discrete points in state space [Buc69, Buc71]; [Spa88, Chapter 6]. This is similar to particle filtering (discussed in Chapter 15), except in particle filtering we choose the particles to be distributed in state space according to the pdf of the state. Grid-based filtering does not distribute the particles in this way, and hence has computational requirements that increase exponentially with the dimension of the state. Grid-based filtering is even more computationally expensive than particle filtering, and this has limited its application. Furthermore, particle filtering is a type of "intelligent" grid-based filtering. This seems to portend very little further work in grid-based filtering.

Richard Bucy suggested yet another approach to nonlinear filtering [Buc65]. Instead of linearizing the system dynamics, compute the theoretically optimal nonlinear filter, and then linearize the nonlinear filter. However, the theoretically optimal nonlinear filter is very difficult to compute except in special cases.

13.4 PARAMETER ESTIMATION

State estimation theory can be used to not only estimate the states of a system, but also to estimate the unknown parameters of a system. This may have first been suggested in [Kop63]. Suppose that we have a discrete-time system model, but the system matrices depend in a nonlinear way on an unknown parameter vector p:

$$\begin{aligned} x_{k+1} &= F_k(p)x_k + G_k(p)u_k + L_k(p)w_k \\ y_k &= H_k x_k + v_k \\ p &= \text{ unknown parameter vector} \end{aligned} \tag{13.109}$$

In this model, we are assuming that the measurement is independent of p, but this is only for notational convenience. The discussion here can easily be extended to include a dependence of y_k on p. Assume that p is a constant parameter vector. We do not really care about estimating the state, but we are interested in estimating p. This is the case, for example, in the aircraft engine health estimation problem [Kob03, Sim05a]. In those papers it was assumed that we want to estimate aircraft engine health (for the purpose of maintenance scheduling), but we do not really care about estimating the states of the engine.

In order to estimate the parameter p, we first augment the state with the parameter to obtain an augmented state vector x':

$$x'_k = \begin{bmatrix} x_k \\ p_k \end{bmatrix} \tag{13.110}$$

If p_k is constant then we model $p_{k+1} = p_k + w_{pk}$, where w_{pk} is a small artificial noise term that allows the Kalman filter to change its estimate of p_k. Our augmented system model can be written as

$$\begin{aligned} x'_{k+1} &= \begin{bmatrix} F_k(p_k)x_k + G_k(p_k)u_k + L_k(p_k)w_k \\ p_k + w_{pk} \end{bmatrix} \\ &= f(x'_k, u_k, w_k, w_{pk}) \\ y_k &= \begin{bmatrix} H_k & 0 \end{bmatrix} \begin{bmatrix} x_k \\ p_k \end{bmatrix} + v_k \end{aligned} \tag{13.111}$$

Note that $f(x'_k, u_k, w_k, w_{pk})$ is a nonlinear function of the augmented state x'_k. We can therefore use an extended Kalman filter (or any other nonlinear filter) to estimate x'_k.

■ EXAMPLE 13.4

This example is taken from [Ste94]. Suppose we have a second-order system governed by the following equations:

$$\ddot{x}_1 + 2\zeta\omega_n\dot{x}_1 + \omega_n^2 x_1 = \omega_n^2 w \tag{13.112}$$

where ω_n is the natural frequency of the system, ζ is the damping ratio, and the input w is zero-mean noise. A state-space model for this system can be written as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\omega_n^2 x_1 - 2\zeta\omega_n x_2 + \omega_n^2 w \\ \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} w \end{aligned} \tag{13.113}$$

Suppose that $-2\zeta\omega_n$ is known, but ζ and ω_n are unknown. We want to estimate $-\omega_n^2$. Suppose that both x_1 and x_2 are available for measurement. We define the known parameter as b; that is, $b = -2\zeta\omega_n$. We define a new state element equal to the parameter that we want to estimate. That is, $x_3 = -\omega_n^2$. We then form an augmented system model as follows:

$$\begin{aligned}
\dot{x}' &= \begin{bmatrix} x_2 \\ x_3x_1 + bx_2 - x_3w \\ w_p \end{bmatrix} \\
&= f(x', w') \\
w' &= \begin{bmatrix} w \\ w_p \end{bmatrix} \\
y &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x' + v
\end{aligned} \tag{13.114}$$

where w_p is an artificial noise term that we add to the system that allows the Kalman filter to modify its estimate of x_3. We can use an extended Kalman filter to estimate the augmented state. First we need to find the partial derivative matrices:

$$\begin{aligned}
F &= \left.\frac{\partial f}{\partial x'}\right|_{\hat{x}', w_0'} \\
&= \begin{bmatrix} 0 & 1 & 0 \\ x_3 & b & x_1 - w \\ 0 & 0 & 0 \end{bmatrix}_{\hat{x}', w_0'} \\
&= \begin{bmatrix} 0 & 1 & 0 \\ \hat{x}_3 & b & \hat{x}_1 \\ 0 & 0 & 0 \end{bmatrix} \\
L &= \left.\frac{\partial f}{\partial w'}\right|_{\hat{x}', w_0'} \\
&= \begin{bmatrix} 0 & 0 \\ -\hat{x}_3 & 0 \\ 0 & 1 \end{bmatrix}
\end{aligned} \tag{13.115}$$

The continuous-time extended Kalman filter can be written as

$$\begin{aligned}
\dot{\hat{x}}' &= f(\hat{x}', 0) + K(y - H\hat{x}') \\
K &= PH^TR^{-1} \\
\dot{P} &= FP + PF^T + LQL^T - PH^TR^{-1}HP
\end{aligned} \tag{13.116}$$

Figure 13.4 illustrates the results of a typical simulation of the extended Kalman filter that is used to estimate $-\omega_n^2$ for this system. The true system parameters are $\omega_n = 2$ and $\zeta = 0.1$, so $-\omega_n^2 = -4$. Suppose that we begin by estimating $-\omega_n^2$ as -8 with an initial estimation variance of 20. Figure 13.4 shows that the error in our estimate of $-\omega_n^2$ gradually decreases toward zero, and the estimation variance gradually decreases. We set the variance of the artificial noise w_p equal to 0.1 in this example. This allows the Kalman filter

Figure 13.4 Example 13.4 results. Typical parameter estimation performance and parameter uncertainty for an extended Kalman filter estimating $-\omega_n^2$ for a second-order system. The estimation error of the unknown parameter and its variance gradually decrease toward zero.

to more readily adjust its estimate of $-\omega_n^2$, but also may prevent the filter from converging to the true value (see Problem 13.23).

$\nabla\nabla\nabla$

13.5 SUMMARY

Optimal state estimators can be derived for general classes of nonlinear systems as shown in [Kus67], but the filters are generally infinite dimensional, which makes them impractical for implementation. Finite-dimensional, optimal, nonlinear state estimators can be derived for more restricted classes of nonlinear systems [Liu80], but the restriction on the classes of applicable systems are significant enough to prevent wide applicability. Because of these factors, nonlinear Kalman filtering is the most widespread approach to state estimation for nonlinear systems.

It is interesting to note that the first applications of Kalman filtering were on nonlinear orbit-estimation problems [Bat62]. Some early investigations in nonlinear Kalman filtering can be found in [Cox64, Fri66]. Whereas stability and convergence results are readily available for the linear Kalman filter, such results are much more difficult to obtain for nonlinear Kalman filtering. Some convergence results for nonlinear Kalman filtering are found in [Urs80]. If the nonlinearities have known bounds then the Riccati equation can be modified in a simple way to guarantee stability for the continuous-time EKF [Rei98]. Conditions needed to guarantee the boundedness of the discrete-time EKF error covariance can be related to the observability of the underlying nonlinear system [Dez92, Son95].

PROBLEMS

Written exercises

13.1 Consider the scalar system

$$\begin{aligned} \dot{x} &= -x + w \\ y &= x + v \end{aligned}$$

The process noise has a mean value of 2, and the measurement noise has a mean value of 3. Redefine the noise quantities and the state to obtain an equivalent system of the form

$$\begin{aligned} \dot{x}' &= Ax' + Bu + w' \\ y &= Cx' + v' \end{aligned}$$

so that the new noise quantities w' and v' both have mean values of 0.

13.2 Consider the scalar system

$$\dot{x} = -x + u + w$$

w is zero-mean process noise with a variance of Q. The control has a mean value of u_0, an uncertainty of 2 (one standard deviation), and is uncorrelated with w. Rewrite the system equations to obtain an equivalent system with a normalized control that is perfectly known. What is the variance of the new process noise term in the transformed system equation?

13.3 Suppose that x is a constant scalar, and $y_k = \sqrt{x}(1 + v_k)$ are noisy measurements, where $v_k \sim N(0, R)$.

a) An intuitive way to estimate x is to set $\hat{x}_k = y_k^2$. Compute the mean and variance of the estimation error for this estimate. Your answer should be a function of x and R. Hint: recall that $E(v_k^3) = 0$ and $E(v_k^4) = 3R^2$.

b) Perhaps a better estimate for x_k could be obtained by averaging all previous values of y_k^2. That is,

$$\hat{x}_k = \frac{1}{k}\sum_{i=1}^{k} y_i^2$$

Compute the mean and variance of the estimation error for this estimate. Your answer should be a function of k, x, and R. Note that if you substitute $k = 1$ into your solution, you should get the same answer as part (a). What is the variance as $k \to \infty$?

c) Write the extended Kalman filter equations to estimate x. What is the theoretical mean and variance of the EKF estimate as $k \to \infty$?

13.4 Consider the system

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + v_k^2 \end{aligned}$$

where w_k and v_k are uniformly distributed, uncorrelated, zero-mean white noise processes with variances Q and R, respectively.

a) What is the mean of the *a posteriori* estimation error for the discrete EKF?

b) Modify the measurement equation by subtracting the known bias of the measurement noise so that the modified measurement noise is zero-mean. What is the variance of the modified measurement noise?

13.5 Consider the nonlinear system

$$\begin{aligned} x_{k+1} &= -x_k^2 + u_k + w_k \\ y_k &= 4x_k^2 + v_k \end{aligned}$$

Find the nominal values for x_k and y_k when $x_0 = 0$ and $u_k = 1$.

13.6 Consider the system $x_{k+1} = x_k^2 + w_k$, where w_k is zero-mean. The initial state x_0 is uniformly distributed between 0 and 1. An EKF is initialized with $\hat{x}_0^+ = E(x_0)$. What is $E(x_1)$? What is $\hat{x}_1^-$? This problem illustrates the fact that the state estimate of an EKF is not always equal to the expected value of the state.

13.7 Find the terminal velocity of the falling body of Example 13.2 if the terminal velocity occurs at an altitude of 1 mile.

13.8 Consider the hybrid scalar system

$$\begin{aligned} \dot{x} &= f(x) + w, \qquad w \sim N(0, Q) \\ y_k &= h(x_k) + v_k, \qquad v_k \sim N(0, R) \end{aligned}$$

The estimator that is used for the system is

$$\hat{x}_k = a + by_k + cy_k^2$$

Suppose that the state $x(t)$ is normally distributed with a mean of zero and a variance of P_x.

a) Find an equation relating a, b, and c that must be satisfied in order for $\hat{x}_k$ to be an unbiased estimate of $x(t_k)$ [Gel74].

b) Find values of a, b, and c so that $\hat{x}_k$ is the minimum-variance estimate. Assume that $h(x)$ is an odd function of x.

13.9 Suppose for a scalar system that $P_k^- = 1$, $R = 1$, and $H = 3$. What is the value of P_k^+ as given by Equation (5.19)? What will be the computed value of P_k^+ if $H = 2$ is used instead? What will be the computed value of P_k^+ if $H = 1$ is used instead? This illustrates how the iterated Kalman filter gets a more accurate estimate of P_k^+ by using a more accurate value for H_k.

13.10 Consider a system with the measurement equation $y_k = x_k^2 + v_k$. At time k the *a priori* state estimate is $\hat{x}_k^- = 1$, the true state is $x_k = 5$, and the measurement is $y_k = 25$. The *a priori* estimation-error variance is $P_k^- = 1$, and the measurement noise variance is $R_k = 4$. Use the iterated EKF algorithm to find $\hat{x}_{k,1}^+$ and $\hat{x}_{k,2}^+$. Although the iterated EKF does not always improve the *a posteriori* state estimate, this problem illustrates how it usually does.

13.11 Prove Lemma 6 for scalar random variables x.

13.12 Suppose you have the process equation $\dot{x} = x^2 + w$ and the state estimate $\hat{x}_k^+ = 0$. What is the differential equation for propagating $\hat{x}$ to the next measurement time using the first-order EKF? What is the differential equation using the second-order EKF?

13.13 Consider the measurement equation $y_k = x_k^2 + v_k$, where $v_k \sim (0, R)$. Suppose that $P_k^- = 1$, and $\hat{x}_k^- = 1$ is unbiased.

a) What is the expected value of $\hat{x}_k^+$ if the first-order EKF is used for the measurement update? Based on your expression for $E(\hat{x}_k^+)$, how does the bias of the state estimate change with R? Does this make intuitive sense?

b) What is the expected value of $\hat{x}_k^+$ if the second-order EKF is used for the measurement update?

13.14 Consider the system

$$\begin{aligned} z_{k+1} &= a z_k + w_k, \qquad w_k \sim (0, Q) \\ y_k &= z_k + v_k, \qquad v_k \sim (0, R) \end{aligned}$$

with unknown parameter a. Suppose that an EKF is used to estimate the state z_k and the parameter a. Further suppose that the artificial noise term used in the estimation of a is zero, and the EKF converges to the correct value of a with zero variance. Show that the EKF in this situation is equivalent to the standard Kalman filter for the scalar system when a is known.

Computer exercises

13.15 Write a program that implements the moving average filter and the extended Kalman filter for the system described in Problem 13.3. Use $R = 1$, $x = 1$, $P_0^+ = 1$, and $\hat{x}_0 = 2$. Which filter appears to perform better?

13.16 A planar model for a satellite orbiting around the earth can be modeled as

$$\begin{aligned} \ddot{r} &= r\dot{\theta}^2 - \frac{GM}{r^2} + w \\ \ddot{\theta} &= \frac{-2\dot{\theta}\dot{r}}{r} \end{aligned}$$

where r is the distance of the satellite from the center of the earth, θ is the angular position of the satellite in its orbit, $G = 6.6742 \times 10^{-11}\text{m}^3/\text{kg/s}^2$ is the universal gravitational constant, $M = 5.98 \times 10^{24}$ kg is the mass of the earth, and $w \sim (0, 10^{-6})$ is random noise due to space debris, atmospheric drag, outgassing, and so on.

a) Write a state-space model for this system with $x_1 = r$, $x_2 = \dot{r}$, $x_3 = \theta$, and $x_4 = \dot{\theta}$.

b) What must $\dot{\theta}$ be equal to in order for the orbit to have a constant radius when $w = 0$?

c) Linearize the model around the point $r = r_0$, $\dot{r} = 0$, $\theta = \omega_0 T$, $\dot{\theta} = \omega_0$. What are the eigenvalues of the system matrix for the linearized system when $r_0 = 6.57 \times 10^6$ m? What would you estimate to be the largest

integration step size that could be used to simulate the system? (Hint: recall that for a second-order transfer function with imaginary poles $\pm ja$, the time constant is equal to $1/a$.)

d) Suppose that measurements of the satellite radius and angular position are obtained every minute, with error standard deviations of 100 meters and 0.1 radians, respectively. Simulate the linearized Kalman filter for three hours. Initialize the system with $x(0) = \begin{bmatrix} r_0 & 0 & 0 & 1.1\omega_0 \end{bmatrix}$, $\hat{x}(0) = x(0)$, and $P(0) = \text{diag}(0,0,0,0)$. Plot the radius estimation error as a function of time. Why is the performance so poor? How could you modify the linearized Kalman filter to get better performance?

e) Implement an extended Kalman filter and plot the radius estimation error as a function of time. How does the performance compare with the linearized Kalman filter?

13.17 Implement the hybrid EKF with a measurement period of $0.1s$ for the system described in Example 13.1. Assume that the winding current measurement noises have a standard deviation of 0.1 amps. Create a table showing the experimental standard deviation of the motor velocity estimation error as a function of the standard deviation of the control input uncertainties q_1 and q_2. Use control input standard deviations from 0 to 0.1 volts in steps of 0.01 (i.e., $\sigma_q = 0$, $\sigma_q = 0.01$, $\cdots$, $\sigma_q = 0.1$). In order to make a fair comparison, you should either run several simulations for each value of σ_q and average the results, or else initialize the random seed in your software so that each simulation runs with the same random noise history.

13.18 Derive the first-order EKF, second-order EKF, and iterated EKF (with one iteration) for the scalar system

$$\begin{aligned} x_{k+1} &= x_k^2 + w_k \\ y_k &= x_k^2 + v_k \end{aligned}$$

where w_k and v_k are independent zero-mean white noise terms with variances 0.1 and 1, respectively. Simulate the first-order, second-order, and iterated extended Kalman filters for five time steps. Set the initial state to 1, the initial estimation-error variance to 1, and the initial state estimate to 2. Compute the RMS error of the filter estimates. How does the performance of the filters compare? (Note that you need more than one simulation, in general, to obtain a fair comparison of filter performance.)

13.19 Use the following procedure [Sor71b] to approximate a uniform pdf that is defined on ± 1 with M Gaussian pdfs; that is, $U(-1,1) \approx \sum_{i=1}^{M} a_i N(\mu_i, \sigma_i^2)$.

- Select the weighting coefficients so that $a_i = 1/M$ for all i.
- Select the means of the Gaussian pdfs to be equally spaced on the range $[-1,1]$ with $\mu_{i+1} - \mu_i = 2/(M+1)$.
- Select the variances σ_i of the Gaussian pdfs to all be the same and to minimize the RMS difference between $U(-1,1)$ and $\sum_{i=1}^{M} a_i N(\mu_i, \sigma_i^2)$ over the range $[-1,1]$.

The above approach reduces the approximation problem to a one-dimensional optimization problem, which can be solved in a number of different ways (for example,

using the golden search method [Pre92]). Plot the true pdf and the approximate pdf for $M = 3$, 5, and 10, and compare the RMS errors.

13.20 Suppose you have a scalar system given as

$$\begin{aligned} x_{k+1} &= x_k \\ y_k &= x_k^2 + v_k \end{aligned}$$

where v_k is white Gaussian noise with a variance of 0.01. The pdf of the initial state x_0 is uniform between -1 and $+1$. Note from the measurement equation that there is no way to distinguish between a positive state and a negative state.

a) What will the extended Kalman filter estimate of the system be equal to?

b) The pdf of x_0 can be approximated with two Gaussian pdfs, each with a variance of 0.43, and with respective means of $-1/3$ and $+1/3$. Suppose that $x_0 = -1/2$. Plot the true state and the individual state estimates of a two-term Gaussian sum filter for 20 time steps. Plot the Gaussian pdfs at the final time for each estimate of the two-term Gaussian sum filter.

13.21 Consider the problem of tracking a moving vehicle in two dimensions (north is one dimension and east is the other dimension). The vehicle's acceleration in the north and east directions consists of independent white noise. Two tracking stations, located at north–east coordinates (N_1, E_1) and (N_2, E_2), respectively, measure the range to the vehicle. The system model can therefore be written as

$$\begin{aligned} \begin{bmatrix} n_{k+1} \\ e_{k+1} \\ \dot{n}_{k+1} \\ \dot{e}_{k+1} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_k \\ e_k \\ \dot{n}_k \\ \dot{e}_k \end{bmatrix} + w_k \\ y_k &= \begin{bmatrix} \sqrt{(n_k - N_1)^2 + (e_k - E_1)^2} \\ \sqrt{(n_k - N_2)^2 + (e_k - E_2)^2} \end{bmatrix} + v_k \end{aligned}$$

where n_k and e_k are the vehicle's north and east coordinates at time step k, T is the time step of the system, w_k is the zero-mean process noise, and v_k is the zero-mean measurement noise. Suppose that the time step $T = 0.1s$, the process noise covariance $Q = \text{diag}(0, 0, 4, 4)$, and the measurement noise covariance $R = \text{diag}(1, 1)$. The tracking stations are located at $(N_1, E_1) = (20, 0)$, and $(N_2, E_2) = (0, 20)$. The initial state of the vehicle $x_0 = \begin{bmatrix} 0 & 0 & 50 & 50 \end{bmatrix}^T$ and is perfectly known. Design an extended Kalman filter to estimate the state of the vehicle. Run the simulation for 60 s. Plot the estimation error for the four states. What is the experimental standard deviation of the estimation error for each of the four states? Based on the steady-state covariance matrix of the filter, what is the theoretical standard deviation of the estimation error for each of the four states?

13.22 Consider the system

$$\begin{aligned} x_{k+1} &= \phi x_k + w_k \\ y_k &= x_k \end{aligned}$$

where $w_k \sim (0, 1)$, and $\phi = 0.9$ is an unknown constant. Design an extended Kalman filter to estimate ϕ. Simulate the filter for 100 time steps with $x_0 = 1$,

$P_0 = I$, $\hat{x}_0 = 0$, and $\hat{\phi}_0 = 0$. Hand in your source code and a plot showing $\hat{\phi}$ as a function of time.

13.23 Simulate Example 13.4 with artificial parameter noise variance values $\sigma_p^2 =$ 0, 1, and 100. How does a change in the artificial parameter noise variance affect the filter's estimate of $-\omega_n^2$?

CHAPTER 14

The unscented Kalman filter

We use the intuition that it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation.

—Simon Julier, Jeffrey Uhlmann, and Hugh Durrant-Whyte [Jul00]

As discussed earlier, the extended Kalman filter (EKF) is the most widely applied state estimation algorithm for nonlinear systems. However, the EKF can be difficult to tune and often gives unreliable estimates if the system nonlinearities are severe. This is because the EKF relies on linearization to propagate the mean and covariance of the state. This chapter discusses the unscented Kalman filter (UKF), an extension of the Kalman filter that reduces the linearization errors of the EKF. The use of the UKF can provide significant improvement over the EKF.

First, we will take a diversion from filtering in Section 14.1 to investigate how means and covariances propagate in nonlinear equations. In Section 14.2, we will present the unscented transformation, which is a way to approximate how the mean and covariance of a random variable change when the random variable undergoes a nonlinear transformation. In Section 14.3, we will use the previous results to derive the UKF and show that it has less linearization error than the EKF. In Section 14.4, we will present some modifications of the standard UKF which can be used to obtain more accurate or faster filtering results.

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

14.1 MEANS AND COVARIANCES OF NONLINEAR TRANSFORMATIONS

In this section, we will show how linearization approximations can result in errors in the transformation of means and covariances when a random variable is operated on by a nonlinear function. This section does not really have anything to do directly with state estimation, Kalman filtering, or the UKF. However, this section provides some background that will allow us to develop the UKF later in this chapter. This section will also give us a more complete background to understand the type of problems that can arise in the EKF (which relies on linearization).

Consider the nonlinear transformation

$$\begin{aligned} y_1 &= r\cos\theta \\ y_2 &= r\sin\theta \end{aligned} \tag{14.1}$$

This is a standard polar-to-rectangular coordinate transformation. For instance, we might have a sensor that measures range r and angle θ, and we want to convert the measured data to rectangular coordinates y_1 and y_2. The coordinate transformation can be written more generally as

$$y = h(x) \tag{14.2}$$

where y is the two-element output of $h(x)$, and the two-element vector x is defined as

$$x = \begin{bmatrix} r \\ \theta \end{bmatrix} \tag{14.3}$$

Suppose that x_1 (which is the range r) is a random variable with a mean of 1 and a standard deviation of σ_r. Suppose that x_2 (which is the angle θ) is a random variable with a mean of $\pi/2$ and a standard deviation of σ_θ. In other words, the means of the components of x are given as $\bar{r} = 1$ and $\bar{\theta} = \pi/2$. In addition, we will assume that r and θ are independent, and that their probability density functions are symmetric around their means (for example, Gaussian or uniform).

14.1.1 The mean of a nonlinear transformation

An initial consideration of the above problem, along with Equation (14.1), would lead us to believe that y_1 has a mean of 0, and y_2 has a mean of 1. In addition, a linearization approach would lead us to the same conclusion. If we perform a first-order linearization of Equation (14.2) and take the expected value of both sides, we obtain

$$\begin{aligned} \bar{y} &= E[h(x)] \\ &\approx E\left[h(\bar{x}) + \left.\frac{\partial h}{\partial x}\right|_{\bar{x}} (x - \bar{x})\right] \\ &= h(\bar{x}) + \left.\frac{\partial h}{\partial x}\right|_{\bar{x}} E(x - \bar{x}) \\ &= h(\bar{x}) \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \tag{14.4}$$

Our intuition, along with a first-order linearization analysis, both lead us to the same conclusion. However, let us pursue this problem with more rigor to check our previous analysis. We can write r and θ as

$$\begin{aligned} r &= \bar{r} + \tilde{r} \\ \theta &= \bar{\theta} + \tilde{\theta} \end{aligned} \tag{14.5}$$

where $\tilde{r}$ and $\tilde{\theta}$ are simply the deviations of r and θ from their means. A rigorous analysis of the mean of y_1 can be performed as follows:

$$\begin{aligned} \bar{y}_1 &= E(r\cos\theta) \\ &= E\left[(\bar{r}+\tilde{r})\cos(\bar{\theta}+\tilde{\theta})\right] \\ &= E\left[(\bar{r}+\tilde{r})(\cos\bar{\theta}\cos\tilde{\theta} - \sin\bar{\theta}\sin\tilde{\theta})\right] \end{aligned} \tag{14.6}$$

Carrying out the multiplication, remembering that $\tilde{r}$ and $\tilde{\theta}$ are independent with symmetric pdfs, and taking the expected value, results in

$$\begin{aligned} \bar{y}_1 &= \bar{r}\cos\bar{\theta} \\ &= 0 \end{aligned} \tag{14.7}$$

Our intuition and our first-order approximation of $\bar{y}_1$ have been confirmed by rigorous analysis. Let us repeat the process for y_2:

$$\begin{aligned} \bar{y}_2 &= E(r\sin\theta) \\ &= E\left[(\bar{r}+\tilde{r})\sin(\bar{\theta}+\tilde{\theta})\right] \\ &= E\left[(\bar{r}+\tilde{r})(\sin\bar{\theta}\cos\tilde{\theta} + \cos\bar{\theta}\sin\tilde{\theta})\right] \end{aligned} \tag{14.8}$$

Carrying out the multiplication, remembering that $\tilde{r}$ and $\tilde{\theta}$ are independent with symmetric pdfs, and taking the expected value, results in

$$\begin{aligned} \bar{y}_2 &= \bar{r}\sin\bar{\theta}E(\cos\tilde{\theta}) \\ &= E(\cos\tilde{\theta}) \end{aligned} \tag{14.9}$$

We cannot go any further unless we assume some distribution for $\tilde{\theta}$, so let us assume that $\tilde{\theta}$ is uniformly distributed between $\pm\theta_m$. In that case, we can compute

$$\begin{aligned} \bar{y}_2 &= E(\cos\tilde{\theta}) \\ &= \frac{\sin\theta_m}{\theta_m} \end{aligned} \tag{14.10}$$

We expected to get 1 for our answer in confirmation of Equation (14.4), but instead we got some number that is less than 1. [Note that $(\sin\theta_m)/\theta_m < 1$ for all $\theta_m > 0$, and $\lim_{\theta_m\to 0}(\sin\theta_m)/\theta_m = 1$.] The analysis reveals a problem with our initial intuition and the first-order linearization that we performed earlier. The mean of y_2 will indeed be less than 1. This can be seen by looking at a plot of 300 randomly generated r and θ values, where $\tilde{r}$ is uniformly distributed between ± 0.01, and $\tilde{\theta}$ is uniformly distributed between ± 0.35 radians. The small variance of $\tilde{r}$ and the large

Figure 14.1 Linearized and nonlinear mean of 300 randomly generated points with $\tilde{r}$ uniformly distributed between ± 0.01 and $\tilde{\theta}$ uniformly distributed between ± 0.35 radians.

variance of $\tilde{\theta}$ result in an arc-shaped distribution of points as seen in Figure 14.1. This arc-shaped distribution results in $\bar{y}_2 < 1$.

This is not a Kalman filtering example. But since the EKF uses first-order linearization to update the mean of the state, this example shows the kind of error that can creep into the EKF when it is applied to a nonlinear system.

For a more general analysis of the mean of a nonlinear transformation, recall from Equation (1.89) that $y = h(x)$ can be expanded in a Taylor series around $\bar{x}$ as follows:

$$\begin{aligned} y &= h(x) \\ &= h(\bar{x}) + D_{\tilde{x}}h + \frac{1}{2!}D_{\tilde{x}}^2 h + \frac{1}{3!}D_{\tilde{x}}^3 h + \cdots \end{aligned} \tag{14.11}$$

where $\tilde{x} = x - \bar{x}$. The mean of y can therefore be expanded as

$$\begin{aligned} \bar{y} &= E\left[h(\bar{x}) + D_{\tilde{x}}h + \frac{1}{2!}D_{\tilde{x}}^2 h + \frac{1}{3!}D_{\tilde{x}}^3 h + \cdots\right] \\ &= h(\bar{x}) + E\left[D_{\tilde{x}}h + \frac{1}{2!}D_{\tilde{x}}^2 h + \frac{1}{3!}D_{\tilde{x}}^3 h + \cdots\right] \end{aligned} \tag{14.12}$$

By using $D_{\tilde{x}}h$ from Equation (1.88) we can see that

$$\begin{aligned} E[D_{\tilde{x}}h] &= E\left[\sum_{i=1}^{n} \tilde{x}_i \frac{\partial}{\partial x_i} h(x)\Big|_{x=\bar{x}}\right] \\ &= \sum_{i=1}^{n} E(\tilde{x}_i) \frac{\partial}{\partial x_i} h(x)\Big|_{x=\bar{x}} \\ &= 0 \end{aligned} \tag{14.13}$$

because $E(\tilde{x}_i) = 0$. Likewise, we can see that

$$\begin{aligned} E[D_{\tilde{x}}^3 h] &= E\left[\left(\sum_{i=1}^{n} \tilde{x}_i \frac{\partial}{\partial x_i}\right)^3 h(x)\Big|_{x=\bar{x}}\right] \\ &= 0 \end{aligned} \tag{14.14}$$

This is because the sum in the above equation consists only of third-order moments $[E(\tilde{x}_1^3)$, $E(\tilde{x}_1^2\tilde{x}_2)$, etc.]. These expected values will always be zero as shown at the end of Section 2.2. Similarly all of the odd terms in Equation (14.12) will be zero, which leads to the simplification

$$\bar{y} = h(\bar{x}) + \frac{1}{2!}E[D_{\tilde{x}}^2 h] + \frac{1}{4!}E[D_{\tilde{x}}^4 h] + \cdots \tag{14.15}$$

This shows why the mean calculation in Equation (14.4) was incorrect; that calculation was only correct up to the first order. If we approximate $\bar{y}$ for our polar-to-rectangular transformation using terms up to the second order from Equation (14.15), we obtain

$$\begin{aligned} \bar{y} &\approx h(\bar{x}) + \frac{1}{2!}E[D_{\tilde{x}}^2 h] \\ &= h(\bar{x}) + \frac{1}{2}E\left[\left(\sum_{i=1}^{2} \tilde{x}_i \frac{\partial}{\partial x_i}\right)^2 h(x)\Big|_{x=\bar{x}}\right] \\ &= h(\bar{x}) + \frac{1}{2}\left(E(\tilde{x}_1^2)\left.\frac{\partial^2 h(x)}{\partial x_1^2}\right|_{x=\bar{x}} + 2E(\tilde{x}_1\tilde{x}_2)\left.\frac{\partial^2 h(x)}{\partial x_1 \partial x_2}\right|_{x=\bar{x}} + E(\tilde{x}_2^2)\left.\frac{\partial^2 h(x)}{\partial x_2^2}\right|_{x=\bar{x}}\right) \\ &= h(\bar{x}) + \frac{1}{2}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \sigma_\theta^2 \begin{bmatrix} -r\cos\theta \\ -r\sin\theta \end{bmatrix}_{x=\bar{x}}\right) \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \frac{1}{2}\sigma_\theta^2 \begin{bmatrix} 0 \\ -1 \end{bmatrix} \end{aligned} \tag{14.16}$$

We therefore obtain

$$\begin{aligned} \bar{y}_1 &\approx 0 \\ \bar{y}_2 &\approx 1 - \frac{\sigma_\theta^2}{2} \\ &= 1 - \frac{E(\tilde{\theta}^2)}{2} \end{aligned} \tag{14.17}$$

Note that we found the exact value of $\bar{y}_2$ in Equation (14.9) to be equal to $E(\cos\tilde{\theta})$. The approximate expression found in Equation (14.17) is the first two nonzero terms of the Taylor series expansion of $E(\cos\tilde{\theta})$.

14.1.2 The covariance of a nonlinear transformation

Now we turn our attention to the covariance of a random variable that undergoes a nonlinear transformation. The covariance of y is given as

$$P_y = E\left[(y - \bar{y})(y - \bar{y})^T\right] \tag{14.18}$$

We can use Equations (14.11) and (14.15) to write $(y - \bar{y})$ as

$$\begin{aligned} y - \bar{y} &= \left[h(\bar{x}) + D_{\tilde{x}}h + \frac{1}{2!}D_{\tilde{x}}^2 h + \cdots\right] - \\ &\quad \left[h(\bar{x}) + \frac{1}{2!}E(D_{\tilde{x}}^2 h) + \frac{1}{4!}E(D_{\tilde{x}}^4 h) + \cdots\right] \\ &= \left[D_{\tilde{x}}h + \frac{1}{2!}D_{\tilde{x}}^2 h + \cdots\right] - \left[\frac{1}{2!}E(D_{\tilde{x}}^2 h) + \frac{1}{4!}E(D_{\tilde{x}}^4 h) + \cdots\right] \end{aligned} \tag{14.19}$$

We substitute this expression into Equation (14.18) and use the same type of reasoning as in the previous section to see that all of the odd-powered terms in the expected value evaluate to zero (assuming that $\tilde{x}$ is zero-mean with a symmetric pdf). This results in

$$\begin{aligned} P_y &= E\left[D_{\tilde{x}}h(D_{\tilde{x}}h)^T\right] + \\ &\quad E\left[\frac{D_{\tilde{x}}h(D_{\tilde{x}}^3 h)^T}{3!} + \frac{D_{\tilde{x}}^2 h(D_{\tilde{x}}^2 h)^T}{2!2!} + \frac{D_{\tilde{x}}^3 h(D_{\tilde{x}}h)^T}{3!}\right] + \\ &\quad E\left(\frac{D_{\tilde{x}}^2 h}{2!}\right) E\left(\frac{D_{\tilde{x}}^2 h}{2!}\right)^T + \cdots \end{aligned} \tag{14.20}$$

The first term on the right side of the above equation can be written as

$$\begin{aligned} E\left[D_{\tilde{x}}h(D_{\tilde{x}}h)^T\right] &= E\left[\left(\sum_{i=1}^{n} \tilde{x}_i \left.\frac{\partial h}{\partial x_i}\right|_{x=\bar{x}}\right)\left(\cdots\right)^T\right] \\ &= E\left[\sum_{i,j} \tilde{x}_i \left.\frac{\partial h}{\partial x_i}\right|_{x=\bar{x}} \left.\frac{\partial h^T}{\partial x_j}\right|_{x=\bar{x}} \tilde{x}_j\right] \\ &= \sum_{i,j} H_i E(\tilde{x}_i \tilde{x}_j) H_j^T \\ &= \sum_{i,j} H_i P_{ij} H_j^T \end{aligned} \tag{14.21}$$

where the partial derivative vector H_i and the expected value P_{ij} are defined by the above equation. Recall from Equation (1.16) that an equation in this form can be written as

$$\begin{aligned} E[D_{\tilde{x}}h(D_{\tilde{x}}h)^T] &= \left.\frac{\partial h}{\partial x}\right|_{x=\bar{x}} P \left.\frac{\partial h^T}{\partial x}\right|_{x=\bar{x}} \\ &= HPH^T \end{aligned} \tag{14.22}$$

where the partial derivative matrix H and the covariance matrix P are defined by the above equation. H_i in Equation (14.21) is the ith column of H, and P_{ij} in Equation (14.21) is the element in the ith row and jth column of $P = E(\tilde{x}\tilde{x}^T)$. We can use this in Equation (14.20) to write the covariance of a nonlinear transformation $y = h(x)$ as follows:

$$
\begin{aligned}
P_y &= HPH^T + E\left[\frac{D_{\bar{x}}h(D_{\bar{x}}^3h)^T}{3!} + \frac{D_{\bar{x}}^2h(D_{\bar{x}}^2h)^T}{2!2!} + \frac{D_{\bar{x}}^3h(D_{\bar{x}}h)^T}{3!}\right] + \\
&\quad E\left(\frac{D_{\bar{x}}^2h}{2!}\right)E\left(\frac{D_{\bar{x}}^2h}{2!}\right)^T + \cdots
\end{aligned} \tag{14.23}
$$

This is the complete Taylor series expansion for the covariance of a nonlinear transformation.

In the EKF, we use only the first term of this expansion to approximate the covariance of the estimation error. For example, if the measurement $y = h(x) + v$ then we see from Equation (10.98) that the covariance of y is approximated as $P_y = HP_xH^T + R$, where H is the partial derivative of h with respect to x, and R is the covariance of v. Likewise, if the state propagates as $x_{k+1} = f(x_k) + w_k$ then we see from Equation (10.100) that the covariance of x is approximately updated as $P_k^- = FP_{k-1}^+F^T + Q$, where F is the partial derivative of $f(x)$ with respect to x, and Q is the covariance of w_k. However, these covariance approximations can result in significant errors if the underlying functions $h(x)$ and $f(x)$ are highly nonlinear.

For example, consider the nonlinear transformation introduced at the beginning of this section. A linear covariance approximation would indicate that $P_y \approx HP_xH^T$, where H and P_x are given as

$$
\begin{aligned}
H &= \left.\frac{\partial h}{\partial x}\right|_{x=\bar{x}} \\
&= \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix}_{x=\bar{x}} \\
&= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \\
P_x &= E\left(\begin{bmatrix} r-\bar{r} \\ \theta-\bar{\theta} \end{bmatrix}\begin{bmatrix} \cdots \end{bmatrix}^T\right) \\
&= \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}
\end{aligned} \tag{14.24}
$$

This gives P_y as follows.

$$
\begin{aligned}
P_y &\approx HP_xH^T \\
&= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \sigma_\theta^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}
\end{aligned} \tag{14.25}
$$

This is an approximation of P_y. However, a more rigorous analysis of P_y can be conducted using Equations (14.1), (14.7), and (14.10):

$$
\begin{aligned}
P_y &= E\left[(y-\bar{y})(y-\bar{y})^T\right] \qquad (14.26) \\
&= E\left[\begin{pmatrix} r\cos\theta \\ r\sin\theta - (\sin\theta_m)/\theta_m \end{pmatrix}\begin{pmatrix} \cdots \end{pmatrix}^T\right] \\
&= E\begin{bmatrix} r^2\cos^2\theta & r^2\cos\theta\sin\theta - r\cos\theta(\sin\theta_m)/\theta_m \\ r^2\cos\theta\sin\theta - r\cos\theta(\sin\theta_m)/\theta_m & (r\sin\theta - (\sin\theta_m)/\theta_m)^2 \end{bmatrix}
\end{aligned}
$$

We again use our assumption that r and θ are independent, r is uniformly distributed with a mean of 1 and a standard deviation of σ_r, and $\theta = \pi/2 + \tilde{\theta}$, with $\tilde{\theta}$ uniformly distributed between $\pm\theta_m$. We can therefore compute

$$\begin{aligned} E(r^2) &= 1 + \sigma_r^2 \\ E(\cos^2\tilde{\theta}) &= \frac{1 - E(\cos 2\tilde{\theta})}{2} \\ E(\cos 2\tilde{\theta}) &= \frac{\sin 2\theta_m}{2\theta_m} \\ E(\sin\theta) &= E(\cos\tilde{\theta}) \\ &= \frac{\sin\theta_m}{\theta_m} \end{aligned} \tag{14.27}$$

We can use these expressions in Equation (14.26) to compute

$$P_y = \begin{bmatrix} \frac{1}{2}(1+\sigma_r^2)(1 - \sin 2\theta_m/2\theta_m) & 0 \\ 0 & \frac{1}{2}(1+\sigma_r^2)(1 + \sin 2\theta_m/2\theta_m) - \sin^2\theta_m/\theta_m^2 \end{bmatrix} \tag{14.28}$$

This matrix defines a two-dimensional ellipse, where $P_y(1,1)$ specifies the square of the y_1 axis length, and $P_y(2,2)$ specifies the square of the y_2 axis length. Figure 14.2 shows the linearized covariance defined by Equation (14.25), and the exact covariance defined by Equation (14.28). The linearized covariance is centered at the linearized mean, and the exact covariance is centered around at the exact mean. It can be seen that the linearized covariance is not a very good approximation to the exact covariance, at least not in the y_2 direction.

Figure 14.2 Linearized and nonlinear mean and covariance of 300 randomly generated points with $\tilde{r}$ uniformly distributed between ± 0.01 and $\tilde{\theta}$ uniformly distributed between ± 0.35 radians.

This is not a Kalman filtering example. But since the EKF uses first-order linearization to update the covariance of the state, this example shows the kind of error that can creep into the EKF when it is applied to a nonlinear system.

14.2 UNSCENTED TRANSFORMATIONS

The problem with nonlinear systems is that it is difficult to transform a probability density function through a general nonlinear function. In the previous section, we were able to obtain exact nonlinear transformations of the mean and covariance, but only for a simple two-dimensional transformation. The extended Kalman filter works on the principle that a linearized transformation of means and covariances is approximately equal to the true nonlinear transformation, but we saw in the previous section that the approximation could be unsatisfactory.

An unscented transformation is based on two fundamental principles. First, it is easy to perform a nonlinear transformation on a single point (rather than an entire pdf). Second, it is not too hard to find a set of individual points in state space whose sample pdf approximates the true pdf of a state vector.

Taking these two ideas together, suppose that we know the mean $\bar{x}$ and covariance P of a vector x. We then find a set of deterministic vectors called sigma points whose ensemble mean and covariance are equal to $\bar{x}$ and P. We next apply our known nonlinear function $y = h(x)$ to each deterministic vector to obtain transformed vectors. The ensemble mean and covariance of the transformed vectors will give a good estimate of the true mean and covariance of y. This is the key to the unscented transformation.

As an example, suppose that x is an $n \times 1$ vector that is transformed by a nonlinear function $y = h(x)$. Choose $2n$ sigma points $x^{(i)}$ as follows:

$$\begin{aligned} x^{(i)} &= \bar{x} + \tilde{x}^{(i)} \qquad i = 1, \cdots, 2n \\ \tilde{x}^{(i)} &= \left(\sqrt{nP}\right)_i^T \qquad i = 1, \cdots, n \\ \tilde{x}^{(n+i)} &= -\left(\sqrt{nP}\right)_i^T \qquad i = 1, \cdots, n \end{aligned} \tag{14.29}$$

where $\sqrt{nP}$ is the matrix square root of nP such that $(\sqrt{nP})^T\sqrt{nP} = nP$, and $(\sqrt{nP})_i$ is the ith row of $\sqrt{nP}$.[1] In the next couple of subsections, we will see how the ensemble mean of the above sigma points can be used to approximate the mean and covariance of a nonlinearly transformed vector.

14.2.1 Mean approximation

Suppose that we have a vector x with a known mean $\bar{x}$ and covariance P, a nonlinear function $y = h(x)$, and we want to approximate the mean of y. We propose transforming each individual sigma point of Equation (14.29) using the nonlinear function $h(\cdot)$, and then taking the weighted sum of the transformed sigma points to approximate the mean of y. The transformed sigma points are computed as follows:

$$y^{(i)} = h\left(x^{(i)}\right) \qquad i = 1, \cdots, 2n \tag{14.30}$$

[1]MATLAB's Cholesky factorization routine CHOL can be used to find a matrix square root. See Section 6.3.1, but note the slight difference between the matrix square root definition used in that section and here.

The true mean of y is denoted as $\bar{y}$. The approximated mean of y is denoted as $\bar{y}_u$ and is computed as follows:

$$\bar{y}_u = \sum_{i=1}^{2n} W^{(i)} y^{(i)} \tag{14.31}$$

The weighting coefficients $W^{(i)}$ are defined as follows:

$$W^{(i)} = \frac{1}{2n} \qquad i = 1, \cdots, 2n \tag{14.32}$$

Equation (14.31) can therefore be written as

$$\bar{y}_u = \frac{1}{2n} \sum_{i=1}^{2n} y^{(i)} \tag{14.33}$$

Now let's compute the value of $\bar{y}_u$ to see how well it matches the true mean of y. To do this we first use Equation (1.89) to expand each $y^{(i)}$ in Equation (14.33) in a Taylor series around $\bar{x}$. This results in

$$\begin{aligned} \bar{y}_u &= \frac{1}{2n} \sum_{i=1}^{2n} \left(h(\bar{x}) + D_{\tilde{x}^{(i)}} h + \frac{1}{2!} D^2_{\tilde{x}^{(i)}} h + \cdots \right) \\ &= h(\bar{x}) + \frac{1}{2n} \sum_{i=1}^{2n} \left(D_{\tilde{x}^{(i)}} h + \frac{1}{2!} D^2_{\tilde{x}^{(i)}} h + \cdots \right) \end{aligned} \tag{14.34}$$

Now notice that for any integer $k \geq 0$ we have

$$\begin{aligned} \sum_{j=1}^{2n} D^{2k+1}_{\tilde{x}^{(j)}} h &= \sum_{j=1}^{2n} \left[\left(\sum_{i=1}^{n} \tilde{x}_i^{(j)} \frac{\partial}{\partial x_i} \right)^{2k+1} h(x) \Big|_{x=\bar{x}} \right] \\ &= \sum_{j=1}^{2n} \left[\sum_{i=1}^{n} \left(\tilde{x}_i^{(j)} \right)^{2k+1} \frac{\partial^{2k+1}}{\partial x_i^{2k+1}} h(x) \Big|_{x=\bar{x}} \right] \\ &= \sum_{i=1}^{n} \left[\sum_{j=1}^{2n} \left(\tilde{x}_i^{(j)} \right)^{2k+1} \frac{\partial^{2k+1}}{\partial x_i^{2k+1}} h(x) \Big|_{x=\bar{x}} \right] \\ &= 0 \end{aligned} \tag{14.35}$$

because from Equation (14.29) $\tilde{x}^{(j)} = -\tilde{x}^{(n+j)}$ $(j = 1, \cdots, n)$. Therefore, all of the odd terms in Equation (14.34) evaluate to zero and we have

$$\begin{aligned} \bar{y}_u &= h(\bar{x}) + \frac{1}{2n} \sum_{i=1}^{2n} \left(\frac{1}{2!} D^2_{\tilde{x}^{(i)}} h + \frac{1}{4!} D^4_{\tilde{x}^{(i)}} h + \cdots \right) \\ &= h(\bar{x}) + \frac{1}{2n} \sum_{i=1}^{2n} \frac{1}{2!} D^2_{\tilde{x}^{(i)}} h + \\ &\quad \frac{1}{2n} \sum_{i=1}^{2n} \left(\frac{1}{4!} D^4_{\tilde{x}^{(i)}} h + \frac{1}{6!} D^6_{\tilde{x}^{(i)}} h + \cdots \right) \end{aligned} \tag{14.36}$$

Now look at the second term on the right side of the above equation:

$$\begin{aligned}
\frac{1}{2n}\sum_{i=1}^{2n}\frac{1}{2!}D^2_{\tilde{x}^{(i)}}h &= \frac{1}{2n}\sum_{k=1}^{2n}\frac{1}{2!}\left(\sum_{i=1}^{n}\tilde{x}_i^{(k)}\frac{\partial}{\partial x_i}\right)^2 h(x)\Big|_{x=\bar{x}} \\
&= \frac{1}{4n}\sum_{k=1}^{2n}\sum_{i,j=1}^{n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2}{\partial x_i\partial x_j}h(x)\Big|_{x=\bar{x}} \\
&= \frac{1}{4n}\sum_{i,j=1}^{n}\sum_{k=1}^{2n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2}{\partial x_i\partial x_j}h(x)\Big|_{x=\bar{x}} \\
&= \frac{1}{2n}\sum_{i,j=1}^{n}\sum_{k=1}^{n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2}{\partial x_i\partial x_j}h(x)\Big|_{x=\bar{x}} \qquad (14.37)
\end{aligned}$$

where we have again used the fact from Equation (14.29) that $\tilde{x}^{(k)} = -\tilde{x}^{(k+n)}$ $(k = 1, \cdots, n)$. Substitute for $\tilde{x}_i^{(k)}$ and $\tilde{x}_j^{(k)}$ from Equation (14.29) in the above equation to obtain

$$\begin{aligned}
\frac{1}{2n}\sum_{i,j=1}^{n}\sum_{k=1}^{n}\tilde{x}_i^{(k)}\tilde{x}_j^{(k)}\frac{\partial^2 h(x)}{\partial x_i\partial x_j}\Big|_{x=\bar{x}} &= \frac{1}{2n}\sum_{i,j=1}^{n}\sum_{k=1}^{n}\left(\sqrt{nP}\right)_{ki}\left(\sqrt{nP}\right)_{kj}\frac{\partial^2 h(x)}{\partial x_i\partial x_j}\Big|_{x=\bar{x}} \\
&= \frac{1}{2n}\sum_{i,j=1}^{n}nP_{ij}\frac{\partial^2 h(x)}{\partial x_i\partial x_j}\Big|_{x=\bar{x}} \\
&= \frac{1}{2}\sum_{i,j=1}^{n}P_{ij}\frac{\partial^2 h(x)}{\partial x_i\partial x_j}\Big|_{x=\bar{x}} \qquad (14.38)
\end{aligned}$$

Equation (14.36) can therefore be written as

$$\begin{aligned}
\bar{y}_u &= h(\bar{x}) + \frac{1}{2}\sum_{i,j=1}^{n}P_{ij}\frac{\partial^2 h}{\partial x_i\partial x_j}\Big|_{x=\bar{x}} + \\
&\quad \frac{1}{2n}\sum_{i=1}^{2n}\left(\frac{1}{4!}D^4_{\tilde{x}^{(i)}}h + \frac{1}{6!}D^6_{\tilde{x}^{(i)}}h + \cdots\right) \qquad (14.39)
\end{aligned}$$

Now recall that the true mean of y is given by Equation (14.15) as

$$\bar{y} = h(\bar{x}) + \frac{1}{2!}E\left[D^2_{\tilde{x}}h\right] + \frac{1}{4!}E\left[D^4_{\tilde{x}}h\right] + \cdots \qquad (14.40)$$

Look at the second term on the right side of the above equation. It can be written as follows:

$$\begin{aligned}
\frac{1}{2!}E\left[D^2_{\tilde{x}}h\right] &= \frac{1}{2!}E\left[\left(\sum_{i=1}^{n}\tilde{x}_i\frac{\partial}{\partial x_i}\right)^2 h(x)\Big|_{x=\bar{x}}\right] \\
&= \frac{1}{2!}E\left[\sum_{i,j=1}^{n}\tilde{x}_i\tilde{x}_j\frac{\partial^2 h}{\partial x_i\partial x_j}\Big|_{x=\bar{x}}\right]
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2!}\sum_{i,j=1}^{n} E(\tilde{x}_i\tilde{x}_j)\left.\frac{\partial^2 h}{\partial x_i \partial x_j}\right|_{x=\bar{x}} \\
&= \frac{1}{2!}\sum_{i,j=1}^{n} P_{ij}\left.\frac{\partial^2 h}{\partial x_i \partial x_j}\right|_{x=\bar{x}}
\end{aligned} \tag{14.41}
$$

We therefore see that $\bar{y}$ can be written from Equation (14.40) as

$$
\begin{aligned}
\bar{y} &= h(\bar{x}) + \frac{1}{2}\sum_{i,j=1}^{n} P_{ij}\left.\frac{\partial^2 h}{\partial x_i \partial x_j}\right|_{x=\bar{x}} + \\
&\quad \frac{1}{4!}E\left[D_{\tilde{x}}^4 h\right] + \frac{1}{6!}E\left[D_{\tilde{x}}^6 h\right] + \cdots
\end{aligned} \tag{14.42}
$$

Comparing this with Equation (14.39) we see that $\bar{y}_u$ (the approximated mean of y) matches the true mean of y correctly up to the third order, whereas linearization only matches the true mean of y up to the first order (see Section 14.1.1). If we compute $\bar{y}_u$ using Equations (14.29), (14.30), and (14.33), then the value of $\bar{y}_u$ will match the true mean of y up to the third order. The biggest difficulty with this algorithm is the matrix square root that is required in Equation (14.29). But the unscented transformation has the computational advantage that the linearization matrix H does not need to be computed. Of course, the greatest advantage of the unscented transformation (relative to linearization) is the increased accuracy of the mean transformation.

14.2.2 Covariance approximation

Now suppose that we want to approximate the covariance of the nonlinearly transformed vector x. That is, we have an n-element vector x with known mean $\bar{x}$ and covariance P, and we have a known nonlinear function $y = h(x)$. We want to estimate the covariance of y. We will denote the estimate as P_u, and we propose using the following equation:

$$
\begin{aligned}
P_u &= \sum_{i=1}^{2n} W^{(i)}(y^{(i)} - y_u)(y^{(i)} - y_u)^T \\
&= \frac{1}{2n}\sum_{i=1}^{2n}(y^{(i)} - y_u)(y^{(i)} - y_u)^T
\end{aligned} \tag{14.43}
$$

where the $y^{(i)}$ vectors are the transformed sigma points that were computed in Equation (14.30), and the $W^{(i)}$ weighting coefficients are the same as those given in Equation (14.32). Expanding this approximation using Equations (1.89) and (14.36) gives the following:

$$
P_u = \frac{1}{2n}\sum_{i=1}^{2n}\left[h(x^{(i)}) - y_u\right]\left[h(x^{(i)}) - y_u\right]^T \tag{14.44}
$$

$$
\begin{aligned}
&= \frac{1}{2n}\sum_{i=1}^{2n}\Bigg[h(\bar{x}) + D_{\tilde{x}^{(i)}}h + \frac{1}{2}D^2_{\tilde{x}^{(i)}}h + \frac{1}{3!}D^3_{\tilde{x}^{(i)}}h + \cdots \\
&\quad -h(\bar{x}) - \frac{1}{2n}\sum_{j=1}^{2n}\left(\frac{1}{2}D^2_{\tilde{x}^{(j)}}h + \frac{1}{4!}D^4_{\tilde{x}^{(j)}}h + \cdots\right)\Bigg]\Bigg[\cdots\Bigg]^T \qquad (14.45)
\end{aligned}
$$

Multiplying this equation out gives

$$
\begin{aligned}
P_u &= \frac{1}{2n}\sum_{i=1}^{2n}\Bigg\{\left(D_{\tilde{x}^{(i)}}h\right)(\cdots)^T + \underbrace{\left[\left(\frac{1}{2}D_{\tilde{x}^{(i)}}h\right)\left(D^2_{\tilde{x}^{(i)}}h\right)^T\right]}_{0} + \underbrace{\Bigg[\cdots\Bigg]^T}_{0} + \\
&\quad \frac{1}{4!}\left(D^2_{\tilde{x}^{(i)}}h\right)(\cdots)^T - \underbrace{\left[D_{\tilde{x}^{(i)}}h\left(\frac{1}{2n}\sum_j \frac{1}{2}D^2_{\tilde{x}^{(i)}}h\right)^T\right]}_{0} - \underbrace{\Bigg[\cdots\Bigg]^T}_{0} + \\
&\quad \frac{1}{4n^2}\left(\sum_j D^2_{\tilde{x}^{(j)}}h\right)\Bigg(\cdots\Bigg)^T - \left[\frac{1}{4n}D^2_{\tilde{x}^{(i)}}h\left(\sum_j D^2_{\tilde{x}^{(j)}}h\right)^T\right] - \Bigg[\cdots\Bigg]^T + \\
&\quad \left[D_{\tilde{x}^{(i)}}h\left(\frac{1}{3!}D^3_{\tilde{x}^{(j)}}h\right)^T\right] + \Bigg[\cdots\Bigg]^T + \cdots\Bigg\} \qquad (14.46)
\end{aligned}
$$

Some of the terms in the above equation are zero as noted above because $\tilde{x}^{(i)} = -\tilde{x}^{(i+n)}$ for $i = 1, \cdots, n$. So the covariance approximation can be written as

$$
P_u = \frac{1}{2n}\sum_{i=1}^{2n}\left(D_{\tilde{x}^{(i)}}h\right)(\cdots)^T + \text{HOT} \qquad (14.47)
$$

where HOT means higher-order terms (i.e., terms to the fourth power and higher). Expanding this equation for P_u while neglecting the higher order terms gives

$$
P_u = \frac{1}{2n}\sum_{i=1}^{2n}\sum_{j,k=1}^{n}\left(\tilde{x}_j^{(i)}\frac{\partial h(\bar{x})}{\partial x_j}\right)\left(\tilde{x}_k^{(i)}\frac{\partial h(\bar{x})}{\partial x_k}\right)^T \qquad (14.48)
$$

Now recall that $\tilde{x}_j^{(i)} = -\tilde{x}_j^{(i+n)}$ and $\tilde{x}_k^{(i)} = -\tilde{x}_k^{(i+n)}$ for $i = 1, \cdots, n$. Therefore, the covariance approximation becomes

$$
\begin{aligned}
P_u &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j,k=1}^{n}\left(\tilde{x}_j^{(i)}\frac{\partial h(\bar{x})}{\partial x_j}\right)\left(\tilde{x}_k^{(i)}\frac{\partial h(\bar{x})}{\partial x_k}\right)^T \\
&= \sum_{j,k=1}^{n} P_{jk}\frac{\partial h(\bar{x})}{\partial x_j}\left(\frac{\partial h(\bar{x})}{\partial x_k}\right)^T \\
&= HPH^T \qquad (14.49)
\end{aligned}
$$

where the last equality comes from Equation (14.22). Comparing this equation for P_u with the true covariance of y from Equation (14.23), we see that Equation (14.43)

approximates the true covariance of y up to the third order (i.e., only terms to the fourth and higher powers are incorrect). This is the same approximation order as the linearization method, as seen on page 439. However, we would intuitively expect the magnitude of the error of the unscented approximation in Equation (14.43) to be smaller than the linear approximation HPH^T, because the unscented approximation at least contains correctly signed terms to the fourth power and higher, whereas the linear approximation does not contain any terms other than HPH^T.

The unscented transformation can be summarized as follows.

The unscented transformation

1. We begin with an n-element vector x with known mean $\bar{x}$ and covariance P. Given a known nonlinear transformation $y = h(x)$, we want to estimate the mean and covariance of y, denoted as $\bar{y}_u$ and P_u.

2. Form $2n$ sigma point vectors $x^{(i)}$ as follows:

$$\begin{aligned} x^{(i)} &= \bar{x} + \tilde{x}^{(i)} \qquad i = 1, \cdots, 2n \\ \tilde{x}^{(i)} &= \left(\sqrt{nP}\right)_i^T \qquad i = 1, \cdots, n \\ \tilde{x}^{(n+i)} &= -\left(\sqrt{nP}\right)_i^T \qquad i = 1, \cdots, n \end{aligned} \tag{14.50}$$

   where $\sqrt{nP}$ is the matrix square root of nP such that $(\sqrt{nP})^T\sqrt{nP} = nP$, and $(\sqrt{nP})_i$ is the ith row of $\sqrt{nP}$.

3. Transform the sigma points as follows:

$$y^{(i)} = h(x^{(i)}) \qquad i = 1, \cdots, 2n \tag{14.51}$$

4. Approximate the mean and covariance of y as follows:

$$\begin{aligned} \bar{y}_u &= \frac{1}{2n}\sum_{i=1}^{2n} y^{(i)} \\ P_u &= \frac{1}{2n}\sum_{i=1}^{2n}\left(y^{(i)} - y_u\right)\left(y^{(i)} - y_u\right)^T \end{aligned} \tag{14.52}$$

■ EXAMPLE 14.1

To illustrate the unscented transformation, consider the nonlinear transformation shown in Equation (14.1). Since there are two independent variables (r and θ), we have $n = 2$. The covariance of P is given as $P = \text{diag}(\sigma_r^2, \sigma_\theta^2)$. Equation (14.32) shows that $W^{(i)} = 1/4$ for $i = 1, 2, 3, 4$. Equation (14.29) shows that the sigma points are determined as

$$\begin{aligned} x^{(1)} &= \bar{x} + \left(\sqrt{nP}\right)_1^T \\ &= \begin{bmatrix} 1 + \sigma_r\sqrt{2} \\ \pi/2 \end{bmatrix} \end{aligned}$$

$$
\begin{aligned}
x^{(2)} &= \bar{x} + \left(\sqrt{nP}\right)_2^T \\
&= \begin{bmatrix} 1 \\ \pi/2 + \sigma_\theta\sqrt{2} \end{bmatrix} \\
x^{(3)} &= \bar{x} - \left(\sqrt{nP}\right)_1^T \\
&= \begin{bmatrix} 1 - \sigma_r\sqrt{2} \\ \pi/2 \end{bmatrix} \\
x^{(4)} &= \bar{x} - \left(\sqrt{nP}\right)_2^T \\
&= \begin{bmatrix} 1 \\ \pi/2 - \sigma_\theta\sqrt{2} \end{bmatrix}
\end{aligned} \tag{14.53}
$$

Computing the nonlinearly transformed sigma points $y^{(i)} = h(x^{(i)})$ gives

$$
\begin{aligned}
y^{(1)} &= \begin{bmatrix} x_1^{(1)} \cos x_2^{(1)} \\ x_1^{(1)} \sin x_2^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 + \sigma_r\sqrt{2} \end{bmatrix} \\
y^{(2)} &= \begin{bmatrix} x_1^{(2)} \cos x_2^{(2)} \\ x_1^{(2)} \sin x_2^{(2)} \end{bmatrix} = \begin{bmatrix} \cos(\pi/2 + \sigma_\theta\sqrt{2}) \\ \sin(\pi/2 + \sigma_\theta\sqrt{2}) \end{bmatrix} \\
y^{(3)} &= \begin{bmatrix} x_1^{(3)} \cos x_2^{(3)} \\ x_1^{(3)} \sin x_2^{(3)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 - \sigma_r\sqrt{2} \end{bmatrix} \\
y^{(4)} &= \begin{bmatrix} x_1^{(4)} \cos x_2^{(4)} \\ x_1^{(4)} \sin x_2^{(4)} \end{bmatrix} = \begin{bmatrix} \cos(\pi/2 - \sigma_\theta\sqrt{2}) \\ \sin(\pi/2 - \sigma_\theta\sqrt{2}) \end{bmatrix}
\end{aligned} \tag{14.54}
$$

Now we can compute the unscented approximation of the mean and covariance of $y = h(x)$ as

$$
\begin{aligned}
\bar{y}_u &= \sum_{i=1}^{4} W^{(i)} y^{(i)} \\
P_u &= \sum_{i=1}^{4} W^{(i)} \left(y^{(i)} - y_u\right)\left(y^{(i)} - y_u\right)^T
\end{aligned} \tag{14.55}
$$

The results of these transformations are shown in Figure 14.3. This shows the improved accuracy of mean and covariance estimation when unscented transformations are used instead of linear approximations. The true mean and the approximate unscented mean are so close that they are plotted on top of each other. The true mean and the approximate unscented mean are both equal to $(0, 0.9797)$ to four significant digits.

$\nabla\nabla\nabla$

14.3 UNSCENTED KALMAN FILTERING

The unscented transformation developed in the previous section can be generalized to give the unscented Kalman filter. After all, the Kalman filter algorithm attempts

Figure 14.3 Results of Example 14.1. A comparison of the exact, linearized, and unscented mean and covariance of 300 randomly generated points with $\tilde{r}$ uniformly distributed between ± 0.01 and $\tilde{\theta}$ uniformly distributed between ± 0.35 radians.

to propagate the mean and covariance of a system using a time-update and a measurement update. If the system is linear, then the mean and covariance can be exactly updated with the Kalman filter (Chapter 5). If the system is nonlinear, then the mean and covariance can be approximately updated with the extended Kalman filter (Section 13.2). However, the EKF is based on linearization, and the previous section showed that unscented transformations are more accurate than linearization for propagating means and covariances. Therefore, we simply replace the EKF equations with unscented transformations to obtain the UKF algorithm. The UKF algorithm can be summarized as follows.

The unscented Kalman filter

1. We have an n-state discrete-time nonlinear system given by

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, t_k) + w_k \\ y_k &= h(x_k, t_k) + v_k \\ w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \end{aligned} \tag{14.56}$$

2. The UKF is initialized as follows.

$$\begin{aligned} \hat{x}_0^+ &= E(x_0) \\ P_0^+ &= E\left[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T\right] \end{aligned} \tag{14.57}$$

3. The following time update equations are used to propagate the state estimate and covariance from one measurement time to the next.

(a) To propagate from time step $(k-1)$ to k, first choose sigma points $x_{k-1}^{(i)}$ as specified in Equation (14.29), with appropriate changes since the current best guess for the mean and covariance of x_k are $\hat{x}_{k-1}^{+}$ and P_{k-1}^{+}:

$$\begin{aligned}
\hat{x}_{k-1}^{(i)} &= \hat{x}_{k-1}^{+} + \tilde{x}^{(i)} \qquad i = 1, \cdots, 2n \\
\tilde{x}^{(i)} &= \left(\sqrt{nP_{k-1}^{+}}\right)_i^T \qquad i = 1, \cdots, n \\
\tilde{x}^{(n+i)} &= -\left(\sqrt{nP_{k-1}^{+}}\right)_i^T \qquad i = 1, \cdots, n
\end{aligned} \tag{14.58}$$

(b) Use the known nonlinear system equation $f(\cdot)$ to transform the sigma points into $\hat{x}_k^{(i)}$ vectors as shown in Equation (14.30), with appropriate changes since our nonlinear transformation is $f(\cdot)$ rather than $h(\cdot)$:

$$\hat{x}_k^{(i)} = f(\hat{x}_{k-1}^{(i)}, u_k, t_k) \tag{14.59}$$

(c) Combine the $\hat{x}_k^{(i)}$ vectors to obtain the *a priori* state estimate at time k. This is based on Equation (14.33):

$$\hat{x}_k^{-} = \frac{1}{2n} \sum_{i=1}^{2n} \hat{x}_k^{(i)} \tag{14.60}$$

(d) Estimate the *a priori* error covariance as shown in Equation (14.43). However, we should add Q_{k-1} to the end of the equation to take the process noise into account:

$$P_k^{-} = \frac{1}{2n} \sum_{i=1}^{2n} \left(\hat{x}_k^{(i)} - \hat{x}_k^{-}\right)\left(\hat{x}_k^{(i)} - \hat{x}_k^{-}\right)^T + Q_{k-1} \tag{14.61}$$

4. Now that the time update equations are done, we implement the measurement-update equations.

(a) Choose sigma points $x_k^{(i)}$ as specified in Equation (14.29), with appropriate changes since the current best guess for the mean and covariance of x_k are $\hat{x}_k^{-}$ and P_k^{-}:

$$\begin{aligned}
\hat{x}_k^{(i)} &= \hat{x}_k^{-} + \tilde{x}^{(i)} \qquad i = 1, \cdots, 2n \\
\tilde{x}^{(i)} &= \left(\sqrt{nP_k^{-}}\right)_i^T \qquad i = 1, \cdots, n \\
\tilde{x}^{(n+i)} &= -\left(\sqrt{nP_k^{-}}\right)_i^T \qquad i = 1, \cdots, n
\end{aligned} \tag{14.62}$$

This step can be omitted if desired. That is, instead of generating new sigma points we can reuse the sigma points that were obtained from the time update. This will save computational effort if we are willing to sacrifice performance.

(b) Use the known nonlinear measurement equation $h(\cdot)$ to transform the sigma points into $\hat{y}_k^{(i)}$ vectors (predicted measurements) as shown in Equation (14.30):

$$\hat{y}_k^{(i)} = h(\hat{x}_k^{(i)}, t_k) \tag{14.63}$$

(c) Combine the $\hat{y}_k^{(i)}$ vectors to obtain the predicted measurement at time k. This is based on Equation (14.33):

$$\hat{y}_k = \frac{1}{2n}\sum_{i=1}^{2n} \hat{y}_k^{(i)} \tag{14.64}$$

(d) Estimate the covariance of the predicted measurement as shown in Equation (14.43). However, we should add R_k to the end of the equation to take the measurement noise into account:

$$P_y = \frac{1}{2n}\sum_{i=1}^{2n} \left(\hat{y}_k^{(i)} - \hat{y}_k\right)\left(\hat{y}_k^{(i)} - \hat{y}_k\right)^T + R_k \tag{14.65}$$

(e) Estimate the cross covariance between $\hat{x}_k^-$ and $\hat{y}_k$ based on Equation (14.43):

$$P_{xy} = \frac{1}{2n}\sum_{i=1}^{2n} \left(\hat{x}_k^{(i)} - \hat{x}_k^-\right)\left(\hat{y}_k^{(i)} - \hat{y}_k\right)^T \tag{14.66}$$

(f) The measurement update of the state estimate can be performed using the normal Kalman filter equations as shown in Equation (10.100):

$$\begin{aligned} K_k &= P_{xy}P_y^{-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - \hat{y}_k) \\ P_k^+ &= P_k^- - K_k P_y K_k^T \end{aligned} \tag{14.67}$$

The algorithm above assumes that the process and measurement equations are linear with respect to the noise, as shown in Equation (14.56). In general, the process and measurement equations may have noise that enters the process and measurement equations nonlinearly. That is,

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, w_k, t_k) \\ y_k &= h(x_k, v_k, t_k) \end{aligned} \tag{14.68}$$

In this case, the UKF algorithm presented above is not rigorous because it treats the noise as additive, as seen in Equations (14.61) and (14.65). To handle this situation, we can augment the noise onto the state vector as shown in [Jul04, Wan01]:

$$x_k^{(a)} = \begin{bmatrix} x_k \\ w_k \\ v_k \end{bmatrix} \tag{14.69}$$

Then we can use the UKF to estimate the augmented state $x_k^{(a)}$. The UKF is initialized as

$$\begin{aligned} \hat{x}_0^{a+} &= \begin{bmatrix} E(x_0) \\ 0 \\ 0 \end{bmatrix} \\ P_0^{a+} &= \begin{bmatrix} E\left[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T\right] & 0 & 0 \\ 0 & Q_0 & 0 \\ 0 & 0 & R_0 \end{bmatrix} \end{aligned} \tag{14.70}$$

Then we use the UKF algorithm presented above, except that we are estimating the augmented mean and covariance, so we remove Q_{k-1} and R_k from Equations (14.61) and (14.65).

■ EXAMPLE 14.2

Suppose we are trying to estimate the altitude x_1, velocity x_2, and constant ballistic coefficient x_3 of a body as it falls toward earth. A range measuring device is located at an altitude a and the horizontal range between the measuring device and the body is M. This system is the same as the one in Example 13.3. The equations for this system are

$$\begin{aligned} \dot{x}_1 &= x_2 + w_1 \\ \dot{x}_2 &= \rho_0 \exp(-x_1/k) x_2^2 x_3/2 - g + w_2 \\ \dot{x}_3 &= w_3 \\ y(t_k) &= \sqrt{M^2 + (x_1(t_k) - a)^2} + v_k \end{aligned} \tag{14.71}$$

As usual, w_i is the noise that affects the ith process equation, and v is the measurement noise. ρ_0 is the air density at sea level, k is a constant that defines the relationship between air density and altitude, and g is the acceleration due to gravity. We will use the continuous-time system equations to simulate the system, and suppose that we obtain range measurements every 0.5 seconds. The constants that we will use are given as

$$\begin{aligned} \rho_0 &= 2 \text{ lb-sec}^2/\text{ft}^4 \\ g &= 32.2 \text{ ft/sec}^2 \\ k &= 20,000 \text{ ft} \\ E[v_k^2] &= 10,000 \text{ ft}^2 \\ E[w_i^2(t)] &= 0 \quad i = 1,2,3 \\ M &= 100,000 \text{ ft} \\ a &= 100,000 \text{ ft} \end{aligned} \tag{14.72}$$

The initial conditions of the system and the estimator are given as

$$\begin{aligned} x_0 &= \begin{bmatrix} 300,000 & -20,000 & 0.001 \end{bmatrix}^T \\ \hat{x}_0^+ &= x_0 \\ P_0^+ &= \begin{bmatrix} 1,000,000 & 0 & 0 \\ 0 & 4,000,000 & 0 \\ 0 & 0 & 10 \end{bmatrix} \end{aligned} \tag{14.73}$$

We use rectangular integration with a step size of 1 msec to simulate the system, the extended Kalman filter, and the unscented Kalman filter for 30 seconds. Figure 14.4 shows the altitude and velocity of the falling body. For the first few seconds, the velocity is constant. But then the air density increases and drag slows the falling object. Toward the end of the simulation, the object has reached a constant terminal velocity as the acceleration due to gravity is canceled by drag.

Figure 14.5 shows typical EKF and UKF estimation-error magnitudes for this system. It is seen that the altitude and velocity estimates both spike around 10 seconds, at which point the altitude of the measuring device and the falling body are about the same, so the measurement gives less information about the body's altitude and velocity. It is seen from the figure that the UKF consistently gives estimates that are one or two orders of magnitude better than the EKF.

Figure 14.4 Altitude and velocity of a falling body for Example 14.2.

$\nabla\nabla\nabla$

14.4 OTHER UNSCENTED TRANSFORMATIONS

The unscented transformation discussed in the previous section is not the only one that exists. In this section, we discuss several other possible transformations. These other transformations can be used if we have some information about the statistics of the noise, or if we are interested in computational savings.

14.4.1 General unscented transformations

We have seen that an accurate mean and covariance approximation for a nonlinear transformation $y = h(x)$ can be obtained by choosing $2n$ sigma points (where n is the dimension of x) as given in Equation (14.29), and approximating the mean and

Figure 14.5 Kalman filter and unscented filter estimation-error magnitudes of the altitude, velocity, and ballistic coefficient of a falling body for Example 14.2.

covariance as given in Equations (14.33) and (14.43). However, it can be shown that the same order of mean and covariance estimation accuracy can be obtained by choosing $(2n+1)$ sigma points $x^{(i)}$ as follows:

$$\begin{aligned} x^{(0)} &= \bar{x} \\ x^{(i)} &= \bar{x} + \tilde{x}^{(i)} \qquad i = 1, \cdots, 2n \\ \tilde{x}^{(i)} &= \left(\sqrt{(n+\kappa)P}\right)_i^T \qquad i = 1, \cdots, n \\ \tilde{x}^{(n+i)} &= -\left(\sqrt{(n+\kappa)P}\right)_i^T \qquad i = 1, \cdots, n \end{aligned} \tag{14.74}$$

The $(2n+1)$ weighting coefficients are given as

$$\begin{aligned} W^{(0)} &= \frac{\kappa}{n+\kappa} \\ W^{(i)} &= \frac{1}{2(n+\kappa)} \qquad i = 1, \cdots, 2n \end{aligned} \tag{14.75}$$

The unscented mean and covariance approximations are computed as

$$\begin{aligned} y^{(i)} &= h\left(x^{(i)}\right) \\ \bar{y}_u &= \sum_{i=0}^{2n} W^{(i)} y^{(i)} \\ P_u &= \sum_{i=0}^{2n} W^{(i)} \left(y^{(i)} - y_u\right)\left(y^{(i)} - y_u\right)^T \end{aligned} \tag{14.76}$$

It can be seen that if $\kappa = 0$ then these definitions reduce to the quantities given in Section 14.2. Any κ value can be used [as long as $(n+\kappa) \neq 0$] and will give a mean

and covariance estimation accuracy with the same order of accuracy as derived in Section 14.2. However, κ can be used to reduce the higher-order errors of the mean and covariance approximation. For example, if x is Gaussian then $\kappa = 3 - n$ will minimize some of the errors in the fourth-order terms in the mean and covariance approximation [Jul00, Jul04].

14.4.2 The simplex unscented transformation

If computational effort is a primary consideration, then a minimum number of sigma points can be chosen to give the order of estimation accuracy derived in the previous section. It can be shown [Jul02a, Jul04] that if x has n elements then the minimum number of sigma points that gives the order of estimation accuracy of the previous section is equal to $(n+1)$. These sigma points are called simplex sigma points. The following algorithm results in $(n+2)$ sigma points, but the number can be reduced to $(n+1)$ by choosing one of the weights to be zero. The simplex sigma-point algorithm can be summarized as follows.

The simplex sigma-point algorithm

1. Choose the weight $W^{(0)} \in [0,1)$. The choice of $W^{(0)}$ affects only the fourth and higher order moments of the set of sigma points [Jul00, Jul02a].

2. Choose the rest of the weights as follows:

$$W^{(i)} = \begin{cases} 2^{-n}(1 - W^{(0)}) & i = 1,2 \\ 2^{i-2}W^{(1)} & i = 3, \cdots, n+1 \end{cases} \tag{14.77}$$

3. Initialize the following one-element vectors:

$$\begin{aligned} \sigma_0^{(1)} &= 0 \\ \sigma_1^{(1)} &= \frac{-1}{\sqrt{2W^{(1)}}} \\ \sigma_2^{(1)} &= \frac{1}{\sqrt{2W^{(1)}}} \end{aligned} \tag{14.78}$$

4. Recursively expand the σ vectors by performing the following steps for $j = 2, \cdots, n$:

$$\sigma_i^{(j)} = \begin{cases} \begin{bmatrix} \sigma_0^{(j-1)} \\ 0 \end{bmatrix} & i = 0 \\ \begin{bmatrix} \sigma_i^{(j-1)} \\ \frac{-1}{\sqrt{2W^{(j+1)}}} \end{bmatrix} & i = 1, \cdots, j \\ \begin{bmatrix} 0_{j-1} \\ \frac{j}{\sqrt{2W^{(j+1)}}} \end{bmatrix} & i = j+1 \end{cases} \tag{14.79}$$

where 0_j is the column vector containing j zeros.

5. After the above recursion is complete we have the n-element vectors $\sigma_i^{(n)}$ $(i = 0, \cdots, n+1)$. We modify the unscented transformation of Equation (14.29) and obtain the sigma points for the unscented transformation as follows:

$$x^{(i)} = \bar{x} + \sqrt{P}\sigma_i^{(n)} \qquad (i = 0, \cdots, n+1) \tag{14.80}$$

We actually have $(n+2)$ sigma points instead of the $(n+1)$ sigma points as we claimed, but if we choose $W^{(0)} = 0$ then the $x^{(0)}$ sigma point can be ignored in the ensuing unscented transformation. The unscented Kalman filter algorithm in Section 14.3 is then modified in the obvious way based on this minimal set of sigma points.

The problem with the simplex UKF is that the ratio of $W^{(n)}$ to $W^{(1)}$ is equal to 2^{n-2}, where n is the dimension of the state vector x. As the dimension of the state increases, this ratio increases and can quickly cause numerical problems. The only reason for using the simplex UKF is the computational savings, and computational savings is an issue only for problems of high dimension (in general). This makes the simplex UKF of limited utility and leads to the spherical unscented transformation in the following section.

14.4.3 The spherical unscented transformation

The unscented transformation discussed in Section 14.2 is numerically stable. However, it requires $2n$ sigma points and may be too computationally expensive for some applications. The simplex unscented transformation discussed in Section 14.4.2 is the cheapest computational unscented transformation but loses numerical stability for problems with a moderately large number of dimensions. The spherical unscented transformation was developed with the goal of rearranging the sigma points of the simplex algorithm in order to obtain better numerical stability [Jul03, Jul04]. The spherical sigma points are chosen with the following algorithm.

The spherical sigma-point algorithm

1. Choose the weight $W^{(0)} \in [0, 1)$. The choice of $W^{(0)}$ affects only the fourth- and higher-order moments of the set of sigma points [Jul00, Jul02a].

2. Choose the rest of the weights as follows:

$$W^{(i)} = \frac{1 - W^{(0)}}{n+1} \qquad i = 1, \cdots, n+1 \tag{14.81}$$

Note that (in contrast to the simplex unscented transformation) all of the weights are identical except for $W^{(0)}$.

3. Initialize the following one-element vectors:

$$\begin{aligned} \sigma_0^{(1)} &= 0 \\ \sigma_1^{(1)} &= \frac{-1}{\sqrt{2W^{(1)}}} \\ \sigma_2^{(1)} &= \frac{1}{\sqrt{2W^{(1)}}} \end{aligned} \tag{14.82}$$

4. Recursively expand the σ vectors by performing the following steps for $j = 2, \cdots, n$:

$$\sigma_i^{(j)} = \begin{cases} \begin{bmatrix} \sigma_0^{(j-1)} \\ 0 \end{bmatrix} & i = 0 \\ \begin{bmatrix} \sigma_{i-1}^{(j-1)} \\ \hline \sqrt{j(j+1)W^{(1)}} \end{bmatrix} & i = 1, \cdots, j \\ \begin{bmatrix} 0_{j-1} \\ j \\ \hline \sqrt{j(j+1)W^{(1)}} \end{bmatrix} & i = j+1 \end{cases} \tag{14.83}$$

where 0_j is the column vector containing j zeros.

5. After the above recursion is complete, we have the n-element vectors $\sigma_i^{(n)}$ $(i = 0, \cdots, n+1)$. As with the simplex sigma points, we actually have $(n+2)$ sigma points above, but if we choose $W^{(0)} = 0$ then the $x^{(0)}$ sigma point can be ignored in the ensuing unscented transformation. We modify the unscented transformation of Equation (14.29) and obtain the sigma points for the unscented transformation as follows:

$$x^{(i)} = \bar{x} + \sqrt{P}\sigma_i^{(n)} \qquad (i = 0, \cdots, n+1) \tag{14.84}$$

The unscented Kalman filter algorithm in Section 14.3 is then modified in the obvious way based on this set of sigma points.

The ratio of the largest element of $\sigma_i^{(n)}$ to the smallest element is

$$\frac{n}{\sqrt{n(n+1)W^{(1)}}} \bigg/ \frac{1}{\sqrt{n(n+1)W^{(1)}}} = n \tag{14.85}$$

so numerical problems should not be an issue for the spherical unscented transformation.

■ EXAMPLE 14.3

Here we consider the falling-body system described in Example 14.2. The initial conditions of the system and the estimator are given as

$$\begin{aligned} x_0 &= \begin{bmatrix} 300,000 & -20,000 & 1/1000 \end{bmatrix}^T \\ \hat{x}_0^+ &= \begin{bmatrix} 303,000 & -20,200 & 1/1010 \end{bmatrix}^T \\ P_0^+ &= \begin{bmatrix} 30,000 & 0 & 0 \\ 0 & 2,000 & 0 \\ 0 & 0 & 1/10,000 \end{bmatrix} \end{aligned} \tag{14.86}$$

We ran 100 Monte Carlo simulations, each with a 60 s simulation time. The average RMS estimation errors of the EKF, standard UKF (six sigma points), simplex UKF (four sigma points since we chose $W^{(0)} = 0$), and spherical UKF (four sigma points since we chose $W^{(0)} = 0$) are given in Table 14.1. The simplex UKF performs best for altitude estimation, with the standard UKF

not far behind. The standard UKF performs best for velocity estimation, and the spherical UKF performs best for ballistic coefficient estimation. The EKF is generally the worst performing of the four state estimators.

Table 14.1 Example 14.3 estimation errors for the extended Kalman filter, the standard unscented Kalman filter with $2n$ sigma points, and the spherical unscented Kalman filter with $(n+1)$ sigma points. The standard UKF generally performs best. The spherical UKF performance and computational effort lie between those of the EKF and the standard UKF.

	Altitude	Velocity	Ballistic Coefficient Reciprocal
EKF	615	173	11.6
UKF	460	112	7.5
Simplex UKF	449	266	80.8
Spherical UKF	578	142	0.4

$\nabla\nabla\nabla$

14.5 SUMMARY

The unscented filter can give greatly improved estimation performance (compared with the extended Kalman filter) for nonlinear systems. In addition, the EKF requires the computation of Jacobians (partial derivative matrices), and the UKF does not use Jacobians. For systems with analytic process and measurement equations (such as Example 14.2), it is easy to compute Jacobians. But some systems are not given in analytical form and it is numerically difficult to compute Jacobians.

The UKF was first published in 1995 [Jul95] and since then has been expounded upon in many publications.[2] Although the UKF is a relatively recent development, it is rapidly finding applications in such areas as aircraft engine health estimation [Dew03], aircraft model estimation [Cam01], neural network training [Wan01], financial forecasting [Wan01], and motor state estimation [Aki03]. In addition, just as in the Kalman filter, the UKF can be implemented in a square root form to effectively increase numerical precision [Van01, Wan01]. Note that a filter based on polynomial approximations of nonlinear functions is presented in [Nor00], and it seems that the UKF is a special case of this filter.

There is a lot of room for development in the area of unscented filtering. A glance through this book's table of contents shows many specialized topics that have been applied to Kalman and H_∞ filtering, revealing a rich source of research topics for unscented filtering. These include UKF stability properties, constrained unscented filtering, unscented smoothing, reduced-order unscented filtering, robust unscented filtering, unscented filtering with delayed measurements, hybrid unscented/H_∞ filtering, and others.

[2]It is interesting to note that the first journal publication of the UKF was submitted for publication in 1994, but did not appear in print until 2000 [Jul00]. Alternative technologies that are highly different than existing approaches tend to meet with resistance, but persistence (if accompanied by technical rigor) can break down barriers.

PROBLEMS

Written exercises

14.1 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = x^2$. What is $\bar{y}$? What is the first-order approximation to $\bar{y}$? What is the second-order approximation to $\bar{y}$?

14.2 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is $\bar{y}$? What is the first-order approximation to $\bar{y}$? What is the second-order approximation to $\bar{y}$? What is the third-order approximation to $\bar{y}$? What is the fourth-order approximation to $\bar{y}$?

14.3 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is the variance of y? What is the first-order approximation to the variance of y? What is the fourth-order approximation to the variance of y?

14.4 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is $\bar{y}$? What is the unscented approximation to $\bar{y}$?

14.5 Consider the matrix

$$P = \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix}$$

Find an upper triangular matrix S (using only paper and pencil) such that $S^T S = P$. Find a lower triangular matrix S such that $S^T S = P$. (Note the difference between your solution to this problem and the solution to Problem 6.7.)

14.6 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is the variance of y? What is unscented approximation to the variance of y?

14.7 Show that for a system with an identity transition matrix, the UKF algorithm gives $\hat{x}_k^- = \hat{x}_{k-1}^+$.

14.8 Show that for a system with $y_k = x_k$, the UKF gain K_k is positive definite.

14.9 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is $\bar{y}$? Use the generalized unscented transformation to approximate $\bar{y}$ with $\kappa = 0$, $\kappa = 1$, and $\kappa = 2$.

14.10 Suppose the RV x is uniformly distributed on $[-1, 1]$, and $y = e^x$. What is the variance of y? Use the generalized unscented transformation to approximate the variance of y with $\kappa = 0$, $\kappa = 1$, and $\kappa = 2$.

14.11 Consider the simplex sigma-point algorithm. Prove that $\sum_i W^{(i)} \sigma_i^{(j)} = 0$ (i.e., the weighted sample mean of the $\sigma_i^{(j)}$ vectors is zero).

14.12 Prove that the sum of the weights in the simplex sigma-point algorithm is equal to 1.

14.13 Consider the simplex sigma-point algorithm. Prove that the $\sum_i W^{(i)} x^{(i)} = \bar{x}$ (i.e., the weighted sample mean of the sigma points is equal to $\bar{x}$). (Hint: Use the results of Problems 14.11 and 14.12.)

Computer exercises

14.14 Design an unscented Kalman filter for the system described in Problem 13.21. Simulate the system and the filter for 60 s. Plot the estimation error for the four states. What is the experimental standard deviation of the estimation error for each of the four states? Based on the steady-state covariance matrix of the filter, what is the theoretical standard deviation of the estimation error for each of the four states? How does this compare with the extended Kalman filter results of Problem 13.21?

14.15 An inverted pendulum on a cart can be modeled as follows [Bay99, Che99].

$$\begin{aligned}
\ddot{\theta} &= \frac{mgl\sin\theta(M+m) - ml\cos\theta(u + ml\dot{\theta}^2\sin\theta - B\dot{d})}{(J+ml^2)(M+m) - m^2l^2\cos^2\theta} \\
\ddot{d} &= \frac{u - ml\ddot{\theta}\cos\theta + ml\dot{\theta}^2\sin\theta - B\dot{d}}{M+m}
\end{aligned}$$

The quantities in the system model are as follows:

$$\begin{aligned}
\theta(0) &= \text{initial angle (0.1 rad)} \\
d(0) &= \text{initial cart displacement (0 rad)} \\
m &= \text{pendulum mass (0.2 kg)} \\
M &= \text{cart mass (1 kg)} \\
g &= \text{acceleration due to gravity (9.81 m/s}^2\text{)} \\
B &= \text{coefficient of friction between cart and ground [0.1 N/(m/s)]} \\
l &= \text{pendulum length (1 m)} \\
r &= \text{pendulum mass radius (0.02 m)} \\
u &= \text{external force applied to cart} \\
J &= \text{pendulum moment of inertia} \\
&= mr^2/2
\end{aligned}$$

where we have assumed that the pendulum mass is concentrated in a cylinder at the end of the pendulum. Define the state of the system as $x = \begin{bmatrix} d & \dot{d} & \theta & \dot{\theta} \end{bmatrix}^T$. The horizontal displacement d is measured every 5 ms with a standard deviation of 0.1 m. The continuous-time process noise is $Q_c = \text{diag}(0, 0.0004, 0, 0.04)$. The system can be linearized (so that an EKF can be used to estimate the state) by assuming that θ is small, so $\cos\theta \approx 1$, $\sin\theta \approx \theta$, and $\dot{\theta}^2 \approx 0$. Suppose that the feedback control signal is given as $u = 40\theta$ and the initial state is perfectly known. Write an EKF and a UKF to estimate the state, where the control is assumed by the filters to be $\hat{u} = 40\hat{\theta}$. Plot the true states and estimated states for a 2 second simulation. Which filter appears to perform better?

CHAPTER 15

The particle filter

> In view of all that we have said in the foregoing sections, the many obstacles we appear to have surmounted, what casts the pall over our victory celebration? It is the curse of dimensionality, a malediction that has plagued the scientist from earliest days.
>
> —Richard Bellman [Bel61]

> We want now to point out that modern computing machines are extremely well suited to perform the procedures described.
>
> —Nicholas Metropolis and S. Ulam [Met49]

Particle filters had their beginnings in the 1940s with the work of Metropolis, and Norbert Wiener suggested something much like particle filtering as early as 1940 [Wie56]. But only since the 1980s has computational power been adequate for their implementation. Even now it is the computational burden of the particle filter that is its primary obstacle to more widespread use. The particle filter is a statistical, brute-force approach to estimation that often works well for problems that are difficult for the conventional Kalman filter (i.e., systems that are highly nonlinear). Particle filtering goes by many other names, including sequential importance sampling [Dou01, Chapter 11], bootstrap filtering [Gor93], the condensation algorithm [Isa96, Mac99], interacting particle approximations [Mor98], Monte Carlo filtering [Kit96], sequential Monte Carlo (SMC) filtering [And04, Cri02], and sur-

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

vival of the fittest [Kan95]. A short discussion on the origins of particle filtering can be found in [Iba01]. Reference books on the particle filter include [Dou01, Ris04].

Particle filters had their origin in Nicolas Metropolis's work in 1949 [Met49], in which he proposed studying systems by investigating the properties of sets of particles rather than the properties of individual particles. He used the analogy of the card game of solitaire. What is the probability of success in a game of solitaire? The probability may be impossible to compute analytically (because of *all* of the possible permutations of play). But if a person plays several hundred games and succeeds in a certain proportion of those games, then the probability of success can be approximated on that basis:

$$\Pr(\text{Success}) \approx \frac{\text{Number of successes}}{\text{Number of trials}} \tag{15.1}$$

This simple idea hearkens back to the definition of probability in Section 2.1. Given the recent invention of the electronic computer at the time, Metropolis's work was certainly ahead of its time. Now that fast, parallel computers are available, his work is beginning to see its fruition in the methods described in this chapter.

As discussed in Chapter 13, the extended Kalman filter (EKF) is the most widely applied state estimation algorithm for nonlinear systems. However, the EKF can be difficult to tune and often gives unreliable estimates if the system nonlinearities are severe. This is because the EKF relies on linearization to propagate the mean and covariance of the state. Chapter 14 discussed the unscented Kalman filter and showed how it reduces linearization errors. We saw that the UKF can provide significant improvements in estimation accuracy over the EKF. However, the UKF is still only an approximate nonlinear estimator. The EKF estimates the mean of a nonlinear system with first-order accuracy, and the UKF improves on this by providing an estimate with higher-order accuracy. However, this simply defers the inevitable divergence that will occur when the system or measurement nonlinearities become too severe.

This chapter presents the particle filter, which is a completely nonlinear state estimator. Of course, there is no free lunch [Ho02]. The price that must be paid for the high performance of the particle filter is an increased level of computational effort. There may be problems for which the improved performance of the particle filter is worth the increased computational effort. There may be other applications for which the improved performance is not worth the extra computational effort. These trade-offs are problem dependent and must be investigated on an individual basis.

The particle filter is a probability-based estimator. Therefore, in Section 15.1, we will discuss the Bayesian approach to state estimation, which will provide a foundation for the derivation of the particle filter. In Section 15.2, we will derive the particle filter. In Section 15.3, we will explore some implementation issues and methods for improving the performance of the particle filter.

15.1 BAYESIAN STATE ESTIMATION

In this section, we will briefly discuss the Bayesian approach to state estimation. This is based on Bayes' Rule, which is discussed in Chapter 2. This section is based on the presentation given in [Gor93], which is similar to many other books and papers on the subject of Bayesian estimation [Dou01, Ris04].

Suppose we have a nonlinear system described by the equations

$$\begin{aligned} x_{k+1} &= f_k(x_k, w_k) \\ y_k &= h_k(x_k, v_k) \end{aligned} \tag{15.2}$$

where k is the time index, x_k is the state, w_k is the process noise, y_k is the measurement, and v_k is the measurement noise. The functions $f_k(\cdot)$ and $h_k(\cdot)$ are time-varying nonlinear system and measurement equations. The noise sequences $\{w_k\}$ and $\{v_k\}$ are assumed to be independent and white with known pdf's. The goal of a Bayesian estimator is to approximate the conditional pdf of x_k based on measurements $y_1, y_2, \cdots, y_k$. This conditional pdf is denoted as

$$p(x_k|Y_k) = \text{pdf of } x_k \text{ conditioned on measurements } y_1, y_2, \cdots, y_k \tag{15.3}$$

The first measurement is obtained at $k = 1$, so the initial condition of the estimator is the pdf of x_0, which can be written as

$$p(x_0) = p(x_0|Y_0) \tag{15.4}$$

since Y_0 is defined as the set of no measurements. Once we compute $p(x_k|Y_k)$ then we can estimate x_k in whatever way we think is most appropriate, depending on the problem. The conditional pdf $p(x_k|Y_k)$ may be multimodal, in which case we may not want to use the mean of x_k as our estimate. For example, suppose that the conditional pdf is computed as shown in Figure 15.1. In this case, the mean of x is 0, but there is zero probability that x is equal to 0, so we may not want to use 0 as our estimate of x. Instead we might want to use fuzzy logic and say that $\hat{x} = \pm 2$, each with a level of membership equal to 0.5 [Lew97].

Figure 15.1 An example of a multimodal probability density function. What single number should be used as an estimate of x?

Our goal is to find a recursive way to compute the conditional pdf $p(x_k|Y_k)$. Before we find this conditional pdf, we will find the conditional pdf $p(x_k|Y_{k-1})$. This

is the pdf of x_k given all measurements *prior to* time k. We can use Equations (2.17) and (2.51) to write this pdf as

$$\begin{aligned} p(x_k|Y_{k-1}) &= \int p[(x_k, x_{k-1})|Y_{k-1}]\, dx_{k-1} \\ &= \int p[x_k|(x_{k-1}, Y_{k-1})]p(x_{k-1}|Y_{k-1})\, dx_{k-1} \end{aligned} \tag{15.5}$$

But notice from our system description in Equation (15.2) that x_k is entirely determined by x_{k-1} and w_{k-1}; therefore $p[x_k|(x_{k-1}, Y_{k-1})] = p(x_k|x_{k-1})$ and we see that

$$p(x_k|Y_{k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|Y_{k-1})\, dx_{k-1} \tag{15.6}$$

The second pdf on the right side of the above equation is not available yet, but it is available at the initial time [see Equation (15.4)]. Later in this section we will see how to compute it recursively. The first pdf on the right side of the above equation is available. The pdf $p(x_k|x_{k-1})$ is simply the pdf of the state at time k given a specific state at time $(k-1)$. We know this pdf because we know the system equation $f_k(\cdot)$ and we know the pdf of the noise w_k (see Section 2.3). For example, suppose that the system equation is given as $x_{k+1} = x_k + w_k$ and suppose that $x_{k-1} = 1$ and w_{k-1} is uniformly distributed on $[-1, 1]$. Then the pdf $p(x_k|x_{k-1})$ is uniformly distributed on $[0, 2]$.

Now consider the *a posteriori* conditional pdf of x_k. We can again use Equations (2.17) and (2.51) to write this pdf as

$$\begin{aligned} p(x_k|Y_k) &= \frac{p(Y_k|x_k)}{p(Y_k)} p(x_k) \\ &= \frac{p[(y_k, Y_{k-1})|x_k]}{p(y_k, Y_{k-1})} \underbrace{\frac{p(x_k|Y_{k-1})p(Y_{k-1})}{p(Y_{k-1}|x_k)}}_{p(x_k)} \\ &= \frac{p(x_k, y_k, Y_{k-1})}{p(x_k)p(y_k, Y_{k-1})} \frac{p(x_k, Y_{k-1})p(Y_{k-1})}{p(Y_{k-1})p(Y_{k-1}|x_k)} \end{aligned} \tag{15.7}$$

We can multiply both the numerator and denominator of this equation by $p(x_k, y_k)$ to obtain

$$p(x_k|Y_k) = \frac{p(x_k, y_k, Y_{k-1})p(x_k, Y_{k-1})p(Y_{k-1})}{p(x_k)p(y_k, Y_{k-1})p(Y_{k-1})p(Y_{k-1}|x_k)} \frac{p(x_k, y_k)}{p(x_k, y_k)} \tag{15.8}$$

Now we use the ratios of various joint pdfs to marginal pdfs in the above equation to obtain conditional pdfs. This gives

$$p(x_k|Y_k) = \frac{p[Y_{k-1}|(x_k, y_k)]p(y_k|x_k)p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})p(Y_{k-1}|x_k)} \tag{15.9}$$

Note that y_k is a function of x_k, so $p[Y_{k-1}|(x_k, y_k)] = p(Y_{k-1}|x_k)$. These two terms cancel in the above equation and we obtain

$$p(x_k|Y_k) = \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})} \tag{15.10}$$

All of the pdf's on the right side of the above equation are available. The pdf $p(y_k|x_k)$ is available from our knowledge of the measurement equation $h_k(\cdot)$ and our knowledge of the pdf of the measurement noise v_k. The pdf $p(x_k|Y_{k-1})$ is available from Equation (15.6). Finally, the pdf $p(y_k|Y_{k-1})$ is obtained (in the same way that Equation (15.5) was obtained) as follows:

$$\begin{aligned} p(y_k|Y_{k-1}) &= \int p[(y_k, x_k)|Y_{k-1}]\, dx_k \\ &= \int p[y_k|(x_k, Y_{k-1})]p(x_k|Y_{k-1})\, dx_k \end{aligned} \tag{15.11}$$

But y_k is completely determined by x_k and v_k, so $p[y_k|(x_k, Y_{k-1})] = p(y_k|x_k)$ and

$$p(y_k|Y_{k-1}) = \int p(y_k|x_k)p(x_k|Y_{k-1})\, dx_k \tag{15.12}$$

Both of the pdf's on the right side of the above equation are available as discussed above. $p(y_k|x_k)$ is available from our knowledge of the measurement equation $h(\cdot)$ and the pdf of v_k, and $p(x_k|Y_{k-1})$ is available from Equation (15.6).

Summarizing the development of this section, the recursive equations of the Bayesian state estimation filter can be summarized as follows.

The recursive Bayesian state estimator

1. The system and measurement equations are given as follows:

$$\begin{aligned} x_{k+1} &= f_k(x_k, w_k) \\ y_k &= h_k(x_k, v_k) \end{aligned} \tag{15.13}$$

   where $\{w_k\}$ and $\{v_k\}$ are independent white noise processes with known pdf's.

2. Assuming that the pdf of the initial state $p(x_0)$ is known, initialize the estimator as follows:

$$p(x_0|Y_0) = p(x_0) \tag{15.14}$$

3. For $k = 1, 2, \cdots$, perform the following.

   (a) The *a priori* pdf is obtained from Equation (15.6).

$$p(x_k|Y_{k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|Y_{k-1})\, dx_{k-1} \tag{15.15}$$

   (b) The *a posteriori* pdf is obtained from Equations (15.10) and (15.12).

$$p(x_k|Y_k) = \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{\int p(y_k|x_k)p(x_k|Y_{k-1})\, dx_k} \tag{15.16}$$

Analytical solutions to these equations are available only for a few special cases. In particular, if $f(\cdot)$ and $h(\cdot)$ are linear, and x_0, $\{w_k\}$, and $\{v_k\}$ are additive, independent, and Gaussian, then the solution is the Kalman filter discussed in Chapter 5. This way of obtaining the Kalman filter is more complicated than the least squares approach that we used in Chapter 5. The Bayesian derivation of the

Kalman filter can be found in many references, including [Rho71], [Spa88, Chapter 6], [Ho64, Wes85], [Kit96a, Chapter 6]. When the Kalman filter is derived this way, then no conclusions can be drawn about the optimality of the filter when the noise is not Gaussian. In fact, other optimal (nonKalman) filters have been derived for other noise distributions [Ser81]. Nevertheless, the Bayesian derivation proves that when the noise is Gaussian, the Kalman filter is the optimal filter. However, the least squares derivation that we used in Chapter 5 shows that the Kalman filter is the optimal *linear* filter, regardless of the pdf of the noise.

15.2 PARTICLE FILTERING

In this section, we derive the basic idea of the particle filter. The particle filter was invented to numerically implement the Bayesian estimator of the previous section. The main idea is intuitive and straightforward. At the beginning of the estimation problem, we randomly generate a given number N state vectors based on the initial pdf $p(x_0)$ (which is assumed to be known). These state vectors are called particles and are denoted as $x_{0,i}^+$ $(i = 1, \cdots, N)$. At each time step $k = 1, 2, \cdots$, we propagate the particles to the next time step using the process equation $f(\cdot)$:

$$x_{k,i}^- = f_{k-1}(x_{k-1,i}^+, w_{k-1}^i) \qquad (i = 1, \cdots, N) \tag{15.17}$$

where each w_{k-1}^i noise vector is randomly generated on the basis of the known pdf of w_{k-1}. After we receive the measurement at time k, we compute the conditional relative likelihood of each particle $x_{k,i}^-$. That is, we evaluate the pdf $p(y_k|x_{k,i}^-)$. As discussed in Section 15.1, this can be done if we know the nonlinear measurement equation and the pdf of the measurement noise. For example, if an m-dimensional measurement equation is given as $y_k = h(x_k) + v_k$ and $v_k \sim N(0, R)$ then a relative likelihood q_i that the measurement is equal to a specific measurement y^*, given the premise that x_k is equal to the particle $x_{k,i}^-$, can be computed as follows [compare with Equation (2.73)].

$$\begin{aligned} q_i &= P[(y_k = y^*)|(x_k = x_{k,i}^-)] \\ &= P[v_k = y^* - h(x_{k,i}^-)] \\ &\sim \frac{1}{(2\pi)^{m/2}|R|^{1/2}} \exp\left(\frac{-[y^* - h(x_{k,i}^-)]^T R^{-1}[y^* - h(x_{k,i}^-)]}{2}\right) \end{aligned} \tag{15.18}$$

The $\sim$ symbol in the above equation means that the probability is not really given by the expression on the right side, but the probability is directly proportional to the right side. So if this equation is used for all the particles $x_{k,i}^-$ $(i = 1, \cdots, N)$, then the *relative* likelihoods that the state is equal to each particle will be correct. Now we normalize the relative likelihoods obtained in Equation (15.18) as follows.

$$q_i = \frac{q_i}{\sum_{j=1}^N q_j} \tag{15.19}$$

This ensures that the sum of all the likelihoods is equal to one. Next we resample the particles from the computed likelihoods. That is, we compute a brand new set of particles $x_{k,i}^+$ that are randomly generated on the basis of the relative likelihoods q_i.

This can be done several different ways. One straightforward (but not necessarily efficient) way is the following [Ris04]. For $i = 1, \cdots, N$, perform the following two steps.

1. Generate a random number r that is uniformly distributed on $[0, 1]$.

2. Accumulate the likelihoods q_i into a sum, one at a time, until the accumulated sum is greater than r. That is, $\sum_{m=1}^{j-1} q_m < r$ but $\sum_{m=1}^{j} q_m \geq r$. The new particle $x_{k,i}^{+}$ is then set equal to the old particle $x_{k,j}^{-}$.

This resampling idea is formally justified in [Smi92], where it is shown that the ensemble pdf of the new particles $x_{k,i}^{+}$ tends to the pdf $p(x_k|y_k)$ as the number of samples N approaches ∞. The resampling step can be summarized as follows:

$$x_{k,i}^{+} = x_{k,j}^{-} \text{ with probability } q_j \qquad (i, j = 1, \cdots, N) \tag{15.20}$$

This is illustrated in Figure 15.2.

Figure 15.2 Illustration of resampling in the particle filter. For example, if a random number $r = 0.3$ is generated (from a distribution that is uniform on $[0, 1]$), the smallest value of j for which $\sum_{m=1}^{j} q_m \geq r$ is $j = 3$. Therefore the resampled particle is set equal to $x_{k,3}^{-}$.

The computational effort of the particle filter is often a bottleneck to its implementation. With this in mind, more efficient resampling methods can be implemented, such as order statistics [Car99, Rip87], stratified sampling and residual sampling [Liu98], and systematic resampling [Kit96]. Other ways of resampling have also been proposed [Mul91]. For example, the *a priori* samples $x_{k,j}^{-}$ $(j = 1, \cdots, N)$ could be accepted as *a posteriori* samples with a probability that is proportional to q_j. However, in this case additional logic must be incorporated to maintain a constant sample size N.

Now we have a set of particles $x_{k,i}^{+}$ that are distributed according to the pdf $p(x_k|y_k)$. We can compute any desired statistical measure of this pdf. For example,

if we want to compute the expected value $E(x_k|y_k)$ then we can approximate it as the algebraic mean of the particles:

$$E(x_k|y_k) \approx \frac{1}{N}\sum_{i=1}^{N} x_{k,i}^{+} \tag{15.21}$$

The particle filter can be summarized as follows.

The particle filter

1. The system and measurement equations are given as follows:

$$\begin{aligned} x_{k+1} &= f_k(x_k, w_k) \\ y_k &= h_k(x_k, v_k) \end{aligned} \tag{15.22}$$

   where $\{w_k\}$ and $\{v_k\}$ are independent white noise processes with known pdf's.

2. Assuming that the pdf of the initial state $p(x_0)$ is known, randomly generate N initial particles on the basis of the pdf $p(x_0)$. These particles are denoted $x_{0,i}^{+}$ $(i = 1, \cdots, N)$. The parameter N is chosen by the user as a trade-off between computational effort and estimation accuracy.

3. For $k = 1, 2, \cdots$, do the following.

   (a) Perform the time propagation step to obtain *a priori* particles $x_{k,i}^{-}$ using the known process equation and the known pdf of the process noise:

$$x_{k,i}^{-} = f_{k-1}(x_{k-1,i}^{+}, w_{k-1}^{i}) \qquad (i = 1, \cdots, N) \tag{15.23}$$

   where each w_{k-1}^{i} noise vector is randomly generated on the basis of the known pdf of w_{k-1}.

   (b) Compute the relative likelihood q_i of each particle $x_{k,i}^{-}$ conditioned on the measurement y_k. This is done by evaluating the pdf $p(y_k|x_{k,i}^{-})$ on the basis of the nonlinear measurement equation and the pdf of the measurement noise.

   (c) Scale the relative likelihoods obtained in the previous step as follows:

$$q_i = \frac{q_i}{\sum_{j=1}^{N} q_j} \tag{15.24}$$

   Now the sum of all the likelihoods is equal to one.

   (d) Generate a set of *a posteriori* particles $x_{k,i}^{+}$ on the basis of the relative likelihoods q_i. This is called the resampling step (for example, see Figure 15.2).

   (e) Now that we have a set of particles $x_{k,i}^{+}$ that are distributed according to the pdf $p(x_k|y_k)$, we can compute any desired statistical measure of this pdf. We typically are most interested in computing the mean and the covariance.

■ **EXAMPLE 15.1**

Suppose that we have a scalar system given by the following equations:

$$\begin{aligned} x_k &= \frac{1}{2}x_{k-1} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos[1.2(k-1)] + w_k \\ y_k &= \frac{1}{20}x_k^2 + v_k \end{aligned} \tag{15.25}$$

where $\{w_k\}$ and $\{v_k\}$ are zero-mean Gaussian white noise sequences, both with variances equal to 1. This system has become a benchmark in the nonlinear estimation literature [Kit87, Gor93]. The high degree of nonlinearity in both the process and measurement equations makes this a difficult state estimation problem for a Kalman filter. We take the initial state as $x_0 = 0.1$, the initial state estimate as $\hat{x}_0 = x_0$, and the initial estimation covariance for the Kalman filter as $P_0^+ = 2$. We can simulate the EKF and the particle filter to estimate the state x. We used a simulation length of 50 time steps, and 100 particles in the particle filter. Figure 15.3 shows the EKF and particle filter estimates of the state. Not only is the EKF estimate poor, but the EKF *thinks* (on the basis of the computed covariance) that the estimate is much better than it really is. The true state is usually farther away from the estimated state than the 95% confidence measure of the EKF (as determined from the covariance P). On the other hand, Figure 15.3 shows that the particle filter does a nice job of estimating the state for this example. The RMS estimation errors for the Kalman and particle filters were 16.3 and 2.6, respectively.

Note that it might be possible to modify the Kalman filter to obtain better performance. For example, some of the procedures discussed in Section 5.5 to prevent divergence could improve the Kalman filter performance in this example. Sometimes, changing the coordinate system of the state space equation or measurement equation can improve performance [Aid83]. Nevertheless, this example shows the type of improvement that can be obtained with the use of particle filtering.

∇∇∇

15.3 IMPLEMENTATION ISSUES

In this section, we discuss a few implementation issues that often arise in the application of particle filters. The methods discussed in this section can significantly improve the performance of the particle filter, and in fact can make the difference between success and failure.

15.3.1 Sample impoverishment

Sample impoverishment occurs when the region of state space in which the pdf $p(y_k|x_k)$ has significant values does not overlap with the pdf $p(x_k|y_{k-1})$. This means that if all of our *a priori* particles are distributed according to $p(x_k|y_{k-1})$, and we then use the computed pdf $p(y_k|x_k)$ to resample the particles, only a few particles will be resampled to become *a posteriori* particles. This is because only a few of the

Figure 15.3 Example 15.1 results. Extended Kalman filter and particle filter estimation performance for a highly nonlinear scalar system.

a priori particles will be in a region of state space where the computed pdf $p(y_k|x_k)$ has a significant value. This means that the resampling process will select only a few distinct *a priori* particles to become *a posteriori* particles. Eventually, all of the particles will collapse to the same value.[1] This problem will be exacerbated if the measurements are not consistent with the process model (modeling errors). This can be overcome by a brute-force method of simply increasing the number of particles N, but this can quickly lead to unreasonable computational demands, and often simply delays the inevitable sample impoverishment. Other more intelligent ways of dealing with this problem can be used [Aru02, Gor93]. In the following subsections we discuss several remedies for sample impoverishment, including roughening, prior editing, regularized particle filtering, Markov chain Monte Carlo resampling, and auxiliary particle filtering.

15.3.1.1 Roughening Roughening can be used to prevent sample impoverishment, as shown in [Dou01, Chapter 14], [Gor93]. In this method, random noise is added to each particle after the resampling process. This is similar to adding artificial process noise to the Kalman filter (see Section 5.5). In the roughening approach, the *a posteriori* particles (i.e., the outputs of the resampling step) are modified as follows:

$$\begin{aligned} x_{k,i}^{+}(m) &= x_{k,i}^{+}(m) + \Delta x(m) \qquad (m = 1, \cdots, n) \\ \Delta x(m) &\sim (0, KM(m)N^{-1/n}) \end{aligned} \tag{15.26}$$

$\Delta x(m)$ is a zero-mean random variable (usually Gaussian). K is a scalar tuning parameter, N is the number of particles, n is the dimension of the state space, and M is a vector containing the maximum difference between the particle elements

[1]This is called the black hole of particle filtering.

before roughening. The mth element of the M vector is given as

$$M(m) = \max_{i,j} |x_{k,i}^{+}(m) - x_{k,j}^{+}(m)| \quad (m = 1, \cdots, n) \tag{15.27}$$

where k is the time step, and i and j are particle numbers. K is a tuning parameter that specifies the amount of jitter that is added to each particle. In [Gor93] the value $K = 0.2$ is used.

■ EXAMPLE 15.2

In this example, we consider the same problem as discussed in Example 14.2. That is, we will try to estimate the altitude, velocity, and ballistic coefficient of a body as it falls toward earth. We use the extended Kalman filter, the unscented Kalman filter, and the particle filter to estimate the system state. A straightforward implementation of the particle filter does not work very well in this example. In order to get good results we had to use the roughening procedure of Equation (15.26) with a tuning parameter $K = 0.2$. We also had to constrain each particle's third element (ballistic coefficient) to a nonnegative value so that the integration of the time-update equations in the particle filter did not diverge. We used 1000 particles. Figure 15.4 shows typical EKF, UKF, and particle filter estimation error magnitudes for this system. It is seen that the particle filter provides performance on par with the UKF, but at the price of much higher computational effort. The UKF is essentially an "intelligent" particle filter with only seven particles (twice the number of states plus one), whereas the particle filter can be viewed as a "brute-force" filter with 1000 particles. Perhaps some additional modifications could be made to the particle filter to obtain better performance, but the same could be said for the UKF.

Figure 15.4 Example 15.2 results. Kalman filter, unscented filter, and particle filter estimation-error magnitudes.

∇∇∇

15.3.1.2 Prior editing If roughening does not prevent sample impoverishment, then prior editing can be tried. This involves rejection of an *a priori* sample if it is in a region of state space with small q_i. If an *a priori* sample is in a region of small probability, then it can be roughened as many times as necessary, using a procedure like Equation (15.26), until it is in a region of significant probability q_i. In [Gor93] prior editing is implemented as follows: if the magnitude of $[y_k - h(x_{k,i}^-)]$ is more than six standard deviations of the measurement noise, then it is highly unlikely to be selected as an *a posteriori* particle. In this case, $x_{k-1,i}^+$ is roughened and then passed through the system equation again to obtain a new $x_{k,i}^-$. This is repeated as many times as necessary until $x_{k,i}^-$ is in a region of nonnegligible probability.

15.3.1.3 Regularized particle filtering Another way of preventing sample impoverishment is through the use of the regularized particle filter (RPF) [Dou01, Chapter 12], [Ris04]. This performs resampling from a continuous approximation of the pdf $p(y_k|x_{k,i}^-)$ rather than from the discrete pdf samples used thus far. Recall in our resampling step in Equation (15.18) that we used the probability

$$q_i = P[(y_k = y^*)|(x_k = x_{k,i}^-)] \tag{15.28}$$

to determine the likelihood of selecting an *a priori* particle to be an *a posteriori* particle. Instead, we can use the pdf $p(x_k|y_k)$ to perform resampling. That is, the probability of selecting the particle $x_{k,i}^-$ to be an *a posteriori* particle is proportional to the pdf $p(x_k|y_k)$ evaluated at $x_k = x_{k,i}^-$. In the RPF, this pdf is approximated as

$$\hat{p}(x_k|y_k) = \sum_{i=1}^{N} w_{k,i} K_h(x_k - x_{k,i}) \tag{15.29}$$

where $w_{k,i}$ are the weights that are used in the approximation. Later on, we will see that these weights should be set equal to the q_i probabilities that were computed in Equation (15.18). $K_h(\cdot)$ is given as

$$K_h(x) = h^{-n} K(x/h) \tag{15.30}$$

where h is the positive scalar kernel bandwidth, and n is the dimension of the state vector. $K(\cdot)$ is a kernel density that is a symmetric pdf that satisfies

$$\begin{aligned} \int x K(x)\, dx &= 0 \\ \int ||x||_2^2 K(x)\, dx &< \infty \end{aligned} \tag{15.31}$$

The kernel $K(\cdot)$ and the bandwidth h are chosen to minimize a measure of the error between the assumed true density $p(x_k|y_k)$ and the approximate density $\hat{p}(x_k|y_k)$:

$$\{K(x), h\} = \text{argmin} \int \left[\hat{p}(x|y_k) - p(x|y_k)\right]^2 dx \tag{15.32}$$

In the classic case of equal weights ($w_{k,i} = 1/N$ for $i = 1, \cdots, N$) the optimal kernel is given as

$$K(x) = \begin{cases} \frac{n+2}{2v_n}(1 - ||x||_2^2) & \text{if } ||x||_2 < 1 \\ 0 & \text{otherwise} \end{cases} \tag{15.33}$$

where v_n is the volume of the n-dimensional unit hypersphere. $K(x)$ is called the Epanechnikov kernel [Dou01, Chapter 12].

An n-dimensional unit hypersphere is a volume in n dimensions in which all points are one unit from the origin [Cox73]. In one dimension, the unit hypersphere is a line with a length of two and a "volume" of two. In two dimensions, the unit hypersphere is a circle with a radius of one and volume π. In three dimensions, the unit hypersphere is a ball with a radius of one and volume $4\pi/3$. In n dimensions, the unit hypersphere has a volume $v_n = 2\pi v_{n-2}/n$.

If $p(x|y_k)$ is Gaussian with an identity covariance matrix then the optimal bandwidth is given as

$$h^* = \left[8v_n^{-1}(n+4)(2\sqrt{\pi})^n\right]^{1/(n+4)} N^{-1/(n+4)} \tag{15.34}$$

In order to handle the case of multimodal pdf's,[2] we should use $h = h^*/2$ [Dou01, Chapter 12],[Sil86]. These choices for the kernel and the bandwidth are optimal only for the case of equal weights and a Gaussian pdf, but they still are often used in other situations to obtain good particle filtering results. Instead of selecting *a priori* particles to become *a posteriori* particles using the probabilities of Equation (15.28), we instead select *a posteriori* particles based on the pdf approximation given in Equation (15.29). This allows more diversity as we perform the update from the *a priori* particles to *a posteriori* particles. In general, we should set the $w_{k,i}$ weights in Equation (15.29) equal to the q_i probabilities shown in Equation (15.28).

Since this procedure assumes that the true density $p(x_k|y_k)$ has a unity covariance matrix, we numerically compute the covariance of the $x_{k,i}^-$ at each time step. Suppose that this covariance is computed as S (an $n \times n$ matrix). Then we compute the matrix square root of S, denoted as A, such that $AA^T = S$ (e.g., we can use Cholesky decomposition for this computation). Then we compute the kernel as

$$K_h(x) = (\det A)^{-1} h^{-n} K(A^{-1}x/h) \tag{15.35}$$

The RPF resampling algorithm can be summarized as follows.

Regularized particle filter resampling

This resampling strategy replaces Step (3d) in the particle filter algorithm on page 468. We have an n-state system, the N *a priori* particles $x_{k,i}^-$, and the N corresponding (normalized) *a priori* probabilities q_i. Generate the *a posteriori* particles $x_{k,i}^+$ as follows.

1. Compute the ensemble mean μ and covariance S of the *a priori* particles as follows.

$$\begin{aligned} \mu &= \frac{1}{N}\sum_{i=1}^{N} x_{k,i}^- \\ S &= \frac{1}{N-1}\sum_{i=1}^{N}(x_{k,i}^- - \mu)(x_{k,i}^- - \mu)^T \end{aligned} \tag{15.36}$$

   Some authors use an N in the denominator of the S equation, but $(N-1)$ gives an unbiased estimate (see Problem 3.6).

[2] A multimodal pdf is one with more than one local maxima. See, for example, Figure 15.1.

2. Perform a square root factorization of S (e.g., a Cholesky factorization) to compute the $n \times n$ matrix A such that $AA^T = S$.

3. Compute the volume of the n-dimensional unit sphere as $v_n = 2\pi v_{n-2}/n$. The starting values for this recursion are $v_1 = 2$, $v_2 = \pi$, and $v_3 = 4\pi/3$.

4. Compute the optimal kernel bandwidth h as follows:

$$h = \frac{1}{2}\left[8v_n^{-1}(n+4)(2\sqrt{\pi})^n\right]^{1/(n+4)} N^{-1/(n+4)} \tag{15.37}$$

   The bandwidth h can be considered a tuning parameter for the particle filter.

5. Approximate the pdf $p(x_k|y_k)$ as follows:

$$\hat{p}(x_k|y_k) = \sum_{i=1}^{N} q_i K_h(x_k - x_{k,i}) \tag{15.38}$$

   where the kernel $K_h(x)$ is given as

$$K_h(x) = (\det A)^{-1} h^{-n} K(A^{-1}x/h) \tag{15.39}$$

   and the Epanechnikov kernel $K(x)$ is given as

$$K(x) = \begin{cases} \frac{n+2}{2v_n}(1 - ||x||_2^2) & \text{if } ||x||_2 < 1 \\ 0 & \text{otherwise} \end{cases} \tag{15.40}$$

   Note that other kernels can also be used in the pdf approximation (see Problem 15.14). Equation (15.38) must be implemented digitally, so the user must choose a certain number of digital values at which to evaluate Equation (15.38). As with the number of particles N, the number of digital values is a trade-off between computational resources and estimation accuracy.

6. Now that we have $\hat{p}(x_k|y_k)$ from the previous step, we generate the *a posteriori* particles $x_{k,i}^+$ by probabilistically selecting points from the pdf approximation $\hat{p}(x_k|y_k)$.

■ EXAMPLE 15.3

Consider the same system as in Example 15.1, except use a process-noise covariance of 0.001 and only three particles ($N = 3$). In this case, the particles in the standard particle filter can quickly degenerate into a single point, but the use of an RPF can prevent this degeneration, increase diversity among the particles, and provide a better state estimate. Twenty Monte Carlo runs of this system result in average RMS errors of 4.6 for the standard particle filter and 3.0 for the RPF. Figure 15.5 shows the improvement that is possible with the use of an RPF.

Figure 15.6 shows the difference between the resampling step of the standard particle filter and the RPF. The standard particle filter has an *a priori* pdf approximation that consists of the sum of impulse functions. Therefore, the *a posteriori* particles are all set equal to one of the *a priori* particles.

However, the RPF has a pdf approximation that is a continuous function of the state estimate. Therefore, the *a posteriori* particles can be equal to any value on the horizontal axis. Of course, when we implement the RPF we have to discretize the horizontal axis in order to choose the *a posteriori* particles, but we can use as fine a discretization as our computational resources will allow.

Figure 15.5 Particle filter estimation performance for the highly nonlinear scalar system of Example 15.3. This shows the improvement that is possible with the use of an RPF.

Figure 15.6 This shows the discrete pdf approximation of the standard particle filter (with three particles), and the continuous pdf approximation of the RPF. This plot is a snapshot of the pdf approximations at one time instant.

∇∇∇

15.3.1.4 Markov chain Monte Carlo resampling Another approach for preventing sample impoverishment is the Markov chain Monte Carlo (MCMC) move step [Gil01, Ris04]. This approach moves the *a priori* particle $x_{k,i}^-$ to a new randomly generated state $\tilde{x}_{k,i}^-$ if a uniformly distributed random number is less than an acceptance probability. The acceptance probability is computed as the probability that the *a priori* sample is consistent with the measurement, relative to the probability that the resampled state is consistent with the measurement. The Metropolis–Hastings acceptance probability [Rob99] is given as

$$\alpha = \min\left[1, \frac{p(y_k|\tilde{x}_{k,i}^-)}{p(y_k|x_{k,i}^-)} \frac{p(\tilde{x}_{k,i}^-|x_{k-1,i}^+)}{p(x_{k,i}^-|x_{k-1,i}^+)}\right] \tag{15.41}$$

The first fraction in the above equation is the ratio of the measurement probability conditioned on the new particle to the measurement probability conditioned on the old particle. The second fraction is the ratio of the probability of the new particle to the probability of the old particle, both conditioned on the particle at the previous time. The acceptance probability is the product of these two fractions, which increases as the probability of the new particle increases. The old *a priori* particle $x_{k,i}^-$ is therefore changed to a new particle $\tilde{x}_{k,i}^-$ if the old particle has a low probability of being selected with the resampling step. This helps to maintain diversity in the particles that come out of the resampling step.

15.3.1.5 Auxiliary particle filtering Another approach to evening out the probability of the *a priori* particles (and thus increasing diversity in the *a posteriori* particles) is called the auxiliary particle filter [Pit99, Ris04]. This approach was developed by augmenting each *a priori* particle by one element (an auxiliary variable). This increases the dimension of the problem and thus adds a degree of freedom to the choice of the resampling weights in Equation (15.19), which allows the resampling weights to be more evenly distributed. Recall from Section 15.2 that the resampling step of the standard particle filter is performed by selecting particles based on their probabilities. These probabilities are given by

$$q_i = P[(y_k = y^*)|(x_k = x_{k,i}^-)] \tag{15.42}$$

where y^* is the actual measurement at time k. The problem with this is that outliers in the batch of *a priori* particles are ignored due to their low probabilities, and the particles can therefore collapse into a single point. Auxiliary particle filtering addresses this issue by changing the resampling probability to the following:

$$q_i = \frac{P[(y_k = y^*)|(x_k = x_{k,i}^-)]}{P[(y_k = y^*)|\mu_{k,i}]} \tag{15.43}$$

where $\mu_{k,i}$ is some statistical characterization of x_k based on $x_{k,i}^-$. For example, we could use $\mu_{k,i} = E(x_k|x_{k,i}^-)$, or $\mu_{k,i} = \text{pdf}(x_k|x_{k,i}^-)$. So compared to the standard particle filter, the resampling probability of the auxiliary particle filter is smaller by a factor of $P[(y_k = y^*)|\mu_{k,i}]$. If the actual measurement is highly likely given $\mu_{k,i}$, then the actual measurement is highly likely given $x_{k,i}^-$. The auxiliary particle filter will then tend to decrease q_i relative to the standard particle filter. Likewise, the auxiliary particle filter will tend to increase q_i for highly unlikely particles. This tends to promote diversity in the population of particles.

Another easy way to smooth out the q_i probabilities is to use something like the following formula.

$$\tilde{q}_i = \frac{(\alpha - 1)q_i + \bar{q}}{\alpha} \tag{15.44}$$

where $\bar{q}$ is the sample mean of all of the q_i probabilities. The parameter $\alpha \in [1, \infty]$ controls how much regularization occurs. If $\alpha \to \infty$ then the regularized probabilities $\tilde{q}_i$ are equal to the standard probabilities q_i. If $\alpha = 1$ then all of the regularized probabilities $\tilde{q}_i$ are equal.

If the dynamics of the state-space system are linear, then there should not be any reason to use auxiliary particle filtering. The existence of outliers in the particles results from nonlinearities. This implies that the use of auxiliary particle filtering is more appropriate when the system nonlinearities are severe. In fact, if the nonlinearities are mild or nonexistent, then the use of auxiliary particle filtering could corrupt the probabilities q_i in an inappropriate way and degrade performance relative to the standard particle filter.

■ **EXAMPLE 15.4**

Consider the same system as in Examples 14.2 and 15.2. That is, we will try to estimate the altitude x_1, velocity x_2, and constant ballistic coefficient x_3 of a body as it falls toward earth. The equations for this system are given in Example 14.2. We use fourth-order Runge–Kutta integration with a step size of 0.5 sec to simulate the system for 30 seconds. We estimate the system states with the standard particle filter and the auxiliary particle filter. As mentioned in Example 15.2, a straightforward implementation of the particle filter does not work very well in this example. In Example 15.2, we used the roughening procedure of Equation (15.26). In this example, we use the auxiliary particle filter of Equation (15.44) with 200 particles. In the standard particle filter, the particles quickly collapse to a single point in state space. In the auxiliary particle filter with $\alpha = 1.1$ (obtained by manual tuning) the diversity of the particles is preserved. Averaged over 10 simulations, the use of the auxiliary particle filter improves altitude estimation by 73%, and improves velocity estimation by 55%. However, the auxiliary particle filter makes the estimate of the ballistic coefficient worse. This may be because the ballistic coefficient is not involved in any nonlinear dynamics in either the system equation or the measurement equation.

∇∇∇

15.3.2 Particle filtering combined with other filters

One approach that has been proposed for improving particle filtering is to combine it with another filter such as the EKF or the UKF [Wan01, Ris04]. In this approach, each particle is updated at the measurement time using the EKF or the UKF, and then resampling is performed using the measurement. This is like running a bank of N Kalman filters (one for each particle) and then adding a resampling step after each measurement. After $x_{k,i}^-$ is obtained as shown in Equation (15.17), it can be refined using the EKF or UKF measurement-update equations. For example, if we want to combine the particle filter with the EKF, then after the measurement is

obtained at time k, $x_{k,i}^-$ is updated to $x_{k,i}^+$ according to the EKF equations shown in Section 13.2:

$$\begin{aligned} P_{k,i}^- &= F_{k-1,i}P_{k-1,i}^+F_{k-1,i}^T + Q_{k-1} \\ K_{k,i} &= P_{k,i}^-H_{k,i}^T(H_{k,i}P_{k,i}^-H_{k,i}^T + R_k)^{-1} \\ x_{k,i}^+ &= x_{k,i}^- + K_{k,i}\left[y_k - h(x_{k,i}^-)\right] \\ P_{k,i}^+ &= (I - K_{k,i}H_{k,i})P_{k,i}^- \end{aligned} \tag{15.45}$$

$K_{k,i}$ is the Kalman gain for the ith particle, and $P_{k,i}^-$ is the *a priori* estimation-error covariance for the ith particle. The partial derivative matrices F and H are defined as

$$\begin{aligned} F_{k-1,i} &= \left.\frac{\partial f}{\partial x}\right|_{x=x_{k-1,i}^+} \\ H_{k,i} &= \left.\frac{\partial h}{\partial x}\right|_{x=x_{k,i}^-} \end{aligned} \tag{15.46}$$

Next, resampling is performed as discussed in Section 15.2 to modify the $x_{k,i}^+$ particles (and their associated covariances $P_{k,i}^+$). This is another way to prevent sample impoverishment because the *a priori* particles $x_{k,i}^-$ are updated on the basis of the measurement at time k before they are resampled. The measurement updates of the particles could be performed with any type of filter – an EKF, a UKF, an H_∞ filter, another particle filter, and so on. The extended Kalman particle filter can be summarized as follows.

The extended Kalman particle filter

1. The system and measurement equations are given as follows:

$$\begin{aligned} x_{k+1} &= f_k(x_k, w_k) \\ y_k &= h_k(x_k, v_k) \end{aligned} \tag{15.47}$$

   where $\{w_k\}$ and $\{v_k\}$ are independent white noise processes with known pdf's.

2. Assuming that the pdf of the initial state $p(x_0)$ is known, randomly generate N initial particles on the basis of the pdf $p(x_0)$. These particles are denoted $x_{0,i}^+$ and their covariances are denoted $P_{0,i}^+ = P_0^+$ $(i = 1, \cdots, N)$. The parameter N is chosen by the user as a trade-off between computational effort and estimation accuracy.

3. For $k = 1, 2, \cdots$, do the following.

   (a) Perform the time propagation step to obtain *a priori* particles $x_{k,i}^-$ and covariances $P_{k,i}^-$ using the known process equation and the known pdf of the process noise:

$$\begin{aligned} x_{k,i}^- &= f_{k-1}(x_{k-1,i}^+, w_{k-1}^i) \\ P_{k,i}^- &= F_{k-1,i}P_{k-1,i}^+F_{k-1,i}^T + Q_{k-1} \\ F_{k-1,i} &= \left.\frac{\partial f}{\partial x}\right|_{x=x_{k-1,i}^+} \end{aligned} \tag{15.48}$$

where each w_{k-1}^i noise vector is randomly generated on the basis of the known pdf of w_{k-1}.

(b) Update the *a priori* particles and covariances to obtain *a posteriori* particles and covariances:

$$\begin{aligned} H_{k,i} &= \left.\frac{\partial h}{\partial x}\right|_{x=x_{k,i}^-} \\ K_{k,i} &= P_{k,i}^- H_{k,i}^T (H_{k,i} P_{k,i}^- H_{k,i}^T + R_k)^{-1} \\ x_{k,i}^+ &= x_{k,i}^- + K_{k,i}\left[y_k - h(x_{k,i}^-)\right] \\ P_{k,i}^+ &= (I - K_{k,i} H_{k,i}) P_{k,i}^- \end{aligned} \tag{15.49}$$

(c) Compute the relative likelihood q_i of each particle $x_{k,i}^+$ conditioned on the measurement y_k. This is done by evaluating the pdf $p(y_k|x_{k,i}^+)$ on the basis of the nonlinear measurement equation and the pdf of the measurement noise.

(d) Scale the relative likelihoods obtained in the previous step as follows:

$$q_i = \frac{q_i}{\sum_{j=1}^N q_j} \tag{15.50}$$

Now the sum of all the likelihoods is equal to one.

(e) Refine the set of *a posteriori* particles $x_{k,i}^+$ and covariances $P_{k,i}^+$ on the basis of the relative likelihoods q_i. This is the resampling step.

(f) Now we have a set of *a posteriori* particles $x_{k,i}^+$ and covariances $P_{k,i}^+$. We can compute any desired statistical measure of this set of particles. We typically are most interested in computing the mean and the covariance.

■ EXAMPLE 15.5

Consider the same system as in Example 15.4. That is, we will try to estimate the altitude x_1, velocity x_2, and constant ballistic coefficient x_3 of a body as it falls toward earth. The equations for this system are given in Example 14.2. We use fourth-order Runge–Kutta integration with a step size of 0.5 sec to simulate the system for 30 s. We use the standard particle filter and the EKF particle filter to estimate the states. The EKF particle filter updates the *a priori* particles at each time based on the measurement, and then the resampling step is performed as usual. In this example, we use 200 particles for the estimator. As mentioned in Example 15.4, in the standard particle filter the particles quickly collapse to a single trajectory. In Example 15.2, we used roughening to improve the particle filter. In Example 15.3, we used the regularized particle filter to improve performance. In Example 15.4, we used the auxiliary particle filter to improve performance. Here we use an EKF particle filter to improve performance. Averaged over 10 simulations, the use of the EKF particle filter improves altitude estimation accuracy by an astounding 99.6%, almost three orders of magnitude. The velocity estimation is only marginally improved, and the ballistic coefficient estimation is marginally degraded.

15.4 SUMMARY

In this chapter, we laid the foundation of Bayesian state estimation, and from there we developed the particle filter. In a linear system with Gaussian noise, the Kalman filter is optimal. In a system that is nonlinear, the Kalman filter can be used for state estimation, but the particle filter may give better results at the price of additional computational effort. In a system that has non-Gaussian noise, the Kalman filter is the optimal linear filter, but again the particle filter may perform better. The unscented Kalman filter provides a balance between the low computational effort of the Kalman filter and the high performance of the particle filter. This is depicted in Figure 15.7.

(a) The above figure depicts the increasing computational effort and increasing accuracy that is obtained by going from an EKF to a UKF to a particle filter. This applies to systems that are nonlinear or non-Gaussian.

(b) The above figure depicts the fact that the Kalman filter is optimal for linear Gaussian systems. Going from a Kalman filter to a UKF to a particle filter will increase computational effort but will not improve estimation accuracy.

Figure 15.7 State estimation trade-offs.

The particle filter has some similarities with the UKF (see Chapter 14) in that it transforms a set of points via known nonlinear equations and combines the results to estimate the mean and covariance of the state. However, in the particle filter the points are chosen randomly, whereas in the UKF the points are chosen on the basis of a specific algorithm. Because of this, the number of points used in a particle filter generally needs to be much greater than the number of points in a UKF. Another difference between the two filters is that the estimation error in a UKF does not converge to zero in any sense, but the estimation error in a particle filter does converge to zero as the number of particles (and hence the computational effort) approaches infinity.

Particle filters have found application in a wide variety of areas, including tracking problems [Ris04], demodulation of communication signals [Dou01, Chapter 4], estimation of ecological parameters and populations [Dou01, Chapter 5], image processing [Dou01, Chapter 16], neural network training [Dou01, Chapter 17], fault detection [deF02], speech recognition [Ver02], and pattern recognition [Dou01, Chapter 26]. Particle filtering is a growing area of research with many unexplored avenues and applications. Some of the more important areas of open research include the avoidance of sample impoverishment, methods for determining how many particles are required for a given problem, convergence results [Cri02], application to control and parameter estimation [Mor03, And04], connections with genetic algo-

rithms [Dou01, Chapter 20], real-time implementation issues [Kwo04], and hardware implementations of parallel particle filters (e.g., in field programmable gate arrays).

PROBLEMS

Written exercises

15.1 Consider the scalar system

$$\begin{aligned} x_{k+1} &= x_k + w_k, \quad & w_k \sim U(-1,1) \\ y_k &= x_k + v_k, \quad & v_k \sim U(-1,1) \end{aligned}$$

where $x_0 \sim U(-1,1)$. Suppose that the first measurement $y_1 = 1$.

a) Use the recursive Bayesian state estimator to find $\text{pdf}(x_1|Y_0)$ and $\text{pdf}(x_1|Y_1)$.

b) What is the Kalman filter estimate $\hat{x}_1^+$? How is $\hat{x}_1^+$ related to $\text{pdf}(x_1|Y_1)$?

15.2 Suppose the pdf of an RV x is given as

$$\text{pdf}(x) = \begin{cases} 1 - x/2 & x \in [0,2] \\ 0 & \text{otherwise} \end{cases}$$

The value of x can be estimated several ways.

a) The maximum-likelihood estimate is written as $\hat{x} = \text{argmax}_x \text{pdf}(x)$. Find the maximum-likelihood estimate of x.

b) The min-max estimate of x is that value of $\hat{x}$ that minimizes the magnitude of the maximum estimation error. Find the min-max estimate of x.

c) The minimum mean square estimate of x is that value of $\hat{x}$ that minimizes $E[(x - \hat{x})^2]$. Find the minimum mean square estimate of x.

d) The expected value estimate of x is given as $\hat{x} = E(x)$. Find $E(x)$.

15.3 Suppose you have a measurement $y_k = x_k^2 + v_k$, where v_k has a triangular pdf that is given as

$$\text{pdf}(v_k) = \begin{cases} 1/2 + v_k/4 & v_k \in [-2,0] \\ 1/2 - v_k/4 & v_k \in [0,2] \\ 0 & \text{otherwise} \end{cases}$$

Suppose that five *a priori* particles $x_{k,i}^-$ are given as -2, -1, 0, 1, and 2, and that the measurement is obtained as $y_k = 1$. What are the normalized likelihoods q_i of each *a priori* particle $x_{k,i}^-$?

15.4 Suppose you have a measurement $y_k = v_k/x_k$, where $v_k \sim N(9,1)$. Suppose that five *a priori* particles $x_{k,i}^-$ are given as 0.8, 0.9, 1.0, 1.1, and 1.2, and that the measurement is obtained as $y_k = 10$. What are the relative likelihoods q_i of each *a priori* particle $x_{k,i}^-$?

15.5 Suppose that five *a priori* particles are found to have probabilities 0.1, 0.1, 0.1, 0.2, and 0.5. The particles are resampled with the basic strategy depicted in Equation (15.20).

a) What is the probability that the first particle will be chosen as an *a posteriori* particle at least once?
b) What is the probability that the fifth particle will be chosen as an *a posteriori* particle at least once?
c) What is the probability that the five *a posteriori* particles will be equal to the five *a priori* particles (disregarding order)?

15.6 Suppose you have the five particles $x_{k,i}^{+} = \{\ 1,\ \ 2,\ \ 3,\ \ -2,\ \ 6\ \}$. What would you propose to use for the estimate of x_k? What would you estimate as the variance of $\hat{x}_k$?

15.7 Suppose that you have five particles -1, -1, 0, 1, and 1. You want to use the roughening procedure of Section 15.3.1.1 to add a uniform random variable with a variance of $KMN^{-1/n}$ to each particle. What range of K will give a probability of at least 1/8 that at least one of the roughened particles is less than -2?

15.8 Suppose you have the system equation $x_{k+1} = x_k$ and the measurement equation $y_k = x_k^2 + v_k$, where v_k has a triangular pdf that is given as

$$\text{pdf}(v_k) = \begin{cases} 1/2 + v_k/4 & v_k \in [-2, 0] \\ 1/2 - v_k/4 & v_k \in [0, 2] \\ 0 & \text{otherwise} \end{cases}$$

Suppose that five *a posteriori* particles $x_{k-1,i}^{+}$ are given as -2, -1, 0, 1, and 2, and that the measurement is obtained as $y_k = 1$. You want to use prior editing to ensure that the -2 particle has at least a 10% chance (after one roughening step) of being selected as an *a posteriori* particle at the next time step. What value of K should you use in your roughening step?

15.9 Suppose you have two particles -1 and $+1$, both with *a priori* probabilities 1/2. Use the kernel bandwidth $h = 1$ with the regularized particle filter to find the pdf approximations $\hat{p}(x_k = -2|y_k)$, $\hat{p}(x_k = -1|y_k)$, $\hat{p}(x_k = 0|y_k)$, $\hat{p}(x_k = 1|y_k)$, and $\hat{p}(x_k = 2|y_k)$. For what values of x_k is the pdf approximation $\hat{p}(x_k|y_k)$ equal to zero?

15.10 Suppose you have N resampling probabilities q_i with sample mean μ and sample variance S. What is the sample mean and variance of the auxiliary probabilities given by Equation (15.44)?

Computer exercises

15.11 Plot the volume of the n-dimensional unit hypersphere as a function of n for $n \in [1, 20]$.

15.12 Consider two particles $x_1 = 1$ and $x_2 = 2$, with equal probabilities. Generate the approximate pdf using the Epanechnikov kernel with bandwidth $h = h^*$. Generate two separate plots (on the same figure) of the two individual terms in the summation of Equation (15.29), and also generate a plot (on the same figure) of their sum. Repeat for three particles $x_1 = 1$, $x_2 = 2$, and $x_3 = 3$ with equal probabilities.

15.13 Repeat Problem 15.12 with $h = h^*/2$ and with $h = 2h^*$. This shows that the bandwidth selection can have a strong effect on the pdf approximation.

15.14 Kernels other than the Epanechnikov kernel can also be used for pdf approximation [Sim98, Dev01]. Some of the more popular kernels can be described in one dimension as follows.

$$\begin{aligned}
\text{Epanechnikov: } K(x) &= \begin{cases} \frac{3}{4}(1-x^2) & |x| < 1 \\ 0 & \text{otherwise} \end{cases} \\
\text{Gaussian: } K(x) &= (2\pi)^{-1/2}\exp(-x^2/2) \\
\text{Uniform: } K(x) &= \begin{cases} \frac{1}{2} & |x| < 1 \\ 0 & \text{otherwise} \end{cases} \\
\text{Triangular: } K(x) &= \begin{cases} 1-x^2 & |x| < 1 \\ 0 & \text{otherwise} \end{cases} \\
\text{Biweight: } K(x) &= \begin{cases} \frac{15}{16}(1-x^2)^2 & |x| < 0 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

Bandwidth selection is another matter, but for this problem you can simply use the optimal Epanechnikov bandwidth for all of the kernels.

a) Repeat Problem 15.12 using Gaussian kernels.
b) Repeat Problem 15.12 using uniform kernels.
c) Repeat Problem 15.12 using triangular kernels.
d) Repeat Problem 15.12 using biweight kernels.

15.15 In this problem, we will explore the performance of the EKF and the particle filter for the system described in Example 15.1.

a) Run 100 simulations of the EKF and the particle filter with $N = 10$, $N = 100$, and $N = 1000$. What is the average RMS state-estimation error for each case?
b) Run 100 simulations of the EKF and the particle filter with $N = 100$ using $Q = 0.1$, $Q = 1$, and $Q = 10$. What is the average RMS state-estimation error for each case?

APPENDIX A

HISTORICAL PERSPECTIVES

> We are like dwarfs on the shoulders of giants, by whose grace we see farther than they. Our study of the works of the ancients enables us to give fresh life to their finer ideas, and rescue them from time's oblivion and man's neglect.
>
> —Peter of Blois, late twelfth century[1]

The Kalman filter has its roots in the early 1700s in the least squares work of Roger Cotes, who died in 1716.[2] However, Cotes's research was vague, without example, and therefore had little influence on later developments in estimation [Sti86]. Least squares estimation began to be more firmly developed in the middle 1700s by Tobias Mayer (estimating the motion of the moon in 1750), Leonard Euler in 1749 and Pierre Laplace in 1787 (estimating the motion of Jupiter and Saturn), Roger Boscovich in 1755 (estimating the dimensions of Earth), and Daniel Bernoulli in 1777 [Ken61]. At the age of 77, Daniel Bernoulli developed the idea of

[1]This quote is usually attributed to Isaac Newton, but as seen from this quote, the idea did not originate with Newton. Peter of Blois penned this analogy in the context of theological knowledge, and Newton, who himself spent much time studying theology, may have been familiar with the idea from Peter of Blois's writings.

[2]Cotes died at the age of 33, having published only one paper during his entire life. Some of his work was published posthumously. Isaac Newton said of him, "had he lived we might have known something."

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

maximum likelihood estimation. Recursive least squares was essentially established by the early 1800s with the work of Karl Gauss (published in 1809, but claimed to have been completed in 1795), Adrien Legendre (1805), and Robert Adrain (1808). Gauss and Legendre's application was estimating the locations of planets and comets on the basis of imperfect measurements, and Adrain's application was surveying. Additional information on the early history of the development of least squares estimation can be found in [Sea67, Sor80, Sor85, Sti86].

In 1880 the Danish astronomer Thorvald Nicolai Thiele extended earlier least squares work and developed a recursive algorithm very similar to the Kalman filter [Hal81, Lau81]. Thiele's filter is equivalent to the Kalman filter for the special case of a scalar state, scalar measurement, state transition and measurement matrices both equal to unity, and deterministic initial state. Thiele also proposed a way to estimate the variances of the state and measurement noise, a precursor to adaptive filtering.

It is interesting to note that most of the early contributors to estimation theory were primarily astronomers rather than mathematicians. They used mathematics as a means to an end. Then, as now, the most outstanding and lasting contributions to theory were driven by practical engineering interests. "There is nothing so practical as a good theory" [Lew51, page 169].

Wiener and Kolmogorov's work in the 1940s was similar to the Kalman filter (see Section 3.4). However, their work did not arise within the context of state-space theory. It is more statistical in nature than Kalman filtering, and requires knowledge of covariances such as $E(x_i x_j^T)$ and $E(y_i x_j^T)$. In order to implement a Wiener filter in a closed form, the theory assumes that the state and measurements are stationary random processes. Furthermore, Wiener filtering is a steady-state process; that is, it assumes that the measurements have been generated from the infinite past. The 1950s saw a lot of work on relaxing the assumptions of the Wiener filter [Zad50, Boo52]. NASA spent several years investigating Wiener theory in the 1950s, but could not see any practical way to implement it in space navigation problems [Sch81].

Later in the 1950s, work began on replacing the covariance knowledge required by the Wiener filter with state-space descriptions. The results of this work were algorithms that are very close to the Kalman filter as we know it today. Work in this direction at Johns Hopkins University was motivated by missile tracking and appeared in unpublished work as early as 1956 [Spa88]. Peter Swerling's work at the RAND Corporation in the late 1950s was motivated by satellite orbit estimation [Swe59]. Swerling essentially developed (and published in 1959) the Kalman filter for the case of noise-free system dynamics. Furthermore, he considered nonlinear system dynamics and measurement equations (because of his application). Similar to the dispute between Gauss and Legendre regarding credit for the development of least squares, there has been a smaller dispute regarding credit for the development of the Kalman filter. After the Kalman filter became well known, Peter Swerling wrote a letter to the *AIAA Journal* claiming credit for the algorithm that bears Kalman's name [Swe63]. For the most part, Swerling's claim has been ignored, but his place in the development of the Kalman filter will surely be remembered. He wrote an appendix to [Bro98] comparing his work with Kalman's. Ruslan Stratonovich in the USSR also obtained the Kalman filter equations in 1960. Richard Battin independently developed the Kalman filter equations from a maximum likelihood point of view. He published his results internally at MIT in

1961 (MIT Instrumentation Laboratory Report R-341), and in the open literature in 1962 [Bat62].

Results similar to Kalman filtering were also derived prior to 1960 in fields other than engineering. For example, work as early as 1950 in statistics and economics resulted in a recursive least squares "Kalman filter" for the case of constant parameter estimation with noisy measurements [Pla50, Thi61]. More details about the relationship between this early work and the Kalman filter can be found in [Did85, Wel87].

Rudolph Kalman developed the discrete-time Kalman filter that we presented in this book in 1960 [Kal60]. Kalman and Bucy developed the continuous-time Kalman filter (discussed in Chapter 8) in 1961 [Kal61].

In view of all the earlier work along the same lines, why was the filter named after Kalman? There were probably several factors that contributed to this [Spa88]. First of all, Kalman wrote his papers in a relatively straightforward way and did not focus on any specific applications. Other similar papers were more application oriented, which tended to obscure the fundamental nature of the theory.[3] Second, Kalman discussed the duality between state estimation and optimal control, which allowed him to specify mathematical conditions for filter stability. Third, the increasing popularity of digital computers at the time of Kalman's papers helped bring his work into the mainstream. Finally, Kalman made his work known to NASA, which needed just such an estimator for the Apollo space program [Sch81, McG85]. The use of the Kalman filter for the Apollo program was facilitated by Stanley Schmidt, who was the branch chief of the NASA Ames Dynamic Analysis Branch in the late 1950s and early 1960s when NASA was beginning feasibility studies of lunar missions. Kalman and Schmidt happened to be close acquaintances during the time that Kalman developed his theory and Schmidt needed a navigation algorithm. During the early 1960s, the Kalman filter was often referred to in papers as the Kalman–Schmidt filter [Bel67]. It is something of an accident of history that the filter was named after Rudolph Kalman, although it is difficult to overstate his contributions to the development of modern control and estimation theory.

Additional interesting notes on the early history of the Kalman filter can be found in [Sor70, Kai74, Lai74, Bat82, Hut84, Gre01].

[3]This can be a lesson for researchers today. As engineers our goal is to gear our work towards practical applications. But (as a rule of thumb) it is the more general, theoretical work that has greater influence on the world and stands the test of time.

APPENDIX B

OTHER BOOKS ON KALMAN FILTERING

Of making many books there is no end, and much study wearies the body.
—Solomon [Ecclesiastes 12:12]

Many books have been written over the years that include Kalman filtering. In this appendix, we give a brief review of some of these books in approximately chronological order.

The earliest book that includes Kalman filtering is probably the one by Richard Battin [Bat64]. His book deals primarily with orbital dynamics and spacecraft guidance, but also includes a chapter titled "Recursive navigation theory," which essentially provides an independent derivation of the Kalman filter and applies it to spacecraft navigation. Battin's book includes an interesting section that discusses the determination of the measurement schedule that minimizes the state estimation-error covariance. Richard Lee's book [Lee64], published a few months later, gives more extensive coverage of the Kalman filter, also referred to in the book as the "Wiener–Kalman filter." Ralph Deutsch's book [Deu65] mostly deals with least squares estimation of constants and Wiener filtering, essentially an expanded version of Chapter 3 of the present book. Deutsch's book is notable in that it contains one chapter on "Kalman and Bucy's recently provided alternate approach to the Wiener–Kolmogorov theory which has certain inviting features." Deutsch's book also contains an interesting chapter that reproduces part of Karl Gauss's orig-

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

inal work on least squares estimation. Other early books that include coverage of Kalman filtering include those by Masanao Aoki [Aok67] and Paul Liebelt [Lie67]. Richard Bucy and Peter Joseph's book [Buc68] deals mostly with continuous-time filtering. It also discusses the second-order Kalman filter for nonlinear systems (see Section 13.3 of the present book) and includes a lot of material related to aerospace applications. James Meditch's book [Med69] was the first to include extensive coverage of both Kalman filtering and its dual, linear quadratic control. These early books on estimation theory are interesting because they were written when a lot of linear systems material that we take for granted today (state-space descriptions, observability, controllability, etc.) were relatively new concepts in the engineering literature.

Andrew Jazwinski's book [Jaz70] emphasizes the Bayesian approach to optimal filtering and includes much material on nonlinear filtering. Andrew Sage and James Melsa's book [Sag71] includes a chapter on decision theory, which is closely related to (but distinct from) estimation theory. Arthur Gelb's book [Gel74] is still considered a classic in the field, probably because it was the earliest readily accessible text on the topic (in terms of mathematical clarity). The book is dated by now, but still continues to provide a good introduction to Kalman filtering. Gerald Bierman's book [Bie77b] is an excellent reference on square root filtering and related topics. He gives a one chapter review of the Kalman filter, and then spends the rest of the book delving into topics such as matrix factorizations and transformations, square root filtering, and U-D filtering. Mark Davis's brief book [Dav77] includes an interesting section on Kalman filtering for distributed parameter systems (i.e., systems with an infinite number of states). Thomas Kailath's edited volume [Kai77] contains reprints of 20 historically important papers on the topics of Wiener filtering and Kalman filtering. Brian Anderson and John Moore's book [And79] has been an important text and reference for many students of optimal filtering, and is noted for its mathematical rigor.

Peter Maybeck wrote a three-volume series covering state estimation and optimal control [May79, May82, May84] that is another classic in the field. The first volume covers the standard linear filtering material, along with one of the earliest discussions of Kalman filtering for GPS/INS integration. The second volume covers more advanced topics in Kalman filtering, such as smoothing, model uncertainties, and nonlinear estimation. The third volume deals with optimal control.

Harold Sorenson's text [Sor80] includes interesting notes about the historical development of parameter estimation techniques, starting with Babylonian astronomers in 300 BC, continuing with least squares estimation in the 18th and 19th centuries, and concluding with the development of Kalman filtering in the 1960s. Sorenson's later edited volume [Sor85] includes reprints of 45 historically important papers in the area of Kalman filtering. It includes reprints of Swerling's paper [Swe59], Kalman and Bucy's papers [Kal60, Kal61], and many other foundational papers. Frank Lewis's book on state estimation [Lew86b] is notable for the amount of material devoted to connections between Wiener filtering and Kalman filtering. Charles Chui and Guanrong Chen's book [Chu87] has a good discussion of decoupled Kalman filtering, which can reduce computational effort (without necessarily using a steady-state filter). Donald Catlin's book [Cat89] is interesting in that it covers Kalman filtering more from a mathematical and statistical point of view rather than from a systems and engineering point of view.

Athanasios Antoulas's edited volume [Ant91] is also worth mentioning. It was published as a tribute to Rudolf Kalman on the occasion of his 60th birthday. It contains 31 papers on topics that were invented or largely influenced by Kalman, such as system theory, Kalman filtering, optimal control, system realization, and system identification. Guanrong Chen's edited volume [Che93] contains a sequence of chapters that deal with Kalman filtering when the underlying assumptions of the filter are not exactly satisfied. These situations include nonlinear systems (see Chapter 13 of the present book), unknown initial conditions, and unmodeled system information. Bozic's brief book [Boz94] presents a treatment of Kalman filtering within the context of digital signal processing. George Siouris's book [Sio96] is quite useful and contains a chapter on decentralized Kalman filtering, and also includes several flowcharts and Fortran code listings for various algorithms. Robert Brown and Patrick Hwang's excellent book [Bro96], currently in its third edition, is an extensive treatment of Kalman filtering and contains two chapters showing how it can be applied to navigation problems. Yaakov Bar-Shalom, X.-Rong Li, and Thiagalingam Kirubarajan's books [Bar98, Bar01] include extensive discussion of tracking and navigation examples, including adaptive estimation and target tracking. They also include companion software that is an interactive MATLAB-based Kalman filter design tool. Eli Brookner's book [Bro98] deals mostly with tracking applications and includes a lot of discussion of the α-β and α-β-γ filters (see Section 7.3 of the present book). It also includes detailed discussions of the transformations that are required for square root filtering (see Section 6.3 of the present book), and concludes with an appendix written by Peter Swerling that compares his work with Kalman's. Thomas Kailath, Ali Sayed, and Babak Hassibi's compendious volumes [Has99, Kai00] are well worth the effort for the serious researcher. Their first book is more of a research monograph, while their second book is more suitable for general classroom use and self-study. Their material is mostly restricted to linear filtering, and is motivated by the Krein space approach that they pioneered. They also scatter a lot of complementary historical background throughout the text.

Paul Zarchan and Howard Musoff's book [Zar00] is light on theory but is full of practical, real-world examples illustrating applications of the Kalman filter. Mohinder Grewal and Angus Andrews's book [Gre01] contains a useful chapter on practical considerations in Kalman filter implementations. John Crassidis and John Junkins's highly recommended book [Cra04] includes a chapter discussing the duality between Kalman filtering and optimal control (see Section 8.5 of the present book). They also have a Web site with MATLAB code for the examples in the book.

Other books that focus on the topic of Kalman filtering include [Sch73, McG74, Kai81, Goo84, Kri84, Che85, Ott85, Ruy85, Bal87, Men87, Cai88, Min93, Bra89, Har89, Ste94], In addition to all of these texts, there are many other books on topics such as optimal control, signal processing, and time series analysis that include chapters or sections devoted to Kalman filtering.

APPENDIX C

STATE ESTIMATION AND THE MEANING OF LIFE

The discipline of the scholar is a consecration to the pursuit of the truth.

—Norbert Wiener [Wie56, p. 358]

The truth will set you free.

—Jesus Christ [John 8:32]

This appendix places state estimation in a larger, more meaningful context in the life of the reader. At first glance, state estimation may not seem to have much to do with The Meaning of Life. After all,

- State estimation is the concern of engineers (and in particular, control engineers). The Meaning of Life is the concern of philosophers.
- State estimation deals with mathematical and physical realities. The Meaning of Life is concerned with spiritual realities.
- State estimation is concerned with the things of this world (the planet Earth and its immediate surroundings). The Meaning of Life is concerned with the things of God.

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

However, in spite of these superficial differences, it is my contention that state estimation is intimately connected with The Meaning of Life. After all, there is only one reality, and both state estimation and The Meaning of Life are both a part of that reality.[1] An analogy from physics can be brought to bear on this point. If we look at a banana and an airplane, they would appear on the surface to be two completely different things with very little in common. However, at a deeper level they are actually similar in many ways. They are both part of the same reality. In fact, both bananas and airplanes are made up of exactly the same electrons, protons, neutrons, and other subatomic particles. Similarly, on a superficial level it appears that state estimation and The Meaning of Life may not have a lot in common. However, at a deeper level they are closely related. Consider the following:

- The Meaning of Life is based on philosophical and theological truth. State estimation is based on mathematical truth.
- God created the universe and all that is in it. This includes philosophical and theological truth, and it also includes mathematical truth.

Many readers will have reasonable doubts about the existence of God, and whether or not he[2] created the universe. Nevertheless, the vast majority of people believe in something or someone that they call God. Without this assumption, I don't think we can go any farther, and so we will use God's existence as a working assumption for now. We will return to the question of this assumption's validity at the end of this appendix.

To be fair, I should also state that I write as a Christian. That is, I believe that the Bible is God's Word, I believe that Jesus Christ offers rescue from evil and death, and I believe the host of other doctrines that historically have characterized evangelical Christianity. Nevertheless, I believe that other religions and worldviews also have a lot of truth, and I believe that all religions (including Christianity) have more similarities with each other than differences (recall the banana/airplane analogy). So although I am a follower of Jesus, I choose to focus in this appendix on the commonalities of all religions and worldviews.

Another implicit assumption that I have made is that The Meaning of Life exists. That is, I am assuming that there is some meaning to life. Again, many readers will have reasonable doubts about this assumption, but the majority of people believe that life does have some meaning. So what is the meaning of life? Philosophies and religions have given a variety of answers to this question. Most of them include something like the following.

- The meaning of life is to pursue pleasure.
- The meaning of life is to love and serve others.
- The meaning of life is to know God.
- The meaning of life is to grow and improve as a person.

[1] At this point I begin making assumptions, such as the assumption here that there is only one reality. Most of the implicit assumptions made in this appendix are widely accepted, but it should be noted that they are indeed assumptions rather than proven facts.

[2] There is no intent here to classify God as male. The pronoun "he" is used for purely historical reasons.

Many religions and worldviews would agree (to at least some extent) with each of these hypotheses for The Meaning of Life.

If God exists, and he created everything (including philosophical, spiritual, and mathematical truth), it follows that there may be some underlying connections between the two seemingly disparate ideas of state estimation and The Meaning of Life.

Consider another analogy. If a certain artist paints a portrait one week, and a landscape the next week, the two paintings may appear upon initial examination to be quite different. But since they were both painted by the same artist, a close examination of the paintings will reveal similarities in style and other interesting connections. Similarly if an author writes a novel one year, and a biography the next year, the two books may appear on the surface to be quite different. But since they were both written by the same author, a close examination of the books will reveal similarities in style and other interesting connections. So we see that if the same God creates both theological truth and mathematical truth, there may not be any apparent connection between the two sets of truth. But since they were both created by the same God, a close examination of the two sets of truth will reveal similarities in style and other interesting connections. Some thought shows that there are indeed interesting connections between state estimation and The Meaning of Life. These connections are explored in the following sections.

Forgiveness and noise suppression

Forgiveness is an essential part of The Meaning of Life. God's nature is such that he forgives humans, and he also requires his followers to forgive others. Many people have a shallow view of forgiveness, thinking that forgiveness of an offense is equivalent to ignoring that offense (hence the popular but damaging phrase "forgive and forget"). A careful examination of religious philosophy shows that forgiveness is actually active rather than passive. Far from ignoring or forgetting an offense, true forgiveness consists of confronting the offense, recognizing it as the wrong that it truly is, actively seeking to benefit the offender, and consciously revoking any attempts at revenge. A person who refuses to forgive hurts himself more than the offender, for the unforgiving person allows a destructive root of hate and bitterness to grow inside him.

Noise suppression in state estimation is similar to forgiveness. A state estimator that does not consider noise is incomplete and does not reflect an accurate view of reality. In fact, noise suppression (filtering) can be considered as one of the primary purposes of state estimation. A state estimator that ignores the presence of noise might exhibit undesirable oscillatory behavior or even instability. The estimator might operate wonderfully in a noise-free environment, but the introduction of noise could render the system useless. A state estimator that is designed to perform well in the presence of noise is like a person who acknowledges the presence of sin in the world but does not allow it to ruin him. Just as the spiritual person deals with offense in a constructive and active way, the optimal state estimator minimizes the effects of noise.

Discernment and bandwidth

In order to grow spiritually, we need to listen and learn from a variety of sources (from all religions and worldviews) because we never know when and how God may try to speak to us. In that sense we need to be essentially open to the data that comes into our lives from others. But if we listen to everything that is within earshot we will be "tossed back and forth by the waves, and blown here and there by every wind of teaching."[3] We need to reject unhealthy data in order to prevent ourselves from being misled. In other words, we can't believe everything we hear or read.[4] We need to strike a healthy balance between skepticism and acceptance of the views of others. We need to exercise discernment in order to allow ourselves to be influenced by beneficial information while rejecting data that may be detrimental.

The band-limited frequency response of a state estimator is similar to spiritual discernment. A state estimator needs to be responsive to input measurements, yet it also needs to reject those parts of the measurements that consist of noise. A state estimator that rejects all measurements is clearly ineffective. Yet a state estimator that is equally sensitive to all measurement data will be "tossed back and forth by the waves, and blown here and there by every wind of measurement." The state estimator needs to strike a healthy balance between acceptance of information content and rejection of the noisy part of measurements.

Fellowship and persistent excitation

People need to be actively involved in fellowship (i.e., spiritually constructive friendships with others) in order to grow spiritually. We need to interact with others, share insights and burdens, and receive the encouragement that others offer. Many people adopt the "Lone Ranger" approach to religion and consider themselves beyond the need for fellowship. But they are like the scientist who tries to conduct research without considering the contributions of the past. We need to be aware that interaction with others will enrich our spiritual lives as we draw on the variegated experiences and insights of others. We will make more progress in our spiritual lives if we stand on the shoulders of the giants who went before us (or at least on the shoulders of the average sized people who accompany us).

Persistent excitation in system identification is similar to spiritual fellowship. In order to estimate the state of a system, we need to have a mathematical model of that system (in general). Even in those systems in which estimation can be performed without a mathematical model, the availability of an accurate system model will always improve estimation performance. One way to obtain a system model is to execute some sort of system identification algorithm. But in order for the system identification algorithm to be effective, it must be excited by an adequate variety of input signals. This is called the "persistent excitation" condition for system identification methods [Jua93, Lju98]. The system model will not be accurate unless the inputs are persistently exciting. Likewise, our lives as spiritual persons will not be all that they can be unless we receive sufficient input from others.

[3] Ephesians 4:14.
[4] Of course, you can't take my word for it.

Spiritual growth and adaptive state estimation

As spiritual beings, we need to grow spiritually in order to be healthy as balanced individuals. Many people appear to be satisfied with their present spiritual status, but God requires us to grow on a continual basis throughout our lives. God is more concerned about the spiritual *direction* that we are moving in than he is with our present spiritual condition. In other words, he is more concerned with velocity than position. We should adopt a mindset that is never complacent but rather continually looks for areas in our lives where we can grow and improve. One of the apostles of the early Christian church, Saint Paul, said toward the end of his life, "Not that I have already obtained all this, or have already been made perfect, but I press on ... forgetting what is behind and straining toward what is ahead, I press on toward the goal..."[5]

Adaptive state estimation is similar to spiritual growth. Some state estimators are static and unchanging in their dynamic characteristics. But a variety of adaptive state estimators have been proposed over the past few decades that exhibit continuous improvement in performance. These adaptive algorithms are never satisfied with their present performance, but continually adjust their parameters in order to obtain incremental improvements over time (see Section 10.4). These adaptive estimators promise the benefit of improved performance and robustness relative to more traditional estimators. In a similar manner, the person who constantly maintains a lookout for areas of possible growth has the promise of many spiritual benefits.

Spiritual perfection and estimator optimality

God requires us to be perfect. To the control engineer, this statement raises the questions, "Perfect in what way? What is the standard for perfection?" Jesus told his followers, "Be perfect, therefore, as your heavenly Father is perfect."[6] So we see that it is God himself who provides the standard for perfection. God himself is the divine objective function. Some people will disagree with the statement that "God requires us to be perfect" because of its obvious impossibility. But in spite of its impossibility, it is a standard toward which God requires us to strive. We will never reach the standard of perfection (at least in this life), but we can continually get (asymptotically) closer to it throughout our lives.[7]

Optimality in state estimation is similar to perfection in life. An optimal state estimator attempts to minimize some objective function. Theoretically, optimality can be achieved. But practically speaking, optimality will never be attained. This is because of modeling errors, incomplete knowledge of noise statistics, sampling and resolution limitations, and other deviations from ideal conditions. Although optimality will never be completely attained, optimal estimators are still quite effective in practice. We do not give up on the notion of optimality just because it is not completely attainable. We continue with our efforts toward optimality, thankful for the performance that we can obtain. The state estimator churns away in its quest for optimality, never quite attaining it, yet continually getting closer

[5]Philippians 3:12-14.
[6]Matthew 5:48.
[7]Those who claim to have already achieved perfection are referred to the paragraphs above.

and never giving up. In a similar manner, we churn away in our quest for spiritual perfection, never quite attaining it, yet continually getting closer and never giving up.

The one true way and the single best estimator

In this book we have discussed a number of different estimators (e.g., Kalman filtering, H_∞ filtering, robust filtering, unscented filtering, and particle filtering). Which filter is the best approach for a given problem? It is not an easy question because one filter may be computationally more effective, another filter may be better from an RMS error viewpoint, another filter may be better from a worst-case error viewpoint, and yet another filter may be better from some other viewpoint. Nevertheless, if the problem and the optimality criterion are well defined, then there is a single filter that is the best. We may not know what the best filter is, but there is a single best filter for the problem. One reason that we may never find the best filter for our problem is because we are stuck on a specific filtering approach and are unable to take the time to learn other competing approaches. If we are comfortable with filter x and we have never been exposed to competing approaches, then we will probably use filter x for every problem. This will prevent us from obtaining the better performance that we might have gotten with a different filter. To some extent this problem is unavoidable. After all, who has the time or energy to learn every filtering algorithm that has ever been proposed? But to some extent this problem is avoidable. After all, with some expenditure of effort on our part we can learn about new filters and have a better chance of knowing the right filter for new problems that we encounter.

As we spend our lives searching for The Meaning of Life, we are confronted with the question of which worldview is the best approach to use in our search. It is not an easy question because one worldview may be better from one point of view, while another worldview may be better from another point of view. Nevertheless, there is a single worldview that is ultimately the best. We may end our lives never having found the best worldview, but it is out there somewhere. One reason that we may never find the best worldview is because we are stuck on the specific worldview that we grew up with and are unable to take the time to learn about others. To some extent this problem is unavoidable. After all, who has the time or energy to conduct an exhaustive study of every religion and philosophy that has ever been proposed? But to some extent this problem is avoidable. After all, with some expenditure of effort on our part we can learn about the most widely adopted religions and have a better chance of knowing the best approach to finding The Meaning of Life.

Conclusion

Earlier in this appendix (page 494) I promised to return to the question of the validity of the assumption of God's existence. In order to deal with this question, we turn to Occam's razor. William of Occam, who lived in the 14th century, was an English philosopher and Christian theologian. He is most famous for the invention of Occam's razor, also called the principle of parsimony. The idea of Occam's razor is that the simplest explanation is the most reasonable explanation. Occam's razor is used to "shave off" those concepts that are not really needed to explain

some observed phenomenon. This idea is used in system identification to accept the simplest model structure that fits the observed data. Occam's razor is used implicitly in all fields of science and engineering (and in everyday life as well) to support the simplest explanation for observed data.

Consider the following example. If I come home and find crayon marks on the wall, I can theorize that a mysterious chemical reaction caused the paint on the wall to change color, or perhaps a burglar broke into the house and colored my walls, or perhaps my young daughter with a fondness for coloring did it. Which explanation is most likely? Occam's razor says to accept the simplest explanation. The simplest explanation is not always correct, but experience has taught us that it is usually correct, and it is certainly more satisfying (although it is not necessarily more satisfying to my daughter).

When we look at the complexity of life with its underlying unity, Occam's razor says to accept the simplest explanation. Bananas and airplanes are both made from the same stuff, and state estimation and The Meaning of Life have an underlying commonality. We see two paintings with similar artistic styles. We see two books with similar writing styles. Is it a coincidence, or is there a simpler explanation? Occam's razor says to accept God as the simplest explanation. The underlying unity that we see in the complexity of life is an evidence for the existence of God.

Some would say that God is more complicated than anything that we directly observe. Therefore, introducing God as an explanation introduces unwarranted complexity and thus actually violates Occam's razor. In this brief appendix, I have neither the time nor the ability to delve into the many deep philosophical arguments for and against the existence of God. Nevertheless, I believe that the existence of God explains so many things that we observe in life that it is a clear example of Occam's razor. Although God is certainly complicated and cannot be proven to be necessary, the addition of one complicated factor to explain a million simple observations is appealing from both an aesthetic and an engineering viewpoint.

REFERENCES

[Abo03] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank, *Matrix Riccati Equations in Control and Systems Theory,* Birkhauser, Boston, Massachusetts, 2003.

[Aid83] V. Aidala and S. Hammel, "Utilization of modified polar coordinates for bearings-only tracking," *IEEE Transactions on Automatic Control,* **AC-28**(3), pp. 283-294 (March 1983).

[Aki03] B. Akin, U. Orguner, and A. Ersak, "State estimation of induction motor using unscented Kalman filter," *Conference on Control Applications,* pp. 915-919 (June 2003).

[Ale91] H. Alexander, "State estimation for distributed systems with sensing delay," *Proceedings of the SPIE, Data Structures and Target Classification,* **1470**, pp. 103-111 (1991).

[Als72] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Transactions on Automatic Control,* **AC-17**(4), pp. 439-448 (August 1972).

[Als74] D. Alspach, "Gaussian sum approximations in nonlinear filtering and control," *Information Sciences,* **7**, pp. 271-290 (1974).

[Als74b] D. Alspach, "A parallel filtering algorithm for linear systems with unknown time varying noise statistics," *IEEE Transactions on Automatic Control,* **AC-19**(5), pp. 552-556 (October 1974).

[And79] B. Anderson and J. Moore, *Optimal Filtering*, Prentice-Hall, Englewood Cliffs, New Jersey, 1979.

[And68] A. Andrews, "A square root formulation of the Kalman covariance equations," *AIAA Journal,* **6**(6), pp. 1165-1166 (June 1968).

ISBN 0471708585

©2006 John Wiley & Sons, Inc.

[And04] C. Andrieu, A. Doucet, S. Singh, and V. Tadic, "Particle methods for change detection, system identification, and control," *Proceedings of the IEEE,* **92**(3), pp. 42-438 (March 2004).

[Ant91] A. Antoulas, *Mathematical System Theory*, Springer-Verlag, New York, 1991.

[Aok65] M. Aoki, "Optimal Bayesian and min-max control of a class of stochastic and adaptive dynamic systems," *Symposium on Systems Engineering for Control System Design,* pp. 77-84 (1965).

[Aok67] M. Aoki, *Optimization of Stochastic Systems,* Academic Press, New York, 1967.

[Ara94] J. Aranda, J. De La Cruz, S. Dormido, P. Ruiperez, and R. Hernandez, "Reduced-order Kalman filter for alignment," *Cybernetics and Systems,* **25**, pp. 1-16 (1994).

[Aru02] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing,* **50**(2), pp. 174-188 (February 2002).

[Ath68] M. Athans, R. Wishner, and A. Bertolini, "Suboptimal state estimation for continuous-time nonlinear systems from discrete measurements," *IEEE Transactions on Automatic Control,* **AC-13**(5), pp. 504-514 (October 1968).

[Ath77] M. Athans, R. Whiting, and M. Gruber, "A suboptimal estimation algorithm with probabilistic editing for false measurements with applications to target tracking with wake phenomena," *IEEE Transactions on Automatic Control,* **AC-22**(3), pp. 372-384 (June 1977).

[Atk89] K. Atkinson, *An Introduction to Numerical Analysis,* John Wiley & Sons, New York, 1989.

[Bai95] E. Bai, K. Nagpal, and R. Tempo, "Bounded error parameter estimation: Noise models, recursive algorithms and H_∞ optimality," *American Control Conference,* pp. 3065-3069, June 1995.

[Bal87] A. Balakrishnan, *Kalman Filtering Theory,* Optimization Software, New York, 1987.

[Bal01] J. Ballabrera-Poy, A. Busalacchi, and R. Murtugudde, "Application of a reduced-order Kalman filter to initialize a coupled atmosphere-ocean model: Impact on the prediction of El Nino," *Journal of Climate,* **14**, pp. 1720-1737 (April 2001).

[Ban91] R. Banavar and J. Speyer, "A linear-quadratic game approach to estimation and smoothing," *American Control Conference,* pp. 2818-2822 (1991).

[Ban92] R. Banavar, "A game theoretic approach to linear dynamic estimation," Doctoral Dissertation, University of Texas at Austin, May 1992.

[Bar83] I. Bar-Itzhack and Y. Medan, "Efficient square root algorithm for measurement update in Kalman filtering," *Journal of Guidance, Control, and Dynamics,* **6**(3), pp. 129-134 (May 1983).

[Bar78] Y. Bar-Shalom, "Tracking methods in a multitarget environment," *IEEE Transactions on Automatic Control,* **AC-23**(4), pp. 618-626 (August 1978).

[Bar95] Y. Bar-Shalom and X. Li, *Multitarget-Multisensor Tracking: Principles and Techniques,* YBS Publishing, Storrs, Connecticut, 1995.

[Bar98] Y. Bar-Shalom and X. Li, *Estimation and Tracking: Principles, Techniques, and Software,* YBS Publishing, Storrs, Connecticut, 1998.

[Bar01] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation,* John Wiley & Sons, New York, 2001.

[Bar02] Y. Bar-Shalom, "Update with out-of-sequence measurements in tracking: Exact solution," *IEEE Transactions on Aerospace and Electronic Systems,* **38**(3), pp. 769-778 (July 2002).

[Bar05] K. Barbosa, C. de Souza, and A. Trofino, "Robust H_2 filtering for uncertain linear systems: LMI based methods with parametric Lyapunov functions," *Systems and Control Letters,* **54**(3), pp. 251-262 (March 2005).

[Bas99] O. Bas, B. Shafai, and S. Linder, "Design of optimal gains for the proportional integral Kalman filter with application to single particle tracking," *IEEE Conference on Decision and Control,* pp. 4567-4571 (December 1999).

[Bas66] R. Bass, V. Norum, and L. Swartz, "Optimal multichannel nonlinear filtering," *Journal of Mathematical Analysis and Applications,* **16**(1), pp. 152-164 (1966).

[Bat62] R. Battin, "A statistical optimizing navigation procedure for space flight," *American Rocket Society Journal,* **32**, pp. 1681-1696 (November 1962).

[Bat64] R. Battin, *Astronautical Guidance,* McGraw-Hill, New York, 1964.

[Bat82] R. Battin, "Space guidance evolution – A personal narrative," *Journal of Guidance,* **5**(2), pp. 97-110 (March-April 1982).

[Bay99] J. Bay, *Fundamentals of Linear State Space Systems,* McGraw-Hill, New York, 1999.

[Bel67] J. Bellantoni and K. Dodge, "A square root formulation of the Kalman-Schmidt filter," *AIAA Journal,* **5**, pp. 1309-1314, 1967.

[Bel60] R. Bellman, *Introduction to Matrix Analysis,* McGraw-Hill, New York, 1960.

[Bel61] R. Bellman, *Adaptive Control Processes: A Guided Tour,* Princeton University Press, Princeton, New Jersey, 1961.

[Bel80] R. Bellman, "Some consequences of the nonnegativity of the elements of the matrix exponential," *Nonlinear Analysis, Theory, Methods and Applications,* **4**(4), pp. 735-736 (1980).

[Ben62] T. Benedict and G. Bordner, "Synthesis of an optimal set of radar track-while-scan smoothing equations," *IRE Transactions on Automatic Control,* **AC-7**(4), pp. 27-32 (July 1962).

[Ber85] D. Bernstein and D. Hyland, "The optimal projection equations for reduced-order state estimation," *IEEE Transactions on Automatic Control,* **AC-30**(6), pp. 583-585 (June 1985).

[Ber89] D. Bernstein and W. Haddad, "Steady-state Kalman filtering with an H_∞ error bound," *Systems and Control Letters,* **12**, pp. 9-16 (1989).

[Ber05] D. Bernstein, *Matrix Mathematics,* Princeton University Press, Princeton, New Jersey, 2005.

[Bet94] M. Bettayeb and D. Kavranoglu, "Reduced order H_∞ filtering," *American Control Conference,* pp. 1884-1888 (June 1994).

[Bie73] G. Bierman, "A comparison of discrete linear filtering algorithms," *IEEE Transactions on Aerospace and Electronic Systems,* **AES-9**(1), pp. 28-37 (January 1973).

[Bie76] G. Bierman, "Measurement updating using the *U-D* factorization," *Automatica,* **12**(4), pp. 375-382 (July 1976).

[Bie77a] G. Bierman and C. Thornton, "Numerical comparison of Kalman filter algorithms: Orbit determination case study," *Automatica,* **13**(1), pp. 23-35 (January 1977).

[Bie77b] G. Bierman, *Factorization Methods for Discrete Sequential Estimation,* Academic Press, San Diego, California, 1977.

[Bil95] P. Billingsley, *Probability and Measure,* Wiley-Interscience, New York, 1995.

[Bit85] R. Bitmead, M. Gevers, I. Petersen, and R. Kaye, "Monotonicity and stabilizability properties of solutions of the Riccati difference equation: Propositions, lemmas, theorems, fallacious conjectures and counterexamples," *Systems and Control Letters,* **5**(5), pp. 309-315 (April 1985).

[Bit91] S. Bittanti, A. Laub, and J. Willems (eds.), *The Riccati Equation,* Springer-Verlag, New York, 1991.

[Bjo67] A. Bjorck, "Solving linear least squares problems by Gram-Schmidt orthogonalization," *BIT,* **7**, pp. 1-21 (1967).

[Bla99] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems,* Artech House, Dedham, Massachusetts, 1999.

[Boo52] R. Booton, "An optimization theory for time-varying linear systems with nonstationary statistical inputs," *Proceedings of the IRE,* **40**, pp. 977-981 (August 1952).

[Bor79] P. Borjesson and C. Sundberg, "Simple approximations of the error function $Q(x)$ for communications applications," *IEEE Transactions on Communications,* **COM-27**(3), pp. 639-643 (March 1979).

[Boz94] S. Bozic, *Digital and Kalman Filtering,* Halsted Press, New York, 1994.

[Bra89] K. Brammer and G. Siffling, *Kalman-Bucy Filters,* Artech House, Dedham, Massachusetts, 1989.

[Bro91] P. Brockwell and R. Davis, *Time Series: Theory and Methods,* Springer-Verlag, New York, 1991.

[Bro98] E. Brookner, *Tracking and Kalman Filtering Made Easy,* John Wiley & Sons, New York, 1998.

[Bro96] R. Brown and P. Hwang, *Introduction to Random Signals and Applied Kalman Filtering,* John Wiley & Sons, New York, 1996.

[Bry65] A. Bryson and D. Johanson, "Linear filtering for time-varying systems using measurements containing colored noise," *IEEE Transactions on Automatic Control,* **AC-10**(1), pp. 4-10 (January 1965).

[Bry68] A. Bryson and L. Henrikson, "Estimation using sampled data containing sequentially correlated noise," *Journal of Spacecraft and Rockets,* **5**(6), pp. 662-665 (June 1968).

[Buc65] R. Bucy, "Nonlinear filtering," *IEEE Transactions on Automatic Control,* **10**(1), p. 198 (January 1965).

[Buc67] R. Bucy, "Global theory of the Riccati equation," *Journal of Computer and System Sciences,* **1**, pp. 349-361 (December 1967).

[Buc68] R. Bucy and P. Joseph, *Filtering for Stochastic Processes with Applications to Guidance,* John Wiley & Sons, New York, 1968.

[Buc69] R. Bucy, "Bayes theorem and digital realizations for non-linear filters," *The Journal of the Astronautical Sciences,* **17**(2), pp. 80-94 (September-October 1969).

[Buc71] R. Bucy and K. Senne, "Digital synthesis of non-linear filters," *Automatica,* **7**(3), pp. 287-298 (May 1971).

[Bur93] J. Burl, "A reduced order extended Kalman filter for sequential images containing a moving object," *IEEE Transactions on Image Processing,* **2**(3), pp. 285-295 (July 1993).

[Bur99] J. Burl, *Linear Optimal Control,* Addison-Wesley, Reading, Massachusetts, 1999.

[Cai88] P. Caines, *Linear Stochastic Systems,* John Wiley & Sons, New York, 1988.

[Cam68] A. Cameron, "Control and estimation of linear systems with nongaussian *a priori* distributions," *Sixth Annual Allerton Conference on Circuit and System Theory* (1968).

[Cam01] M. Campbell and S. Brunke, "Nonlinear estimation of aircraft models for online control customization," *IEEE Aerospace Conference,* pp. 621-628 (2001).

[Car73] N. Carlson, "Fast triangular formulation of the square root filter," *AIAA Journal,* **11**(9), pp. 1259-1265 (September 1973).

[Car99] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *IEE Proceedings Part F: Radar and Sonar Navigation,* **146**, pp. 2-7 (February 1999).

[Cat89] D. Catlin, *Estimation, Control, and the Discrete Kalman Filter,* Springer-Verlag, New York, 1989.

[Cha47] S. Chandrasekhar, "On the radiative equilibrium of a stellar atmosphere, Part XXI," *The Astrophysical Journal,* **106**, pp. 152-216 (1947).

[Cha48] S. Chandrasekhar, "On the radiative equilibrium of a stellar atmosphere, Part XXII," *The Astrophysical Journal,* **107**, pp. 48-72 (1948).

[Cha96] S. Charleston and M. Azimi-Sadjadi, "Reduced order Kalman filtering for the enhancement of respiratory sounds," *IEEE Transactions on Biomedical Engineering,* **43**(4), pp. 421-424 (April 1996).

[Che99] C. Chen, *Linear System Theory and Design,* Oxford University Press, 1999.

[Che93] G. Chen, *Approximate Kalman Filtering,* World Scientific, Singapore, 1993.

[Che85] H. Chen, *Recursive Estimation and Control for Stochastic Systems,* John Wiley & Sons, New York, 1985.

[Chi85] T. Chia, *Parameter Identification and State Estimation of Constrained Systems,* Doctoral Dissertation, Case Western Reserve University, Cleveland, Ohio, 1985.

[Chu87] C. Chui and G. Chen, *Kalman Filtering with Real-Time Applications,* Springer-Verlag, New York, 1987.

[Coh96] S. Cohn and R. Todling, "Approximate data assimilation scheme for stable and unstable dynamics," *Journal of the Meteorological Society of Japan,* **74**(1), pp. 63-75 (1996).

[Cox64] H. Cox, "On the estimation of state variables and parameters for noisy dynamic systems," *IEEE Transactions on Automatic Control,* **AC-9**(1), pp. 5-12 (January 1964).

[Cox73] H. Coxeter, *Regular Polytopes,* Dover, New York, 1973.

[Cra04] J. Crassidis and J. Junkins, *Optimal Estimation of Dynamic Systems,* CRC Press, New York, 2004.

[Cri02] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Transactions on Signal Processing,* **50**(3), pp. 736-746 (March 2002).

[Dar84] J. Darragh and D. Looze, "Noncausal minimax linear state estimation for systems with uncertain second-order statistics," *IEEE Transactions on Automatic Control,* **AC-29**(6), pp. 555-557 (June 1984).

[Dav77] M. Davis, *Linear Estimation and Stochastic Control,* Chapman & Hall, London, 1977.

[deF71] R. de Figueiredo and Y. Jan, "Spline filters," *Symposium on Nonlinear Estimation Theory and Its Applications,* pp. 127-138 (1971).

[deF74] B. de Finetti, *Theory of Probability: A Critical Introductory Treatment,* John Wiley & Sons, New York, 1974.

[deF02] N. de Freitas, "Rao-Blackwellised particle filtering for fault diagnosis," *IEEE Aerospace Conference,* pp. 1767-1772 (2002).

[Deu65] R. Deutsch, *Estimation Theory,* Prentice-Hall, Englewood Cliffs, New Jersey, 1965.

[Dev01] L. Devroye and G. Lugosi, *Combinatorial Methods in Density Estimation,* Springer-Verlag, New York, 2001.

[Dew03] P. Dewallef and O. Leonard, "On-line performance monitoring and engine diagnostic using robust Kalman filtering techniques," *ASME Turbo Expo,* Paper GT2003-38379 (2003).

[Dew04] P. Dewallef, O. Leonard, and K. Mathioudakis, "On-line aircraft engine diagnostic using a soft-constrained Kalman filter," *ASME Turbo Expo,* Paper GT2004-53539 (2004).

[Dez92] F. Deza, E. Busvelle, J. Gauthier, and D. Rakotopara, "High gain estimation for nonlinear systems," *Systems and Control Letters,* **18**, pp. 295-299 (1992).

[Did85] G. Diderrich, "The Kalman filter from the perspective of the Goldberger-Thiel estimators," *The American Statistician,* **39**(3), pp. 193-198 (August 1985).

[Did95] G. Didinsky, Z. Pan, and T. Basar, "Parameter identification for uncertain plants using H^∞ methods," *Automatica,* **31**(9), pp. 1227-1250 (September 1995).

[Dou01] A. Doucet, N. de Freitas, and N. Gordon (eds.), *Sequential Monte Carlo Methods in Practice,* Springer-Verlag, New York, 2001.

[Dun44] W. Duncan, "Some devices for the solution of large sets of simultaneous linear equations (with an appendix on the reciprocation of partitioned matrices)," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, Seventh Series,* **35**, pp. 660-670 (1944).

[Dye69] P. Dyer and S. McReynolds, "Extension of square-root filtering to include process noise," *Journal of Optimization Theory and Applications,* **3**(6), pp. 444-459 (1969).

[Fad59] V. Fadeeva, *Computational Methods of Linear Algebra,* Dover, New York, 1959.

[Fey63] R. Feynman, R. Leighton, and M. Sands, *The Feynman Lectures on Physics: Volume 1,* Addison-Wesley, Reading, Massachusetts, 1963.

[Fit71] R. Fitzgerald, "Divergence of the Kalman filter," *IEEE Transactions on Automatic Control,* **AC-16**, pp. 736-747 (December 1971).

[Fle81] R. Fletcher, *Practical Methods of Optimization – Volume 2: Constrained Optimization,* John Wiley & Sons, New York, 1981.

[Fom00] S. Fomin and I. Gelfand, *Calculus of Variations,* Dover, New York, 2000.

[Fra69] D. Fraser and J. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Transactions on Automatic Control,* **14**(4), pp. 387-390 (August 1969).

[Fri66] B. Friedland and I. Bernstein, "Estimation of the state of a nonlinear process in the presence of nongaussian noise and disturbances," *Journal of the Franklin Institute,* **281**(6), pp. 455-480 (June 1966).

[Fri89] B. Friedland, "On the properties of reduced-order Kalman filters," *IEEE Transactions on Automatic Control,* **34**(3), pp. 321-324 (March 1989).

[Gau04] C. Gauss, *Theory of Motion of the Heavenly Bodies Moving About the Sun in Conic Sections: A Translation of Theoria Motus*, Dover, New York, 2004.

[Gee97] J. De Geeter, H. Van Brussel, and J. De Schutter, "A smoothly constrained Kalman filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **19**(10), pp. 1171-1177 (October 1997).

[Gel74] A. Gelb (ed.), *Applied Optimal Estimation,* MIT Press, Cambridge, Massachusetts, 1974.

[Gil01] W. Gilks and C. Berzuini, "Following a moving target – Monte Carlo inference for dynamic Bayesian models," *Journal of the Royal Statistical Society, B,* **63**, pp. 127-146 (2001).

[Gil81] P. Gill, W. Murray, and M. Wright, *Practical Optimization,* Academic Press, New York, 1981.

[Gli94] L. Glielmo, P. Marino, R. Setola, and F. Vasca, "Reduced Kalman filtering for indirect adaptive control of the induction motor," *International Journal of Adaptive Control and Signal Processing*, **8**, pp. 527-541 (1994).

[Gol65] G. Golub, "Numerical methods for solving linear least squares problems," *Numerische Mathematik,* **7**, pp. 206-216 (1965).

[Gol89] G. Golub and C. Van Loan, *Matrix Computations,* The Johns Hopkins University Press, Baltimore, Maryland, 1989.

[Goo84] G. Goodwin and K. Sin, *Adaptive Filtering, Prediction and Control,* Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[Goo05a] G. Goodwin, M. Seron, and J. De Dona, *Constrained control and estimation*, Springer-Verlag, London, 2005.

[Goo05b] G. Goodwin, J. De Dona, M. Seron, and X. Zhuo, "Lagrangian duality between constrained estimation and control," *Automatica,* **41**(6), pp. 935-944 (June 2005).

[Gor93] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEEE Proceedings-F,* **140**(2), pp. 107-113 (April 1993).

[Gra93] J. Gray and W. Murray, "A derivation of an analytic expression for the tracking index for the alpha-beta-gamma filter," *IEEE Transactions on Aerospace and Electronic Systems*, **29**(3), pp. 1064-1065 (July 1993).

[Gre01] M. Grewal and A. Andrews, *Kalman Filtering,* John Wiley & Sons, New York, 2001.

[Gri97] K. Grigoriadis and J. Watson, "Reduced-order H_∞ and $L_2 - L_\infty$ filtering via linear matrix inequalities," *IEEE Transactions on Aerospace and Electronic Systems,* **33**(4), pp. 1326-1338 (October 1997).

[Gri88] M. Grimble, "H_∞ design of optimal linear filters," in *Linear Circuits, Systems and Signal Processing: Theory and Applications,* C. Byrnes, C. Martin, and R. Saeks (Eds.) North-Holland, Amsterdam, The Netherlands, 1988, pp. 533540.

[Gri90] M. Grimble and A. El Sayed, "Solution of the H_∞ optimal linear filtering problem for discrete-time systems," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* **38**(7), pp. 1092-1104 (July 1990).

[Gri91a] M. Grimble, "H_∞ fixed lag smoothing filter for scalar systems," *IEEE Transactions on Signal Processing,* **39**(9), pp. 1955-1963 (September 1991).

[Gri91b] M. Grimble and M. Johnson, "H_∞ robust control design – A tutorial review," *Computing and Control Engineering Journal,* **2**(6), pp. 275-282 (November 1991).

[Hab03] S. Habibi and R. Burton, "The variable structure filter," *ASME Journal of Dynamic Systems, Measurement, and Control,* **125**(3), pp. 287-293 (September 2003).

[Had91] W. Haddad, D. Bernstein, and D. Mustafa, "Mixed-norm H_2/H_∞ regulation and estimation: The discrete-time case," *Systems and Control Letters,* **16**, pp. 235-247 (1991).

[Hae98] D. Haessig and B. Friedland, "Separate-bias estimation with reduced-order Kalman filters," *IEEE Transactions on Automatic Control,* **43**(7), pp. 983-987. (July 1998).

[Hal81] A. Hald, "T. N. Thiele's contributions to statistics," *International Statistics Review,* **49**, pp. 1-20 (1981).

[Har89] A. Harvey, *Forecasting, Structural Time Series Models and the Kalman Filter,* Cambridge University Press, New York, 1989.

[Has95] B. Hassibi and T. Kailath, "H^∞ adaptive filtering," *IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 949-952 (1995).

[Has96a] B. Hassibi, A. Sayed, and T. Kailath, "Linear estimation in Krein spaces – Part I: Theory," *IEEE Transactions on Automatic Control,* **41**(1), pp. 18-33 (January 1996).

[Has96b] B. Hassibi, A. Sayed, and T. Kailath, "Linear estimation in Krein spaces – Part II: Applications," *IEEE Transactions on Automatic Control,* **41**(1), pp. 34-49 (January 1996).

[Has99] B. Hassibi, A. Sayed, T. Kailath, *Indefinite-Quadratic Estimation and Control,* SIAM, Philadelphia, Pennsylvania, 1999.

[Hay98] S. Hayward, "Constrained Kalman filter for least-squares estimation of time-varying beamforming weights," in *Mathematics in Signal Processing IV,* J. McWhirter and I. Proudler (Eds.), Oxford University Press, pp. 113-125, 1998.

[Hec71] C. Hecht, "Digital realization of non-linear filters," *Symposium on Nonlinear Estimation Theory and Its Applications,* pp. 152-158 (1971).

[Hec73] C. Hecht, "Systems identification using Bayesian estimation," *Symposium on Nonlinear Estimation Theory and Its Applications,* pp. 107-113 (1973).

[Hef66] H. Heffes, "The effect of erroneous models on the Kalman filter response," *IEEE Transactions on Automatic Control,* **AC-11**, pp. 541-543 (July 1966).

[Hel94] Y. Hel-Or and M. Werman, "Recognition and localization of articulated objects," *IEEE Workshop on Motion of Non-Rigid and Articulated Objects,* pp. 116-123 (November 1994).

[Hen81] H. Henderson and S. Searle, "On deriving the inverse of a sum of matrices," *SIAM Review,* **23**(1), pp. 53-60 (January 1981).

[Hen82] R. Henriksen, "The truncated second-order nonlinear filter revisited," *IEEE Transactions on Automatic Control,* **AC-27**(1), pp. 247-251 (February 1982).

[Hid03] C. Hide, T. Moore, and M. Smith, "Adaptive Kalman filtering for low-cost INS/GPS," *The Journal of Navigation,* **56**, pp. 143-152 (2003).

[Ho64] Y. Ho and R. Lee, "A Bayesian approach to problems in stochastic estimation and control," *IEEE Transactions on Automatic Control,* **AC-9**, pp. 333-339 (1964).

[Ho02] Y. Ho and D. Pepyne, "Simple explanation of the no-free-lunch theorem and its implications," *Journal of Optimization Theory and Applications,* **155**(3). 549-570. (December 2002).

[Hor85] R. Horn and C. Johnson, *Matrix Analysis,* Cambridge University Press, New York, 1985.

[Hou53] A. Householder, *Principles of Numerical Analysis,* McGraw-Hill, New York, 1953.

[Hou64] A. Householder, *The Theory of Matrices in Numerical Analysis,* Blaisdell, New York, 1964.

[Hsi96] F. Hsiao and S. Pan, "Robust Kalman filter synthesis for uncertain multiple time-delay stochastic systems," *Journal of Dynamic Systems, Measurement, and Control,* **118**, pp. 803-808 (December 1996).

[Hug88] R. Hughes, *Analog Automatic Control Loops in Radar and EW,* Artech House, Dedham, Massachusetts, 1988.

[Hun03] Y. Hung and F. Yang, "Robust H_∞ filtering with error variance constraints for discrete time-varying systems with uncertainty," *Automatica,* **39**(7), pp. 1185-1194 (July 2003).

[Hut84] C. Hutchinson, "The Kalman filter applied to aerospace and electronic systems," *IEEE Transactions on Aerospace and Electronic Systems,* **AES-20**(4), pp. 500-504 (July 1984).

[Iba01] Y. Iba, "Population Monte Carlo algorithms," *Transactions of the Japanese Society for Artificial Intelligence,* **16**(2), pp. 279-286. (2001).

[Ion99] V. Ionescu, C. Oara, and M. Weiss, *Generalized Riccati Theory and Robust Control,* John Wiley & Sons, New York, 1999.

[Isa96] M. Isaard and A. Blake, "Contour tracking by stochastic propagation of conditional density," *European Conference on Computer Vision,* Cambridge, pp. 343-356, 1996.

[Isa98a] M. Isard and A. Blake, "Condensation – Conditional density propagation for visual tracking," *International Journal of Computer Vision,* **29**(1), pp. 5-28. (1998).

[Isa98b] M. Isard and A. Blake, "ICondensation: Unifying low-level and high-level tracking in a stochastic framework," *Lecture Notes in Computer Science,* **1406**, pp. 893-908. (1998).

[Jaz66] A. Jazwinski, "Filtering for nonlinear dynamical systems," *IEEE Transactions on Automatic Control,* **AC-11**(5), pp. 765-766 (October 1966).

[Jaz69] A. Jazwinski, "Adaptive filtering," *Automatica,* **5**(4), pp. 475-485 (July 1969).

[Jaz70] A. Jazwinski, *Stochastic Processes and Filtering Theory,* Academic Press, New York, 1970.

[Joh69] G. Johnson, "A deterministic theory of estimation and control," *IEEE Transactions on Automatic Control,* **14**(4), pp. 380-384 (August 1969).

[Jor68] T. Jordan, "Experiments on error growth associated with some linear least-squares procedures," *Mathematics of Computation,* **22**, pp. 579-588 (1968).

[Jua93] J. Juang, *Applied System Identification,* Prentice-Hall, Upper Saddle River, New Jersey, 1993.

[Jul95] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," *American Control Conference,* pp. 1628-1632 (1995).

[Jul00] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control,* **45**(3), pp. 477-482 (March 2000).

[Jul02a] S. Julier and J. Uhlmann, "Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations," *American Control Conference,* pp. 887-892, 2002.

[Jul03] S. Julier, "The spherical simplex unscented transformation," *American Control Conference,* pp. 2430-2434, 2003.

[Jul04] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE,* **92**(3), pp. 401-422 (March 2004).

[Kai68] T. Kailath, "An innovations approach to least-squares estimation – Part I: Linear filtering in additive white noise," *IEEE Transactions on Automatic Control,* **AC-13**(6), pp. 646-655 (December 1968).

[Kai73] T. Kailath, "Some new algorithms for recursive estimation in constant linear systems," *IEEE Transactions on Information Theory,* **IT-19**(6), pp. 750-760 (November 1973).

[Kai74] T. Kailath, "A view of three decades of linear filtering theory," *IEEE Transactions on Information Theory,* **IT-20**(2), pp. 146-181 (March 1974).

[Kai77] T. Kailath (ed.), *Linear Least Squares Estimation,* Dowden, Hutchinson & Ross, Stroudsburg, Pennsylvania, 1977.

[Kai80] T. Kailath, *Linear Systems,* Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

[Kai81] T. Kailath, *Lectures on Wiener and Kalman Filtering,* Springer-Verlag, New York, 1981.

[Kai00] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation,* Prentice-Hall, Upper Saddle River, New Jersey, 2000.

[Kal84] P. Kalata, "The tracking index: A generalized parameter for α-β and α-β-γ target trackers," *IEEE Transactions on Aerospace and Electronic Systems,* **AES-20**(2), pp. 174-182.

[Kal60] R. Kalman, "A new approach to linear filtering and prediction problems," *ASME Journal of Basic Engineering,* **82**, pp. 35-45 (March 1960).

[Kal60b] R. Kalman, "Contributions to the theory of optimal control," *Boletin de la Sociedad Matematica Mexicana,* **5**, pp. 102-119 (1960).

[Kal60c] R. Kalman, "On the general theory of control systems," *Proceedings of the 1st IFAC Congress,* pp. 481492, Butterworth's, London, 1960.

[Kal61] R. Kalman and R. Bucy, "New results in linear filtering and prediction theory," *ASME Journal of Basic Engineering,* **83**, pp. 95-108 (March 1961).

[Kal63] R. Kalman, Y. Ho, and K. Narendra, "Controllability of linear dynamical systems," *Contributions to Differential Equations,* **1**(2), pp. 189-213 (1963).

[Kam71] P. Kaminski, A. Bryson, and S. Schmidt, "Discrete square root filtering: A survey of current techniques," *IEEE Transactions on Automatic Control,* **AC-16**, pp. 727-736 (December 1971).

[Kan95] K. Kanazawa, D. Koller, and S. Russell, "Stochastic simulation algorithms for dynamics probabilistic networks," *Eleventh Annual Conference on Uncertainty in AI,* pp. 346-351, 1995.

[Kas96] E. Kaskurewicz and A. Bhaya, "Discrete-time state estimation with two counters and measurement delay," *IEEE Conference on Decision and Control,* pp. 1472-1476, 1996.

[Kel98] J. Keller and M. Darouach, "Reduced-order Kalman filter with unknown inputs," *Automatica,* **34**(11), pp. 1463-1468 (November 1998).

[Kel99] J. Keller, "A new strategy for designing a reduced-order Kalman filter," *International Journal of Systems Science,* **30**(11), pp. 1161-1166 (1999).

[Ken61] M. Kendall, "Daniel Bernoulli on maximum likelihood," *Biometrika,* **48**, pp. 1-18 (1961).

[Kit87] G. Kitagawa, "Non-Gaussian state-space modelling of non-stationary time series (with discussion)," *Journal of the American Statistical Association,* **82**(400), pp. 1032-1063 (December 1987).

[Kit89] G. Kitagawa, "Non-Gaussian seasonal adjustment," *Computers and Mathematics with Applications,* **18**(6,7), pp. 503-514 (1989).

[Kit94] G. Kitagawa, "The two-filter formula for smoothing and an implementation of the Gaussian-sum smoother," *Annals of the Institute of Statistical Mathematics,* **46**(4), pp. 605-623 (1994).

[Kit96a] G. Kitagawa, *Smoothness Priors Analysis of Time Series,* Springer-Verlag, New York, 1996.

[Kit96] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics,* **5**(1), pp. 1-25 (1996).

[Ko06] S. Ko and R. Bitmead, "State estimation for linear systems with state equality constraints," *Automatica,* 2006.

[Kob03] T. Kobayashi and D.L. Simon, "Application of a bank of Kalman filters for aircraft engine fault diagnosis," *ASME Turbo Expo,* Atlanta, paper GT2003-38550, June 2003.

[Kol41] A. Kolmogorov, "Interpolation and extrapolation of stationary random sequences," in *Selected Works of A. N. Kolmogorov, Volume II,* V. Tikhomirov (Ed.), Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 272-280 (1992).

[Kop63] R. Kopp and R. Orford, "Linear regression applied to system identification for adaptive control systems," *AIAA Journal,* **1**, pp. 2300-2306 (October 1963).

[Kos04] S. Kosanam and D. Simon, "Kalman filtering with uncertain noise covariances," *Intelligent Systems and Control,* Honolulu, Hawaii, pp. 375-379, August 2004.

[Kra88] S. Kramer and H. Sorenson, "Recursive Bayesian estimation using piece-wise constant approximations," *Automatica,* **24**(6), pp. 789-801 (November 1988).

[Kri84] V. Krishnan, *Nonlinear Filtering and Smoothing: An Introduction to Martingales, Stochastic Integrals and Estimation,* John Wiley & Sons, New York, 1984.

[Kus67a] H. Kushner, "Dynamical equations for optimal nonlinear filtering," *Journal of Differential Equations,* **3**, pp. 179-190 (1967).

[Kus67] H. Kushner, "Approximations to optimal nonlinear filters," *IEEE Transactions on Automatic Control,* **AC-12**(5), pp. 546-556 (October 1967).

[Kwa72] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems,* John Wiley & Sons, New York, 1972.

[Kwo04] C. Kwok, D. Fox, and M. Meila, "Real-time particle filters," *Proceedings of the IEEE,* **92**(3), pp. 469-484 (March 2004).

[Kwo03] S. Kwon and W. Chung, "Combined synthesis of state estimator and perturbation observer," *ASME Journal of Dynamic Systems, Measurement, and Control,* **125**(1), pp. 19-26 (March 2003).

[Lai74] D. Lainiotis, "Estimation: A brief survey," *Information Sciences,* **7**, pp. 191-202 (1974).

[Lan95] P. Lancaster and L. Rodman, *Algebraic Riccati Equations,* Oxford University Press, New York, 1995.

[Lar98] T. Larsen, N. Andersen, O. Ravn, and N. Poulsen, "Incorporation of time delayed measurements in a discrete-time Kalman filter," *IEEE Conference on Decision and Control,* pp. 3972-3977, 1998.

[Lau81] S. Lauritzen, "Time series analysis in 1880: A discussion of contributions made by T. N. Thiele," *International Statistics Review,* **49**, pp. 319-331 (1981).

[Lee64] R. Lee, *Optimal Estimation, Identification, and Control,* The MIT Press, Cambridge, Massachusetts, 1964.

[Lew51] K. Lewin, *Field Theory in Social Science; Selected Theoretical Papers,* Harper & Row, New York, 1951.

[Lew86a] F. Lewis, *Optimal Control,* John Wiley & Sons, New York, 1986.

[Lew86b] F. Lewis, *Optimal Estimation,* John Wiley & Sons, New York, 1986.

[Lew97] H. Lewis, *Foundations of Fuzzy Control,* Plenum, New York, 1997.

[Lie67] P. Liebelt, *An Introduction to Optimal Estimation,* Addison-Wesley, Reading, Massachusetts, 1967.

[Liu80] C. Liu and S. Marcus, "Estimator performance for a class of nonlinear estimation problems," *IEEE Transactions on Automatic Control,* **AC-25**(2), pp. 299-302 (April 1980).

[Liu98] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamical systems," *Journal of the American Statistical Association,* **93**, pp. 1032-1044 (1998).

[Lju98] L. Ljung, *System Identification: Theory for the User,* Prentice-Hall, Upper Saddle River, New Jersey, 1998.

[Lue84] D. Luenberger, *Linear and Nonlinear Programming,* Addison-Wesley, Reading, Massachusetts, 1984.

[Mac99] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," *International Conference on Computer Vision,* pp. 572-578, 1999.

[Mah04a] K. Mahata and T. Soderstrom, "Improved estimation performance using known linear constraints," *Automatica,* **40**(8), pp. 1307-1318 (August 2004).

[Mah04b] M. Mahmoud, "Resilient linear filtering of uncertain systems," *Automatica,* **40**(10), pp. 1797-1802 (October 2004).

[Mai84] J. Maindonald, *Statistical Computation,* John Wiley & Sons, New York, 1984.

[Mal01] M. Mallick, S. Coraluppi, and C. Carthel, "Advances in asynchronous and decentralized estimation," *IEEE Aerospace Conference,* pp. 1873-1888, 2001.

[Mar05] H. Marquez and M. Riaz, "Robust state observer design with application to an industrial boiler system," *Control Engineering Practice,* **13**(6), pp. 713-728 (June 2005).

[Mas75] C. Masreliez, "Approximate non-Gaussian filtering with linear state and observation relations," *IEEE Transactions on Automatic Control,* **AC-20**, pp. 107-110 (February 1975).

[Mas77] C. Masrielez and R. Martin, "Robust Bayesian estimation for the linear model and robustifying the Kalman filter," *IEEE Transactions on Automatic Control,* **AC-22**(3), pp. 361-371 (June 1977).

[May79] P. Maybeck, *Stochastic Models, Estimation, and Control - Volume 1,* Academic Press, New York, 1979.

[May82] P. Maybeck, *Stochastic Models, Estimation, and Control - Volume 2,* Academic Press, New York, 1982.

[May84] P. Maybeck, *Stochastic Models, Estimation, and Control - Volume 3,* Academic Press, New York, 1984.

[McG74] T. McGarty, *Stochastic Systems and State Estimation,* John Wiley & Sons, New York, 1974.

[McG85] L. McGee and S. Schmidt, "Discovery of the Kalman filter as a practical tool for aerospace and industry," NASA Technical Memo 86847, November 1985.

[Mcr75] S. McReynolds, "Multidimensional Hermite-Gaussian quadrature formulae and their application to nonlinear estimation," *Symposium on Nonlinear Estimation Theory and Its Applications,* pp. 188-191 (1975).

[Mec49] E. Mechler, J. Russell, and M. Preston, "The basis for the optimum aided-tracking time constant," *Journal of the Franklin Institute,* **248**, pp. 327-334 (October 1949).

[Med69] J. Meditch, *Stochastic Optimal Linear Estimation and Control,* McGraw-Hill, New York, 1969.

[Med73] J. Meditch, "A survey of data smoothing for linear and nonlinear dynamic systems," *Automatica,* **9**(2), pp. 151-162 (March 1973).

[Meh70] R. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Transactions on Automatic Control,* **AC-15**(2), pp. 175-184 (April 1970).

[Meh72] R. Mehra, "Approaches to adaptive filtering," *IEEE Transactions on Automatic Control,* **AC-17**(5), pp. 693-698 (October 1972).

[Men87] J. Mendel, *Lessons in Digital Estimation Theory,* Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

[Met49] N. Metropolis and S. Ulam, "The Monte Carlo method," *Journal of the American Statistical Association,* **44**(247), pp. 335-341 (September 1949).

[Min93] G. Minkler and J. Minkler, *Theory and Application of Kalman Filtering,* Magellan Book Company, Palm Bay, Florida, 1993.

[Mir90] L. Mirsky, *An Introduction to Linear Algebra,* Dover, New York, 1990.

[Mol03] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM Review,* **45**(1), pp. 3-49 (January 2003).

[Moo00] T. Moon and W. Stirling, *Mathematical Methods and Algorithms for Signal Processing,* Prentice-Hall, Upper Saddle River, New Jersey, 2000.

[Mor98] P. Moral, "Measure valued processes and interacting particle systems: Application to non-linear filtering problems," *Annals of Applied Probability,* **8**(2), pp. 438-495, 1998.

[Mor03] R. Morales-Menendez, N. de Freitas, and D. Poole, "Estimation and control of industrial processes with particle filters," *American Control Conference,* pp. 579-584, 2003.

[Mor78] M. Morf, R. Levy, and T. Kailath, "Square-root algorithms for the continuous-time linear least-square estimation problem," *IEEE Transactions on Automatic Control,* **AC-23**(5), pp. 907-911 (October 1978).

[Mul91] P. Muller, "Monte Carlo integration in general dynamic models," *Contemporary Mathematics,* **115**, pp. 145-163 (1991).

[Mye76] K. Myers and B. Tapley, "Adaptive sequential estimation with unknown noise statistics," *IEEE Transactions on Automatic Control,* **AC-21**(4), pp. 520-523 (August 1976).

[Nag87] K. Nagpal, R. Helmick, and C. Sims, "Reduced-order estimation: Part I. Filtering," *International Journal of Control,* **45**, pp. 1867-1888 (1987).

[Nag91] K. Nagpal and P. Khargonekar, "Filtering and smoothing in an H^∞ setting," *IEEE Transactions on Automatic Control,* **36**(2), pp. 152-166 (February 1991).

[Nah69] N. Nahi, "Optimal recursive estimation with uncertain observation," *IEEE Transactions on Information Theory,* **IT-15**(4), pp. 457-462 (July 1969).

[Nel87] E. Nelson, *Radically Elementary Probability Theory,* Princeton University Press, Princeton, New Jersey, 1987.

[Net78] M. Netto, L. Gimeno, and M. Mendes, "On the optimal and suboptimal nonlinear filtering problem for discrete-time systems," *IEEE Transactions on Automatic Control,* **AC-23**(6), pp. 1062-1067 (December 1978).

[Nis66] T. Nishimura, "On the a priori information in sequential estimation problems," *IEEE Transactions on Automatic Control,* **AC-11**, pp. 197-204 (April 1966).

[Nor00] M. Norgaard, N. Poulsen, and O. Ravn, "New developments in state estimation for nonlinear systems," *Automatica,* **36**(11), pp. 1627-1638 (November 2000).

[OCo96] J. O'Connor and E. Robertson, *Matrices and Determinants,* http://www-history.mcs.st-andrews.ac.uk/HistTopics/Matrices_and_determinants.html (1996).

[Ott85] P. Otter, *Dynamic Feature Space Modelling, Filtering and Self-Tuning Control of Stochastic Systems,* Springer-Verlag, New York, 1985.

[Pan96] Z. Pan and T. Basar, "Parameter identification for uncertain linear systems with partial state measurements under an H^∞ criterion," *IEEE Transactions on Automatic Control,* **41**(9), pp. 1295-1311 (September 1996).

[Pap02] A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Processes,* McGraw-Hill, New York, 2002.

[Pat98] A. Patti, A. Tekalp, and M. Sezan, "A new motion-compensated reduced-order model Kalman filter for space-varying restoration of progressive and interlaced video," *IEEE Transactions on Image Processing,* **7**(4), pp. 543-554 (April 1998).

[Pee01] P. Peebles, *Probability, Random Variables, and Random Signal Principles,* McGraw-Hill, New York, 2001.

[Pha98] D. Pham, J. Verron, and M. Roubaud, "A singular evolutive extended Kalman filter for data assimilation in oceanography," *Journal of Marine Systems,* **16**, pp. 323-240 (1998).

[Pit99] M. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association,* **94**(446), pp. 590-599 (1999).

[Pla50] R. Plackett, "Some theorems in least squares," *Biometrika,* **37**, pp. 149-157 (1950).

[Poo81] V. Poor and D. Looze, "Minimax state estimation for linear stochastic systems with noise uncertainty," *IEEE Transactions on Automatic Control,* **AC-26**(4), pp. 902-906 (August 1981).

[Por88] J. Porrill, "Optimal combination and constraints for geometrical sensor data," *International Journal of Robotics Research,* **7**(6), pp. 66-77 (December 1988).

[Pou86] M. Poubelle, I. Petersen, M. Gevers, and R. Bitmead, "A miscellany of results on an equation of Count J. F. Riccati," *IEEE Transactions on Automatic Control,* **AC-31**(7), pp. 651-654 (July 1986).

[Pre92] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C : The Art of Scientific Computing,* Cambridge University Press, New York, 1992.

[Rao03] C. Rao, J. Rawlings, and D. Mayne, "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations," *IEEE Transactions on Automatic Control,* **48**(2), pp. 246-258 (February 2003).

[Rau65] H. Rauch, F. Tung, and C. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal,* **3**(8), pp. 1445-1450 (August 1965).

[Rei72] W. Reid, *Riccati Differential Equations,* Academic Press, New York, 1972.

[Rei98] K. Reif, F. Sonnemann, and R. Unbehauen, "An EKF-based nonlinear observer with a prescribed degree of stability," *Automatica,* **34**(9), pp. 1119-1123 (September 1998).

[Rei99] K. Reif, F. Sonnemann, and R. Unbehauen, "Nonlinear state observation using H_∞-filtering Riccati design," *IEEE Transactions on Automatic Control,* **44**(1), pp. 203-208 (January 1999).

[Rhe89] I. Rhee and J. Speyer, "A game theoretic controller and its relationship to H_∞ and linear-exponential-Gaussian synthesis," *IEEE Conference on Decision and Control,* pp. 909-915 (1989).

[Rho71] I. Rhodes, "A tutorial introduction to estimation and filtering," *IEEE Transactions on Automatic Control,* **AC-16**(6), pp. 688-706 (December 1971).

[Rip87] B. Ripley, *Stochastic Simulation,* John Wiley & Sons, New York, 1987.

[Ris04] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications,* Artech House, Norwell, Massachusetts, 2004.

[Rob99] C. Robert and G. Casella, *Monte Carlo Statistical Methods,* Springer-Verlag, New York, 1999.

[Ruy85] P. Ruymgaart and T. Soong, *Mathematics of Kalman-Bucy Filtering,* Springer-Verlag, New York, 1985.

[Saf78] M. Safonov and M. Athans, "Robustness and computational aspects of nonlinear stochastic estimators and regulators," *IEEE Transactions on Automatic Control,* **AC-23**(4), pp. 717-725 (August 1978).

[Sag71] A. Sage and J. Melsa, *Estimation Theory with Applications to Communications and Control,* McGraw-Hill, New York, 1971.

[Sch67] F. Schlee, C. Standish, and N. Toda, "Divergence in the Kalman filter," *AIAA Journal,* **5**, pp. 1114-1120 (June 1967).

[Sch66] S. Schmidt, "Applications of state space methods to navigation problems," in C. Leondes (Ed.), *Advances in Control Systems*, Vol. 3, pp. 293-340, Academic Press, New York, 1966.

[Sch81] S. Schmidt, "The Kalman filter: Its recognition and development for aerospace applications," *Journal of Guidance and Control,* **4**(1), pp. 4-7 (January-February 1981).

[Sch68] L. Schwartz and E. Stear, "A computational comparison of several nonlinear filters," *IEEE Transactions on Automatic Control,* **13**(1), pp. 83-86 (February 1968).

[Sch73] F. Schweppe, *Uncertain Dynamic Systems,* Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[Sea67] H. Seal, "The historical development of the Gauss linear model," *Biometrika,* **54**(1-2), pp. 1-24 (1967).

[Sch17] I. Schur, "Uber potenzreihen, die in innern des einheitskrises beschrankt sind," *Journal fuer die Reine und Angewandte Mathematik,* **147**, pp. 205-232 (1917).

[Sch86] I. Schur, "On power series which are bounded in the interior of the unit circle, Part I," in I. Gohberg (Ed.), *I. Schur Methods in Operator Theory and Signal Processing*, pp. 31-88, Birkhauser, Boston, Massachusetts, 1986.

[Ser81] L. Servi and Y. Ho, "Recursive estimation in the presence of uniformly distributed measurement noise," *IEEE Transactions on Automatic Control,* **26**(2), pp. 563-564 (April 1981).

[Sha82] W. Shanahan, "Circuit models for prediction, Wiener filtering, Levinson and Kalman filters for ARMA time series," *IEEE International Conference on Acoustics, Speech and Signal Processing,* pp. 266-269 (May 1982).

[Sha92] U. Shaked and Y. Theodor, "H_∞-optimal estimation: a tutorial," *IEEE Conference on Decision and Control,* pp. 2278-2286 (December 1992).

[She95] X. Shen and L. Deng, "Discrete H_∞ filter design with application to speech enhancement," *IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 1504-1507 (1995).

[She97] X. Shen and L. Deng, "Game theory approach to discrete H_∞ filter design," *IEEE Transactions on Signal Processing,* pp. 1092-1095 (April 1997).

[She99] K. Shen, J. Crossley, A. Lun, S. Kangshen, and H. Liu, *The Nine Chapters on the Mathematical Art: Companion and Commentary,* Oxford University Press, Oxford (1999).

[Shi98] N. Shimada, Y. Shirai, Y. Kuno, and J. Miura, "Hand gesture estimation and model refinement using monocular camera – Ambiguity limitation by inequality constraints," *Third IEEE International Conference on Automatic Face and Gesture Recognition,* pp. 268-273 (April 1998).

[Sil86] B. Silverman, *Density Estimation for Statistics and Data Analysis,* Chapman & Hall, London, 1986.

[Sim96] D. Simon and H. El-Sherief, "Hybrid Kalman/minimax filtering in phase-locked loops," *Control Engineering Practice,* **4**, pp. 615-623 (October 1996).

[Sim02] D. Simon and T. Chia, "Kalman filtering with state equality constraints," *IEEE Transactions on Aerospace and Electronic Systems,* **38**(1), pp. 128-136 (January 2002).

[Sim05a] D. Simon and D.L. Simon, "Aircraft turbofan engine health estimation using constrained Kalman filtering," *ASME Journal of Engineering for Gas Turbines and Power,* **127**(2), pp. 323-328 (April 2005).

[Sim06a] D. Simon, "Reduced order Kalman filtering without model reduction," *Control and Intelligent Systems,* 2006.

[Sim06b] D. Simon and D.L. Simon, "Constrained Kalman filtering via density function truncation for turbofan engine health estimation," submitted for publication.

[Sim06c] D. Simon, "A game theory approach to constrained minimax state estimation," *IEEE Transactions on Signal Processing,* **54**(2), pp. 405-412 (February 2006).

[Sim69] K. Simon and A. Stubberud, "Reduced order Kalman filter," *International Journal of Control,* **10**(5), pp. 501-509 (1969).

[Sim98] J. Simonoff, *Smoothing Methods in Statistics*, Springer-Verlag, New York, 1998.

[Sim82] C. Sims, "Reduced-order modelling and filtering," in C. Leondes (Ed.), *Control and Dynamic Systems, Vol. 18,* pp. 55-103, 1982.

[Sin73] R. Singer and R. Sea, "New results in optimizing surveillance system tracking and data correlation performance in dense multitarget environment," *IEEE Transactions on Automatic Control,* pp. 571-582 (December 1973).

[Sio96] G. Siouris, *An Engineering Approach to Optimal Control and Estimation Theory,* John Wiley & Sons, New York, 1996.

[Ske98] R. Skelton, T. Iwasaki, and K. Grigoriadis, *A Unified Algebraic Approach to Linear Control Design,* Taylor & Francis, London, 1998.

[Skl57] J. Sklansky, "Optimizing the dynamic parameters of a track-while-scan system," *RCA Review,* **18**, pp. 163-185 (June 1957).

[Smi92] A. Smith and A. Gelfand, "Bayesian statistics without tears: A sampling-resampling perspective," *The American Statistician,* **46**(2), pp. 84-88, (May 1992).

[Smi78] M. Smith and A. Roberts, "An exact equivalence between the discrete- and continuous-time formulations of the Kalman filter," *Mathematics and Computers in Simulation,* **20**(2), pp. 102-109 (June 1978).

[Sne97] J. Snell, "A conversation with Joe Doob," *Statistical Science,* **12**(4), pp. 301-311 (November 1997).

[Sob63] V. Sobolev, *A Treatise on Radiative Transfer,* Van Nostrand, Princeton, New Jersey, 1963.

[Son95] Y. Song and J. Grizzle, "The extended Kalman filter as a local asymptotic observer for discrete-time nonlinear systems," *Journal of Mathematical Systems, Estimation, and Control,* **5**(1), pp. 59-78.

[Soo65] T. Soong, "On *a priori* statistics in minimum-variance estimation problems," *ASME Journal of Basic Engineering,* **87D**, pp. 109-112 (March 1965).

[Sor68] H. Sorenson and A. Stubberud, "Nonlinear filtering by approximation of the *a posteriori* density," *International Journal of Control,* **8**(1), pp. 33-51 (1968).

[Sor70] H. Sorenson, "Least squares estimation: from Gauss to Kalman," *IEEE Spectrum,* **7**, pp. 63-68 (July 1970).

[Sor71a] H. Sorenson and J. Sacks, "Recursive fading memory filtering," *Information Science,* **3**, pp. 101-119 (January 1971).

[Sor71b] H. Sorenson and D. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica,* **7**(4), pp. 465-479 (July 1971).

[Sor80] H. Sorenson, *Parameter Estimation,* Marcel Dekker, New York, 1980.

[Sor85] H. Sorenson (ed.), *Kalman Filtering: Theory and Application,* IEEE Press, New York, 1985.

[Sot99] G. Soto, E. Mendes, and A. Razek, "Reduced-order observers for rotor flux, rotor resistance and speed estimation for vector controlled induction motor

drives using the extended Kalman filter technique," *IEE Proceedings – Electric Power Applications,* **146**(3), pp. 282-288 (May 1999).

[Spa88] J. Spall (editor), *Bayesian Analysis of Time Series and Dynamic Models,* Marcel Dekker, New York, 1988.

[Sri70] K. Srinivasan, "State estimation by orthogonal expansion of probability distributions," *IEEE Transactions on Automatic Control,* **AC-15**(1), pp. 3-10 (February 1970).

[Ste68] E. Stear and A. Stubberud, "Optimal filtering for Gauss-Markov noise," *International Journal of Control,* **8**(2), pp. 123-130 (July 1968).

[Ste52] P. Stein, "Some general theorems on iterants," *Journal of Research of the National Bureau of Standards,* **48**, pp. 82-82 (1952).

[Ste94] R. Stengel, *Optimal Control and Estimation,* Dover, New York, 1994.

[Sti86] S. Stigler, *The History of Statistics: The Measurement of Uncertainty Before 1900,* Cambridge, Massachusetts, Harvard University Press, 1986.

[Sto94] A. Stoorvogel and J. Van Schuppen, "An H_∞-parameter estimator and its interpretation," *IFAC Symposium on System Identification,* pp. 1223-1226 (July 1994).

[Str90] P. Strobach, *Linear Prediction Theory,* Springer-Verlag, New York, 1990.

[Swe59] P. Swerling, "First order error propagation in a stagewise smoothing procedure for satellite observations," *The Journal of the Astronautical Sciences,* **6**, pp. 46-52 (1959).

[Swe63] P. Swerling, "Comment on 'A statistical optimizing navigation procedure for space flight'," *AIAA Journal,* **1**, p. 1968 (1963).

[Swe71] P. Swerling, "Modern state estimation methods from the viewpoint of the method of least squares," *IEEE Transactions on Automatic Control,* **AC-16**, pp. 707-719 (1971).

[Tap80] B. Tapley and J. Peters, "Sequential estimation algorithm using a continuous UDU^T covariance factorization," *Journal of Guidance and Control,* **3**(4), pp. 326-331 (July 1980).

[The94a] Y. Theodor, U. Shaked, and C. de Souza, "A game theory approach to robust discrete-time H_∞ estimation," *IEEE Transactions on Signal Processing,* **42**(6), pp. 1486-1495 (June 1994).

[The94b] Y. Theodor and U. Shaked, "Game theory approach to H_∞-optimal discrete-time fixed-point and fixed-lag smoothing," *IEEE Transactions on Automatic Control,* **39**(9), pp. 1944-1948 (September 1994).

[The96] Y. Theodor and U. Shaked, "Robust discrete-time minimum-variance filtering," *IEEE Transactions on Signal Processing,* **44**(2), pp. 181-189 (February 1996).

[Thi61] H. Thiel and A. Goldberger, "On pure and mixed statistical estimation in economics," *International Economic Review,* **2**(1), pp. 65-78 (January 1961).

[Tho94a] S. Thomopoulos and L. Zhang, "Decentralized filtering with random sampling and delay," *Information Sciences,* **81**, pp. 117-131 (1994).

[Tho94b] S. Thomopoulos, "Decentralized filtering and control in the presence of delays: Discrete-time and continuous-time case," *Information Sciences,* **81**, pp. 133-153 (1994).

[Tho77] C. Thornton and G. Bierman, "Gram-Schmidt algorithms for covariance propagation," *International Journal of Control,* **25**(2), pp. 243-260 (February 1977).

[Tsa83] C. Tsai and L. Kurz, "An adaptive robustizing approach to Kalman filtering," *Automatica,* **19**(3), pp. 278-288 (May 1983).

[Tse94] J. Tse, J. Bentsman, and N. Miller, "Minimax long range parameter estimation," *IEEE Conference on Decision and Control,* pp. 277-282, December 1994.

[Urs80] B. Ursin, "Asymptotic convergence properties of the extended Kalman filter using filtered state estimates," *IEEE Transactions on Automatic Control,* **AC-25**(6), pp. 1207-1211 (December 1980).

[Van01] R. Van der Merwe and E. Wan, "The square-root unscented Kalman filter for state and parameter estimation," *IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 3461-3464 (May 2001).

[Vau70] D. Vaughan, "A nonrecursive algebraic solution for the discrete Riccati equation," *IEEE Transactions on Automatic Control,* **AC-15**(5), pp. 597-599 (October 1970).

[Ver84] S. Verdu and H. Poor, "Minimax linear observers and regulators for stochastic systems with uncertain second-order statistics," *IEEE Transactions on Automatic Control,* **AC-29**(6), pp. 499-511 (June 1984).

[Ver86] M. Verhaegen and P. Van Dooren, "Numerical aspects of different Kalman filter implementations," *IEEE Transactions on Automatic Control,* **AC-31**(10), pp. 907-917 (October 1986).

[Ver95] M. Verlaan and A. Hemmink, "Reduced rank square root filters for large scale data assimilation problem," *Second International Symposium on Assimilation of Observations in Meteorology and Oceanography,* pp. 247-252 (1995).

[Ver02] J. Vermaak, C. Andrieu, A. Doucet, and S. Godsill, "Particle methods for Bayesian modeling and enhancement of speech signals," *IEEE Transactions on Speech and Audio Processing,* **10**, pp. 173-185 (March 2002).

[Wan01] E. Wan and R. van der Merwe, "The unscented Kalman filter," in *Kalman Filtering and Neural Networks,* S. Haykin (ed.), John Wiley & Sons, New York, 2001.

[Wat22] G. Watson, *A Treatise on the Theory of Bessel Functions,* Cambridge University Press, Cambridge, United Kingdom, 1922.

[Wei74] R. Weinstock, *Calculus of Variations,* Dover, New York, 1974.

[Wel87] M. Welch, "A Kalman filter perspective," *The American Statistician,* **41**, pp. 90-91 (1987).

[Wen92] W. Wen and H. Durrant-Whyte, "Model-based multi-sensor data fusion," *IEEE International Conference on Robotics and Automation,* pp. 1720-1726, May 1992.

[Wes85] M. West, P. Harrison, and H. Mignon, "Dynamic generalized linear models and Bayesian forecasting," *Journal of the American Statistical Association,* **80**(389), pp. 73-97 (March 1985).

[Wie56] N. Wiener, *I Am a Mathematician,* MIT Press, Cambridge, Massachusetts, 1956.

[Wie64] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series,* MIT Press, Cambridge, Massachusetts, 1964.

[Wil81] W. Willman, "Edgeworth expansions in state perturbation estimation," *IEEE Transactions on Automatic Control,* **AC-26**(2), pp. 493-498 (April 1981).

[Wis69] R. Wishner, J. Tabaczynski, and M. Athans, "A comparison of three non-linear filters," *Automatica,* **5**(4), pp. 487-496 (July 1969).

[Wis70] R. Wishner, R. Larson, and M. Athans, "Status of radar tracking algorithms," *Symposium on Nonlinear Estimation Theory and its Applications,* San Diego, pp. 32-54, 1970.

[Wri97] S. Wright, *Primal-Dual Interior-Point Methods,* SIAM Press, Philadelphia, Pennsylvania, 1997.

[Xie94] L. Xie, C. Soh, and C. de Souza, "Robust Kalman filtering for uncertain discrete-time systems," *IEEE Transactions on Automatic Control,* **39**(6), pp. 1310-1314 (June 1994).

[Xie04] L. Xie, L. Lu, D. Zhang, and H. Zhang, "Improved robust H_2 and H_∞ filtering for uncertain discrete-time systems," *Automatica,* **40**(5), pp. 873-880 (May 2004).

[Xu02] S. Xu and T. Chen, "Reduced-order H_∞ filtering for stochastic systems," *IEEE Transactions on Signal Processing,* **50**(12), pp. 2998-3007 (December 2002).

[Yae91] I. Yaesh and U. Shaked, "A transfer function approach to the problems of discrete-time systems: H_∞-optimal linear control and filtering," *IEEE Transactions on Automatic Control,* **36**(11), pp. 1264-1271 (November 1991).

[Yae92] I. Yaesh and U. Shaked, "Game theory approach to state estimation of linear discrete-time processes and its relation to H_∞-optimal estimation," *International Journal of Control,* **55**(6), pp. 1443-1452 (1992).

[Yae04] I. Yaesh and U. Shaked, "Min-max Kalman filtering," *Systems and Control Letters,* **53**(3-4), pp. 217-228 (November 2004).

[Yon80] K. Yonezawa, "Reduced-order Kalman filtering with incomplete observability," *Journal of Guidance and Control,* **3**(3), pp. 280-282 (June 1980).

[Yoo04] M. Yoon, V. Ugrinovskii, and I. Petersen, "Robust finite horizon minimax filtering for discrete-time stochastic uncertain systems," *Systems and Control Letters,* **52**(2), pp. 99-112 (June 2004).

[Zad50] L. Zadeh and J. Ragazzini, "An extension of Wiener's theory of prediction," *Journal of Applied Physics,* **21**, pp. 645-655 (July 1950).

[Zar00] P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach,* American Institute of Aeronautics and Astronautics, Reston, Virginia, 2000.

[Zha05a] H. Zhang, L. Xie, Y. Chai Soh, and D. Zhang, "H_∞ fixed-lag smoothing for discrete linear time-varying systems," *Automatica,* **41**(5), pp. 839-846 (May 2005).

[Zha95] X. Zhang, A. Heemink, and J. Van Eijkeren, "Performance robustness analysis of Kalman filter for linear discrete-time systems under plant and noise uncertainty," *International Journal of Systems Science,* **26**(2), pp. 257-275 (February 1995).

[Zha05b] Y. Zhang, Y. Soh, and W. Chen, "Robust information filter for decentralized estimation," *Automatica,* **41**(12), pp. 2141-2146 (December 2005).

INDEX

A

α filter, 203
α-β filter, 199, 491
α-β-γ filter, 202, 491
acronyms, xv
active set method, 216, 340, 384
aircraft, 181, 184, 397, 423, 457, 494, 499
applications of state estimation, xxii
astrophysics, 100
auxiliary particle filter, 470

B

Babylonians, 17, 490
banana, 494
bandwidth, 496
batch filter, 178
battery, 102–103
Bayes' rule, 52
Bayesian estimator, 462, 465
biomedical engineering, 227
black hole, 470
bootstrap filter, *see* particle filter

C

calculus of variations, 97
Cayley–Hamilton theorem, 18, 46
central limit theorem, 62
Chandrasekhar algorithm, 178, 238, 242, 261
Chapman–Kolmogorov equation, 54
chemicals, 91, 118
Chinese, ancient, 17
Cholesky decomposition, 160, 180, 473
Cholesky triangle, 160
choose function, 50
communication system, 480
condensation algorithm, *see* particle filter
consider state, 311
constrained filter, 212, 223, 381
  H_∞, 381
  model reduction, 212
  pdf truncation, 212, 218
  perfect measurements, 212–213
  projection approach, 212, 214
    general, 216
    least squares, 215
    maximum probability, 214
constrained optimization, 337
control vector, 19
controllability grammian, 38–39
controllability matrix, 38–39
convolution, 94
correction term, 84
cost function, 80
  augmented, 338
covariance filter, 178
curse of dimensionality, 461
curve fitting, 92
  linear, 92
  quadratic, 93

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

D

decentralized filtering, 317
decision theory, 490
delayed measurement, 317, 320
digital signal processing, 491
discretization, 26, 379
duality, 258, 487, 490–491

E

e-mail address, xxiii
Earth, dimensions of, 485
ecology, 480
error function, 220
estimation
  a posteriori, 124
  a priori, 125
  covariance, 85
  frequency domain, 94
  gain matrix, 84
  least squares, 79
    alternate forms, 86
    history, 485–486
    recursive, 84
    weighted, 82
  maximum likelihood, 481, 486
  mean based, 103, 481
  median based, 103
  minimax based, 104, 481
  minimum mean square, 481
  mode based, 103
  of a constant, 80
  of the mean, 103
  of the variance, 103
  predicted, 125
  smoothed, 125
  unbiased, 84

F

f-g filter, *see* α-β filter
f-g-h filter, *see* α-β-γ filter
fading-memory filter, 140, 143, 208
falling body, 405, 417, 427, 451, 456, 471, 477, 479
fault detection, 480
feedback control, xxii
finite precision arithmetic, 140
first order hold, 111
fish, 145
football, 154, 157
Fourier transform, 71
FPGA, 481
free lunch, 462
fuzzy logic, 463

G

g-h filter, *see* α-β filter
g-h-k filter, *see* α-β-γ filter
game theory, 343, 381
Gaussian elimination, 17
Gaussian sum filter, 420–421
giants, shoulders of, 485
Givens transformation, 163, 171
GPS, 223, 386, 490
Gram–Schmidt transformation, 163, 171, 219
  modified, 163, 171–172, 180
grid-based filter, 422
gyroscope, 240, 261–262

H

H_∞ filter, 336, 367
  a posteriori, 365
  a priori, 365
  combined with Kalman, 374, 376–377
  comparison with Kalman, 354, 361, 368
  constrained, 381
  continuous, 361
  limitations, 360, 368
  model uncertainties, 377
  nonlinear, 368
  optimality, 374
  performance bound, 344, 365
  philosophy, 343
  polynomial systems approach, 368
  sensitivity to design parameters, 368
  stability, 357
  steady state, 354
  system identification, 368
  transfer function, 357, 364–365
  weighting matrices, 344
Hamiltonian function, 341, 345
Hamiltonian matrix, 205, 226, 366, 370
Householder transformation, 163, 171
hypersphere, 473, 482

I

image processing, 480
impulse function, 72
  sifting property, 112, 119, 231
infimum, 383
information filter, 156, 178
  backward, 282
information matrix, 334
input matrix, 19
interacting particles, *see* particle filter
inverted pendulum, 459
investments, 12

J

j (complex variable), 4
Jacobian, 457
Johns Hopkins University, 229, 486
Jordan form, 18, 20, 43, 74, 153, 205, 219
Jupiter, 485

K

Kalman–Bucy filter, 229, 400
Kalman–Schmidt filter, 487
Kalman filter
  adaptive, 298, 300, 313, 497
  bank, 300–301

batch, 154
combined with H_∞, 374, 376–377
comparison with H_∞, 354, 361, 368
comparison with Wiener filter, 490
continuous, 229
alternate form, 238
gain, 236
innovations, 301, 326
steady state, 252
decentralized, 491
decoupled, 490
discrete, 123
alternate form, 135, 149, 265, 318, 334
closed form, 138
gain, 128
innovations, 130, 298, 300, 326
Joseph stabilized version, 145
measurement update, 127
offline calculation, 129
one-step equations, 131
steady state, 137, 193
time update, 126–127
divergence, 140
extended, 400, 416, 462
continuous, 400–401
difficulties, 433, 439, 462
discrete, 407, 409
hybrid, 403, 405
iterated, 410–411
second order, 413, 419
steady state, 404
fictitious process noise, 140, 142
history, 485, 490–491
initialization, 140
Joseph stabilized version, 129
limitations, 336
linearized, 397, 399
multiple model, 301
optimality, 336, 374, 465, 480
philosophy, 343
properties, 129, 145
proportional integral, 326
reduced order, 305
Anderson's approach, 306
covariance approximation, 305
decoupled, 305
model reduction, 305
Schmidt's approach, 309
robust, 312
sequential, 150
square root, 140
stability, 196, 254, 297, 308, 317
symmetrization, 140
transition matrix approach, 238
verification, 298
kernel
bandwidth, 472, 483
biweight, 483
density, 472
Epanechnikov, 473, 483
Gaussian, 483
triangular, 483
uniform, 483
Krein space, 369, 491
Kronecker delta function, 72, 124

L

Lagrange multiplier, 214–215, 337–338, 345
linearization, 22, 397, 434
covariance approximation, 446
mean approximation, 444
operating point, 23
Lyapunov equation
continuous, 39, 41, 115
discrete, 39, 42, 108

M

MATLAB, xxiii, 491
functions
CARE, 253, 256, 262
CHOL, 160, 441
COND, 159
DARE, 194, 225, 355
DLYAP, 110
EXP, 19
EXPM, 19
LYAP, 119
QP, 217
RESIDUE, 99
matrix, 4
matrix inversion lemma, 11
matrix
algebra, 6
calculus, 14
characteristic roots, 9
condition number, 159, 328
degenerate, 4
determinant, 7
product rules, 14
diagonalization, 153
dimension, 4
eigendata, 9
eigenvalues, 8–9
eigenvectors, 9
exponential, 19
hermitian, 5
hermitian transpose, 5
history, 17–18
identity, 7
indefinite, 10
infinity-norm, 359
inverse, 8, 10
derivative, 14
trace, 294
Laplace expansion, 7
latent roots, 9
latent vectors, 9
negative definite, 10
negative semidefinite, 10
noninvertible, 8
nonnegative definite, 10
nonpositive definite, 10

nullity, 5
positive definite, 10
positive semidefinite, 10
product, 9
proper numbers, 9
proper vectors, 9
rank, 4–5
singular, 8
singular values, 10, 159, 180
square root, 160, 246, 441, 458, 473
symmetric, 5, 10
symplectic, 205, 226, 370
trace, 9
inverse, 294
transpose, 5
triangular, 261
triangularization, 169
MCMC resampling, 470, 476
meaning of life, 493
measurement residual, 80
Metropolis–Hastings acceptance, 476
minimax filter, 336, 367
see also H_∞ filter
missile tracking, 486
MIT, 486
model errors, 140
model uncertainties, 377
modified matrices formula, 12
monotonic function, 59
Monte Carlo filter, *see* particle filter
moon, 485, 487
motor, 20, 25, 47, 379, 401, 429, 457

N

NASA, xiii, 158, 165, 246, 486–487
neural networks, 480
noise
artificial, 423
bandlimited, 252
biased, 369, 398, 426
colored, 71, 183, 188, 247, 249, 309
simulation, 73
control, 398, 426
correlated, 184, 247–248
nonlinear, 450
power spectrum, 252
white, 71, 112, 230–231
zero measurement, 190, 214, 250
nominal operating point, 23, 397
nonlinear filter, 395, 425, 490
benchmark problem, 469
stability, 425
tradeoffs, 396, 480
notation, xxiii, xxv
nuisance state, 311
nuisance variable, 311
numerical difficulties, 158, 246

O

objective function, 80
observability grammian, 41–42
observability matrix, 41–42
Occam's razor, 498
Ohm's Law, 395
open research areas, xxv, 224, 317, 377, 388, 396, 457, 480
optimal control, 258
optimization
constrained, 214
order statistics, 467
output matrix, 19
output vector, 19

P

parameter estimation, 422, 428, 487
parsimony, 498
particle filter, 466, 468
auxiliary, 470, 476
comparison with unscented filter, 480
difficulties, 462, 467
extended, 477
H_∞, 478
implementation issues, 469
prior editing, 470, 472
regularized, 470, 472
resampling, 466–468
roughening, 470
sample impoverishment, 469
unscented, 477
pattern recognition, 480
perfection, 497
persistent excitation, 496
perturbation estimator, 326
pleasure, 494
poker, 51
population, 109, 327
prerequisites, xxiii, xxv
primal-dual interior-point method, 341
prior editing, 470
probability density function, 53
conditional, 54
joint, 61
marginal, 62
multimodal, 463, 473
probability distribution function, 53
conditional, 54
joint, 61
marginal, 61
probability
a posteriori, 51
a priori, 51
conditional, 51
definition, 50
independence, 52
joint, 51
proportional integral Kalman filter, 326
pseudo inverse, 81

Q

Q-function, 54
quadratic programming, 216
quaternions, 18

R

radar, 104
radioactivity, 145, 179, 370
RAND, 486
random process, 68
random sequence
  continuous, 68
  discrete, 68
random variable, 49, 53
  average, 54
  Cauchy, 76
  central moment, 56
  continuous, 53
  correlation coefficient, 62
  covariance, 62
  definition, 53
  discrete, 53
  expectation, 54
  expected value, 54
  exponential, 75
  function, 55, 59, 67
  Gaussian, 57
  independence, 62
  Laplace, 57
  mean, 54
  moment, 56
  multivariable, 65
  normal, 57
  orthogonal, 63
  realization, 53
  skew, 56
  skewness, 56
  standard deviation, 56
  transformation, 59
  uncorrelated, 62
  uniform, 56
  variance, 55
  vector, 65
    autocorrelation, 66
    autocovariance, 66
    correlation, 66
    covariance, 66
    Gaussian, 67
recursion, *see* recursion
regularized particle filter, 470
research funding, 144
residual sampling, 467
retrodiction, 322
retrodiction filter, 294
return function, 80
Riccati equation
  continuous, 235, 248, 253, 259, 361
  discrete, 131, 158, 194, 196, 314, 355, 374, 378, 389
RL circuit, 112
RLC circuit, 39, 42, 47, 119, 148, 181
rocket, 48
roughening, 470
running average, 90

S

saddle point, 351, 383
satellite, 264, 274, 309, 428, 486
Saturn, 485
scalar, 4
sequential filter, 178
sequential importance, *see* particle filter
Sherman-Morrison formula, 12
sigma point, 441, 449, 452
signal, two-norm, 358
simulation, 27, 232
  colored noise, 73
  Euler integration, 29
  rectangular integration, 29
  Runge Kutta integration, 31
  trapezoidal integration, 29
slot machine, rigged, 63
smoother
  constant states, 274
  fixed-interval, 264, 279
    forward–backward, 280, 285
    RTS, 286
  fixed-lag, 264, 274
  fixed-point, 264, 267
  Gaussian sum, 422
  improvement, 270
  stability, 295
solution manual, xxiv
spacecraft, 400, 489
speech recognition, 480
square root filter, 158, 178, 238, 246, 490–491
  Potter's algorithm, 158, 165
  triangularization, 169
state, xxi
state space models, 18
state transition matrix, 19, 109, 111, 114, 320
  invertibility, 282, 320, 377
state vector, 19
stationary point, 81, 351
steel production, 105
Stein equation, 39, 42, 108
stochastic process, 68
  autocorrelation, 69
  autocovariance, 69
  colored, 71
  continuous, 68
  cross correlation, 71
  cross covariance, 71
  cross power spectrum, 72
  discrete, 68
  ergodic, 70
  power, 72
  power density, 72
  power density spectrum, 72
  power spectral density, 72
  power spectrum, 71

probability density function
joint, 69
second-order, 69
probability distribution function
joint, 69
second-order, 69
stationary, 69
strict-sense stationary, 69
time autocorrelation, 70
time average, 70
uncorrelated, 71
white, 71
wide-sense stationary, 69
stock market, 70
stratified sampling, 467
student, tired, 82
superposition, 95
supremum, 360, 383
surveying, 486
survival of the fittest, *see* particle filter
Sylvester equation, 39, 41, 115
synchronization error, 317
system identification, 303, 496
system matrix, 19
system
algebraically equivalent, 43
anticausal, 99
augmented, 424
causal, 96
continuous
covariance, 114
mean, 114
controllability, 38, 195, 197, 253–254
detectability, 43, 197, 254
discrete
covariance, 108
mean, 107
distributed parameter, 490
infinity-norm, 359
linear, 18–19, 27
nonexistence of, 395
modes, 43
Newtonian, 132, 199, 202, 226, 261, 271, 279, 285, 316, 390
nonlinear, 22, 395
observability, 38
sampled-data
covariance, 111
mean, 111
second order, 296, 303, 423
stability, 33–34, 37
stabilizability, 43, 197, 254
systematic resampling, 467

T

target maneuvering index, 201, 203
target tracking index, 201
Taylor series, 23, 337, 397, 407, 413, 439
tire tread, 328–329
toast, 343
tracking, 317, 480, 491

U

U-D filter, 174, 178, 180, 490
unscented filter, 447
comparison with particle filter, 480
difficulties, 462
simplex, 454
spherical, 455
square root, 457
weighting coefficient, 442, 444, 453
unscented transformation, 441, 446, 452
covariance approximation, 446
mean approximation, 444

V

variable structure filter, 326
vector, 4
calculus, 14
column, 4
Euclidean norm, 9
quadratic form, 10
row, 4
two-norm, 9–10, 358
vehicle navigation, 223, 370, 386, 430, 459

W

weather forecasting, 305
Web site, xxiii
Wiener–Hopf equation, 100, 259
Wiener–Kalman filter, 489
Wiener–Khintchine relations, 71
Wiener filter, 94, 257, 486
causal, 100
comparison to Kalman filter, 490
noncausal, 98
parametric, 96
wombat, 147
Woodbury's identity, 12
workhorse of state estimation, xxiv
World War II, 94

Y

yellow peril, 94