
THIRD EDITION

• Digital Control •
of Dynamic Systems

Gene F. Franklin
Stanford University

J. David Powell
Stanford University

Michael L. Workman
IBM Corporation



ADDISON-WESLEY

An imprint of Addison Wesley Longman, Inc.

Menlo Park, California • Reading, Massachusetts • Harlow, England
Berkeley, California • Don Mills, Ontario • Sydney • Bonn • Amsterdam • Tokyo • Mexico City

**World Student Series**

*Editorial Assistant, Royden Tonomura Proofreader, Holly McLean-Aldis
Senior Production Editor, Teri Hyde Composer, Eigentype Compositors
Art and Design Supervisor, Kevin Berry Cover Design, Yvo Riezebos
Manufacturing Supervisor, Janet Weaver Illustrations, Scientific Illustrators.
Copyeditor, Steve Feinstein Bruce Saltzman*

Copyright © 1998, Addison Wesley Longman, Inc.

All rights reserved. No part of this publication may be reproduced, or stored in a database or retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America. Printed simultaneously in Canada.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial caps or in all caps.

MATLAB is a registered trademark of The Math Works, Inc.

24 Prime Park Way, Natick, MA 01760-1520.
Phone: (508) 653-1415, Fax: (508) 653-2997

Email: info@mathworks.com

Cover Photo: Telegraph Colour Library/FPG International LLC.

Library of Congress Cataloging-in-Publication Data

Franklin, Gene F.
Digital control of dynamic systems / Gene F. Franklin, J. David

Powell, Michael L. Workman. – 3rd ed.

p. cm.

Includes index.

ISBN 0-201-33153-5

1. Digital control systems. 2. Dynamics. I. Powell, J. David, 1938– . II. Workman, Michael L. III. Title.
TJ223.M53F73 1997

629.8'9 – dc21

97-35994

CIP

Instructional Material Disclaimer:

The programs presented in this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. Neither the publisher nor the authors offer any warranties or representations, nor do they accept any liabilities with respect to the programs.

ISBN 0-201-33153-5

1 2 3 4 5 6 7 8 9 10-MA-01 00 99 98 97

Addison Wesley Longman, Inc.
2725 Sand Hill Road
Menlo Park, CA 94025

Additional Addison Wesley Longman Control Engineering titles:

Feedback Control of Dynamic Systems,
Third Edition, 0-201-52747-2
Gene F. Franklin and J. David Powell

Modern Control Systems,
Eighth Edition, 0-201-30864-9
Richard C. Dorf and
Robert H. Bishop

The Art of Control Engineering,
0-201-17545-2
Ken Dutton, Steve Thompson,
and Bill Barraclough

Introduction to Robotics,
Second Edition, 0-201-09529-9
John J. Craig

Fuzzy Control,
0-201-18074-X
Kevin M. Passino and Stephen Yurkovich

Adaptive Control,
Second Edition, 0-201-55866-1
Karl J. Åstrom and Bjorn Wittenmark

Control Systems Engineering,
Second Edition, 0-8053-5424-7
Norman S. Nise

Computer Control of Machines and Processes,
0-201-10645-0
John G. Bollinger and Neil A. Duffie

Multivariable Feedback Design,
0-201-18243-2
Jan Maciejowski

• Contents •

Preface xix

1 Introduction 1

1.1 Problem Definition	1
1.2 Overview of Design Approach	5
1.3 Computer-Aided Design	7
1.4 Suggestions for Further Reading	7
1.5 Summary	8
1.6 Problems	8

2 Review of Continuous Control 11

2.1 Dynamic Response	11
2.1.1 Differential Equations	12
2.1.2 Laplace Transforms and Transfer Functions	12
2.1.3 Output Time Histories	14
2.1.4 The Final Value Theorem	15
2.1.5 Block Diagrams	15
2.1.6 Response versus Pole Locations	16
2.1.7 Time-Domain Specifications	20
2.2 Basic Properties of Feedback	22

2.2.1 Stability	22	4.2.1 The z -Transform	79
2.2.2 Steady-State Errors	23	4.2.2 The Transfer Function	80
2.2.3 PID Control	24	4.2.3 Block Diagrams and State-Variable Descriptions	82
2.3 Root Locus	24	4.2.4 Relation of Transfer Function to Pulse Response	90
2.3.1 Problem Definition	25	4.2.5 External Stability	93
2.3.2 Root Locus Drawing Rules	26	4.3 Discrete Models of Sampled-Data Systems	96
2.3.3 Computer-Aided Loci	28	4.3.1 Using the z -Transform	96
2.4 Frequency Response Design	31	4.3.2 *Continuous Time Delay	99
2.4.1 Specifications	32	4.3.3 State-Space Form	101
2.4.2 Bode Plot Techniques	34	4.3.4 *State-Space Models for Systems with Delay	110
2.4.3 Steady-State Errors	35	4.3.5 *Numerical Considerations in Computing Φ and Γ	114
2.4.4 Stability Margins	36	4.3.6 *Nonlinear Models	117
2.4.5 Bode's Gain-Phase Relationship	37	4.4 Signal Analysis and Dynamic Response	119
2.4.6 Design	38	4.4.1 The Unit Pulse	120
2.5 Compensation	39	4.4.2 The Unit Step	120
2.6 State-Space Design	41	4.4.3 Exponential	121
2.6.1 Control Law	42	4.4.4 General Sinusoid	122
2.6.2 Estimator Design	46	4.4.5 Correspondence with Continuous Signals	125
2.6.3 Compensation: Combined Control and Estimation	48	4.4.6 Step Response	128
2.6.4 Reference Input	48	4.5 Frequency Response	131
2.6.5 Integral Control	49	4.5.1 *The Discrete Fourier Transform (DFT)	134
2.7 Summary	50	4.6 Properties of the z-Transform	137
2.8 Problems	52	4.6.1 Essential Properties	137
3 Introductory Digital Control	57	4.6.2 *Convergence of z -Transform	142
3.1 Digitization	58	4.6.3 *Another Derivation of the Transfer Function	146
3.2 Effect of Sampling	63	4.7 Summary	148
3.3 PID Control	66	4.8 Problems	149
3.4 Summary	68	5 Sampled-Data Systems	155
3.5 Problems	69	5.1 Analysis of the Sample and Hold	156
4 Discrete Systems Analysis	73	5.2 Spectrum of a Sampled Signal	160
4.1 Linear Difference Equations	73	5.3 Data Extrapolation	164
4.2 The Discrete Transfer Function	78	5.4 Block-Diagram Analysis of Sampled-Data Systems	170
		5.5 Calculating the System Output Between Samples: The Ripple	180

5.6 Summary	182	8.1.1 Pole Placement	282
5.7 Problems	183	8.1.2 Controllability	285
5.8 Appendix	186	8.1.3 Pole Placement Using CACSD	286
6 Discrete Equivalents	187	8.2 Estimator Design	289
6.1 Design of Discrete Equivalents via Numerical Integration	189	8.2.1 Prediction Estimators	290
6.2 Zero-Pole Matching Equivalents	200	8.2.2 Observability	293
6.3 Hold Equivalents	202	8.2.3 Pole Placement Using CACSD	294
6.3.1 Zero-Order Hold Equivalent	203	8.2.4 Current Estimators	295
6.3.2 A Non-Causal First-Order-Hold Equivalent: The Triangle-Hold Equivalent	204	8.2.5 Reduced-Order Estimators	299
6.4 Summary	208	8.3 Regulator Design: Combined Control Law and Estimator	302
6.5 Problems	209	8.3.1 The Separation Principle	302
7 Design Using Transform Techniques	211	8.3.2 Guidelines for Pole Placement	308
7.1 System Specifications	212	8.4 Introduction of the Reference Input	310
7.2 Design by Emulation	214	8.4.1 Reference Inputs for Full-State Feedback	310
7.2.1 Discrete Equivalent Controllers	215	8.4.2 Reference Inputs with Estimators: The State-Command Structure	314
7.2.2 Evaluation of the Design	218	8.4.3 Output Error Command	317
7.3 Direct Design by Root Locus in the z -Plane	222	8.4.4 A Comparison of the Estimator Structure and Classical Methods	319
7.3.1 z -Plane Specifications	222	8.5 Integral Control and Disturbance Estimation	322
7.3.2 The Discrete Root Locus	227	8.5.1 Integral Control by State Augmentation	323
7.4 Frequency Response Methods	234	8.5.2 Disturbance Estimation	328
7.4.1 Nyquist Stability Criterion	238	8.6 Effect of Delays	337
7.4.2 Design Specifications in the Frequency Domain	243	8.6.1 Sensor Delays	338
7.4.3 Low Frequency Gains and Error Coefficients	259	8.6.2 Actuator Delays	341
7.4.4 Compensator Design	260	8.7 *Controllability and Observability	345
7.5 Direct Design Method of Ragazzini	264	8.8 Summary	351
7.6 Summary	269	8.9 Problems	352
7.7 Problems	270		
8 Design Using State-Space Methods	279	9 Multivariable and Optimal Control	359
8.1 Control Law Design	280	9.1 Decoupling	360
		9.2 Time-Varying Optimal Control	364
		9.3 LQR Steady-State Optimal Control	371
		9.3.1 Reciprocal Root Properties	372
		9.3.2 Symmetric Root Locus	373

9.3.3	Eigenvector Decomposition	374
9.3.4	Cost Equivalents	379
9.3.5	Emulation by Equivalent Cost	380
9.4	Optimal Estimation	382
9.4.1	Least-Squares Estimation	383
9.4.2	The Kalman Filter	389
9.4.3	Steady-State Optimal Estimation	394
9.4.4	Noise Matrices and Discrete Equivalents	396
9.5	Multivariable Control Design	400
9.5.1	Selection of Weighting Matrices Q_1 and Q_2	400
9.5.2	Pincer Procedure	401
9.5.3	Paper-Machine Design Example	403
9.5.4	Magnetic-Tape-Drive Design Example	407
9.6	Summary	419
9.7	Problems	420

10 Quantization Effects 425

10.1	Analysis of Round-Off Error	426
10.2	Effects of Parameter Round-Off	437
10.3	Limit Cycles and Dither	440
10.4	Summary	445
10.5	Problems	445

11 Sample Rate Selection 449

11.1	The Sampling Theorem's Limit	450
11.2	Time Response and Smoothness	451
11.3	Errors Due to Random Plant Disturbances	454
11.4	Sensitivity to Parameter Variations	461
11.5	Measurement Noise and Antialiasing Filters	465
11.6	Multirate Sampling	469
11.7	Summary	474
11.8	Problems	476

12 System Identification 479

12.1	Defining the Model Set for Linear Systems	481
12.2	Identification of Nonparametric Models	484
12.3	Models and Criteria for Parametric Identification	495
12.3.1	Parameter Selection	496
12.3.2	Error Definition	498
12.4	Deterministic Estimation	502
12.4.1	Least Squares	503
12.4.2	Recursive Least Squares	506
12.5	Stochastic Least Squares	510
12.6	Maximum Likelihood	521
12.7	Numerical Search for the Maximum-Likelihood Estimate	526
12.8	Subspace Identification Methods	535
12.9	Summary	538
12.10	Problems	539

13 Nonlinear Control 543

13.1	Analysis Techniques	544
13.1.1	Simulation	545
13.1.2	Linearization	550
13.1.3	Describing Functions	559
13.1.4	Equivalent Gains	573
13.1.5	Circle Criterion	577
13.1.6	Lyapunov's Second Method	579
13.2	Nonlinear Control Structures: Design	582
13.2.1	Large Signal Linearization: Inverse Nonlinearities	582
13.2.2	Time-Optimal Servomechanisms	599
13.2.3	Extended PTOS for Flexible Structures	611
13.2.4	Introduction to Adaptive Control	615
13.3	Design with Nonlinear Cost Functions	635
13.3.1	Random Neighborhood Search	635
13.4	Summary	642
13.5	Problems	643

14 Design of a Disk Drive Servo: A Case Study 649

14.1 Overview of Disk Drives	650
14.1.1 High Performance Disk Drive Servo Profile	652
14.1.2 The Disk-Drive Servo	654
14.2 Components and Models	655
14.2.1 Voice Coil Motors	655
14.2.2 Shorted Turn	658
14.2.3 Power Amplifier Saturation	659
14.2.4 Actuator and HDA Dynamics	660
14.2.5 Position Measurement Sensor	663
14.2.6 Runout	664
14.3 Design Specifications	666
14.3.1 Plant Parameters for Case Study Design	667
14.3.2 Goals and Objectives	669
14.4 Disk Servo Design	670
14.4.1 Design of the Linear Response	671
14.4.2 Design by Random Numerical Search	674
14.4.3 Time-Domain Response of XPTOS Structure	678
14.4.4 Implementation Considerations	683
14.5 Summary	686
14.6 Problems	687

Appendix A Examples 689

A.1 Single-Axis Satellite Attitude Control	689
A.2 A Servomechanism for Antenna Azimuth Control	691
A.3 Temperature Control of Fluid in a Tank	694
A.4 Control Through a Flexible Structure	697
A.5 Control of a Pressurized Flow Box	699

Appendix B Tables 701

B.1 Properties of z -Transforms	701
B.2 Table of z -Transforms	702

Appendix C A Few Results from Matrix Analysis 705

C.1 Determinants and the Matrix Inverse	705
C.2 Eigenvalues and Eigenvectors	707

C.3 Similarity Transformations	709
C.4 The Cayley-Hamilton Theorem	711

Appendix D Summary of Facts from the Theory of Probability and Stochastic Processes 713

D.1 Random Variables	713
D.2 Expectation	715
D.3 More Than One Random Variable	717
D.4 Stochastic Processes	719

Appendix E MATLAB Functions 725**Appendix F Differences Between MATLAB v5 and v4 727**

F.1 System Specification	727
F.2 Continuous to Discrete Conversion	729
F.3 Optimal Estimation	730

References 731**Index 737**

• Preface •

This book is about the use of digital computers in the real-time control of dynamic systems such as servomechanisms, chemical processes, and vehicles that move over water, land, air, or space. The material requires some understanding of the Laplace transform and assumes that the reader has studied linear feedback controls. The special topics of discrete and sampled-data system analysis are introduced, and considerable emphasis is given to the z -transform and the close connections between the z -transform and the Laplace transform.

The book's emphasis is on designing digital controls to achieve good dynamic response and small errors while using signals that are sampled in time and quantized in amplitude. Both transform (classical control) and state-space (modern control) methods are described and applied to illustrative examples. The transform methods emphasized are the root-locus method of Evans and frequency response. The root-locus method can be used virtually unchanged for the discrete case; however, Bode's frequency response methods require modification for use with discrete systems. The state-space methods developed are the technique of pole assignment augmented by an estimator (observer) and optimal quadratic-loss control. The optimal control problems use the steady-state constant-gain solution; the results of the separation theorem in the presence of noise are stated but not proved.

Each of these design methods—classical and modern alike—has advantages and disadvantages, strengths and limitations. It is our philosophy that a designer must understand all of them to develop a satisfactory design with the least effort.

Closely related to the mainstream of ideas for designing linear systems that result in satisfactory dynamic response are the issues of sample-rate selection, model identification, and consideration of nonlinear phenomena. Sample-rate selection is discussed in the context of evaluating the increase in a least-squares performance measure as the sample rate is reduced. The topic of model making is treated as measurement of frequency response, as well as least-squares parameter estimation. Finally, every designer should be aware that all models are nonlinear

and be familiar with the concepts of the describing functions of nonlinear systems, methods of studying stability of nonlinear systems, and the basic concepts of nonlinear design.

Material that may be new to the student is the treatment of signals which are discrete in time and amplitude and which must coexist with those that are continuous in both dimensions. The philosophy of presentation is that new material should be closely related to material already familiar, and yet, by the end, indicate a direction toward wider horizons. This approach leads us, for example, to relate the z -transform to the Laplace transform and to describe the implications of poles and zeros in the z -plane to the known meanings attached to poles and zeros in the s -plane. Also, in developing the design methods, we relate the digital control design methods to those of continuous systems. For more sophisticated methods, we present the elementary parts of quadratic-loss Gaussian design with minimal proofs to give some idea of how this powerful method is used and to motivate further study of its theory.

The use of computer-aided design (CAD) is universal for practicing engineers in this field, as in most other fields. We have recognized this fact and provided guidance to the reader so that learning the controls analysis material can be integrated with learning how to compute the answers with MATLAB, the most widely used CAD software package in universities. In many cases, especially in the earlier chapters, actual MATLAB scripts are included in the text to explain how to carry out a calculation. In other cases, the MATLAB routine is simply named for reference. All the routines given are tabulated in Appendix E for easy reference; therefore, this book can be used as a reference for learning how to use MATLAB in control calculations as well as for control systems analysis. In short, we have tried to describe the entire process, from learning the concepts to computing the desired results. But we hasten to add that it is mandatory that the student retain the ability to compute simple answers by hand so that the computer's reasonableness can be judged. The First Law of Computers for engineers remains "Garbage In, Garbage Out."

Most of the graphical figures in this third edition were generated using MATLAB® supplied by The Mathworks, Inc. The files that created the figures are available from Addison Wesley Longman at ftp.aw.com or from The Mathworks, Inc. at ftp.mathworks.com/pub/books/franklin. The reader is encouraged to use these MATLAB figure files as an additional guide in learning how to perform the various calculations.

To review the chapters briefly: Chapter 1 contains introductory comments. Chapters 2 and 3 are new to the third edition. Chapter 2 is a review of the prerequisite continuous control; Chapter 3 introduces the key effects of sampling in order to elucidate many of the topics that follow. Methods of linear analysis are presented in Chapters 4 through 6. Chapter 4 presents the z -transform. Chapter 5 introduces combined discrete and continuous systems, the sampling theorem,

and the phenomenon of aliasing. Chapter 6 shows methods by which to generate discrete equations that will approximate continuous dynamics. The basic deterministic design methods are presented in Chapters 7 and 8—the root-locus and frequency response methods in Chapter 7 and pole placement and estimators in Chapter 8. The state-space material assumes no previous acquaintance with the phase plane or state space, and the necessary analysis is developed from the ground up. Some familiarity with simultaneous linear equations and matrix notation is expected, and a few unusual or more advanced topics such as eigenvalues, eigenvectors, and the Cayley-Hamilton theorem are presented in Appendix C. Chapter 9 introduces optimal quadratic-loss control: First the control by state feedback is presented and then the estimation of the state in the presence of system and measurement noise is developed, based on a recursive least-squares estimation derivation.

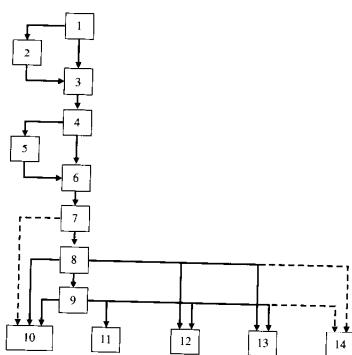
In Chapter 10 the nonlinear phenomenon of amplitude quantization and its effects on system error and system dynamic response are studied. Chapter 11 presents methods of analysis and design guidelines for the selection of the sampling period in a digital control system. It utilizes the design methods discussed in Chapters 7, 8, and 9, in examples illustrating the effects of sample rate. Chapter 12 introduces both nonparametric and parametric identification. Nonparametric methods are based on spectral estimation. Parametric methods are introduced by starting with deterministic least squares, introducing random errors, and completing the solution with an algorithm for maximum likelihood. Sub-space methods are also introduced for estimating the state matrices directly. Nonlinear control is the subject of Chapter 13, including examples of plant nonlinearities and methods for the analysis and design of controllers for nonlinear models. Simulation, stability analysis, and performance enhancement by nonlinear controllers and by adaptive designs are also included in Chapter 13. The chapter ends with a nonlinear design optimization alternative to the techniques presented in Chapter 9. The final chapter, 14, is a detailed design example of a digital servo for a disk drive head. Table P.1 shows the differences between the second and third editions of the book.

For purposes of organizing a course, Fig. P.1 shows the dependence of material in each chapter on previous chapters. By following the solid lines, the reader will have all the background required to understand the material in a particular chapter, even if the path omits some chapters. Furthermore, sections with a star (*) are optional and may be skipped with no loss of continuity. Chapters may also be skipped, as suggested by the dashed lines, if the reader is willing to take some details on faith; however, the basic ideas of the later chapters will be understood along these paths.

The first seven chapters (skipping or quickly reviewing Chapter 2) constitute a comfortable one-quarter course that would follow a course in continuous linear control using a text such as Franklin, Powell, and Emami-Naeini (1994). For a one-semester course, the first eight chapters represent a comfortable load. The

Table P.1 Comparison of the Table of Contents

<i>Chapter Title</i>	<i>3rd Edition Chapter Number</i>	<i>2nd Edition Chapter Number</i>
Introduction	1	1
Review of Continuous Control	2	-
Introductory Digital Control	3	-
Discrete Analysis and the z-Transform	4	2
Sampled Data Systems	5	3
Discrete Equivalents	6	4
Design Using Transform Methods	7	5
Design Using State-Space Methods	8	6
Multivariable and Optimal Control	9	9
Quantization Effects	10	7
Sample-Rate Selection	11	10
System Identification	12	8
Nonlinear Control	13	11
Application of Digital Control	14	12

Figure P.1

content of a second course has many possibilities. One possibility is to combine Chapters 8 and 9 with Chapter 10, 11, or 12. As can be seen from the figure, many options exist for including the material in the last five chapters. For a full-year course, all fourteen chapters can be covered. One of the changes made in

this third edition is that the optimal control material no longer depends on the least-squares development in the system identification chapter, thus allowing for more flexibility in the sequence of teaching.

It has been found at Stanford that it is very useful to supplement the lectures with laboratory work to enhance learning. A very satisfactory complement of laboratory equipment is a digital computer having an A/D and a D/A converter, an analog computer (or equivalent) with ten operational amplifiers, a digital storage scope, and a CAD package capable of performing the basic computations and plotting graphs. A description of the laboratory equipment and experiments at Stanford is described in Franklin and Powell, *Control System Magazine* (1989).

There are many important topics in control that we have not been able to include in this book. There is, for example, no discussion of mu analysis or design, linear matrix inequalities, or convex optimization. It is our expectation, however, that careful study of this book will provide the student engineer with a sound basis for design of sampled-data controls and a foundation for the study of these and many other advanced topics in this most exciting field.

As do all authors of technical works, we wish to acknowledge the vast array of contributors on whose work our own presentation is based. The list of references gives some indication of those to whom we are in debt. On a more personal level, we wish to express our appreciation to Profs. S. Boyd, A. Bryson, R. Cannon, S. Citron, J. How, and S. Rock for their valuable suggestions for the book and especially to our long-time colleague, Prof. Dan DeBra, for his careful reading and many spirited suggestions. We also wish to express our appreciation for many valuable suggestions to the current and former students of E207 and E208, for whom this book was written.

In addition, we want to thank the following people for their helpful reviews of the manuscript: Fred Bailey, University of Minnesota; John Fleming, Texas A&M University; J.B. Pearson, Rice University; William Perkins, University of Illinois; James Carroll, Clarkson University; Walter Higgins, Jr., Arizona State University; Stanley Johnson, Lehigh University; Thomas Kurfess, Georgia Institute of Technology; Stephen Phillips, Case Western Reserve University; Chris Rahm, Clemson University; T. Srinivasan, Wilkes University; Hal Tharp, University of Arizona; Russell Trahan, Jr., University of New Orleans; and Gary Young, Oklahoma State University.

We also wish to express our appreciation to Laura Cheu, Emilie Bauer, and all the staff at Addison-Wesley for their quality production of the book.

Stanford, California

G.F.F.
J.D.P.
M.L.W.

• 1 •

Introduction

A Perspective on Digital Control

The control of physical systems with a digital computer or microcontroller is becoming more and more common. Examples of electromechanical servomechanisms exist in aircraft, automobiles, mass-transit vehicles, oil refineries, and paper-making machines. Furthermore, many new digital control applications are being stimulated by microprocessor technology including control of various aspects of automobiles and household appliances. Among the advantages of digital approaches for control are the increased flexibility of the control programs and the decision-making or logic capability of digital systems, which can be combined with the dynamic control function to meet other system requirements. In addition, one hardware design can be used with many different software variations on a broad range of products, thus simplifying and reducing the design time.

Chapter Overview

In Section 1.1, you will learn about what a digital control system is, what the typical structure is, and what the basic elements are. The key issues are discussed and an overview of where those issues are discussed in the book is given. Section 1.2 discusses the design approaches used for digital control systems and provides an overview of where the different design approaches appear in the book. Computer Aided Control System Design (CACSD) issues and how the book's authors have chosen to handle those issues are discussed in Section 1.3.

1.1 Problem Definition

The digital controls studied in this book are for closed-loop (feedback) systems in which the dynamic response of the process being controlled is a major consideration in the design. A typical structure of the elementary type of system

that will occupy most of our attention is sketched schematically in Fig. 1.1. This figure will help to define our basic notation and to introduce several features that distinguish digital controls from those implemented with analog devices. The process to be controlled (sometimes referred to as the **plant**) may be any of the physical processes mentioned above whose satisfactory response requires control action.

By "satisfactory response" we mean that the plant output, $y(t)$, is to be forced to follow or track the reference input, $r(t)$, despite the presence of disturbance inputs to the plant [$w(t)$ in Fig. 1.1] and despite errors in the sensor [$v(t)$ in Fig. 1.1]. It is also essential that the tracking succeed even if the dynamics of the plant should change somewhat during the operation. The process of holding $y(t)$ close to $r(t)$, including the case where $r \equiv 0$, is referred to generally as the process of **regulation**. A system that has good regulation in the presence of disturbance signals is said to have good **disturbance rejection**. A system that has good regulation in the face of changes in the plant parameters is said to have low **sensitivity** to these parameters. A system that has both good disturbance rejection and low sensitivity we call **robust**.

The means by which robust regulation is to be accomplished is through the control inputs to the plant [$u(t)$ in Fig. 1.1]. It was discovered long ago¹ that a scheme of feedback wherein the plant output is measured (or sensed) and compared directly with the reference input has many advantages in the effort to design robust controls over systems that do not use such feedback. Much of our effort in later parts of this book will be devoted to illustrating this discovery and demonstrating how to exploit the advantages of feedback. However, the problem of control as discussed thus far is in no way restricted to digital control. For that we must consider the unique features of Fig. 1.1 introduced by the use of a digital device to generate the control action.

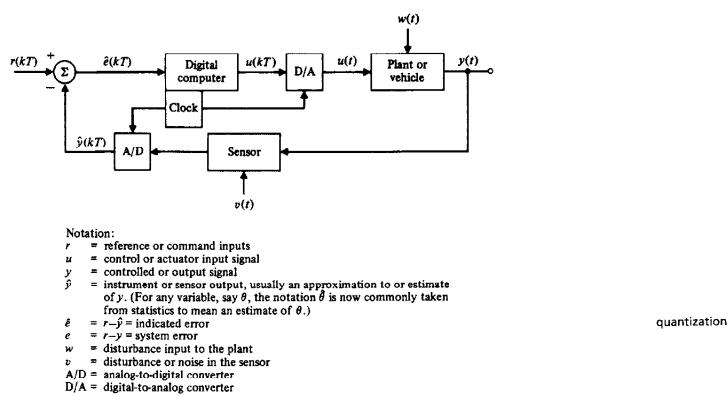
We consider first the action of the analog-to-digital (A/D) converter on a signal. This device acts on a physical variable, most commonly an electrical voltage, and converts it into a stream of numbers. In Fig. 1.1, the A/D converter acts on the sensor output and supplies numbers to the digital computer. It is common for the sensor output, $\hat{y}(t)$, to be sampled and to have the error formed in the computer. We need to know the times at which these numbers arrive if we are to analyze the dynamics of this system.

In this book we will make the assumption that all the numbers arrive with the same fixed period T , called the **sample period**. In practice, digital control systems sometimes have varying sample periods and/or different periods in different feedback paths. Usually there is a clock as part of the computer logic which supplies a pulse or **interrupt** every T seconds, and the A/D converter sends a number to the computer each time the interrupt arrives. An alternative implementation is simply to access the A/D upon completion of each cycle of the code execution, a scheme often referred to as **free running**. A further alternative is to use some other device to determine a sample, such as an encoder on an engine crankshaft that supplies a pulse to trigger a computer cycle. This scheme is referred to as **event-based sampling**. In the first case the sample period is precisely fixed; in the second case the sample period is essentially fixed by the length of the code, providing no logic branches are present that could vary the amount of code executed; in the third case, the sample period varies with the engine speed. Thus in Fig. 1.1 we identify the sequence of numbers into the computer as $\hat{e}(kT)$. We conclude from the periodic sampling action of the A/D converter that some of the signals in the digital control system, like $\hat{e}(kT)$, are variable only at discrete times. We call these variables **discrete signals** to distinguish them from variables like w and y , which change continuously in time. A system having both discrete and continuous signals is called a **sampled-data system**.

In addition to generating a discrete signal, however, the A/D converter also provides a **quantized** signal. By this we mean that the output of the A/D converter must be stored in digital logic composed of a finite number of digits. Most commonly, of course, the logic is based on binary digits (i.e., bits) composed

¹ See especially the book by Bode (1945).

Figure 1.1
Block diagram of a basic digital control system

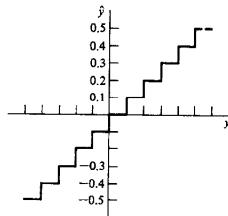


of 0's and 1's, but the essential feature is that the representation has a finite number of digits. A common situation is that the conversion of y to \hat{y} is done so that \hat{y} can be thought of as a number with a fixed number of places of accuracy. If we plot the values of y versus the resulting values of \hat{y} we can obtain a plot like that shown in Fig. 1.2. We would say that \hat{y} has been truncated to one decimal place, or that \hat{y} is *quantized* with a q of 0.1, since \hat{y} changes only in fixed quanta of, in this case, 0.1 units. (We will use q for quantum size, in general.) Note that quantization is a nonlinear function. A signal that is both discrete and quantized is called a **digital signal**. Not surprisingly, digital computers in this book process digital signals.

In a real sense the problems of analysis and design of *digital controls* are concerned with taking account of the effects of the sampling period T and the quantization size q . If both T and q are extremely small (sampling frequency 30 or more times the system bandwidth with a 16-bit word size), digital signals are nearly continuous, and continuous methods of analysis and design can be used. The resulting design could then be converted to the digital format for implementation in a computer by using the simple methods described in Chapter 3 or the **emulation** method described in Chapter 7. We will be interested in this text in gaining an understanding of the effects of all sample rates, fast and slow, and the effects of quantization for large and small word sizes. Many systems are originally conceived with fast sample rates, and the computer is specified and frozen early in the design cycle; however, as the designs evolve, more demands are placed on the system, and the only way to accommodate the increased computer load is to slow down the sample rate. Furthermore, for cost sensitive digital systems, the best design is the one with the lowest cost computer that will do the required job. That translates into being the computer with the slowest speed and the smallest word size. We will, however, treat the problems of varying T and q separately. We first consider q to be zero and study discrete and sampled-data (combined discrete and continuous) systems that are linear. In Chapter 10 we will analyze

emulation

Figure 1.2
Plot of output versus input characteristics of the A/D converter



aliasing

in more detail the source and the effects of quantization, and we will discuss in Chapters 7 and 11 specific effects of sample-rate selection.

Our approach to the design of digital controls is to assume a background in continuous systems and to relate the comparable digital problem to its continuous counterpart. We will develop the essential results, from the beginning, in the domain of discrete systems, but we will call upon previous experience in continuous-system analysis and in design to give alternative viewpoints and deeper understanding of the results. In order to make meaningful these references to a background in continuous-system design, we will review the concepts and define our notation in Chapter 2.

1.2 Overview of Design Approach

An overview of the path we plan to take toward the design of digital controls will be useful before we begin the specific details. As mentioned above, we place systems of interest in three categories according to the nature of the signals present. These are discrete systems, sampled-data systems, and digital systems.

In discrete systems all signals vary at discrete times only. We will analyze these in Chapter 4 and develop the z -transform of discrete signals and "pulse"-transfer functions for linear constant discrete systems. We also develop discrete transfer functions of continuous systems that are sampled, systems that are called sampled-data systems. We develop the equations and give examples using both transform methods and state-space descriptions. Having the discrete transfer functions, we consider the issue of the dynamic response of discrete systems.

A sampled-data system has both discrete and continuous signals, and often it is important to be able to compute the continuous time response. For example, with a slow sampling rate, there can be significant **ripple** between sample instants. Such situations are studied in Chapter 5. Here we are concerned with the question of data extrapolation to convert discrete signals as they might emerge from a digital computer into the continuous signals necessary for providing the input to one of the plants described above. This action typically occurs in conjunction with the D/A conversion. In addition to data extrapolation, we consider the analysis of sampled signals from the viewpoint of continuous analysis. For this purpose we introduce impulse modulation as a model of sampling, and we use Fourier analysis to give a clear picture for the ambiguity that can arise between continuous and discrete signals, also known as **aliasing**. The plain fact is that more than one continuous signal can result in exactly the same sample values. If a sinusoidal signal, y_1 , at frequency f_1 has the same samples as a sinusoid y_2 of a different frequency f_2 , y_1 is said to be an **alias** of y_2 . A corollary of aliasing is the **sampling theorem**, which specifies the conditions necessary if this ambiguity is to be removed and only one continuous signal allowed to correspond to a given set of samples.

digital filters

As a special case of discrete systems and as the basis for the emulation design method, we consider discrete equivalents to continuous systems, which is one aspect of the field of **digital filters**. Digital filters are discrete systems designed to process discrete signals in such a fashion that the digital device (a digital computer, for example) can be used to replace a continuous filter. Our treatment in Chapter 6 will concentrate on the use of discrete filtering techniques to find discrete equivalents of continuous-control compensator transfer functions. Again, both transform methods and state-space methods are developed to help understanding and computation of particular cases of interest.

modern control

Once we have developed the tools of analysis for discrete and sampled systems we can begin the design of feedback controls. Here we divide our techniques into two categories: **transform**² and **state-space**³ methods. In Chapter 7 we study the transform methods of the root locus and the frequency response as they can be used to design digital control systems. The use of state-space techniques for design is introduced in Chapter 8. For purposes of understanding the design method, we rely mainly on **pole placement**, a scheme for forcing the closed-loop poles to be in desirable locations. We discuss the selection of the desired pole locations and point out the advantages of using the optimal control methods covered in Chapter 9. Chapter 8 includes control design using feedback of all the "state variables" as well as methods for estimating the state variables that do not have sensors directly on them. In Chapter 9 the topic of **optimal control** is introduced, with emphasis on the steady-state solution for linear constant discrete systems with quadratic loss functions. The results are a valuable part of the designer's repertoire and are the only techniques presented here suitable for handling multivariable designs. A study of quantization effects in Chapter 10 introduces the idea of random signals in order to describe a method for treating the "average" effects of this important nonlinearity.

identification

The last four chapters cover more advanced topics that are essential for most complete designs. The first of these topics is sample rate selection, contained in Chapter 11. In our earlier analysis we develop methods for examining the effects of different sample rates, but in this chapter we consider for the first time the question of sample rate as a design parameter. In Chapter 12, we introduce **system identification**. Here the matter of model making is extended to the use of experimental data to verify and correct a theoretical model or to supply a dynamic description based only on input-output data. Only the most elementary of the concepts in this enormous field can be covered, of course. We present the method of least squares and some of the concepts of maximum likelihood.

In Chapter 13, an introduction to the most important issues and techniques for the analysis and design of nonlinear sampled-data systems is given. The

2 Named because they use the Laplace or Fourier transform to represent systems.

3 The state space is an extension of the space of displacement and velocity used in physics. Much that is called **modern control theory** uses differential equations in state-space form. We introduce this representation in Chapter 4 and use it extensively afterwards, especially in Chapters 8 and 9.

MATLAB**Digital Control Toolbox**

analysis methods treated are the describing function, equivalent linearization, and Lyapunov's second method of stability analysis. Design techniques described are the use of inverse nonlinearity, optimal control (especially time-optimal control), and adaptive control. Chapter 14 includes a case study of a disk-drive design, and treatment of both implementation and manufacturing issues is discussed.

1.3 Computer-Aided Design

As with any engineering design method, design of control systems requires many computations that are greatly facilitated by a good library of well-documented computer programs. In designing practical digital control systems, and especially in iterating through the methods many times to meet essential specifications, an interactive computer-aided control system design (CACSD) package with simple access to plotting graphics is crucial. Many commercial control system CACSD packages are available which satisfy that need. MATLAB® and Matrixx, being two very popular ones. Much of the discussion in the book assumes that a designer has access to one of the CACSD products. Specific MATLAB routines that can be used for performing calculations are indicated throughout the text and in some cases the full MATLAB command sequence is shown. All the graphical figures were developed using MATLAB and the files that created them are contained in the Digital Control Toolbox which is available on the Web at no charge. Files based on MATLAB v4 with Control System Toolbox v3, as well as files based on MATLAB v5 with Control System Toolbox v4 are available at [ftp.mathworks.com/pub/books/franklin/digital](http://mathworks.com/pub/books/franklin/digital). These figure files should be helpful in understanding the specifics on how to do a calculation and are an important augmentation to the book's examples. The MATLAB statements in the text are valid for MATLAB v5 and the Control System Toolbox v4. For those with older versions of MATLAB, Appendix F describes the adjustments that need to be made.

CACSD support for a designer is universal; however, it is essential that the designer is able to work out very simple problems by hand in order to have some idea about the reasonableness of the computer's answers. Having the knowledge of doing the calculations by hand is also critical for identifying trends that guide the designer; the computer can identify problems but the designer must make intelligent choices in guiding the refinement of the computer design.

1.4 Suggestions for Further Reading

Several histories of feedback control are readily available, including a *Scientific American Book* (1955), and the study of Mayr (1970). A good discussion of the historical developments of control is given by Dorf (1980) and by Fortmann and Hitz (1977), and many other references are cited by these authors for the

interested reader. One of the earliest published studies of control systems operating on discrete time data (sampled-data systems in our terminology) is given by Hurewicz in Chapter 5 of the book by James, Nichols, and Phillips (1947).

The ideas of tracking and robustness embody many elements of the objectives of control system design. The concept of tracking contains the requirements of system stability, good transient response, and good steady-state accuracy, all concepts fundamental to every control system. Robustness is a property essential to good performance in practical designs because real parameters are subject to change and because external, unwanted signals invade every system. Discussion of performance specifications of control systems is given in most books on introductory control, including Franklin, Powell, and Emami-Naeini (1994). We will study these matters in later chapters with particular reference to digital control design.

To obtain a firm understanding of dynamics, we suggest a comprehensive text by Cannon (1967). It is concerned with writing the equations of motion of physical systems in a form suitable for control studies.

1.5 Summary

- In a digital control system, the analog electronics used for compensation in a continuous system is replaced with a digital computer or microcontroller, an analog-to-digital (A/D) converter, and a digital-to-analog (D/A) converter.
- Design of a digital control system can be accomplished by transforming a continuous design, called emulation, or designing the digital system directly. Either method can be carried out using transform or state-space system description.
- The design of a digital control system includes determining the effect of the sample rate and selecting a rate that is sufficiently fast to meet all specifications.
- Most designs today are carried out using computer-based methods; however the designer needs to know the hand-based methods in order to intelligently guide the computer design as well as to have a sanity check on its results.

1.6 Problems

- 1.1** Suppose a radar search antenna at the San Francisco airport rotates at 6 rev/min, and data points corresponding to the position of flight 1081 are plotted on the controller's screen once per antenna revolution. Flight 1081 is traveling directly toward the airport at 540 mi/hr. A feedback control system is established through the controller who gives course corrections to the pilot. He wishes to do so each 9 mi of travel of the aircraft, and his instructions consist of course headings in integral degree values.

- (a) What is the sampling rate, in seconds, of the range signal plotted on the radar screen?
- (b) What is the sampling rate, in seconds, of the controller's instructions?
- (c) Identify the following signals as continuous, discrete, or digital:
 - i. the aircraft's range from the airport.
 - ii. the range data as plotted on the radar screen.
 - iii. the controller's instructions to the pilot.
 - iv. the pilot's actions on the aircraft control surfaces.
- (d) Is this a continuous, sampled-data, or digital control system?
- (e) Show that it is possible for the pilot of flight 1081 to fly a zigzag course which would show up as a straight line on the controller's screen. What is the (lowest) frequency of a sinusoidal zigzag course which will be hidden from the controller's radar?
- 1.2** If a signal varies between 0 and 10 volts (called the **dynamic range**) and it is required that the signal must be represented in the digital computer to the nearest 5 millivolts, that is, if the **resolution** must be 5 mv, determine how many bits the analog-to-digital converter must have.
- 1.3** Describe five digital control systems that you are familiar with. State what you think the advantages of the digital implementation are over an analog implementation.
- 1.4** Historically, house heating system thermostats were a bimetallic strip that would make or break the contact depending on temperature. Today, most thermostats are digital. Describe how you think they work and list some of their benefits.
- 1.5** Use MATLAB (obtain a copy of the Student Edition or use what's available to you) and plot y vs x for $x = 1$ to 10 where $y = x^2$. Label each axis and put a title on it.
- 1.6** Use MATLAB (obtain a copy of the Student Edition or use what's available to you) and make two plots (use MATLAB's subplot) of y vs x for $x = 1$ to 10. Put a plot of $y = x^2$ on the top of the page and $y = \sqrt{x}$ on the bottom.

• 2 •

Review of Continuous Control

A Perspective on the Review of Continuous Control

The purpose of this chapter is to provide a ready reference source of the material that you have already taken in a prerequisite course. The presentation is not sufficient to learn the material for the first time; rather, it is designed to state concisely the key relationships for your reference as you move to the new material in the ensuing chapters. For a more in-depth treatment of any of the topics, see an introductory control text such as *Feedback Control of Dynamic Systems*, by Franklin, Powell, and Emami-Naeini (1994).

Chapter Overview

The chapter reviews the topics normally covered in an introductory controls course: dynamic response, feedback properties, root locus design, frequency response design, and state-space design.

2.1 Dynamic Response

In control system design, it is important to be able to predict how well a trial design matches the desired performance. We do this by analyzing the equations of the system model. The equations can be solved using linear analysis approximations or simulated via numerical methods. Linear analysis allows the designer to examine quickly many candidate solutions in the course of design iterations and is, therefore, a valuable tool. Numerical simulation allows the designer to check the final design more precisely including all known characteristics and is discussed in Section 13.2. The discussion below focuses on linear analysis.

state-variable form

2.1.1 Differential Equations

Linear dynamic systems can be described by their differential equations. Many systems involve coupling between one part of a system and another. Any set of differential equations of any order can be transformed into a coupled set of first-order equations called the **state-variable form**. So a general way of expressing the dynamics of a linear system is

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} \quad (2.1)$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{J}\mathbf{u}. \quad (2.2)$$

where the column vector \mathbf{x} is called the **state** of the system and contains n elements for an n th-order system, \mathbf{u} is the $m \times 1$ input vector to the system, \mathbf{y} is the $p \times 1$ output vector, \mathbf{F} is an $n \times n$ system matrix, \mathbf{G} is an $n \times m$ input matrix, \mathbf{H} is a $p \times n$ output matrix, and \mathbf{J} is $p \times m$.¹ Until Chapter 9, all systems will have a scalar input, u , and a scalar output y ; in this case, \mathbf{G} is $n \times 1$, \mathbf{H} is $1 \times n$, and J is a scalar.

Using this system description, we see that the second-order differential equation

$$\ddot{y} + 2\zeta\omega_o\dot{y} + \omega_o^2 y = K_o u, \quad (2.3)$$

can be written in the state-variable form as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_o^2 & -2\zeta\omega_o \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ K_o \end{bmatrix} u \quad (2.4)$$

$$y = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

state

where the state

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$$

is the vector of variables necessary to describe the future behavior of the system, given the initial conditions of those variables.

2.1.2 Laplace Transforms and Transfer Functions

The analysis of linear systems is facilitated by use of the Laplace transform. The most important property of the Laplace transform (with zero initial conditions) is the transform of the derivative of a signal.

$$\mathcal{L}[\dot{f}(t)] = sF(s). \quad (2.5)$$

¹ It is also common to use **A**, **B**, **C**, **D** in place of **F**, **G**, **H**, **J** as MATLAB does throughout. We prefer to use **F**, **G**... for a continuous plant description, **A**, **B**... for compensation, and **Φ**, **Γ**... for the discrete plant description in order to delineate the various system equation usages.

This relation enables us to find easily the transfer function, $G(s)$, of a linear continuous system, given the differential equation of that system. So we see that Eq. (2.3) has the transform

$$(s^2 + 2\zeta\omega_o s + \omega_o^2)Y(s) = K_o U(s),$$

and, therefore, the transfer function, $G(s)$, is

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K_o}{s^2 + 2\zeta\omega_o s + \omega_o^2}.$$

CACSD software typically accepts the specification of a system in either the state-variable form or the transfer function form. The quantities specifying the state-variable form (Eqs. 2.1 and 2.2) are **F**, **G**, **H**, and **J**. This is referred to as the "ss" form in MATLAB. The transfer function is specified in a polynomial form ("tf") or a factored zero-pole-gain form ("zpk"). The transfer function in polynomial form is

$$G(s) = \frac{b_1 s^m + b_2 s^{m-1} + \cdots + b_{m+1}}{a_1 s^n + a_2 s^{n-1} + \cdots + a_{n+1}}, \quad (2.6)$$

where the MATLAB quantity specifying the numerator is a $1 \times (m+1)$ matrix of the coefficients, for example

$$\text{num} = [b_1 \ b_2 \ \dots \ b_{m+1}]$$

and the quantity specifying the denominator is a $1 \times (n+1)$ matrix, for example

$$\text{den} = [a_1 \ a_2 \ \dots \ a_{n+1}].$$

In MATLAB v5 with Control System Toolbox v4² the numerator and denominator are combined into one system specification with the statement

$$\text{sys} = \text{tf}(\text{num}, \text{den}).$$

In the zero-pole-gain form, the transfer function is written as the ratio of two factored polynomials,

$$G(s) = K \frac{\prod_{i=1}^m (s - z_i)}{\prod_{j=1}^n (s - p_j)}, \quad (2.7)$$

and the quantities specifying the transfer function are an $m \times 1$ matrix of the zeros, an $n \times 1$ matrix of the poles, and a scalar gain, for example

$$z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix}, \quad p = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}, \quad k = K$$

² All MATLAB statements in the text assume the use of MATLAB version 5 with Control System Toolbox version 4. See Appendix F if you have prior versions.

and can be combined into a system description by

`sys = zpk(z,p,k)`

For the equations of motion of a system with second-order or higher equations, the easiest way to find the transfer function is to use Eq. (2.5) and do the math by hand. If the equations of motion are in the state-variable form and the transfer function is desired, the Laplace transform of Eqs. (2.1) and (2.2) yields

$$G(s) = \frac{Y(s)}{U(s)} = H(sI - F)^{-1}G + J.$$

In MATLAB, given F , G , H , and J , one can find the polynomial transfer function form by the MATLAB script

`sys = tf(ss(F,G,H,J))`

or the zero-pole-gain form by

`sys = zpk(ss(F,G,H,J))`

Likewise, one can find a state-space realization of a transfer function by

`sys = ss(tf(num,den))`

2.1.3 Output Time Histories

Given the transfer function and the input, $u(t)$, with the transform $U(s)$, the output is the product,

$$Y(s) = G(s)U(s). \quad (2.8)$$

The transform of a time function can be found by use of a table (See Appendix B.2); however, typical inputs considered in control system design are steps

$$u(t) = R_o \mathbb{1}(t), \Rightarrow U(s) = \frac{R_o}{s},$$

ramps

$$u(t) = V_o t \mathbb{1}(t), \Rightarrow U(s) = \frac{V_o}{s^2},$$

parabolas

$$u(t) = \frac{A_o t^2}{2} \mathbb{1}(t), \Rightarrow U(s) = \frac{A_o}{s^3},$$

and sinusoids

$$u(t) = B \sin(\omega t) \mathbb{1}(t), \Rightarrow U(s) = \frac{B\omega}{s^2 + \omega^2},$$

Using Laplace transforms, the output $Y(s)$ from Eq. (2.8) is expanded into its elementary terms using partial fraction expansion, then the time function associated with each term is found by looking it up in the table. The total time function, $y(t)$, is the sum of these terms. In order to do the partial fraction expansion, it is necessary to factor the denominator. Typically, only the simplest cases are analyzed this way. Usually, system output time histories are solved numerically using computer based methods such as MATLAB's `step.m` for a step input or `lsim.m` for an arbitrary input time history. However, useful information about system behavior can be obtained by finding the individual factors without ever solving for the time history, a topic to be discussed later. These will be important because specifications for a control system are frequently given in terms of these time responses.

2.1.4 The Final Value Theorem

A key theorem involving the Laplace transform that is often used in control system analysis is the **final value theorem**. It states that, if the system is stable and has a final, constant value

$$\lim_{t \rightarrow \infty} x(t) = x_{\infty} = \lim_{s \rightarrow 0} sX(s). \quad (2.9)$$

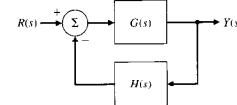
The theorem allows us to solve for that final value without solving for the system's entire response. This will be very useful when examining steady-state errors of control systems.

2.1.5 Block Diagrams

Manipulating block diagrams is useful in the study of feedback control systems. The most common and useful result is that the transfer function of the feedback system shown in Fig. 2.1 reduces to

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + H(s)G(s)}. \quad (2.10)$$

Figure 2.1
An elementary feedback system



2.1.6 Response versus Pole Locations

Given the transfer function of a linear system,

$$H(s) = \frac{b(s)}{a(s)},$$

the values of s such that $a(s) = 0$ will be places where $H(s)$ is infinity, and these values of s are called **poles** of $H(s)$. On the other hand, values of s such that $b(s) = 0$ are places where $H(s)$ is zero, and the corresponding s locations are called **zeros**. Since the Laplace transform of an impulse is unity, the **impulse response** is given by the time function corresponding to the transfer function. Each pole location in the s -plane can be identified with a particular type of response. In other words, the poles identify the classes of signals contained in the impulse response, as may be seen by a partial fraction expansion of $H(s)$. For a first order pole

$$H(s) = \frac{1}{s + \sigma}.$$

Table B.2, Entry 8, indicates that the impulse response will be an exponential function; that is

$$h(t) = e^{-\sigma t} 1(t).$$

poles

zeros
impulse response

stability

time constant

When $\sigma > 0$, the pole is located at $s < 0$, the exponential decays, and the system is said to be **stable**. Likewise, if $\sigma < 0$, the pole is to the right of the origin, the exponential grows with time and is referred to as **unstable**. Figure 2.2 shows a typical response and the **time constant**

$$\tau = \frac{1}{\sigma} \quad (2.11)$$

as the time when the response is $\frac{1}{e}$ times the initial value.

Complex poles can be described in terms of their real and imaginary parts, traditionally referred to as

$$s = -\sigma \pm j\omega_d.$$

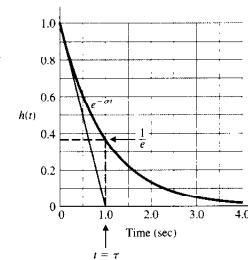
This means that a pole has a negative real part if σ is positive. Since complex poles always come in complex conjugate pairs for real polynomials, the denominator corresponding to a complex pair will be

$$a(s) = (s + \sigma - j\omega_d)(s + \sigma + j\omega_d) = (s + \sigma)^2 + \omega_d^2. \quad (2.12)$$

When finding the transfer function from differential equations, we typically write the result in the polynomial form

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \quad (2.13)$$

Figure 2.2
First-order system response

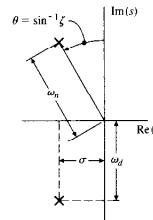


By expanding the form given by Eq. (2.12) and comparing with the coefficients of the denominator of $H(s)$ in Eq. (2.13), we find the correspondence between the parameters to be

$$\sigma = \zeta\omega_n \quad \text{and} \quad \omega_d = \omega_n\sqrt{1 - \zeta^2}, \quad (2.14)$$

where the parameter ζ is called the **damping ratio**, and ω_n is called the **undamped natural frequency**. The poles of this transfer function are located at a radius ω_n in the s -plane and at an angle $\theta = \sin^{-1}\zeta$, as shown in Fig. 2.3. Therefore, the damping ratio reflects the level of damping as a fraction of the critical damping value where the poles become real. In rectangular coordinates, the poles are at $s = -\sigma \pm j\omega_d$. When $\zeta = 0$ we have no damping, $\theta = 0$, and ω_d , the damped natural frequency, equals ω_n , the undamped natural frequency.

Figure 2.3
s-plane plot for a pair of complex poles



For the purpose of finding the time response corresponding to a complex transfer function from Table B.2, it is easiest to manipulate the $H(s)$ so that the complex poles fit the form of Eq. (2.12), because then the time response can be found directly from the table. The $H(s)$ from Eq. (2.13) can be written as

$$H(s) = \frac{\omega_n^2}{(s + \zeta\omega_n)^2 + \omega_n^2(1 - \zeta^2)},$$

therefore, from Entry 21 in Table B.2 and the definitions in Eq. (2.14), we see that the impulse response is

$$h(t) = \omega_n e^{-\sigma t} \sin(\omega_d t) u(t).$$

For $\omega_n = 3$ rad/sec and $\zeta = 0.2$, the impulse response time history could be obtained and plotted by the MATLAB statements:

```
Wn = 3
Ze = 0.2
num = Wn^2
den = [1 Ze*Wn Wn^2]
sys = tf(num,den)
impulse(sys)
```

step response

It is also interesting to examine the **step response** of $H(s)$, that is, the response of the system $H(s)$ to a unit step input $u = 1(t)$ where $U(s) = \frac{1}{s}$. The step response transform given by $Y(s) = H(s)U(s)$, contained in the tables in Entry 22, is

$$y(t) = 1 - e^{-\sigma t} \left(\cos \omega_d t + \frac{\sigma}{\omega_d} \sin \omega_d t \right), \quad (2.15)$$

where $\omega_d = \omega_n \sqrt{1 - \zeta^2}$ and $\sigma = \zeta \omega_n$. This could also be obtained by modifying the last line in the MATLAB description above for the impulse response to

`step(sys)`

Figure 2.4 is a plot of $y(t)$ for several values of ζ plotted with time normalized to the undamped natural frequency ω_n . Note that the actual frequency, ω_d , decreases slightly as the damping ratio increases. Also note that for very low damping the response is oscillatory, while for large damping (ζ near 1) the response shows no oscillation. A few step responses are sketched in Fig. 2.5 to show the effect of pole locations in the s -plane on the step responses. It is very useful for control designers to have the mental image of Fig. 2.5 committed to memory so that there is an instant understanding of how changes in pole locations influence the time response. The negative real part of the pole, σ , determines the decay rate of an exponential envelope that multiplies the sinusoid. Note that if σ

Figure 2.4
Step responses of second-order systems versus ζ

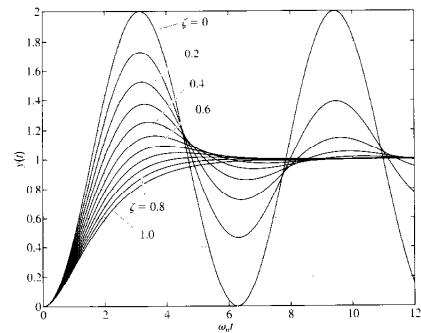
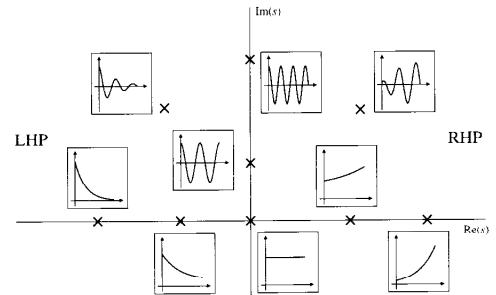


Figure 2.5
Time functions associated with points in the s -plane



is negative, the pole is in the right-half plane, the response will grow with time, and the system is said to be unstable. If $\sigma = 0$, the response neither grows nor decays, so stability is a matter of definition. If σ is positive, the natural response

decays and the system is said to be stable. Note that, as long as the damping is strictly positive, the system will eventually converge to the commanded value.

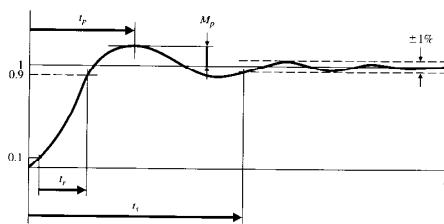
All these notions about the correspondence between pole locations and the time response pertained to the case of the step response of the system of Eq. (2.13), that is, a second-order system with no zeros. If there had been a zero, the effect would generally be an increased overshoot; the presence of an additional pole would generally cause the response to be slower. If there had been a zero in the right-half plane, the overshoot would be repressed and the response would likely go initially in the opposite direction to its final value. Nevertheless, the second-order system response is useful in guiding the designer during the iterations toward the final design, no matter how complex the system is.

2.1.7 Time-Domain Specifications

Specifications for a control system design often involve certain requirements associated with the time response of the system. The requirements for a step response are expressed in terms of the standard quantities illustrated in Fig. 2.6:

- t_r : The rise time t_r is the time it takes the system to reach the vicinity of its new set point.
- t_s : The settling time t_s is the time it takes the system transients to decay.
- M_p : The overshoot M_p is the maximum amount that the system overshoots its final value divided by its final value (and often expressed as a percentage).

Figure 2.6
Definition of rise time t_r , settling time t_s , and overshoot M_p



For a second-order system, the time responses of Fig. 2.4 yield information about the specifications that is too complex to be remembered unless approximated. The commonly used approximations for the second order case with no zeros are

$$t_r \approx \frac{1.8}{\omega_n} \quad (2.16)$$

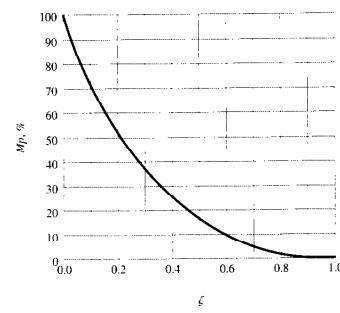
$$t_s \approx \frac{4.6}{\zeta \omega_n} = \frac{4.6}{\sigma} \quad (2.17)$$

$$M_p \approx e^{-\pi \zeta / \sqrt{1-\zeta^2}} \quad 0 \leq \zeta < 1 \quad (2.18)$$

The overshoot M_p is plotted in Fig. 2.7. Two frequently used values from this curve are $M_p = 16\%$ for $\zeta = 0.5$ and $M_p = 5\%$ for $\zeta = 0.7$.

Equations (2.16)–(2.18) characterize the transient response of a system having no finite zeros and two complex poles with undamped natural frequency ω_n , damping ratio ζ , and negative real part σ . In analysis and design, they are used to obtain a rough estimate of rise time, overshoot, and settling time for just about any system. It is important to keep in mind, however, that they are qualitative guides and not precise design formulas. They are meant to provide a starting point for the design iteration and the time response should always be checked after the control design is complete by an exact calculation, usually by numerical simulation, to verify whether the time specifications are actually met. If they have not been met, another iteration of the design is required. For example, if the rise

Figure 2.7
Plot of the peak overshoot M_p versus the damping ratio ζ for the second-order system



time of the system turns out to be longer than the specification, the target natural frequency would be increased and the design repeated.

2.2 Basic Properties of Feedback

An open-loop system described by the transfer function $G(s)$ can be improved by the addition of feedback including the dynamic compensation $D(s)$ as shown in Fig. 2.8. The feedback can be used to improve the stability, speed up the transient response, improve the steady-state error characteristics, provide disturbance rejection, and decrease the sensitivity to parameter variations.

2.2.1 Stability

The dynamic characteristics of the open-loop system are determined by the poles of $G(s)$ and $D(s)$, that is, the roots of the denominators of $G(s)$ and $D(s)$. Using Eq. (2.10), we can see that the transfer function of the closed-loop system in Fig. 2.8 is

$$\frac{Y(s)}{R(s)} = \frac{D(s)G(s)}{1 + D(s)G(s)} = T(s), \quad (2.19)$$

sometimes referred to as the **complementary sensitivity**. In this case, the dynamic characteristics and stability are determined by the poles of the closed-loop transfer function, that is, the roots of

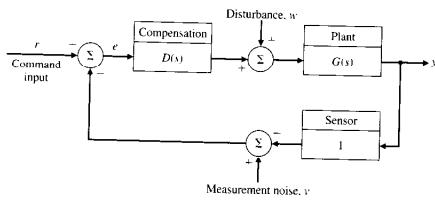
$$1 + D(s)G(s) = 0. \quad (2.20)$$

characteristic equation

This equation is called the **characteristic equation** and is very important in feedback control analysis and design. The roots of the characteristic equation represent the types of motion that will be exhibited by the feedback system. It is clear from Eq. (2.20) that they can be altered at will by the designer via the selection of $D(s)$.

Figure 2.8

A unity feedback system



system type

2.2.2 Steady-State Errors

The difference between the command input r (see Fig. 2.8) and the output y is called the system error, e . Using Eq. (2.10) for the case where the desired output is e , we find that

$$\frac{E(s)}{R(s)} = \frac{1}{1 + D(s)G(s)} = S(s), \quad (2.21)$$

sometimes referred to as the **sensitivity**. For the case where $r(t)$ is a step input and the system is stable, the Final Value Theorem tells us that

$$e_{ss} = \frac{1}{1 + K_p}$$

where

$$K_p = \lim_{s \rightarrow 0} D(s)G(s)$$

and is called the **position-error constant**. If $D(s)G(s)$ has a denominator that does not have s as a factor, K_p and e_{ss} are finite. This kind of system is referred to as **type 0**.

These results can also be seen qualitatively by examining Fig. 2.8. In order for y to be at some desired value ($= r$), the higher the forward loop gain of DG (defined to be K_p), the lower the value of the error, e . An integrator has the property that a zero steady input can produce a finite output, thus producing an infinite gain. Therefore, if there is an integrator in D or G , the steady-state gain will be ∞ and the error will be zero.

Continuing, we define the **velocity constant** as

$$K_v = \lim_{s \rightarrow 0} s D(s)G(s)$$

and the **acceleration constant** as

$$K_a = \lim_{s \rightarrow 0} s^2 D(s)G(s).$$

When K_v is finite, we call the system **type 1**; likewise, when K_a is finite, we call the system **type 2**. For the unity feedback case, it is convenient to categorize the error characteristics for command inputs consisting of steps, ramps, and parabolas. Table 2.1 summarizes the results.

Table 2.1

Errors versus system type for unity feedback

	Step	Ramp	Parabola
Type 0	$\frac{1}{(1-K_p)}$	∞	∞
Type 1	0	$\frac{1}{K_p}$	∞
Type 2	0	0	$\frac{1}{K_p}$

System type can also be defined with respect to the disturbance inputs w . The same ideas hold, but in this case the type is determined by the number of integrators in $D(s)$ only. Thus, if a system had a disturbance as shown in Fig. 2.8 which was constant, the steady-state error e_{ss} of the system would only be zero if $D(s)$ contained an integrator.

2.2.3 PID Control

Proportional, integral, and derivative (PID) control contains three terms. They are proportional control

$$u(t) = K e(t) \Rightarrow D(s) = K, \quad (2.22)$$

integral control

$$u(t) = \frac{K}{T_I} \int_0^t e(\eta) d\eta \Rightarrow D(s) = \frac{K}{T_I s}, \quad (2.23)$$

and derivative control

$$u(t) = K T_D \dot{e}(t) \Rightarrow D(s) = K T_D s. \quad (2.24)$$

T_I is called the integral (or reset) time, T_D the derivative time, and K the position feedback gain. Thus, the combined transfer function is

$$D(s) = \frac{u(s)}{e(s)} = K(1 + \frac{1}{T_I s} + T_D s). \quad (2.25)$$

Proportional feedback control can lead to reduced errors to disturbances but still has a small steady-state error. It can also increase the speed of response but typically at the cost of a larger transient overshoot. If the controller also includes a term proportional to the integral of the error, the error to a step can be eliminated as we saw in the previous section. However, there tends to be a further deterioration of the dynamic response. Finally, addition of a term proportional to the error derivative can add damping to the dynamic response. These three terms combined form the classical PID controller. It is widely used in the process industries and commercial controller hardware can be purchased where the user only need "tune" the gains on the three terms.

2.3 Root Locus

The root locus is a technique which shows how changes in the system's open-loop characteristics influence the closed-loop dynamic characteristics. This technique allows us to plot the locus of the closed-loop roots in the s -plane as an open-loop parameter varies, thus producing a root locus. The root locus method is most commonly used to study the effect of the loop gain (K in Eq. (2.25)); however, the method is general and can be used to study the effect of any parameter in $D(s)$

or $G(s)$. In fact, the method can be used to study the roots of any polynomial versus parameters in that polynomial.

A key attribute of the technique is that it allows you to study the **closed-loop roots** while only knowing the factors (poles and zeros) of the **open-loop system**.

2.3.1 Problem Definition

The first step in creating a root locus is to put the polynomials in the **root locus form**

$$1 + K \frac{b(s)}{a(s)} = 0. \quad (2.26)$$

Typically, $Kb(s)/a(s)$ is the open loop transfer function $D(s)G(s)$ of a feedback system; however, this need not be the case. The root locus is the set of values of s for which Eq. (2.26) holds for some real value of K . For the typical case, Eq. (2.26) represents the characteristic equation of the closed-loop system.

The purpose of the root locus is to show in a graphical form the general trend of the roots of a closed-loop system as we vary some parameter. Being able to do this by hand (1) gives the designer the ability to design simple systems without a computer. (2) helps the designer verify and understand computer-generated root loci, and (3) gives insight to the design process.

Equation (2.26) shows that, if K is real and positive, $b(s)/a(s)$ must be real and negative. In other words, if we arrange $b(s)/a(s)$ in polar form as magnitude and phase, then the phase of $b(s)/a(s)$ must be 180° . We thus define the root locus in terms of the phase condition as follows.

180° locus definition: The root locus of $b(s)/a(s)$ is the set of points in the s -plane where the phase of $b(s)/a(s)$ is 180° .

Since the phase is unchanged if an integral multiple of 360° is added, we can express the definition as³

$$\angle \frac{b(s)}{a(s)} = 180^\circ + l360^\circ,$$

where l is any integer. The significance of the definition is that, while it is very difficult to solve a high-order polynomial, computation of phase is relatively easy. When K is positive, we call this the **positive or 180° locus**. When K is real and negative, $b(s)/a(s)$ must be real and positive for s to be on the locus. Therefore, the phase of $b(s)/a(s)$ must be 0° . This case is called the **0° or negative locus**.

³ \angle refers to the phase of (\cdot) .

2.3.2 Root Locus Drawing Rules

The steps in drawing a 180° root locus follow from the basic phase definition. They are

STEP 1 On the s-plane, mark poles (roots of $a(s)$) by an \times and zeros (roots of $a(s)$) by a \circ . There will be a branch of the locus departing from every pole and a branch arriving at every zero.

STEP 2 Draw the locus on the real axis to the left of an odd number of real poles plus zeros.

STEP 3 Draw the asymptotes, centered at α and leaving at angles ϕ_i , where

$$\begin{aligned} n - m &= \text{number of asymptotes} \\ n &= \text{order of } a(s) \\ m &= \text{order of } b(s) \\ \alpha &= \frac{\sum P_i - \sum Z_i}{n - m} = \frac{-a_1 + b_1}{n - m}; \\ \phi_i &= \frac{180^\circ + (l - 1)360^\circ}{n - m}, \quad l = 1, 2, \dots, n - m. \end{aligned}$$

For $n - m > 0$, there will be a branch of the locus approaching each asymptote and departing to infinity. For $n - m < 0$, there will be a branch of the locus arriving from infinity along each asymptote.

STEP 4 Compute locus departure angles from the poles and arrival angles at the zeros where

$$\begin{aligned} q\phi_{dep} &= \sum \psi_i - \sum \phi_i - 180^\circ - l360^\circ \\ q\psi_{arr} &= \sum \phi_i - \sum \psi_i + 180^\circ + l360^\circ \end{aligned}$$

where q is the order of the pole or zero and l takes on q integer values so that the angles are between $\pm 180^\circ$. ψ_i is the angle of the line going from the i_{th} pole to the pole or zero whose angle of departure or arrival is being computed. Similarly, ϕ_i is the angle of the line from the i_{th} zero.

STEP 5 If further refinement is required at the stability boundary, assume $s_0 = j\omega_0$ and compute the point(s) where the locus crosses the imaginary axis for positive K .

STEP 6 For the case of multiple roots, two loci come together at 180° and break away at $\pm 90^\circ$. Three loci segments approach each other at angles of 120° and depart at angles rotated by 60°.

STEP 7 Complete the locus, using the facts developed in the previous steps and making reference to the illustrative loci for guidance. The loci branches start at poles and end at zeros or infinity.

STEP 8 Select the desired point on the locus that meets the specifications (s_o), then use the magnitude condition from Eq. (2.26) to find that the value of K associated with that point is

$$K = \frac{1}{|b(s_o)/a(s_o)|}.$$

When K is negative, the definition of the root locus in terms of the phase relationship is

0° locus definition: The root locus of $b(s)/a(s)$ is the set of points in the s-plane where the phase of $b(s)/a(s)$ is 0°.

For this case, the steps above are modified as follows

STEP 2 Draw the locus on the real axis to the left of an even number of real poles plus zeros.

STEP 3 The asymptotes depart at

$$\phi_i = \frac{(l - 1)360^\circ}{n - m}, \quad l = 1, 2, \dots, n - m.$$

STEP 4 The locus departure and arrival angles are modified to

$$\begin{aligned} q\phi_{dep} &= \sum \psi_i - \sum \phi_i - l360^\circ \\ q\psi_{arr} &= \sum \phi_i - \sum \psi_i + l360^\circ. \end{aligned}$$

Note that the 180° term has been removed.

◆ Example 2.1 Root Locus Sketch

Sketch the root locus versus K (positive and negative) for the case where the open-loop system is given by

$$G(s) = K \frac{s}{s^2 + 1}.$$

Solution. First let's do the 180° locus.

STEP 1: There is a zero at $s = 0$ and poles at $s = \pm j\omega$.

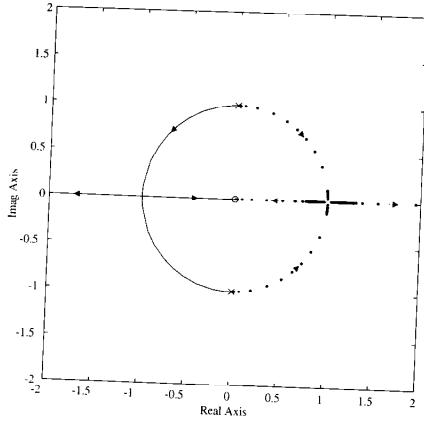
STEP 2: There is a locus on the entire negative real axis.

STEP 3: $n - m = 1$, therefore, there is one asymptote and it departs at 180°; that is, along the negative real axis.

STEP 4: The departure angle at the upper pole is calculated as

$$\phi_1 = 90^\circ - 90^\circ - 180^\circ = -180^\circ.$$

Figure 2.9
Example root locus sketch



thus, the locus departs from the upper pole horizontally to the left. The departure angle from the lower pole also turns out to be -180° and that branch of the locus also departs horizontally to the left.

We know that there is a locus segment along the entire negative real axis; however, we also know that there is a locus branch moving to the right and arriving at the zero, and that there is a branch departing to infinity at the far left. Therefore, the two branches from the poles must join the real axis at some point and split in opposite directions. It turns out that the two complex branches form a semi-circle as they approach the real axis. The solid lines in Fig. 2.9 show the sketch of this 180° locus.

For the 0° locus, there is a segment along the positive real axis and the angles of departure are both 0° . The result is shown in the figure by the dotted lines.

2.3.3 Computer-Aided Loci

The most common approach to machine computation of the root locus is to cast the problem as a polynomial in the form $a(s) + Kb(s) = 0$, and, for a sequence of values of K varying from near zero to a large value, solve the polynomial for

its n roots by any of many available numerical methods. A disadvantage of this method is that the resulting root locations are very unevenly distributed in the s -plane. For example, near a point of multiple roots, the sensitivity of the root locations to the parameter value is very great, and the roots just fly through such points, the plots appear to be irregular, and sometimes important features are missed. As a result, it is useful to have the root locus plotting rules in mind when interpreting computer plots. The polynomial is generally solved by transforming the problem to state-variable form and using the QR algorithm which solves for the eigenvalues of the closed-loop system matrix.

◆ Example 2.2 CACSD Root Locus

- Plot the root locus using MATLAB for the open-loop system shown in Fig. 2.8 with $G(s) = \frac{10}{s(s+2)}$, and $D(s) = K \frac{s+3}{s+10}$.
- Find the gain K associated with the point of maximum damping and plot the step response with that value of K .
- Reconcile the root locus plot with the hand plotting rules and compare the computer-based step response with the rules of thumb in Eqs. (2.16)-(2.18).

Solution.

- The MATLAB script following will generate the desired locus plot which is shown in Fig. 2.10(a).

```
numD = [1 3]; denD = [1 10];
numG = 10; denG = [1 2 0];
sys = tf(numD,denD)*tf(numG,denG);
k = 0:0.1:4;
rlocus(sys,k);
```

- The statement

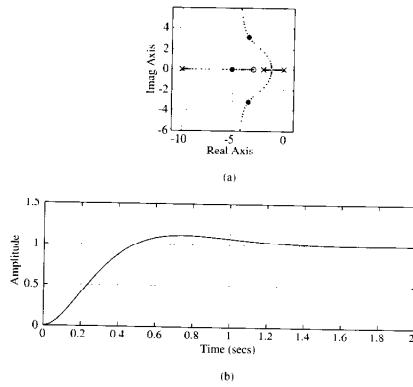
```
[K,p] = rlocfind(sys)
```

will place a cross-hair on the plot which can be moved with the mouse to the desired point on the locus in order to find the associated gain K and pole locations p . Given this value of K , (≈ 3.7) the script

```
K = 3.7;
sysCL = feedback(K*sys,1);
step(sysCL)
```

produces the desired step response shown in Fig. 2.10(b).

Figure 2.10
Example of CACSD for
(a) root locus and (b)
Step response



3. The root locus in Fig. 2.10(a) has locus segments to the left of odd numbers of poles and zeros (Step 2), has two asymptotes departing at $\pm 90^\circ$ and centered at

$$\alpha = \frac{-2 + 3 - 10}{2} = -4.5,$$

(Step 3); and has branches departing the real axis at the multiple roots between the two poles at $\pm 90^\circ$ (Step 6). The gain associated with the desired root at $s = -3.5 \pm j3.1$ can be determined from Step 8 by

$$K = \frac{(4.7)(3.5)(7.2)}{(3.2)(10)} = 3.7$$

where 4.7 is the distance from the root to the pole at $s = 0$; 3.5 is the distance to the pole at $s = -2$; 7.2 is the distance to the pole at $s = -10$; 3.2 is the distance to the zero at $s = -3$; and 10 is from the gain of $G(s)$.

The step response shown in Fig. 2.10(b) has $t_r \approx 0.4$ sec, $t_i \approx 1.4$ sec, and $M_p \approx 10\%$. The closed-loop roots with $K = 3.7$ are at $s = -5.1, -3.5 \pm j3.1$; thus, for the complex roots, $\zeta = 0.74$, $\omega_n = 4.7$ rad/sec, and $\sigma = 3.5$. The rules of thumb given in Section 2.1.7 suggest that

$$\begin{aligned} t_r &\approx \frac{1.8}{\omega_n} = 0.38 \text{ sec} \\ t_i &\approx \frac{4.6}{\sigma} = 1.3 \text{ sec} \\ M_p &\approx e^{-\pi\zeta/\sqrt{1-\zeta^2}} = 4\% \quad (\text{Fig. 2.7}), \end{aligned}$$

The rule-of-thumb values based on the second order system with no zeros predict a t_r and t_i that are a little slow due to the presence of the extra pole. The predicted M_p is substantially too small due to the presence of the zero at $s = 3$. \blacklozenge

2.4 Frequency Response Design

The response of a linear system to a sinusoidal input is referred to as the system's **frequency response**. A system described by

$$\frac{Y(s)}{U(s)} = G(s).$$

where the input $u(t)$ is a sine wave with an amplitude of U_o and frequency ω

$$u(t) = U_o \sin \omega_t,$$

which has a Laplace transform

$$U(s) = \frac{U_o \omega_1}{s^2 + \omega_1^2},$$

has a response with the transform,

$$Y(s) = G(s) \frac{U_o \omega_1}{s^2 + \omega_1^2}. \quad (2.27)$$

A partial fraction expansion of Eq. (2.27) will result in terms that represent the natural behavior of $G(s)$ and terms representing the sinusoidal input. Providing that all the natural behavior is stable, those terms will die out and the only terms left in the steady state are those due to the sinusoidal excitation, that is

$$Y(s) = \dots + \frac{\alpha_o}{s + j\omega_1} + \frac{\alpha_o^*}{s - j\omega_1} \quad (2.28)$$

where α_o and α_o^* would be found by performing the partial fraction expansion. After the natural transients have died out, the time response is

$$y(t) = 2|\alpha_o| \sin(\omega_1 t + \phi) = U_o A \sin(\omega_1 t + \phi)$$

where

$$A = |G(j\omega_1)| = |G(s)| \Big|_{s=j\omega_1}. \quad (2.29)$$

$$\phi = \tan^{-1} \frac{\text{Im}[G(j\omega_1)]}{\text{Re}[G(j\omega_1)]} = \angle G(j\omega_1). \quad (2.30)$$

So, a stable linear system $G(s)$ excited by a sinusoid will eventually exhibit a sinusoidal output y with the same frequency as the input u . The **magnitude**, $A(\omega)$ of y with respect to the input, $= |G(j\omega)|$ and the **phase**, $\phi(\omega)$, is $\angle G(j\omega)$; that is, the magnitude and phase of $G(s)$ is evaluated by letting s take on values along the imaginary ($j\omega$) axis. In addition to the response to a sinusoid, the analysis of the frequency response of a system is very useful in the determination of stability of a closed-loop system given its open-loop transfer function.

A key reason that the frequency response is so valuable is that the designer can determine the frequency response experimentally with no prior knowledge of the system's model or transfer function. The system is excited by a sinusoid with varying frequency and the magnitude $A(\omega)$ is obtained by a measurement of the ratio of the output sinusoid to input sinusoid in the steady-state at each frequency. The phase $\phi(\omega)$ is the measured difference in phase between input and output signals. As an example, frequency responses of the second-order system

$$G(s) = \frac{1}{(s/\omega_n)^2 + 2\xi(s/\omega_n) + 1}$$

are plotted for various values of ξ in Fig. 2.11 which is done by MATLAB with `bode(sys)`.

2.4.1 Specifications

bandwidth

A natural specification for system performance in terms of frequency response is the **bandwidth**, defined to be the maximum frequency at which the output of a system will track an input sinusoid in a satisfactory manner. By convention, for the system shown in Fig. 2.12 with a sinusoidal input r , the bandwidth is the frequency of r at which the output y is attenuated to a factor of 0.707 times the input (or down 3 dB). Figure 2.13 depicts the idea graphically for the frequency response of the *closed-loop* transfer function (defined to be $T(s)$ in Eq. (2.19))

$$\frac{Y(s)}{R(s)} - T(s) = \frac{KG(s)}{1 + KG(s)}.$$

The plot is typical of most closed-loop systems in that 1) the output follows the input, $|T| \approx 1$, at the lower excitation frequencies, and 2) the output ceases to follow the input, $|T| < 1$, at the higher excitation frequencies.

The bandwidth ω_{pw} is a measure of the speed of response and is therefore similar to the time-domain measure of rise time t_r or the s -plane measure of natural frequency ω_n . In fact, it can be seen from Fig. 2.11 that the bandwidth will be equal to the natural frequency when $\xi = 0.7$. For other damping ratios, the bandwidth is approximately equal to the natural frequency with an error typically less than a factor of 2.

The resonant peak M_r is a measure of the damping, as evidenced by Fig. 2.11 where the peak is approximately the value at $\omega = \omega_r$, which is $\frac{1}{2\xi}$ for $\xi < 0.5$.

Figure 2.11
(a) Magnitude and (b)
phase of a second-order
system

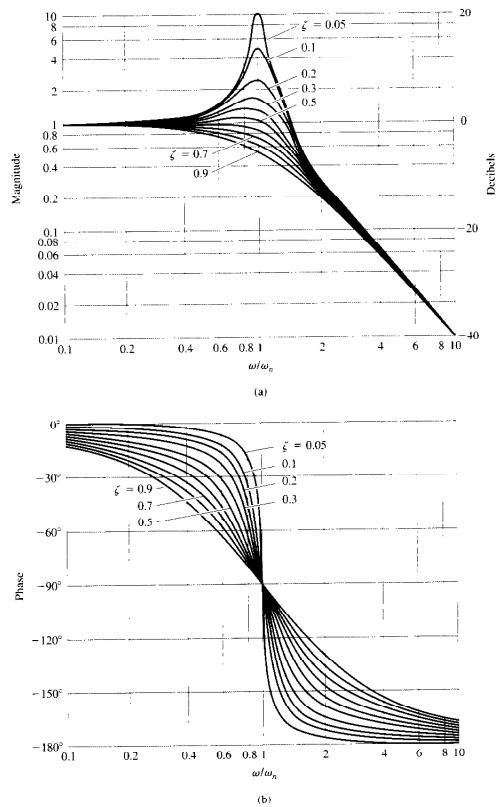


Figure 2.12
Simplified system definition

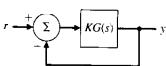
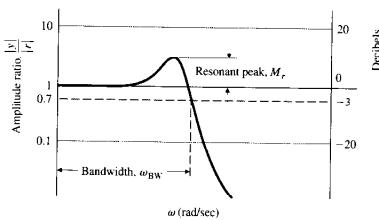


Figure 2.13
Definitions of bandwidth and resonant peak



2.4.2 Bode Plot Techniques

It is useful to be able to plot the frequency response of a system by hand in order to (a) design simple systems without the aid of a computer, (b) check computer-based results, and (c) understand the effect of compensation changes in design iterations. H. W. Bode developed plotting techniques in the 1930s that enabled quick hand plotting of the frequency response. His rules are:

STEP 1 Manipulate the transfer function into the **Bode form**

$$KG(j\omega) = K_o(j\omega)^n \frac{(j\omega\tau_1 + 1)(j\omega\tau_2 + 1)\dots}{(j\omega\tau_a + 1)(j\omega\tau_b + 1)\dots}$$

STEP 2 Determine the value of n for the $K_o(j\omega)^n$ term. Plot the low-frequency magnitude asymptote through the point K_o at $\omega = 1$ rad/sec with a slope of n (or $n \times 20$ dB per decade).

STEP 3 Determine the **break points** where $\omega = 1/\tau_i$. Complete the composite magnitude asymptotes by extending the low frequency asymptote until the first frequency break point, then stepping the slope by $+1$ or ± 2 , depending on whether the break point is from a first or second order term in the numerator or denominator, and continuing through all break points in ascending order.

STEP 4 Sketch in the approximate magnitude curve by increasing from the asymptote by a factor of $1.4 (+3 \text{ dB})$ at first order numerator breaks and decreasing it by a factor of $0.707 (-3 \text{ dB})$ at first order denominator breaks. At second order break points, sketch in the resonant peak (or valley) according to Fig. 2.11(a) using the relation $|G(j\omega)| = 1/(2\zeta)$ at the break.

STEP 5 Plot the low frequency asymptote of the phase curve, $\phi = n \times 90^\circ$.

STEP 6 As a guide, sketch in the approximate phase curve by changing the phase gradually over two decades by $\pm 90^\circ$ or $\pm 180^\circ$ at each break point in ascending order. For first order terms in the numerator, the gradual change of phase is $+90^\circ$; in the denominator, the change is -90° . For second order terms, the change is $\pm 180^\circ$.

STEP 7 Locate the asymptotes for each individual phase curve so that their phase change corresponds to the steps in the phase from the approximate curve indicated by Step 6. Sketch in each individual phase curve as indicated by Fig. 2.14 or Fig. 2.11(b).

STEP 8 Graphically add each phase curve. Use dividers if an accuracy of about $\pm 5^\circ$ is desired. If lesser accuracy is acceptable, the composite curve can be done by eye, keeping in mind that the curve will start at the lowest frequency asymptote and end on the highest frequency asymptote, and will approach the intermediate asymptotes to an extent that is determined by the proximity of the break points to each other.

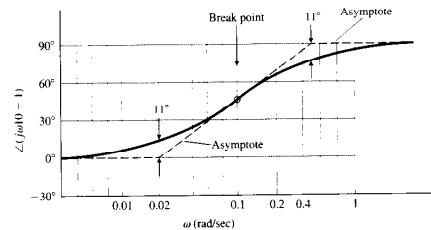
2.4.3 Steady-State Errors

Section 2.2.2 showed that the steady-state error of a feedback system decreases as the gain of the open loop transfer function increases. In plotting a composite magnitude curve, the low frequency asymptote is given by

$$KG(j\omega) = K_o(j\omega)^n. \quad (2.31)$$

Therefore, we see that the larger the value of the magnitude on the low-frequency asymptote, the lower the steady-state errors will be for the closed loop system. This idea is very useful in designing compensation.

Figure 2.14
Phase plot for $j\omega\tau + 1$; $\tau = 0.1$



For a system with $n = 0$, (a type 0 system) the low frequency asymptote is a constant and the gain K_0 of the open loop system is equal to the position error constant, K_p . For a system where $n = -1$, (a type 1 system) the low frequency asymptote has a slope of -1 and $K_0 = K_v$.

The easiest way of determining the value of K_v in a type 1 system is to read the magnitude of the low frequency asymptote at a frequency low enough to be well below any of the break points because $\frac{K}{\omega}$ equals the magnitude at these frequencies. In some cases, the lowest frequency break point will be below $\omega = 1$ rad/sec, therefore the asymptote can be extended to $\omega = 1$ rad/sec in order to read K_v directly.

2.4.4 Stability Margins

If the closed-loop transfer function of a system is known, the stability of the system can be determined by simply inspecting the denominator in factored form to observe whether the real parts are positive or negative. However, the closed-loop transfer function is not usually known; therefore, we would like to determine closed-loop stability by evaluating the frequency response of the *open-loop* transfer function $KG(j\omega)$ and then performing a simple test on that response. This can be done without a math model of the system by experimentally determining the open-loop frequency response.

We saw in Section 2.3.1 that all points on the root locus have the property that

$$|KG(s)| = 1 \quad \text{and} \quad \angle(KG(s)) = 180^\circ.$$

At the point of neutral stability we see that these root-locus conditions hold for $s = j\omega$, so

$$|KG(j\omega)| = 1 \quad \text{and} \quad \angle(KG(j\omega)) = 180^\circ. \quad (2.32)$$

Thus a Bode plot of a system that is neutrally stable (that is, with the value of K such that the closed-loop roots fall on the imaginary axis) will satisfy the conditions of Eq. (2.32). That means that the magnitude plot must equal 1 at the same frequency that the phase plot equals 180° . Typically, a system becomes less stable as the gain increases; therefore, we have the condition for stability

$$|KG(j\omega)| < 1 \quad \text{at} \quad \angle(KG(j\omega)) = -180^\circ. \quad (2.33)$$

This stability criterion holds for all systems where increasing gain leads to instability and $|KG(j\omega)|$ crosses the magnitude = 1 once, the most common situation. However, there are systems where an increasing gain can lead from instability to stability and in this case, the stability condition is

$$|KG(j\omega)| > 1 \quad \text{at} \quad \angle(KG(j\omega)) = -180^\circ.$$

gain margin
phase margin

crossover frequency

One way that will frequently resolve the ambiguity is to perform a rough sketch of the root locus to resolve the question of whether increasing gain leads to stability or instability. The rigorous way to resolve the ambiguity is to use the Nyquist stability criterion, which is reviewed in Section 7.5.1 for continuous systems.

Two quantities that measure the stability margin of a system are directly related to the stability criterion of Eq. (2.33): gain margin and phase margin. The **gain margin** (GM) is the factor by which the gain is less than the neutral stability value when the phase = 180° . The **phase margin** (PM) is the amount by which the phase of $G(s)$ exceeds -180° when $|KG(j\omega)| = 1$. The two margins are alternate ways of measuring the degree to which the stability conditions of Eq. (2.33) are met.

The phase margin is generally related to the damping of a system. For a second-order system, the approximation that

$$\zeta \cong \frac{\text{PM}}{100}$$

is commonly used. Therefore, if it were known that a system was to be designed using frequency response methods, it would make sense to specify the speed of response of the system in terms of a required bandwidth and the stability of the system in terms of a required phase margin.

2.4.5 Bode's Gain-Phase Relationship

One of Bode's important contributions is his theorem that states

For any minimum phase system (that is, one with no time delays, RHP zeros or poles), the phase of $G(j\omega)$ is uniquely related to the integral of the magnitude of $G(j\omega)$.

When the slope of $|G(j\omega)|$ versus ω on a log-log scale persists at a constant value for nearly a decade of frequency, the relationship is particularly simple

$$\angle G(j\omega) \cong n \times 90^\circ. \quad (2.34)$$

where n is the slope of $|G(j\omega)|$ in units of decade of amplitude per decade of frequency.

Equation (2.34) is used as a guide to infer stability from $|G(j\omega)|$ alone. When $|KG(j\omega)| = 1$, the **crossover frequency**, the phase

$$\begin{aligned} \angle G(j\omega) &\cong -90^\circ && \text{if } n = -1, \\ \angle G(j\omega) &\cong -180^\circ && \text{if } n = -2. \end{aligned}$$

For stability we want $\angle G(j\omega) > -180^\circ$ for a PM > 0. Therefore we adjust the $|KG(j\omega)|$ curve so that it has a slope of -1 at the crossover frequency. If the slope is -1 for a decade above and below the crossover frequency, the PM would be approximately 90° ; however, to ensure a reasonable PM, it is usually only

necessary to insist on a -1 slope (-20 dB per decade) persisting for a decade in frequency that is centered at the crossover frequency.

2.4.6 Design

One of the very useful aspects of frequency-response design is the ease with which we can evaluate the effects of gain changes. In fact, we can determine the PM for any value of K without redrawing the magnitude or phase information. We need only indicate on the figure where $|KG(j\omega)| = 1$ for selected trial values of K since varying K has the effect of sliding the magnitude plot up or down.

◆ Example 2.3 Frequency-Response Design

For a plant given by

$$G(s) = K \frac{1}{s(s+1)^2}$$

determine the PM and GM for the system with unity feedback and (a) $K = 1$, (b) determine if the system is stable if $K = 5$, and (c) find what value of K is required to achieve a PM of (i) 45° , and (ii) 70° .

Solution.

Using the hand plotting rules, we see that the low frequency asymptote has a slope of -1 and goes thru magnitude $= 1$ at $\omega = 1$ rad/sec. The slope changes to -3 at the break point ($\omega = 1$). We can then sketch in the actual magnitude curve, noting (STEP 4 in Section 2.4.2) that it will go below the asymptote intersection by -6 dB because there is a slope change of -2 at that break point. The curve is sketched in Fig. 2.15. The phase curve starts out at -90° and drops to -270° along the asymptote as sketched in the figure according to STEP 7.

Using MATLAB⁴, the statements

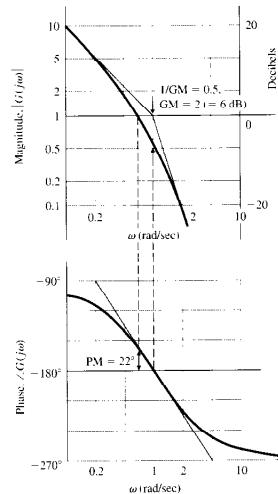
```
num = 1; den = [1 2 1 0];
sys = tf(num,den);
bode(sys);
```

will also create the plots of magnitude and phase for this example. The curves are drawn in Fig. 2.15 showing the PM and GM for $K = 1$ and the same curves are drawn in Fig. 2.16 showing the PM's for $K = 5, 0.5$, & 0.2 .

- (a) We can read the PM from Fig. 2.15 to be 22° .
- (b) Fig. 2.16 shows that the system is unstable for $K=5$.
- (c) (i) PM = 45° when $K = 0.5$, and (ii) PM = 70° when $K = 0.2$

⁴ All MATLAB statements in the text assume the use of MATLAB version 5 with Control System Toolbox version 4. See Appendix F if you have prior versions.

Figure 2.15
Magnitude and phase plots with PM and GM for $1/s(\omega + 1)^2$



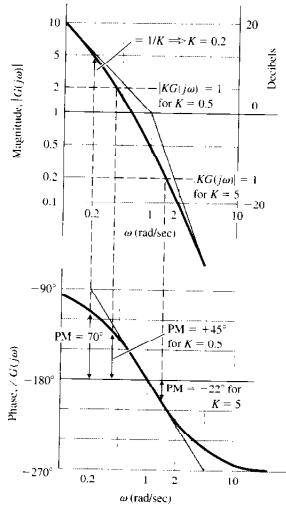
2.5 Compensation

If the plant dynamics are of such a nature that a satisfactory design cannot be achieved by adjustment of the feedback gain alone, then some modification or compensation must be made in the feedback to achieve the desired specifications. Typically, it takes the form

$$D(s) = K \frac{s+z}{s+p}$$

where it is called **lead compensation** if $z < p$ and **lag compensation** if $z > p$. Lead compensation approximates the addition of a derivative control term and tends to increase the bandwidth and the speed of response while decreasing the overshoot. Lag compensation approximates integral control and tends to improve the steady state error.

lead compensation
lag compensation

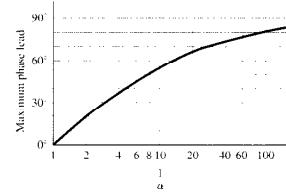
Figure 2.16
PM versus K for
 $1/(s+1)^2$ 

The design of lead compensation typically entails placing the zero z at a frequency that is lower than the magnitude = 1 crossover frequency and the pole higher than the crossover frequency. Lead compensation provides an increased magnitude slope and an increased phase in the interval between these two break points; the maximum being halfway between the two break points on a logarithmic scale. The maximum phase increase is

$$\delta\phi = \sin^{-1} \frac{1-\alpha}{1+\alpha} \quad \text{where} \quad \alpha = \frac{z}{p}$$

and is plotted versus α in Fig. 2.17.

The design of lag compensation typically entails placing both break points well below the crossover frequency. Lag compensation decreases the phase in the vicinity of the two break points; therefore, z should be well below the crossover

Figure 2.17
Maximum phase increase for lead compensation

frequency in order to prevent the compensation from degrading the PM and the system stability. The primary role of lag compensation is to increase the gain (magnitude of the frequency response) at the low frequencies. As we saw in Section 2.4.3, this will decrease the steady-state error.

2.6 State-Space Design

We saw in Section 2.1.1 that equations of motion could be written in the state-variable form of Eqs. (2.1) and (2.2). The **state-space** design approach utilizes this way of describing the plant and arrives directly with feedback controllers (compensation) without the need to determine transforms. Advantages of state-space design are especially apparent when the system to be controlled has more than one control input or more than one sensed output, called multivariable or multi input-multi output (MIMO). However, we will review only the single input-single output (SISO) case here. For readers not familiar with state-space design, the material in this section is not required for comprehension of the remainder of the book. The basic ideas of state-space design are covered in detail in Chapter 8 for the discrete case and that chapter does not require that the reader be knowledgeable about continuous state-space design. Chapter 9 extends state-space design for discrete systems to optimal control design for the multivariable case.

One of the attractive features of the state-space design method is that it consists of a sequence of independent steps. The first step, discussed in Section 2.6.1, is to determine the control. The purpose of the control law is to allow us to design a set of pole locations for the closed-loop system that will correspond to satisfactory dynamic response in terms of rise-time, overshoot, or other measures of transient response.

The second step—necessary if the full state is not available—is to design an **estimator** (sometimes called an **observer**), which computes an estimate of the

estimator
observer

entire state vector when provided with the measurements of the system indicated by Eq. (2.2). We review estimator design in Section 2.6.2.

The third step consists of combining the control law and the estimator. Figure 2.18 shows how the control law and the estimator fit together and how the combination takes the place of what we have been previously referring to as compensation.

The fourth and final step is to introduce the reference input in such a way that the plant output will track external commands with acceptable rise-time, overshoot and settling time values. Figure 2.18 shows the command input r introduced in the same relative position as was done with the transform design methods; however, in Section 2.6.4 we will show how to introduce the reference input in a different way that results in a better system response.

2.6.1 Control Law

The first step is to find the control law as feedback of a linear combination of all the state variables—that is,

$$u = -\mathbf{K}\mathbf{x} = -[K_1 \ K_2 \ \dots \ K_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \quad (2.35)$$

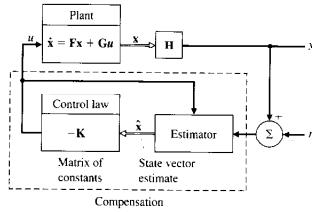
controllability

We assume for design purposes that all the elements of the state vector are at our disposal, an infrequent situation for an actual system, but an expedient assumption for the time being.

For an n -th-order system there will be n feedback gains K_1, \dots, K_n , and since there are n roots (or poles) of the system, it is possible that there are enough degrees of freedom to select arbitrarily any desired root location by choosing the proper values of K_j .

pole placement

Figure 2.18
Schematic diagram of state-space design elements



Substituting the feedback law, Eq. (2.35) into the system described by Eq. (2.1) yields

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u}. \quad (2.36)$$

The characteristic equation of this closed-loop system is

$$\det[\mathbf{I} - (\mathbf{F} - \mathbf{G}\mathbf{K})] = 0. \quad (2.37)$$

When evaluated, this yields an n -th-order polynomial in s containing the gains K_1, \dots, K_n . The control-law design then consists of picking the gains \mathbf{K} so that the roots of Eq. (2.37) are in desirable locations. Selection of desirable locations for the roots is an inexact science and may require some iteration by the designer. For now, we will assume that the desired locations are known, say

$$s = s_1, s_2, \dots, s_n.$$

Then the corresponding desired (control) characteristic equation is

$$\alpha_c(s) = (s - s_1)(s - s_2) \cdots (s - s_n) = 0. \quad (2.38)$$

Hence the required elements of \mathbf{K} are obtained by matching coefficients in Eq. (2.37) and Eq. (2.38). This forces the system characteristic equation to be identical with the desired characteristic equation and the closed-loop poles to be placed at the desired locations.

The calculation of \mathbf{K} can be done providing the system is **controllable**. Systems that are not controllable have certain modes or subsystems that are unaffected by the control. This usually means that parts of the system are physically disconnected from the input. Although there is a mathematical test for controllability, it is good practice to insist on the stronger condition that the control input be as strongly coupled to the modes of interest as possible.

It is theoretically possible to solve for \mathbf{K} by hand with Eq. (2.37) and Eq. (2.38). In practice, this is almost never done. Ackermann's formula for this calculation has been implemented in MATLAB as the function `acker.m` and can be used for the design of SISO systems with a small (≤ 10) number of state variables. For more complex cases a more reliable formula is available, implemented in MATLAB as the function `place`. A modest limitation on `place.m` is that none of the desired closed-loop poles are repeated; i.e., that the poles are *distinct*, a requirement that does not apply to `acker`. Both `acker` and `place` require inputs consisting of the system description matrices, \mathbf{F} and \mathbf{G} , and a vector, \mathbf{p} , of n desired pole locations. Their output is the feedback gain \mathbf{K} . Thus the MATLAB statements

$$\mathbf{K} = \text{acker}(\mathbf{F}, \mathbf{G}, \mathbf{p}) \quad \text{or} \quad \mathbf{K} = \text{place}(\mathbf{F}, \mathbf{G}, \mathbf{P})$$

will provide the desired value of \mathbf{K} . When selecting the desired root locations, it is always useful to keep in mind that the control effort required is related to how far the open-loop poles are moved by the feedback. Furthermore, when a zero is near a pole, the system may be nearly uncontrollable and moving such a pole may

LQR
optimal control

require large control effort. Therefore, a pole placement philosophy that aims to fix only the undesirable aspects of the open-loop response and avoids either large increases in bandwidth or efforts to move poles that are near zeros will typically allow smaller gains and thus smaller control actuators.

A method called the **linear quadratic regulator (LQR)** specifically addresses the issue of achieving a balance between good system response and the control effort required. The method consists of calculating the gain K that minimizes a **cost function**

$$\mathcal{J} = \int_0^\infty [\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}] dt \quad (2.39)$$

where \mathbf{Q} is an $n \times n$ state weighting matrix, \mathbf{R} is an $m \times m$ control weighting matrix, and m is the number of control inputs in a multi-input system. For the SISO systems that we are primarily concerned with here, $m = 1$ and \mathbf{R} is a scalar R . The weights \mathbf{Q} and R are picked by the designer by trial-and-error in order to arrive at the desired balance between state errors $\mathbf{x}^T \mathbf{x}$ and control usage u^2 , thus avoiding the necessity of picking desired pole locations that do not use excessive control. Generally, \mathbf{Q} is a diagonal matrix with a weighting factor on one or more of the state-vector elements while $R = 1$. It is perfectly acceptable to only weight one element; in fact, the element representing the system output is often the only element weighted. Rules of thumb that help in picking the weights are that (1) the bandwidth of the system increases as overall values in \mathbf{Q} increase, (2) the damping increases as the term in \mathbf{Q} that weights the velocity type state elements increase, and (3) a portion of a system can be made faster by increasing the weights on the state elements representing that portion of the system. The MATLAB statement

$K = lqr(F, G, Q, R)$

solves for the K that minimizes the cost, \mathcal{J} .

◆ Example 2.4 State-Space Control Design

For a plant given by

$$G(s) = \frac{1}{s^2},$$

(a) Find the feedback gain matrix K that yields closed-loop roots with $\omega_n = 3$ rad/sec and $\zeta = 0.8$.

(b) Investigate the roots obtained by using LQR with

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 100 & 0 \\ 0 & 0 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 100 & 0 \\ 0 & 5 \end{bmatrix}$$

and $R = 1$.

Solution. The state-variable description of $G(s)$ is (Eq. (2.4)) with $\omega_n = 0$, $\zeta = 0$, and $K_v = 1$)

$$\begin{aligned} F &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, & G &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ H &= \begin{bmatrix} 1 & 0 \end{bmatrix}, & J &= 0. \end{aligned}$$

(a) The desired characteristic equation is

$$\sigma_c(s) = s^2 + 2\zeta\omega_n s + \omega_n^2 = 0.$$

Therefore, the MATLAB script

```
F = [0 1; 0 0]
G = [0; 1]
Wn = 3
Ze = 0.8
p = roots([1 - 2*Wn*Ze - Wn^2])
K = acker(F,G,p)
```

provides the answer

$$K = [9 - 4.8].$$

(b) The scripts

```
Q = [1 0; 0 0], [100 0; 0 0], and [100 0; 0 5]
R = 1
K = lqr(F,G,Q,R)
p = eig(F - G*K)
[Wn, Ze] = damp(p)
```

compute feedback gains of

$$K = [1 - 1.4], [10 - 4.5], \text{ and } [10 - 5].$$

which produces natural frequencies of

$$\omega_n = 1, 3.2, \text{ and } 3.2 \text{ rad/sec}$$

and damping of

$$\zeta = 0.71, 0.71, \text{ and } 0.79.$$

For this simple example, use of `acker` is the easier way to find K ; however, in more complex systems with higher order roots, it is easier to use `lqr` rather than iterate on the best value for all the roots.

2.6.2 Estimator Design

For a system described by Eqs. (2.1) and (2.2), an estimate, \hat{x} , of the full state vector, x , can be obtained based on measurements of the output, y , from

$$\dot{\hat{x}} = F\hat{x} + Gu + L(y - H\hat{x}). \quad (2.40)$$

Here L is a proportional gain defined as

$$L = [l_1, l_2, \dots, l_n]^T, \quad (2.41)$$

and is chosen to achieve satisfactory error characteristics. The dynamics of the error can be obtained by subtracting the estimate (Eq. 2.40) from the state (Eq. 2.1), to get the error equation

$$\dot{\tilde{x}} = (F - LH)\tilde{x}. \quad (2.42)$$

The characteristic equation of the error is now given by

$$\det[sI - (F - LH)] = 0. \quad (2.43)$$

We choose L so that $F - LH$ has stable and reasonably fast eigenvalues, so \tilde{x} decays to zero, independent of the control $u(t)$ and the initial conditions. This means that $\tilde{x}(t)$ will converge to $x(t)$.

Errors in the model of the plant (F, G, H) cause additional errors to the state estimate from those predicted by Eq. (2.42). However, L can typically be chosen so that the error is kept acceptably small. It is important to emphasize that the nature of the plant and the estimator are quite different. The plant is a physical system such as a chemical process or servomechanism whereas the estimator is usually an electronic unit computing the estimated state according to Eq. (2.40).

The selection of L is approached in exactly the same fashion as K is selected in the control-law design. If we specify the desired location of the estimator error poles as

$$\beta_i = \beta_1, \beta_2, \dots, \beta_n,$$

then the desired estimator characteristic equation is

$$\alpha_e(s) \stackrel{\Delta}{=} (s - \beta_1)(s - \beta_2) \cdots (s - \beta_n). \quad (2.44)$$

We can solve for L by comparing coefficients in Eq. (2.43) and Eq. (2.44).

As in the control case, this is almost never done by hand. Rather, the functions `acker.m` and `place.m` are used, but with a slight twist. The transpose of Eq. (2.43) is

$$\det[sI - (F^T - H^T L^T)] = 0. \quad (2.45)$$

and we now see that this is identical in form to Eq. (2.37) where K and L^T play the same role. Therefore, we compute L to achieve estimator poles at the desired location, p , by typing in MATLAB

$$L = \text{acker}(F^T, H^T, p) \quad \text{or} \quad L = \text{place}(F^T, H^T, p)$$

observability

optimal estimation

where F^T is indicated in MATLAB as F' , etc.

There will be a unique solution for L for a SISO system provided that the system is **observable**. Roughly speaking, observability refers to our ability to deduce information about all the modes of the system by monitoring only the sensed outputs. Unobservability results when some mode or subsystem has no effect on the output.

The selection of the estimator poles that determine L are generally chosen to be a factor of 2 to 6 faster than the controller poles. This ensures a faster decay of the estimator errors compared with the desired dynamics, thus causing the controller poles to dominate the total system response. If sensor noise is particularly large, it sometimes makes sense for the estimator poles to be slower than two times the controller poles, which would yield a system with lower bandwidth and more noise smoothing. On the other hand, the penalty in making the estimator poles too fast is that the system becomes more noise sensitive.

The tradeoff between fast and slow estimator roots can also be made using results from **optimal estimation theory**. First, let's consider that there is a random input affecting the plant, called **process noise**, w , that enters Eq. (2.1) as

$$\dot{x} = Fx + Gu + G_1w, \quad (2.46)$$

and a random **sensor noise**, v entering Eq. (2.1) as

$$y = Hx + v. \quad (2.47)$$

The estimator error equation with these additional inputs is

$$\dot{\tilde{x}} = (F - LH)\tilde{x} + G_1w - Lv. \quad (2.48)$$

In Eq. (2.48) the sensor noise is multiplied by L and the process noise is not. If L is very small, then the effect of sensor noise is removed but the estimator's dynamic response will be "slow", so the error will not reject effects of w very well. The state of a low-gain estimator will not track uncertain plant inputs very well or plants with modeling errors. On the other hand, if L is "large", then the estimator response will be fast and the disturbance or process noise will be rejected, but the sensor noise, multiplied by L , results in large errors. Clearly, a balance between these two effects is required.

It turns out that the optimal solution to this balance can be found as a function of the process noise intensity, R_w , and the sensor noise intensity, R_v , both of which are scalars for the SISO case under consideration. Since the only quantity affecting the result is the ratio R_w/R_v , it makes sense to let $R_v = 1$ and vary R_w only. An important advantage of using the optimal solution is that only one parameter, R_w , needs to be varied by the designer rather than picking n estimator poles for an n^{th} -order system. The solution is calculated by MATLAB as

$$L = \text{kalman(sys,Rw,Rv)}.$$

2.6.3 Compensation: Combined Control and Estimation

We now put all this together, ignoring for the time being the effect of a command input, r . If we take the control law (Eq. 2.35), combine it with the estimator (Eq. 2.40), and implement the control law using the estimated state elements, the design is complete and the equations describing the result are

$$\begin{aligned}\dot{\hat{x}} &= (\mathbf{F} - \mathbf{GK} - \mathbf{LH})\hat{x} + \mathbf{Ly}, \\ u &= -\mathbf{K}\hat{x}.\end{aligned}\quad (2.49)$$

These equations describe what we previously called compensation: that is, the control, u , is calculated given the measured output, y . Figure 2.19 shows schematically how the pieces fit together. The roots of this new closed-loop system can be shown to consist of the chosen roots of the controller plus the chosen roots of the estimator that have been designed in separate procedures in Sections 2.6.1 and 2.6.2. The poles and zeros of the compensator alone could be obtained by examining the system described by Eq. (2.49); however, that step need not be carried out unless the designer is curious how the compensation from this approach compares with compensation obtained using a transform based design method.

2.6.4 Reference Input

One obvious way to introduce a command input is to subtract y from r in exactly the same way it has been done for the transform design methods discussed previously. This scheme is shown schematically in Fig. 2.20(b). Using this approach, a step command in r enters directly into the estimator, thus causing an estimation error that decays with the estimator dynamic characteristics in addition to the response corresponding to the control poles.

An alternative approach consists of entering the command r directly into the plant and estimator in an identical fashion as shown in Fig. 2.20(a). Since the

Figure 2.19
Estimator and controller mechanism

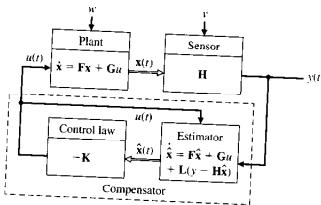
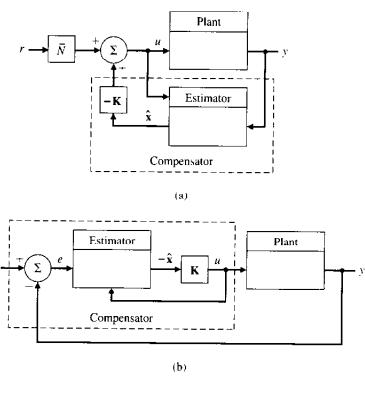


Figure 2.20
Possible locations for introducing the reference input:
(a) compensation in the feedback path,
(b) compensation in the feedforward path



command creates a step in u that affects the plant and estimator in an identical fashion, both respond identically, and no estimator error is induced. Therefore, there are no estimator error characteristics in the response and the total response consists of controller characteristics only. This approach is usually superior.

The feedforward gain, \bar{N} , can be computed so that no steady-state error exists. Its value is based on computing the steady-state value of the control, u_{ss} , and the steady-state values of the state, x_{ss} , that result in no steady-state error, e . The result is

$$\bar{N} = N_u + \mathbf{K}N_x \quad (2.50)$$

where

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{H} & \mathbf{J} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

2.6.5 Integral Control

In many cases, it is difficult to obtain an accurate value for the plant gain, in part because plants are typically nonlinear and the plant model is a linearization at a particular point. Therefore, the value of \bar{N} will not be accurate and steady-state errors will result even though the model is sufficiently accurate for good

feedback control design. The solution is to incorporate an integral control term in the feedback similar to the integral control discussed in Section 2.2.3.

Integral control is accomplished using state space design by augmenting the state vector with the desired integral x_I . It obeys the differential equation

$$\dot{x}_I = \mathbf{H}\mathbf{x} - r \quad (= e).$$

Thus

$$x_I = \int^t e dt.$$

This equation is augmented to the state equations (Eq. 2.1) and they become

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{x}_I \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{H} \\ 0 & \mathbf{F} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ x_I \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{G} \end{bmatrix} u - \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} r. \quad (2.51)$$

The feedback law is

$$u = -[\mathbf{K}_1 \quad \mathbf{K}_0] \begin{bmatrix} x_I \\ \mathbf{x} \end{bmatrix},$$

or simply

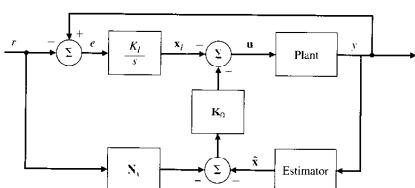
$$u = -\mathbf{K} \begin{bmatrix} x_I \\ \mathbf{x} \end{bmatrix}.$$

With this revised definition of the system, the design techniques from Section 2.6.1 can be applied in a similar fashion. The elements of \mathbf{K} obtained are implemented as shown in Fig. 2.21.

2.7 Summary

- System dynamics can be represented by a state-space description, Eq. (2.1), or by a transfer function, Eqs. (2.6) or (2.7).

Figure 2.21
Integral control structure



- The key property of the Laplace transform that allows solution of differential equations is Eq. (2.5)

$$\mathcal{L}\{\dot{f}(t)\} = s F(s).$$

- A system's output can be determined by the inverse Laplace transform for very simple cases or, more often the case, by numerical methods such as impulse.m, step.m, or lsim.m in MATLAB.

- If a system's output is described by $X(s)$ and is stable, the Final Value Theorem states that

$$\lim_{t \rightarrow \infty} x(t) = x_{ss} = \lim_{s \rightarrow 0} s X(s).$$

- One can associate certain time response behavior with pole locations in the s -plane as summarized in Fig. 2.5.

- Control system specifications are usually defined in terms of the rise time t_r , settling time t_s , and overshoot M_p which are defined by Eqs. (2.16)–(2.18).

- For an open loop system given by $D(s)G(s)$, the closed loop system as defined by Fig. 2.8 is given by Eq. (2.19)

$$\frac{Y(s)}{R(s)} = \frac{D(s)G(s)}{1 + D(s)G(s)} = T(s).$$

- The basic types of feedback are proportional, integral, and derivative, and are defined by Eqs. (2.22)–(2.24).

- The root locus is a method to sketch the location of the closed-loop roots of a system vs. some parameter of interest, usually the feedback gain. It is based on phase considerations which can easily be determined graphically by hand, and are therefore very useful in checking computer based results.

- The frequency response of the open-loop transfer function of a system can be easily analyzed to determine the stability of the related closed-loop system. The open-loop transfer function can be determined experimentally or analytically.

- Design of control systems using the state space approach is carried out by specifying the desired closed-loop root location, called pole-placement, or by selecting weighting matrices in a cost function, called optimal or LQR control. Either method tends to reduce the design iterations required over root locus or frequency response design, especially for higher order systems and those with multiple inputs and/or outputs.

- State space design requires that all elements of the state vector are available for the control; therefore, they must be measured directly or estimated using measurements of a portion of the state vector. Pole placement or optimal methods can also be used to arrive at the best estimator for this purpose.

2.8 Problems

- 2.1 Design feedback with lead compensation for the open-loop system

$$G(s) = \frac{10}{s^2}.$$

The rise time should be 1 sec or less and the overshoot should be less than 10%.

- 2.2 Design feedback with lead compensation for the open-loop system

$$G(s) = \frac{5}{s^2}.$$

The bandwidth should be faster than 1 rad/sec and the phase margin should be better than 50°.

- 2.3 For the open-loop system

$$G(s) = \frac{2}{s^3},$$

- (a) design feedback assuming you have access to all the state elements. Ensure that there are closed-loop system poles at $s = -3 \pm 3j$.
- (b) Design an estimator for the system so that it has poles at $s = -6 \pm 6j$.
- (c) Find the transfer function of the complete controller consisting of the control from part (a) and the estimator from part (b).

- 2.4 For the open-loop system

$$G(s) = \frac{1}{s(s+4)},$$

- (a) design feedback assuming you have access to all the state elements. Ensure that there are closed-loop system poles that provide a natural frequency of $\omega_n = 3$ rad/sec with $\zeta = 0.5$.
- (b) Design an estimator for the system so that it has poles that provide a natural frequency of $\omega_n = 6$ rad/sec with $\zeta = 0.5$.
- (c) Find the transfer function of the complete controller consisting of the control from part (a) and the estimator from part (b).

- 2.5 Can you stabilize the system

$$G(s) = \frac{1}{s^2(s^2 + 25)}$$

with a single lead compensation? If you can, do it. If you can't, show why not.

- 2.6 For the open-loop system

$$G(s) = \frac{1}{s^2(s^2 + 25)}.$$

- (a) design feedback assuming you have access to all the state elements. Place the closed-loop system poles at $s = -1 \pm 1j, -0.5 \pm 5j$.
- (b) Design an estimator for the system so that it has poles at $s = -2 + 2j, -2 \pm 8j$.
- (c) Find the transfer function of the complete controller consisting of the control from part (a) and the estimator from part (b).

- 2.7 Consider a pendulum with control torque T_c and disturbance torque T_d whose differential equation is

$$\ddot{\theta} + 4\theta = T_c + T_d.$$

Assume there is a potentiometer at the pin that measures the output angle θ , that is, $y = \theta$.

- (a) Design a lead compensation using a root locus that provides for an $M_p < 10\%$ and a rise time, $t_r < 1$ sec.
- (b) Add an integral term to your controller so that there is no steady-state error in the presence of a constant disturbance, T_d , and modify the compensation so that the specifications are still met.

- 2.8 Consider a pendulum with control torque T_c and disturbance torque T_d whose differential equation is

$$\ddot{\theta} + 4\theta = T_c + T_d.$$

Assume there is a potentiometer at the pin that measures the output angle θ , that is, $y = \theta$.

- (a) Design a lead compensation using frequency response that provides for a PM > 50° and a bandwidth, $\omega_{RH} > 1$ rad/sec.
- (b) Add an integral term to your controller so that there is no steady-state error in the presence of a constant disturbance, T_d , and modify the compensation so that the specifications are still met.

- 2.9 Consider a pendulum with control torque T_c and disturbance torque T_d whose differential equation is

$$\ddot{\theta} + 4\theta = T_c + T_d.$$

Assume there is a potentiometer at the pin that measures the output angle θ , that is, $y = \theta$.

- (a) Taking the state vector to be

$$\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix},$$

write the system equations in state form. Give values for the matrices F, G, H .

- (b) Show, using state-variable methods, that the characteristic equation of the model is $s^2 + 4 = 0$.

- (c) Write the estimator equations for

$$\begin{bmatrix} \hat{\theta} \\ \dot{\hat{\theta}} \end{bmatrix}.$$

Pick estimator gains $[L_1, L_2]^T$ to place both roots of the estimator-error characteristic equation at $s = -10$.

- (d) Using state feedback of the estimated state variables θ and $\dot{\theta}$, derive a control law to place the closed-loop control poles at $s = -2 \pm 2j$.

- (e) Draw a block diagram of the system, that is, estimator, plant, and control law.

- (f) Demonstrate the performance of the system by plotting the step response to a reference command on (i) θ , and (ii) T_d .

- (g) Design a controller with an integral term and demonstrate its performance to the step inputs as in (f).

2.10 For the open-loop system

$$G(s) = \frac{3}{s^2 + 2s - 3},$$

determine

- (a) the final value to a unit step input.
 (b) Answer (a) for the case where

$$G(s) = \frac{3}{s^2 + 2s + 3}.$$

2.11 For the open-loop system

$$G(s) = \frac{3}{s^2 + 2s - 3},$$

assume there is a feedback with a proportional gain, K , and sketch a locus of the closed-loop roots vs. K . What is the minimum value of K to achieve a stable system?

2.12 For the open-loop system

$$G(s) = \frac{1}{s^2(s^2 + 2s + 100)},$$

use a single lead compensation in the feedback to achieve as fast a response as possible, keeping the damping of the resonant mode better than $\zeta = 0.05$.

2.13 Sketch the locus of roots vs. the parameter b for

$$s^2 + bs + b + 1 = 0.$$

2.14 Sketch the root locus with respect to K for the open-loop system

$$G(s) = \frac{K(s+3)}{s(s+2)(s+1)^2}.$$

After completing the hand sketch, verify your result using MATLAB.

2.15 Sketch the root locus with respect to K for the open-loop system

$$G(s) = \frac{K(s+2)}{s^4}.$$

After completing the hand sketch, verify your result using MATLAB.

2.16 Sketch the root locus with respect to K for the open-loop system

$$G(s) = \frac{K(s+2)}{s^2}.$$

After completing the hand sketch, verify your result using MATLAB.

2.17 Sketch the root locus with respect to K for the open-loop system

$$G(s) = \frac{K(s+1)}{s(s+2)(s^2 + 25)}.$$

After completing the hand sketch, verify your result using MATLAB.

- 2.18** Sketch a Bode plot for the open-loop system

$$G(s) = \frac{(s+0.1)}{s(s+1)(s^2 + 2s + 100)}.$$

After completing the hand sketch, verify your result using MATLAB. With unity feedback, would the system be stable?

- 2.19** Sketch a Bode plot for the open-loop system

$$G(s) = \frac{100(s+1)}{s^2(s+10)}.$$

After completing the hand sketch, verify your result using MATLAB. With unity feedback, would the system be stable? What is the PM?

- 2.20** Sketch a Bode plot for the open-loop system

$$G(s) = \frac{5000(s+1)}{s^2(s+10)(s+50)}.$$

After completing the hand sketch, verify your result using MATLAB. With unity feedback, would the system be stable? If not, how would you stabilize it?

• 3 •

Introductory Digital Control

A Perspective on Introductory Digital Control

The continuous controllers you have studied so far are built using analog electronics such as resistors, capacitors, and operational amplifiers. However, most control systems today use digital computers (usually microprocessors or microcontrollers) with the necessary input/output hardware to implement the controllers. The intent of this chapter is to show the very basic ideas of designing control laws that will be implemented in a digital computer. Unlike analog electronics, digital computers cannot integrate. Therefore, in order to solve a differential equation in a computer, the equation must be approximated by reducing it to an algebraic equation involving sums and products only. These approximation techniques are often referred to as **numerical integration**. This chapter shows a simple way to make these approximations as an introduction to digital control. Later chapters expand on various improvements to these approximations, show how to analyze them, and show that digital compensation may also be carried out directly without resorting to these approximations. In the final analysis, we will see that direct digital design provides the designer with the most accurate method and the most flexibility in selection of the sample rate.

From the material in this chapter, you should be able to design and implement a digital control system. The system would be expected to give adequate performance if the sample rate is at least 30 times faster than the bandwidth of the system.

Chapter Overview

In Section 3.1, you will learn how to approximate a continuous $D(s)$ with a set of difference equations, a design method sometimes referred to as **emulation**. Section 3.1 is sufficient to enable you to approximate a continuous feedback controller in a digital control system. Section 3.2 shows the basic effect of

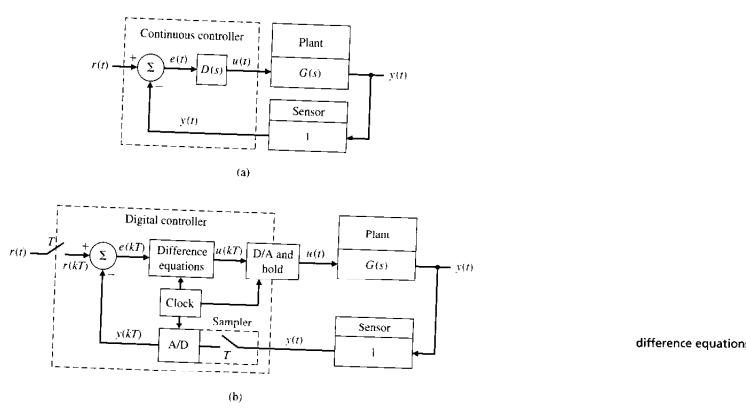
sampling on the performance of the system and a simple way to analyze that effect. Section 3.3 shows how to convert a continuous PID control law to the digital form.

3.1 Digitization

Figure 3.1(a) shows the topology of the typical continuous system. The computation of the error signal, e , and the dynamic compensation, $D(s)$, can all be accomplished in a digital computer as shown in Fig. 3.1(b). The fundamental differences between the two implementations are that the digital system operates on *samples* of the sensed plant output rather than on the continuous signal and that the dynamics represented by $D(s)$ are implemented by algebraic recursive equations called **difference equations**.

We consider first the action of the analog-to-digital (A/D) converter on a signal. This device acts on a physical variable, most commonly an electrical voltage, and converts it into a binary number that usually consists of 10 or 12 bits. A binary number with 10 bits can take on $2^{10} = 1024$ values; therefore, an A/D converter with 10 bits has a resolution of 0.1%. The conversion from the analog signal $y(t)$ occurs repetitively at instants of time that are T seconds

Figure 3.1
Basic control-system block diagrams.
(a) continuous system,
(b) with a digital computer



apart. T is called the **sample period** and $1/T$ is the **sample rate** in cycles per second or Hz (also sometimes given in radians/second or $2\pi/T$). The sampled signal is $y(kT)$ where k can take on any integer value. It is often written simply as $y(k)$. We call this type of variable a **discrete signal** to distinguish it from a continuous variable like $y(t)$, which changes continuously in time. We make the assumption here that the sample period is fixed; however, it may vary depending on the implementation as discussed in Section 1.1.

There also may be a sampler and A/D converter for the input command, $r(t)$, producing the discrete $r(kT)$ from which the sensed $y(kT)$ would be subtracted to arrive at the discrete error signal, $e(kT)$. The differential equation of the continuous compensation is approximated by a difference equation which is the discrete approximation to the differential equation and can be made to duplicate the dynamic behavior of a $D(s)$ if the sample period is short enough. The result of the difference equation is a discrete $u(kT)$ at each sample instant. This signal is converted to a continuous $u(t)$ by the D/A and hold. The D/A converts the binary number to an analog voltage, and a **zero-order hold** (ZOH) maintains that same voltage throughout the sample period. The resulting $u(t)$ is then applied to the actuator in precisely the same manner as the continuous implementation.

One particularly simple way to make a digital computer approximate the real time solution of differential equations is to use Euler's method. It follows from the definition of a derivative that

$$\dot{x} = \lim_{\delta t \rightarrow 0} \frac{\delta x}{\delta t} \quad (3.1)$$

where δx is the change in x over a time interval δt . Even if δt is not quite equal to zero, this relationship will be approximately true, and

$$\dot{x}(k) \cong \frac{x(k+1) - x(k)}{T} \quad (3.2)$$

where

$T = t_{k+1} - t_k$ (the sample interval in seconds),

$t_k = kT$ (for a constant sample interval),

k is an integer,

$x(k)$ is the value of x at t_k , and

$x(k+1)$ is the value of x at t_{k+1} .

This approximation¹ can be used in place of all derivatives that appear in the controller differential equations to arrive at a set of equations that can be solved by a digital computer. These equations are called **difference equations** and are solved repetitively with time steps of length T . For systems having bandwidths

¹ This particular version is called the **forward rectangular rule**. See Problem 3.2 for the **backward rectangular** version.

of a few Hertz, sample rates are often on the order of 100 Hz, so that sample periods are on the order of 10 msec and errors from the approximation can be quite small.

◆ Example 3.1 Difference Equations Using Euler's Method

Using Euler's method, find the difference equations to be programmed into the control computer in Fig. 3.1(b) for the case where the $D(s)$ in Fig. 3.1(a) is

$$D(s) = \frac{U(s)}{E(s)} = K_o \frac{s+a}{s+b}. \quad (3.3)$$

Solution. First find the differential equation that corresponds to $D(s)$. After cross multiplying Eq. (3.3) to obtain

$$(s+b)U(s) = K_o(s+a)E(s),$$

we can see by inspection that the corresponding differential equation is

$$\dot{u} + bu = K_o(\dot{e} + ae). \quad (3.4)$$

Using Euler's method to approximate Eq. (3.4) according to Eq. (3.2), we get the approximating difference equation

$$\frac{u(k+1) - u(k)}{T} + bu(k) = K_o \left[\frac{e(k+1) - e(k)}{T} + ae(k) \right]. \quad (3.5)$$

Rearranging Eq. (3.5) puts the difference equation in the desired form

$$u(k+1) = u(k) + T \left[-bu(k) + K_o \left(\frac{e(k+1) - e(k)}{T} + ae(k) \right) \right]. \quad (3.6)$$

Equation (3.6) shows how to compute the new value of the control, $u(k+1)$, given the past value of the control, $u(k)$, and the new and past values of the error signal, $e(k+1)$ and $e(k)$. For computational efficiency, it is convenient to re-arrange Eq. (3.6) to

$$u(k+1) = (1 - bT)u(k) + K_o(aT - 1)e(k) + K_oae(k+1). \quad (3.7)$$

In principle, the difference equation is evaluated initially with $k = 0$, then $k = 1, 2, 3, \dots$. However, there is usually no requirement that values for all times be saved in memory. Therefore, the computer need only have variables defined for the current and past values for this first-order difference equation. The instructions to the computer to implement the feedback loop in Fig. 3.1(b) with the difference equation from Eq. (3.7) would call for a continual looping through the code in Table 3.1. Note in the table that the calculations have been arranged so as to minimize the computations required between the reading of the A/D and the writing to the D/A, thus keeping the computation delay to a minimum.

Table 3.1

Real Time Controller implementation

```
x = 0 (initialization of past values for first loop through)
Define constants:
α₁ = 1 - bT
α₂ = K_o(aT - 1)
READ A/D to obtain y and r
e = r - y
u = x + K_o e
OUTPUT u to D/A and ZOH
now compute x for the next loop through
x = α₁ u + α₂ e
go back to READ when T seconds have elapsed since last READ
```

The sample rate required depends on the closed-loop bandwidth of the system. Generally, sample rates should be faster than 30 times the bandwidth in order to assure that the digital controller can be made to closely match the performance of the continuous controller. Discrete design methods described in later chapters will show how to achieve this performance and the consequences of sampling even slower if that is required for the computer being used. However, when using the techniques presented in this chapter, a good match to the continuous controller is obtained when the sample rate is greater than approximately 30 times the bandwidth.

◆ Example 3.2 Lead Compensation Using a Digital Computer

Find digital controllers to implement the lead compensation

$$D(s) = 70 \frac{s+2}{s+10} \quad (3.8)$$

for the plant

$$G(s) = \frac{1}{s(s+1)}$$

using sample rates of 20 Hz and 40 Hz. Implement the control equations on an experimental laboratory facility like that depicted in Fig. 3.1, that is, one that includes a microprocessor for the control equations, a ZOH, and analog electronics for the plant. Compute the theoretical step response of the continuous system and compare that with the experimentally determined step response of the digitally controlled system.

Solution. Comparing the compensation transfer function in Eq. (3.8) with Eq. (3.3) shows that the values of the parameters in Eq. (3.6) are $a = 2$, $b = 10$, and $K_o = 70$. For a sample rate of 20 Hz, $T = 0.05$ sec and Eq. (3.6) can be simplified to

$$u(k+1) = 0.5u(k) + 70[e(k+1) - 0.9e(k)].$$

For a sample rate of 40 Hz, $T = 0.025$ sec and Eq. (3.6) simplifies to

$$u(k+1) = 0.75u(k) + 70[e(k+1) - 0.95e(k)].$$

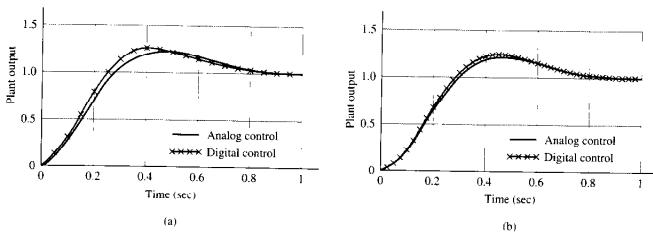
The statements in MATLAB² to compute the continuous step response is

```
numD = 70*[1 2]; denD = [1 10];
numG = 1; denG = [1 1 0];
sys1 = tf(numD,denD)*tf(numG,denG);
sysCL = feedback(sys1,1);
step(sysCL);
```

Figure 3.2 shows the step response of the two digital controllers compared to the continuous step response. Note that the 40 Hz sample rate (about 30 \times bandwidth) behaves essentially like the continuous case, whereas the 20 Hz sample rate (about 15 \times bandwidth) has a detectable increased overshoot signifying some degradation in the damping. The damping would degrade further if the sample rate were made any slower.

The MATLAB file that created Fig. 3.2 (fig32.m) computed the digital responses as well as the continuous response. You will learn how to compute the response of a digital system in Chapter 4.

Figure 3.2
Continuous and digital step response using Euler's method for discretization: (a) 20 Hz sample rate, (b) 40 Hz sample rate



² Assumes the use of MATLAB v5 and Control System Toolbox v4. For prior versions, see Appendix F.

In Chapter 6, you will see that there are several ways to approximate a continuous transfer function, each with different merits, and most with better qualities than the Euler method presented here. In fact, MATLAB provides a function (c2d.m) that computes these approximations. However, before those methods can be examined, it will be necessary to understand discrete transfer functions, a topic covered in Chapter 4.

3.2 Effect of Sampling

It is worthy to note that *the single most important* impact of implementing a control system digitally is the delay associated with the hold. A delay in any feedback system degrades the stability and damping of the system. Because each value of $u(kT)$ in Fig. 3.1(b) is held constant until the next value is available from the computer, the continuous value of $u(t)$ consists of steps (see Fig. 3.3) that, on the average, lag $u(kT)$ by $T/2$, as shown by the dashed line in the figure. By incorporating a continuous approximation of this $T/2$ delay in a continuous analysis of the system, an assessment can be made of the effect of the delay in the digitally controlled system. The delay can be approximated by the method of Padé. The simplest first-order approximation is

$$G_p(s) = \frac{2/T}{s + 2/T}. \quad (3.9)$$

Figure 3.4 compares the responses from Fig. 3.2 with a continuous analysis that includes a delay approximation according to Eq. (3.9).

This linear approximation of the sampling delay (Eq. (3.9)) could also be used to determine the effect of a particular sample rate on the roots of a system via linear analysis, perhaps a locus of roots vs. T . Alternatively, the effect of a delay can be analyzed using frequency response techniques because a time delay of $T/2$ translates into a phase decrease of

$$\delta\phi = -\frac{\omega T}{2}. \quad (3.10)$$

Figure 3.3
The delay due to the hold operation

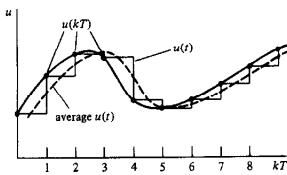
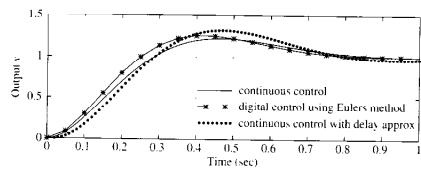


Figure 3.4

Continuous and digital step response at 20 Hz sample rate showing results with a $7/2$ delay approximation



Thus, we see that the loss of phase margin due to sampling can be estimated by invoking Eq. (3.10) with ω equal to the frequency where the magnitude equals one, that is, the "gain crossover frequency."

◆ Example 3.3 Approximate Analysis of the Effect of Sampling

For the system in Example 3.2, determine the decrease in damping that would result from sampling at 10 Hz. Use both linear analysis and the frequency response method. Compare the time response of the continuous system with the discrete implementation to validate the analysis.

Solution. The damping of the system in Example 3.2 can be obtained from the MATLAB statement

```
damp(sysCL)
```

where sysCL is that computed in Example 3.2. The result is $\zeta = .56$.

The damping of the system with the simple delay approximation added (Eq. (3.9)) is obtained from

```
T = 1/10
numDL = 2/T; denDL = [1 - 2/T]
sys2 = tf(numDL,denDL)*sys1
sysCL = feedback(sys2,1)
damp(sysCL)
```

where sys1 is that computed in Example 3.2. The result of this calculation is $\zeta = .33$.

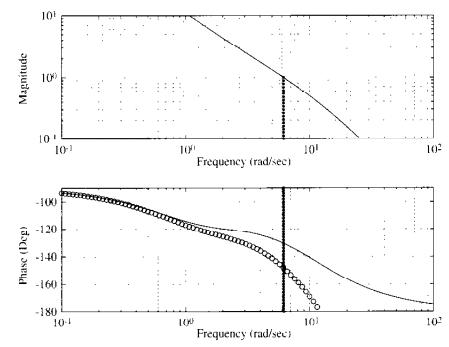
The frequency response of the continuous system is shown by the solid line in Fig. 3.5 and shows that the crossover frequency is about 6 rad/sec and the PM is about 50°. The line of small circles shows the phase corrected by Eq. (3.10) and, therefore, that the PM decreases to about 30°. For more precision, the use of margin.m in MATLAB shows that the continuous system has a PM of 49.5° at a crossover frequency of 6.17 rad/sec. Equation (3.10) then indicates that the correction due to sampling should be 17.7°, thus the PM of the digital system would be

31.8°. Since the PM is approximately $100 \times \zeta$, this analysis shows that the ζ decreases from approximately 0.5 for the continuous system to 0.32 for the digital system.

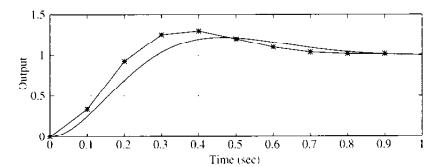
Both analysis methods indicate a similar reduction in the damping of the system. One should, therefore, expect that the overshoot of the step response should increase. For the case with no zeros, Fig. 2.7 indicates that this decrease in ζ should result in the step response overshoot, M_{∞} , going from 16% to 35% for a 2nd-order system with no zeros. The actual step responses in Fig. 3.6 have about 20% overshoot for the continuous system and about 30% for the digital case. So, we see that the approximate analysis was somewhat conservative in the prediction of the decreased damping and increased overshoot in the digital case. The trend that decreasing sample rate causes decreasing damping and stability will be analyzed more completely throughout the book. ◆

Figure 3.5

Frequency response for Example 3.3

**Figure 3.6**

Continuous and digital responses for Example 3.3 (at 10 Hz sample rate)



3.3 PID Control

The notion of proportional, integral, and derivative (PID) control is reviewed in Section 2.2.3. Reviewing again briefly, the three terms are proportional control

$$u(t) = K e(t), \quad (3.11)$$

integral control

$$u(t) = \frac{K}{T_i} \int_0^t e(\eta) d\eta, \quad (3.12)$$

and derivative control

$$u(t) = K T_d \dot{e}(t), \quad (3.13)$$

where K is called the proportional gain, T_i the integral time, and T_d the derivative time. These three constants define the control.

The approximations of these individual control terms to an algebraic equation that can be implemented in a digital computer are proportional control

$$u(k) = K e(k). \quad (3.14)$$

integral control

$$u(k) = u(k-1) + \frac{K}{T_i} T e(k), \quad (3.15)$$

and derivative control

$$u(k) = \frac{K T_d}{T} [e(k) - e(k-1)]. \quad (3.16)$$

Equation (3.11) is already algebraic, therefore Eq. (3.14) follows directly while Eqs. (3.15) and (3.16) result from an application of Euler's method (Eq. (3.2)) to Eqs. (3.12) and (3.13). However, normally these terms are used together and, in this case, the combination needs to be done carefully. The combined continuous transfer function (Eq. 2.24) is

$$D(s) = \frac{u(s)}{e(s)} = K \left(1 + \frac{1}{T_i s} + T_d s \right).$$

Therefore, the differential equation relating $u(t)$ and $e(t)$ is

$$\dot{u} = K (\dot{e} + \frac{1}{T_i} e + T_d \ddot{e})$$

and the use of Euler's method (twice for \ddot{e}) results in

$$u(k) = u(k-1) + K \left[\left(1 + \frac{T}{T_i} + \frac{T_d}{T} \right) e(k) - \left(1 + 2 \frac{T_d}{T} \right) e(k-1) + \frac{T_d}{T} e(k-2) \right]. \quad (3.17)$$

◆ Example 3.4 Transforming a Continuous PID to a Digital Computer

A micro-servo motor has a transfer function from the input applied voltage to the output speed (rad/sec),

$$G(s) = \frac{360000}{(s+60)(s+600)}. \quad (3.18)$$

It has been determined that PID control with $K = 5$, $T_d = 0.0008$ sec, and $T_i = 0.003$ sec gives satisfactory performance for the continuous case. Pick an appropriate sample rate, determine the corresponding digital control law, and implement on a digital system. Compare the digital step response with the calculated response of a continuous system. Also, separately investigate the effect of a higher sample rate and re-tuning the PID parameters on the ability of the digital system to match the continuous response.

Solution. The sample rate needs to be selected first. But before we can do that, we need to know how fast the system is or what its bandwidth is. The solid line in Fig. 3.7 shows the step response of the continuous system and indicates that the rise time is about 1 msec. Based on Eq. (2.16), this suggests that $\omega_n \approx 1800$ rad/sec, and so the bandwidth would be on the order of 2000 rad/sec or 320 Hz. Therefore, the sample rate would be about 3.2 kHz if 10 times bandwidth. So let's pick $T = 0.3$ msec. Use of Eq. (3.17) results in the difference equation

$$u(k) = u(k-1) + 5[3.7667e(k) - 6.3333e(k-1) + 2.6667e(k-2)]$$

which, when implemented in the digital computer results in the line with stars in Fig. 3.7. This implementation shows a considerably increased overshoot over the continuous case. The line with circles in the figure shows the improved performance obtained by increasing the sample rate to 10 kHz; i.e., a sample rate about 30 times bandwidth, while using the same PID parameters as before. It shows that the digital performance has improved to be essentially the same as the continuous case.

Increasing the sample rate, however, will increase the cost of the computer and the A/D converter; therefore, there will be a cost benefit by improving the performance while maintaining the 3.2 kHz sample rate. A look at Fig. 3.7 shows that the digital response ($T = 0.3$ msec) has a faster rise time and less damping than the continuous case. This suggests that the proportional gain K should be reduced to slow the system down and the derivative time, T_d , should be increased to increase the damping. Some trial and error, keeping these ideas in mind, produces the results in Fig. 3.8. The revised PID parameters that produced these results are $K = 3.2$ and $T_d = 0.0011$ sec. The integral reset time, T_i , was left unchanged.

This example once again showed the characteristics of a digital control system. The damping was degraded an increasing amount as the sample rate was reduced. Furthermore, it was possible to restore the damping with suitable adjustments to the control.

Figure 3.7
Step response of a micro-motor, Example 3.4, same PID parameters

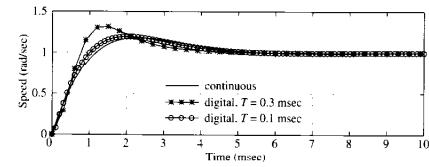
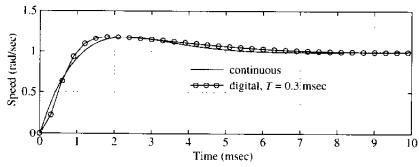


Figure 3.8
Effect of PID tuning on the digital response, Example 3.4



3.4 Summary

- Digitization methods allow the designer to convert a continuous compensation, $D(s)$, into a set of difference equations that can be programmed directly into a control computer.
 - Euler's method can be used for the digitization
- $$\dot{x}(k) \cong \frac{x(k+1) - x(k)}{T}. \quad (3.2)$$
- As long as the sample rate is on the order of $30 \times$ bandwidth or faster, the digitally controlled system will behave close to its continuous counterpart and the continuous analysis that has been the subject of your continuous control systems study will suffice.
- For sample rates on the order of 10 to 30 times the bandwidth, a first order analysis can be carried out by introducing a delay of $T/2$ in the continuous analysis to see how well the digital implementation matches the continuous analysis. A zero-pole approximation for this delay is

$$G_h(s) = \frac{2/T}{s + 2/T}. \quad (3.9)$$

The delay can be analyzed more accurately using frequency response where the phase from the continuous analysis should be decreased by

$$\delta\phi = \frac{\omega T}{2}. \quad (3.10)$$

- A continuous PID control law whose transfer function is

$$D(s) = \frac{u(s)}{e(s)} = K \left(1 + \frac{1}{T_i s} + T_o s \right)$$

can be implemented digitally using Eq. (3.17)

$$u(k) = u(k-1) + K \left[\left(1 + \frac{T_p}{T_i} + \frac{T_o}{T} \right) e(k) - \left(1 + 2 \frac{T_p}{T} \right) e(k-1) + \frac{T_o}{T} e(k-2) \right].$$

The digital control system will behave reasonably close to the continuous system providing the sample rate is faster than 30 times the bandwidth.

- In order to analyze the system accurately for any sample rate, but especially for sample rates below about 30 times bandwidth, you will have to proceed on to the next chapters to learn about z-transforms and how to apply them to the study of discrete systems.
- For digital control systems with sample rates less than 30 times bandwidth, design is often carried out directly in the discrete domain, eliminating approximation errors.

3.5 Problems

- 3.1 Do the following:

- (a) Design a continuous lead compensation for the satellite attitude control example ($G(s) = 1/s^2$) described in Appendix A.1 so that the complex roots are at approximately $s = -4.4 \pm j4.4$ rad/sec.
- (b) Assuming the compensation is to be implemented digitally, approximate the effect of the digital implementation to be a delay of $T/2$ as given by

$$G_h(s) = \frac{jT}{s + 2/T}$$

and determine the revised root locations for sample rates of $\omega_s = 5$ Hz, 10 Hz, and 20 Hz where $T = 1/\omega_s$ sec.

- 3.2 Repeat Example 3.1, but use the approximation that

$$\dot{x}(k) \cong \frac{x(k) - x(k-1)}{T}.$$

the backward rectangular version of Euler's method. Compare the resulting difference equations with the forward rectangular Euler method. Also compute the numerical value of the coefficients for both cases vs. sample rate for $\omega_s = 1 - 100$ Hz. Assume the continuous values from Eq. (3.8). Note that the coefficients of interest are given in Eq. (3.7) for the forward rectangular case as $(1 - bT)$ and $(aT - 1)$.

3.3 For the compensation

$$D(s) = 25 \frac{s + 1}{s + 15}.$$

use Euler's forward rectangular method to determine the difference equations for a digital implementation with a sample rate of 80 Hz. Repeat the calculations using the backward rectangular method (see Problem 3.2) and compare the difference equation coefficients.

3.4 For the compensation

$$D(s) = 5 \frac{s + 2}{s + 20}.$$

use Euler's forward rectangular method to determine the difference equations for a digital implementation with a sample rate of 80 Hz. Repeat the calculations using the backward rectangular method (see Problem 3.2) and compare the difference equation coefficients.

3.5 The read arm on a computer disk drive has the transfer function

$$G(s) = \frac{1000}{s^2}.$$

Design a digital PID controller that has a bandwidth of 100 Hz, a phase margin of 50°, and has no output error for a constant bias torque from the drive motor. Use a sample rate of 6 kHz.

3.6 The read arm on a computer disk drive has the transfer function

$$G(s) = \frac{1000}{s^2}.$$

Design a digital controller that has a bandwidth of 100 Hz and a phase margin of 50°. Use a sample rate of 6 kHz.

3.7 For

$$G(s) = \frac{1}{s^2}.$$

- (a) design a continuous compensation so that the closed-loop system has a rise time $t_r < 1$ sec and overshoot $M_p < 15\%$ to a step input command.
- (b) revise the compensation so the specifications would still be met if the feedback was implemented digitally with a sample rate of 5 Hz, and
- (c) find difference equations that will implement the compensation in the digital computer.

3.8 The read arm on a computer disk drive has the transfer function

$$G(s) = \frac{500}{s^2}.$$

Design a continuous lead compensation so that the closed-loop system has a bandwidth of 100 Hz and a phase margin of 50°. Modify the MATLAB file fig32.m so that you can evaluate the digital version of your lead compensation using Euler's forward rectangular method. Try different sample rates, and find the slowest one where the overshoot does not exceed 30%.

3.9 The antenna tracker has the transfer function

$$G(s) = \frac{10}{s(s + 2)}.$$

Design a continuous lead compensation so that the closed-loop system has a rise time $t_r < 0.3$ sec and overshoot $M_p < 10\%$. Modify the MATLAB file fig32.m so that you can evaluate the digital version of your lead compensation using Euler's forward rectangular method. Try different sample rates, and find the slowest one where the overshoot does not exceed 20%.

3.10 The antenna tracker has the transfer function

$$G(s) = \frac{10}{s(s + 2)}.$$

Design a continuous lead compensation so that the closed-loop system has a rise time $t_r < 0.3$ sec and overshoot $M_p < 10\%$. Approximate the effect of a digital implementation to be

$$G_d(s) = \frac{2/T}{s + 2/T}.$$

and estimate M_p for a digital implementation with a sample rate of 10 Hz.

• 4 •

Discrete Systems Analysis

A Perspective on Discrete Systems Analysis

The unique element in the structure of Fig. 3.1 is the digital computer. The fundamental character of the digital computer is that it takes a finite time to compute answers, and it does so at discrete steps in time. The purpose of this chapter is to develop tools of analysis necessary to understand and to guide the design of programs for a computer sampling at discrete times and acting as a linear, dynamic control component. Needless to say, digital computers can do many things other than control linear dynamic systems; it is our purpose in this chapter to examine their characteristics when doing this elementary control task and to develop the basic analysis tools needed to write programs for real-time computer control.

Chapter Overview

Section 4.1 restates the difference equations used by a computer to represent a dynamic system, a topic covered very briefly in Section 3.1. The tool for analyzing this sort of system, the z -transform, is introduced and developed in Section 4.2. Use of the z -transform is developed further in Section 4.3 to show how it applies to the combined system in Fig. 3.1. Furthermore, state-space models of discrete systems are developed in this section. Section 4.4 shows the correspondence between roots in the z -plane and time response characteristics while Section 4.5 discusses characteristics of the discrete frequency response. The last section, 4.6, derives properties of the z -transform.

4.1 Linear Difference Equations

We assume that the analog-to-digital converter (A/D) in Fig. 1.1 takes samples of the signal y at discrete times and passes them to the computer so that $\hat{y}(kT) =$

$y(kT)$. The job of the computer is to take these sample values and compute in some fashion the signals to be put out through the digital-to-analog converter (D/A). The characteristics of the A/D and D/A converters will be discussed later. Here we consider the treatment of the data inside the computer. Suppose we call the input signals up to the k th sample $e_0, e_1, e_2, \dots, e_k$, and the output signals prior to that time $u_0, u_1, u_2, \dots, u_{k-1}$. Then, to get the next output, we have the machine compute some function, which we can express in symbolic form as

$$u_k = f(e_0, \dots, e_k; u_0, \dots, u_{k-1}). \quad (4.1)$$

Because we plan to emphasize the elementary and the dynamic possibilities, we assume that the function f in Eq. (4.1) is *linear* and depends on only a *finite* number of past e 's and u 's. Thus we write

$$\begin{aligned} u_k = & -a_1 u_{k-1} - a_2 u_{k-2} - \cdots - a_n u_{k-n} \\ & + b_0 e_k + b_1 e_{k-1} + \cdots + b_m e_{k-m}. \end{aligned} \quad (4.2)$$

Equation (4.2) is called a linear **recurrence equation** or difference equation and, as we shall see, has many similarities with a linear differential equation. The name "difference equation" derives from the fact that we could write Eq. (4.2) using u_k plus the differences in u_k , which are defined as

$$\begin{aligned} \nabla u_k &= u_k - u_{k-1} \\ \nabla^2 u_k &= \nabla u_k - \nabla u_{k-1} \\ \nabla^n u_k &= \nabla^{n-1} u_k - \nabla^{n-1} u_{k-n}. \end{aligned} \quad (4.3)$$

If we solve Eq. (4.3) for the values of u_k , u_{k-1} , and u_{k-2} in terms of differences, we find

$$\begin{aligned} u_k &= u_k \\ u_{k-1} &= u_k - \nabla u_k \\ u_{k-2} &= u_k - 2\nabla u_k + \nabla^2 u_k. \end{aligned}$$

Thus, for a second-order equation with coefficients a_1 , a_2 , and b_0 (we let $b_1 = b_2 = 0$ for simplicity), we find the equivalent difference equation to be

$$a_2 \nabla^2 u_k - (a_1 + 2a_2) \nabla u_k + (a_2 + a_1 + 1) u_k = b_0 e_k.$$

Although the two forms are equivalent, the recurrence form of Eq. (4.2) is more convenient for computer implementation; we will drop the form using differences. We will continue, however, to refer to our equations as "difference equations." If the a 's and b 's in Eq. (4.2) are constant, then the computer is solving a **constant-coefficient difference equation** (CCDE). We plan to demonstrate later that with such equations the computer can control linear constant dynamic systems and approximate most of the other tasks of linear, constant, dynamic systems, including performing the functions of electronic filters. To do so, it is necessary first to examine methods of obtaining solutions to Eq. (4.2) and to study the general properties of these solutions.

constant coefficients

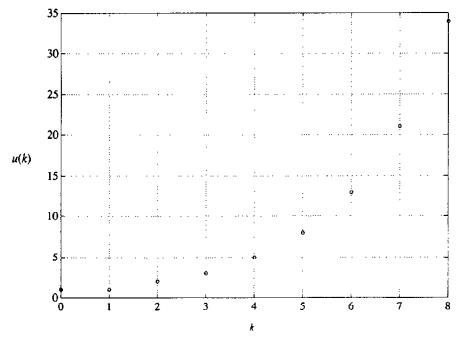
To solve a specific CCDE is an elementary matter. We need a starting time (k -value) and some initial conditions to characterize the contents of the computer memory at this time. For example, suppose we take the case

$$u_k = u_{k-1} + u_{k-2} \quad (4.4)$$

and start at $k = 2$. Here there are no input values, and to compute u_k we need to know the (initial) values for u_0 and u_1 . Let us take them to be $u_0 = u_1 = 1$. The first nine values are 1, 1, 2, 3, 5, 8, 13, 21, 34, . . . A plot of the values of u_k versus k is shown in Fig. 4.1.

The results, the **Fibonacci numbers**, are named after the thirteenth-century mathematician¹ who studied them. For example, Eq. (4.4) has been used to model the growth of rabbits in a protected environment². However that may be, say the least. If the response of a dynamic system to any finite initial conditions can grow without bound, we call the system **unstable**. We would like to be able to examine equations like Eq. (4.2) and, without having to solve them explicitly, see if they are stable or unstable and even understand the general shape of the solution.

Figure 4.1
The Fibonacci numbers



¹ Leonardo Fibonacci of Pisa, who introduced Arabic notation to the Latin world about 1200 A.D.
² Wilde (1964). Assume that u_k represents pairs of rabbits and that babies are born in pairs. Assume that no rabbits die and that a new pair begin reproduction after one period. Thus at time k , we have all the old rabbits, u_{k-1} , plus the newborn pairs born to the mature rabbits, which are u_{k-2} .

One approach to solving this problem is to assume a form for the solution with unknown constants and to solve for the constants to match the given initial conditions. For continuous, ordinary, differential equations that are constant and linear, exponential solutions of the form e^{rt} are used. In the case of linear, constant, difference equations, it turns out that solutions of the form z^k will do where z has the role of s and k is the discrete independent variable replacing time, t . Consider Eq. (4.4). If we assume that $u(k) = A z^k$, we get the equation

$$A z^k = A z^{k-1} + A z^{k-2}.$$

Now if we assume $z \neq 0$ and $A \neq 0$, we can divide by A and multiply by z^{-k} , with the result

$$1 = z^{-1} + z^{-2}$$

or

$$z^2 = z + 1.$$

This polynomial of second degree has two solutions, $z_{1,2} = 1/2 \pm \sqrt{5}/2$. Let's call these z_1 and z_2 . Since our equation is linear, a sum of the individual solutions will also be a solution. Thus, we have found that a solution to Eq. (4.4) is of the form

$$u(k) = A_1 z_1^k + A_2 z_2^k.$$

We can solve for the unknown constants by requiring that this general solution satisfy the specific initial conditions given. If we substitute $k = 0$ and $k = 1$, we obtain the simultaneous equations

$$\begin{aligned} 1 &= A_1 + A_2, \\ 1 &= A_1 z_1 + A_2 z_2. \end{aligned}$$

These equations are easily solved to give

$$\begin{aligned} A_1 &= \frac{\sqrt{5} + 1}{2\sqrt{5}}, \\ A_2 &= \frac{\sqrt{5} - 1}{2\sqrt{5}}. \end{aligned}$$

And now we have the complete solution of Eq. (4.4) in a closed form. Furthermore, we can see that since $z_1 = (1 + \sqrt{5})/2$ is greater than 1, the term in z_1^k will grow without bound as k grows, which confirms our suspicion that the equation represents an unstable system. We can generalize this result. The equation in z that we obtain after we substitute $u = z^k$ is a polynomial in z known as the **characteristic equation** of the difference equation. If any solution of this equation is outside the unit circle (has a magnitude greater than one),

characteristic equation

the corresponding difference equation is unstable in the specific sense that for some finite initial conditions the solution will grow without bound as time goes to infinity. If all the roots of the characteristic equation are *inside* the unit circle, the corresponding difference equation is stable.

◆ Example 4.1 Discrete Stability

Is the equation

$$u(k) = 0.9u(k-1) - 0.2u(k-2)$$

stable?

Solution. The characteristic equation is

$$z^2 - 0.9z + 0.2 = 0,$$

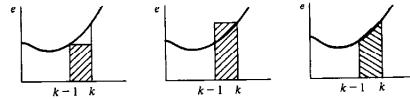
and the characteristic roots are $z = 0.5$ and $z = 0.4$. Since both these roots are inside the unit circle, the equation is stable. ◆

As an example of the origins of a difference equation with an external input, we consider the discrete approximation to integration. Suppose we have a continuous signal, $e(t)$, of which a segment is sketched in Fig. 4.2, and we wish to compute an approximation to the integral

$$\mathcal{I} = \int_0^t e(t) dt, \quad (4.5)$$

using only the discrete values $e(0), \dots, e(t_{k-1}), e(t_k)$. We assume that we have an approximation for the integral from zero to the time t_{k-1} and we call it u_{k-1} . The problem is to obtain u_k from this information. Taking the view of the integral as the area under the curve $e(t)$, we see that this problem reduces to finding an approximation to the area under the curve between t_{k-1} and t_k . Three alternatives are sketched in Fig. 4.2. We can use the rectangle of height e_{k-1} , or the rectangle

Figure 4.2
Plot of a function and alternative approximations to the area under the curve over a single time interval



of height e_k , or the trapezoid formed by connecting e_{k-1} to e_k by a straight line. If we take the third choice, the area of the trapezoid is

$$A = \frac{t_k - t_{k-1}}{2} (e_k + e_{k-1}). \quad (4.6)$$

Finally, if we assume that the sampling period, $t_k - t_{k-1}$, is a constant, T , we are led to a simple formula for discrete integration called the **trapezoid rule**

$$u_k = u_{k-1} + \frac{T}{2} (e_k + e_{k-1}). \quad (4.7)$$

If $e(t) = t$, then $e_i = kT$ and substitution of $u_k = (T^2/2)k^2$ satisfies Eq. (4.7) and is exactly the integral of e . (It should be, because if $e(t)$ is a straight line, the trapezoid is the *exact* area.) If we approximate the area under the curve by the rectangle of height e_{k-1} , the result is called the **forward rectangular rule** (sometimes called Euler's method, as discussed in Chapter 3 for an approximation to differentiation) and is described by

$$u_k = u_{k-1} + Te_{k-1}.$$

The other possibility is the **backward rectangular rule**, given by

$$u_k = u_{k-1} + Te_k.$$

Each of these integration rules is a special case of our general difference equation Eq. (4.2). We will examine the properties of these rules later, in Chapter 6, while discussing means to obtain a difference equation that will be equivalent to a given differential equation.

Thus we see that difference equations can be evaluated directly by a digital computer and that they can represent models of physical processes and approximations to integration. It turns out that if the difference equations are linear with coefficients that are constant, we can describe the relation between u and e by a transfer function, and thereby gain a great aid to analysis and also to the design of linear, constant, discrete controls.

4.2 The Discrete Transfer Function

We will obtain the transfer function of linear, constant, discrete systems by the method of z -transform analysis. A logical alternative viewpoint that requires a bit more mathematics but has some appeal is given in Section 4.6.2. The results are the same. We also show how these same results can be expressed in the state space form in Section 4.2.3.

4.2.1 The z -Transform

If a signal has discrete values $e_0, e_1, \dots, e_k, \dots$ we define the z -transform of the signal as the function^{3,4}

$$\begin{aligned} E(z) &\equiv Z\{e(k)\} \\ &\equiv \sum_{k=-\infty}^{\infty} e_k z^{-k}, \quad r_o < |z| < R_o. \end{aligned} \quad (4.8)$$

and we assume we can find values of r_o and R_o as bounds on the magnitude of the complex variable z for which the series Eq. (4.8) converges. A discussion of convergence is deferred until Section 4.6.

◆ Example 4.2 The z -Transform

The data e_k are taken as samples from the time signal $e^{-at}1(t)$ at sampling period T where $1(t)$ is the unit step function, zero for $t < 0$, and one for $t \geq 0$. Then $e_k = e^{-akT}1(kT)$. Find the z -transform of this signal.

Solution. Applying Eq. (4.8), we find that

$$\begin{aligned} \sum_{k=-\infty}^{\infty} e_k z^{-k} &= \sum_{k=0}^{\infty} e_k z^{-k} \\ &= \sum_{k=0}^{\infty} (e^{-akT} z^{-1})^k \\ &= \frac{1}{1 - e^{-aT} z^{-1}} \\ &= \frac{z}{z - e^{-aT}}, \quad |z| > e^{-aT}. \end{aligned}$$

We will return to the analysis of signals and development of a table of useful z -transforms in Section 4.4; we first examine the use of the transform to reduce

³ We use the notation $\hat{=}$ to mean "is defined as."

⁴ In Eq. (4.8) the lower limit is $-\infty$ so that values of e_k on both sides of $k = 0$ are included. The transform so defined is sometimes called the two-sided z -transform to distinguish it from the one-sided definition, which would be $\sum_{k=0}^{\infty} e_k z^{-k}$. For signals that are zero for $k < 0$, the transforms obviously give identical results. To take the one-sided transform of u_{k-1} , however, we must handle the value of u_{-1} , and thus are initial conditions introduced by the one-sided transform. Examination of this property and other features of the one-sided transform are invited by the problems. We select the two-sided transform because we need to consider signals that extend into negative time when we study random signals in Chapter 12.

difference equations to algebraic equations and techniques for representing these as block diagrams.

4.2.2 The Transfer Function

The z transform has the same role in discrete systems that the Laplace transform has in analysis of continuous systems. For example, the z -transforms for e_i and u_i in the difference equation (4.2) or in the trapezoid integration (4.7) are related in a simple way that permits the rapid solution of linear, constant, difference equations of this kind. To find the relation, we proceed by direct substitution. We take the definition given by Eq. (4.8) and, in the same way, we define the z -transform of the sequence $\{u_k\}$ as

$$U(z) = \sum_{k=-\infty}^{\infty} u_k z^{-k}. \quad (4.9)$$

Now we multiply Eq. (4.7) by z^{-k} and sum over k . We get

$$\sum_{k=-\infty}^{\infty} u_k z^{-k} = \sum_{k=-\infty}^{\infty} u_{k-1} z^{-k} + \frac{T}{2} \left(\sum_{k=-\infty}^{\infty} e_k z^{-k} + \sum_{k=-\infty}^{\infty} e_{k-1} z^{-k} \right). \quad (4.10)$$

From Eq. (4.9), we recognize the left-hand side as $U(z)$. In the first term on the right, we let $k-1=j$ to obtain

$$\sum_{k=-\infty}^{\infty} u_{k-1} z^{-k} = \sum_{j=-\infty}^{\infty} u_j z^{-(j+1)} = z^{-1} U(z). \quad (4.11)$$

By similar operations on the third and fourth terms we can reduce Eq. (4.10) to

$$U(z) = z^{-1} U(z) + \frac{T}{2} [E(z) + z^{-1} E(z)]. \quad (4.12)$$

Equation (4.12) is now simply an algebraic equation in z and the functions U and E . Solving it we obtain

$$U(z) = \frac{T}{2} \frac{1+z^{-1}}{1-z^{-1}} E(z). \quad (4.13)$$

We define the ratio of the transform of the output to the transform of the input as the **transfer function**, $H(z)$. Thus, in this case, the transfer function for trapezoid-rule integration is

$$\frac{U(z)}{E(z)} \equiv H(z) = \frac{Tz+1}{2z-1}. \quad (4.14)$$

For the more general relation given by Eq. (4.2), it is readily verified by the same techniques that

$$H(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}.$$

zeros
poles

and if $n \geq m$, we can write this as a ratio of polynomials in z as

$$H(z) = \frac{b_0 z^n + b_1 z^{n-1} + \cdots + b_m z^{n-m}}{z^n + a_1 z^{n-1} + a_2 z^{n-2} + \cdots + a_n} \quad (4.15)$$

or

$$H(z) = \frac{b(z)}{a(z)}.$$

This transfer function is represented in MATLAB in the `tf` form similarly to the continuous case as discussed after Eq. (2.6). The numerator of Eq. (4.15) would be specified in MATLAB as a $1 \times (n+1)$ matrix of the coefficients, for example, when $m=n$

$$\text{num} = [b_0 \ b_1 \ b_2 \ \dots \ b_m]$$

and when $n > m$, there would be $n-m$ zeros after b_m . The quantity specifying the denominator would be specified as a $1 \times (n+1)$ matrix, for example

$$\text{den} = [1 \ a_1 \ a_2 \ \dots \ a_n].$$

Note that $H(z)$ was assumed to be in the form given by Eq. (4.15), that is, with positive powers of z . The discrete system is specified as⁵

$$\text{sys} = \text{tf}(\text{num}, \text{den}, T)$$

where T is the sample period.

The general input-output relation between transforms with linear, constant, difference equations is

$$U(z) = H(z)E(z). \quad (4.16)$$

Although we have developed the transfer function with the z -transform, it is also true that the transfer function is the ratio of the output to the input when both vary as z^k .

Because $H(z)$ is a rational function of a complex variable, we use the terminology of that subject. Suppose we call the numerator polynomial $b(z)$ and the denominator $a(z)$. The places in z where $b(z)=0$ are **zeros** of the transfer function, and the places in z where $a(z)=0$ are the **poles** of $H(z)$. If z_0 is a pole and $(z - z_0)^p H(z)$ has neither pole nor zero at z_0 , we say that $H(z)$ has a pole of order p at z_0 . If $p=1$, the pole is simple. The transfer function Eq. (4.14) has a simple pole at $z=1$ and a simple zero at $z=-1$. When completely factored, the transfer function would be

$$H(z) = K \frac{\prod_{i=1}^m (z - z_i)}{\prod_{i=1}^n (z - p_i)}. \quad (4.17)$$

⁵ Assumes the use of MATLAB v5 and Control Toolbox v4. For prior versions, see Appendix F.

and the quantities specifying the transfer function in the MATLAB zpk form are an $m \times 1$ matrix of the zeros, an $n \times 1$ matrix of the poles, and a scalar gain, for example

$$z = \begin{bmatrix} z_1 \\ z_2 \\ \dots \\ z_m \end{bmatrix}, \quad p = \begin{bmatrix} p_1 \\ p_2 \\ \dots \\ p_n \end{bmatrix}, \quad k = K.$$

The system is then

$$\text{sys} = \text{zpk}(z, p, k, T).$$

We can now give a physical meaning to the variable z . Suppose we let all coefficients in Eq. (4.15) be zero except b_i and we take b_i to be 1. Then $H(z) = z^{-1}$. But $H(z)$ represents the transform of Eq. (4.2), and with these coefficient values the difference equation reduces to

$$u_k = e_{k-1}. \quad (4.18)$$

z^{-1} and cycle delay

The present value of the output, u_k , equals the input delayed by one period. Thus we see that a transfer function of z^{-1} is a *delay* of one time unit. We can picture the situation as in Fig. 4.3, where both time and transform relations are shown.

Since the relations of Eqs. (4.7), (4.14), (4.15) are all composed of delays, they can be expressed in terms of z^{-1} . Consider Eq. (4.7). In Fig. 4.4 we illustrate the difference equation (4.7) using the transfer function z^{-1} as the symbol for a unit delay.

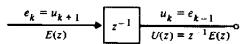
We can follow the operations of the discrete integrator by tracing the signals through Fig. 4.4. For example, the present value of e_k is passed to the first summer, where it is added to the previous value e_{k-1} , and the sum is multiplied by $T/2$ to compute the area of the trapezoid between e_{k-1} and e_k . This is the signal marked a_k in Fig. 4.4. After this, there is another sum, where the previous output, u_{k-1} , is added to the new area to form the next value of the integral estimate, u_k . The discrete integration occurs in the loop with one delay, z^{-1} , and unity gain.

4.2.3 Block Diagrams and State-Variable Descriptions

Because Eq. (4.16) is a linear algebraic relationship, a system of such relations is described by a system of linear equations. These can be solved by the methods of linear algebra or by the graphical methods of block diagrams in the same

Figure 4.3

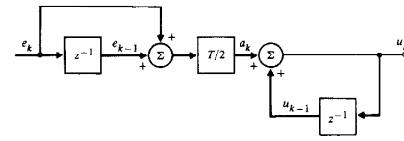
The unit delay



$$e_k \xrightarrow{z^{-1}} u_k = e_{k-1}$$

$$E(z) = z^{-1}$$

Figure 4.4
A block diagram of trapezoid integration as represented by Eq. (4.7)



way as for continuous system transfer functions. To use block-diagram analysis to manipulate these discrete-transfer-function relationships, there are only four primitive cases:

1. The transfer function of paths in parallel is the sum of the single-path transfer functions (Fig. 4.5).
2. The transfer function of paths in series is the *product* of the path transfer functions (Fig. 4.6).
3. The transfer function of a single loop of paths is the transfer function of the forward path divided by one minus the loop transfer function (Fig. 4.7).
4. The transfer function of an arbitrary multipath diagram is given by combinations of these cases. Mason's rule⁶ can also be used.

For the general difference equation of (4.2), we already have the transfer function in Eq. (4.15). It is interesting to connect this case with a block diagram

Figure 4.5
Block diagram of parallel blocks

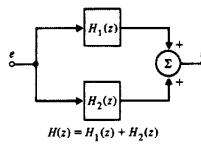
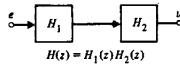
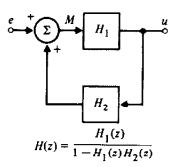


Figure 4.6
Block diagram of cascade blocks



⁶ Mason (1956). See Franklin, Powell, and Emami-Naeini (1986) for a discussion.

Figure 4.7
Feedback transfer function



using only simple delay forms for z in order to see several "canonical" block diagrams and to introduce the description of discrete systems using equations of state.

*Canonical Forms

There are many ways to reduce the difference equation (4.2) to a block diagram involving z only as the delay operator, z^{-1} . The first one we will consider leads to the "control" canonical form. We begin with the transfer function as a ratio of polynomials

$$U(z) = H(z)E(z) = \frac{b(z)}{a(z)} E(z) = b(z)\xi,$$

where

$$\xi = \frac{E(z)}{a(z)}$$

and thus

$$a(z)\xi = E(z).$$

At this point we need to get specific; and rather than carry through with a system of arbitrary order, we will work out the details for the third-order case. In the development that follows, we will consider the variables u , e , and ξ as time variables and z as an advance operator such that $zu(k) = u(k+1)$ or $z^{-1}u(k) = u(k-1)$. With this convention (which is simply using the property of z derived earlier), consider the equations

$$(z^3 + a_1z^2 + a_2z + a_3)\xi = e, \quad (4.19)$$

$$(b_0z^3 + b_1z^2 + b_2z + b_3)\xi = u. \quad (4.20)$$

We can write Eq. (4.19) as

$$\begin{aligned} z^3\xi &= e - a_1z^2\xi - a_2z\xi - a_3\xi, \\ \xi(k+3) &= e(k) - a_1\xi(k+2) - a_2\xi(k+1) - a_3\xi(k). \end{aligned} \quad (4.21)$$

Now assume we have $z^3\xi$, which is to say that we have $\xi(k+3)$ because z^3 is an advance operator of three steps. If we operate on this with z^{-1} three times in a row, we will get back to $\xi(k)$, as shown in Fig. 4.8(a). From Eq. (4.21), we can now compute $z^3\xi$ from e and the lower powers of z and ξ given in the block diagram; the picture is now as given in Fig. 4.8(b). To complete the representation of Eqs. (4.19) and (4.20), we need only add the formation of the output u as a weighted sum of the variables $z^3\xi$, $z^2\xi$, $z\xi$, and ξ according to Eq. (4.20). The completed picture is shown in Fig. 4.8(c).

In Fig. 4.8(c), the internal variables have been named x_1 , x_2 , and x_3 . These variables comprise the state of this dynamic system in this form. Having the block diagram shown in Fig. 4.8(c), we can write down, almost by inspection, the difference equations that describe the evolution of the state, again using the fact that the transfer function z^{-1} corresponds to a one-unit delay. For example, we see that $x_1(k+1) = x_1(k)$ and $x_2(k+1) = x_2(k)$. Finally, expressing the sum at the far left of the figure, we have

$$x_1(k+1) = -a_1x_1(k) - a_2x_2(k) - a_3x_3(k) + e(k).$$

We collect these three equations together in proper order, and we have

$$x_1(k+1) = -a_1x_1(k) - a_2x_2(k) - a_3x_3(k) + e(k), \quad (4.22)$$

$$x_2(k+1) = x_1(k),$$

$$x_3(k+1) = x_2(k).$$

Using vector-matrix notation,⁷ we can write this in the compact form

$$\mathbf{x}(k+1) = \mathbf{A}_c \mathbf{x}(k) + \mathbf{B}_c e(k),$$

where

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \\ \mathbf{A}_c &= \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{aligned} \quad (4.23)$$

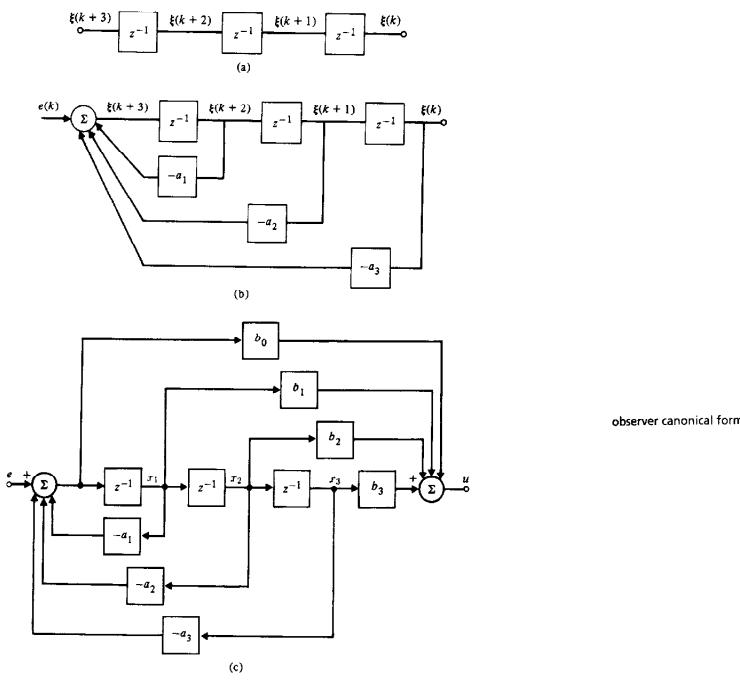
and

$$\mathbf{B}_c = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \quad (4.24)$$

The output equation is also immediate except that we must watch to catch all paths by which the state variables combine in the output. The problem is caused

⁷ We assume the reader has some knowledge of matrices. The results we require and references to study material are given in Appendix C.

Figure 4.8
Block diagram development of control canonical form.
(a) Solving for $\xi(k)$;
(b) solving for $\xi(k+3)$ from $e(k)$ and past ξ 's;
(c) solving for $U(k)$ from ξ 's



observer canonical form

by the b_0 term. If $b_0 = 0$, then $u = b_1x_1 + b_2x_2 + b_3x_3$, and the corresponding matrix form is immediate. However, if b_0 is not 0, x_i for example not only reaches the output through b_i , but also by the parallel path with gain $-b_i a_i$. The complete equation is

$$u = (b_1 - a_1 b_0)x_1 + (b_2 - a_2 b_0)x_2 + (b_3 - a_3 b_0)x_3 + b_0 e.$$

In vector/matrix notation, we have

$$u = \mathbf{C}_c x + \mathbf{D}_c e$$

where

$$\mathbf{C}_c = [b_1 - a_1 b_0 \quad b_2 - a_2 b_0 \quad b_3 - a_3 b_0] \quad (4.25)$$

$$\mathbf{D}_c = b_0. \quad (4.26)$$

We can combine the equations for the state evolution and the output to give the very useful and most compact equations for the dynamic system,

$$\mathbf{x}(k+1) = \mathbf{A}_c \mathbf{x}(k) + \mathbf{B}_c e(k).$$

where \mathbf{A}_c and \mathbf{B}_c for this control canonical form are given by Eq. (4.23), and \mathbf{C}_c and \mathbf{D}_c are given by Eq. (4.25).

The other canonical form we want to illustrate is called the "observer" canonical form and is found by starting with the difference equations in operator/transform form as

$$z^3 u + a_1 z^2 u + a_2 z u + a_3 u = b_0 z^3 e + b_1 z^2 e + b_2 z e + b_3 e.$$

In this equation, the external input is $e(k)$, and the response is $u(k)$, which is the solution of this equation. The terms with factors of z are time-shifted toward the future with respect to k and must be eliminated in some way. To do this, we assume at the start that we have the $u(k)$, and of course the $e(k)$, and we rewrite the equation as

$$b_3 e - a_3 u = z^3 u + a_1 z^2 u + a_2 z u - b_0 z^3 e - b_1 z^2 e - b_2 z e.$$

Here, every term on the right is multiplied by at least one power of z , and thus we can operate on the lot by z^{-1} as shown in the partial block diagram drawn in Fig. 4.9(a).

Now in this internal result there appear $a_3 u$ and $-b_3 e$, which can be canceled by adding proper multiples of u and e , as shown in Fig. 4.9(b), and once they have been removed, the remainder can again be operated on by z^{-1} .

If we continue this process of subtracting out the terms at k and operating on the rest by z^{-1} , we finally arrive at the place where all that is left is u alone! But that is just what we assumed we had in the first place, so connecting this term back to the start finishes the block diagram, which is drawn in Fig. 4.9(c).

Figure 4.9
Block diagram development of observer canonical form: (a) the first partial sum and delay, (b) the second partial sum and delay, (c) the completion with the solution for $u(k)$

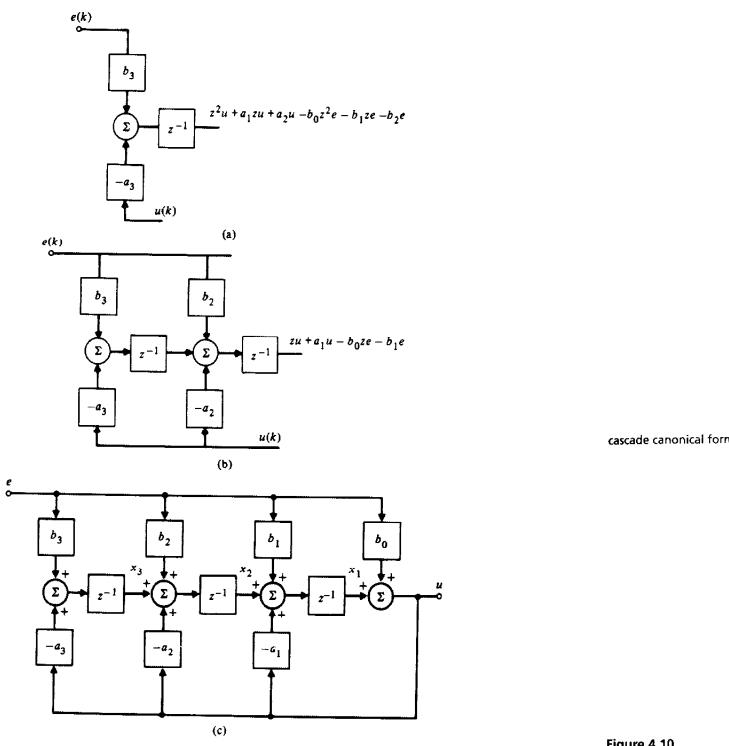
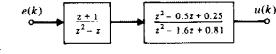


Figure 4.10
Block diagram of a cascade realization



A preferred choice of numbering for the state components is also shown in the figure. Following the technique used for the control form, we find that the matrix equations are given by

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_o \mathbf{x}(k) + \mathbf{B}_o e(k) \\ u(k) &= \mathbf{C}_o \mathbf{x}(k) + \mathbf{D}_o e(k), \end{aligned} \quad (4.27)$$

where

$$\begin{aligned} \mathbf{A}_o &= \begin{bmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{bmatrix} \\ \mathbf{B}_o &= \begin{bmatrix} b_1 - b_0 a_1 \\ b_2 - b_0 a_2 \\ b_3 - b_0 a_3 \end{bmatrix} \\ \mathbf{C}_o &= [1 \ 0 \ 0] \\ \mathbf{D}_o &= [b_0]. \end{aligned}$$

The block diagrams of Figs. 4.8 and 4.9 are called **direct canonical form** realizations of the transfer function $H(z)$ because the gains of the realizations are coefficients in the transfer-function polynomials.

Another useful form is obtained if we realize a transfer function by placing several first- or second-order direct forms in series with each other, a **cascade canonical form**. In this case, the $H(z)$ is represented as a product of factors, and the poles and zeros of the transfer function are clearly represented in the coefficients.

For example, suppose we have a transfer function

$$\begin{aligned} H(z) &= \frac{z^3 + 0.5z^2 - 0.25z + 0.25}{z^4 - 2.6z^3 + 2.4z^2 - 0.8z} \\ &= \frac{(z+1)(z^2 - 0.5z + 0.25)}{(z^2 - z)(z^2 - 1.6z + 0.8)}. \end{aligned}$$

The zero factor $z+1$ can be associated with the pole factor $z^2 - z$ to form one second-order system, and the zero factor $z^2 - 0.5z + 0.25$ can be associated with the second-order pole factor $z^2 - 1.6z + 0.8$ to form another. The cascade factors, which could be realized in a direct form such as control or observer form, make a cascade form as shown in Fig. 4.10.

4.2.4 Relation of Transfer Function to Pulse Response

We have shown that a transfer function of z^{-1} is a unit delay in the time domain. We can also give a time-domain meaning to an arbitrary transfer function. Recall that the z -transform is defined by Eq. (4.8) to be $E(z) = \sum e_k z^{-k}$, and the transfer function is defined from Eq. (4.16) as $H(z)$ when the input and output are related by $U(z) = H(z)E(z)$. Now suppose we deliberately select $e(k)$ to be the unit discrete pulse defined by

$$e_k = \begin{cases} 1, & (k = 0), \\ 0, & (k \neq 0), \end{cases} \stackrel{\triangle}{=} \delta_k. \quad (4.28)$$

Then it follows that $E(z) = 1$ and therefore that

$$U(z) = H(z). \quad (4.29)$$

Thus the transfer function $H(z)$ is seen to be the *transform* of the response to a unit-pulse input. For example, let us look at the system of Fig. 4.4 and put a unit pulse in at the e_1 -node (with no signals in the system beforehand).⁸ We can readily follow the pulse through the block and build Table 4.1.

Thus the unit-pulse response is zero for negative k , is $T/2$ at $k = 0$, and equals T thereafter. The z -transform of this sequence is

$$H(z) = \sum_{-\infty}^{\infty} h_k z^{-k} \stackrel{\triangle}{=} \sum_{-\infty}^{\infty} h_k z^{-k}.$$

Table 4.1

Step-by-step construction of the unit pulse response for Fig. 4.4

k	e_{k-1}	e_k	a_k	u_{k-1}	$u_k = h_k$
0	0	1	$T/2$	0	$T/2$
1	1	0	$T/2$	$T/2$	T
2	0	0	0	T	T
3	0	0	0	T	T

⁸ In this development we assume that Eq. (4.7) is intended to be used as a formula for computing values of u_k as k increases. There is no reason why we could not also solve for u_k as k takes on negative values. The direction of time comes from the application and not from the recurrence equation.

If we add $T/2$ to the z^0 -term and subtract $T/2$ from the whole series, we have a simpler sum, as follows

$$\begin{aligned} H(z) &= \sum_{k=0}^{\infty} Tz^{-k} - \frac{T}{2} \\ &= \frac{T}{1-z^{-1}} - \frac{T}{2} \quad (1 < |z|) \\ &= \frac{2T - T(1-z^{-1})}{2(1-z^{-1})} \\ &= \frac{T+z^{-1}}{2(1-z^{-1})} \\ &= \frac{Tz+1}{2z-1} \quad (1 < |z|). \end{aligned} \quad (4.30)$$

Of course, this is the transfer function we obtained in Eq. (4.13) from direct analysis of the difference equation.

A final point of view useful in the interpretation of the discrete transfer function is obtained by multiplying the infinite polynomials of $E(z)$ and $H(z)$ as suggested in Eq. (4.16). For purposes of illustration, we will assume that the unit-pulse response, h_k , is zero for $k < 0$. Likewise, we will take $k = 0$ to be the starting time for e_k . Then the product that produces $U(z)$ is the polynomial product given in Fig. 4.11.

Figure 4.11
Representation of the product $E(z)H(z)$ as a product of polynomials

$$\begin{array}{cccccc} e_0 & +e_1 z^{-1} & +e_2 z^{-2} & +e_3 z^{-3} & +\dots \\ h_0 & +h_1 z^{-1} & +h_2 z^{-2} & +h_3 z^{-3} & +\dots \\ \hline e_0 h_0 + e_1 h_0 z^{-1} & +e_2 h_0 z^{-2} & +e_3 h_0 z^{-3} & & & \\ +e_0 h_1 z^{-1} & +e_1 h_1 z^{-2} & +e_2 h_1 z^{-3} & & & \\ & +e_0 h_2 z^{-2} & +e_1 h_2 z^{-3} & & & \\ & & +e_0 h_3 z^{-3} & & & \\ \hline e_0 h_0 + (e_0 h_1 + e_1 h_0) z^{-1} + (e_0 h_2 + e_1 h_1 + e_2 h_0) z^{-2} + (e_0 h_3 + e_1 h_2 + e_2 h_1 + e_3 h_0) z^{-3} + \dots \end{array}$$

Since this product has been shown to be $U(z) = \sum u_k z^{-k}$, it must therefore follow that the coefficient of z^{-k} in the product is u_k . Listing these coefficients, we have the relations

$$\begin{aligned} u_0 &= e_0 h_0 \\ u_1 &= e_1 h_1 + e_0 h_0 \\ u_2 &= e_2 h_2 + e_1 h_1 + e_0 h_0 \\ u_3 &= e_3 h_3 + e_2 h_2 + e_1 h_1 + e_0 h_0 \end{aligned}$$

The extrapolation of this simple pattern gives the result

$$u_k = \sum_{j=0}^k e_j h_{k-j}.$$

By extension, we let the lower limit of the sum be $-\infty$ and the upper limit be $+\infty$:

$$u_k = \sum_{j=-\infty}^{\infty} e_j h_{k-j}. \quad (4.31)$$

Negative values of j in the sum correspond to inputs applied before time equals zero. Values for j greater than k occur if the unit-pulse response is nonzero for negative arguments. By definition, such a system, which responds *before* the input that causes it occurs, is called **noncausal**. This is the discrete **convolution** sum and is the analog of the convolution integral that relates input and impulse response to output in linear, constant, continuous systems.

To verify Eq. (4.31) we can take the z -transform of both sides

$$\sum_{k=-\infty}^{\infty} u_k z^{-k} = \sum_{k=-\infty}^{\infty} z^{-k} \sum_{j=-\infty}^{\infty} e_j h_{k-j}.$$

Interchanging the sum on j with the sum on k leads to

$$U(z) = \sum_{j=-\infty}^{\infty} e_j \sum_{k=-\infty}^{\infty} z^{-k} h_{k-j}.$$

Now let $k - j = l$ in the second sum

$$U(z) = \sum_{j=-\infty}^{\infty} e_j \sum_{l=-\infty}^{\infty} h_l z^{-(l+j)}.$$

but $z^{-(l+j)} = z^{-l} z^{-j}$, which leads to

$$U(z) = \sum_{j=-\infty}^{\infty} e_j z^{-j} \sum_{l=-\infty}^{\infty} h_l z^{-l},$$

and we recognize these two separate sums as

$$U(z) = E(z)H(z).$$

convolution

We can also derive the convolution sum from the properties of linearity and stationarity. First we need more formal definitions of "linear" and "stationary."

- Linearity:** A system with input e and output u is *linear* if superposition applies, which is to say, if $u_1(k)$ is the response to $e_1(k)$ and $u_2(k)$ is the response to $e_2(k)$, then the system is linear if and only if, for every scalar α and β , the response to $\alpha e_1 + \beta e_2$ is $\alpha u_1 + \beta u_2$.
- Stationarity:** A system is *stationary*, or time invariant, if a time shift in the input results in only a time shift in the output. For example, if we take the system at rest (no internal energy in the system) and apply a certain signal $e(k)$, suppose we observe a response $u(k)$. If we repeat this experiment at any later time when the system is again at rest and we apply the shifted input, $e(k - N)$, if we see $u(k - N)$, then the system is stationary. A constant coefficient difference equation is stationary and typically referred to as a constant system.

These properties can be used to derive the convolution in Eq. (4.31) as follows. If response to a unit pulse at $k = 0$ is $h(k)$, then response to a pulse of intensity e_0 is $e_0 h(k)$ if the system is linear. Furthermore, if the system is stationary then a delay of the input will delay the response. Thus, if

$$\begin{aligned} e &= e_l, & k &= l \\ &= 0, & k &\neq l. \end{aligned}$$

Finally, by linearity again, the total response at time k to a sequence of these pulses is the *sum* of the responses, namely,

$$u_k = e_0 h_k + e_1 h_{k-1} + \cdots + e_l h_{k-l} + \cdots + e_k h_0,$$

or

$$u_k = \sum_{l=0}^k e_l h_{k-l}.$$

Now note that if the input sequence began in the distant past, we must include terms for $l < 0$, perhaps back to $l = -\infty$. Similarly, if the system should be noncausal, future values of e where $l > k$ may also come in. The general case is thus (again)

$$u_k = \sum_{l=-\infty}^{\infty} e_l h_{k-l}. \quad (4.32)$$

4.2.5 External Stability

A very important qualitative property of a dynamic system is stability, and we can consider internal or external stability. Internal stability is concerned with the responses at all the internal variables such as those that appear at the delay

elements in a canonical block diagram as in Fig. 4.8 or Fig. 4.9 (the state). Otherwise we can be satisfied to consider only the **external stability** as given by the study of the input-output relation described for the linear stationary case by the convolution Eq. (4.32). These differ in that some internal modes might not be connected to both the input and the output of a given system.

For external stability, the most common definition of *appropriate response* is that for every Bounded Input, we should have a Bounded Output. If this is true we say the system is BIBO stable. A test for BIBO stability can be given directly in terms of the unit-pulse response, h_k . First we consider a sufficient condition. Suppose the input e_k is bounded, that is, there is an M such that

$$|e_l| \leq M < \infty \quad \text{for all } l. \quad (4.33)$$

If we consider the magnitude of the response given by Eq. (4.32), it is easy to see that

$$|u_k| \leq \left| \sum e_l h_{k-l} \right|,$$

which is surely less than the sum of the magnitudes as given by

$$\leq \sum_{l=-\infty}^{\infty} |e_l| |h_{k-l}|.$$

But, because we assume Eq. (4.33), this result is in turn bounded by

$$\leq M \sum_{l=-\infty}^{\infty} |h_{k-l}|. \quad (4.34)$$

Thus the output will be bounded for every bounded input if

$$\sum_{l=-\infty}^{\infty} |h_{k-l}| < \infty. \quad (4.35)$$

This condition is also necessary, for if we consider the bounded (by 1!) input

$$\begin{aligned} e_l &= \frac{h_{-l}}{|h_{-l}|} & (h_{-l} \neq 0) \\ &= 0 & (h_{-l} = 0) \end{aligned}$$

and apply it to Eq. (4.32), the output at $k = 0$ is

$$\begin{aligned} u_0 &= \sum_{l=-\infty}^{\infty} e_l h_{-l} \\ &= \sum_{l=-\infty}^{\infty} \frac{(h_{-l})^2}{|h_{-l}|} \\ &= \sum_{l=-\infty}^{\infty} |h_{-l}|. \end{aligned} \quad (4.36)$$

Thus, unless the condition given by Eq. (4.35) is true, the system is not BIBO stable.

◆ Example 4.3 Integration Stability

Is the discrete approximation to integration (Eq. 4.7) BIBO stable?

Solution. The test given by Eq. (4.35) can be applied to the unit pulse response used to compute the u_k -column in Table 4.1. The result is

$$\begin{aligned} h_0 &= T/2 \\ h_k &= T, \quad k > 0 \\ \sum |h_k| &= T/2 + \sum_{k=1}^{\infty} T = \text{unbounded}. \end{aligned} \quad (4.37)$$

Therefore, this discrete approximation to integration is not BIBO stable! ◆

◆ Example 4.4 General Difference Equation Stability

Consider the difference equation (4.2) with all coefficients except a_1 and b_0 equal to zero

$$u_k = a_1 u_{k-1} + b_0 e_k. \quad (4.38)$$

Is this equation stable?

Solution. The unit-pulse response is easily developed from the first few terms to be

$$\begin{aligned} u_0 &= b_0, & u_1 &= a_1 b_0, & u_2 &= a_1^2 b_0, \dots \\ u_k &= h_k = b_0 a_1^k, & k \geq 0. \end{aligned} \quad (4.39)$$

Applying the test, we have

$$\begin{aligned} \sum_{l=-\infty}^{\infty} |h_l| &= \sum_{l=0}^{\infty} b_0 |a_1|^l = b_0 \frac{1}{1 - |a_1|} & (|a_1| < 1) \\ &= \text{unbounded} & (|a_1| \geq 1). \end{aligned}$$

Thus we conclude that the system described by this equation is BIBO stable if $|a_1| < 1$, and unstable otherwise. ◆

For a more general rational transfer function with many simple poles, we can expand the function in partial fractions about its poles, and the corresponding pulse response will be a sum of respective terms. As we saw earlier, if a pole

is inside the unit circle, the corresponding pulse response decays with time geometrically and is stable. Thus, if all poles are inside the unit circle, the system with rational transfer function is stable; if at least one pole is on or outside the unit circle, the corresponding system is not BIBO stable. With modern computer programs available, finding the poles of a particular transfer function is no big deal. Sometimes, however, we wish to test for stability of an entire class of systems; or, as in an adaptive control system, the potential poles are constantly changing and we wish to have a quick test for stability in terms of the literal polynomial coefficients. In the continuous case, such a test was provided by Routh; in the discrete case, the most convenient such test was worked out by Jury and Blanchard(1961).⁹

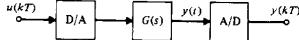
4.3 Discrete Models of Sampled-Data Systems

The systems and signals we have studied thus far have been defined in discrete time only. Most of the dynamic systems to be controlled, however, are continuous systems and, if linear, are described by continuous transfer functions in the Laplace variable s . The interface between the continuous and discrete domains are the A/D and the D/A converters as shown in Fig. 1.1. In this section we develop the analysis needed to compute the discrete transfer function between the samples that come from the digital computer to the D/A converter and the samples that are picked up by the A/D converter.¹⁰ The situation is drawn in Fig. 4.12.

4.3.1 Using the z -Transform

We wish to find the discrete transfer function from the input samples $u(kT)$ (which probably come from a computer of some kind) to the output samples $y(kT)$ picked up by the A/D converter. Although it is possibly confusing at first, we follow convention and call the discrete transfer function $G(z)$ when the continuous transfer function is $G(s)$. Although $G(z)$ and $G(s)$ are entirely different functions, they do describe the *same* plant, and the use of s for the continuous transform and z for the discrete transform is always maintained. To

Figure 4.12
The prototype sampled-data system



⁹ See Franklin, Powell, and Workman, 2nd edition, 1990, for a discussion of the Jury test.

¹⁰ In Chapter 5, a comprehensive frequency analysis of sampled data systems is presented. Here we undertake only the special problem of finding the sample-to-sample discrete transfer function of a continuous system between a D/A and an A/D.

ZOH

find $G(z)$ we need only observe that the $y(kT)$ are samples of the plant output when the input is from the D/A converter. As for the D/A converter, we assume that this device, commonly called a zero-order hold or ZOH, accepts a sample $u(kT)$ at $t = kT$ and holds its output constant at this value until the next sample is sent at $t = kT + T$. The piecewise constant output of the D/A is the signal $u(t)$, that is applied to the plant.

Our problem is now really quite simple because we have just seen that the discrete transfer function is the z -transform of the samples of the output when the input samples are the unit pulse at $k = 0$. If $u(kT) = 1$ for $k = 0$ and $u(kT) = 0$ for $k \neq 0$, the output of the D/A converter is a pulse of width T seconds and height 1, as sketched in Fig. 4.13. Mathematically, this pulse is given by $1(t) - 1(t - T)$. Let us call the particular output in response to the pulse shown in Fig. 4.13 $y_i(t)$. This response is the difference between the step response [to $1(t)$] and the delayed step response [to $1(t - T)$]. The Laplace transform of the step response is $G(s)/s$. Thus in the transform domain the unit pulse response of the plant is

$$Y_i(s) = (1 - e^{-Ts}) \frac{G(s)}{s}, \quad (4.40)$$

and the required transfer function is the z -transform of the samples of the inverse of $Y_i(s)$, which can be expressed as

$$\begin{aligned} G(z) &= \mathcal{Z}\{Y_i(kT)\} \\ &= \mathcal{Z}\{\mathcal{L}^{-1}\{Y_i(s)\}\} \triangleq \mathcal{Z}\{Y_i(s)\} \\ &= \mathcal{Z}\{(1 - e^{-Ts}) \frac{G(s)}{s}\}. \end{aligned}$$

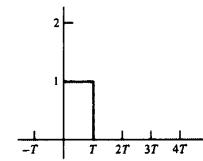
This is the sum of two parts. The first part is $\mathcal{Z}\{\frac{G(s)}{s}\}$, and the second is

$$\mathcal{Z}\{e^{-Ts} \frac{G(s)}{s}\} = z^{-1} \mathcal{Z}\{\frac{G(s)}{s}\}$$

because e^{-Ts} is exactly a delay of one period. Thus the transfer function is

$$G(z) = (1 - z^{-1}) \mathcal{Z}\left\{\frac{G(s)}{s}\right\}. \quad (4.41)$$

Figure 4.13
D/A output for unit-pulse input



◆ Example 4.5 Discrete Transfer Function of 1st-Order System

What is the discrete transfer function of

$$G(s) = a/(s + a)$$

preceded by a ZOH?

Solution. We will apply the formula (4.41)

$$\frac{G(s)}{s} = \frac{a}{s(s + a)} = \frac{1}{s} - \frac{1}{s + a},$$

and the corresponding time function is

$$\mathcal{L}^{-1} \left\{ \frac{G(s)}{s} \right\} = 1(t) - e^{-at} 1(t).$$

The samples of this signal are $1(kT) - e^{-akT} 1(kT)$, and the z -transform of these samples is

$$\begin{aligned} \mathcal{Z} \left\{ \frac{G(s)}{s} \right\} &= \frac{z}{z-1} - \frac{z}{z - e^{-aT}} \\ &= \frac{z(1 - e^{-aT})}{(z-1)(z - e^{-aT})}. \end{aligned}$$

We could have gone to the tables in Appendix B and found this result directly as Entry 12. Now we can compute the desired transform

$$\begin{aligned} G(z) &= \frac{z-1}{z} \frac{z(1 - e^{-aT})}{(z-1)(z - e^{-aT})} \\ &= \frac{1 - e^{-aT}}{z - e^{-aT}}. \end{aligned} \quad (4.42)$$

◆ Example 4.6 Discrete Transfer Function of a $1/s^2$ Plant

What is the discrete transfer function of

$$G(s) = \frac{1}{s^2}$$

preceded by a ZOH?

Solution. We have

$$G(z) = (1 - z^{-1}) \mathcal{Z} \left\{ \frac{1}{s^2} \right\}.$$

This time we refer to the tables in Appendix B and find that the z -transform associated with $1/s^2$ is

$$\frac{T^2}{2} \frac{z(z+1)}{(z-1)^3},$$

and therefore Eq. (4.41) shows that

$$G(z) = \frac{T^2(z+1)}{2(z-1)^3}. \quad (4.43)$$

The MATLAB function, c2d.m computes Eq. (4.41) (the ZOH method is the default) as well as other discrete equivalents discussed in Chapter 6. It is able to accept the system in any of the forms.

◆ Example 4.7 Discrete Transfer Function of a $1/s^2$ Plant Using MATLAB

Use MATLAB to find the discrete transfer function of

$$G(s) = \frac{1}{s^2}$$

preceded by a ZOH, assuming the sample period is $T = 1$ sec.

Solution. The MATLAB script

```
T = 1
numC = 1, denC = [1 0 0]
sysC = tf(numC,denC)
sysD = c2d(sysC,T)
[numD,denD,T] = tfdata(sysD)
```

produces the result that

$$\text{numD} = [0 \ 0.5 \ 0.5] \quad \text{and} \quad \text{denD} = [1 \ -2 \ 1]$$

which means that

$$G(z) = \frac{0z^2 + 0.5z + 0.5}{z^2 - 2z + 1} = 0.5 \frac{z+1}{(z-1)^2}$$

which is the same as Eq. (4.43) with $T = 1$.

4.3.2 *Continuous Time Delay

We now consider computing the discrete transfer function of a continuous system preceded by a ZOH with pure time delay. The responses of many chemical process-control plants exhibit pure time delay because there is a finite time of transport of fluids or materials between the process and the controls and/or the

sensors. Also, we must often consider finite computation time in the digital controller, and this is exactly the same as if the process had a pure time delay. With the techniques we have developed here, it is possible to obtain the discrete transfer function of such processes exactly, as Example 4.8 illustrates.

◆ Example 4.8 Discrete Transfer Function of 1st-Order System with Delay

Find the discrete transfer function of the mixer in Appendix A.3 with $a = 1$, $T = 1$, and $\lambda = 1.5$.

Solution. The fluid mixer problem in Appendix A.3 is described by

$$G(s) = e^{-\lambda s} H(s).$$

The term $e^{-\lambda s}$ represents the delay of λ seconds, which includes both the process delay and the computation delay, if any. We assume that $H(s)$ is a rational transfer function. To prepare this function for computation of the z -transform, we first define an integer ℓ and a positive number m less than 1.0 such that $\lambda = \ell T - mT$. With these definitions we can write

$$\frac{G(s)}{s} = e^{-\lambda s} \frac{e^{mT}}{s} H(s).$$

Because ℓ is an integer, this term reduces to $z^{-\ell}$ when we take the z -transform. Because $m < 1$, the transform of the other term is quite direct. We select $H(s) = a/(s + a)$ and, after the partial fraction expansion of $H(s)/s$, we have

$$G(z) = \frac{z-1}{z^{\ell+1}} \mathcal{Z} \left\{ \frac{e^{mT}}{s} \right\} = \frac{z-1}{z^{\ell+1}} \left\{ \frac{e^{mT}}{s+a} \right\}.$$

To complete the transfer function, we need the z -transforms of the inverses of the terms in the braces. The first term is a unit step shifted left by mT seconds, and the second term is an exponential shifted left by the same amount. Because $m < 1$, these shifts are less than one full period, and no sample is picked up in negative time. The signals are sketched in Fig. 4.14.

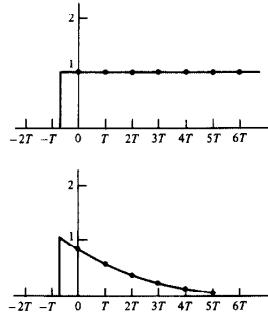
The samples are given by $1(4T)$ and $z^{-\ell} e^{-amT}(1(4T))$. Consequently the final transfer function is

$$\begin{aligned} G(z) &= \frac{z-1}{z^{\ell+1}} \left\{ \frac{z}{z-1} - \frac{ze^{-amT}}{z-e^{-amT}} \right\} \\ &= \frac{z-1}{z^{\ell+1}} \left\{ \frac{z(z-e^{-amT})(z-1)e^{-amT}}{(z-1)(z-e^{-amT})} \right\} \\ &= (1 - e^{-amT}) \frac{z+\alpha}{z^{\ell+1}(z-e^{-amT})} \end{aligned}$$

where the zero position is at $-\alpha = -(e^{-amT} - e^{-amT})/(1 - e^{-amT})$. Notice that this zero is near the origin of the z -plane when m is near 1 and moves outside the unit circle to near $-\infty$ when m approaches 0. For specific values of the mixer, we take $a = 1$, $T = 1$, and $\lambda = 1.5$. Then we can compute that $\ell = 2$ and $m = 0.5$. For these values, we get

$$G(z) = \frac{z+0.6025}{z^3(z-0.3679)}. \quad (4.44)$$

Figure 4.14
Sketch of the shifted
signals showing sample
points



In MATLAB, the transfer function for this system would be computed by

```
Td = 1.5, a = 1, T = 1
sysC = tf([1 - a], 'td', Td)
sysD = c2d(sysC, T)
```

4.3.3 State-Space Form

Computing the z -transform using the Laplace transform as in Eq. (4.41) is a very tedious business that is unnecessary with the availability of computers. We will next develop a formula using state descriptions that moves the tedium to the computer. A continuous, linear, constant-coefficient system of differential equations was expressed in Eq. (2.1) as a set of first-order matrix differential equations. For a scalar input, it becomes

$$\dot{x} = Fx + Gu + G_1w. \quad (4.45)$$

where u is the scalar control input to the system and w is a scalar disturbance input. The output was expressed in Eq. (2.2) as a linear combination of the state, x , and the input, u , which becomes for scalar output

$$y = Hx + Ju. \quad (4.46)$$

Often the sampled-data system being described is the plant of a control problem, and the parameter J in Eq. (4.46) is zero and will frequently be omitted.

◆ Example 4.9 State Representation of a $1/s^2$ Plant

Apply Eqs. (4.45) and (4.46) to the double integrator plant of the satellite control problem in Appendix A.1

$$G(s) = \frac{1}{s^2}.$$

Solution. The satellite attitude-control example is shown in block diagram form in Fig. 4.15 and the attitude (θ) and attitude rate ($\dot{\theta}$) are defined to be x_1 and x_2 , respectively. Therefore, the equations of motion can be written as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_F \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_G u.$$

$$\theta = y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_H \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.47)$$

which, in this case, turns out to be a rather involved way of writing

$$\ddot{\theta} = u.$$

The representations given by Eqs. (4.45) and (4.46) are not unique. Given one state representation, any nonsingular linear transformation of that state such as $Bx = Tx$ is also an allowable alternative realization of the same system.

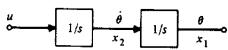
If we let $\xi = Tx$ in Eqs. (4.45) and (4.46), we find

$$\begin{aligned} \dot{\xi} &= T\dot{x} = T(Fx + Gu + G_i w) \\ &= TFx + TGu + TG_i w, \\ \dot{\xi} &= TFT^{-1}\xi + TGu + TG_i w, \\ y &= T^{-1}\xi + Ju. \end{aligned}$$

If we designate the system matrices for the new state ξ as A , B , C , and D , then

$$\dot{\xi} = Ax + Bu + B_i w, \quad y = C\xi + Du,$$

Figure 4.15
Satellite attitude control
in classical
representation



where

$$A = TFT^{-1}, \quad B = TG, \quad B_i = TG_i, \quad C = T^{-1}, \quad D = J.$$

◆ Example 4.10 State Transformation for $1/s^2$ Plant

Find the state representation for the case with the state definitions of the previous example interchanged.

Solution. Let $\xi_1 = x_2$ and $\xi_2 = x_1$ in Eq. (4.47); or, in matrix notation, the transformation to interchange the states is

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

In this case $T^{-1} = T$, and application of the transformation equations to the system matrices of Eq. (4.47) gives

$$A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = [0 \ 1].$$

Most often, a change of state is made to bring the description matrices into a useful canonical form. We saw earlier how a single high-order difference equation could be represented by a state description in control or in observer canonical form. Also, there is a very useful state description corresponding to the partial-fraction expansion of a transfer function. State transformations can take a general description for either a continuous or a discrete system and, subject to some technical restrictions, convert it into a description in one or the other of these forms, as needed.

We wish to use the state description to establish a general method for obtaining the difference equations that represent the behavior of the continuous plant. Fig. 4.16 again depicts the portion of our system under consideration. Ultimately, the digital controller will take the samples $v(k)$, operate on that sequence by means of a difference equation, and put out a sequence of numbers, $u(k)$, which are the inputs to the plant. The loop will, therefore, be closed. To analyze the result, we must be able to relate the samples of the output $y(k)$ to the samples of the control $u(k)$. To do this, we must solve Eq. (4.45).

We will solve the general equation in two steps. We begin by solving the equation with only initial conditions and no external input. This is the homogeneous equation

$$\dot{x}_n = Fx_n(t), \quad x_n(t_0) = x_0. \quad (4.48)$$

Figure 4.16
System definition with sampling operations shown



To solve this, we assume the solution is sufficiently smooth that a series expansion of the solution is possible

$$\mathbf{x}_b(t) = \mathbf{A}_0 + \mathbf{A}_1(t - t_0) + \mathbf{A}_2(t - t_0)^2 + \dots \quad (4.49)$$

If we let $t = t_0$, we find immediately that $\mathbf{A}_0 = \mathbf{x}_b$. If we differentiate Eq. (4.49) and substitute into Eq. (4.48), we have

$$\mathbf{A}_1 + 2\mathbf{A}_2(t - t_0) + 3\mathbf{A}_3(t - t_0)^2 + \dots = \mathbf{F}\mathbf{x}_b$$

and, at $t = t_0$, $\mathbf{A}_1 = \mathbf{F}\mathbf{x}_b$. Now we continue to differentiate the series and the differential equation and equate them at t_0 to arrive at the series

$$\mathbf{x}_b(t) = \left[\mathbf{I} + \mathbf{F}(t - t_0) + \frac{\mathbf{F}^2(t - t_0)^2}{2!} + \frac{\mathbf{F}^3(t - t_0)^3}{3!} + \dots \right] \mathbf{x}_b.$$

This series is defined as the matrix exponential and written

$$\mathbf{x}_b(t) = e^{\mathbf{F}(t-t_0)}\mathbf{x}(t_0), \quad (4.50)$$

where, by definition, the matrix exponential is

$$\begin{aligned} e^{\mathbf{F}(t-t_0)} &= \mathbf{I} + \mathbf{F}(t - t_0) + \mathbf{F}^2 \frac{(t - t_0)^2}{2!} + \mathbf{F}^3 \frac{(t - t_0)^3}{3!} + \dots \\ &= \sum_{k=0}^{\infty} \mathbf{F}^k \frac{(t - t_0)^k}{k!}. \end{aligned} \quad (4.51)$$

It can be shown that the solution given by Eq. (4.50) is unique, which leads to very interesting properties of the matrix exponential. For example, consider two values of t : t_1 and t_2 . We have

$$\mathbf{x}(t_1) = e^{\mathbf{F}(t_1-t_0)}\mathbf{x}(t_0)$$

and

$$\mathbf{x}(t_2) = e^{\mathbf{F}(t_2-t_0)}\mathbf{x}(t_0).$$

Because t_0 is arbitrary also, we can express $\mathbf{x}(t_2)$ as if the equation solution began at t_1 , for which

$$\mathbf{x}(t_2) = e^{\mathbf{F}(t_2-t_1)}\mathbf{x}(t_1).$$

Substituting for $\mathbf{x}(t_1)$ gives

$$\mathbf{x}(t_2) = e^{\mathbf{F}(t_2-t_1)}e^{\mathbf{F}(t_1-t_0)}\mathbf{x}(t_0).$$

We now have two separate expressions for $\mathbf{x}(t_2)$, and if the solution is unique, these must be the same. Hence we conclude that

$$e^{\mathbf{F}(t_2-t_0)} = e^{\mathbf{F}(t_2-t_1)}e^{\mathbf{F}(t_1-t_0)} \quad (4.52)$$

for all t_2, t_1, t_0 . Note especially that if $t_2 = t_0$, then

$$\mathbf{I} = e^{-\mathbf{F}(t_1-t_0)}e^{\mathbf{F}(t_1-t_0)}.$$

Thus we can obtain the inverse of $e^{\mathbf{F}t}$ by merely changing the sign of t . We will use this result in computing the particular solution to Eq. (4.45).

The particular solution when u is not zero is obtained by using the method of **variation of parameters**.¹¹ We guess the solution to be in the form

$$\mathbf{x}_p(t) = e^{\mathbf{F}(t-t_0)}v(t), \quad (4.53)$$

where $v(t)$ is a vector of variable parameters to be determined [as contrasted to the constant parameters $\mathbf{x}(t_0)$ in Eq. (4.50)]. Substituting Eq. (4.53) into Eq. (4.45), we obtain

$$\mathbf{F}e^{\mathbf{F}(t-t_0)}v + e^{\mathbf{F}(t-t_0)}v = \mathbf{F}e^{\mathbf{F}(t-t_0)}v + \mathbf{Gu},$$

and, using the fact that the inverse is found by changing the sign of the exponent, we can solve for v as

$$\dot{v}(t) = e^{-\mathbf{F}(t-t_0)}\mathbf{Gu}(t).$$

Assuming that the control $u(t)$ is zero for $t < t_0$, we can integrate \dot{v} from t_0 to t to obtain

$$v(t) = \int_{t_0}^t e^{-\mathbf{F}(t-\tau)}\mathbf{Gu}(\tau)d\tau.$$

Hence, from Eq. (4.53), we get

$$\mathbf{x}_p(t) = e^{\mathbf{F}(t-t_0)} \int_{t_0}^t e^{-\mathbf{F}(t-\tau)}\mathbf{Gu}(\tau)d\tau,$$

and simplifying, using the results of Eq. (4.52), we obtain the particular solution (convolution)

$$\mathbf{x}_p(t) = \int_{t_0}^t e^{\mathbf{F}(t-\tau)}\mathbf{Gu}(\tau)d\tau. \quad (4.54)$$

The total solution for $w = 0$ and $u \neq 0$ is the sum of Eqs. (4.50) and (4.54):

$$\mathbf{x}(t) = e^{\mathbf{F}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^t e^{\mathbf{F}(t-\tau)}\mathbf{Gu}(\tau)d\tau. \quad (4.55)$$

¹¹ Due to Joseph Louis Lagrange, French mathematician (1736–1813). We assume $w = 0$, but because the equations are linear, the effect of w can be added later.

We wish to use this solution over one sample period to obtain a difference equation; hence we juggle the notation a bit (let $t = kT + T$ and $t_0 = kT$) and arrive at a particular version of Eq. (4.55):

$$\mathbf{x}(kT + T) = e^{\mathbf{F}T}\mathbf{x}(kT) + \int_{kT}^{kT+T} e^{\mathbf{F}(kT+\tau-t)}\mathbf{G}u(\tau)d\tau. \quad (4.56)$$

This result is not dependent on the type of hold because u is specified in terms of its continuous time history, $u(t)$, over the sample interval. A common and typically valid assumption is that of a zero-order hold (ZOH) with no delay, that is,

$$u(\tau) = u(kT), \quad kT \leq \tau < kT + T.$$

If some other hold is implemented or if there is a delay between the application of the control from the ZOH and the sample point, this fact can be accounted for in the evaluation of the integral in Eq. (4.56). The equations for a delayed ZOH will be given in the next subsection. To facilitate the solution of Eq. (4.56) for a ZOH with no delay, we change variables in the integral from τ to η such that

$$\eta = kT + T - \tau.$$

Then we have

$$\mathbf{x}(kT + T) = e^{\mathbf{F}T}\mathbf{x}(kT) + \int_0^T e^{\mathbf{F}\eta}d\eta \mathbf{G}u(kT). \quad (4.57)$$

If we define

$$\begin{aligned} \Phi &= e^{\mathbf{F}T} \\ \Gamma &= \int_0^T e^{\mathbf{F}\eta}d\eta \mathbf{G}, \end{aligned} \quad (4.58)$$

Eqs. (4.57) and (4.46) reduce to difference equations in standard form

$$\begin{aligned} \mathbf{x}(k+1) &= \Phi\mathbf{x}(k) + \Gamma u(k) + \Gamma_1 w(k), \\ y(k) &= \mathbf{H}\mathbf{x}(k), \end{aligned} \quad (4.59)$$

where we include the effect of an impulsive or piecewise constant disturbance, w , and assume that $J = 0$ in this case. If w is a constant, then Γ_1 is given by Eq. (4.58) with \mathbf{G} replaced by \mathbf{G}_1 . If w is an impulse, then $\Gamma_1 = \mathbf{G}_1$.¹² The Φ series expansion

$$\Phi = e^{\mathbf{F}T} = \mathbf{I} + \mathbf{FT} + \frac{\mathbf{F}^2T^2}{2!} + \frac{\mathbf{F}^3T^3}{3!} + \dots$$

¹² If $w(t)$ varies significantly between its sample values, then an integral like that of Eq. (4.56) is required to describe its influence on $\mathbf{x}(k+1)$. Random disturbances are treated in Chapter 9.

can also be written

$$\Phi = \mathbf{I} + \mathbf{FT}\Psi, \quad (4.60)$$

where

$$\Psi = \mathbf{I} + \frac{\mathbf{FT}}{2!} + \frac{\mathbf{F}^2T^2}{3!} + \dots$$

The Γ integral in Eq. (4.58) can be evaluated term by term to give

$$\begin{aligned} \Gamma &= \sum_{k=0}^{\infty} \frac{\mathbf{F}^k T^{k+1}}{(k+1)!} \mathbf{G} \\ &= \sum_{k=0}^{\infty} \frac{\mathbf{F}^k T^k}{(k+1)!} T \mathbf{G} \\ &= \Psi T \mathbf{G}. \end{aligned} \quad (4.61)$$

We evaluate Ψ by a series in the form

$$\Psi \approx \mathbf{I} + \frac{\mathbf{FT}}{2} \left(\mathbf{I} + \frac{\mathbf{FT}}{3} \left(\dots \frac{\mathbf{FT}}{N-1} \left(\mathbf{I} + \frac{\mathbf{FT}}{N} \right) \dots \right) \right). \quad (4.62)$$

which has better numerical properties than the direct series of powers. We then find Γ from Eq. (4.61) and Φ from Eq. (4.60). A discussion of the selection of N and a technique to compute Ψ for comparatively large T is given by Källström (1973), and a review of various methods is found in a classic paper by Moler and Van Loan (1978). The program logic for computation of Φ and Γ for simple cases is given in Fig. 4.17. MATLAB's `c2d.m` and all control design packages that we know of compute Φ and Γ from the continuous matrices \mathbf{F} , \mathbf{G} , and the sample period T .

To compare this method of representing the plant with the discrete transfer functions, we can take the z -transform of Eq. (4.59) with $w = 0$ and obtain

$$\begin{aligned} [z\mathbf{I} - \Phi]\mathbf{X} &= \Gamma I/(z), \\ Y(z) &= \mathbf{H}\mathbf{X}(z) \end{aligned} \quad (4.63)$$

Figure 4.17
Program logic to
compute Φ and Γ from
 \mathbf{F} , \mathbf{G} , and T for simple
cases. (The left arrow \leftarrow
is read as "is replaced
by")

1. Select sampling period T and description matrices \mathbf{F} and \mathbf{G} .
2. Matrix $\mathbf{I} \leftarrow$ Identity
3. Matrix $\Psi \leftarrow \mathbf{I}$
4. $k \leftarrow 11$ [We are using $N = 11$ in Eq. (4.62).]
5. If $k = 1$, go to step 9.
6. Matrix $\Psi \leftarrow \mathbf{I} + \frac{\mathbf{FT}}{k} \Psi$
7. $k \leftarrow k + 1$
8. Go to step 5.
9. Matrix $\Gamma \leftarrow T \Psi \mathbf{G}$
10. Matrix $\Phi \leftarrow \mathbf{I} + \mathbf{FT}\Psi$

therefore

$$\frac{Y(z)}{U(z)} = H[zI - \Phi]^{-1}\Gamma. \quad (4.64)$$

◆ Example 4.11 Φ and Γ Calculation

By hand, calculate the Φ and Γ matrices for the satellite attitude-control system of Example 4.9.

Solution. Use Eqs. (4.60) and (4.61) and the values for F and G defined in Eq. (4.47). Since $F^2 = 0$ in this case, we have

$$\begin{aligned} \Phi &= I + FT + \frac{F^2T^2}{2!} + \dots \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}T = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \\ \Gamma &= \left[IT + F \frac{T^2}{2!} + \frac{F^2T^3}{3!} \right] G \\ &= \left[\begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \frac{T^2}{2} \right] \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}. \end{aligned}$$

hence, using Eq. (4.64), we obtain

$$\begin{aligned} \frac{Y(z)}{U(z)} &= [1 \ 0] \left\{ z \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \right\}^{-1} \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \\ &= \frac{T^2}{2} \frac{(z+1)}{(z-1)^2}, \end{aligned}$$

which is the same result that would be obtained using Eq. (4.41) and the z -transform tables. Note that the values for Φ and Γ could have been obtained for a specific value of T by the MATLAB statements

```
sysC = ss(F,G,H,J)
sysD = c2d(sysC,T)
[phi,gam,h,J] = ssdata(sysD)
```

Note that to compute Y/U we find that the denominator is the determinant $\det(zI - \Phi)$, which comes from the matrix inverse in Eq. (4.64). This determinant is the characteristic polynomial of the transfer function, and the zeros of the determinant are the poles of the plant. We have two poles at $z = 1$ in this case, corresponding to the two integrations in this plant's equations of motion.

We can explore further the question of poles and zeros and the state-space description by considering again the transform equations (4.63). An interpretation of transfer-function poles from the perspective of the corresponding difference equation is that a pole is a value of z such that the equation has a nontrivial solution when the forcing input is zero. From Eq. (4.63a), this implies that the linear eigenvalue equations

$$[zI - \Phi]X(z) = [0]$$

have a nontrivial solution. From matrix algebra the well-known requirement for this is that $\det(zI - \Phi) = 0$. Using the Φ from the previous example, we have

$$\begin{aligned} \det[zI - \Phi] &= \det \left[\begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \right] \\ &= \det \begin{bmatrix} z-1 & -T \\ 0 & z-1 \end{bmatrix} \\ &= (z-1)^2 = 0, \end{aligned}$$

which is the characteristic equation, as we have seen. To compute the poles numerically when the matrices are given, one would use an eigenvalue routine. In MATLAB, the statement

`lam=eig(phi)`

will produce a vector, `lam`, of the poles of Φ .

Along the same line of reasoning, a system zero is a value of z such that the system output is zero even with a nonzero state-and-input combination. Thus if we are able to find a nontrivial solution for $X(z_0)$ and $U(z_0)$ such that $Y(z_0)$ is zero, then z_0 is a zero of the system. Combining the two parts of Eq. (4.59), we must satisfy the requirement

$$\begin{bmatrix} zI - \Phi & -\Gamma \\ H & 0 \end{bmatrix} \begin{bmatrix} X(z) \\ U(z) \end{bmatrix} = [0]. \quad (4.65)$$

Once more the condition for the existence of nontrivial solutions is that the determinant of the square coefficient system matrix be zero.¹³ For the satellite example, we have

$$\begin{aligned} \det \begin{bmatrix} z-1 & -T & -T^2/2 \\ 0 & z-1 & -T \\ 1 & 0 & 0 \end{bmatrix} &= 1 \cdot \det \begin{bmatrix} -T & -T^2/2 \\ z-1 & -T \end{bmatrix} \\ &= +T^2 + \left(\frac{T^2}{2}\right)(z-1) \end{aligned}$$

¹³ We do not consider here the case of different numbers of inputs and outputs.

$$= + \frac{T^2}{2} z + \frac{T^2}{2}$$

$$= + \frac{T^2}{2} (z + 1).$$

Thus we have a single zero at $z = -1$, as we have seen from the transfer function. These zeros are called **transmission zeros** and are easily computed using MATLAB's tzero.m.¹⁴ Using the discrete model sysD found in Example 4.11 the statement

```
zer=tzero(sysD)
```

produces the transmission zeros in the quantity zer.

4.3.4 *State-Space Models for Systems with Delay

Thus far we have discussed the calculation of discrete state models from continuous, ordinary differential equations of motion. Now we present the formulas for including a time delay in the model and also a time prediction up to one period which corresponds to the **modified z-transform** as defined by Jury. We begin with a state-variable model that includes a delay in control action. The state equations are

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{G}u(t - \lambda), \quad (4.66)$$

$$\mathbf{y} = \mathbf{H}\mathbf{x}.$$

The general solution to Eq. (4.66) is given by Eq. (4.55); it is

$$\mathbf{x}(t) = e^{\mathbf{F}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^t e^{\mathbf{F}(t-\tau)}\mathbf{G}u(\tau - \lambda) d\tau.$$

If we let $t_0 = kT$ and $t = kT + T$, then

$$\mathbf{x}(kT + T) = e^{\mathbf{F}T}\mathbf{x}(kT) + \int_{kT}^{kT+T} e^{\mathbf{F}(kT+\tau-t)}\mathbf{G}u(\tau - \lambda) d\tau.$$

If we substitute $\eta = kT + T - \tau$ for τ in the integral, we find a modification of Eq. (4.57)

$$\mathbf{x}(kT + T) = e^{\mathbf{F}T}\mathbf{x}(kT) + \int_T^0 e^{\mathbf{F}\eta}\mathbf{G}u(kT + T - \lambda - \eta)(-d\eta)$$

$$= e^{\mathbf{F}T}\mathbf{x}(kT) + \int_0^T e^{\mathbf{F}\eta}\mathbf{G}u(kT + T - \lambda - \eta) d\eta.$$

¹⁴ In using this function, one must be careful to account properly for the zeros that are at infinity; the function might return them as very large numbers that the user must remove to "uncover" the finite zeros; that is, to scale the finite numbers so they don't appear to be zero by the computer.

If we now separate the system delay λ into an integral number of sampling periods plus a fraction, we can define an integer ℓ and a positive number m less than one such that

$$\lambda = \ell T - mT, \quad (4.67)$$

and

$$\begin{array}{c} \ell \geq 0, \\ 0 \leq m < 1. \end{array}$$

With this substitution, we find that the discrete system is described by

$$\mathbf{x}(kT + T) = e^{\mathbf{F}T}\mathbf{x}(kT) + \int_0^T e^{\mathbf{F}\eta}\mathbf{G}u(kT + T - \ell T + mT - \eta) d\eta. \quad (4.68)$$

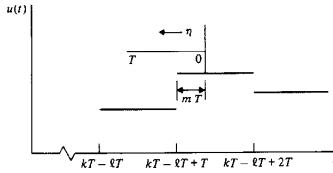
If we sketch a segment of the time axis near $t = kT - \ell T$ (Fig. 4.18), the nature of the integral in Eq. (4.68) with respect to the variable η will become clear. The integral runs for η from 0 to T , which corresponds to t from $kT - \ell T + T + mT$ backward to $kT - \ell T + mT$. Over this period, the control, which we assume is piecewise constant, takes on first the value $u(kT - \ell T + T)$ and then the value $u(kT - \ell T)$. Therefore, we can break the integral in (2.66) into two parts as follows

$$\begin{aligned} \mathbf{x}(kT + T) &= e^{\mathbf{F}T}\mathbf{x}(kT) + \int_0^{mT} e^{\mathbf{F}\eta}\mathbf{G}d\eta u(kT - \ell T + T) \\ &\quad + \int_{mT}^T e^{\mathbf{F}\eta}\mathbf{G}d\eta u(kT - \ell T) \\ &= \Phi\mathbf{x}(kT) + \Gamma_1 u(kT - \ell T) + \Gamma_2 u(kT - \ell T + T). \end{aligned} \quad (4.69)$$

In Eq. (4.69) we defined

$$\Phi = e^{\mathbf{F}T}, \quad \Gamma_1 = \int_{mT}^T e^{\mathbf{F}\eta}\mathbf{G}d\eta, \quad \text{and} \quad \Gamma_2 = \int_0^{mT} e^{\mathbf{F}\eta}\mathbf{G}d\eta. \quad (4.70)$$

Figure 4.18
Sketch of a piecewise input and time axis for a system with time delay



To complete our analysis it is necessary to express Eq. (4.69) in standard state-space form. To do this we must consider separately the cases of $\ell = 0$, $\ell = 1$, and $\ell > 1$.

For $\ell = 0$, $\lambda = -mT$ according to Eq. (4.67), which implies no delay but prediction. Because mT is restricted to be less than T , however, the output will not show a sample before $k = 0$, and the discrete system will be causal. The result is that the discrete system computed with $\ell = 0$, $m \neq 0$ will show the response at $t = 0$, which the same system with $\ell = 0$, $m = 0$ would show at $t = mT$. In other words, by taking $\ell = 0$ and $m \neq 0$ we pick up the response values between the normal sampling instants. In z -transform theory, the transform of the system with $\ell = 0$, $m \neq 0$ is called the **modified z -transform**.¹⁵ The state-variable form requires that we evaluate the integrals in Eq. (4.70). To do so we first convert Γ_1 to a form similar to the integral for Γ_2 . From Eq. (4.70) we factor out the constant matrix \mathbf{G} to obtain

$$\Gamma_1 = \int_{-mT}^T e^{\mathbf{F}\eta} d\eta \mathbf{G}.$$

If we set $\sigma = \eta - mT$ in this integral, we have

$$\begin{aligned} \Gamma_1 &= \int_0^{T-mT} e^{\mathbf{F}(mT+\sigma)} d\sigma \mathbf{G} \\ &= e^{\mathbf{F}m} \int_0^{T-mT} e^{\mathbf{F}\sigma} d\sigma \mathbf{G}. \end{aligned} \quad (4.71)$$

For notational purposes we will define, for any positive nonzero scalar number, a , the two matrices

$$\Phi(a) = e^{\mathbf{F}a}, \quad \Psi(a) = \frac{1}{a} \int_0^a e^{\mathbf{F}\sigma} d\sigma. \quad (4.72)$$

In terms of these matrices, we have

$$\begin{aligned} \Gamma_1 &= (T - mT)\Phi(mT)\Psi, \\ \Gamma_2 &= mT\Psi. \end{aligned} \quad (4.73)$$

The definitions in Eqs. (4.72) are also useful from a computational point of view. If we recall the series definition of the matrix exponential

$$\Phi(a) = e^{\mathbf{F}a} = \sum_{k=0}^{\infty} \frac{\mathbf{F}^k a^k}{k!},$$

then we get

$$\begin{aligned} \Psi(a) &= \frac{1}{a} \int_0^a \sum_{k=0}^{\infty} \frac{\mathbf{F}^k a^k}{k!} d\sigma \\ &= \frac{1}{a} \sum_{k=0}^{\infty} \frac{\mathbf{F}^k}{k!} \frac{a^{k-1}}{k+1} \end{aligned}$$

¹⁵ See Jury (1964) or Ogata (1987).

$$= \sum_{k=0}^{\infty} \frac{\mathbf{F}^k a^k}{(k+1)!}. \quad (4.74)$$

But now we note that the series for $\Phi(a)$ can be written as

$$\Phi(a) = \mathbf{I} + \sum_{k=1}^{\infty} \frac{\mathbf{F}^k a^k}{k!}.$$

If we let $k = j + 1$ in the sum, then, as in Eq. (4.60), we have

$$\begin{aligned} \Phi(a) &= \mathbf{I} + \sum_{j=0}^{\infty} \frac{\mathbf{F}^{j+1} a^{j+1}}{(j+1)!} \\ &= \mathbf{I} + \sum_{j=0}^{\infty} \frac{\mathbf{F}^j a^j}{(j+1)!} a \mathbf{F} \\ &= \mathbf{I} + a \Psi(a) \mathbf{F}. \end{aligned} \quad (4.75)$$

The point of Eq. (4.75) is that only the series for $\Psi(a)$ needs to be computed and from this single sum we can compute Φ and Γ .

If we return to the case $\ell = 0$, $m \neq 0$, the discrete state equations are

$$\mathbf{x}(k+1) = \Phi \mathbf{x}(k) + \Gamma_1 u(k) + \Gamma_2 u(k+1),$$

where Γ_1 and Γ_2 are given by Eq. (4.73). In order to put these equations in state-variable form, we must eliminate the term in $u(k+1)$. To do this, we define a new state, $\xi(k) = \mathbf{x}(k) - \Gamma_2 u(k)$. Then the equations are

$$\begin{aligned} \xi(k+1) &= \mathbf{x}(k+1) - \Gamma_2 u(k+1) \\ &= \Phi \mathbf{x}(k) + \Gamma_1 u(k) + \Gamma_2 u(k+1) - \Gamma_2 u(k+1), \\ \xi(k+1) &= \Phi \xi(k) + \Gamma_2 u(k) + \Gamma_1 u(k) \\ &= \Phi \xi(k) + (\Phi \Gamma_2 + \Gamma_1) u(k) \\ &= \Phi \xi(k) + \Gamma u(k). \end{aligned} \quad (4.76)$$

The output equation is

$$\begin{aligned} \mathbf{y}(k) &= \mathbf{H} \mathbf{x}(k) \\ &= \mathbf{H}[\xi(k) + \Gamma_2 u(k)] \\ &= \mathbf{H} \xi(k) + \mathbf{H} \Gamma_2 u(k) \\ &= \mathbf{H}_d \xi(k) + J_d u(k). \end{aligned} \quad (4.77)$$

Thus for $\ell = 0$, the state equations are given by Eqs. (4.73), (4.76), and (4.77). Note especially that if $m = 0$, then $\Gamma_2 = 0$, and these equations reduce to the previous model with no delay.

Our next case is $\ell = 1$. From Eq. (4.69), the equations are given by

$$\mathbf{x}(k+1) = \Phi \mathbf{x}(k) + \Gamma_1 u(k-1) + \Gamma_2 u(k).$$

In this case, we must eliminate $u(k-1)$ from the right-hand side, which we do by defining a new state $x_{n+1}(k) = u(k-1)$. We have thus an increased dimension of the state, and the equations are

$$\begin{bmatrix} \mathbf{x}(k+1) \\ x_{n+1}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma_1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ x_{n+1}(k) \end{bmatrix} + \begin{bmatrix} \Gamma_2 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [\mathbf{H} \ 0] \begin{bmatrix} \mathbf{x} \\ x_{n+1} \end{bmatrix}. \quad (4.78)$$

For our final case, we consider $\ell > 1$. In this case, the equations are

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma_1 u(k-\ell) + \Gamma_2 u(k-\ell+1)$$

and we must eliminate the past controls up to $u(k)$. To do this we introduce ℓ new variables such that

$$x_{n+1}(k) = u(k-\ell), \quad x_{n+2}(k) = u(k-\ell+1), \quad \dots, \quad x_{n+\ell}(k) = u(k-1).$$

The structure of the equations is

$$\begin{bmatrix} \mathbf{x}(k+1) \\ x_{n+1}(k+1) \\ x_{n+2}(k+1) \\ \vdots \\ x_{n+\ell}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma_1 & \Gamma_2 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ x_{n+1}(k) \\ x_{n+2}(k) \\ \vdots \\ x_{n+\ell}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [\mathbf{H} \ 0 \ \cdots \ 0] \begin{bmatrix} \mathbf{x}(k) \\ x_{n+1}(k) \\ \vdots \\ x_{n+\ell}(k) \end{bmatrix}. \quad (4.79)$$

This final solution is easily visualized in terms of a block diagram, as shown in Fig. 4.19.

4.3.5 *Numerical Considerations in Computing Φ and Γ

The numerical considerations of these computations are centered in the approximation to the infinite sum for Ψ given by Eq. (4.74) or, for $a = T$, by Eq. (4.62). The problem is that if FT is large, then $(FT)^N/N!$ becomes extremely large before it becomes small, and before acceptable accuracy is realized most computer number representations will overflow, destroying the value of the computation. Källström (1973) has analyzed a technique used by Kalman and Englar (1966),



Figure 4.19
Block diagram of system with delay of more than one period. Double line indicates vector valued variables

which has been found effective by Moler and Van Loan (1978). The basic idea comes from Eq. (4.52) with $t_2 - t_0 = 2T$ and $t_1 - t_0 = T$, namely

$$(e^{FT})^2 = e^{FT} e^{FT} = e^{2FT}. \quad (4.80)$$

Thus, if T is too large, we can compute the series for $T/2$ and square the result. If $T/2$ is too large, we compute the series for $T/4$, and so on, until we find a k such that $T/2^k$ is not too large. We need a test for deciding on the value of k . We propose to approximate the series for Ψ , which can be written

$$\Psi\left(\frac{T}{2^k}\right) = \sum_{j=0}^{N-1} \frac{(\mathbf{F}(T/2^k))^j}{(j+1)!} + \sum_{j=N}^{\infty} \frac{(\mathbf{F}T/2^k)^j}{(j+1)!} = \hat{\Psi} + \mathbf{R}.$$

We will select k , the factor that decides how much the sample period is divided down, to yield a small remainder term \mathbf{R} . Källström suggests that we estimate the size of \mathbf{R} by the size of the first term ignored in Ψ , namely,

$$\hat{\mathbf{R}} \cong (\mathbf{F}T)^N/(N+1)2^{Nk}.$$

A simpler method is to select k such that the size of $\mathbf{F}T$ divided by 2^k is less than 1. In this case, the series for $\mathbf{F}T/2^k$ will surely converge. The rule is to select k such that

$$2^k > \|\mathbf{F}T\| = \max_j \sum_{i=1}^n |F_{ij}| T.$$

Taking the log of both sides, we find

$$k > \log_2 \|\mathbf{F}T\|,$$

from which we select

$$k = \max(\lceil \log_2 \|\mathbf{F}T\| \rceil, 0). \quad (4.81)$$

where the symbol $\lceil x \rceil$ means the smallest integer greater than x . The maximum of this integer and zero is taken because it is possible that $\|FT\|$ is already so small that its log is negative, in which case we want to select $k = 0$.

Having selected k , we now have the problem of computing $\hat{\Psi}(T)$ from $\hat{\Psi}(T/2^k)$. Our original concept was based on the series for Φ , which satisfied Eq. (4.80). To obtain the suitable formula for Ψ , we use the relation between Φ and Ψ given by Eq. (4.60) as follows to obtain the “doubling” formula for Ψ

$$\begin{aligned}\Phi(2T) &= \Phi(T)\Phi(T), \\ I + 2TF\Psi(2T) &= [I + TF\Psi(T)][I + TF\Psi(T)] \\ &= I + 2TF\Psi(T) + T^2F^2\Psi^2(T),\end{aligned}$$

therefore

$$2TF\Psi(2T) = 2TF\Psi(T) + T^2F^2\Psi^2(T).$$

This is equivalent to

$$\Psi(2T) = \left(I + \frac{T}{2}F\Psi(T) \right) \Psi(T),$$

which is the form to be used. The program logic for computing Ψ is shown in Fig. 4.20.¹⁶ This algorithm does not include the delay discussed in Section 4.3.4.

Figure 4.20

Logic for a program to compute Ψ using automatic time scaling

1. Select F and T .
2. Comment: Compute $\|FT\|$.
3. $V \leftarrow \max\{\sum_i |F_{ij}| \} \times T$
4. $k \leftarrow$ smallest nonnegative integer greater than $\log_2 V$.
5. Comment: compute $\Psi(T/2^k)$.
6. $T_j \leftarrow T/2^k$
7. $I \leftarrow$ Identity
8. $\Psi \leftarrow I$
9. $j \leftarrow 11$
10. If $j = 1$, go to step 14.
11. $\Psi \leftarrow I + \frac{FT_j}{j}\Psi$
12. $j \leftarrow j - 1$
13. Go to step 10.
14. Comment: Now double Ψ k times.
15. If $k = 0$, stop.
16. $\Psi \leftarrow (I + \frac{FT}{2^{k-1}})\Psi$
17. $k \leftarrow k - 1$
18. Go to step 15.

¹⁶ Similar logic is used by MATLAB in C2D.M.

For that, we must implement the logic shown in Fig. 4.19. In the Control Toolbox, the function c2d.m executes the logic with a delay if one is specified.

4.3.6 *Nonlinear Models

Contrary to the predominant developments in this book, models of dynamic systems are generally nonlinear. However, it is more difficult to apply analysis to nonlinear models and thus, less insight is gained if models are left in their nonlinear form throughout the entire design process. Controls engineers commonly use numerical simulation of nonlinear models to evaluate the performance of control systems, a technique that should always be a part of any control system design. To aid in the design synthesis of controllers and to gain insight into approximate behavior, it is often advantageous to linearize the system so the methods in this text can be utilized.

We begin with the assumption that our plant dynamics are adequately described by a set of ordinary differential equations in state-variable form as

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, \dots, x_n, u_1, \dots, u_m, t), \\ \dot{x}_2 &= f_2(x_1, \dots, x_n, u_1, \dots, u_m, t), \\ &\vdots \\ \dot{x}_n &= f_n(x_1, \dots, x_n, u_1, \dots, u_m, t), \\ y_1 &= h_1(x_1, \dots, x_n, u_1, \dots, u_m, t), \\ &\vdots \\ y_p &= h_p(x_1, \dots, x_n, u_1, \dots, u_m, t).\end{aligned}\quad (4.82)$$

or, more compactly in matrix notation, we assume that our plant dynamics are described by

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \\ \mathbf{x}(t_0) &= \mathbf{x}_0, \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}, \mathbf{u}, t).\end{aligned}\quad (4.83)$$

One proceeds as follows with the process of linearization and small-signal approximations. We assume stationarity by the approximation that \mathbf{f} and \mathbf{h} do not change significantly from their initial values at t_0 . Thus we can set

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t_0)$$

or, simply

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{u}). \quad (4.84)$$

The assumption of small signals can be reflected by taking \mathbf{x} and \mathbf{u} to be always close to their reference values \mathbf{x}_0 , \mathbf{u}_0 , and these values, furthermore, to be an equilibrium point of Eq. (4.82), where

$$\mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) = 0. \quad (4.85)$$

Now, if \mathbf{x} and \mathbf{u} are "close" to \mathbf{x}_0 and \mathbf{u}_0 , they can be written as $\mathbf{x} = \mathbf{x}_0 + \delta\mathbf{x}$; $\mathbf{u} = \mathbf{u}_0 + \delta\mathbf{u}$, and these can be substituted into Eq. (4.84). The fact that $\delta\mathbf{x}$ and $\delta\mathbf{u}$ are small is now used to motivate an expansion of Eq. (4.84) about \mathbf{x}_0 and \mathbf{u}_0 and to suggest that the only terms in the first power of the small quantities $\delta\mathbf{x}$ and $\delta\mathbf{u}$ need to be retained. We thus have a vector equation and need the expansion of a vector-valued function of a vector variable,

$$\frac{d}{dt}(\mathbf{x}_0 + \delta\mathbf{x}) = \mathbf{f}(\mathbf{x}_0 + \delta\mathbf{x}, \mathbf{u}_0 + \delta\mathbf{u}). \quad (4.86)$$

If we go back to Eq. (4.82) and do the expansion of the components f_i one at a time, it is tedious but simple to verify that Eq. (4.86) can be written as¹⁷

$$\delta\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) + \mathbf{f}_{,\mathbf{x}}(\mathbf{x}_0, \mathbf{u}_0)\delta\mathbf{x} + \mathbf{f}_{,\mathbf{u}}(\mathbf{x}_0, \mathbf{u}_0)\delta\mathbf{u} + \dots \quad (4.87)$$

where we define the partial derivative of a scalar f_i with respect to the vector \mathbf{x} by a subscript notation:

$$\mathbf{f}_{,\mathbf{x}} \stackrel{\text{def}}{=} \left(\frac{\partial f_1}{\partial x_1} \cdots \frac{\partial f_1}{\partial x_n} \right). \quad (4.88)$$

The row vector in Eq. (4.88) is called the **gradient** of the scalar f_1 with respect to the vector \mathbf{x} . If \mathbf{f} is a vector, we define its partial derivatives with respect to the vector \mathbf{x} as the matrix (called the **Jacobian**) composed of rows of gradients. In the subscript notation, if we mean to take the partial of *all* components, we omit the specific subscript such as 1 or 2 but hold its place by the use of a comma

$$\mathbf{f}_{,\mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial \mathbf{x}} \\ \frac{\partial f_2}{\partial \mathbf{x}} \\ \vdots \\ \frac{\partial f_n}{\partial \mathbf{x}} \end{bmatrix}. \quad (4.89)$$

Now, to return to Eq. (4.87), we note that by Eq. (4.85) we chose \mathbf{x}_0 , \mathbf{u}_0 to be an equilibrium point, so the first term on the right of Eq. (4.87) is zero, and because the terms beyond those shown depend on higher powers of the small signals $\delta\mathbf{x}$ and $\delta\mathbf{u}$, we are led to the approximation

$$\begin{aligned} \delta\dot{\mathbf{x}} &\approx \mathbf{f}_{,\mathbf{x}}(\mathbf{x}_0, \mathbf{u}_0)\delta\mathbf{x} + \mathbf{f}_{,\mathbf{u}}(\mathbf{x}_0, \mathbf{u}_0)\delta\mathbf{u}, \\ \delta\mathbf{y} &= \mathbf{h}_{,\mathbf{x}}\delta\mathbf{x} + \mathbf{h}_{,\mathbf{u}}\delta\mathbf{u}. \end{aligned} \quad (4.90)$$

¹⁷ Note that $d\mathbf{x}_0/dt = 0$ because our "reference trajectory" \mathbf{x}_0 is a constant here.

But now the notation is overly clumsy, so we drop the $\delta\mathbf{x}$, $\delta\mathbf{u}$ and $\delta\mathbf{y}$ notation and simply call them \mathbf{x} , \mathbf{u} and \mathbf{y} and define the constant matrices

$$\begin{aligned} \mathbf{F} &= \mathbf{f}_{,\mathbf{x}}(\mathbf{x}_0, \mathbf{u}_0), & \mathbf{G} &= \mathbf{f}_{,\mathbf{u}}(\mathbf{x}_0, \mathbf{u}_0), \\ \mathbf{H} &= \mathbf{h}_{,\mathbf{x}}(\mathbf{x}_0, \mathbf{u}_0), & \mathbf{J} &= \mathbf{h}_{,\mathbf{u}}(\mathbf{x}_0, \mathbf{u}_0). \end{aligned}$$

This results in the form we used earlier in Section 2.1.1

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{Gu}, \quad \mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{Ju}. \quad (4.91)$$

We go even further and restrict ourselves to the case of single input and single output and discrete time. We then write the model as

$$\begin{aligned} \mathbf{x}(k+1) &= \Phi\mathbf{x}(k) + \Gamma u(k), \\ \mathbf{y}(k) &= \mathbf{Hx}(k) + Ju(k), \end{aligned} \quad (4.92)$$

from which the transfer function is

$$\mathbf{G}(z) = \mathbf{Y}(z)/U(z) = \mathbf{H}(z\mathbf{I} - \Phi)^{-1}\Gamma + J. \quad (4.93)$$

Thus we see that nonlinear models can be approximated as linear state-space models or as transfer function models. The accuracy of the approximation varies with the problem, but is generally useful in designing the control system. The final design of the control system should always be checked via numerical simulation of the nonlinear equations.

4.4 Signal Analysis and Dynamic Response

In Section 4.2 we demonstrated that if two variables are related by a linear constant difference equation, then the ratio of the z -transform of the output signal to that of the input is a function of the system equation alone, and the ratio is called the transfer function. A method for study of linear constant discrete systems is thereby indicated, consisting of the following steps:

1. Compute the transfer function of the system $H(z)$.
2. Compute the transform of the input signal, $E(z)$.
3. Form the product, $E(z)H(z)$, which is the transform of the output signal, U .
4. Invert the transform to obtain $u(kT)$.

If the system description is available in difference-equation form, and if the input signal is elementary, then the first three steps of this process require very little effort or computation. The final step, however, is tedious if done by hand, and, because we will later be preoccupied with design of transfer functions to give desirable responses, we attach great benefit to gaining intuition for the kind of response to be expected from a transform without actually inverting it or numerically evaluating the response. Our approach to this problem is to present a

repertoire of elementary signals with known features and to learn their representation in the transform or z -domain. Thus, when given an unknown transform, we will be able, by reference to these known solutions, to infer the major features of the time-domain signal and thus to determine whether the unknown is of sufficient interest to warrant the effort of detailed time-response computation. To begin this process of attaching a connection between the time domain and the z -transform domain, we compute the transforms of a few elementary signals.

4.4.1 The Unit Pulse

We have already seen that the unit pulse is defined by¹⁸

$$\begin{aligned} e_1(k) &= 1 & (k = 0) \\ &= 0 & (k \neq 0) \\ &\equiv \delta_k; \end{aligned}$$

therefore we have

$$E_1(z) = \sum_{k=-\infty}^{\infty} \delta_k z^{-k} = z^0 = 1. \quad (4.94)$$

This result is much like the continuous case, wherein the Laplace transform of the unit impulse is the constant 1.

The quantity $E_1(z)$ gives us an instantaneous method to relate signals to systems: To characterize the system $H(z)$, consider the signal $u(k)$, which is the unit pulse response; then $U(z) = H(z)$.

4.4.2 The Unit Step

Consider the unit step function defined by

$$\begin{aligned} e_s(k) &= 1 & (k \geq 0) \\ &= 0 & (k < 0) \\ &\equiv 1(k). \end{aligned}$$

In this case, the z -transform is

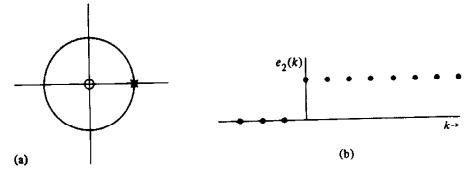
$$\begin{aligned} E_s(z) &= \sum_{k=-\infty}^{\infty} e_s(k) z^{-k} = \sum_{k=0}^{\infty} z^{-k} \\ &= \frac{1}{1 - z^{-1}} & (|z^{-1}| < 1) \\ &= \frac{z}{z - 1} & (|z| > 1). \end{aligned} \quad (4.95)$$

¹⁸ We have shifted notation here to use $e(k)$ rather than e_i for the k th sample. We use subscripts to identify different signals.

Here the transform is characterized by a zero at $z = 0$ and a pole at $z = 1$. The significance of the convergence being restricted to $|z| > 1$ will be explored later when we consider the inverse transform operation. The Laplace transform of the unit step is $1/s$; we may thus keep in mind that a pole at $s = 0$ for a continuous signal corresponds in some way to a pole at $z = 1$ for discrete signals. We will explore this further later. In any event, we record that a pole at $z = 1$ with convergence outside the unit circle, $|z| = 1$, will correspond to a constant for positive time and zero for negative time.

To emphasize the connection between the time domain and the z -plane, we sketch in Fig. 4.21 the z -plane with the unit circle shown and the pole of $E_s(z)$ marked \times and the zero marked \circ . Beside the z -plane, we sketch the time plot of $e_s(k)$.

Figure 4.21
(a) Pole and zero of $E_s(z)$ in the z -plane. The unit circle is shown for reference. (b) Plot of $e_s(k)$



4.4.3 Exponential

The one-sided exponential in time is

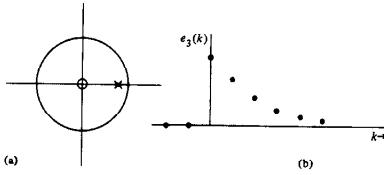
$$e_r(k) = r^k \quad (k \geq 0) \\ = 0 \quad (k < 0) \quad (4.96)$$

which is the same as $r^k 1(k)$, using the symbol $1(k)$ for the unit step function. Now we get

$$\begin{aligned} E_r(z) &= \sum_{k=0}^{\infty} r^k z^{-k} \\ &= \sum_{k=0}^{\infty} (rz^{-1})^k \\ &= \frac{1}{1 - rz^{-1}} & (|rz^{-1}| < 1) \\ &= \frac{z}{z - r} & (|z| > |r|). \end{aligned} \quad (4.97)$$

The pole of $E_3(z)$ is at $z = r$. From Eq. (4.96) we know that $e_3(k)$ grows without bound if $|r| > 1$. From Eq. (4.97) we conclude that a z -transform that converges for large z and has a real pole *outside* the circle $|z| = 1$ corresponds to a growing signal. If such a signal were the unit-pulse response of our system, such as our digital control program, we would say the program was *unstable* as we saw in Eq. (4.39). We plot in Fig. 4.22 the z -plane and the corresponding time history of $E_3(z)$ as $e_3(k)$ for the stable value, $r = 0.6$.

Figure 4.22
(a) Pole and zero of $E_3(z)$ in the z -plane. (b) Plot of $e_3(k)$



4.4.4 General Sinusoid

Our next example considers the modulated sinusoid $e_4(k) = [r^k \cos(k\theta)]1(k)$, where we assume $r > 0$. Actually, we can decompose $e_4(k)$ into the sum of two complex exponentials as

$$e_4(k) = r^k \left(\frac{e^{j\theta k} + e^{-j\theta k}}{2} \right) 1(k),$$

and because the z -transform is linear,¹⁹ we need only compute the transform of each single complex exponential and add the results later. We thus take first

$$e_3(k) = r^k e^{j\theta k} 1(k) \quad (4.98)$$

and compute

$$\begin{aligned} E_4(z) &= \sum_{k=0}^{\infty} r^k e^{j\theta k} z^{-k} \\ &= \sum_{k=0}^{\infty} (re^{j\theta} z^{-1})^k \end{aligned}$$

¹⁹ We have not shown this formally. The demonstration, using the definition of linearity given above, is simple and is given in Section 4.6.

$$\begin{aligned} &= \frac{1}{1 - re^{j\theta} z^{-1}} \\ &= \frac{z}{z - re^{j\theta}} \quad (|z| > r). \end{aligned} \quad (4.99)$$

The signal $e_4(k)$ grows without bound as k gets large if and only if $r > 1$, and a system with this pulse response is BIBO stable if and only if $|r| < 1$. The boundary of stability is the unit circle. To complete the argument given before for $e_4(k) = r^k \cos(k\theta) 1(k)$, we see immediately that the other half is found by replacing θ by $-\theta$ in Eq. (4.99).

$$\mathcal{Z}\{r^k e^{-jk\theta} 1(k)\} = \frac{z}{z - re^{-j\theta}} \quad (|z| > r), \quad (4.100)$$

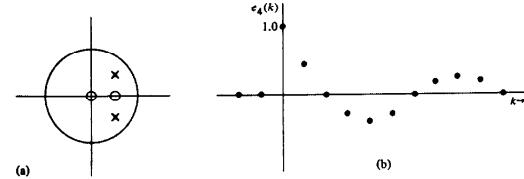
and thus that

$$\begin{aligned} E_4(z) &= \frac{1}{2} \left\{ \frac{z}{z - re^{j\theta}} + \frac{z}{z - re^{-j\theta}} \right\} \\ &= \frac{z(z - r \cos \theta)}{z^2 - 2r(\cos \theta)z + r^2}. \quad (|z| > r). \end{aligned} \quad (4.101)$$

The z -plane pole-zero pattern of $E_4(z)$ and the time plot of $e_4(k)$ are shown in Fig. 4.23 for $r = 0.7$ and $\theta = 45^\circ$.

We note in passing that if $\theta = 0$, then e_4 reduces to e_3 and, with $r = 1$, to e_1 , so that three of our signals are special cases of e_4 . By exploiting the features of $E_4(z)$, we can draw a number of conclusions about the relation between pole locations in the z -plane and the time-domain signals to which the poles correspond. We collect these for later reference.

Figure 4.23
(a) Poles and zeros of $E_4(z)$ for $\theta = 45^\circ$, $r = 0.7$ in the z -plane. (b) Plot of $e_4(k)$



- The settling time of a transient, defined as the time required for the signal to decay to one percent of its maximum value, is set mainly by the value of the radius, r , of the poles.
- (a) $r > 1$ corresponds to a growing signal that will not decay at all.
- (b) $r = 1$ corresponds to a signal with constant amplitude (which is *not* BIBO stable as a pulse response).
- (c) For $r < 1$, the closer r is to 0 the shorter the settling time. The corresponding system is BIBO stable. We can compute the settling time in samples, N , in terms of the pole radius, r .

Pole Radius r	Response Duration N
0.9	43
0.8	21
0.6	9
0.4	5

- (d) A pole at $r = 0$ corresponds to a transient of finite duration.

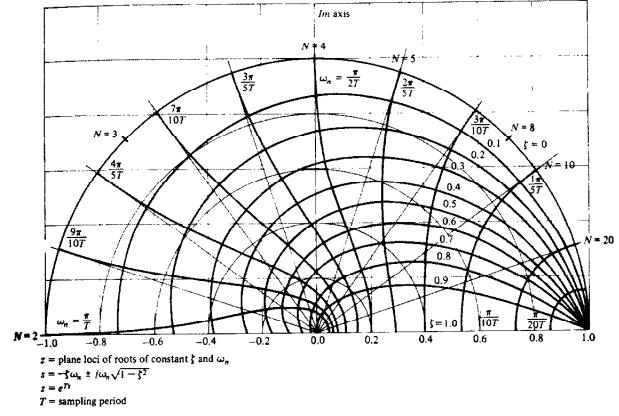
- The number of samples per oscillation of a sinusoidal signal is determined by θ . If we require $\cos(\theta k) = \cos(\theta(k + N))$, we find that a period of 2π rad contains N samples, where

$$N = \frac{2\pi}{\theta} \Big|_{\text{rad}} = \frac{360}{\theta \text{ deg}} \text{ samples/cycle.}$$

For $\theta = 45^\circ$, we have $N = 8$, and the plot of $e_4(k)$ given in Fig. 4.23(b) shows the eight samples in the first cycle very clearly. A sketch of the unit circle with several points corresponding to various numbers of samples per cycle marked is drawn in Fig. 4.24 along with other contours that will be explained in the next section. The sampling frequency in Hertz is $1/T$, and the signal frequency is $f = 1/NT$ so that $N = f_s/T$ and $1/N$ is a *normalized* signal frequency. Since $\theta = (2\pi)/N$, θ is the normalized signal frequency in radians/sample. θ/T is the frequency in radians/second.

A compilation of signal responses versus their pole location in the z -plane is shown in Fig. 4.25. It demonstrates visually the features just summarized for the general sinusoid, which encompasses all possible signals.

Figure 4.24
Sketch of the unit circle with angle θ marked in numbers of samples per cycle



4.4.5 Correspondence with Continuous Signals

From the calculation of these few z -transforms, we have established that the duration of a time signal is related to the radius of the pole locations and the number of samples per cycle is related to the angle, θ . Another set of very useful relationships can be established by considering the signals to be samples from a continuous signal, $e(t)$, with Laplace transform $E(s)$. With this device we can exploit our knowledge of s -plane features by transferring them to equivalent z -plane properties. For the specific numbers represented in the illustration of e_4 , we take the continuous signal

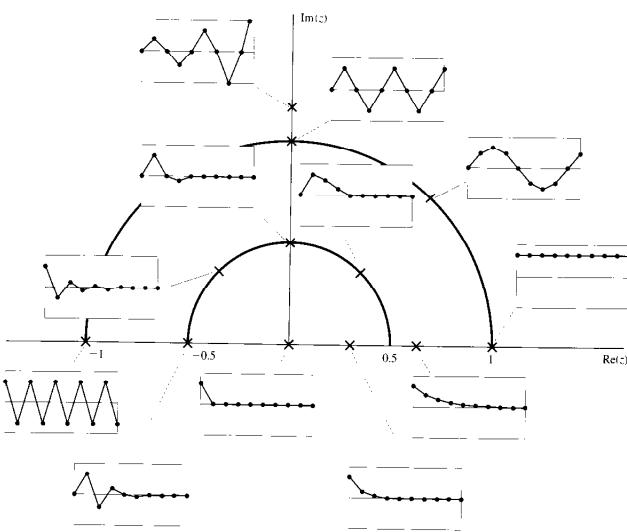
$$v(t) = e^{-at} \cos bt \, u(t) \quad (4.102)$$

with

$$aT = 0.3567.$$

$$bT = \pi/4.$$

Figure 4.25
Time sequences associated with pole locations in the z -plane



And, taking samples one second apart ($T = 1$), we have

$$\begin{aligned} y(kT) &= (e^{-0.3567})^k \cos \frac{\pi k}{4} l(k) \\ &= (0.7)^k \cos \frac{\pi k}{4} l(k) \\ &= e_4(k). \end{aligned}$$

The poles of the Laplace transform of $y(t)$ (in the s -plane) are at

$$s_{1,2} = -a + jb, -a - jb.$$

From Eq. (4.101), the z -transform of $E_4(z)$ has poles at

$$z_{1,2} = re^{j\theta}, re^{-j\theta},$$

but because $y(kT)$ equals $e_4(k)$, it follows that

$$\begin{aligned} r &= e^{-aT}, \quad \theta = bT \\ z_{1,2} &= e^{j\pi T}, e^{-j\pi T}. \end{aligned}$$

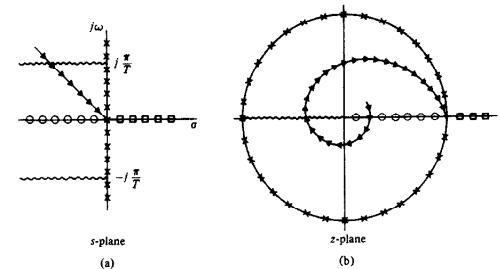
If $E(z)$ is a ratio of polynomials in z , which will be the case if $e(k)$ is generated by a linear difference equation with constant coefficients, then by partial fraction expansion, $E(z)$ can be expressed as a sum of elementary terms like E_4 and E_V .²⁰ In all such cases, the discrete signal can be generated by samples from continuous signals where the relation between the s -plane poles and the corresponding z -plane poles is given by

$$z = e^{j\pi T}. \quad (4.103)$$

If we know what it means to have a pole in a certain place in the s -plane, then Eq. (4.103) shows us where to look in the z -plane to find a representation of discrete samples having the same time features. It is useful to sketch several major features from the s -plane to the z -plane according to Eq. (4.103) to help fix these ideas. Such a sketch is shown in Fig. 4.26.

Each feature should be traced in the mind to obtain a good grasp of the relation. These features are given in Table 4.2. We note in passing that the map

Figure 4.26
Corresponding lines in the s -plane and the z -plane according to $z = e^{j\pi T}$



²⁰ Unless a pole of $E(z)$ is repeated. We have yet to compute the discrete version of a signal corresponding to a higher-order pole. The result is readily shown to be a polynomial in k multiplying $r^k e^{jk\omega}$.

Description of corresponding lines in s-plane and z-plane		
s-plane	Symbol	z-plane
$\{ s = j\omega_n$	$\times \times \times$	$\{ z = 1$
Real frequency axis	$\square \square \square$	Unit circle
$s = a \geq 0$	$\circ \circ \circ$	$z = r, 0 \leq r \leq 1$
$s = a \leq 0$		
$\{ s = -\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2}$	$\triangle \triangle \triangle$	$\{ z = re^{j\theta} \text{ where } r = \exp(-\zeta\omega_n T)$
$= -a + jb$		$= e^{j\theta}, \theta = \omega_n T$
Constant damping ratio		$\theta = \omega_n T \sqrt{1-\zeta^2} = bT$
if ζ is fixed and ω_n varies		Logarithmic spiral
$s = \pm j(\pi/T) + \alpha_i$	$\alpha \leq 0$	$z = -r$

$z = e^{j\tau}$ of Eq. (4.103) is many-to-one. There are many values of s for each value of z . In fact, if

$$s_2 = s_1 + j \frac{2\pi}{T} N,$$

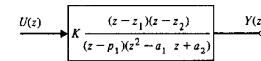
then $e^{j\tau'} = e^{j\tau}$. The (great) significance of this fact will be explored in Chapter 5.

Lines of constant damping in the s -plane are mapped into the z -plane according to Eq. (4.103) for several values of z in Fig. 4.24. We often refer to the damping of a pole in the z -plane in terms of this **equivalent s-plane damping**, or sometimes we simply refer to the damping of a z -plane pole. Likewise, lines of constant natural frequency, ω_n , in the s -plane (semi-circles centered at the origin) are also mapped into the z -plane according to Eq. (4.103) for several values of ω_n in Fig. 4.24. It's interesting to note that in the immediate vicinity of $z = +1$, the map of ζ and ω_n looks exactly like the s -plane in the vicinity of $s = 0$. Because of the usefulness of this mapping, the Control System Toolbox has the function `zgrid.m` that allows one to superimpose this mapping on various plots to help in the interpretation of the results. You will see its use in the figure files of discrete root loci in Chapter 7.

4.4.6 Step Response

Our eventual purpose, of course, is to design digital controls, and our interest in the relation between z -plane poles and zeros and time-domain response comes from our need to know how a proposed design will respond in a given dynamic situation. The generic dynamic test for controls is the step response, and we will conclude this discussion of discrete system dynamic response with an examination of the relationships between the pole-zero patterns of elementary systems and the corresponding step responses for a discrete transfer function from u to y of a hypothetical plant. Our attention will be restricted to the step responses of the discrete system shown in Fig. 4.27 for a selected set of values of the parameters:

Figure 4.27
Definition of the parameters of the system whose step responses are to be catalogued



Note that if $z_1 = p_1$, the members of the one pole-zero pair cancel out; and if at the same time $z_2 = r \cos(\theta)$, $a_1 = -2r \cos(\theta)$, and $a_2 = r^2$, the system response, $Y(z)$, to the input with transform $U(z) = 1$ (a unit pulse) is

$$Y(z) = \frac{z - r \cos \theta}{z^2 - 2r \cos \theta z + r^2}. \quad (4.104)$$

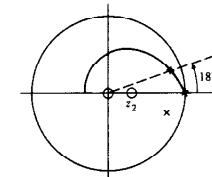
This transform, when compared with the transform $E_1(z)$ given in Eq. (4.101), is seen to be

$$Y(z) = z^{-1} E_1(z),$$

and we conclude that under these circumstances the system pulse response is a delayed version of $E_1(k)$, a typical second-order system pulse response.

For our first study we consider the effect of zero location. We let $z_1 = p_1$ and explore the effect of the (remaining) zero location, z_2 , on the step-response overshoot for three sets of values of a_1 and a_2 . We select a_1 and a_2 so that the poles of the system correspond to a response with an equivalent s -plane damping ratio $\zeta = 0.5$ and consider values of θ of 18, 45, and 72 degrees. In every case, we will take the gain K to be such that the steady-state output value equals the step size. The situation in the z -plane is sketched in Fig. 4.28 for $\theta = 18^\circ$. The curve for $\zeta = 0.5$ is also shown for reference. In addition to the two poles and one zero of $H(z)$, we show the pole at $z = 1$ and the zero at $z = 0$, which come from the transform of the input step, $U(z)$, given by $z/(z - 1)$.

Figure 4.28
Pole-zero pattern of $Y(z)$ for the system of Fig. 4.27, with $z_1 = p_1$, $U(z) = z/(z - 1)$, a_1 and a_2 selected for $\theta = 18^\circ$, and $\zeta = 0.5$



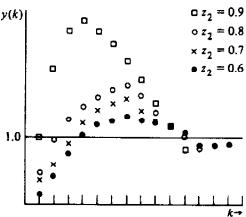
The major effect of the zero z_2 on the step response $y(k)$ is to change the percent overshoot, as can be seen from the four step responses for this case plotted in Fig. 4.29. To summarize all these data, we plot the percent overshoot versus zero location in Fig. 4.30 for $\zeta = 0.5$ and in Fig. 4.31, for $\zeta = 0.707$. The major feature of these plots is that the zero has very little influence when on the negative axis, but its influence is dramatic as it comes near +1. Also included on the plots of Fig. 4.30 are overshoot figures for a zero in the unstable region on the positive real axis. These responses go in the *negative* direction at first, and for the zero very near +1, the negative peak is larger than 1.²¹

Our second class of step responses corresponds to a study of the influence of a third pole on a basically second-order response. For this case we again consider the system of Fig. 4.27, but this time we fix $z_1 = z_3 = -1$ and let p_1 vary from near -1 to near +1. In this case, the major influence of the moving singularity is on the rise time of the step response. We plot this effect for $\theta = 18^\circ, 45^\circ$, and 72° and $\zeta = 0.5$ on Fig. 4.32. In the figure we defined the rise time as the time required for the response to rise to 0.95, which is to 5% of its final value. We see here that the extra pole causes the rise time to get very much longer as the location of p_1 moves toward $z = +1$ and comes to dominate the response.

Our conclusions from these plots are that the addition of a pole or zero to a given system has only a small effect if the added singularities are in the range from 0 to -1. However, a zero moving toward $z = +1$ greatly increases the system overshoot. A pole placed toward $z = +1$ causes the response to slow down and thus primarily affects the rise time, which is progressively increased.

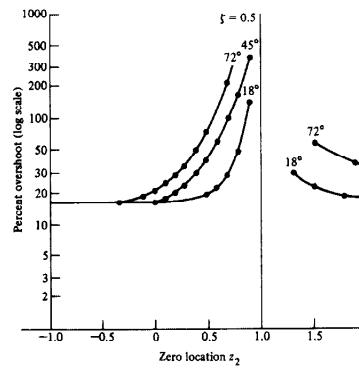
The understanding of how poles and zeros affect the time response is very useful for the control system designer. The knowledge helps guide the iterative

Figure 4.29
Plot of step responses for a discrete plant described by the pole-zero pattern of Fig. 4.28 for various values of z_2 .



21. Such systems are called nonminimum phase by Bode because the phase shift they impart to a sinusoidal input is greater than the phase of a system whose magnitude response is the same but that has a zero in the stable rather than the unstable region.

Figure 4.30
Effects of an extra zero on a discrete second-order system, $\zeta = 0.5$; $\theta = 18^\circ, 45^\circ$, and 72° .



design process and helps the designer understand why a response is the way it is. Ultimately, however, the test of a design is typically the actual time response, either by numerical simulation or an experimental evaluation. Today, transform inversion would never be carried out. In MATLAB, the numerical simulation of the impulse response for a discrete system, sysD is accomplished by

$y = \text{impulse}(\text{sysD})$

and the discrete step response by

$y = \text{step}(\text{sysD})$

Invoked without a left-hand argument ($y =$), both functions result in a plot of the response on the screen.

4.5 Frequency Response

A very important concept in linear systems analysis is the frequency response. If a sinusoid at frequency ω_o is applied to a stable, linear, constant, continuous system, the response is a transient plus a sinusoidal steady state at the same frequency, ω_o , as the input. If the transfer function is written in gain-phase form as $H(j\omega) = A(\omega)e^{j\phi(\omega)}$, then the steady-state response to a unit-amplitude

Figure 4.31
Effects of an extra zero
on a discrete
second-order system,
 $\zeta = 0.707$, $\theta = 18^\circ, 45^\circ$,
and 72° ; percent
overshoot versus zero
location

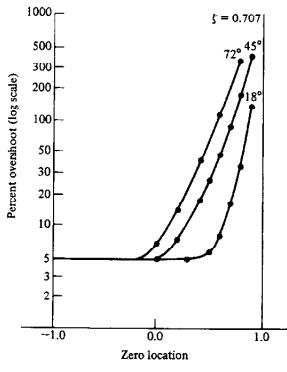
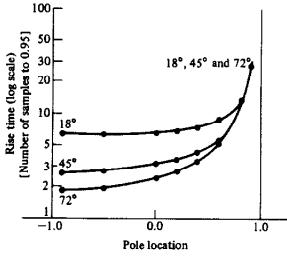


Figure 4.32
Effects of an extra pole
on rise time for a
discrete third-order
system, two zeros at -1 ,
one zero at ∞ : $\zeta =$
 0.5 ; $\theta = 18^\circ, 45^\circ, 72^\circ$



sinusoidal signal has amplitude $A(\omega_o)$ and phase $\psi(\omega_o)$ relative to the input signal.

We can say almost exactly the same respecting the frequency response of a stable, linear, constant, discrete system. If the system has a transfer function $H(z)$,

we define its magnitude and phase for z taking on values around the unit circle by $H(e^{j\omega_o T}) = A(\omega_o T)e^{j\psi(\omega_o T)}$. If a unit-amplitude sinusoid is applied, then in the steady state, the response samples will be on a sinusoid of the same frequency with amplitude $A(\omega_o T)$ and phase $\psi(\omega_o T)$. It is worthwhile going through the calculations to fix ideas on this point.

From Eq. (4.16), the discrete response transform is

$$U(z) = H(z)E(z). \quad (4.105)$$

If $e(k) = \cos(\omega_o T k)1(k)$, then, from Eq. (4.101) with $r = 1$ and $\theta = \omega_o T$, we have

$$E(z) = \frac{1}{2} \left[\frac{z}{z - e^{j\omega_o T}} + \frac{z}{z - e^{-j\omega_o T}} \right]. \quad (4.106)$$

If we substitute Eq. (4.106) into Eq. (4.105), we obtain

$$U(z) = \frac{1}{2} \left[\frac{zH(z)}{z - e^{j\omega_o T}} + \frac{zH(z)}{z - e^{-j\omega_o T}} \right]. \quad (4.107)$$

The steady state of $u(kT)$ corresponds to the terms in the expansion of Eq. (4.107) associated with the two poles on the unit circle. If we expand $U(z)/z$ into partial fractions and multiply back by z , the steady state part can be found as

$$U_{ss}(z) = \frac{1}{2} \frac{H(e^{j\omega_o T})z}{z - e^{j\omega_o T}} + \frac{1}{2} \frac{H(e^{-j\omega_o T})z}{z - e^{-j\omega_o T}}.$$

If $H(e^{j\omega_o T}) = A(\omega_o T)e^{j\psi(\omega_o T)}$, then we have

$$U_{ss}(z) = \frac{A}{2} \frac{e^{j\psi} z}{z - e^{j\omega_o T}} + \frac{A}{2} \frac{e^{-j\psi} z}{z - e^{-j\omega_o T}}, \quad (4.108)$$

and the inverse transform of $U_{ss}(z)$ is

$$U_{ss}(kT) = \frac{A}{2} e^{j\psi} e^{j\omega_o T k} + \frac{A}{2} e^{-j\psi} e^{-j\omega_o T k} = A \cos(\omega_o T k + \psi), \quad (4.109)$$

which, of course, are samples at kT instants on a sinusoid of amplitude A , phase ψ , and frequency ω_o .

We will defer the plotting of particular frequency responses until later chapters (see, for example, Figs. 6.3, 6.8, 7.16, and 7.28). However, it should be noticed here that although a sinusoid of frequency ω_o could be passed through the samples of Eq. (4.109), there are other continuous sinusoids of frequency $\omega_o + \ell 2\pi/T$ for integer ℓ which also pass through these points. This is the phenomenon of aliasing, to which we will return in Chapter 5. Here, we define the discrete frequency response of a transfer function $H(z)$ to sinusoids of frequency ω_o as $H(e^{j\omega_o T})$ so that the amplitude A and phase ψ are

$$A = |H(e^{j\omega_o T})| \quad \text{and} \quad \psi = \angle(H(e^{j\omega_o T})) \quad (4.110)$$

which can be evaluated and plotted by MATLAB's `bode.m` with the scripts

```
sysD = tf(num,den,T)
bode(sysD)
```

where amplitude is plotted in decibels (dB), or

```
[mag,phase,w] = bode(sysD)
subplot(2,1,1), loglog(w,mag)
subplot(2,1,2), semilogx(w,phase)
```

where amplitude is plotted as a ratio as in the figures in this text. If the system is described by the state-space matrices, the scripts above can be invoked with

```
sysD = ss(F,G,H,J,T).
```

4.5.1 *The Discrete Fourier Transform (DFT)

The analysis developed above based on the z -transform is adequate for considering the theoretical frequency response of a linear, constant system or the corresponding difference equation, but it is not the best for the analysis of real-time signals as they occur in the laboratory or in other experimental situations. For the analysis of real data, we need a transform defined over a finite data record, which can be computed quickly and accurately. The required formula is that of the **Discrete Fourier Transform**, the DFT, and its numerical cousin, the **Fast Fourier Transform**, the FFT. Implementation of a version of the FFT algorithm is contained in all signal-processing software and in most computer-aided control-design software.

To understand the DFT, it is useful to consider two properties of a signal and its Fourier transform that are complements of each other: the property of being periodic and the property of being discrete. In ordinary Fourier analysis, we have a signal that is neither periodic nor discrete and its Fourier transform is also neither discrete nor periodic. If, however, the time function $f(t)$ is periodic with period T_0 , then the appropriate form of the transform is the Fourier series, and the transform is defined only for the discrete frequencies $\omega = 2\pi n/T_0$. In other words, if the function in time is periodic, the function in frequency is discrete. The case where the properties are reversed is the z -transform we have just been studying. In this case, the time functions are discrete, being sampled, and the z -transform is periodic in ω ; for if $z = e^{j\omega T}$, corresponding to real frequencies, then replacing $\omega = \omega + 2\pi k/T$ leaves z unchanged. We can summarize these results with the following table:

Fast Fourier Transform

which can be evaluated and plotted by MATLAB's `bode.m` with the scripts

```
sysD = tf(num,den,T)
bode(sysD)
```

where amplitude is plotted in decibels (dB), or

```
[mag,phase,w] = bode(sysD)
subplot(2,1,1), loglog(w,mag)
subplot(2,1,2), semilogx(w,phase)
```

where amplitude is plotted as a ratio as in the figures in this text. If the system is described by the state-space matrices, the scripts above can be invoked with

```
sysD = ss(F,G,H,J,T).
```

4.5.1 *The Discrete Fourier Transform (DFT)

The analysis developed above based on the z -transform is adequate for considering the theoretical frequency response of a linear, constant system or the corresponding difference equation, but it is not the best for the analysis of real-time signals as they occur in the laboratory or in other experimental situations. For the analysis of real data, we need a transform defined over a finite data record, which can be computed quickly and accurately. The required formula is that of the **Discrete Fourier Transform**, the DFT, and its numerical cousin, the **Fast Fourier Transform**, the FFT. Implementation of a version of the FFT algorithm is contained in all signal-processing software and in most computer-aided control-design software.

To understand the DFT, it is useful to consider two properties of a signal and its Fourier transform that are complements of each other: the property of being periodic and the property of being discrete. In ordinary Fourier analysis, we have a signal that is neither periodic nor discrete and its Fourier transform is also neither discrete nor periodic. If, however, the time function $f(t)$ is periodic with period T_0 , then the appropriate form of the transform is the Fourier series, and the transform is defined only for the discrete frequencies $\omega = 2\pi n/T_0$. In other words, if the function in time is periodic, the function in frequency is discrete. The case where the properties are reversed is the z -transform we have just been studying. In this case, the time functions are discrete, being sampled, and the z -transform is periodic in ω ; for if $z = e^{j\omega T}$, corresponding to real frequencies, then replacing $\omega = \omega + 2\pi k/T$ leaves z unchanged. We can summarize these results with the following table:

Fast Fourier Transform

	Time	Frequency
z -transform	periodic discrete	discrete periodic

Suppose we now have a time function that is both periodic and discrete. Based on what we have seen, we would expect the transform of this function also to be both periodic and discrete. And this is the case, which leads us to the finite discrete Fourier transform and its finite inverse. Let the time function in question be $f(kT) = f(kT + NT)$. Because the function is periodic, the transform can be defined as the finite sum

$$F\left(\frac{2\pi n}{NT}\right) = \sum_{k=0}^{N-1} f(kT) e^{-j2\pi(nkT)/(NT)}.$$

This is the same as the z -transform over one period evaluated at the discrete frequencies of a Fourier series $\omega = 2\pi n/NT$. It is standard practice to suppress all the arguments except the indices of time and frequency and write

$$F_n = \sum_{k=0}^{N-1} f_k e^{-j2\pi(nk)/N}. \quad (4.111)$$

To complete the DFT, we need the inverse transform, which, by analogy with the standard Fourier transform, we guess to be the sum

$$\sum_{n=0}^{N-1} F_n e^{j2\pi(nk)/N}.$$

If we substitute Eq. (4.111) with summing index ℓ into this, we find

$$\sum_{n=0}^{N-1} \left\{ \sum_{k=0}^{N-1} f_k e^{-j2\pi(nk)/N} \right\} e^{j2\pi(nk)/N}.$$

Interchanging the order of the summations gives

$$\sum_{\ell=0}^{N-1} f_\ell \left\{ \sum_{n=0}^{N-1} e^{j2\pi[n(k-\ell)]/N} \right\} e^{j2\pi(nk)/N}.$$

The sum in the braces is a finite geometric series, which we can evaluate as follows

$$\begin{aligned} \sum_{n=0}^{N-1} e^{j2\pi[n(k-\ell)]/N} &= \frac{1 - e^{j2\pi(k-\ell)/N}}{1 - e^{j2\pi(k-\ell)/N}} \\ &= \begin{cases} N & k - \ell = 0 \\ 0 & k - \ell = 1, 2, \dots, N-1. \end{cases} \end{aligned}$$

The sum is periodic with period N . With this evaluation, we see that the sum we have been considering is Nf_k , and thus we have the inverse sum

$$f_k = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{j2\pi nk/N}. \quad (4.112)$$

Equations (4.111) and (4.112) comprise the DFT

$$\begin{aligned} F_n &= \sum_{k=0}^{N-1} f_k e^{-j2\pi nk/N}, \\ f_k &= \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{j2\pi nk/N}. \end{aligned}$$

Because there are N terms in the sum in Eq. (4.111), it would appear that to compute the DFT for one frequency it will take on the order of N multiply and add operations; and to compute the DFT for all N frequencies, it would take on the order of N^2 multiply and add operations. However, several authors, especially Cooley and Tukey (1965), have showed how to take advantage of the circular nature of the exponential so that all N values of F_n can be computed with the order of $N \log(N)$ operations if N is a power of 2. For $N = 1024$, this is a saving of a factor of 100, a very large value. Their algorithm and related schemes are called the Fast Fourier Transform or FFT.

To use the DFT/FFT in evaluating frequency response, consider a system described by Eq. (4.105) and where the input is a sinusoid at frequency $\omega_c = 2\pi\ell/N$ so that $e(kT) = A \sin(2\pi\ell k/N)$. We apply this input to the system and wait until all transients have died away. At this time, the output is given by $u(kT) = B \sin(2\pi\ell k/N + \psi)$. The DFT of $e(k)$ is

$$\begin{aligned} E_n &= \sum_{k=0}^{N-1} A \sin\left(\frac{2\pi\ell k}{N}\right) e^{-j2\pi nk/N} \\ &= \sum_{k=0}^{N-1} \frac{A}{2j} [e^{j(2\pi\ell k)/N} - e^{-j(2\pi\ell k)/N}] e^{-j(2\pi nk)/N} \\ &= \begin{cases} 0, & \ell \neq n \\ \frac{NA}{2j}, & \ell = n. \end{cases} \end{aligned}$$

The DFT of the output is

$$\begin{aligned} U_n &= \sum_{k=0}^{N-1} B \sin\left(\frac{2\pi\ell k}{N} + \psi\right) e^{-j2\pi nk/N} \\ &= \sum_{k=0}^{N-1} \frac{B}{2j} [e^{j\psi} e^{j(2\pi\ell k)/N} - e^{-j\psi} e^{-j(2\pi\ell k)/N}] e^{-j(2\pi nk)/N} \end{aligned}$$

Dividing these results, we see that with sinusoidal input and output, the frequency response at the frequency $\omega = (2\pi\ell)/NT$ is given by

$$H(e^{j2\pi\ell/N}) = \frac{U_\ell}{E_\ell},$$

where $U_\ell = FFT(u_k)$ and $E_\ell = FFT(e_k)$, each evaluated at $n = \ell$. We will discuss in Chapter 12 the general problem of estimation of the total frequency response from experimental data using the DFT/FFT as well as other tools.

4.6 Properties of the z -Transform

We have used the z -transform to show that linear, constant, discrete systems can be described by a transfer function that is the z -transform of the system's unit-pulse response, and we have studied the relationship between the pole-zero patterns of transfer functions in the z -plane and the corresponding time responses. We began a table of z -transforms, and a more extensive table is given in Appendix B. In Section 4.6.1 we turn to consideration of some of the properties of the z -transform that are essential to the effective and correct use of this important tool. In Section 4.6.2 convergence issues concerning the z -transform are discussed and in Section 4.6.3 an alternate derivation of the transfer function is given.

4.6.1 Essential Properties

In order to make maximum use of a table of z -transforms, one must be able to use a few simple properties of the z -transform which follow directly from the definition. Some of these, such as linearity, we have already used without making a formal statement of it, and others, such as the transform of the convolution, we have previously derived. For reference, we will demonstrate a few properties here and collect them into Appendix B for future reference. In all the properties listed below, we assume that $F_i(z) = \mathcal{Z}\{f_i(kT)\}$.

- Linearity:** A function $f(x)$ is linear if $f(\alpha x_1 + \beta x_2) = \alpha f(x_1) + \beta f(x_2)$. Applying this result to the definition of the z transform, we find immediately that

$$\begin{aligned} \mathcal{Z}\{\alpha f_1(kT) + \beta f_2(kT)\} &= \sum_{k=-\infty}^{\infty} [\alpha f_1(k) + \beta f_2(k)] z^{-k} \\ &= \alpha \mathcal{Z}\{f_1(k)\} + \beta \mathcal{Z}\{f_2(k)\} \\ &= \alpha F_1(z) + \beta F_2(z). \end{aligned}$$

Thus the z -transform is a linear function. It is the linearity of the transform that makes the partial-fraction technique work.

2. Convolution of Time Sequences:

$$\mathcal{Z} \left\{ \sum_{l=-\infty}^{\infty} f_1(l) f_2(k-l) \right\} = F_1(z) F_2(z).$$

We have already developed this result in connection with Eq. (4.32). It is this result with linearity that makes the transform so useful in linear-constant-system analysis because the analysis of a combination of such dynamic systems can be done by linear algebra on the transfer functions.

3. Time Shift:

$$\mathcal{Z}\{f(k+n)\} = z^n F(z). \quad (4.113)$$

We demonstrate this result also by direct calculation:

$$\mathcal{Z}\{f(k+n)\} = \sum_{k=-\infty}^{\infty} f(k+n) z^{-k}.$$

If we let $k+n=j$, then

$$\begin{aligned} \mathcal{Z}\{f(k+n)\} &= \sum_{j=-\infty}^{\infty} f(j) z^{-(j-n)} \\ &= z^n F(z). \quad \text{QED} \end{aligned}$$

This property is the essential tool in solving linear constant-coefficient difference equations by transforms. We should note here that the transform of the time shift is not the same for the one-sided transform because a shift can introduce terms with negative argument which are not included in the one-sided transform and must be treated separately. This effect causes initial conditions for the difference equation to be introduced when solution is done with the one-sided transform. See Problem 4.13.

4. Scaling in the z -Plane:

$$\mathcal{Z}\{r^{-k} f(k)\} = F(rz). \quad (4.114)$$

By direct substitution, we obtain

$$\begin{aligned} \mathcal{Z}\{r^{-k} f(k)\} &= \sum_{k=-\infty}^{\infty} r^{-k} f(k) z^{-k} \\ &= \sum_{k=-\infty}^{\infty} f(k) (rz)^{-k} \\ &= F(rz). \quad \text{QED} \end{aligned}$$

As an illustration of this property, we consider the z -transform of the unit step, $1(k)$, which we have computed before

$$\mathcal{Z}\{1(k)\} = \sum_{k=0}^{\infty} z^{-k} = \frac{z}{z-1}.$$

By property 4 we have immediately that

$$\mathcal{Z}\{r^{-k} 1(k)\} = \frac{rz}{rz-1} = \frac{z}{z-(1/r)}.$$

As a more general example, if we have a polynomial $a(z) = z^2 + a_1 z + a_2$ with roots $r e^{\pm j\theta}$, then the scaled polynomial $\omega^2 z^2 + a_1 \omega z + a_2$ has roots $(r/\omega) e^{\pm j\theta}$. This is an example of radial projection whereby the roots of a polynomial can be projected radially simply by changing the coefficients of the polynomial. The technique is sometimes used in pole-placement designs as described in Chapter 8, and sometimes used in adaptive control as described in Chapter 13.

5. Final-Value Theorem: If $F(z)$ converges for $|z| > 1$ and all poles of $(z-1)F(z)$ are inside the unit circle, then

$$\lim_{z \rightarrow \infty} f(k) = \lim_{z \rightarrow 1} (z-1)F(z). \quad (4.115)$$

The conditions on $F(z)$ assure that the only possible pole of $F(z)$ not strictly inside the unit circle is a simple pole at $z=1$, which is removed in $(z-1)F(z)$. Furthermore, the fact that $F(z)$ converges as the magnitude of z gets arbitrarily large ensures that $f(k)$ is zero for negative k . Therefore, all components of $f(k)$ tend to zero as k gets large, with the possible exception of the constant term due to the pole at $z=1$. The size of this constant is given by the coefficient of $1/(z-1)$ in the partial-fraction expansion of $F(z)$, namely

$$C = \lim_{z \rightarrow 1} (z-1)F(z).$$

However, because all other terms in $f(k)$ tend to zero, the constant C is the final value of $f(k)$, and Eq. (4.115) results. QED

As an illustration of this property, we consider the signal whose transform is given by

$$U(z) = \frac{z - Tz + 1}{z - 0.5z - 1}, \quad |z| > 1.$$

Because $U(z)$ satisfies the conditions of Eq. (4.115), we have

$$\begin{aligned} \lim_{k \rightarrow \infty} u(k) &= \lim_{z \rightarrow 1} (z-1) \frac{z - Tz + 1}{z - 0.5z - 1} \frac{Tz + 1}{z - 1} \\ &= \lim_{z \rightarrow 1} \frac{z - T}{z - 0.5z} (z+1) \end{aligned}$$

$$= \frac{1}{1 - 0.5 \frac{T}{2}} (1 + 1) \\ = 2T.$$

This result can be checked against the closed form for $u(k)$ given by Eq. (4.121) below.

- 6. Inversion:** As with the Laplace transform, the z -transform is actually one of a pair of transforms that connect functions of time to functions of the complex variable z . The z -transform computes a function of z from a sequence in k . (We identify the sequence number k with time in our analysis of dynamic systems, but there is nothing in the transform *per se* that requires this.) The inverse z -transform is a means to compute a sequence in k from a given function of z . We first examine two elementary schemes for inversion of a given $F(z)$ which can be used if we know beforehand that $F(z)$ is rational in z and converges as z approaches infinity. For a sequence $f(k)$, the z -transform has been defined as

$$F(z) = \sum_{k=-\infty}^{\infty} f(k)z^{-k}, \quad r_0 < |z| < R_0. \quad (4.116)$$

If any value of $f(k)$ for negative k is nonzero, then there will be a term in Eq. (4.116) with a positive power of z . This term will be unbounded if the magnitude of z is unbounded; and thus if $F(z)$ converges as $|z|$ approaches infinity, we know that $f(k)$ is zero for $k < 0$. In this case, Eq. (4.116) is one-sided, and we can write

$$F(z) = \sum_{k=0}^{\infty} f(k)z^{-k}. \quad r_0 < |z|. \quad (4.117)$$

The right-hand side of Eq. (4.117) is a series expansion of $F(z)$ about infinity or about $z^{-1} = 0$. Such an expansion is especially easy if $F(z)$ is the ratio of two polynomials in z^{-1} . We need only divide the numerator by the denominator in the correct way, and when the division is done, the coefficient of z^{-k} is automatically the sequence value $f(k)$. An example we have worked out before will illustrate the process.

◆ Example 4.12 z -Transform Inversion by Long Division

The system for trapezoid-rule integration has the transfer function given by Eq. (4.14)

$$H(z) = \frac{Tz + 1}{2z - 1}, \quad |z| > 1.$$

Determine the output for an input which is the geometric series represented by $e_3(k)$ with $r = 0.5$. That is

$$E_3(z) = \frac{z}{z - 0.5}, \quad |z| > 0.5.$$

Solution. The z -transform of the output is

$$U(z) = E_3(z)H(z) \\ = \frac{z}{z - 0.5} \frac{Tz + 1}{2z - 1}, \quad |z| > 1. \quad (4.118)$$

Equation (4.118) represents the transform of the system output, $u(k)$. Keeping out the factor of $T/2$, we write $U(z)$ as a ratio of polynomials in z^{-1}

$$U(z) \rightarrow \frac{T}{2} \frac{1 + z^{-1}}{1 - 1.5z^{-1} + 0.5z^{-2}}, \quad (4.119)$$

and divide as follows

$$\begin{array}{r} \frac{T[1 + 2.5z^{-1} + 3.25z^{-2} + 3.625z^{-3} + \dots]}{1 - 1.5z^{-1} + 0.5z^{-2}} \\ \underline{\frac{1 + z^{-1}}{1 - 1.5z^{-1} + 0.5z^{-2}}} \\ \frac{2.5z^{-1} - 0.5z^{-2}}{2.5z^{-1} - 3.75z^{-2} + 1.25z^{-3}} \\ \underline{\frac{2.5z^{-1} - 3.75z^{-2} + 1.25z^{-3}}{3.25z^{-2} - 1.25z^{-3}}} \\ \frac{3.25z^{-2} - 4.875z^{-3} + 1.625z^{-4}}{3.625z^{-3} - 1.625z^{-4}} \\ \underline{\frac{3.625z^{-3} - 1.625z^{-4}}{3.625z^{-3} - \dots}} \end{array}$$

By direct comparison with $U(z) = \sum_{k=0}^{\infty} u(k)z^{-k}$, we conclude that

$$\begin{aligned} u_0 &= T/2, \\ u_1 &= (T/2)2.5, \\ u_2 &= (T/2)3.25, \\ &\vdots \end{aligned} \quad (4.120)$$

Clearly, the use of a computer will greatly aid the speed of this process in all but the simplest of cases. Some may prefer to use synthetic division and omit copying over all the extraneous z 's in the division. The process is identical to converting $F(z)$ to the equivalent difference equation and solving for the unit-pulse response.

The second special method for the inversion of z -transforms is to decompose $F(z)$ by partial-fraction expansion and look up the components of the sequence $f(k)$ in a previously prepared table.

◆ Example 4.13 z -Transform Inversion by Partial Fraction Expansion

Repeat Example 4.12 using the partial fraction expansion method.

Solution. Consider again Eq. (4.118) and expand $U(z)$ as a function of z^{-1} as follows

$$U(z) = \frac{T}{2} \frac{1+z^{-1}}{1-z^{-1}} \frac{1}{1-0.5z^{-1}} = \frac{A}{1-z^{-1}} + \frac{B}{1-0.5z^{-1}}.$$

We multiply both sides by $1-z^{-1}$, let $z^{-1}=1$, and compute

$$A = \frac{T}{2} \frac{2}{1-2} = -T.$$

Similarly, at $z^{-1}=2$, we evaluate

$$B = \frac{T}{2} \frac{1+2}{1-2} = -\frac{3T}{2}.$$

Looking back now at e_2 and e_3 , which constitute our "table" for the moment, we can copy down that

$$\begin{aligned} u_k &= Ae_2(k) + Be_3(k) \\ &= 2Te_2(k) - \frac{3T}{2}e_3(k) \\ &= \left(2T - \frac{3T}{2} \left(\frac{1}{2}\right)^k\right)1(k) \\ &= \frac{T}{2} \left[4 - \frac{3}{2^k}\right]1(k). \end{aligned} \quad (4.121)$$

Evaluation of Eq. (4.121) for $k=0, 1, 2, \dots$ will, naturally, give the same values for $u(k)$ as we found in Eq. (4.120). \blacklozenge

4.6.2 *Convergence of z -Transform

We now examine more closely the role of the region of convergence of the z -transform and present the inverse-transform integral. We begin with another example. The sequence

$$f(k) = \begin{cases} -1 & k < 0 \\ 0 & k \geq 0 \end{cases}$$

has the transform

$$\begin{aligned} F(z) &= \sum_{k=-\infty}^{-1} -z^{-k} = -\left[\sum_{k=0}^{\infty} z^k - 1\right] \\ &= \frac{z}{z-1}, \quad |z| < 1. \end{aligned}$$

This transform is exactly the same as the transform of the unit step $1(k)$, Eq. (4.95), except that this transform converges *inside* the unit circle and the transform of the $1(k)$ converges outside the unit circle. Knowledge of the region of convergence

is obviously essential to the proper inversion of the transform to obtain the time sequence. The inverse z -transform is the closed, complex integral²²

$$f(k) = \frac{1}{2\pi j} \oint_C F(z) z^k \frac{dz}{z}. \quad (4.122)$$

where the contour is a circle in the region of convergence of $F(z)$. To demonstrate the correctness of the integral and to use it to compute inverses, it is useful to apply Cauchy's residue calculus [see Churchill and Brown (1984)]. Cauchy's result is that a closed integral of a function of z which is analytic on and inside a closed contour C except at a finite number of isolated singularities z_i is given by

$$\frac{1}{2\pi j} \oint_C F(z) dz = \sum_i \text{Res}(z_i). \quad (4.123)$$

In Eq. (4.123), $\text{Res}(z_i)$ means the residue of $F(z)$ at the singularity at z_i . We will be considering only rational functions, and these have only poles as singularities. If $F(z)$ has a pole of order n at z_i , then $(z - z_i)^n F(z)$ is regular at z_i and can be expanded in a Taylor series near z_i as

$$(z - z_i)^n F(z) = A_{-n} + A_{-n+1}(z - z_i) + \cdots + A_{-1}(z - z_i)^{n-1} + A_0(z - z_i)^n + \cdots \quad (4.124)$$

The residue of $F(z)$ at z_i is A_{-1} .

First we will use Cauchy's formula to verify Eq. (4.123). If $F(z)$ is the z -transform of $f(k)$, then we write

$$\mathcal{I} = \frac{1}{2\pi j} \oint_C \sum_{l=-\infty}^{\infty} f(l) z^{-l} \frac{dz}{z}.$$

We assume that the series for $F(z)$ converges uniformly on the contour of integration, so the series can be integrated term by term. Thus we have

$$\mathcal{I} = \frac{1}{2\pi j} \sum_{l=-\infty}^{\infty} f(l) \oint_C z^{-l-1} \frac{dz}{z}.$$

The argument of the integral has no pole inside the contour if $k-l \geq 1$, and it has zero residue at the pole at $z=0$ if $k-l < 0$. Only if $k=l$ does the integral have a residue, and that is 1. By Eq. (4.123), the integral is zero if $k \neq l$ and is $2\pi j$ if $k=l$. Thus $\mathcal{I} = f(k)$, which demonstrates Eq. (4.122).

²² If it is known that $f(k)$ is causal, that is, $f(k)=0$ for $k < 0$, then the region of convergence is outside the smallest circle that contains all the poles of $F(z)$ for rational transforms. It is this property that permits inversion by partial-fraction expansion and long division.

To illustrate the use of Eq. (4.123) to compute the inverse of a z -transform, we will use the function $z/(z - 1)$ and consider first the case of convergence for $|z| > 1$ and second the case of convergence for $|z| < 1$. For the first case

$$f_1(k) = \frac{1}{2\pi j} \oint_{|z|=R>1} \frac{z}{z-1} z^k \frac{dz}{z}, \quad (4.125)$$

where the contour is a circle of radius greater than 1. Suppose $k < 0$. In this case, the argument of the integral has two poles inside the contour: one at $z = 1$ with residue

$$\lim_{z \rightarrow 1^-} (z-1) \frac{z^k}{z-1} = 1,$$

and one pole at $z = 0$ with residue found as in (2.109) if $k < 0$, then z^{-k} removes the pole)

$$\begin{aligned} z^{-k} \frac{z^k}{z-1} &= \frac{-1}{z-1} \\ &= -(1 + z^{-1} + z^{-2} + \dots + z^{-k} + \dots). \end{aligned}$$

The residue is thus -1 for all k , and the sum of the residues is zero, and

$$f_1(k) = 0, \quad k < 0. \quad (4.126)$$

For $k \geq 0$, the argument of the integral in Eq. (4.125) has only the pole at $z = 1$ with residue 1. Thus

$$f_1(k) = 1, \quad k \geq 0. \quad (4.127)$$

Equations (4.123) and (4.124) correspond to the unit-step function, as they should. We would write the inverse transform symbolically $\mathcal{Z}^{-1}\{\cdot\}$ as, in this case

$$\mathcal{Z}^{-1}\left\{\frac{z}{z-1}\right\} = 1(k) \quad (4.128)$$

when $z/(z - 1)$ converges for $|z| > 1$.

If, on the other hand, convergence is inside the unit circle, then for $k \geq 0$, there are no poles of the integrand contained in the contour, and

$$f_2(k) = 0, \quad k \geq 0.$$

At $k < 0$, there is a pole at the origin of z , and as before, the residue is equal to -1 there, so

$$f_2(k) = -1, \quad k < 0.$$

In symbols, corresponding to Eq. (4.128), we have

$$\mathcal{Z}^{-1}\left\{\frac{z}{z-1}\right\} = 1(k) - 1$$

when $z/(z - 1)$ converges for $|z| < 1$.

Although, as we have just seen, the inverse integral can be used to compute an expression for a sequence to which a transform corresponds, a more effective use of the integral is in more general manipulations. We consider one such case that will be of some interest later. First, we consider an expression for the transform of a product of two sequences. Suppose we have

$$f_3(k) = f_1(k)f_2(k),$$

and f_1 and f_2 are such that the transform of the product exists. An expression for $F_3(z)$ in terms of $F_1(z)$ and $F_2(z)$ can be developed as follows. By definition

$$F_3(z) = \sum_{k=-\infty}^{\infty} f_1(k)f_2(k)z^{-k}.$$

From the inversion integral, Eq. (4.122), we can replace $f_2(k)$ by an integral

$$F_3(z) = \sum_{k=-\infty}^{\infty} f_1(k)z^{-k} \frac{1}{2\pi j} \oint_{C_z} F_2(\xi) \xi^k \frac{d\xi}{\xi}.$$

We assume that we can find a region where we can exchange the summation with the integration. The contour will be called C_3 in this case

$$F_3(z) = \frac{1}{2\pi j} \oint_{C_3} F_2(\xi) \sum_{k=-\infty}^{\infty} f_1(k) \left(\frac{z}{\xi}\right)^{-k} \frac{d\xi}{\xi}.$$

The sum can now be recognized as $F_1(z/\xi)$ and, when we substitute this,

$$F_3(z) = \frac{1}{2\pi j} \oint_{C_3} F_2(\xi) F_1\left(\frac{z}{\xi}\right) \frac{d\xi}{\xi}. \quad (4.129)$$

the contour C_3 must be in the overlap of the convergence regions of $F_1(\xi)$ and $F_2(z/\xi)$. Then $F_3(z)$ will converge for the range of values of z for which C_3 can be found.

If we let $f_1 = f_2$ and $z = 1$ in Eq. (4.129), we have the discrete version of Parseval's theorem, where convergence is on the unit circle

$$F_3(1) = \sum_{k=-\infty}^{\infty} f_1^2 = \frac{1}{2\pi j} \oint_C F_1(\xi) F_1\left(\frac{1}{\xi}\right) \frac{d\xi}{\xi}. \quad (4.130)$$

This particular theorem shows how we can compute the sum of squares of a time sequence by evaluating a complex integral in the z -domain. The result is useful in the design of systems by least squares.

4.6.3 *Another Derivation of the Transfer Function

Let \mathcal{D} be a discrete system which maps an input sequence, $\{e(k)\}$, into an output sequence $\{u(k)\}$.²³ Then, expressing this as an operator on $e(k)$, we have

$$u(k) = \mathcal{D}\{e(k)\}.$$

If \mathcal{D} is linear, then

$$\mathcal{D}(\alpha e_1(k) + \beta e_2(k)) = \alpha \mathcal{D}\{e_1(k)\} + \beta \mathcal{D}\{e_2(k)\}. \quad (4.131)$$

If the system is time invariant, a shift in $e(k)$ to $e(k+j)$ must result in no other effects but a shift in the response, u . We write

$$\mathcal{D}\{e(k+j)\} = u(k+j) \quad \text{for all } j \quad (4.132)$$

if

$$\mathcal{D}\{e(k)\} = u(k).$$

Theorem

If \mathcal{D} is linear and time invariant and is given an input z^k for a value of z for which the output is finite at time k , then the output will be of the form $H(z)z^k$.

In general, if $e(k) = z^k$, then an arbitrary finite response can be written

$$u(k) = H(z, k)z^k.$$

Consider $e_j(k) = z^{k+j} = z^j z^k$ for some fixed j . From Eq. (4.131), if we let $\alpha = z^j$, it must follow that

$$\begin{aligned} u_j &= z^j u(k) \\ &= z^j H(z, k)z^k \\ &= H(z, k)z^{k+j}. \end{aligned} \quad (4.133)$$

From Eq. (4.132), we must have

$$\begin{aligned} u_j(k) &= u(k+j) \\ &= H(z, j+k)z^{k+j} \quad \text{for all } j. \end{aligned} \quad (4.134)$$

From a comparison of Eqs. (4.133) and (4.134), it follows that

$$H(z, k) = H(z, k+j) \quad \text{for all } j$$

that is, H does not depend on the second argument and can be written $H(z)$. Thus for the elemental signal $e(k) = z^k$, we have a solution $u(k)$ of the same (exponential) shape but modulated by a ratio $H(z)$, $u(k) = H(z)z^k$.

²³ This derivation was suggested by L. A. Zadeh in 1952 at Columbia University.

Can we represent a general signal as a *linear sum* (integral) of such elements? We can, by the inverse integral derived above, as follows

$$e(k) = \frac{1}{2\pi j} \oint E(z)z^k \frac{dz}{z}, \quad (4.135)$$

where

$$E(z) = \sum_{k=-\infty}^{\infty} e(k)z^{-k}, \quad r < |z| < R. \quad (4.136)$$

for signals with $r < R$ for which Eq. (4.136) converges. We call $E(z)$ the z -transform of $e(k)$, and the (closed) path of integration is in the annular region of convergence of Eq. (4.136). If $e(k) = 0$, $k < 0$, then $R \rightarrow \infty$, and this region is the whole z -plane *outside* a circle of finite radius.

The consequences of linearity are that the response to a sum of signals is the sum of the responses as given in Eq. (4.131). Although Eq. (4.135) is the limit of a sum, the result still holds, and we can write

$$u(k) = \frac{1}{2\pi j} \oint E(z)[\text{response to } z^k] \frac{dz}{z},$$

but, by the theorem, the response to z^k is $H(z)z^k$. Therefore we can write

$$\begin{aligned} u(k) &= \frac{1}{2\pi j} \oint E(z)[H(z)z^k] \frac{dz}{z} \\ &= \frac{1}{2\pi j} \oint H(z)E(z)z^k \frac{dz}{z}. \end{aligned} \quad (4.137)$$

We can define $U(z) = H(z)E(z)$ by comparison with Eq. (4.135) and note that

$$U(z) = \sum_{k=-\infty}^{\infty} u(k)z^{-k} = H(z)E(z). \quad (4.138)$$

Thus $H(z)$ is the *transfer function*, which is the ratio of the transforms of $e(k)$ and $u(k)$ as well as the amplitude response to inputs of the form z^k .

This derivation begins with linearity and stationarity and derives the z -transform as the natural tool of analysis from the fact that input signals in the form z^k produce an output that has the same shape.²⁴ It is somewhat more satisfying to derive the necessary transform than to start with the transform and see what systems it is good for. Better to start with the problem and find a tool than start with a tool and look for a problem. Unfortunately, the direct approach requires extensive use of the inversion integral and more sophisticated analysis to develop the main result, which is Eq. (4.138). *Chacun à son goût.*

²⁴ Because z^k is unchanged in shape by passage through the linear constant system, we say that z^k is an eigenfunction of such systems.

4.7 Summary

- The z -transform can be used to solve discrete difference equations in the same way that the Laplace transform is used to solve continuous differential equations.
- The key property of the z -transform that allows solution of difference equations is

$$\mathcal{Z}\{f(k-1)\} = z^{-1}F(z). \quad (4.113)$$

- A system will be stable in the sense that a *Bounded Input* will yield a *Bounded Output* (BIBO stability) if

$$\sum_{k=-\infty}^{\infty} |h_{k-1}| < \infty. \quad (4.35)$$

- A discrete system can be defined by its transfer function (in z) or its state-space difference equation.
- The z -transform of the samples of a continuous system $G(s)$ preceded by a zero-order-hold (ZOH) is

$$G(z) = (1 - z^{-1})\mathcal{Z}\left\{\frac{G(s)}{s}\right\} \quad (4.41)$$

which is typically evaluated using MATLAB's c2d.m.

- For the continuous state-space model

$$\dot{x} = Fx + Gu, \quad (4.45)$$

$$y = Hx + Ju, \quad (4.46)$$

preceded by a zero-order-hold, the discrete state-space difference equations are

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k), \\ y(k) &= Hx(k) + Ju(k), \end{aligned} \quad (4.59)$$

where

$$\begin{aligned} \Phi &= e^{Ft} \\ \Gamma &= \int_0^T e^{F\eta} d\eta G, \end{aligned} \quad (4.58)$$

which can be evaluated by MATLAB's c2d.m.

- The discrete transfer function in terms of the state-space matrices is

$$\frac{Y(z)}{U(z)} = H[zI - \Phi]^{-1}\Gamma. \quad (4.64)$$

which can be evaluated in MATLAB by the tf function.

- The characteristic behavior associated with poles in the z -plane is shown in Figs. 4.21 through 4.23 and summarized in Fig. 4.25. Responses are typically determined via MATLAB's impulse.m or step.m.
- A system represented by $H(z)$ has a discrete frequency response to sinusoids at ω_n , given by amplitude, A , and phase, ψ ,

$$A = |H(e^{j\omega_n T})| \quad \text{and} \quad \psi = \angle(H(e^{j\omega_n T})) \quad (4.110)$$

which can be evaluated by MATLAB's bode.m.

- The discrete Final Value Theorem, for an $F(z)$ that converges and has a final value, is given by

$$\lim_{k \rightarrow \infty} f(k) = \lim_{z \rightarrow 1} (z-1)F(z). \quad (4.115)$$

4.8 Problems

- 4.1 Check the following for stability:

- (a) $u(k) = 0.5u(k-1) - 0.3u(k-2)$
- (b) $u(k) = 1.6u(k-1) - u(k-2)$
- (c) $u(k) = 0.8u(k-1) + 0.4u(k-2)$

- 4.2 (a) Derive the difference equation corresponding to the approximation of integration found by fitting a parabola to the points e_{k-2}, e_{k-1}, e_k and taking the area under this parabola between $t = kT - T$ and $t = kT$ as the approximation to the integral of $e(t)$ over this range.

- (b) Find the transfer function of the resulting discrete system and plot the poles and zeros in the z -plane.

- 4.3 Verify that the transfer function of the system of Fig. 4.8(c) is given by the same $H(z)$ as the system of Fig. 4.9(c).

- 4.4 (a) Compute and plot the unit-pulse response of the system derived in Problem 4.2.
(b) Is this system BIBO stable?

- 4.5 Consider the difference equation

$$u(k+2) = 0.25u(k).$$

- (a) Assume a solution $u(k) = A_1 z^k$ and find the characteristic equation in z .
- (b) Find the characteristic roots z_1 and z_2 and decide if the equation solutions are stable or unstable.

- (c) Assume a general solution of the form

$$u(k) = A_1 z_1^k + A_2 z_2^k$$

and find A_1 and A_2 to match the initial conditions $u(0) = 0, u(1) = 1$.

- (d) Repeat parts (a), (b), and (c) for the equation

$$u(k+2) = -0.25u(k).$$

- (e) Repeat parts (a), (b), and (c) for the equation

$$u(k+2) = u(k+1) - 0.5u(k).$$
- 4.6** Show that the characteristic equation

$$z^2 - 2r \cos(\theta)z + r^2$$
- has the roots

$$z_{1,2} = r e^{\pm j\theta}.$$
- 4.7** (a) Use the method of block-diagram reduction, applying Figs. 4.5, 4.6, and 4.7 to compute the transfer function of Fig. 4.8(c).
(b) Repeat part (a) for the diagram of Fig. 4.9(c).
- 4.8** Use MATLAB to determine how many roots of the following are outside the unit circle.
(a) $z^2 + 0.25 = 0$
(b) $z^2 - 1.1z^2 + 0.01z + 0.405 = 0$
(c) $z^3 - 3.6z^2 + 4z - 1.6 = 0$
- 4.9** Compute by hand and table look-up the discrete transfer function if the $G(s)$ in Fig. 4.12 is
(a) $\frac{K}{s}$
(b) $\frac{1}{(s+1)(s+3)}$
(c) $\frac{3}{(s+1)(s+3)}$
(d) $\frac{(s+1)}{s^2}$
(e) $\frac{e^{j\pi/2}}{s^2}$
(f) $\frac{(1-s)}{s^2}$
(g) $\frac{3e^{-1.9s}}{(s+1)(s+3)}$
(h) Repeat the calculation of these discrete transfer functions using MATLAB. Compute for the sampling period $T = 0.05$ and $T = 0.5$ and plot the location of the poles and zeros in the z -plane.
- 4.10** Use MATLAB to compute the discrete transfer function if the $G(s)$ in Fig. 4.12 is
(a) the two-mass system with the non-collocated actuator and sensor of Eq. (A.21) with sampling periods $T = 0.02$ and $T = 0.1$. Plot the zeros and poles of the results in the z -plane. Let $\omega_p = 5$, $\zeta_p = 0.01$.
(b) the two-mass system with the colocated actuator and sensor given by Eq. (A.23). Use $T = 0.02$ and $T = 0.1$. Plot the zeros and poles of the results in the z -plane. Let $\omega_p = 5$, $\omega_s = 3$, $\zeta_p = \zeta_s = 0$.
(c) the two-input-two-output paper machine described in Eq. (A.24). Let $T = 0.1$ and $T = 0.5$.
- 4.11** Consider the system described by the transfer function

$$\frac{Y(s)}{U(s)} = G(s) = \frac{3}{(s+1)(s+3)}.$$
- (a) Draw the block diagram corresponding to this system in control canonical form, define the state vector, and give the corresponding description matrices F , G , H .

- (b) Write $G(s)$ in partial fractions and draw the corresponding parallel block diagram with each component part in control canonical form. Define the state ξ and give the corresponding state description matrices A , B , C , D .
(c) By finding the transfer functions X_1/U and X_2/U of part (a) in partial fraction form, express x_1 and x_2 in terms of ξ_1 and ξ_2 . Write these relations as the two-by-two transformation T such that $x = T\xi$.
(d) Verify that the matrices you have found are related by the formulas

$$A = T^{-1}FT$$

$$B = T^{-1}G$$

$$C = HT$$

$$D = J$$
- 4.12** The first-order system $(z - \alpha)/(1 - \alpha)z$ has a zero at $z = \alpha$.
(a) Plot the step response for this system for $\alpha = 0.8, 0.9, 1.1, 1.2, 2$.
(b) Plot the overshoot of this system on the same coordinates as those appearing in Fig. 4.30 for $1 < \alpha < 1$.
(c) In what way is the step response of this system unusual for $\alpha > 1$?
- 4.13** The one-sided z -transform is defined as

$$F(z) = \sum_{k=0}^{\infty} f(k)z^{-k}.$$
- (a) Show that the one-sided transform of $f(k+1)$ is

$$\mathcal{Z}\{f(k+1)\} = zF(z) - zf(0).$$
- (b) Use the one-sided transform to solve for the transforms of the Fibonacci numbers by writing Eq. (4.4) as $u_{k+2} = u_{k+1} + u_k$. Let $u_0 = u_1 = 1$. [You will need to compute the transform of $f(k+2)$.]
(c) Compute the location of the poles of the transform of the Fibonacci numbers.
(d) Compute the inverse transform of the numbers.
(e) Show that if u_k is the k th Fibonacci number, then the ratio u_{k+1}/u_k will go to $(1 + \sqrt{5})/2$, the golden ratio of the Greeks.
(f) Show that if we add a forcing term, $e(k)$, to Eq. (4.4) we can generate the Fibonacci numbers by a system that can be analyzed by the two-sided transform: i.e., let $u_k = u_{k-1} + u_{k-2} + e_k$ and let $e_k = \delta_0(k)$ ($\delta_0(k) - 1$ at $k = 0$ and zero elsewhere). Take the two-sided transform and show the same $U(z)$ results as in part (b).
- 4.14** Substitute $u = A_z^k$ and $e = B_z^k$ into Eqs. (4.2) and (4.7) and show that the transfer functions, Eqs. (4.15) and (4.14), can be found in this way.
- 4.15** Consider the transfer function

$$H(z) = \frac{(z+1)(z^2 - 1.3z + 0.81)}{(z^2 - 1.2z + 0.5)(z^2 - 1.4z + 0.81)}.$$

Draw a cascade realization, using observer canonical forms for second-order blocks and in such a way that the coefficients as shown in $H(z)$ above are the parameters of the block diagram.

- 4.16 (a)** Write the $H(z)$ of Problem 4.15 in partial fractions in two terms of second order each, and draw a *parallel* realization, using the observer canonical form for each block and showing the coefficients of the partial-fraction expansion as the parameters of the realization.

- (b)** Suppose the two factors in the denominator of $H(z)$ were identical (say we change the 1.4 to 1.2 and the 0.81 to 0.5). What would the parallel realization be in this case?

- 4.17** Show that the observer canonical form of the system equations shown in Fig. 4.9 can be written in the state-space form as given by Eq. (4.77).

- 4.18** Draw out each block of Fig. 4.10 in (a) control and (b) observer canonical form. Write out the state-description matrices in each case.

- 4.19** For a second-order system with damping ratio 0.5 and poles at an angle in the z -plane of $\theta = 30^\circ$, what percent overshoot to a step would you expect if the system had a zero at $z_2 = 0.6$?

- 4.20** Consider a signal with the transform (which converges for $|z| > 2$)

$$U(z) = \frac{z}{(z-1)(z-2)}.$$

- (a) What value is given by the formula (Final Value Theorem) of (2.100) applied to this $U(z)$?
- (b) Find the final value of $u(k)$ by taking the inverse transform of $U(z)$, using partial-fraction expansion and the tables.

- (c) Explain why the two results of (a) and (b) differ.

- 4.21 (a)** Find the z -transform and be sure to give the region of convergence for the signal

$$u(k) = r^{-|k|}, \quad r < 1.$$

[Hint: Write u as the sum of two functions, one for $k \geq 0$ and one for $k < 0$, find the individual transforms, and determine values of z for which *both* terms converge.]

- (b) If a rational function $U(z)$ is known to converge on the unit circle $|z| = 1$, show how partial-fraction expansion can be used to compute the inverse transform. Apply your result to the transform you found in part (a).

- 4.22** Compute the inverse transform, $f(k)$, for each of the following transforms:

- (a) $F(z) = \frac{1}{1+z^{-2}}$, $|z| > 1$;
 (b) $F(z) = \frac{z(z-1)}{z^2-1.2z+0.35}$, $|z| > 1$;
 (c) $F(z) = \frac{z^2-2z+1}{z^2-2z+1}$, $|z| > 1$;
 (d) $F(z) = \frac{z^2}{(z-\frac{1}{2})(z-2)}$, $1/2 < |z| < 2$.

- 4.23** Use MATLAB to plot the time sequence associated with each of the transforms in Problem 4.22.

- 4.24** Use the z -transform to solve the difference equation

$$y(k) - 3y(k-1) + 2y(k-2) = 2u(k-1) - 2u(k-2),$$

$$\begin{aligned} u(k) &= \begin{cases} k, & k \geq 0 \\ 0 & k < 0 \end{cases} \\ y(k) &= 0, \quad k < 0. \end{aligned}$$

- 4.25** For the difference equation in Problem 4.24, solve using MATLAB.

- 4.26** Compute by hand and table look-up the discrete transfer function if the $G(s)$ in Fig. 4.12 is

$$G(s) = \frac{10(s+1)}{s^2+s+10}$$

and the sample period is $T = 10$ msec. Verify the calculation using MATLAB.

- 4.27** Find the discrete state-space model for the system in Problem 4.26.

- 4.28** Compute by hand and table look-up the discrete transfer function if the $G(s)$ in Fig. 4.12 is

$$G(s) = \frac{10(s+1)}{s^2+s+10}$$

and the sample period is $T = 10$ msec. Verify the calculation using MATLAB and find the DC gain of both the $G(x)$ and the $G(z)$.

- 4.29** Find the discrete state-space model for the system in Problem 4.28. Then compute the eigenvalues of Φ and the transmission zeros of the state-space model.

- 4.30** Find the state-space model for Fig. 4.12 with

$$G(s) = \frac{1}{s^2}$$

where there is a one cycle delay after the A/D converter.

• 5 •

Sampled-Data Systems

A Perspective on Sampled-Data Systems

The use of digital logic or digital computers to calculate a control action for a continuous, dynamic system introduces the fundamental operation of sampling. Samples are taken from the continuous physical signals such as position, velocity, or temperature and these samples are used in the computer to calculate the controls to be applied. Systems where discrete signals appear in some places and continuous signals occur in other parts are called *sampled-data systems* because continuous data are sampled before being used. In many ways the analysis of a purely continuous system or of a purely discrete system is simpler than that of sampled-data systems. The analysis of linear, time-invariant continuous systems can be done with the Laplace transform and the analysis of linear time-invariant discrete systems can be done with the z -transform alone. If one is willing to restrict attention to only the samples of all the signals in a digital control one can do much useful analysis and design on the system as a purely discrete system using the z -transform. However the physical reality is that the computer operations are on discrete signals while the plant signals are in the continuous world and in order to consider the behavior of the plant between sampling instants, it is necessary to consider both the discrete actions of the computer and the continuous response of the plant. Thus the role of sampling and the conversion from continuous to discrete and back from discrete to continuous are very important to the understanding of the complete response of digital control, and we must study the process of sampling and how to make mathematical models of analog-to-digital conversion and digital-to-analog conversion. This analysis requires careful treatment using the Fourier transform but the effort is well rewarded with the understanding it provides of sampled-data systems.

Chapter Overview

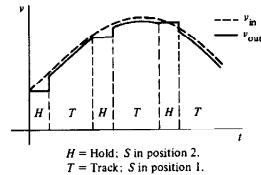
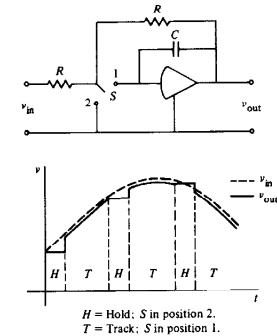
In this chapter, we introduce the analysis of the sampling process and describe both a time-domain and a frequency-domain representation. We also describe the companion process of data extrapolation or data holding to construct a continuous time signal from samples. As part of this analysis we show that a sampled-data system is made time varying by the introduction of sampling, and thus it is not possible to describe such systems exactly by a continuous-time transfer function. However, a continuous signal is recovered by the hold process and we can approximate the sinusoidal response of a sampler and hold by fitting another sinusoid of the same frequency to the complete response. We show how to compute this best-fit sinusoidal response analytically and use it to obtain a good approximation to a transfer function. For those familiar with the idea, this approach is equivalent to the use of the “describing function” that is used to approximate a transfer function for simple nonlinear systems. In Section 5.1 the analysis of the sample and hold operation is considered and in Section 5.2 the frequency analysis of a sampled signal is given. Here the important phenomenon of signal aliasing caused by sampling is introduced. In Section 5.3 the zero-order hold and some of its generalizations are considered. Analysis of sampled-data systems in the frequency domain is introduced in Section 5.4 including block diagram analysis of these combined systems. Finally in Section 5.5 computation of intersample ripple is discussed.

5.1 Analysis of the Sample and Hold

To get samples of a physical signal such as a position or a velocity into digital form, we typically have a sensor that produces a voltage proportional to the physical variable and an **analog-to-digital converter**, commonly called an **A/D converter** or ADC, that transforms the voltage into a digital number. The physical conversion always takes a non-zero time, and in many instances this time is significant with respect to the sample period of the control or with respect to the rate of change of the signal being sampled. In order to give the computer an accurate representation of the signal exactly at the sampling instants kT , the A/D converter is typically preceded by a **sample-and-hold circuit** (SHC). A simple electronic schematic is sketched in Fig. 5.1, where the switch, S , is an electronic device driven by simple logic from a clock. Its operation is described in the following paragraph.

With the switch, S , in position 1, the amplifier output $v_{\text{out}}(t)$ tracks the input voltage $v_{\text{in}}(t)$ through the transfer function $1/(RCs + 1)$. The circuit bandwidth of the SHC, $1/RC$, is selected to be high compared to the input signal bandwidth. Typical values are $R = 1000$ ohms, $C = 30 \times 10^{-12}$ farads for a bandwidth of $f = 1/2\pi RC = 5.3$ MHz. During this “tracking time,” the ADC is turned off and ignores v_{out} . When a sample is to be taken at $t = kT$ the switch S is set

Figure 5.1
Analog-to-digital converter with sample and hold



to position 2 and the capacitor C holds the output of the operational amplifier frozen from that time at $v_{\text{out}}(kT) = v_{\text{in}}(kT)$. The ADC is now signaled to begin conversion of the constant input from the SHC into a digital number which will be a true representation of the input voltage at the sample instant. When the conversion is completed, the digital number is presented to the computer at which time the calculations based on this sample value can begin. The SHC switch is now moved to position 1, and the circuit is again tracking, waiting for the next command to freeze a sample. The SHC needs only to hold the voltage for a short time on the order of microseconds in order for the conversion to be completed before it starts tracking again. The value converted is held inside the computer for the entire sampling period of the system, so the combination of the electronic SHC plus the ADC operate as a sample-and-hold for the sampling period, T , which may be many milliseconds long. The number obtained by the ADC is a quantized version of the signal represented in a finite number of bits, 12 being a typical number. As a result, the device is nonlinear. However, the signals are typically large with respect to the smallest quantum and the effect of this nonlinearity can be ignored in a first analysis. A detailed study of quantization is included in Chapter 10.

For the purpose of the analysis, we separate the sample and hold into two mathematical operations: a sampling operation represented by impulse modulation and a hold operation represented as a linear filter. The symbol or schematic of the ideal sampler is shown in Fig. 5.2; its role is to give a mathematical representation of the process of taking periodic samples from $r(t)$ to produce $r(kT)$ and

Figure 5.2

The sampler

$$r(t) \xrightarrow{T} r^*(t) = \sum_{k=-\infty}^{\infty} r(t)\delta(t - kT)$$

to do this in such a way that we can include the sampled signals in the analysis of continuous signals using the Laplace transform.¹ The technique is to use *impulse modulation* as the mathematical representation of sampling. Thus, from Fig. 5.2, we picture the output of the sampler as a string of impulses

$$r^*(t) = \sum_{k=-\infty}^{\infty} r(t)\delta(t - kT). \quad (5.1)$$

The impulse can be visualized as the limit of a pulse of unit area that has growing amplitude and shrinking duration. The essential property of the impulse is the sifting property that

$$\int_{-\infty}^{\infty} f(t)\delta(t - a)dt = f(a) \quad (5.2)$$

for all functions f that are continuous at a . The integral of the impulse is the unit step

$$\int_{-\infty}^t \delta(\tau)d\tau = 1(t), \quad (5.3)$$

and the Laplace transform of the unit impulse is 1, because

$$\mathcal{L}\{\delta(t)\} = \int_{-\infty}^{\infty} \delta(\tau)e^{-s\tau}d\tau = 1. \quad (5.4)$$

Using these properties we can see that $r^*(t)$, defined in Eq. (5.1), depends only on the discrete sample values $r(kT)$. The Laplace transform of $r^*(t)$ can be computed as follows

$$\mathcal{L}\{r^*(t)\} = \int_{-\infty}^{\infty} r^*(\tau)e^{-s\tau}d\tau.$$

If we substitute Eq. (5.1) for $r^*(t)$, we obtain

$$= \int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} r(\tau)\delta(\tau - kT)e^{-s\tau}d\tau,$$

and now, exchanging integration and summation and using Eq. (5.2), we have

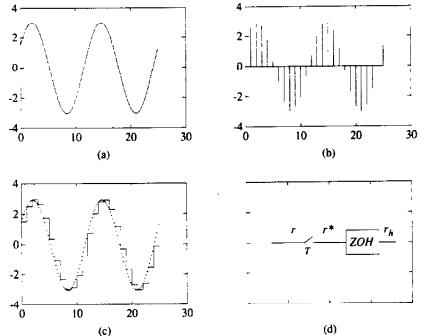
$$R^*(s) = \sum_{k=-\infty}^{\infty} r(kT)e^{-skT}. \quad (5.5)$$

¹ We assume that the reader has some familiarity with Fourier and Laplace transform analysis. A general reference is Bracewell (1978).

The notation $R^*(s)$ is used to symbolize the (Laplace) transform of $r^*(t)$, the sampled or impulse-modulated $r(t)$.² Notice that if the signal $r(t)$ in Eq. (5.1) is shifted a small amount then different samples will be selected by the sampling process for the output proving that sampling is not a time-invariant process. Consequently one must be very careful in using transform analysis in this context.

Having a model of the sampling operation as impulse modulation, we need to model the hold operation to complete the description of the physical sample-and-hold which will take the impulses that are produced by the mathematical sampler and produce the piecewise constant output of the device. Typical signals are sketched in Fig. 5.3. Once the samples are taken, as represented by $r^*(t)$ in

Figure 5.3
The sample and hold, showing typical signals
(a) input signal r ;
(b) sampled signal r^* ;
(c) output signal r_h ;
(d) sample and hold



2 It will be necessary, from time to time, to consider sampling a signal that is not continuous. The only case we will consider will be equivalent to applying a step function, $1(t)$, to a sampler. For the purposes of this book we will define the unit step to be continuous from the right and assume that the impulse, $\delta(t)$, picks up the full value of unity. By this convention and Eq. (5.1) we compute

$$1^*(t) = \sum_{k=0}^{\infty} \delta(t - kT). \quad (a)$$

and, using Eq. (5.2), we obtain

$$\mathcal{L}\{1^*(t)\} = 1/(1 - e^{-Ts}). \quad (b)$$

The reader should be warned that the Fourier integral converges to the *average* value of a function at a discontinuity and not the value approached from the right as we assume. Because our use of the transform theory is elementary and the convenience of equation (b) above is substantial, we have selected the continuous-from-the-right convention. In case of doubt, the discontinuous term should be separated and treated by special analysis, perhaps in the time domain.

Eq. (5.1), the hold is defined as the means whereby these impulses are extrapolated to the piecewise constant signal $r_h(t)$, defined as

$$r_h(t) = r(kT) \quad kT \leq t < kT + T. \quad (5.6)$$

A general technique of data extrapolation from samples is to use a polynomial fit to the past samples. If the extrapolation is done by a constant, which is a zero-order polynomial, then the extrapolator is called a **zero-order hold**, and its transfer function is designated as $ZOH(s)$. We can compute $ZOH(s)$ as the transform of its impulse response.³ If $r^*(t) = \delta(t)$, then $r_h(t)$, which is now the impulse response of the ZOH , is a pulse of height 1 and duration T seconds. The mathematical representation of the impulse response is simply

$$p(t) = 1(t) - 1(t - T).$$

The required transfer function is the Laplace transform of $p(t)$ as

$$\begin{aligned} ZOH(s) &= \mathcal{L}(p(t)) \\ &= \int_0^\infty [1(t) - 1(t - T)]e^{-st} dt \\ &= (1 - e^{-sT})/s. \end{aligned} \quad (5.7)$$

Thus the linear behavior of an A/D converter with sample and hold can be modeled by Fig. 5.3. We must emphasize that the impulsive signal $r^*(t)$ in Fig. 5.3 is not expected to represent a physical signal in the A/D converter circuit; rather it is a hypothetical signal introduced to allow us to obtain a transfer-function model of the hold operation and to give an input-output model of the sample-and-hold suitable for transform and other linear systems analysis.

5.2 Spectrum of a Sampled Signal

We can get further insight into the process of sampling by an alternative representation of the transform of $r^*(t)$ using Fourier analysis. From Eq. (5.1) we see that $r^*(t)$ is a product of $r(t)$ and the train of impulses, $\sum \delta(t - kT)$. The latter series, being periodic, can be represented by a Fourier series

$$\sum_{k=-\infty}^{\infty} \delta(t - kT) = \sum_{n=-\infty}^{\infty} C_n e^{j(2\pi n/T)t},$$

where the Fourier coefficients, C_n , are given by the integral over one period as

$$C_n = \frac{1}{T} \int_{-T/2}^{T/2} \sum_{k=-\infty}^{\infty} \delta(t - kT) e^{-jn(2\pi t/T)} dt.$$

³ The hold filter in Fig. 5.3(d) will receive one unit-size impulse if the input signal is zero at every sample time except $t = 0$ and is equal to 1 there. That is, if $r(kT) = 0$, $k \neq 0$ and $r(0) = 1$.

The only term in the sum of impulses that is in the range of the integral is the $\delta(t)$ at the origin, so the integral reduces to

$$C_n = \frac{1}{T} \int_{-T/2}^{T/2} \delta(t) e^{-jn(2\pi t/T)} dt;$$

but the sifting property from Eq. (5.2) makes this easy to integrate, with the result

$$C_n = \frac{1}{T}.$$

Thus we have derived the representation for the sum of impulses as a Fourier series

$$\sum_{k=-\infty}^{\infty} \delta(t - kT) = \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{j(2\pi n/T)t}. \quad (5.8)$$

We define $\omega_s = 2\pi/T$ as the sampling frequency (in radians per second) and now substitute Eq. (5.8) into Eq. (5.1) using ω_s . We take the Laplace transform of the output of the mathematical sampler,

$$\mathcal{L}[r^*(t)] = \int_{-\infty}^{\infty} r(t) \left\{ \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{j(n\omega_s)t} \right\} e^{-st} dt$$

and integrate the sum, term by term to obtain

$$R^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} r(t) e^{j(n\omega_s)t} e^{-st} dt.$$

If we combine the exponentials in the integral, we get

$$R^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} r(t) e^{-(s-jn\omega_s)t} dt.$$

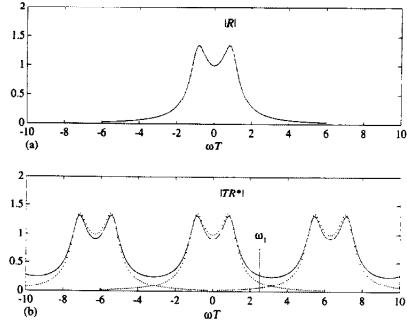
The integral here is the Laplace transform of $r(t)$ with only a change of variable where the frequency goes. The result can therefore be written as

$$R^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} R(s - jn\omega_s), \quad (5.9)$$

where $R(s)$ is the transform of $r(t)$. In communication or radio engineering terms, Eq. (5.8) expresses the fact that the impulse train corresponds to an infinite sequence of carrier frequencies at integral values of $2\pi/T$, and Eq. (5.9) shows that when $r(t)$ modulates all these carriers, it produces a never-ending train of sidebands. A sketch of the elements in the sum given in Eq. (5.9) is shown in Fig. 5.4.

An important feature of sampling, shown in Fig. 5.4, is illustrated at the frequency marked ω_1 . Two curves are drawn representing two of the elements that enter into the sum given in Eq. (5.9). The value of the larger amplitude component located at the frequency ω_1 is the value of $R(j\omega_1)$. The smaller

Figure 5.4
(a) Sketch of a spectrum amplitude and (b) the components of the spectrum after sampling, showing aliasing



aliasing

component shown at ω_1 comes from the spectrum centered at $2\pi/T$ and is $R(j\omega_0)$, where ω_0 is such that $\omega_0 = \omega_1 - 2\pi/T$. This signal at frequency ω_0 which produces a component at frequency ω_1 after sampling is called in the trade an “alias” of ω_1 ; the phenomenon is called **aliasing**.

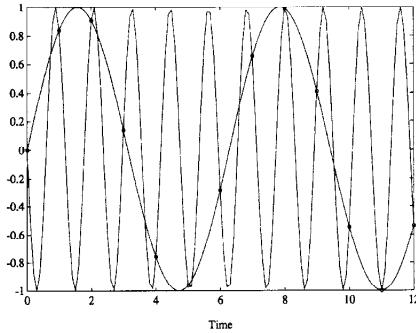
The phenomenon of aliasing has a clear meaning in time. Two continuous sinusoids of different frequencies appear at the same frequency when sampled. We cannot, therefore, distinguish between them based on their samples alone. Figure 5.5 shows a plot of a sinusoid at $\frac{1}{8}$ Hz and of a sinusoid at $\frac{7}{8}$ Hz. If we sample these waves at 1 Hz, as indicated by the dots, then we get the same sample values from both signals and would continue to get the same sample values for all time. Note that the sampling frequency is 1, and, if $f_s = \frac{1}{8}$, then

$$f_s = \frac{1}{8} - 1 = -\frac{7}{8}.$$

The significance of the negative frequency is that the $\frac{7}{8}$ -Hz sinusoid in Fig. 5.5 is a negative sine function.

Thus, as a direct result of the sampling operation, when data are sampled at frequency $2\pi/T$, the total harmonic content at a given frequency ω_1 is to be found not only from the original signal at ω_1 but also from all those frequencies that are aliases of ω_1 , namely, components from all frequencies $\omega_1 + n2\pi/T = \omega_1 + n\omega_s$ as shown in the formula of Eq. (5.9) and sketched in Fig. 5.4. The errors caused by aliasing can be very severe if a substantial quantity of high-frequency components is contained in the signal to be sampled. To minimize the error caused by this

Figure 5.5
Plot of two sinusoids that have identical values at unit sampling intervals—an example of aliasing



sampling theorem

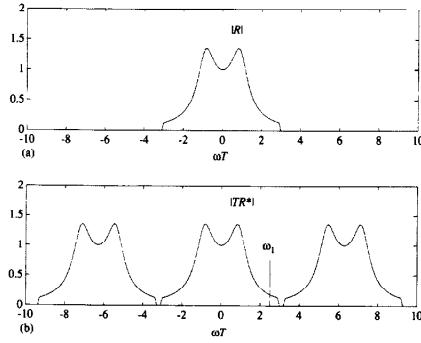
hidden oscillations

effect, it is standard practice to *precede* the sampling operation (such as the sample-and-hold circuit discussed earlier) by a low-pass antialias filter that will remove substantially all spectral content above the half-sampling frequency, i.e., above π/T . A sketch suggesting the result of an anti-aliasing filter is drawn in Fig. 5.6.

If all spectral content above the frequency π/T is removed, then no aliasing is introduced by sampling and the signal spectrum is not distorted, even though it is repeated endlessly, centered at $n2\pi/T$. The critical frequency, π/T , was first reported by H. Nyquist and is called the Nyquist frequency. Band-limited signals that have no components above the Nyquist frequency are represented unambiguously by their samples. A corollary to the aliasing issue is the **sampling theorem**. We have seen that if $R(j\omega)$ has components above the Nyquist frequency $\omega_s/2$ or π/T , then overlap and aliasing will occur. Conversely, we noticed that if $R(j\omega)$ is zero for $|\omega| \geq \pi/T$, then sampling at intervals of T sec. will produce no aliasing and the original spectrum can be recovered exactly from R^* , the spectrum of the samples. Once the spectrum is recovered by inverse transform, we can calculate the original signal itself. This is the sampling theorem: One can recover a signal from its samples if the sampling frequency ($\omega_s = 2\pi/T$) is *at least twice* the highest frequency (π/T) in the signal. Notice that the sampling theorem requires that $R(j\omega)$ is exactly zero for all frequencies above π/T .

A phenomenon somewhat related to aliasing is that of **hidden oscillations**. There is the possibility that a signal could contain some frequencies that the

Figure 5.6
(a) Sketch of a spectrum amplitude and (b) the components of the spectrum after sampling, showing removal of aliasing with an antialiasing filter



samples do not show *at all*. Such signals, when they occur in a digital control system, are called ‘‘hidden oscillations,’’ an example of which is shown in a design problem in Fig. 7.29. Hidden oscillations can only occur at multiples of the Nyquist frequency (π/T).

5.3 Data Extrapolation

The sampling theorem states that under the right conditions it is possible to recover a signal from its samples; we now consider a formula for doing so. From Fig. 5.6 we can see that the spectrum of $R(j\omega)$ is contained in the low-frequency part of $R^*(j\omega)$. Therefore, to recover $R(j\omega)$ we need only process $R^*(j\omega)$ through a low-pass filter and multiply by T . As a matter of fact, if $R(j\omega)$ has zero energy for frequencies in the bands above π/T (such an R is said to be band-limited), then an ideal low-pass filter with gain T for $-\pi/T \leq \omega \leq \pi/T$ and zero elsewhere would recover $R(j\omega)$ from $R^*(j\omega)$ exactly. Suppose we define this ideal low-pass filter characteristic as $L(j\omega)$. Then we have the result

$$R(j\omega) = L(j\omega)R^*(j\omega). \quad (5.10)$$

The signal $r(t)$ is the inverse transform of $R(j\omega)$, and because by Eq. (5.10) $R(j\omega)$ is the product of two transforms, its inverse transform $r(t)$ must be the convolution of the time functions $\ell(t)$ and $r^*(t)$. The form of the filter impulse

response can be computed by using the definition of $L(j\omega)$ from which the inverse transform gives

$$\begin{aligned} \ell(t) &= \frac{1}{2\pi} \int_{-\pi/T}^{\pi/T} T e^{j\omega\tau} d\omega \\ &= \frac{T}{2\pi} \frac{e^{j\omega t}}{j\tau} \Big|_{-\pi/T}^{\pi/T} \\ &= \frac{T}{2\pi j t} (e^{j(\pi t/T)} - e^{-j(\pi t/T)}) \\ &= \frac{\sin(\pi t/T)}{\pi t/T} \\ &\triangleq \text{sinc} \frac{\pi t}{T}. \end{aligned} \quad (5.11)$$

Using Eq. (5.1) for $r^*(t)$ and Eq. (5.11) for $\ell(t)$, we find that their convolution is

$$r(t) = \int_{-\infty}^{\infty} r(\tau) \sum_{k=-\infty}^{\infty} \delta(\tau - kT) \text{sinc} \frac{\pi(t - \tau)}{T} d\tau.$$

Using the sifting property of the impulse, we have

$$r(t) = \sum_{k=-\infty}^{\infty} r(kT) \text{sinc} \frac{\pi(t - kT)}{T}. \quad (5.12)$$

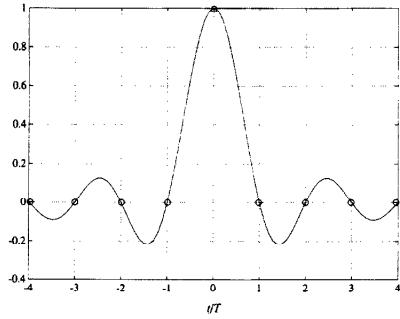
Equation (5.12) is a constructive statement of the sampling theorem. It shows explicitly how to construct a band-limited function $r(t)$ from its samples. The sinc functions are the interpolants that fill in the time gaps between samples with a signal that has no frequencies above π/T . A plot of the impulse response of this ‘‘ideal’’ hold filter is drawn in Fig. 5.7 from the formula of Eq. (5.11).

There is one serious drawback to the extrapolating signal given by Eq. (5.11). Because $\ell(t)$ is the impulse response of the ideal low-pass filter $L(j\omega)$, it follows that this filter is *noncausal* because $\ell(t)$ is nonzero for $t < 0$. $\ell(t)$ starts at $t = -\infty$ while the impulse that triggers it does not occur until $t = 0$! In many communications problems the interpolated signal is not needed until well after the samples are acquired, and the noncausality can be overcome by adding a phase lag, $e^{-j\omega_k t}$, to $L(j\omega)$, which adds a *delay* to the filter and to the signals processed through it. In feedback control systems, a large delay is usually disastrous for stability, so we avoid such approximations to this function and use something else, like the polynomial holds, of which the zero-order hold already mentioned in connection with the ADC is the most elementary and the most common.

In Section 5.2 we introduced the zero-order hold as a model for the storage register in an A/D converter that maintains a constant signal value between samples. We showed in Eq. (5.7) that it has the transfer function

$$ZOH(j\omega) = \frac{1 - e^{-j\omega T}}{j\omega}. \quad (5.13)$$

Figure 5.7
Plot of the impulse response of the ideal low-pass filter



We can discover the frequency properties of $ZOH(j\omega)$ by expressing Eq. (5.13) in magnitude and phase form. To do this, we factor out $e^{-j\omega T/2}$ and multiply and divide by $2j$ to write the transfer function in the form

$$ZOH(j\omega) = e^{-j\omega T/2} \left\{ \frac{e^{j\omega T/2} - e^{-j\omega T/2}}{2j} \right\} \frac{2j}{j\omega}.$$

The term in brackets is recognized as the sine, so this can be written

$$ZOH(j\omega) = Te^{-j\omega T/2} \frac{\sin(\omega T/2)}{\omega T/2}$$

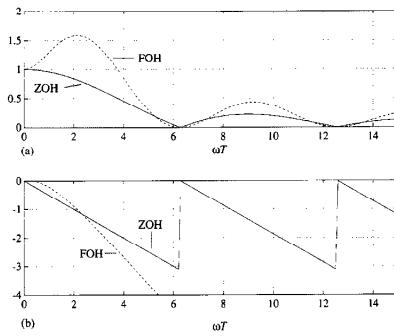
and, using the definition of the sinc function,

$$ZOH(j\omega) = e^{-j\omega T/2} T \text{sinc}(\omega T/2). \quad (5.14)$$

Thus the effect of the zero-order hold is to introduce a phase shift of $\omega T/2$, which corresponds to a time delay of $T/2$ seconds, and to multiply the gain by a function with the magnitude of $\text{sinc}(\omega T/2)$. A plot of the magnitude is shown in Fig. 5.8, which illustrates the fact that although the zero-order hold is a low-pass filter, it has a cut-off frequency well beyond the Nyquist frequency. The magnitude function is

$$|ZOH(j\omega)| = T \left| \text{sinc} \frac{\omega T}{2} \right|. \quad (5.15)$$

Figure 5.8
(a) Magnitude and (b) phase of polynomial hold filters



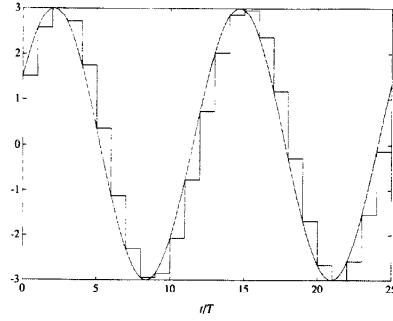
which slowly gets smaller as ω gets larger until it is zero for the first time at $\omega = \omega_s = 2\pi/T$. The phase is

$$\angle ZOH(j\omega) = \frac{-\omega T}{2}, \quad (5.16)$$

plus the 180° shifts where the sinc function changes sign.

We can now give a complete analysis of the sample-and-hold circuit of Fig. 5.3(d) for a sinusoidal input $r(t)$ in both the time and the frequency domains. We consider first the time domain, which is simpler, being just an exercise in construction. For purposes of illustration, we will use $r(t) = 3 \sin(50t + \pi/6)$ as plotted in Fig. 5.9. If we sample $r(t)$ at the instants kT where the sampling frequency is $\omega_s = 2\pi/T = 20\pi$ and $T = 0.01$, then the plot of the resulting $r_h(kT)$ is as shown in Fig. 5.9. Notice that although the input is a single sinusoid, the output is clearly *not* sinusoidal. Thus it is not possible to describe this system by a transfer function, because the fundamental property of linear, time-invariant systems is that a sinusoid input produces an output that is a sinusoid of the same frequency and the relative amplitudes and phases determine the transfer function. The sample-and-hold system is linear but time varying. In the frequency domain, it is clear that the output $r_h(t)$ contains more than one frequency, and a complete analysis requires that we compute the amplitudes and phases of them all. However, in the application to control systems, the output of the hold will typically be applied to a dynamical system that is of low-pass character; thus the most important component in $r_h(t)$ is the fundamental harmonic, at $\omega_o = 50$

Figure 5.9
Plot of $3 \sin(50t + \pi/6)$ and the output of a sample-and-hold with sample period $T = 0.01$



rad/sec in this case. The other harmonics are *impostors*, appearing as part of the output signal when they are really unwanted consequences of the sample-and-hold process. In any event, we can proceed to analyze $r_h(t)$ for all its harmonics and select out the fundamental component, either by analysis or by a low-pass smoothing filter.

First, we need the spectrum of $r(t)$. Because a sinusoid can be decomposed into two exponentials, the spectrum of $r(t) = A \cos(\omega_o t + \phi)$ is given by two impulse functions at ω_o and $-\omega_o$ of intensity πA and phase ϕ and $-\phi$ as⁴

$$R(j\omega) = \pi A [e^{j\phi} \delta(\omega - \omega_o) + e^{-j\phi} \delta(\omega + \omega_o)].$$

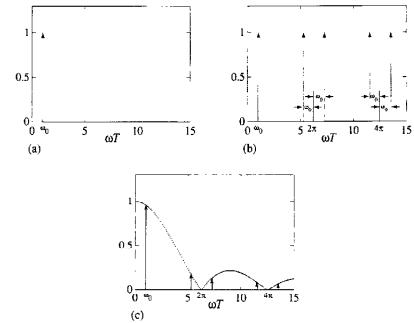
A sketch of this spectrum is shown in Fig. 5.10(a) for $A = 1/\pi$. We represent the impulses by arrows whose heights are proportional to the intensities of the impulses.

After sampling, as we saw in Eq. (5.9), the spectrum of R^* is directly derived from that of R as the sum of multiple copies of that of R shifted by $n2\pi/T$ for all integers n and multiplied by $1/T$. A plot of the result normalized by T is shown in Fig. 5.10(b). Finally, to find the spectrum of R_h , we need only multiply the spectrum of R^* by the transfer function $ZOH(j\omega)$, which is

$$ZOH(j\omega) = Te^{-j\omega T/2} \text{sinc}(\omega T/2).$$

⁴ See the appendix to this chapter for details.

Figure 5.10
Plot of the spectra of
(a) R , (b) R^* ; and (c) R_h



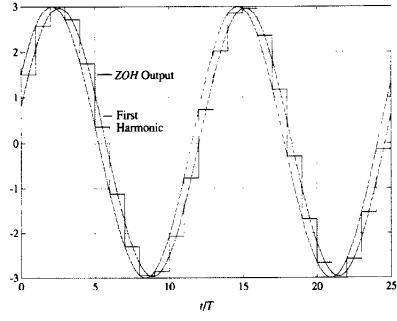
Thus the spectrum of R_h is also a sum of an infinite number of terms, but now with intensities modified by the sinc function and phases shifted by the delay function $\omega T/2$. These intensities are plotted in Fig. 5.10(c). Naturally, when all the harmonics included in R_h are converted to their time functions and added, they sum to the piecewise-constant staircase function plotted earlier in Fig. 5.9.

If we want a best approximation to r_h using only one sinusoid, we need only take out the first or fundamental harmonic from the components of R^* . This component has phase shift ϕ and amplitude $A \text{sinc}(\omega T/2)$. In the time domain, the corresponding sinusoid is given by

$$v_i(t) = A [\text{sinc}(\omega T/2)] \sin[\omega_j(t - \frac{T}{2})]. \quad (5.17)$$

A plot of this approximation for the signal from Fig. 5.9 is given in Fig. 5.11 along with both the original input and the sampled-and-held output to show the nature of the approximation. In control design, we can frequently achieve a satisfactory design for a sampled-data system by approximating the sample and hold with a continuous transfer function corresponding to the delay of $T/2$. The controller design is then done in the continuous domain but is implemented by computing a discrete equivalent. More discussion of this technique, sometimes called *emulation*, will be given in Chapter 6, where some examples illustrate the results.

Figure 5.11
Plot of the output of the sample and hold and the first harmonic approximation



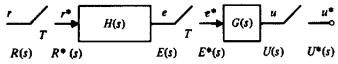
5.4 Block-Diagram Analysis of Sampled-Data Systems

We have thus far talked mainly about discrete, continuous, and sampled signals. To analyze a feedback system that contains a digital computer, we need to be able to compute the transforms of output signals of systems that contain sampling operations in various places, including feedback loops, in the block diagram. The technique for doing this is a simple extension of the ideas of block-diagram analysis of systems that are all continuous or all discrete, but one or two rules need to be carefully observed to assure success. First, we should review the facts of sampled-signal analysis.

We represent the process of sampling a continuous signal and holding it by impulse modulation followed by low-pass filtering. For example, the system of Fig. 5.12 leads to

$$\begin{aligned} E(s) &= R^*(s)H(s), \\ U(s) &= E^*(s)G(s). \end{aligned} \quad (5.18)$$

Figure 5.12
A cascade of samplers and filters



Impulse modulation of continuous-time signals like $e(t)$ and $u(t)$ produces a series of sidebands as given in Eq. (5.9) and plotted in Fig. 5.4, which result in periodic functions of frequency. If the transform of the signal to be sampled is a product of a transform that is already periodic of period $2\pi/T$, and one that is not, as in $U(s) = E^*(s)G(s)$, where $E^*(s)$ is periodic and $G(s)$ is not, we can show that $E^*(s)$ comes out as a factor of the result. This is the most important relation for the block-diagram analysis of sampled-data systems, namely⁵

$$U^*(s) = (E^*(s)G(s))^* = E^*(s)G^*(s). \quad (5.19)$$

We can prove Eq. (5.19) either in the frequency domain, using Eq. (5.9), or in the time domain, using Eq. (5.1) and convolution. We will use Eq. (5.9) here. If $U(s) = E^*(s)G(s)$, then by definition we have

$$U^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E^*(s - jn\omega_i)G(s - jn\omega_i); \quad (5.20)$$

but $E^*(s)$ is

$$E^*(s) = \frac{1}{T} \sum_{k=-\infty}^{\infty} E(s - jk\omega_i),$$

so that

$$E^*(s - jn\omega_i) = \frac{1}{T} \sum_{k=-\infty}^{\infty} E(s - jk\omega_i - jn\omega_i). \quad (5.21)$$

Now in Eq. (5.21) we can let $k = \ell - n$ to get

$$\begin{aligned} E^*(s - jn\omega_i) &= \frac{1}{T} \sum_{\ell=-\infty}^{\infty} E(s - j\ell\omega_i) \\ &= E^*(s). \end{aligned} \quad (5.22)$$

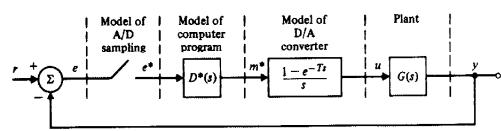
In other words, because E^* is already periodic, shifting it an integral number of periods leaves it unchanged. Substituting Eq. (5.22) into Eq. (5.20) yields

$$\begin{aligned} U^*(s) &= E^*(s) \frac{1}{T} \sum_{-\infty}^{\infty} G(s - jn\omega_i) \\ &= E^*(s)G^*(s). \quad \text{QED} \end{aligned} \quad (5.23)$$

Note especially what is *not* true. If $U(s) = E(s)G(s)$, then $U^*(s) \neq E^*(s)G^*(s)$ but rather $U^*(s) = (EG)^*(s)$. The periodic character of E^* in Eq. (5.19) is crucial.

⁵ We of course assume the existence of $U^*(s)$, which is assured if $G(s)$ tends to zero as s tends to infinity at least as fast as $1/s$. We must be careful to avoid impulse modulation of impulses, for $\delta(t)\delta(t)$ is undefined.

Figure 5.13
Block diagram of digital control as a sampled-data system



The final result we require is that, given a sampled-signal transform such as $U^*(s)$, we can find the corresponding z -transform simply by letting $e^{sT} = z$ or

$$U(z) = U^*(s) \Big|_{e^{sT}=z}. \quad (5.24)$$

There is an important time-domain reflection of Eq. (5.24). The inverse Laplace transform of $U^*(s)$ is the sequence of *impulses* with intensities given by $u(kT)$; the inverse z -transform of $U(z)$ is the sequence of values $u(kT)$. Conceptually, sequences of values and the corresponding z -transforms are easy to think about as being processed by a computer program, whereas the model of sampling as a sequence of impulses is what allows us to analyze a discrete system embedded in a continuous world (see Fig. 5.13). Of course, the impulse modulator must always be eventually followed by a low-pass circuit (hold circuit) in the physical world. Note that Eq. (5.24) can also be used in the other direction to obtain $U^*(s)$, the Laplace transform of the train of impulses, from a given $U(z)$.

◆ Example 5.1 Block Diagram Analysis

Compute the transforms of Y^* and Y for the system block diagram of Fig. 5.13.

Solution. In Fig. 5.13 we have modeled the A/D converter plus computer program plus D/A converter as an impulse modulator [which takes the samples from $e(t)$], a computer program that processes these samples described by $D^*(s)$, and a zero-order hold that constructs the piecewise-constant output of the D/A converter from the impulses of m^* . In the actual computer we assume that the samples of $e(t)$ are manipulated by a difference equation whose input-output effect is described by the z -transform $D(z)$. These operations are represented in Fig. 5.13 as if they were performed on impulses, and hence the transfer function is $D^*(s)$ according to Eq. (5.24). Finally, the manipulated impulses, $m^*(t)$, are applied to the zero-order hold from which the piecewise-constant-control signal $u(t)$ comes. In reality, of course, the computer operates on the sample values of $e(t)$ and the piecewise-constant output is generated via a storage register and a D/A converter. The impulses provide us with a convenient, consistent,

and effective model of the processes to which Laplace-transform methods can be applied. From the results given thus far, we can write relations among Laplace transforms as

$$\begin{aligned} E(s) &= R - Y, & (a) \\ M^*(s) &= E^* D^*, & (b) \\ U &= M^* \left[\frac{1 - e^{-Ts}}{s} \right], & (c) \\ Y &= GU. & (d) \end{aligned} \quad (5.25)$$

The usual idea is to relate the discrete output, Y^* , to the discrete input, R^* . Suppose we sample each of these equations by using the results of Eq. (5.19) to "star" each transform. The equations are⁶

$$\begin{aligned} E^* &= R^* - Y^*, & (a) \\ M^* &= E^* D^*, & (b) \\ U^* &= M^*, & (c) \\ Y^* &= [GU]^*. & (d) \end{aligned} \quad (5.26)$$

Now Eq. (5.26(d)) indicates that we need U , not U^* , to compute Y^* , so we must back up to substitute Eq. (5.25(c)) into Eq. (5.26(d)):

$$Y^* = \left[GM^* \left(\frac{1 - e^{-Ts}}{s} \right) \right]^*. \quad (5.27)$$

Taking out the periodic parts, which are those in which s appears only as e^{sT} [which include $M^*(s)$], we have

$$Y^* = (1 - e^{-Ts})M^* \left(\frac{G}{s} \right)^*. \quad (5.28)$$

Substituting from Eq. (5.26(b)) for M^* gives

$$Y^* = (1 - e^{-Ts})E^* D^* (G/s)^*. \quad (5.29)$$

And substituting Eq. (5.26(a)) for E^* yields

$$Y^* = (1 - e^{-Ts})D^* (G/s)^* |R^* - Y^*|. \quad (5.30)$$

If we call

$$(1 - e^{-Ts})D^* (G/s)^* = H^*, \quad (5.31)$$

then we can solve Eq. (5.30) for Y^* , obtaining

$$Y^* = \frac{H^*}{1 + H^*} R^*. \quad (5.32)$$

⁶ In sampling Eq. (5.25(c)) we obtain Eq. (5.26(c)) by use of the continuous-from-the-right convention for Eq. (5.5) for impulse modulation of discontinuous functions. From the time-domain operation of the zero-order hold, it is clear that the samples of u and m are the same, and then from this Eq. (5.26(c)) follows.

◆ Example 5.2 Analysis of a Specific Block Diagram

Apply the results of Example 1 to compute Y^* and Y for the case where

$$G(s) = \frac{a}{s+a}, \quad (5.33)$$

and the sampling period T is such that $e^{-aT} = \frac{1}{2}$. The computer program corresponds to a discrete integrator

$$u(kT) = u(kT - T) + K_0 e(kT), \quad (5.34)$$

and the computer D/A holds the output constant so that the zero-order hold is the correct model.

Solution. We wish to compute the components of H^* given in Eq. (5.31). For the computer program we have the transfer function of Eq. (5.34), which in terms of z is

$$D(z) = \frac{U(z)}{E(z)} = \frac{K_0}{1 - z^{-1}} = \frac{K_0 z}{z - 1}.$$

Using Eq. (5.24), we get the Laplace-transform form

$$D^*(s) = \frac{K_0 e^{sT}}{e^{sT} - 1}. \quad (5.35)$$

For the plant and zero-order-hold we require

$$\begin{aligned} (1 - e^{-T})G(s)/s)^* &= (1 - e^{-T})\left(\frac{a}{s(s+a)}\right)^* \\ &= (1 - e^{-T})\left(\frac{1}{s} - \frac{1}{s+a}\right)^*. \end{aligned}$$

Using Eq. (5.5), we have

$$(1 - e^{-T})G(s)/s)^* = (1 - e^{-T})\left(\frac{1}{1 - e^{-T}} - \frac{1}{1 - e^{-aT}e^{-T}}\right).$$

Because $e^{-aT} = \frac{1}{2}$, this reduces to

$$\begin{aligned} (1 - e^{-T})G(s)/s)^* &= \frac{(1/2)e^{-T}}{1 - (1/2)e^{-T}} \\ &= \frac{1/2}{e^{T/2} - 1/2}. \end{aligned} \quad (5.36)$$

Combining Eq. (5.36) and Eq. (5.35) then, in this case, we obtain

$$H^*(s) = \frac{K_0}{2} \frac{e^{sT}}{(e^{sT} - 1)(e^{sT} - 1/2)}. \quad (5.37)$$

Equation (5.37) can now be used in Eq. (5.32) to find the closed-loop transfer function from which the dynamic and static responses can be studied, as a function of K_0 , the program gain. We note also that beginning with Eq. (5.25), we can readily calculate that

$$Y(s) = R^* \frac{D^*}{1 + H^*} \frac{(1 - e^{-T})}{s} G(s). \quad (5.38)$$

Equation (5.38) shows how to compute the response of this system in between sampling instants. For a given $r(t)$, the starred terms in Eq. (5.38) and the $(1 - e^{-T})$ -term correspond to a train of impulses whose individual values can be computed by expanding in powers of e^{-T} . These impulses are applied to $G(s)/s$, which is the step response of the plant. Thus, between sampling instants, we will see segments of the plant step response.

With the exception of the odd-looking forward transfer function, Eq. (5.32) looks like the familiar feedback formula: forward-over-one-plus-feedback. Unfortunately, the sequence of equations by which Eq. (5.32) was computed was a bit haphazard, and such an effort might not always succeed. Another example will further illustrate the problem.

◆ Example 5.3 Another Block Diagram Analysis

Compute Y^* and Y for the block diagram of Fig. 5.14.

Solution. The equations describing the system are (all symbols are Laplace transforms)

$$\begin{aligned} E &= R - Y, & (a) \\ U &= H E, & (b) \\ Y &= U^* G; & (c) \end{aligned} \quad (5.39)$$

and after sampling, the equations are

$$\begin{aligned} E^* &= R^* - Y^*, & (a) \\ U^* &= (H E)^*, & (b) \\ Y^* &= U^* G^*. & (c) \end{aligned} \quad (5.40)$$

How do we solve these? In Eq. (5.40(b)) we need E , not E^* . So we must go back to Eq. (5.39(a))

$$\begin{aligned} U^* &= (H(R - Y))^* \\ &= (HR)^* - (HY)^*. \end{aligned}$$

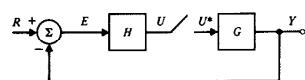
Using Eq. (5.39(c)) for Y , we have

$$U^* = (HR)^* - (HU^* G)^*.$$

Taking out the periodic U^* in the second term on the right gives

$$U^* = (HR)^* - U^*(HG)^*.$$

Figure 5.14
A simple system that does not have a transfer function



Solving, we get

$$U^* = \frac{(HR)^*}{1 + (HG)^*}. \quad (5.41)$$

From Eq. (5.40(c)), we can solve for Y^*

$$Y^* = \frac{(HR)^*}{1 + (HG)^*} G^*. \quad (5.42)$$

Equation (5.42) displays a curious fact. The transform of the input is bound up with $H(s)$ and *cannot* be divided out to give a transfer function! This system displays an important fact that with the manipulations of stars for sampling might be overlooked: A sampled-data system is *time varying*. The response depends on the time *relative to the sampling instants* at which the signal is applied. Only when the input samples *alone* are required to generate the output samples can we obtain a discrete transfer function. The time variation occurs in the taking of samples. In general, as in Fig. 5.14, the entire input signal $r(t)$ is involved in the system response, and the transfer-function concept fails. Even in the absence of a transfer function, however, the techniques developed here permit study of stability and response to specific inputs such as step, ramp, or sinusoidal signals.

We need to know the general rules of block-diagram analysis. In solving Fig. 5.14 we found ourselves working with U , the signal that was sampled. This is in fact the key to the problem. Given a block diagram with several samplers, *always select the variables at the inputs to the samplers as the unknowns*. Being sampled, these variables have periodic transforms that will always factor out after the sampling process and result in a set of equations in the sampled (starred) variables that can be solved.

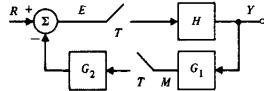
◆ Example 5.4 Another Block Diagram Analysis

Compute the transforms of Y^* and Y for the block diagram drawn in Fig. 5.15.

Solution. We select E and M as independent variables and write

$$\begin{aligned} E(s) &= R - M^* G_2, \\ M(s) &= E^* H G_1. \end{aligned} \quad (5.43)$$

Figure 5.15
A final example for transfer-function analysis of sampled-data systems



Next we sample these signals, and use the "if periodic, then out" rule from Eq. (5.19):

$$\begin{aligned} E^* &= R^* - M^* G_2^*, \\ M^* &= E^* (H G_1)^* G_1^*. \end{aligned} \quad (5.44)$$

We solve these equations by substituting for M^* in Eq. (5.44) from Eq. (5.43)

$$\begin{aligned} E^* &= R^* - E^* (H G_1)^* G_2^*, \\ &= \frac{R^*}{1 + (H G_1)^* G_2^*}. \end{aligned} \quad (5.45)$$

To obtain Y we use the equation

$$Y = E^* H \frac{R^* H}{1 + (H G_1)^* G_2^*}, \quad (5.46)$$

and

$$Y^* = \frac{R^* H^*}{1 + (H G_1)^* G_2^*}. \quad (5.47)$$

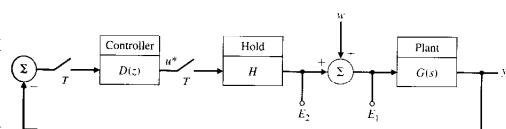
In this case we have a transfer function. Why? Because only the samples of the external input are used to cause the output. To obtain the z -transform of the samples of the output, we would let $e^{-t} = z$ in Eq. (5.47). From Eq. (5.46) we can solve for the continuous output, which consists of impulses applied to $H(s)$ in this case.

As a final example of analysis of sampled-data systems we consider a problem of experimental transfer function measurement in a sampled-data system.

◆ Example 5.5 Measuring the Transfer Function of a Sampled-Data System

It has been proposed to use an experiment to measure the loop gain of a trial sampled-data design on the actual physical system using the setup of Figure 5.16. The proposal is to have zero reference input but to inject a sinusoid into the system at W and to measure the responses

Figure 5.16
A block diagram for experimental measurement of a sampled-data transfer function



at that frequency at locations E_1 and E_2 . It is thought that the (complex) ratio of these signals will give the loop gain from which the gain and phase margins can be determined and with which a frequency response design can be worked out.

1. Compute the transforms of E_1 and E_2 for a general signal input at w .
2. Suppose that the signal w is a sinusoid of frequency ω_0 less than π/T (no aliasing). Plot the spectra of GW and $(GW)^*$ and show that $(GW)^* = \frac{1}{T}GW$ at the frequency ω_0 .
3. Use the results of 2) to get an expression for the complex ratio of the signals E_1 and E_2 when $\omega_0 < \pi/T$.
4. Repeat these calculations for the setup of Fig. 5.17 where the input signal is first sampled and held before being applied to the system.

Solution.

1. Following the procedure just given, we express the signals of interest in terms of sampled signals as follows

$$E_1 = W + U^*H \quad (5.48)$$

$$E_2 = U^*H \quad (5.49)$$

$$Y = WG + U^*HG \quad (5.50)$$

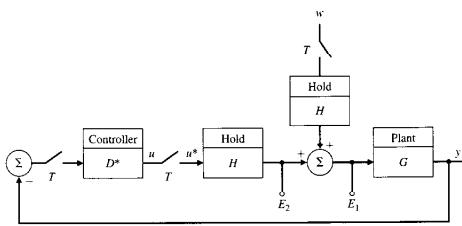
$$Y^* = (WG)^* + U^*(HG)^* \quad (5.51)$$

$$U^* = -D^*Y^* \quad (5.52)$$

Solving Eq. (5.52) for U^*

$$U^*(s) = -\frac{D^*(WG)^*}{1 + D^*(HG)^*} \quad (5.53)$$

Figure 5.17
A block diagram for experimental measurement of a sampled-data transfer function with sampled input



If we now substitute this result into Eq. (5.48) and Eq. (5.49) we have the solution of this part as

$$\begin{aligned} E_1 &= W - \frac{D^*(WG)^*}{1 + D^*(HG)^*} H \\ E_2 &= -\frac{D^*(WG)^*}{1 + D^*(HG)^*} H \end{aligned} \quad (5.54)$$

Clearly we do not have a transfer function since the transform of the signal is imbedded in the signal transforms.

2. For the second part, we can consider the sinusoid one exponential at a time and consider $w = 2\pi\delta(\omega - \omega_0)$. Then

$$(GW)^* = \frac{1}{T} \sum_{k=-\infty}^{t=\infty} G(j\omega - jk\frac{2\pi}{T}) 2\pi\delta(\omega - \omega_0 - \frac{2\pi k}{T}).$$

The spectra involved are easily sketched. Since $\omega_0 < \pi/T$ there is no overlap and at ω_0 the signal is

$$\begin{aligned} (GW)^* &= \frac{1}{T} G(j\omega_0) 2\pi\delta(\omega - \omega_0) \\ &= \frac{1}{T} GW|_{\omega_0}. \end{aligned} \quad (5.55)$$

3. If we substitute Eq. (5.55) into Eq. (5.54) and take the ratio, we find the describing function

$$\frac{E_2}{E_1} = -\frac{\frac{1}{T} D^*GH}{1 + D^*(GH)^* - \frac{1}{T} D^*GH}. \quad (5.56)$$

Notice that if $|G| = 0$ for $|\omega| > \pi/T$ so that $G^* = G$ for frequencies less than π/T , then Eq. (5.56) reduces to

$$\frac{E_2}{E_1} = -D^*(GH)^*.$$

which is the transfer function. Thus the proposed method works well if there is a good antialias filter in the loop.

4. With the input applied through a sample and hold as drawn in Fig. 5.17 the key expressions are given by

$$\begin{aligned} E_1 &= U^*H + W^*H \\ E_2 &= U^*H \\ U^* &= -D^*Y^* \\ Y &= U^*HG + W^*HG. \end{aligned} \quad (5.57)$$

These equations can be readily solved, after taking the "star" of Y to give

$$\begin{aligned} E_1 &= \frac{W^*H}{1 + D^*(HG)^*} \\ E_2 &= -\frac{D^*(HG)^*}{1 + D^*(HG)^*} W^*H. \end{aligned}$$

From these, the ratio gives the true discrete transfer function

$$\frac{E_2}{E_1} = -D^*(HG)^*.$$

5.5 Calculating the System Output Between Samples: The Ripple

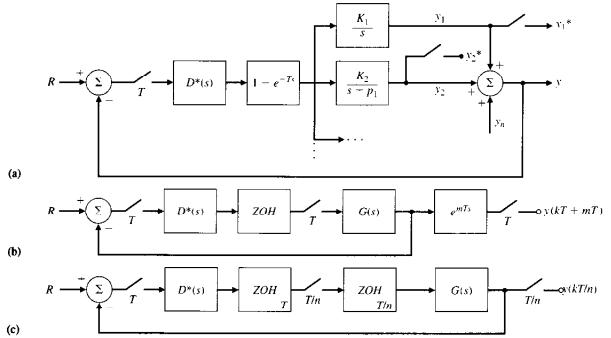
In response to a particular input, the output of a sampled-data system at sampling instants can be computed by the z -transform, even in those cases where there is no transfer function. However, in many problems it is important to examine the response between sampling instants, a response that is called the "ripple" in the response. Often, for example, the maximum overshoot will not occur at a sampling instant but at some intermediate point. In other cases, hidden oscillations are present, oscillations that may or may not decay with time. The ripple is generated by the continuous-time part of the system at the output. For example, in the case drawn in Fig. 5.13, the ripple is given by the response of $G(s)/s$ between sampling instants. Three techniques have been suggested to compute ripple. The first, suggested by J. Sklansky, is based on the partial-fraction expansion of $G(s)/s$. The second, suggested by E. Jury, is based on introducing a time shift in the sampler at the output of the system. If this shift is less than a sample period, the new samples are taken between the system samples. The modified transform from input samples to shifted samples is called the *modified z-transform* of $G(s)/s$. The third technique, introduced by G. Kranc, is based on sampling the output at a faster rate than the feedback loop is updated. Block diagrams representing the three methods are given in Fig. 5.18.

In the partial-fraction method, shown in Fig. 5.18(a), the outputs of the several fractions are sampled and the values of $y_1(kT)$, $y_2(kT)$, ... are computed in the regular way with z -transforms or MATLAB statements. These values at the instant kT represent initial conditions for the several partial fraction continuous dynamics at time kT and from them the transient over the period from kT to $(k+1)T$ can be computed. The total system output is the sum of these components. The method is somewhat tedious but gives an exact expression for the ripple during any given sample period from which, for example, the peak overshoot can be exactly computed.

The modified z -transform of the plant with zero-order hold is defined as

$$G(z, m) = (1 - z^{-1}) \mathcal{Z} \left\{ \frac{G(s)}{s} e^{mT_s} \right\} \quad 0 \leq m < 1,$$

Figure 5.18
Three methods used to evaluate ripple. (a) Partial fraction expansion; (b) Modified z -transform; (c) Multirate sampling



and represents samples taken at the times $kT + mT$. The modified transform of the output of the system shown in Fig. 5.18(b) is given by

$$Y(z, m) = \frac{D(z)G(z, m)}{1 + D(z)G(z)} R(z), \quad (5.58)$$

and its inverse will give samples at $kT + mT$. The modified operation is noncausal but is only being used as a computational device to obtain inter-sample ripple. For example, if $m = 0.5$ then use of Eq. (5.58) will find sample values halfway between sample updates of the control. MATLAB only permits delays (causal models) and can be used to find the output of the modified plant delayed by one sample shown in the figure as $z^{-1}Y(z, m)$. If the plant is given in state form with description matrices $[F, G, H, J]$, then the representation of the delayed modified transform can be computed in MATLAB using

SYS = ss(F, G, H, J).

The delay for sample period T and shift m is set by the command

set(SYS, 'td', (1 - m)T).

and finally, the discrete representation of the system which has a delayed modified $z - transform$ is given by the standard conversion

$$SYS = c2d(SYS, T).$$

The method of multi-rate sampling is shown in Fig. 5.18(c). The output of the controller is held for the full T seconds but this signal is again sampled at the rate T/n for some small n , such as 5. The plant output is also sampled at the rate T/n . The feedback loop is unchanged by these additional samplers but the output ripple is now available at n points in between the regular sample times. This technique is readily programmed in MATLAB and is regularly used in this book to compute the ripple. An interesting case is given in Fig. 7.14 where it can be seen that the maximum overshoot occurs in the ripple.

5.6 Summary

In this chapter we have considered the analysis of mixed systems that are partly discrete and partly continuous, taking the continuous point of view. These systems arise from digital control systems that include A/D and D/A converters. The important points of the chapter are

- The large-signal behavior of an A/D converter can be modeled as a linear impulse modulator followed by a zero-order-hold.
- D/A converter can be modeled as a zero-order-hold.
- The transform of a sampled signal is periodic with period $2\pi/T$ for sample period T .
- Sampling introduces aliasing, which may be interpreted in both the frequency and the time domains.
- The sampling theorem shows how a band-limited signal can be reconstructed from its samples.
- Interconnections of systems that include sampling can be analyzed by block-diagram analysis.
- If the input signal to a sampled data system is not sampled, it is impossible to define a transfer function.
- The output of a sampled-data system between sampling instants can be computed using partial fraction expansion, using the modified $z - transform$, or by multi-rate sampling. With a computer, multi-rate sampling is the most practical method.

5.7 Problems

- 5.1 Sketch a signal that represents bounded hidden oscillations.

5.2 Show how to construct a signal of hidden oscillations that grows in an unstable fashion. Where in the s -plane are the poles of the transforms of your signal(s)?

5.3 A first-order hold is a device that extrapolates a line over the interval from kT to $(k+1)T$ with slope given by the samples $r(kT-T)$ and $r(kT)$ starting from $r(kT)$ as shown in Fig. 5.19. Compute the frequency response of the first-order hold.

5.4 Consider the circuit of Fig. 5.20. By plotting the response to a signal that is zero for all sample instants except $t=0$ and that is 1.0 at $t=0$, show that this circuit implements a first-order hold.

5.5 Sketch the step response $y(t)$ of the system shown in Fig. 5.21 for $K = \frac{1}{2}, 1$, and 2 .

5.6 Sketch the response of a second-order hold circuit to a step unit. What might be the major disadvantage of this data extrapolator?

5.7 A triangle hold is a device that has an output, as sketched in Fig. 5.22 that connects the samples of an input with straight lines.

- (a) Sketch the impulse response of the triangle hold. Notice that it is noncausal.
- (b) Compute the transfer function of the hold.
- (c) Use MATLAB to plot the frequency response of the triangle hold

Figure 5.19
Impulse response of a first-order hold filter

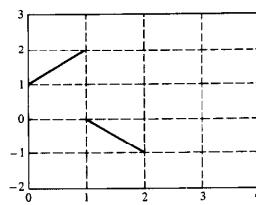


Figure 5.20
Block diagram of a sample and first-order hold

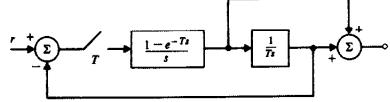


Figure 5.21
A sampled-data system

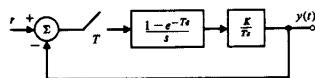
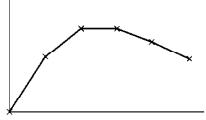


Figure 5.22
Response of a sample and triangle hold



(d) How would the frequency response be changed if the triangle hold is made to be causal by adding a delay of one sample period?

5.8 Sketch the output of a sample and zero-order hold to

- (a) A step input.
- (b) A ramp input.
- (c) A sinusoid of frequency $\omega_0/10$.

5.9 Sketch the output of a sample and first-order hold to

- (a) A step input.
- (b) A ramp input.
- (c) A sinusoid of frequency $\omega_0/10$.

5.10 Sketch the output of a sample and triangle hold to

- (a) A step input.
- (b) A ramp input.
- (c) A sinusoid of frequency $\omega_0/10$.

5.11 Sketch the output of a sample and causal triangle hold to

- (a) A step input.
- (b) A ramp input.
- (c) A sinusoid of frequency $\omega_0/10$.

5.12 A sinusoid of frequency 11 rad/sec . is sampled at the frequency $\omega_s = 5 \text{ rad/sec}$.

- (a) Indicate the component frequencies up to $\omega = 20 \text{ rad/sec}$.
- (b) Indicate the relative amplitudes of the components up to 20 rad/sec . if the sampler is followed by a zero-hold.

5.13 A signal $r(t) = \sin(2t) + \sin(13t)$ is sampled at the frequency $\omega_s = 16$.

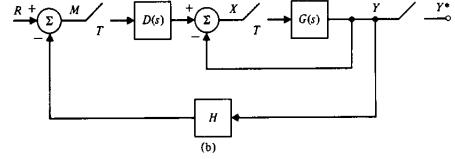
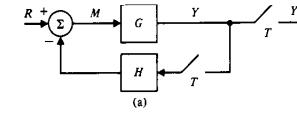
- (a) Indicate the frequency of the components in the sampled signal up to $\omega = 32$.
- (b) Indicate the relative amplitudes of the signals in the output if the signal is passed through the anti-aliasing filter with transfer function $\frac{1}{(z-0.5)^2}$ before sampling. You can use MATLAB to compute the filter gain.

5.14 Derive Eq. (5.38).

5.15 Find the transform of the output $Y(s)$ and its samples $Y^*(s)$ for the block diagrams shown in Fig. 5.23. Indicate whether a transfer function exists in each case.

5.16 Assume the following transfer functions are preceded by a sampler and zero-order hold and followed by a sampler. Compute the resulting discrete transfer functions.

Figure 5.23
Block diagrams of sampled data systems.
(a) Single loop;
(b) multiple loop



- (a) $G_1(s) = 1/s^2$
- (b) $G_2(s) = e^{-1.5s}/(s+1)$
- (c) $G_3(s) = 1/s(s+1)$
- (d) $G_4(s) = e^{-1.5s}/s(s+1)$
- (e) $G_5(s) = 1/(s^2+1)$

5.17 One technique for examining the response of a sampled data system between sampling instants is to shift the response a fraction of a period to the left and to sample the result. The effect is as shown in the block diagram of Fig. 5.24 and described by the equation

$$Y^*(s, m) = R^*(s)(G(s)e^{mT})^*$$

As a function of z , the equivalent equation is

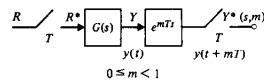
$$Y(z) = R(z)G(z, m).$$

The function $G(z, m)$ is called the modified z -transform of $G(s)$. In the figure, let

$$G(z) = \frac{1}{(z+1)}, \quad R(z) = \frac{1}{s}, \quad \text{and} \quad T = 1.$$

(a) Compute $y(t)$ by calculating $y(kT)$ from the ordinary z -transform and observing that between samples, the output $y(t)$ is an exponential decay with

Figure 5.24
Block diagrams showing the modified z -transform



unit time constant. Sketch the response for five sample intervals. Notice that this technique is the essence of the partial-fraction method of obtaining the ripple.

- (b) Compute the modified z -transform for $m = \frac{1}{2}$ and compute the samples according to the equation for $Y(z, m)$. Plot these on the same plot as that of $y(t)$ and verify that you have found the values at the mid-points of the sampling pattern.

5.8 Appendix

To compute the transform of a sinusoid, we consider first the Fourier transform of $v(t) = e^{j\omega_0 t - j\phi_0}$. For this we have

$$V(j\omega) = \int_{-\infty}^{\infty} e^{j(\omega_0 t - j\phi_0)} e^{-j\omega t} dt. \quad (5.59)$$

This integral does not converge in any obvious way, but we can approach it from the back door, as it were. Consider again the impulse, $\delta(t)$. The direct transform of this object is easy, considering the sifting property, as follows

$$\int_{-\infty}^{\infty} \delta(t) e^{-j\omega t} dt = 1.$$

Now the general form of the inverse Fourier transform is given by

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(j\omega) e^{j\omega t} d\omega.$$

If we apply the inverse transform integral to the impulse and its transform, we take $f(t) = \delta(t)$ and $F(j\omega) = 1$ with the result

$$\delta(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{j\omega t} d\omega.$$

However, except for notation and a simple change of variables, this is exactly the integral we needed to evaluate the spectrum of the single exponential. If we exchange t with ω the integral reads

$$\delta(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{j\omega t} dt.$$

Eq. (5.59) is of this form

$$\begin{aligned} V(j\omega) &= \int_{-\infty}^{\infty} e^{j(\omega_0 t - j\phi_0)} e^{-j\omega t} dt \\ &= e^{j\phi_0} \int_{-\infty}^{\infty} e^{j(\omega_0 - \omega)t} dt \\ &= 2\pi e^{j\phi_0} \delta(\omega - \omega_0). \end{aligned}$$

At the last step in this development, the sign of the argument in the delta function was changed, which is legal because $\delta(t)$ is an even function and $\delta(t) = \delta(-t)$. The argument is more natural as $(\omega - \omega_0)$ rather than the opposite.

• 6 •

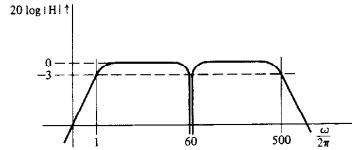
Discrete Equivalents

A Perspective on Computing Discrete Equivalents

One of the exciting fields of application of digital systems¹ is in signal processing and digital filtering. A filter is a device designed to pass desirable signal components and to reject undesirable ones: in signal processing it is common to represent signals as a sum of sinusoids and to define the "desirable components" as those signals whose frequencies are in a specified band. Thus a radio receiver is designed to pass the band of frequencies transmitted by the station we want to hear and reject all others. We would call a filter which does this a **bandpass filter**. In electrocardiography it often happens that power-line frequency signals are strong and unwanted, so a filter is designed to pass signals between 1 and 500 Hz but to eliminate those at 60 Hz. The magnitude of the transfer function for this purpose may look like Fig. 6.1 on a log-frequency scale, where the amplitude response between 59.5 and 60.5 Hz might reach 10^{-3} . Here we have a band-reject filter with a 60-dB rejection ratio in a 1-Hz band centered at 60 Hz.

In long-distance telephony some filters play a conceptually different role. There the issue is that ideal transmission requires that all desired frequencies be

Figure 6.1
Magnitude of a low-frequency bandpass filter with a narrow rejection band



¹ Including microprocessors and special-purpose devices for digital signal processing, called DSP chips.

treated equally but transmission media—wires or microwaves—introduce distortion in the amplitude and phase of the sinusoids that comprise the desired signal and this distortion must be removed. Filters to correct the distortion are called **equalizers**. Finally, the dynamic response of control systems requires modification in order for the complete system to have satisfactory dynamic response. We call the devices that make these changes **compensators**.

Whatever the name—filter, equalizer, or compensator—many fields have used for linear dynamic systems having a transfer function with specified characteristics of amplitude and phase. Increasingly the power and flexibility of digital processors makes it attractive to perform these functions by digital means. The design of continuous electronic filters is a well-established subject that includes not only very sophisticated techniques but also well-tested computer programs to carry out the designs [Van Valkenburg (1982)]. Consequently, an important approach to digital filter design is to start with a good analog design and construct a filter having a discrete frequency response that approximates that of the satisfactory design. For digital control systems we have much the same motivation: Continuous-control designs are well established and one can take advantage of a good continuous design by finding a discrete equivalent to the continuous compensator. This method of design is called **emulation**. Although much of our presentation in this book is oriented toward direct digital design and away from emulation of continuous designs with digital equivalents, it is important to understand the techniques of discrete equivalents both for purposes of comparison and because it is widely used by practicing engineers.

Chapter Overview

The specific problem of this chapter is to find a discrete transfer function that will have approximately the same characteristics over the frequency range of importance as a given transfer function, $H(s)$. Three approaches to this task are presented. The first method is based on *numerical integration* of the differential equations that describe the given design. While there are many techniques for numerical integration, only simple formulas based on rectangular and trapezoid rules are presented. The second approach is based on comparisons of the s and z domains. Note that the natural response of a continuous filter with a pole at some point $s = s_p$ will, when sampled with period T , represent the response of a discrete filter with a pole at $z = e^{s_p T}$. This formula can be used to map the poles and zeros of the given design into poles and zeros of an approximating discrete filter. This is called *pole and zero mapping*. The third and final approach is based on taking the samples of the input signal, extrapolating between samples to form an approximation to the signal, and passing this approximation through the given filter transfer function. This technique is called *hold equivalence*. The methods are compared with respect to the quality of the approximation in the frequency domain as well as the ease of computation of the designs.

6.1 Design of Discrete Equivalents via Numerical Integration

The topic of numerical integration of differential equations is quite complex, and only the most elementary techniques are presented here. For example, we only consider formulas of low complexity and fixed step-size. The fundamental concept is to represent the given filter transfer function $H(s)$ as a differential equation and to derive a difference equation whose solution is an approximation of the differential equation. For example, the system

$$\frac{U(s)}{E(s)} = H(s) = \frac{a}{s + a} \quad (6.1)$$

is equivalent to the differential equation

$$\dot{u} + au = ae. \quad (6.2)$$

Now, if we write Eq. (6.2) in integral form, we have a development much like that described in Chapter 4, except that the integral is more complex here

$$\begin{aligned} u(t) &= \int_0^t [-au(\tau) + ae(\tau)] d\tau. \\ u(kT) &= \int_0^{kT-T} [-au + ae] d\tau + \int_{T-t}^{kT} [-au + ae] d\tau \\ &= u(kT - T) + \left\{ \begin{array}{l} \text{area of } -au + ae \\ \text{over } kT - T \leq \tau < kT. \end{array} \right. \end{aligned} \quad (6.3)$$

Many rules have been developed based on how the incremental area term is approximated. Three possibilities are sketched in Fig. 6.2. The first approximation leads to the **forward rectangular rule**² wherein we approximate the area by the rectangle looking forward from $kT - T$ and take the amplitude of the rectangle to be the value of the integrand at $kT - T$. The width of the rectangle is T . The result is an equation in the first approximation, u_1 ,

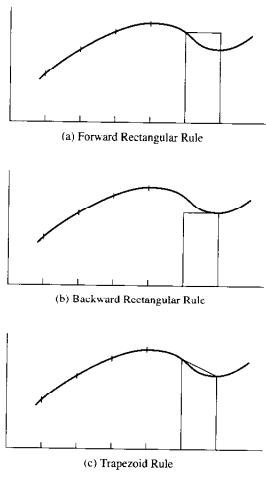
$$\begin{aligned} u_1(kT) &= u_1(kT - T) + T[-au_1(kT - T) + ae(kT - T)] \\ &= (1 - aT)u_1(kT - T) + aTe(kT - T). \end{aligned} \quad (6.4)$$

The transfer function corresponding to the forward rectangular rule in this case is

$$\begin{aligned} H_F(z) &= \frac{aT z^{-1}}{1 - (1 - aT)z^{-1}} \\ &= \frac{a}{(z - 1)/T + a} \quad (\text{forward rectangular rule}). \end{aligned} \quad (6.5)$$

² Also known as *Euler's rule*.

Figure 6.2
Sketches of three ways the area under the curve from kT to $kT + T$ can be approximated:
(a) forward rectangular rule,
(b) backward rectangular rule,
(c) trapezoid rule



A second rule follows from taking the amplitude of the approximating rectangle to be the value looking *backward* from kT toward $kT - T$, namely, $-au(kT) + ae(kT)$. The equation for u_2 , the second approximation,³ is

$$\begin{aligned} u_2(kT) &= u_1(kT - T) + T[-au_1(kT) + ae(kT)] \\ &= \frac{u_1(kT - T)}{1 + aT} + \frac{aT}{1 + aT}e(kT). \end{aligned} \quad (6.6)$$

³ It is worth noting that in order to solve for Eq. (6.6) we had to eliminate $u(kT)$ from the right-hand side where it entered from the integrand. Had Eq. (6.2) been nonlinear, the result would have been an implicit equation requiring an iterative solution. This topic is the subject of predictor-corrector rules, which are beyond our scope of interest. A discussion is found in most books on numerical analysis. See Golub and Van Loan (1983).

Again we take the z -transform and compute the transfer function of the **backward rectangular rule**

$$\begin{aligned} H_B(z) &= \frac{aT}{1 + aT} \frac{1}{1 - z^{-1}/(1 + aT)} = \frac{aTz}{z(1 + aT) - 1} \\ &= \frac{a}{(z - 1)/Tz + a} \quad (\text{backward rectangular rule}). \end{aligned} \quad (6.7)$$

Our final version of integration rules is the **trapezoid rule** found by taking the area approximated in Eq. (6.3) to be that of the trapezoid formed by the average of the previously selected rectangles. The approximating difference equation is

$$\begin{aligned} u_3(kT) &= u_3(kT - T) + \frac{T}{2}[-au_3(kT - T) \\ &\quad + ae(kT - T) - au_3(kT) + ae(kT)] \\ &= \frac{1 - (aT/2)}{1 + (aT/2)}u_3(kT - T) + \frac{aT/2}{1 + (aT/2)}[e_3(kT - T) + e_3(kT)]. \end{aligned} \quad (6.8)$$

The corresponding transfer function from the trapezoid rule is

$$\begin{aligned} H_T(z) &= \frac{aT(z + 1)}{(2 + aT)z + aT - 2} \\ &= \frac{a}{(2/T)((z - 1)/(z + 1)) + a} \quad (\text{trapezoid rule}). \end{aligned} \quad (6.9)$$

Suppose we tabulate our results obtained thus far.

$H(s)$	Method	Transfer function
$\frac{a}{s + a}$	Forward rule	$H_F = \frac{a}{(z - 1)/T + a}$
$\frac{a}{s + a}$	Backward rule	$H_B = \frac{a}{(z - 1)/Tz + a}$
$\frac{a}{s + a}$	Trapezoid rule	$H = \frac{a}{(2/T)((z - 1)/(z + 1)) + a}$

From direct comparison of $H(s)$ with the three approximations in this tabulation, we can see that the effect of each of our methods is to present a discrete transfer function that can be obtained from the given Laplace transfer function

$H(s)$ by substitution of an approximation for the frequency variable as shown below

Method	Approximation
Forward rule	$s \leftarrow \frac{z-1}{T}$
Backward Rule	$s \leftarrow \frac{z-1}{Tz}$
Trapezoid Rule	$s \leftarrow \frac{2z-1}{Tz+1}$

The trapezoid-rule substitution is also known, especially in digital and sampled-data control circles, as **Tustin's method** [Tustin (1947)] after the British engineer whose work on nonlinear circuits stimulated a great deal of interest in this approach. The transformation is also called the **bilinear transformation** from consideration of its mathematical form. The design method can be summarized by stating the rule: Given a continuous transfer function (filter), $H(s)$, a discrete equivalent can be found by the substitution

$$H_T(z) = H(s)|_{s=\frac{z-1}{T}}. \quad (6.12)$$

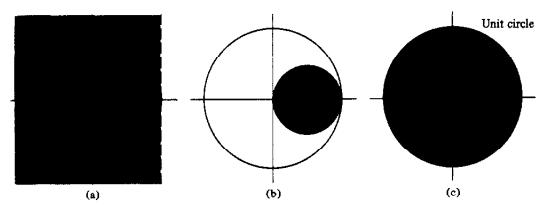
Each of the approximations given in Eq. (6.11) can be viewed as a map from the s -plane to the z -plane. A further understanding of the maps can be obtained by considering them graphically. For example, because the $(s = j\omega)$ -axis is the boundary between poles of stable systems and poles of unstable systems, it would be interesting to know how the $j\omega$ -axis is mapped by the three rules and where the left (stable) half of the s -plane appears in the z -plane. For this purpose we must solve the relations in Eq. (6.11) for z in terms of s . We find

- i) $z = 1 + Ts$, (forward rectangular rule).
- ii) $z = \frac{1}{1 - Ts}$, (backward rectangular rule).
- iii) $z = \frac{1 + Ts/2}{1 - Ts/2}$ (bilinear rule).

If we let $s = j\omega$ in these equations, we obtain the boundaries of the regions in the z -plane which originate from the stable portion of the s -plane. The shaded areas sketched in the z -plane in Fig. 6.3 are these stable regions for each case. To show that rule (ii) results in a circle, $\frac{1}{2}$ is added to and subtracted from the right-hand side to yield

$$\begin{aligned} z &= \frac{1}{2} + \left\{ \frac{1}{1 - Ts} - \frac{1}{2} \right\} \\ &= \frac{1}{2} - \frac{1 + Ts}{2(1 - Ts)}. \end{aligned} \quad (6.14)$$

Figure 6.3
Maps of the left-half of the s -plane by the integration rules of Eq. (6.10) into the z -plane. Stable s -plane poles map into the shaded regions in the z -plane. The unit circle is shown for reference. (a) Forward rectangular rule. (b) Backward rectangular rule. (c) Trapezoid or bilinear rule



Now it is easy to see that with $s = j\omega$, the magnitude of $z - \frac{1}{2}$ is constant

$$|z - \frac{1}{2}| = \frac{1}{2},$$

and the curve is thus a circle as drawn in Fig. 6.3(b). Because the unit circle is the stability boundary in the z -plane, it is apparent from Fig. 6.3 that the forward rectangular rule could cause a stable continuous filter to be mapped into an unstable digital filter.

It is especially interesting to notice that the bilinear rule maps the stable region of the s -plane exactly into the stable region of the z -plane although the entire $j\omega$ -axis of the s -plane is compressed into the 2π -length of the unit circle! Obviously a great deal of distortion takes place in the mapping in spite of the congruence of the stability regions. As our final rule deriving from numerical integration ideas, we discuss a formula that extends Tustin's rule one step in an attempt to correct for the inevitable distortion of real frequencies mapped by the rule. We begin with our elementary transfer function Eq. (6.1) and consider the bilinear rule approximation

$$H_T(z) = \frac{a}{(2/T)[(z-1)/(z+1)] + a}.$$

The original $H(s)$ had a pole at $s = -a$, and for real frequencies, $s = j\omega$, the magnitude of $H(j\omega)$ is given by

$$\begin{aligned} |H(j\omega)|^2 &= \frac{a^2}{\omega^2 + a^2} \\ &= \frac{1}{\omega^2/a^2 + 1}. \end{aligned}$$

Thus our reference filter has a half power point, $|H|^2 = \frac{1}{2}$, at $\omega = a$. It will be interesting to know where $H_r(z)$ has a half-power point.

As we saw in Chapter 4, signals with poles on the imaginary axis in the s -plane (sinusoids) map into signals on the unit circle of the z -plane. A sinusoid of frequency ω_1 corresponds to $z_1 = e^{j\omega_1 T}$, and the response of $H_r(z)$ to a sinusoid of frequency ω_1 is $H_r(z_1)$. We consider now Eq. (6.8) for $H_r(z_1)$ and manipulate it into a more convenient form for our present purposes

$$\begin{aligned} H_r(z_1) &= a / \left(\frac{2 e^{j\omega_1 T} - 1}{T e^{j\omega_1 T} + 1} + a \right) \\ &= a / \left(\frac{2 e^{j\omega_1 T/2} - e^{-j\omega_1 T/2}}{T e^{j\omega_1 T/2} + e^{-j\omega_1 T/2}} + a \right) \\ &= a / \left(\frac{2}{T} j \tan \frac{\omega_1 T}{2} + a \right). \end{aligned} \quad (6.15)$$

The magnitude squared of H_r will be $\frac{1}{2}$ when

$$\frac{2}{T} \tan \frac{\omega_1 T}{2} = a$$

or

$$\tan \frac{\omega_1 T}{2} = \frac{a T}{2}. \quad (6.16)$$

Equation (6.16) is a measure of the frequency distortion or warping caused by Tustin's rule. Whereas we wanted to have a half-power point at $\omega = a$, we realized a half-power point at $\omega_1 = (2/T) \tan^{-1}(aT/2)$, ω_1 will be approximately correct only if $aT/2 \ll 1$ so that $\tan^{-1}(aT/2) \cong aT/2$, that is, if $\omega_1 (= 2\pi/T) \gg a$ and the sample rate is much faster than the half-power frequency. We can turn our intentions around and suppose that we really want the half-power point to be at ω_1 . Equation (6.16) can be made into an equation of prewarping: If we select a according to Eq. (6.16), then, using Tustin's bilinear rule for the design, the half-power point will be at ω_1 . A statement of a complete set of rules for filter design via bilinear transformation with prewarping is

1. Write the desired filter characteristic with transform variable s and critical frequency ω_1 in the form $H(s/\omega_1)$.⁴
2. Replace ω_1 by a such that

$$a = \frac{2}{T} \tan \frac{\omega_1 T}{2},$$

⁴ The critical frequency need not be the band edge. We can use the band center of a bandpass filter or the crossover frequency of a Bode plot compensator. However, we must have $\omega_1 < \pi/T$ if a stable filter is to remain stable after warping.

and in place of $H(s/\omega_1)$, consider the prewarped function $H(s/a)$. For more complicated shapes, such as bandpass filters, the specification frequencies, such as band edges and center frequency, should be prepared before the continuous design is done; and then the bilinear transformation will bring all these points to their correct frequencies in the digital filter.

3. Substitute

$$s = \frac{2}{T} \frac{z-1}{z+1}$$

in $H(s/a)$ to obtain the prewarped equivalent $H_p(z)$.

As a frequency substitution the result can be expressed as

$$H_p(z) = H\left(\frac{s}{\omega_1}\right) \Big|_{\frac{s}{\omega_1} = \frac{1}{\tan(\omega_1 T/2)} \frac{z-1}{z+1}}. \quad (6.17)$$

It is clear from Eq. (6.17) that when $\omega = \omega_1$, $H_p(z_1) = H(j1)$ and the discrete filter has exactly the same transmission at ω_1 as the continuous filter has at this frequency. This is the consequence of prewarping. We also note that as the sampling period gets small, $H_p(z)$ approaches $H(j\omega/\omega_1)$.

◆ Example 6.1 Computing a Discrete Equivalent

The transfer function of a third order low-pass Butterworth filter⁵ designed to have unity passband ($\omega_p = 1$) is

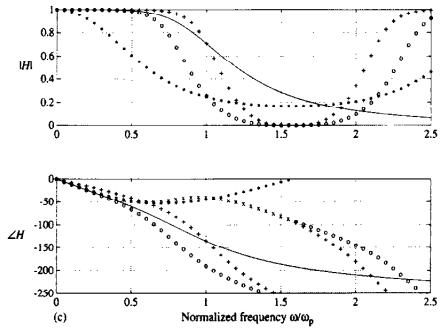
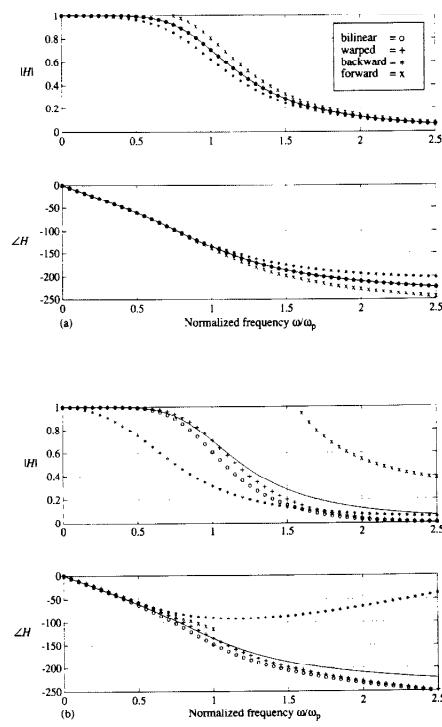
$$H(s) = \frac{1}{s^3 + 2s^2 + 2s + 1}.$$

A simple frequency scaling would of course translate the design to have any desired passband frequency. Compute the discrete equivalents and plot the frequency responses using the forward rectangular rule, the backward rectangular rule, the Tustin bilinear rule and the bilinear rule with prewarping at $\omega = 1$. Use sampling periods $T = 0.1$, $T = 1$, and $T = 2$.

Solution. Computation of the discrete equivalents is numerically tedious and the state-space algorithms described below were used in MATLAB to generate the transfer functions and the response curves plotted in Fig. 6.4. Fig. 6.4(a) shows that at a high sample rate ($T = 0.1$), where the ratio of sampling frequency to passband frequency is $\omega_s/\omega_p \approx 63$, all the rules do reasonably well but the rectangular rules are already showing some deviation. From Fig. 6.4(b) we see that at $\omega_s/\omega_p = 2\pi$ the rectangular rules are useless (the forward rule is unstable). Finally, in Fig. 6.4(c) at very slow sampling frequency with $\omega_s/\omega_p = \pi$ corresponding to a sampling period of $T = 2$ sec, only with prewarping do we have a design that comes even closer to the continuous response. In each case at the Nyquist frequency, $\omega = \pi/T$, the magnitude

⁵ A description of the properties of Butterworth filters is given in most books on filter design and briefly in Franklin, Powell and Emami-Naeini (1986).

Figure 6.4
 (a) Response of third-order lowpass filter and digital equivalents for $\omega_s/\omega_p = 20\pi$.
 (b) Response of third-order lowpass filter and digital equivalents for $\omega_s/\omega_p = 2\pi$.
 (c) Response of third-order lowpass filter and digital equivalents for $\omega_s/\omega_p = \pi$.



responses of the discrete filters start to repeat according to the periodic nature of discrete-transfer-function frequency responses. It can be seen that the magnitude and phase of the prewarped designs match those of the continuous filter exactly at the band edge, $\omega = 1$, for all these cases. This is no surprise, because such matching was the whole idea of prewarping.

The formulas for discrete equivalents are particularly simple and convenient when expressed in state-variable form and used with a computer-aided design package. For example, suppose we have a vector-matrix description of a continuous design in the form of the equations

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}e, \\ u &= \mathbf{C}\mathbf{x} + \mathbf{D}e. \end{aligned} \quad (6.18)$$

The Laplace transform of this equation is

$$\begin{aligned} s\mathbf{X} &= \mathbf{AX} + \mathbf{BE}, \\ U &= \mathbf{CX} + \mathbf{DE}. \end{aligned} \quad (6.19)$$

We can now substitute for s in Eq. (6.19) any of the forms in z corresponding to an integration rule. For example, the forward rectangular rule is to replace s with $(z - 1)/T$ from Eq. (6.11)

$$\begin{aligned} \frac{z-1}{T}\mathbf{X} &= \mathbf{AX} + \mathbf{BE}, \\ U &= \mathbf{CX} + \mathbf{DE}. \end{aligned} \quad (6.20)$$

In the time domain, the operator z corresponds to forward shift; that is, $zx(k) = x(k+1)$. Thus the corresponding discrete equations in the time domain are

$$\begin{aligned} \mathbf{x}(k+1) - \mathbf{x}(k) &= T\mathbf{A}\mathbf{x}(k) + T\mathbf{B}e(k), \\ \mathbf{x}(k+1) &= (\mathbf{I} + T\mathbf{A})\mathbf{x}(k) + T\mathbf{B}e(k), \\ u &= \mathbf{C}\mathbf{x} + \mathbf{D}e. \end{aligned} \quad (6.21)$$

Equation (6.21) is a state-space formula for the forward rule equivalent.

For the backward rule, substitute $s \leftarrow (z-1)/zT$ with the result

$$\frac{z-1}{Tz}\mathbf{X} = \mathbf{AX} + \mathbf{BE},$$

which corresponds to the time domain equations

$$\mathbf{x}(k+1) - \mathbf{x}(k) = T\mathbf{A}\mathbf{x}(k+1) + T\mathbf{B}e(k+1). \quad (6.22)$$

In this equation, there are terms in $k+1$ on both the right- and left-hand sides. In order to get an equation with such terms only on the left, transpose all $k+1$ terms to the left and define them as a new state vector

$$\begin{aligned} \mathbf{x}(k+1) - T\mathbf{A}\mathbf{x}(k+1) - T\mathbf{B}e(k+1) &= \mathbf{x}(k) \\ &\stackrel{\circ}{=} \mathbf{w}(k+1). \end{aligned} \quad (6.23)$$

From this equation, solving for \mathbf{x} in terms of \mathbf{w} and e

$$\begin{aligned} (\mathbf{I} - T\mathbf{A})\mathbf{x} &= \mathbf{w} + T\mathbf{B}e \\ \mathbf{x} - (\mathbf{I} - T\mathbf{A})^{-1}\mathbf{w} + (\mathbf{I} - T\mathbf{A})^{-1}\mathbf{B}Te. \end{aligned} \quad (6.24)$$

With this expression for \mathbf{x} , Eq. (6.23) can be put in standard form as

$$\mathbf{w}(k+1) = (\mathbf{I} - T\mathbf{A})^{-1}\mathbf{w}(k) + (\mathbf{I} - T\mathbf{A})^{-1}\mathbf{B}Te(k), \quad (6.25)$$

and the output equation is now

$$u(k) = \mathbf{C}(\mathbf{I} - T\mathbf{A})^{-1}\mathbf{w} + [\mathbf{D} + \mathbf{C}(\mathbf{I} - T\mathbf{A})^{-1}\mathbf{B}T]\mathbf{e}. \quad (6.26)$$

Equation (6.25) plus Eq. (6.26) are a state-space description of the backward rule equivalent to Eq. (6.18).

Finally, for the trapezoid or bilinear rule, the z -transform equivalent is obtained from

$$\begin{aligned} \frac{2(z-1)}{T(z+1)}\mathbf{X} &= \mathbf{AX} + \mathbf{BE} \\ (z-1)\mathbf{X} &= \frac{AT}{2}(z+1)\mathbf{X} + \frac{BT}{2}(z+1)\mathbf{E} \\ U &= \mathbf{CX} + \mathbf{DE}. \end{aligned} \quad (6.27)$$

and the time domain equation for the state is

$$\mathbf{x}(k+1) - \mathbf{x}(k) = \frac{AT}{2}(\mathbf{x}(k+1) + \mathbf{x}(k)) + \frac{BT}{2}(e(k+1) + e(k)). \quad (6.28)$$

Once more, collect all the $k+1$ terms onto the left and define these as $\mathbf{w}(k+1)$ as follows⁶

$$\begin{aligned} \mathbf{x}(k+1) - \frac{AT}{2}\mathbf{x}(k+1) - \frac{BT}{2}e(k+1) &= \mathbf{x}(k) + \frac{AT}{2}\mathbf{x}(k) + \frac{BT}{2}e(k) \\ &\stackrel{\triangle}{=} \sqrt{T}\mathbf{w}(k+1). \end{aligned} \quad (6.29)$$

Writing the definition of \mathbf{w} at time k , solve for \mathbf{x} as before

$$\begin{aligned} \left(\mathbf{I} - \frac{AT}{2}\right)\mathbf{x} &= \sqrt{T}\mathbf{w} + \frac{BT}{2}e \\ \mathbf{x} &= \left(\mathbf{I} - \frac{AT}{2}\right)^{-1}\sqrt{T}\mathbf{w} + \left(\mathbf{I} - \frac{AT}{2}\right)^{-1}\frac{BT}{2}e. \end{aligned} \quad (6.30)$$

Substituting Eq. (6.30) into Eq. (6.29), we obtain

$$\begin{aligned} \sqrt{T}\mathbf{w}(k+1) &= \left(\mathbf{I} + \frac{AT}{2}\right)\left(\mathbf{I} - \frac{AT}{2}\right)^{-1}\left\{\sqrt{T}\mathbf{w}(k) + \frac{BT}{2}e\right\} + \frac{BT}{2}e(k) \\ \mathbf{w}(k+1) &= \left(\mathbf{I} + \frac{AT}{2}\right)\left(\mathbf{I} - \frac{AT}{2}\right)^{-1}\mathbf{w}(k) \\ &\quad + \left\{\left(\mathbf{I} + \frac{AT}{2}\right)\left(\mathbf{I} - \frac{AT}{2}\right)^{-1} + \mathbf{I}\right\}\frac{B\sqrt{T}}{2}e(k) \\ &= \left(\mathbf{I} + \frac{AT}{2}\right)\left(\mathbf{I} - \frac{AT}{2}\right)^{-1}\mathbf{w}(k) + \left(\mathbf{I} - \frac{AT}{2}\right)^{-1}\mathbf{B}\sqrt{T}e(k). \end{aligned} \quad (6.31)$$

In following this algebra, it is useful to know that in deriving the last part of Eq. (6.31), we expressed the identity \mathbf{I} as $(\mathbf{I} - \frac{AT}{2})(\mathbf{I} - \frac{AT}{2})^{-1}$ and factored out $(\mathbf{I} - \frac{AT}{2})^{-1}$ on the right.

To obtain the output equation for the bilinear equivalent, we substitute Eq. (6.30) into the second part of Eq. (6.27):

$$u(k) = \sqrt{T}\mathbf{C}\left(\mathbf{I} - \frac{AT}{2}\right)^{-1}\mathbf{w}(k) + \left\{\mathbf{D} + \mathbf{C}\left(\mathbf{I} - \frac{AT}{2}\right)^{-1}\frac{BT}{2}\right\}e(k).$$

These results can be tabulated for convenient reference. Suppose we have a continuous system described by

$$\dot{\mathbf{x}}(t) = \mathbf{Ax}(t) + \mathbf{Be}(t),$$

$$u(t) = \mathbf{Cx}(t) + \mathbf{De}(t).$$

⁶ The scale factor of \sqrt{T} is introduced so that the gain of the discrete equivalent will be balanced between input and output, a rather technical condition. See Al-Saggaf and Franklin (1986) for many more details.

Then a discrete equivalent at sampling period T will be described by the equations

$$\begin{aligned}\mathbf{w}(k+1) &= \Phi \mathbf{w}(k) + \Gamma e(k), \\ u(k) &= \mathbf{H} \mathbf{w}(k) + \mathbf{J} e(k),\end{aligned}$$

where Φ , Γ , \mathbf{H} , and \mathbf{J} are given respectively as follows:

	Forward	Backward	Bilinear
Φ	$\mathbf{I} + \Delta T$	$(\mathbf{I} - \Delta T)^{-1} \mathbf{B} T$	$(\mathbf{I} + \frac{\Delta T}{2})(\mathbf{I} - \frac{\Delta T}{2})^{-1}$
Γ	$\mathbf{B} T$	$(\mathbf{I} - \Delta T)^{-1}$	$(\mathbf{I} - \frac{\Delta T}{2})^{-1} \mathbf{B} \sqrt{T}$
\mathbf{H}		$\mathbf{C}(\mathbf{I} - \Delta T)^{-1}$	$\sqrt{T} \mathbf{C}(\mathbf{I} - \frac{\Delta T}{2})^{-1}$
\mathbf{J}	\mathbf{D}	$\mathbf{D} + \mathbf{C}(\mathbf{I} - \Delta T)^{-1} \mathbf{B} T$	$\mathbf{D} + \mathbf{C}(\mathbf{I} - \frac{\Delta T}{2})^{-1} \mathbf{B} T / 2$

The MATLAB Control Toolbox provides for the computation of Tustin bilinear equivalents with the function `c2d`. The syntax of computing the bilinear equivalent `SYSD` of a continuous system `SYS` at sampling period `Ts` is

$$\text{SYSD} = \text{c2d}(\text{SYS}, \text{Ts}, \text{'tustin'}) \quad (6.32)$$

If 'tustin' is replaced with 'prewarp', the bilinear equivalent with prewarping is computed.

6.2 Zero-Pole Matching Equivalents

A very simple but effective method of obtaining a discrete equivalent to a continuous transfer function is to be found by extrapolation of the relation derived in Chapter 4 between the s - and z -planes. If we take the z -transform of samples of a continuous signal $e(t)$, then the poles of the discrete transform $E(z)$ are related to the poles of $E(s)$ according to $z = e^{sT}$. We must go through the z -transform process to locate the zeros of $E(z)$, however. The idea of the zero-pole matching technique is that the map $z = e^{sT}$ could reasonably be applied to the zeros also. The technique consists of a set of heuristic rules for locating the zeros and poles and setting the gain of a z -transform that will describe a discrete, equivalent transfer function that approximates the given $H(s)$. The rules are as follows:

- All poles of $H(s)$ are mapped according to $z = e^{sT}$. If $H(s)$ has a pole at $s = -a$, then $H_{zp}(z)$ has a pole at $z = e^{-aT}$. If $H(s)$ has a pole at $-a + jb$, then $H_{zp}(z)$ has a pole at $re^{j\theta}$, where $r = e^{-aT}$ and $\theta = bT$.
- All finite zeros are also mapped by $z = e^{sT}$. If $H(s)$ has a zero at $s = -a$, then $H_{zp}(z)$ has a zero at $z = e^{-aT}$, and so on.
- The zeros of $H(s)$ at $s = \infty$ are mapped in $H_{zp}(z)$ to the point $z = -1$. The rationale behind this rule is that the map of real frequencies from $j\omega = 0$ to

increasing ω is onto the unit circle at $z = e^{j0} = 1$ until $z = e^{j\pi} = -1$. Thus the point $z = -1$ represents, in a real way, the highest frequency possible in the discrete transfer function, so it is appropriate that if $H(s)$ is zero at the highest (continuous) frequency, $|H_{zp}(z)|$ should be zero at $z = -1$, the highest frequency that can be processed by the digital filter.

- If no delay in the discrete response is desired, all zeros at $s = \infty$ are mapped to $z = -1$.
- If one sample period delay is desired to give the computer time to complete the output calculation, then one of the zeros at $s = \infty$ is mapped to $z = \infty$ and the others mapped to $z = -1$. With this choice, $H_{zp}(z)$ is left with a number of finite zeros one fewer than the number of finite poles.
- The gain of the digital filter is selected to match the gain of $H(s)$ at the band center or a similar critical point. In most control applications, the critical frequency is $s = 0$, and hence we typically select the gain so that

$$H(s)|_{s=0} = H_{zp}(z)|_{z=1}.$$

◆ Example 6.2 A Zero-pole Matching Equivalent

Compute the discrete equivalent to

$$H(s) = \frac{a}{s+a}$$

by zero-pole matching.

Solution. The pole of $H(s)$ at $s = -a$ will map to a pole of $H(z)$ at e^{-aT} . The zero at $s = \infty$ will map to a zero at $z = -1$. The gain of $H(s)$ at $s = 0$ is 1. To match this gain in $H(z)$ at $z = 1$ requires gain of $\frac{1 - e^{-aT}}{1 - e^{-aT}}$. The final function is given by

$$H_{zp}(z) = \frac{(z+1)(1 - e^{-aT})}{2(z - e^{-aT})}. \quad (6.33)$$

or, using rule 3(b), the result is

$$H_{zp}(z) = \frac{1 - e^{-aT}}{z - e^{-aT}}. \quad (6.34)$$

As with the rules based on numerical analysis, an algorithm to generate the matched zero-pole equivalent is also readily constructed. In MATLAB, a matched

zero-pole equivalent, SYSD, at sample period T_s to the continuous system, SYS, is given by

$$\text{SYSD} = \text{c2d}(\text{SYS}, T_s, \text{'matched'}).$$

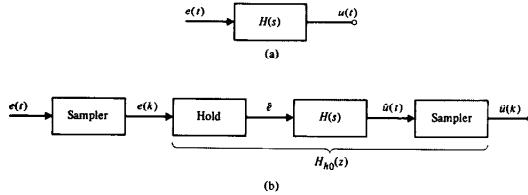
The frequency response of the matched zero-pole equivalent for the third-order Butterworth filter of Example 6.1 is plotted in Fig. 6.9 along with that of other equivalents for purposes of comparison.

6.3 Hold Equivalents

For this technique, we construct the situation sketched in Fig. 6.5. The samplers in Fig. 6.5(b) provide the samples at the input of $H_{h_0}(z)$ and take samples at its output insuring that $H_{h_0}(z)$ can be realized as a discrete transfer function. The philosophy of the design is the following. We are asked to design a discrete system that, with an input consisting of samples of $e(t)$, has an output that approximates the output of the continuous filter $H(s)$ whose input is the continuous $e(t)$. The discrete hold equivalent is constructed by first approximating $e(t)$ from the samples $e(k)$ with a hold filter and then putting this $e_h(t)$ through the given $H(s)$. There are many techniques for taking a sequence of samples and extrapolating or holding them to produce a continuous signal.⁷ Suppose we have the $e(t)$ as sketched in Fig. 6.6. This figure also shows a sketch of a piecewise constant approximation to $e(t)$

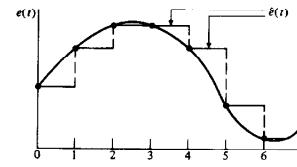
Figure 6.5

System construction for hold equivalents. (a) A continuous transfer function. (b) Block diagram of an equivalent system.



⁷ Some books on digital-signal processing suggest using no hold at all, using the equivalent $H(z) = \mathcal{Z}\{H(s)\}$. This choice is called the z-transform equivalent.

Figure 6.6
A signal, its samples, and its approximation by a zero-order hold



obtained by the operation of holding $e_h(t)$ constant at $e(k)$ over the interval from kT to $(k+1)T$. This operation is the *zero-order hold* (or ZOH) we've discussed before. If we use a first-order polynomial for extrapolation, we have a *first-order hold* (or FOH), and so on for higher-order holds.

6.3.1 Zero-Order Hold Equivalent

If the approximating hold is the zero-order hold, then we have for our approximation exactly the same situation that in Chapter 5 was analyzed as a sampled-data system.⁸ Therefore, the zero-order-hold equivalent to $H(s)$ is given by

$$H_{h_0}(z) = (1 - z^{-1}) \mathcal{Z} \left\{ \frac{H(s)}{s} \right\}. \quad (6.35)$$

◆ Example 6.3 A Hold Equivalent

Find the zero-order-hold equivalent to the first-order transfer function

$$H(s) = \frac{a}{s + a}.$$

Solution. The partial fraction expansion of the s-plane terms of Eq. (6.35) is

$$\frac{H(s)}{s} = \frac{a}{s(s+a)} = \frac{1}{s} - \frac{1}{s+a}$$

and the z-transform is

$$\mathcal{Z} \left\{ \frac{H(s)}{s} \right\} = \mathcal{Z} \left\{ \frac{1}{s} \right\} - \mathcal{Z} \left\{ \frac{1}{s+a} \right\}. \quad (6.36)$$

⁸ Recall that we noticed in Chapter 5 that the signal \hat{e} is, on the average, delayed from e by $T/2$ sec. The size of this delay is one measure of the quality of the approximation and can be used as a guide to the selection of T .

and, by definition of the operation given in Eq. (6.36)

$$\begin{aligned} \mathcal{Z}\left\{\frac{H(s)}{s}\right\} &= \sum_{k=0}^{\infty} z^{-k} - \sum_{k=0}^{\infty} z^{-k} e^{-akT} \\ &= \frac{1}{1-z^{-1}} - \frac{1}{1-e^{-aT}z^{-1}} \\ &= \frac{(1-e^{-aT}z^{-1}) - (1-z^{-1})}{(1-z^{-1})(1-e^{-aT}z^{-1})}. \end{aligned} \quad (6.37)$$

Substituting Eq. (6.37) in Eq. (6.35), the zero-order-hold equivalent of $H(s)$ is found as

$$H_{\text{ho}}(z) = \frac{(1-e^{-aT})}{z - e^{-aT}}. \quad (6.38)$$

We note that for the trivial example given, the zero-order-hold equivalent of Eq. (6.38) is identical to the matched zero-pole equivalent given by Eq. (6.34). However, this is not generally true as is evident in the comparison with frequency responses of other equivalents for the third-order Butterworth filter example plotted in Fig. 6.9. Because a sample and zero-order hold is an exact model for the sample and hold with A/D converter used in the majority of discrete systems, we have already seen the computation of this equivalent in MATLAB as

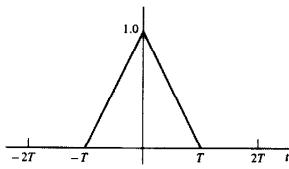
`SYSD = c2d(SYS, Ts, 'zoh')`

where the continuous system is described by `SYS` and the sample period is `Ts`.

6.3.2 A Non-Causal First-Order-Hold Equivalent: The Triangle-Hold Equivalent

An interesting hold equivalent can be constructed by imagining that we have a noncausal first-order-hold impulse response, as sketched in Fig. 6.7. The result is called the triangle-hold equivalent to distinguish it from the causal first order

Figure 6.7
Impulse response of the extrapolation filter for the triangle hold



◆ Example 6.4 A Triangle-Hold Equivalent

Compute the triangle-hold equivalent for $H(s) = 1/s^2$.

Solution. In this case, from the tables of z -transforms

$$\begin{aligned} \mathcal{Z}\left\{\frac{H(s)}{s^2}\right\} &= Z\left\{\frac{1}{s^2}\right\} \\ &= \frac{T^3}{6} \frac{(z^2 + 4z + 1)z}{(z - 1)^4}. \end{aligned} \quad (6.40)$$

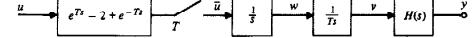
and direct substitution into Eq. (6.39) results in

$$\begin{aligned} H_{\text{th}}(z) &= \frac{(z - 1)^2}{Tz} \frac{T^3}{6} \frac{(z^2 + 4z + 1)z}{(z - 1)^4} \\ &= \frac{T^2 z^2 + 4z + 1}{6} \frac{z}{(z - 1)^2}. \end{aligned} \quad (6.41)$$

An alternative, convenient way to compute the triangle-hold equivalent is again to consider the state-space formulation. The block diagram is shown in Fig. 6.8. The continuous equations are

$$\begin{aligned} \dot{x} &= Fx + Gv, \\ v &= w/T, \\ \dot{w} &= u(t+T)\delta(t+T) - 2u(t)\delta(t) + u(t-T)\delta(t-T). \end{aligned} \quad (6.42)$$

Figure 6.8
Block diagram of the triangle-hold equivalent



and, in matrix form,

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{G} & \mathbf{0} \\ 0 & 0 & 1/T \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ v \\ w \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ 0 \\ 1 \end{bmatrix} \bar{u} \quad (6.43)$$

where \bar{u} represents the input impulse functions. We define the large matrix in Eq. (6.43) as F_T , and the one-step solution to this equation is

$$\xi(kT+1) = e^{F_T T} \xi(kT)$$

because \bar{u} consists only of impulses at the sampling instants. If we define

$$\exp(F_T T) = \begin{bmatrix} \Phi & \Gamma_1 & \Gamma_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.44)$$

then the equation in \mathbf{x} becomes

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) - \Gamma_1 v(k) + \Gamma_2 w(k).$$

With care, the last two equations of Eq. (6.42) can be integrated to show that $v(k) = u(k)$ and that $w(k) = u(k+1) - u(k)$. If a new state is defined as $\xi(k) = \mathbf{x}(k) - \Gamma_2 u(k)$, then the state equation for the triangle equivalent is

$$\begin{aligned} \xi(k+1) &= \Phi(\xi(k) + \Gamma_2 u(k)) + (\Gamma_1 - \Gamma_2)u(k) \\ &= \Phi\xi(k) + (\Gamma_1 + \Phi\Gamma_2 - \Gamma_2)u(k). \end{aligned} \quad (6.45)$$

The output equation is

$$\begin{aligned} y(k) &= \mathbf{H}\mathbf{x}(k) + \mathbf{J}u(k) \\ &= H(\xi(k) + \Gamma_2 u(k)) + Ju(k) \\ &= H\xi(k) + (J + H\Gamma_2)u(k). \end{aligned} \quad (6.46)$$

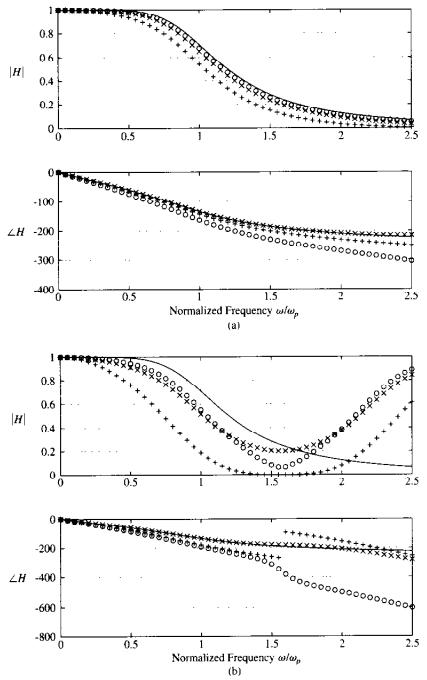
Thus the triangle equivalent of a continuous system described by $[\mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{J}]$ with sample period T is given by

$$\begin{aligned} \mathbf{A} &= \Phi, \\ \mathbf{B} &= \Gamma_1 + \Phi\Gamma_2 - \Gamma_2, \\ \mathbf{C} &= \mathbf{H}, \\ \mathbf{D} &= \mathbf{J} + \mathbf{H}\Gamma_2, \end{aligned} \quad (6.47)$$

where Φ , Γ_1 , and Γ_2 are defined by Eq. (6.44). In the MATLAB Control Toolbox, the function c2d will compute the triangle-hold equivalent (referenced there as a first-order-hold equivalent) of continuous system SYS by

$$\text{SYS}D = \text{c2d}(\text{SYS}, Ts, 'foh').$$

Figure 6.9
Comparison of digital equivalents for sampling period (a) $T = 1$ and $\omega_s/\omega_p = 2\pi$ and (b) $T = 2$ and $\omega_s/\omega_p = \pi$ where ZOH = \circ , and zero-pole = $+$, and triangle = \times



In Fig. 6.9 the frequency responses of the zero-pole, the zero-order hold, and the triangle-hold equivalents are compared again for the third-order Butterworth lowpass filter. Notice in particular that the triangle hold has excellent phase responses, even with the relatively long sampling period of $T = 2$, which corresponds to a sampling frequency to passband frequency ratio of only $\omega_s/\omega_p = \pi$.

6.4 Summary

In this chapter we have presented several techniques for the construction of discrete equivalents to continuous transfer functions so that known design methods for continuous systems can be used as a basis for the design of discrete systems. The methods presented were

1. Numerical integration

- (a) Forward rectangular rule
- (b) Backward rectangular rule
- (c) Trapezoid, bilinear, or Tustin's rule
- (d) Bilinear transformation with prewarping

2. Zero-pole matching

3. Hold equivalents

- (a) Zero-order-hold equivalent
- (b) Noncausal first-order- or triangle-hold equivalent

All methods except the forward rectangular rule guarantee a stable discrete system from a stable continuous prototype with the provision that the warping frequency of the bilinear transformation with prewarping must be less than the Nyquist frequency of $\frac{\pi}{T}$ rad/sec. Zero-pole matching is the simplest method to apply computationally if the zeros and poles of the desired filter are known and takes advantage of the known relations between response and poles and zeros. This is one of the most effective methods in the context of an overall design problem and in later chapters the zero-pole matching method is frequently selected. With a reasonable computer-aided-design tool, the designer can select the method that best meets the requirements of the design. The MATLAB function `c2d` computes the discrete description for most of these discrete equivalents from a continuous system described by `SYS` with sample period T 's as follows.

<code>SYS</code>	<code>=</code>	<code>c2d(SYS, Ts, method)</code>	where
<code>method</code>	<code>=</code>	<code>'zoh'</code>	zero-order hold
<code>method</code>	<code>=</code>	<code>'foh'</code>	first-order hold (triangle hold)
<code>method</code>	<code>=</code>	<code>'tustin'</code>	Tustin's bilinear method
<code>method</code>	<code>=</code>	<code>'prewarp'</code>	bilinear with prewarping
<code>method</code>	<code>=</code>	<code>'matched'</code>	zero-pole matching

6.5 Problems

- 6.1 Sketch the zone in the z -plane where poles corresponding to the left half of the s -plane will be mapped by the zero-pole mapping technique and the zero-order-hold technique.
- 6.2 Show that Eq. (6.15) is true.
- 6.3 The following transfer function is a lead network designed to add about 60° phase lead at $\omega_1 = 3$ rad

$$H(s) = \frac{s + 1}{0.1s + 1}.$$

- (a) For each of the following design methods compute and plot in the z -plane the pole and zero locations and compute the amount of phase lead given by the equivalent network at $z_1 = e^{j\omega_1 T}$ if $T = 0.25$ sec and the design is via
- i. Forward rectangular rule
 - ii. Backward rectangular rule
 - iii. Bilinear rule
 - iv. Bilinear with prewarping (use ω_1 as the warping frequency)
 - v. Zero-pole mapping
 - vi. Zero-order-hold equivalent
 - vii. Triangle-hold equivalent
- (b) Plot over the frequency range $\omega_l = 0.1 \rightarrow \omega_h = 100$ the amplitude and phase Bode plots for each of the above equivalents.
- 6.4 The following transfer function is a lag network designed to increase K_c by a factor of 10 and have negligible phase lag at $\omega_1 = 3$ rad.
- $$H(s) = 10 \frac{10s + 1}{100s + 1}.$$
- (a) For each of the following design methods, compute and plot on the z -plane the zero-pole patterns of the resulting discrete equivalents and give the phase lag at $z_1 = e^{j\omega_1 T}$ corresponding to $\omega_1 = 3$. Let $T = 0.25$ sec.
- i. Forward rectangular rule
 - ii. Backward rectangular rule
 - iii. Bilinear rule
 - iv. Bilinear with prewarping (Use $\omega_1 = 3$ radians as the warping frequency)
 - v. Zero-pole mapping
 - vi. Zero-order-hold equivalent
 - vii. Triangle-hold equivalent
- (b) For each case computed, plot the Bode amplitude and phase curves over the range $\omega_l = 0.01 \rightarrow \omega_h = 10$ rad.

• 7 •

Design Using Transform Techniques

A Perspective on Design Using Transform Techniques

The idea of controlling processes that evolve in time is ubiquitous. Systems from airplanes to the national rate of unemployment, from unmanned space vehicles to human blood pressure, are considered fair targets for control. Over a period of three decades from about 1930 until 1960, a body of control theory was developed based on electronic feedback amplifier design modified for servomechanism problems. This theory was coupled with electronic technology suitable for implementing the required dynamic compensators to give a set of approaches to solve control problems now often called **classical techniques** to distinguish these methods from designs based on a state-space formulation which came to be called **modern techniques**. The landmark contributors to this "classical" theory are Evans (1950) [root locus] and Nyquist (1932) and Bode (1945) [frequency response]. For random inputs, the work of Wiener (1948) should be added. The unifying theme of these methods is the use of Fourier and Laplace transforms to represent signals and system dynamics and to describe the control specifications. Controller design is then carried out in the selected transform domain. From the perspective of the 90's the terms "classical" and "modern" seem a bit pejorative and we prefer to classify the methods as **transform techniques** and **state-space techniques**.

The methods based on transforms were developed before computers were available and the engineer had to depend on hand calculations and careful hand plotting to achieve the design. The availability of computers and software such as MATLAB have made calculations and plotting simple, fast and accurate; and now the hand-plotting guidelines are used as verification of the automatic calculations and as a guide to design decisions. In this role, the understanding of the design process gained by the experience of doing a simple design by hand is well

design by emulation

worth the effort spent in developing the required skills. The introduction of digital control and sampled data adds new constraints and new possibilities to the transform design methods. The z -transform is added to the Laplace and the Fourier transforms and poles and zeros have meaning relative to the unit circle rather than to the imaginary axis. The meaningful part of the frequency response is restricted to half the sampling frequency. Each of these developments must be understood in order to apply transform methods to digital control.

Chapter Overview**design by root locus**

Building on previous understanding of the design of continuous systems, the first method for digital design is based on **emulation** of a continuous design. The continuous controller is simply replaced with a digital equivalent computed by using one of the techniques described in Chapter 6. The result may be evaluated in terms of poles and zeros in the z -plane, magnitude and phase in the frequency response, or transient response to step, impulse or other input.

The second method introduced is the **root locus** where it is demonstrated that the rules of the root locus are unchanged from the continuous case but the relations between pole location and time response must refer to the z -plane rather than the s -plane.

Finally, the Nyquist stability criterion for discrete systems is developed and Bode's design methods for gain and phase margins are extended to discrete systems. In addition to the usual results, the concept of system sensitivity is developed to show how **frequency response** can be used to cause the system to be robust with respect to both stability and performance when the plant transfer function is subjected to bounded but unknown perturbations.

design by frequency response**7.1 System Specifications**

We first consider the design specifications that the controller is expected to achieve. As reviewed in Chapter 2, the central concerns of controller design are for good transient and steady-state response and for sufficient robustness. Requirements on time response and robustness need to be expressed as constraints on s -plane pole and zero locations or on the shape of the frequency response in order to permit design in the transform domains. Dynamic performance in the time domain is defined in terms of parameters of system response to a step in command input. The most frequently used parameters are the rise time, t_r ; the settling time, t_s ; the percent overshoot, M_p ; and the steady-state error, e_{ss} . These parameters, which apply equally well to discrete control as to continuous control, are discussed in Section 4.1.7. The s -plane expressions of these requirements are summarized by the following guidelines:

velocity constant

- The requirement on natural frequency is

$$\omega_n \geq 1.8/t_r. \quad (7.1)$$

- The requirement on the magnitude of the real part of the pole is

$$|\operatorname{Re}\{s_j\}| - \sigma = \xi \omega_n \geq 4.6/t_s. \quad (7.2)$$

- The fractional overshoot, M_p , is given in terms of the damping ratio, ξ , by the plot of Fig. 4.7 which can be very crudely approximated by

$$\xi \approx 0.6(1 - M_p). \quad (7.3)$$

The specifications on steady-state error to polynomial inputs is determined by the error constant appropriate for the case at hand as described in Section 4.2.2. The most common case is for systems of Type 1, which is to say, systems that have zero steady-state error to a step input and finite error to a ramp input of slope r_0 of size $e_{ss} = r_0/K_v$ where K_v is the velocity constant. For a single-loop system with *unity* feedback gain and forward transfer function $D(s)G(s)$ as shown in Fig. 7.1, the system is **Type 1** if $DG(s)$ has a simple pole at $s = 0$.

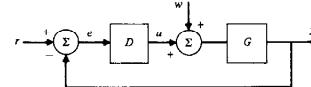
- The **velocity constant** is then given by

$$K_v = \frac{r_0}{e_{ss}} = \lim_{s \rightarrow 0} s D(s)G(s).$$

The fact that in discrete systems the control is applied as a piecewise constant signal causes a roughness in the response that is directly related to the sample frequency. A specification on roughness leads to a specification on sample period, T . This parameter is so important to the design of discrete controls that Chapter 11 is devoted to the decision. At this point, let it suffice to point out that the smaller the T , the better the approximation to continuous control and the smaller the roughness.

- A reasonable choice of T is one that results in at least 6 samples in the closed-loop rise time and better, smoother control results if there are more than 10 samples in the rise time.

Figure 7.1
A unity feedback system



◆ Example 7.1 Selection of Sample Period

What is the relation between sampling frequency and system natural frequency if there are 10 samples in a rise time?

Solution. The sampling frequency in radians/sec is given by $\omega_s = 2\pi/T$ and we assume that rise time and natural frequency are related by Eq. (7.1) so that

$$\begin{aligned}\omega_n &= 1.8/t_r \\ &= \frac{1.8}{10T}.\end{aligned}$$

Substituting for T , we find that

$$\omega_s = \frac{0.18\omega_n}{2\pi}$$

or

$$\omega_s \approx \omega_n/35.$$

In other words, the sample rate, ω_s , should be 35 times faster than the natural frequency, ω_n .

From this example, we conclude that typically the sample frequency should be chosen to be 20 to 35 times the closed loop natural frequency. Slower sampling can be used but one would expect the resulting transients to be excessively rough.

Robustness is the property that the dynamic response (including stability of course) is satisfactory not only for the nominal plant transfer function used for the design but also for the entire class of transfer functions that express the uncertainty of the designer about the dynamic environment in which the real controller is expected to operate. A more comprehensive discussion of robustness will be given when design using frequency response is considered. For root locus design, the natural measure of robustness is, in effect, gain margin. One can readily compare the system gain at the desired operating point and at the point(s) of onset of instability to determine how much gain change is acceptable.

- A typical robustness requirement is that one should have gain margin of two so that the loop gain must double from the design value before reaching the stability boundary.

7.2 Design by Emulation

The elements of design by emulation have been covered already. Continuous control design is reviewed in Chapter 2, and in Chapter 6 the techniques of computing discrete equivalents are described. Control design by emulation is

mainly a combination of these two ideas. A controller design is done as if the system is to be continuous and, after a sample period is selected, a discrete equivalent is computed and used in place of the continuous design. This discrete controller may then be simulated and tested in the discrete control loop and modifications made, if necessary.

7.2.1 Discrete Equivalent Controllers

Techniques to compute discrete equivalents are described in general terms in Chapter 6, and their performance is illustrated on the basis of filter frequency responses. In this chapter, we are interested in controllers for feedback control and in performance comparisons on the basis of time responses. Any of the techniques from Chapter 6 can be used for the purpose; here we illustrate the use of the pole-zero mapping equivalent and explore the choice of sample period by example. An alternative approach that considers directly the performance for the discrete controller in the feedback context has been described by Anderson (1992). The method described in that reference leads to a multirate sampling problem of the sort which will be considered in Chapter 11.

◆ Example 7.2 Design of Antenna Servo Controller

A block diagram of the plant for an antenna angle-tracker is drawn in Fig. 7.2. The transfer function is given by

$$G(s) = \frac{1}{s(10s + 1)}.$$

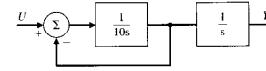
The specifications for this system are

1. Overshoot to a step input less than 16%
2. Settling time to 1% to be less than 10 sec
3. Tracking error to a ramp input of slope 0.01 rad/sec to be less than 0.01 rad
4. Sampling time to give at least 10 samples in a rise-time

Design a controller for this system using the method of emulation.

Solution. From the specifications one can estimate the acceptable region in the s -plane for the closed loop poles. From the overshoot requirement, we conclude that the damping ratio

Figure 7.2
Block diagram of the plant transfer function



must be $\zeta \geq 0.5$. From the settling time requirement, we conclude that the roots must have a real part of $\sigma \geq 4.6/10 = 0.46$. Finally, from the steady-state error requirement, we conclude that the velocity constant is constrained to be $K_v \geq \frac{0.01}{0.001} = 1.0$. Based on the limits on the damping ratio and the real-part of the poles, we can sketch the acceptable region for closed-loop poles in the s -plane as done in Fig. 7.3. Using lead compensation to cancel a plant pole, a first choice for controller might be

$$D(s) = \frac{10s + 1}{s + 1}. \quad (7.1)$$

The root locus for this choice is drawn in Fig. 7.4 using MATLAB commands to enter the plant as a system, ant, the compensation as lead1, and the product as the open loop system, sysol.

```
np = 1;
dp = [10 1 0];
ant = tf(np,dp);
nc = [10 1];
dc = [1 1];
lead1 = tf(nc,dc,0.2);
sysol = lead1*ant;
rlocus(sysol)
```

Figure 7.3
Acceptable pole locations for the antenna control

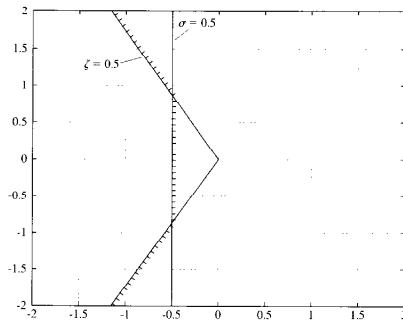
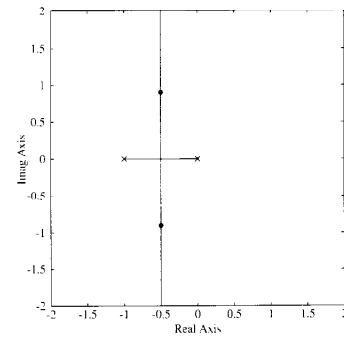


Figure 7.4
Root locus for compensated antenna model



The locations of the roots with $K = 1$ corresponding to a velocity constant of $K_v = 1$ are marked by the dots computed by

$$p = rlocus(sysol, 1, 0).$$

The natural frequency for the given pole locations is essentially $\omega_n = 1$, which corresponds to a rise time of $t_r = 1.8$ sec. The indicated sampling period is thus $T = t_r/10 = 0.18$. A value of $T = 0.2$ will be used for this example and a value of $T = 1.0$ illustrated later to dramatize the effects of the choice of T . The compensation, $D(s)$, given by Eq. (7.4), has two first-order factors: the zero is at $s = -0.1$, and the pole is at $s = -1$. The pole-zero mapping technique requires that each singularity is mapped according to $z = e^{sT}$; therefore, we take $D(z)$ of the form

$$D(z) = K \frac{z - z_1}{z - p_1},$$

and place a zero at

$$z_1 = e^{-(0.1)(0.2)} = 0.9802,$$

and a pole at

$$p_1 = e^{-(1)(0.2)} = 0.8187.$$

To make the dc gain of $D(z)$ and $D(s)$ be identical, we require that

$$\begin{aligned} \text{dc gain} &= \lim_{z \rightarrow 1} D(z) = \lim_{s \rightarrow 0} D(s) = 1 \\ &= K \frac{1 - 0.9802}{1 - 0.8187}. \end{aligned} \quad (7.5)$$

Solving for K we have

$$K = 9.15,$$

and the design of the discrete equivalent compensation has the transfer function

$$D(z) = 9.15 \frac{z - 0.9802}{z - 0.8187} \quad (7.6)$$

To compute this result in MATLAB, the command is
`lead1d = c2d(lead1,0.2,'matched');`

◆ Example 7.3 Implementing the Controller

Give the difference equation that corresponds to the $D(z)$ given by Eq. (7.6).

Solution. The transfer function is converted into a difference equation for implementation using the ideas developed in Chapter 4. Specifically, we first multiply top and bottom by z^{-1} to obtain

$$D(z) = \frac{U(z)}{E(z)} = 9.15 \frac{1 - 0.9802z^{-1}}{1 - 0.8187z^{-1}},$$

which can be restated as

$$(1 - 0.8187z^{-1})U(z) = 9.15(1 - 0.9802z^{-1})E(z).$$

The z -transform expression above is converted to the difference equation form by noting that z^{-1} represents a 1-cycle delay. Thus

$$u(k) = 0.8187u(k-1) + 9.15(e(k) - 0.9802e(k-1)).$$

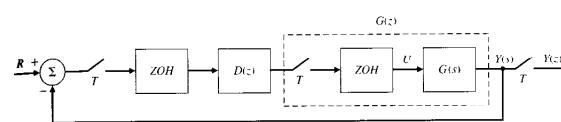
This equation can be directly evaluated by a computer.

7.2.2 Evaluation of the Design

A description of a digital controller that is expected to satisfy the specifications for the antenna controller is now complete. A block diagram of the sampled-data system with discrete controller is given in Fig. 7.5. To analyze the behavior of this compensation, we first determine the z -transform of the continuous plant (Fig. 7.2) preceded by a zero-order hold (ZOH).

$$G(z) = \frac{z-1}{z} \mathcal{Z} \left\{ \frac{a}{s^2(s+a)} \right\}. \quad (7.7)$$

Figure 7.5 Block diagram of sampled-data system



which is

$$G(z) = \frac{z-1}{z} \mathcal{Z} \left\{ \frac{1}{s^2} - \frac{1}{as} + \frac{1}{a} \frac{1}{s+a} \right\}.$$

Using the tables in Appendix B, we find

$$\begin{aligned} G(z) &= \frac{z-1}{z} \left\{ \frac{Tz}{(z-1)^2} - \frac{z}{a(z-1)} + \frac{1}{a} \frac{z}{az - e^{-aT}} \right\} \\ &= \frac{Az + B}{a(z-1)(z - e^{-aT})}, \end{aligned}$$

where

$$A = e^{-aT} + aT - 1, \quad B = 1 - e^{-aT} - aTe^{-aT}.$$

For this example, with $T = 0.2$ and $a = 0.1$, this evaluates to

$$G(z) = 0.00199 \frac{z + 0.9934}{(z-1)(z - 0.9802)}. \quad (7.8)$$

Of course, this can be computed in MATLAB as the discrete model of the antenna by

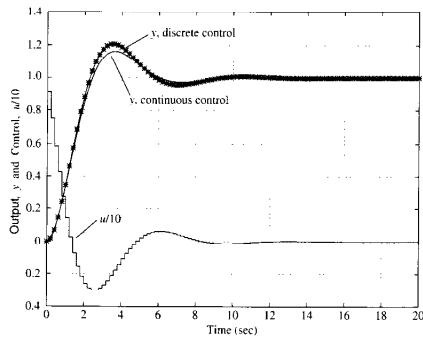
$$\text{antd} = \text{c2d}(\text{ant}, 0.2).$$

With the transfer function of the plant-plus-hold and the discrete controller, we can obtain the system difference equation and compute the step response, obviously most easily using a computer aided design package. The steps are

```
sysold = lead1 * antd
syscl = connect(sysold, [1 - 1])
step(syscl).
```

In this case, the step response of the system with the discrete controller is shown in Fig. 7.6. The figure confirms that the discrete controller will perform satisfactorily, albeit with somewhat increased overshoot. This simulation was carried out using the linear, discrete model of the system. As mentioned earlier, simulations can be embellished with the important nonlinearities such as friction and with

Figure 7.6
Step response of the 5 Hz controller



computation delays in order to assess their effects in addition to the effect of the discretization approximations.

◆ **Example 7.4** Antenna Servo with Slow Sampling

Repeat the antenna design with a sample rate of 1 Hz ($T = 1$ sec); in this case the sample rate is approximately two samples per rise time.

Solution. Repeating the calculations as in Eq. (7.7) with $T = 1$ sec results in

$$G(z) = 0.0484 \frac{z + 0.9672}{(z - 1)(z - 0.9048)} \quad (7.9)$$

Furthermore, repeating the calculations that led to Eq. (7.6) but with $T = 1$ sec, we obtain

$$D(z) = 6.64 \frac{z - 0.9048}{z - 0.3679} \quad (7.10)$$

A plot of the step response of the resulting system is shown in Fig. 7.7 and shows substantial degradation of the response as a result of the slow sampling. A partial explanation of the extra overshoot can be obtained by looking at the Bode plot of the continuous design, computed with `bode(sys0)` and plotted in Fig. 7.8. The designed phase margin in the continuous system is seen to be 51.8°. As was indicated in Chapter 4, the sample and hold can be roughly approximated by a delay of $T/2$ sec. At the crossover frequency of $\omega_p = 0.8$ rad, and with sampling at $T = 0.2$, this corresponds only to $\phi = \omega_p T = 4.5^\circ$. However, at $T = 1.0$, the sample-hold delay corresponds to $\phi = 23^\circ$. Thus the effective phase margin with a sample and hold is reduced to $Pm = 51.8^\circ - 23^\circ = 28.8^\circ$. With this small phase margin, the effective damping

Figure 7.7
Step response of the 1-Hz controller

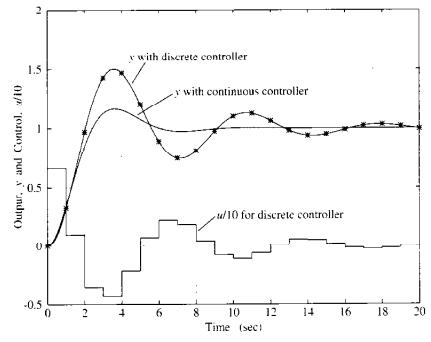
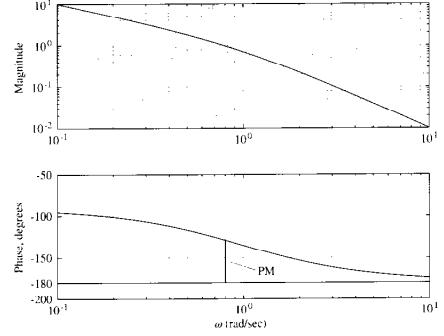


Figure 7.8
Bode plot of the continuous design for the antenna control



ratio is about 0.29 and the overshoot is expected to be about 0.4 rather than 0.16 as designed. The step response shows the actual $M_p = 0.5$, so most of the extra overshoot is explained by the sample-hold delay. ◆

The examples presented here illustrate only a small selection of the alternatives for design by emulation. An immediate improvement would be expected if the continuous design were to include at the outset the characteristic $T/2$ delay of the sample and zero-order hold. Other than this modification, the other algorithms for discrete equivalent design can be tried. These include the very simple Euler rectangular rules, the bilinear transformations, and the several hold-equivalent methods. The triangle hold equivalent appears to be especially promising.¹ There does not seem to be a dominant technique that is best for every case. The designer needs to explore alternatives based on the particular system, the required performance specifications and the practical constraints introduced by the technology to be used for implementation to guide the final choice. Here we now turn to consider the direct discrete design methods, beginning with design by use of the root locus in the z -plane.

7.3 Direct Design by Root Locus in the z -Plane

The root locus introduced by W. Evans is based on graphical rules for plotting the roots of a polynomial as a parameter is varied. The most common root locus is a plot of the roots of a closed-loop characteristic polynomial in the s -plane as the loop gain is varied from 0 to ∞ . In linear discrete systems also the dynamic performance is largely determined by the roots of the closed-loop characteristic polynomial, in this case a polynomial in z with stability represented by having all roots inside the unit circle. The consequences for direct digital design are that one can use Evans' root locus rules unchanged, but that the performance specifications must first be translated into the z -plane.

7.3.1 z -Plane Specifications

Figure 4.26 is a map of the unit disk in the z -plane on which is superimposed discrete system time responses that correspond to several typical z -plane pole locations. These can be used to make the translation of dynamic response performance specifications to a region of acceptable pole locations. For example, we have seen that rise time of a continuous second-order system is found to be inversely proportional to natural frequency as given by Eq. (7.1). Since poles in the s -plane are mapped to $z = e^{sT}$, the natural frequency in s maps to the angle of the pole in polar coordinates in the z -plane as $\theta = \omega_d T$ where $\omega_d = \sqrt{1 - \zeta^2} \omega_n$. Settling time is found to be inversely proportional to the magnitude of the real part of a pole in the s -plane (σ) which maps to the radius of the pole in the z -plane as $r = e^{-\sigma T}$. The step response overshoot varies inversely with the damping

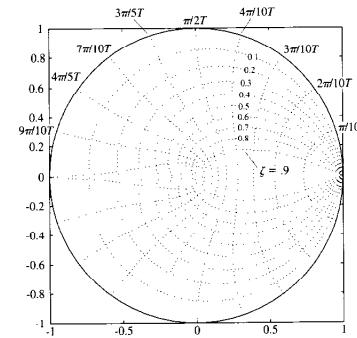
¹ In the MATLAB Control Toolbox function c2d, the triangle hold is called a first-order hold in recognition of the fact that it is a first-order hold although it is *noncausal*.

getting acceptable pole location in the z -plane

ratio. Under the s -to- z mapping, lines of constant damping map into logarithmic spirals in the z -plane. With these guidelines, one can readily estimate the dynamic response parameters based on the pole-zero pattern for simple transfer functions and can derive useful guidelines for design of more complex systems. In summary, to get the specifications on acceptable pole locations in the z -plane

- Estimate the desired ω_n , ζ , and M_p from the continuous-time response specifications. Compute $\sigma = \zeta \omega_n$.
- Compute the radius $r = e^{-\sigma T}$.
- Obtain a plot of the z -plane showing lines of fixed damping and ω_n . The MATLAB command zgrid will do this, plotting ζ in steps of 0.1 from 0.1 to 0.9 and $\omega_n = N\pi/10T$ for integer N from 1 to 10. An example is shown in Fig. 7.9. The command axis equal will cause the unit circle to be plotted as a circle and the command axis([-1 1 0 1]) will cause only the upper half of the circle to be plotted.
- Mark the region of acceptable closed-loop pole locations on the plane.

Figure 7.9
Lines of constant damping and natural frequency in the z -plane



◆ Example 7.5 *Z-Plane Specifications*

Indicate on a *z*-plane map the region of acceptable closed-loop poles for the antenna design of Example 7.2.

Solution. The given specifications are that the system is to have a damping ratio of $\xi \geq 0.5$, natural frequency of $\omega_n \geq 1$, and the real-parts of the roots are to be greater than 0.5. The standard grid of the *z*-plane shows the curve corresponding to $\xi = 0.5$. With the requirement that the roots correspond to a natural frequency greater than $\omega_n = 1$, we need a plot on the standard grid corresponding to $N = 10T\omega_n/\pi = 2/\pi \approx 0.64$. The last requirement means that the roots in the *z*-plane must be inside a circle of radius $r \leq e^{-0.5T} = 0.9048$. The curves corresponding to these criteria are marked in Fig. 7.10.

The specification of steady-state error also follows the continuous case but transferred to the *z*-plane when the controller is implemented in a computer and represented by its discrete transfer function $D(z)$. The discrete transfer function of the plant is given by

$$G(z) = (1 - z^{-1})Z \left[\frac{G(s)}{s} \right]. \quad (7.11)$$

The closed-loop system can now be represented in a purely discrete manner. The discrete transfer functions of the controller, $D(z)$, and the plant, $G(z)$, are

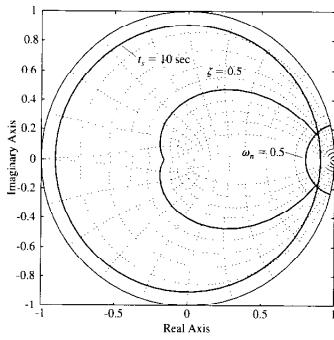


Figure 7.10
Plot of acceptable pole locations in the *z*-plane

combined as before according to Fig. 7.5, where it is now understood that the reference r and the disturbance w are sampled versions of their continuous counterparts. Proceeding as we did for the continuous system, suppose the input r is a step, $r(k) = 1(k)$, and the disturbance w is zero. The transform of the error is computed using the same block-diagram reduction tools that apply for continuous systems represented by their Laplace transforms, except that now we use $D(z)$ and $G(z)$. Doing this yields the transform of the error

$$E(z) = \frac{R(z)}{1 + D(z)G(z)} - \frac{z}{z - 1} \frac{1}{1 + D(z)G(z)}.$$

discrete time final value

The final value of $e(k)$, if the closed loop system is stable with all roots of $1 + DG = 0$ inside the unit circle, is, by Eq. (4.115)

$$\begin{aligned} e(\infty) &= \lim_{z \rightarrow 1} \frac{z}{z - 1} \frac{1}{1 + D(z)G(z)} \\ &= \frac{1}{1 + D(1)G(1)} \\ &= \frac{1}{1 + K_p}. \end{aligned} \quad (7.12)$$

discrete system type

Thus, $D(1)G(1)$ is the position error constant, K_p , of the Type 0 system in discrete time if the limit in Eq. (7.12) is finite. If DG has a pole at $z = 1$, then the error given by Eq. (7.12) is zero. Suppose there is a single pole at $z = 1$. Then we have a Type 1 system and we can compute the error to a unit ramp input, $r(kT) = kT1(kT)$ as

$$E(z) = \frac{Tz}{(z - 1)^2} \frac{1}{1 + D(z)G(z)}.$$

Now the steady-state error is

$$\begin{aligned} e(\infty) &= \lim_{z \rightarrow 1} \frac{Tz}{(z - 1)^2} \frac{1}{1 + DG} \\ &= \lim_{z \rightarrow 1} \frac{Tz}{(z - 1)(1 + DG)} \\ &\triangleq \frac{1}{K_v}. \end{aligned} \quad (7.13)$$

Thus the velocity constant of a Type 1 discrete system with unity feedback (as shown in Fig. 7.5) is

$$K_v = \lim_{z \rightarrow 1} \frac{(z - 1)(1 + D(z)G(z))}{Tz},$$

which simplifies to

$$K_v = \lim_{z \rightarrow 1} \frac{(z-1)D(z)G(z)}{Tz}. \quad (7.14)$$

Although it appears from Eq. (7.14) that K_v is inversely proportional to the sample period, this is not the case if comparing for the same $G(s)$. The reason is that the transfer function of $G(z)$ computed from Eq. (7.11) is typically proportional to the sample period. This proportionality is exact for the very simple case where $G(s) = 1/s$, as can be seen by using Eq. (7.11) and inspecting Entry 4 in Appendix B.2. For systems with a finite K_v and fast sample rates, this proportionality will be approximately correct. The result of this proportionality is that the dc gain of a continuous plant alone preceded by a ZOH is essentially the same as that of the continuous plant.

*Truxal's Rule, Discrete Case

Because systems of Type 1 occur frequently, it is useful to observe that the value of K_v is fixed by the *closed-loop* poles and zeros by a relation given, for the continuous case, by Truxal (1955). Suppose the overall transfer function Y/R is $H(z)$, and that $H(z)$ has poles p_i and zeros z_i . Then we can write

$$H(z) = K \frac{(z - z_1)(z - z_2) \cdots (z - z_n)}{(z - p_1)(z - p_2) \cdots (z - p_n)}. \quad (7.15)$$

Now suppose that $H(z)$ is the closed-loop transfer function that results from a Type 1 system, which implies that the steady-state error of this system to a step is zero and requires that

$$H(1) = 1. \quad (7.16)$$

Furthermore, by definition we can express the error to a ramp as

$$\begin{aligned} E(z) &= \frac{R(z)(1 - H(z))}{Tz} \\ &= \frac{(z-1)^2}{(z-1)^2}(1 - H(z)), \end{aligned}$$

and the final value of this error is given by

$$e(\infty) = \lim_{z \rightarrow 1} (z-1) \frac{Tz}{(z-1)^2}(1 - H(z)) \triangleq \frac{1}{K_v};$$

therefore (omitting a factor of z in the numerator, which makes no difference in the result)

$$\frac{1}{TK_v} = \lim_{z \rightarrow 1} \frac{1 - H(z)}{z - 1}. \quad (7.17)$$

Because of Eq. (7.16), the limit in Eq. (7.17) is indeterminate, and so we can use L'Hôpital's rule

$$\begin{aligned} \frac{1}{TK_v} &= \lim_{z \rightarrow 1} \frac{(d/dz)(1 - H(z))}{(d/dz)(z - 1)} \\ &= \lim_{z \rightarrow 1} \left\{ -\frac{dH(z)}{dz} \right\}. \end{aligned}$$

However, note that by using Eq. (7.16) again, at $z = 1$, we have

$$\frac{d}{dz} \ln H(z) = \frac{1}{H} \frac{d}{dz} H(z) = \frac{d}{dz} H(z).$$

so that

$$\begin{aligned} \frac{1}{TK_v} &= \lim_{z \rightarrow 1} -\frac{d}{dz} \ln H(z) \\ &= \lim_{z \rightarrow 1} \frac{d}{dz} \left\{ \ln K \frac{\prod(z - z_i)}{\prod(z - p_i)} \right\} \\ &= \lim_{z \rightarrow 1} \frac{d}{dz} \left\{ \sum \ln(z - z_i) - \sum \ln(z - p_i) + \ln K \right\} \\ &= \lim_{z \rightarrow 1} \left\{ \sum \frac{1}{z - p_i} - \sum \frac{1}{z - z_i} \right\} \\ &= \sum_{i=1}^n \frac{1}{1 - p_i} - \sum_{i=1}^n \frac{1}{1 - z_i}. \end{aligned}$$

We note especially that the farther the *poles* of the closed-loop system are from $z = 1$, the larger the velocity constant and the smaller the errors. Similarly, K_v can be increased and the errors decreased by *zeros close to $z = 1$* . From the results of Chapter 4 on dynamic response, we recall that a zero close to $z = 1$ usually yields large overshoot and poor dynamic response. Thus is expressed one of the classic trade-off situations: We must balance small steady-state errors against good transient response.

7.3.2 The Discrete Root Locus

The root locus is the locus of points where roots of a characteristic equation can be found as some real parameter varies from zero to large values. From Fig. 7.5 and block-diagram analysis, the characteristic equation of the single-loop system is

$$1 + D(z)G(z) = 0. \quad (7.18)$$

The significant thing about Eq. (7.18) is that this is exactly the same equation as that found for the s -plane root locus. The implication is that the mechanics of drawing the root loci are exactly the same in the z -plane as in the s -plane; the rules for the locus to be on the real axis, for asymptote construction, and

for arrival/departure angles are all unchanged from those developed for the s -plane and reviewed in Chapter 2. As mentioned earlier, the difference lies in the interpretation of the results because the pole locations in the z -plane mean different things than pole locations in the s -plane when we come to interpret system stability and dynamic response.

◆ Example 7.6 Discrete Root Locus Design

Design the antenna system for the slow sampling case with $T = 1$ sec, using the discrete root locus.

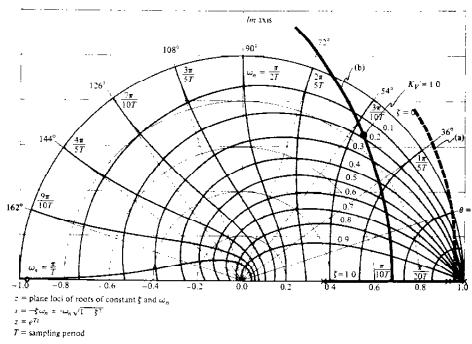
Solution. The exact discrete model of the plant plus hold is given by the $G(z)$ in Eq. (7.9). If the controller consisted simply of a proportional gain [$D(z) = K$], the locus of roots versus K can be found by solving the characteristic equation

$$1 + 0.0484K \frac{z + 0.9672}{(z - 1)(z - 0.9048)} = 0$$

for many values of K . The result computed by `rlocus(antd)` is shown in Fig. 7.11 as the dashed arc marked (a). From study of the root locus we should remember that this locus, with two poles and one zero, is a circle centered at the zero ($z = -0.9672$) and breaking away from the real axis between the two real poles.

From the root locus of the uncompensated system (Fig. 7.11(a)) it is clear that some dynamic compensation is required if we are to get satisfactory response from this system. The

Figure 7.11
Root loci for antenna design: (a)
Uncompensated system;
(b) Locus with $D(z)$
having the poles and
zeros of Eq. (7.10)



radius of the roots never gets less than 0.95, preventing the t_r specification from being met. The system goes unstable at $K \cong 19$ [where $K_c = 0.92$, as can be verified by using Eq. (7.14)], which means that there is no stable value of gain that meets the steady-state error specification with this compensation.

If we cancel the plant pole at 0.9048 with a zero and add a pole at 0.3679, we are using the lead compensation of Eq. (7.10). The root locus for this control versus the gain K [K was equal to 6.64 in Eq. (7.10)] computed with `rlocus(sysold)` is also sketched in Fig. 7.11 as the solid curve (b). The points, p , where $K = 6.64$ are computed with $p = rlocus(sysold, 6.64)$ and marked by dots. We can see that a damping ratio of about 0.2 is to be expected, as we have previously seen from the step response of Fig. 7.7. This gain, however, does result in the specified value of $K_p = 1$ because this criterion was used in arriving at Eq. (7.10). The locus shows that increasing the gain, K , would lower the damping ratio still further. Better damping could be achieved by decreasing the gain, but then the criterion of steady-state error would be violated. It is therefore clear that this choice of compensation pole and zero cannot meet the specifications.

A better choice of compensation can be expected if we transform the specifications into the z -plane and select the compensation so that the closed loop roots meet those values. The original specifications were $K_p \geq 1$, $t_r \leq 10$ sec, and $\zeta \geq 0.5$. If we transform the specifications to the z -plane we compute that the t_r specification requires that the roots be inside the radius $r = e^{-0.5} = 0.61$, and the overshoot requires that the roots are inside the $\zeta = 0.5$ spiral. The requirement that $K_p \geq 1$ applies in either plane but is computed by Eq. (7.14) for the z -plane.

It is typically advantageous to use the design obtained using emulation and to modify it using discrete design methods so that it is acceptable. The problem with the emulation-based design is that the damping is too low at the mandated gain, a situation that is typically remedied by adding more lead in the compensation. More lead is obtained in the s -plane by increasing the separation between the compensation's pole and zero; and the same holds true in the z -plane. Therefore, for a first try, let's keep the zero where it is (canceling the plant pole) and move the compensation pole to the left until the roots and K_p are acceptable. After a few trials, we find that there is no pole location that satisfies all the requirements! Although moving the pole to the left of $z \cong 0$ will produce acceptable z -plane pole locations, the gain K_p is not sufficiently high to meet the criterion for steady-state error. The only way to raise K_p and to meet the requirements for damping and settling time is to move the zero to the left also.

After some trial and error, we see that

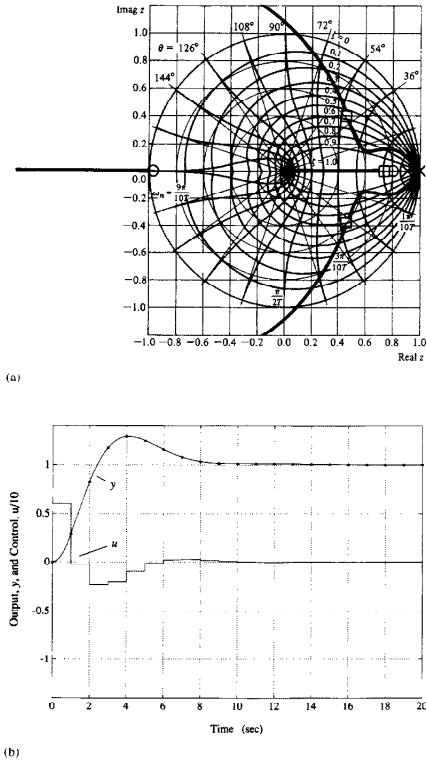
$$D(z) = 6 \frac{z - 0.80}{z - 0.05} \quad (7.19)$$

meets the required z -plane constraints for the complex roots and has a $K_p = 1.26$. The root locus for Eq. (7.19) is shown in Fig. 7.12(a), and the roots corresponding to $K = 6$ are marked by squares. The fact that all requirements seem to be met is encouraging, but there is an additional real root at $z = 0.74$ and a zero at $z = 0.8$, which may degrade the actual response from that expected if it were a second-order system. The actual time history is shown in Fig. 7.12(b). It shows that the overshoot is 29% and the settling time is 15 sec. Therefore, further iteration is required to improve the damping and to prevent the real root from slowing down the response.

A compensation that achieves the desired result is

$$D(z) = 13 \frac{z - 0.88}{z + 0.5} \quad (7.20)$$

Figure 7.12
Antenna design with
 $D(z)$ given by Eq. (7.19);
(a) root locus, (b) step
response



The damping and radius of the complex roots substantially exceed the specified limits, and $K_r = 1.04$. Although the real root is slower than the previous design, it is very close to a zero that attenuates its contribution to the response. The root locus for all K 's is shown in Fig. 7.13(a) and the time response for $K = 13$ in Fig. 7.13(b).

Note that the pole of Eq. (7.20) is on the negative real z -plane axis. In general, placement of poles on the negative real axis should be done with some caution. In this case, however, no adverse effects resulted because all roots were in well-damped locations. As an example of what could happen, consider the compensation

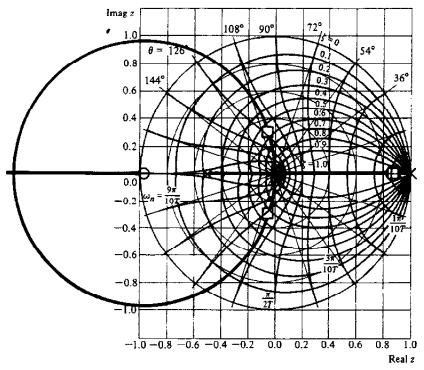
$$D(z) = \frac{9(z - 0.8)}{(z + 0.8)}. \quad (7.21)$$

The root locus versus K and the step response are shown in Fig. 7.14. All roots are real with one root at $z = -0.59$. But this negative real axis root has $\zeta = 0.2$ and represents a damped sinusoid with frequency of $\omega_n/2$. The output has very low overshoot, comes very close to meeting the settling time specification, and has $K_r = 1$; however, the control, u , has large oscillations with a damping and frequency consistent with the negative real root. This indicates that there are "hidden oscillations" or "intersample ripple" in the output that are only apparent by computing the continuous plant output between sample points as is done in Fig. 7.14. The computation of the intersample behavior was carried out by computing it at a much higher sample rate than the digital controller, taking care that the control value was constant throughout the controller sample period. The MATLAB function `ripple`, included in the Digital Control Toolbox, has been written to do these calculations. Note that if only the output at the sample points had been determined, the system would appear to have very good response. This design uses much more control effort than that shown in Fig. 7.13, a fact that is usually very undesirable. So we see that a compensation pole in a lightly damped location on the negative real axis could lead to a poorly damped system pole and undesirable performance. ◆

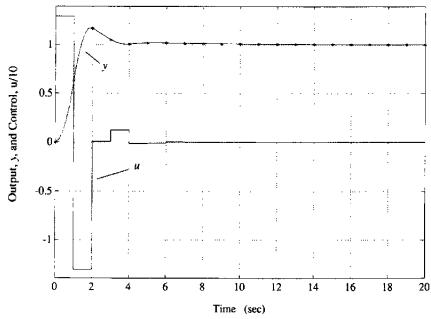
In the design examples to this point, the computed output time histories have assumed that the control, $u(k)$, was available from the computer at the sample instant. However, in a real system this is not always true. In the control implementation example in Table 3.1, we see that some time must pass between the sample of $y(k)$ and the output of $u(k)$ for the computer to calculate the value of $u(k)$. This time delay is called *latency* and usually can be kept to a small fraction of the sample period with good programming and computer design. Its effect on performance can be evaluated precisely using the transform analysis of Section 4.4.2, the state-space analysis of Section 4.3.4, or the frequency response. The designer can usually determine the expected delay and account for it in the design. However, if not taken into account, the results can be serious as can be seen by an analysis using the root locus.

Because a one-cycle delay has a z -transform of z^{-1} , the effect of a full-cycle delay can be analyzed by adding z^{-1} to the numerator of the controller representation. This will result in an additional pole at the origin of the z -plane. If there is a delay of two cycles, two poles will be added to the z -plane origin, and so on.

Figure 7.13
Antenna design with
 $D(z)$ given by Eq. (7.20);
(a) root locus, (b) step
response

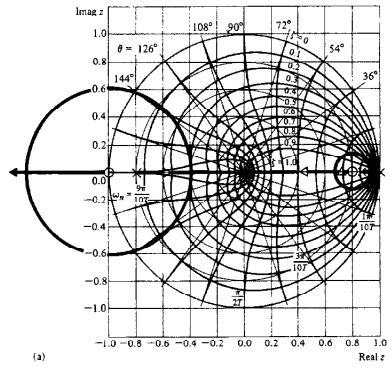


(a)

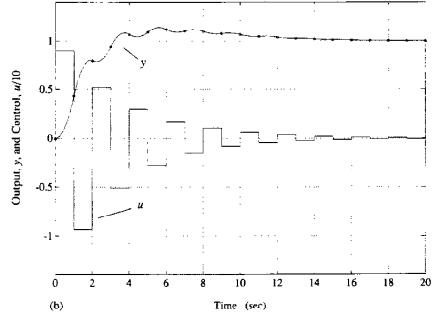


(b)

Figure 7.14
Antenna design with
 $D(z)$ given by Eq. (7.21);
(a) root locus, (b) step
response



(a)



(b)

◆ **Example 7.7 Effect of Unexpected Delay**

Add one cycle delay to the compensation of Eq. (7.21) and plot the resulting root locus and step response.

Solution. The new controller representation is

$$D(z) = 13 \frac{z - 0.88}{z(z + 0.5)}. \quad (7.22)$$

The root locus and time response are shown in Fig. 7.15, which are both substantially changed from the same controller without the delay as shown in Fig. 7.13. The only difference is the new pole at $z = 0$. The severity of the one-cycle delay is due to the fact that this controller is operating at a very slow sample rate (six times the closed loop bandwidth). This sensitivity to delays is one of many reasons why one would prefer to avoid sampling at this slow a rate. ◆

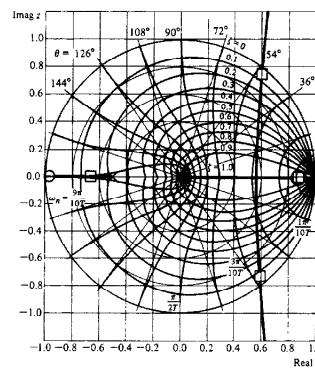
7.4 Frequency Response Methods

The frequency response methods for continuous control system design were developed from the original work of Bode (1945) on feedback-amplifier techniques. Their attractiveness for design of continuous linear feedback systems depends on several ideas.

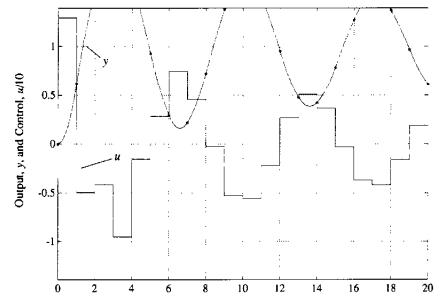
1. The gain and phase curves for a rational transfer function can be easily plotted by hand.
2. If a physical realization of the system is available, the frequency response can be measured experimentally without the necessity of having a mathematical model at all.
3. Nyquist's stability criterion can be applied, and dynamic response specifications can be readily interpreted in terms of gain and phase margins, which are easily seen on the plot of log gain and phase-versus-log frequency.
4. The system error constants, mainly K_p or K_{v} , can be read directly from the low-frequency asymptote of the gain plot.
5. The corrections to the gain and phase curves (and thus the corrections in the gain and phase margins) introduced by a trial pole or zero of a compensator can be quickly and easily computed, using the gain curve alone.
6. The effect of pole, zero, or gain changes of a compensator on the speed of response (which is proportional to the crossover frequency) can be quickly and easily determined using the gain curve alone.

Use of the frequency response in the design of continuous systems has been reviewed in Chapter 2 and the idea of discrete frequency responses has

Figure 7.15
One-cycle-delay antenna design with $D(z)$ given by Eq. (7.22): (a) root locus, (b) step response



(a)



(b)

been introduced in Chapter 4. In order to apply these concepts to the design of digital controls, the basic results on stability and performance must be translated to the discrete domain. The concepts are the same as for continuous systems, but plots of the magnitude and phase of a discrete transfer function, $H(z)$, are accomplished by letting z take on values around the unit circle, $z = e^{j\omega T}$, that is,

$$\begin{aligned} \text{magnitude} &= |H(z)|_{e^{j\omega T}}, \\ \text{phase} &= \angle H(z)|_{e^{j\omega T}}. \end{aligned}$$

◆ Example 7.8 Discrete Bode Plot

Plot the discrete frequency response corresponding to the plant transfer function

$$G(s) = \frac{1}{s(s+1)} \quad (7.23)$$

sampling with a zero order hold at $T = 0.2, 1$, and 2 seconds and compare with the continuous response.

Solution. The discrete transfer functions for the specified sampling periods are computed with c2d.m as

```
sysc = tf([1],[1 1 0]);
sysd1=c2d(sysc,0.2)
sysd2 = c2d(sysc,1)
sysd3=c2d(sysc,2)
```

with transfer functions

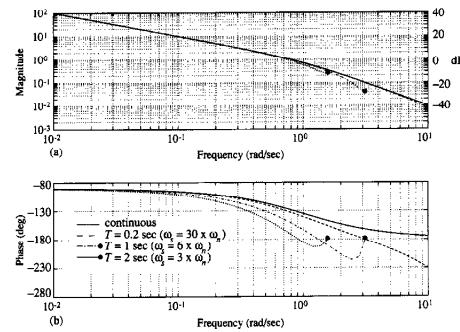
$$\begin{aligned} G_1(z) &= 0.0187 \frac{z + 0.9355}{(z - 1)(z - 0.8187)} \quad \text{for } T = 0.2 \text{ sec} \\ G_2(z) &= 0.368 \frac{z + 0.718}{(z - 1)(z - 0.368)} \quad \text{for } T = 1 \text{ sec} \\ G_3(z) &= 1.135 \frac{z + 0.523}{(z - 1)(z - 0.135)} \quad \text{for } T = 2 \text{ sec}. \end{aligned} \quad (7.24)$$

The frequency responses of Eq. (7.23) and Eq. (7.24) are plotted in Fig. 7.16 using the statement

```
bode(sysc,'-',sysd1,'-.',sysd2,'.',sysd3,'-.-')
```

It is clear that the curves for the discrete systems are nearly coincident with the continuous plot for low frequencies but deviate substantially as the frequency approaches π/T in each case. In particular, the amplitude plots do not approach the simple asymptotes used in the hand-plotting procedures developed by Bode, and his theorem relating the phase to the derivative of the magnitude curve on

Figure 7.16
Frequency responses of continuous and discrete transfer functions



a log-log plot does not apply. The primary effect of sampling is to cause an additional phase lag. Fig. 7.17 shows this additional phase lag by plotting the phase difference, $\Delta\phi$, between the continuous case and the discrete cases. The approximation to the discrete phase lag given by

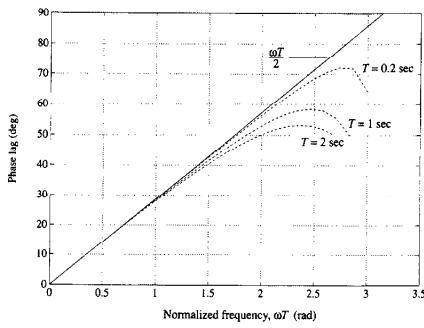
$$\Delta\phi = \frac{\omega T}{2} \quad (7.25)$$

is also shown and demonstrates the accuracy of this approximation for sample rates up to $\omega T = \pi/2$, which corresponds to frequencies up to 1/4 the sample rate. Crossover frequencies (where magnitude = 1) for designs will almost always be lower than 1/4 the sample rate; therefore, one can obtain a good estimate of the phase margin if a sample and hold is introduced into a continuous design by simply subtracting the factor $\omega T/2$ from the phase of the continuous design's phase margin.

The inability to use the standard plotting guidelines detracts from the ease with which a designer can predict the effect of pole and zero changes on the frequency response. Therefore, points 1, 5, and 6 above are less true for discrete frequency-response design using the z -transform than they are for continuous systems and we are more dependent on computer aids in the discrete case. With some care in the interpretations however, points 2, 3, and 4 are essentially unchanged. All these points will be discussed further in this chapter as they pertain to design using the *discrete* frequency response. We begin with the discrete form of the **Nyquist stability criterion** and follow with a discussion of specifications

Nyquist stability criterion

Figure 7.17
Phase lag due to sampling



of performance and stability robustness as expressed in the frequency domain before we introduce the design techniques directly.

7.4.1 Nyquist Stability Criterion

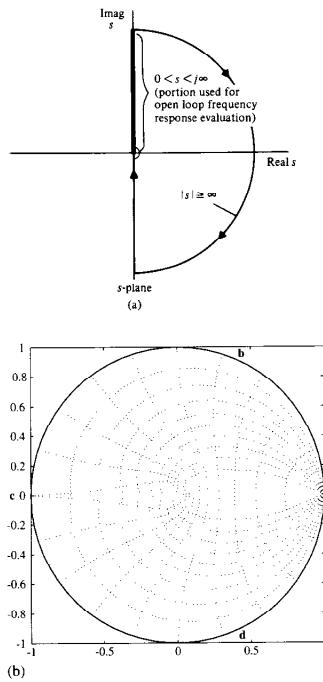
For continuous systems, the Nyquist stability criterion seeks to determine whether there are any zeros of the closed-loop characteristic equation

$$1 + K D(s) G(s) = 0 \quad (7.26)$$

in the right half-plane. The method establishes stability by determining the excess of zeros over poles of the characteristic equation in the right-half plane by plotting $K D(s) G(s)$ for s along the \mathcal{D} contour that encloses the entire right-hand side (unstable region) of the s -plane as sketched in Fig. 7.18(a).

It is assumed that the designer knows the number of (unstable) poles that are inside the contour and from the plot can then determine the number of zeros of Eq. (7.26) in the unstable region that is the same as the number of closed-loop system unstable poles. The entire contour evaluation is fixed by examining $K D(s) G(s)$ over $s = j\omega$ for $0 \leq \omega < \infty$, which is the frequency-response evaluation of the open-loop system. For experimental data, the plot is to be made for $\omega_{low} \leq \omega \leq \omega_{high}$, where ω_{low} is small enough to allow the low-frequency behavior to be decided (usually the gain is very high at ω_{low} and the phase is approaching a fixed multiple of 90°), and where ω_{high} is taken to be high enough that it is known that the magnitude is much less than 1 for all higher

Figure 7.18
Contours used for Nyquist stability criterion:
(a) In the s -plane; (b) in the z -plane



frequencies. Fig. 7.18(a) shows the full \mathcal{D} contour and the portion of the contour for $\omega_{low} \leq \omega \leq \omega_{high}$. The indentation near $\omega = 0$ excludes the (known) poles of $K D G$ at $s = 0$ from the unstable region; the map of this small semicircle is done analytically by letting $s = r^{j\phi}$ for $r \ll 1$, $-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}$.

The specific statement of the Nyquist stability criterion for continuous systems is

$$Z = P + N. \quad (7.27)$$

where

- Z = the number of unstable zeros of Eq. (7.26) (that are unstable closed-loop poles). For stability, $Z = 0$.
- P = the number of unstable (open-loop) poles of $KD(s)G(s)$.
- N = the net number of encirclements of the -1 point for the contour evaluation of $KD(s)G(s)$ in the *same direction* as that taken by s along \mathcal{D} as shown in Fig. 7.18(a). Usually s is taken clockwise around \mathcal{D} and therefore clockwise encirclements are taken as positive.

For the common case of a stable open-loop system ($P = 0$) the closed-loop system is stable if and only if the contour evaluation of $KD(s)G(s)$ does not encircle the -1 point. For unstable open-loop systems, the closed-loop system is stable if and only if the contour evaluation encircles the -1 point counter to the s direction as many times as there are unstable open-loop poles ($N = -P$ in Eq. (7.27)). The proof of this criterion relies on Cauchy's principle of the argument and is given in most introductory textbooks on continuous control systems. The elementary interpretation is based on the following points:

- If we take values of s on a contour in the s -plane that encloses the unstable region, we can plot the corresponding values of the function $1 + KD(s)G(s)$ in an image plane.
- If the s -plane contour encircles a *zero* of $1 + KDG$ in a certain direction, the image contour will encircle the origin in the *same* direction. In the s -plane, the angle of the vector from the zero to s on the contour goes through 360° .
- If the s -plane contour encircles a *pole* of $1 + KDG$, the image contour will encircle the origin in the *opposite direction*. In this case, the s -plane vector angle also goes through 360° but the contribution to the image angle is a negative 360° .
- Thus the *net* number of same-direction encirclements, N , equals the difference $N = Z - P$.²
- The origin of the $1 + KDG$ plane is the same as the point $KDG = -1$ so we can plot KDG and count N as the encirclements of the -1 point just as well.³

² It is much easier to remember same-direction and opposite-direction encirclements than to keep clockwise and counter-clockwise distinguished.

³ When the characteristic equation is written as $1 + KDG$, we can plot only DG and count encirclements of $DG = -\frac{1}{K}$ and thus easily consider the effects of K on stability and stability margins.

- From all of this, Eq. (7.27) follows immediately.

For the discrete case, the ideas are identical: the only difference is that the unstable region of the z -plane is the *outside* of the unit circle and it is awkward to visualize a contour that encloses this region. The problem can be avoided by the simple device of considering the encirclement of the *stable* region and calculating the stability result from that. The characteristic equation of the discrete system is written as

$$1 + KD(z)G(z) = 0. \quad (7.28)$$

and, as in the continuous case, it is assumed that the number, P , of unstable poles of $KD(z)G(z)$, which are also unstable poles of $1 + KD(z)G(z)$, is known and we wish to determine the number, Z , of unstable zeros of Eq. (7.28), which are the unstable closed-loop poles. Examination of Eq. (7.28) reveals that the (possibly unknown) total number of stable plus unstable poles, n , is the same as the total number of zeros of Eq. (7.28). Thus the number of *stable* zeros is $n - Z$ and the number of *stable* poles is $n - P$. Following the mapping result used by Nyquist, the map of $1 + KDG(z)$ for the z contour of Fig. 7.18(b) will encircle the origin N times where

$$\begin{aligned} N &= \{\text{number of stable zeros}\} - \{\text{number of stable poles}\} \\ &= \{n - Z\} - \{n - P\} \\ &= P - Z. \end{aligned}$$

Therefore, the Nyquist stability criterion for discrete systems is

$$Z = P - N. \quad (7.29)$$

In summary, the discrete Nyquist stability criterion is

- Determine the number, P , of unstable poles of KDG .
- Plot $KD(z)G(z)$ for the unit circle, $z = e^{j\omega T}$ and $0 \leq \omega T \leq 2\pi$. This is a counter-clockwise path around the unit circle. Points for the plot can be conveniently taken from a discrete Bode plot of KDG .
- Set N equal to the net number of counter-clockwise (same direction) encirclements of the point -1 on the plot.
- Compute $Z = P - N$. The system is stable if and only if $Z = 0$.

◆ Example 7.9 Nyquist Stability

Evaluate the stability of the unity feedback discrete system with the plant transfer function

$$G(z) = \frac{1}{z(z+1)}. \quad (7.30)$$

with sampling at the rate of 1/2 Hz or $T = 2$ and zero-order hold. The controller is proportional discrete feedback [$K D(z) = K$].

Solution. The discrete transfer function at the specified sampling rate and ZOH is given by sysd3 of Example 7.8 with transfer function

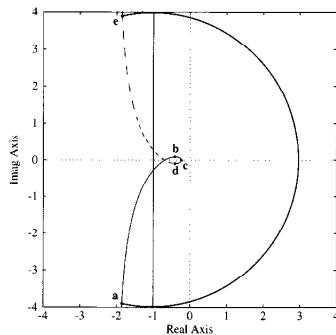
$$G(z) = \frac{1.135(z + 0.523)}{(z - 1)(z - 0.135)} \quad (7.31)$$

and the plot of magnitude and phase of $G(z)$ for $z = e^{j\omega T}$ is included in Fig. 7.16 for $0 \leq \omega T \leq \pi$. Using the data from Fig. 7.16 for $T = 2$, the plot of $K D(z)G(z)$ can be drawn as shown in Fig. 7.19. The plot is marked with corresponding points from Fig. 7.18(b) to facilitate understanding the results. Note that the portion from $a \rightarrow b \rightarrow c$ is directly from Fig. 7.16, and the section from $c \rightarrow d \rightarrow e$ is the same information reflected about the real axis. The large semicircle from $e \rightarrow a$ is the analytically drawn map of the small semicircle about $z = 1$ drawn by letting $(z - 1) = re^{j\phi}$ in Eq. (7.31) for $r \ll 1$ and $-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}$. Because this system is open-loop stable and there are no -1 point encirclements, we conclude that the closed-loop system will be stable as plotted for $K = 1$. Note that all the necessary information to determine stability is contained in the Bode plot information from Fig. 7.16, which determines the portion from $a \rightarrow c$ in Fig. 7.19. Using MATLAB, the plot can be made by the statements

```
nyquist(sysd3)
axis equal
grid
```

The axis statement sets the x and y axes to have equal increments.

Figure 7.19
Nyquist plot of Example 7.9



gain margin
phase margin

7.4.2 Design Specifications in the Frequency Domain

Gain and Phase Margins

The Nyquist plot shows the number of encirclements and thus the stability of the closed-loop system. Gain and phase margins are defined so as to provide a two-point measure of how close the Nyquist plot is to encircling the -1 point, and they are identical to the definitions developed for continuous systems. **Gain margin (GM)** is the factor by which the gain can be increased before causing the system to go unstable, and is usually the inverse of the magnitude of $D(z)G(z)$ when its phase is 180° . The **phase margin (PM)** is the difference between -180° and the phase of $D(z)G(z)$ when its amplitude is 1. The PM is a measure of how much additional phase lag or time delay can be tolerated in the loop before instability results.

Example 7.10 Stability Margins

Consider the open-loop transfer function

$$G(s) = \frac{1}{s(s + 1)^2}$$

with ZOH and sample rate of 5 Hz. The discrete transfer function is given by

$$G(z) = 0.0012 \frac{(z - 3.38)(z - 0.242)}{(z - 1)(z - 0.8187)}$$

What are the gain and phase margins when in a loop with proportional discrete feedback ($D(z) = K = 1$)?

Solution. The discrete Bode plot is given in Fig. 7.20 and the portion of the Nyquist plot representing the frequency response in the vicinity of -1 is plotted in Fig. 7.21. Unlike Example 13.5 which had a very slow sample rate, the higher sample rate here causes the magnitude to be essentially zero at $\omega T = \pi$, and hence the Nyquist plot goes almost to the origin. The plot is very similar to what would result for a continuous controller. Furthermore, just as in the continuous case, there are no -1 point encirclements if $K = 1$ is plotted ($N = 0$), and since there are no unstable poles ($P = 0$), the system will be stable at this gain ($Z = 0$). If the Nyquist plot is multiplied by 1.8, then the plot will go through the -1 point. Thus the gain margin is $GM = 1.8$. For values of $K > 1.8$, the -1 point lies within the contour thus creating two encirclements ($N = 2$) and two unstable closed-loop poles ($Z = 2$). As indicated on the plot, the angle of the plot when the gain is 1 is 18° from the negative axis, so the phase margin is 18° .

phase margin and damping ratio

For continuous systems, it has been observed that the phase margin, PM , is related to the damping ratio, ζ , for a second-order system by the approximate

Figure 7.20
Gain and phase margins
on a Bode plot for
Example 7.10

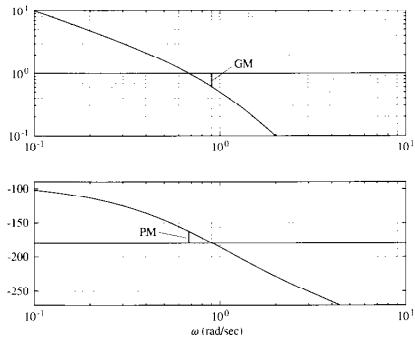
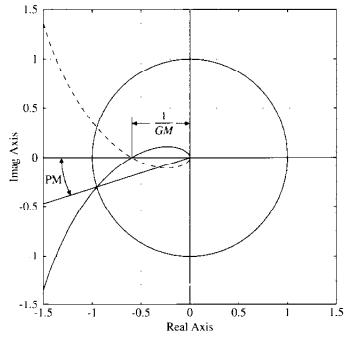
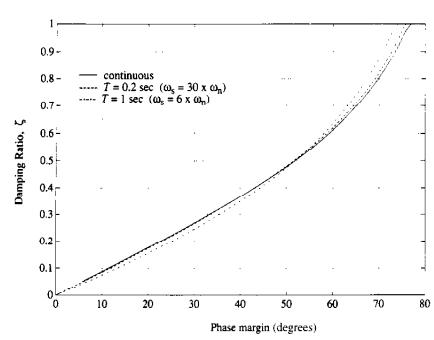


Figure 7.21
Gain and phase margins
on a Nyquist plot for
Example 7.10



relation, $\zeta \cong PM/100$. This relationship is examined in Fig. 7.22 for the continuous case and for discrete systems with two values of the sample rate. Figure 7.22 was generated by evaluating the damping ratio of the closed-loop system

Figure 7.22
Damping ratio of a
second-order system
versus phase margin
(PM)



that resulted when discrete proportional feedback was used with the open-loop system

$$G(s) = \frac{1}{s(s+1)}.$$

A z -transform analysis of this system resulted in z -plane roots that were then transformed back to the s -plane via the inverse of $z = e^{jT}$. The ξ of the resulting s -plane roots are plotted in the figure. As the feedback gain was varied, the damping ratio and phase margin were related as shown in Fig. 7.22. The actual sample rates used in the figure are 1 Hz and 5 Hz, which represent 6 and 30 times the open-loop system pole at 1 rad/sec. The conclusion to be drawn from Fig. 7.22 is that the PM from a discrete z -plane frequency response analysis carries essentially the same implications about the damping ratio of the closed-loop system as it does for continuous systems. For second-order systems without zeros, the relationship between ξ and PM in the figure shows that the approximation of $\xi \cong PM/100$ is equally valid for continuous and discrete systems with reasonably fast sampling. For higher-order systems, the damping of the individual modes needs to be determined using other methods.

Tracking Error in Terms of the Sensitivity Function

The gain and phase margins give useful information about the relative stability of nominal systems but can be very misleading as guides to the design of realistic

sensitivity function

vector gain margin

control problems. A more accurate margin can be given in terms of the sensitivity function. For the unity feedback system drawn in Fig. 7.1, the error is given by

$$E(j\omega) = \frac{1}{1 + DG} R \stackrel{\triangle}{=} S(j\omega) R, \quad (7.32)$$

where we have defined the **sensitivity function** S . In addition to being a factor of the system error, the sensitivity function is also the reciprocal of the distance of the Nyquist curve, DG , from the critical point -1 . A large value for S indicates a Nyquist plot that comes close to the point of instability. The maximum value of $|S|$ is often a more accurate measure of stability margin than either gain or phase margin alone. For example, in Fig. 7.23 a Nyquist plot is sketched that is much closer to instability than either gain or phase margin would indicate. The **vector gain margin** (VGM) is defined as the gain margin in the direction of the worst possible phase. For example, if the Nyquist plot comes closest to -1 on the negative real axis, then the vector margin is the same as the standard gain margin. From the geometry of the Nyquist plot, the distance from the curve to -1 is $1 + DG = \frac{1}{S}$ and with the definition that

$$S_\infty = \max_{\omega} |S|.$$

it follows that the distance of the closest point on the Nyquist curve from -1 is $\frac{1}{S_\infty}$. If the Nyquist curve came this close to the -1 point on the real axis, it would pass through $1 - \frac{1}{S_\infty}$ and by definition, the product $VGM \times (1 - \frac{1}{S_\infty}) = 1$. Therefore we have that

$$VGM = \frac{S_\infty}{S_\infty - 1}. \quad (7.33)$$

The VGM and related geometry are marked on the Nyquist plot in Fig. 7.23.

We can express more complete frequency domain design specifications than any of these margins if we first give frequency descriptions for the external reference and disturbance signals. For example, we have described so far dynamic performance by the transient response to simple steps and ramps. A more realistic description of the actual complex input signals is to represent them as random processes with corresponding frequency spectra. A less sophisticated description which is adequate for our purposes is to assume that the signals can be represented as a sum of sinusoids with frequencies in a specified range. For example, we can usually describe the frequency content of the reference input as a sum of sinusoids with relative amplitudes given by a magnitude function $|R|$ such as that plotted in Fig. 7.24, which represents a signal with sinusoidal components each having about the same amplitude of 150 up to some value ω_i and very small amplitudes for frequencies above that. With this assumption the response specification can be expressed by a statement such as "the magnitude of the system error is to be less than the bound e_b (a value such as 0.01 that defines the required tracking

Figure 7.23
A Nyquist plot showing the vector gain margin

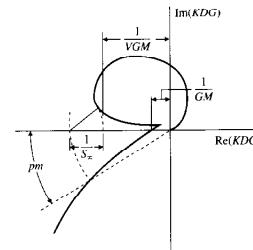
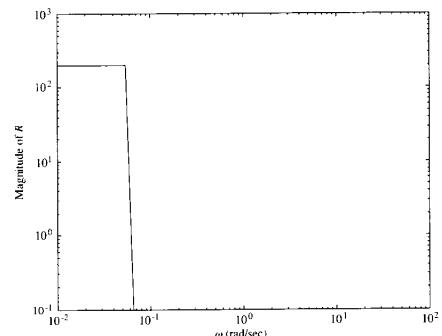


Figure 7.24
Sketch of typical specification of frequency content for reference input tracking



accuracy) for any sinusoid of frequency ω_o and of amplitude given by $|R(j\omega_o)|$." We can now define the size of the error in terms of the sensitivity function and the amplitude of the input. Using Eq. (7.32), the frequency-based error specification can be expressed as $|E| = |S| |R| \leq e_b$. In order to normalize the problem without

defining both the spectrum R and the error bound each time, we define the real function of frequency $W_1(\omega) = |R|/e_b$ and the requirement can be written as

$$|\mathcal{S}| W_1 \leq 1. \quad (7.34)$$

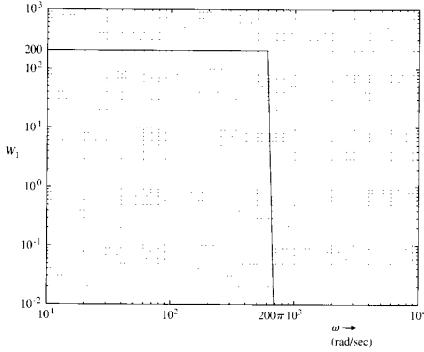
◆ Example 7.11 Performance Bound Function

A unity feedback system is to have an error less than 0.005 for all unity amplitude sinusoids having frequency below 100 Hz. Draw the performance frequency function $W_1(\omega)$ for this design.

Solution. The spectrum, from the problem description, is unity for $0 \leq \omega \leq 200\pi$. Since $e_b = 0.005$, the required function is given by a rectangle of amplitude $1/0.005 = 200$ over the given range. The function is plotted in Fig. 7.25.

The expression in Eq. (7.34) can be translated to the more familiar Bode plot coordinates and given as a requirement on the open-loop gain, DG , by observing

Figure 7.25
Plot of performance frequency function for Example 7.11



that over the frequency range when errors are small the loop gain is large. In that case $|\mathcal{S}| \approx \frac{1}{|DG|}$ and the requirement is approximately

$$\frac{W_1}{|DG|} \leq 1 \quad (7.35)$$

$$|DG| \geq W_1.$$

Stability Robustness in Terms of the Sensitivity Function

In addition to the requirements on dynamic performance, the designer is usually required to design for stability robustness. The models used for design are almost always only approximations to the real system. Many small effects are omitted, such as slight flexibility in structural members or parasitic electrical elements in an electronic circuit. Usually these effects influence the transfer function at frequencies above the control bandwidth and a nominal transfer function, G_o , is used for the design. However, while the design is done for the nominal plant transfer function, the actual system is expected to be stable for an entire class of transfer functions that represent the range of changes that are expected to be faced as all elements are included and as changes due to temperature, age, and other environmental factors vary the plant dynamics from the nominal case. A realistic way to express plant uncertainty is to describe the plant transfer function as having a multiplicative uncertainty as

$$G(j\omega) = G_o(j\omega)[1 + w_2(\omega)\Delta(j\omega)]. \quad (7.36)$$

In Eq. (7.36), $G_o(j\omega)$ is the nominal plant transfer function and the real function, $w_2(\omega)$, is a magnitude function that expresses the size of changes as a function of frequency that the transfer function is expected to experience and is known to be less than some upper bound $W_2(\omega)$. The value of the bound W_2 is almost always very small for low frequencies (we know the model very well there) and increases substantially as we go to high frequencies where parasitic parameters come into play and unmodeled structural flexibility is common.

◆ Example 7.12 Model Uncertainty

A magnetic memory read/write head assembly can be well modelled at low frequencies as

$$G_o(s) = \frac{K}{s}. \quad (7.37)$$

However, the arm supporting the read/write head has some lightly damped flexibility with uncertain resonant frequency. With scaling to place the resonant frequency at ω_o , and damping B , the more accurate model is represented as

$$G(s) = \frac{K}{s^2} \frac{\frac{B}{\omega_o} + 1}{\left(\frac{s}{\omega_o}\right)^2 + B \frac{s}{\omega_o} + 1} \quad (7.38)$$

Compute the model uncertainty function for this case.

Solution. The model transfer function given by Eq. (7.38) can be written as

$$G(s) = \frac{K}{s^2} \left[1 + \frac{-\left(\frac{s}{\omega_o}\right)^2}{\left(\frac{s}{\omega_o}\right)^2 + B \frac{s}{\omega_o} + 1} \right]. \quad (7.39)$$

Comparing Eq. (7.39) with Eq. (7.36), the model uncertainty function is given by

$$w_1(\omega) = \left| \frac{-\left(\frac{s}{\omega_o}\right)^2}{\left(\frac{s}{\omega_o}\right)^2 + B \frac{s}{\omega_o} + 1} \right|_{s=j\omega}. \quad (7.40)$$

A plot of this function is given in Fig. 7.26 for $\omega_o = 1$ and $B = .03$.

In general, a model uncertainty bound is small for low frequencies and large for higher frequencies. A typical shape is sketched in Fig. 7.27. The complex function, $\Delta(j\omega)$, represents the uncertainty in phase and is restricted only by the constraint

$$|\Delta(j\omega)| \leq 1. \quad (7.41)$$

Stability robustness requires that we construct a control design for $G_c(s)$ which will result in a stable system for any transfer function described by Eq. (7.36). To derive the requirement, we begin with the assumption that the nominal design has been done and is stable so that the Nyquist plot of DG_o satisfies the Nyquist stability criterion. In this case, the equation $1 + D(j\omega)G_o(j\omega) = 0$ is never satisfied for any real frequency. If the system is to have stability robustness, the characteristic equation using the uncertain plant as described by Eq. (7.36)

Figure 7.26
Plot of model uncertainty function for disk drive read/write head assembly

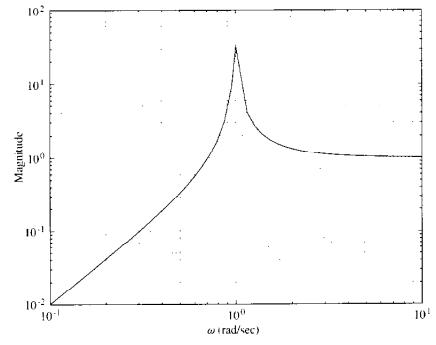
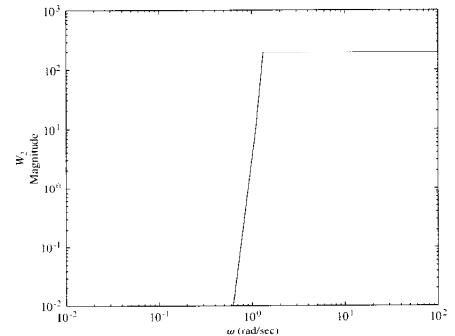


Figure 7.27
A plot of a typical plant uncertainty frequency function



complementary sensitivity function

must not go to zero for any real frequency for any value of either W_2 or Δ . The requirement can be written as a function of $j\omega$ in the form

$$1 + DG \neq 0, \quad (7.42)$$

$$1 + DG_o[1 + w_2\Delta] \neq 0, \quad (7.43)$$

$$(1 + DG_o)\left(1 + \frac{DG_o}{1 + DG_o}w_2\Delta\right) \neq 0,$$

$$(1 + DG_o)(1 + Tw_2\Delta) \neq 0, \quad (7.44)$$

where the **complementary sensitivity function** is defined as $T(j\omega) \triangleq DG_o / (1 + DG_o) - 1 - S$. Because the nominal system is stable, the first term in Eq. (7.42), $(1 + D(j\omega)G_o(j\omega))$, is not zero for any ω . Thus, if Eq. (7.42) is not to be zero for any frequency, any $w_2 \leq W_2$, or for any phase function Δ , then it is necessary and sufficient that

$$|Tw_2\Delta| < 1,$$

$$|T| |w_2| |\Delta| < 1,$$

which reduces to

$$|T| W_2 < 1, \quad (7.45)$$

making use of Eq. (7.41) and the fact that w_2 is bounded by W_2 . As with the performance specification, for single-input-single-output unity feedback systems this requirement can be approximated by a more convenient form. Over the range of high frequencies where there is significant model uncertainty and W_2 is non-negligible, DG_o is small. Therefore we can approximate $T \approx DG_o$ and the constraint becomes

$$|DG_o| W_2 < 1$$

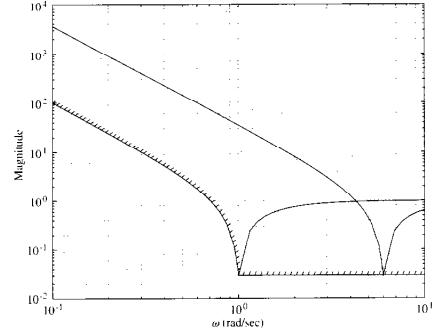
$$|DG_o| < \frac{1}{W_2}. \quad (7.46)$$

◆ Example 7.13 Stability Robustness Function

The uncertainty in the model of a disk read/write head assembly is given in Example 7.12. Suppose it is known that the parameter B is restricted to the range $.03 \leq B \leq .3$ and that the resonant frequency ω_r is known to be no less than 1.0. Plot the stability robustness bound $1/W_2$ for this problem.

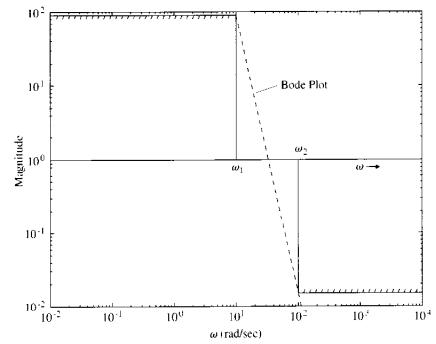
Solution. The function $1/w_2(\omega)$ is plotted for $B = .03$ and $\omega = 1$ and $\omega = 6$ in Fig. 7.28 using bode. It is clear that if the resonant frequency can take on any value greater than 1.0, then the bound $1/W_2$ needs to be extended at the value .03 for all frequencies greater than 1.0. The boundary line is marked with hatching in Fig. 7.28.

Figure 7.28
Plot of the stability robustness frequency function for disk read/write head assembly



In practice, the magnitude of the loop gain is plotted on log-log coordinates, and the constraints of Eq. (7.35) and Eq. (7.46) are included on the same plot. A

Figure 7.29
Typical design limitations as displayed on a Bode magnitude plot



typical sketch is drawn in Fig. 7.29. The designer is expected to construct a loop gain that will stay above W_1 for frequencies below ω_1 , cross over the magnitude-of-1 line ($\log(|DG|) = 0$) in the range $\omega_1 \leq \omega \leq \omega_s$, and stay below $1/W_2$ for frequencies above ω_s . We have developed the design constraints Eq. (7.35) and Eq. (7.46) in terms of $f\omega$ as for continuous systems. The algebra and the equations are the same for the discrete case; one need only substitute the discrete transfer functions for the continuous ones and use the variable $e^{j\omega T}$.

Limitations on Design: Continuous Case

One of the major contributions of Bode was to derive important limitations on transfer functions that set limits on achievable design specifications. For example, we would like to have the system error kept small for the widest possible range of frequencies and yet to have a system that is stable in the presence of uncertainty in the plant transfer function. In terms of the plot in Fig. 7.29, we want W_1 and W_2 to be large in their respective frequency ranges and for ω_1 to be close to ω_s . Thus the loop gain is expected to plunge with a large negative slope from being greater than W_1 to being less than $1/W_2$ in a short span, while maintaining stability which can be expressed as having a good phase margin. Bode showed that this is *impossible* with a linear controller by showing that the minimum possible phase is determined by an integral depending on the slope of the magnitude curve. A common form of the formula for phase is

$$\phi(\omega_o) = \frac{1}{\pi} \int_{-\infty}^{-\infty} \left(\frac{dM}{du} \right) \ln(\coth \left| \frac{u}{2} \right|) du, \quad (7.47)$$

where $M = \ln(|DG|)$ and $u = \ln(\frac{\omega}{\omega_o})$, and thus $\frac{dM}{du}$ is the magnitude slope on the log-log (Bode) plot. The weighting function in Eq. (7.47) is concentrated near ω_o , and if the slope is constant for a substantial range around ω_o , then the formula can be approximated by

$$\phi(\omega_o) \approx \frac{\pi}{2} \left. \frac{dM}{du} \right|_{u=0}.$$

If, for example, the phase is to be kept above -150° to maintain a 30° phase margin, then the magnitude slope is estimated to be

$$\frac{dM}{du} \approx \frac{2}{\pi} \left(-150 - \frac{\pi}{180} \right) \approx -1.667$$

in the neighborhood of crossover. If we try to make the average slope steeper (more negative) than this near crossover, we will lose the phase margin. From this condition there developed the design rule:

- The asymptotes of the Bode plot magnitude, which are restricted to be integral values for rational functions, should be made to crossover the magnitude 1

Bode's gain-phase formula

Bode's gain-phase integral

sensitivity integral

constraint on non-minimum phase systems

line at a slope of -1 over a frequency range of about one decade around the cross-over frequency.

Modifications to this rule need of course to be made in particular cases, but the limitation expressed by Eq. (7.47) is a hard limit that cannot be avoided.

In Freudenberg and Loize (1985) an extension to another of Bode's relations was derived. This is a constraint on the integral of the sensitivity function dependent on the presence of open-loop right-half plane poles. Suppose the loop gain DG , has n_p poles, p_i , in the right-half plane and "rolls off" at high frequencies at a slope faster than -1 . For rational functions, this means that there is an excess of at least two more total poles than zeros. Then it can be shown that

$$\int_0^{\infty} \ln(|S|) d\omega = \pi \sum_{i=1}^{n_p} \operatorname{Re}(p_i). \quad (7.48)$$

If there are no right-half plane poles, then the integral is zero. This means that if we make the log of the sensitivity function very negative over some frequency band to reduce errors in that band, then *of necessity* $\ln|S|$ will be positive over another part of the band and errors will be amplified there. If there are unstable poles, the situation is worse because the positive area where sensitivity magnifies the error must exceed the negative area where the error is reduced by the feedback. There are also consequences if DG , has any *zeros* in the right-half plane. If the open-loop system has no zeros in the right-half plane, then it is in principle possible to keep the magnitude of the sensitivity small by spreading the sensitivity increase over all positive frequencies but such a design requires an excessive bandwidth and is rarely practical. If a specific bandwidth is imposed, then the sensitivity function is constrained to take on a finite, possibly large, positive value at some point below the bandwidth and a large value of $|S|_{\infty}$ leads to a small VGM and generally an unsatisfactory design.

An alternative to Eq. (7.48) is also true if there is a (non-minimum-phase) zero of DG , in the right-half plane. Suppose the zero is located at $z_o = \sigma_o + j\omega_o$, where $\sigma_o > 0$. Again we assume there are n_p right-half plane poles at locations p_i with conjugate values \bar{p}_i . Now the condition can be expressed as a two-sided weighted integral

$$\int_{-\infty}^{\infty} \ln(|S|) \frac{\sigma_o}{\sigma_o + (\omega - \omega_o)^2} d\omega = \pi \sum_{i=1}^{n_p} \ln \left| \frac{\bar{p}_i + z_o}{p_i - z_o} \right|. \quad (7.49)$$

In this case, we do not have the "roll-off" restriction and there is no possibility of spreading the positive area over all high frequencies because the weighting function goes to zero with frequency. The important point about this integral is that if the non-minimum phase zero is close to a right-half plane pole, the right side of the integral can be very large and the excess of positive area is required to be correspondingly large. Based on this result, *one expects especially great*

difficulty meeting specifications on sensitivity with a system having right-half plane poles and zeros close together.

Limitations on Design: Discrete Case

In the discrete case, the relation between gain slope and phase does not hold although it is approximately true for frequencies well below the Nyquist frequency. The situation is illustrated by the Bode plot in Fig. 7.30 for the plant $G = 1/s^2$. Notice that the phase is always slightly more negative than the -180° one would get for this plant in the continuous case and deviates more as we approach the Nyquist limit at π/T . The effect is approximated by the delay of $T/2$ due to the sample and ZOH. From this example, one suspects that the restriction on phase due to gain slope is *more severe* in the discrete case than in the continuous case.

In Sung and Hara (1988) the discrete versions of these design limitations are derived. We consider the single-loop unity feedback structure and define the sensitivity function as $S = \frac{1}{1+DG}$ as before. We also assume that the open-loop transfer function $D(z)G(z)$ has n_p poles outside the unit circle at $\zeta_i = r_i e^{j\phi_i}$, $r_i > 1$. It can be shown that

$$\int_0^\pi \ln(|S(e^{j\phi})|) d\phi = \pi \sum_{i=1}^{n_p} \ln(r_i). \quad (7.50)$$

The implications of Eq. (7.50) are the same as in the continuous case except for the fact that the integral in Eq. (7.50) is over a *finite* limit. If we require that

discrete sensitivity integral

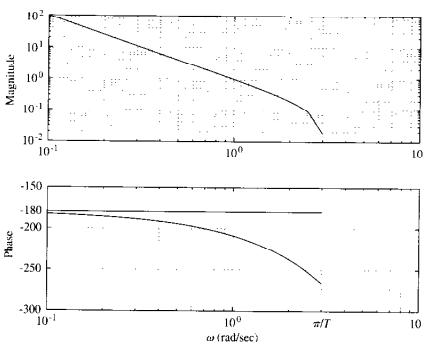


Figure 7.30
Discrete Bode plot of
 $1/s^2$ plant with
zero-order hold

sensitivity is small (negative log) over a given range of frequencies, there is only a finite frequency range over which we can spread the required sensitivity gain or "positive log" area. Again, unstable poles make the situation worse and the effect increases if the poles are located far from the unit circle. We can illustrate the implications of Eq. (7.50) by two simple examples. Consider a stable generic model of a chemical process and an unstable model of a magnetic levitation system. The two transfer functions are

$$G_1(s) = \frac{1}{(s+1)^2} \quad \text{chemical process},$$

$$G_2(s) = \frac{1}{s^2 - 1} \quad \text{magnetic levitation}.$$

In each case we will include a zero-order hold and sample with $T = 0.2$ and in each case the controller transfer function is $D = 15 \frac{s+0.5}{s-0.5}$, selected to give approximately the same rise time and bandwidth. The step responses are shown in Fig. 7.31(a) and the sensitivity plots are shown in Fig. 7.31(b). Notice the substantially larger value of the sensitivity for the unstable plant compared to that of the stable one. The vector gain margin for G_1 is 5.4 while that for G_2 is 2.25, less than half as much. To plot the sensitivity magnitude, it is necessary to obtain a system having the transfer function of S . This can be done using the feedback function if a representation of the open loop system DG is given as, for example, the open loop discrete system $sys0d$. The expression $sys1 = \text{feedback}(sys1, sys2)$ generates the loop with forward system $sys1$ and feedback system $sys2$. For sensitivity, DG is the feedback system, and we need to construct the dummy gain of 1 for the forward system. This can be done with the statement $sysf = ss(0, 0, 0, 1, Ts)$. Finally, the sensitivity is given by $sens = \text{feedback}(sysf, sys0d)$. The plot is given by the statements

```
[mag,ph,w]=bode(sens);
semilogw(mag);
grid.
```

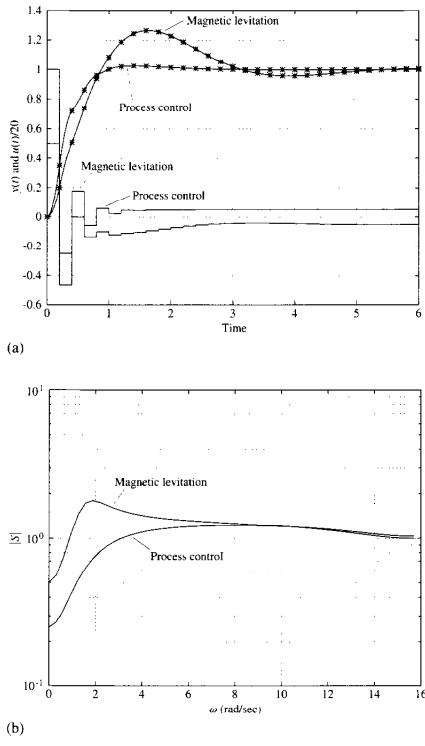
The weighted integral of the discrete sensitivity function is similar to that of the continuous case. We assume again that the system has n_p poles outside the unit circle at $\alpha_i = r_i e^{j\phi_i}$, $r_i > 1$, and conjugate $\bar{\alpha}_i$, and also has a zero outside the unit circle at $\beta_o = r_o e^{j\phi_o}$, $r_o > 1$. Then

$$\int_{-\pi}^{\pi} \ln(|S(e^{j\phi})|) \frac{r_o^2 - 1}{r_o^2 - 2r_o \cos(\phi - \phi_o) + 1} d\phi = 2\pi \sum_{i=1}^{n_p} \ln \left| \frac{1 - \bar{\alpha}_i \beta_o}{\beta_o - \alpha_i} \right|. \quad (7.51)$$

The main consequence of this constraint is that it expresses a limitation imposed by the non-minimum phase zero on the sensitivity function. The constraint is especially severe if there is a non-minimum phase zero near an unstable pole ($\beta_o \approx \alpha_i$).

weighted integral of the
discrete sensitivity
function

Figure 7.31
Comparisons of a stable process control with an unstable magnetic levitation: (a) step responses (b) sensitivity plots



7.4.3 Low Frequency Gains and Error Coefficients

The steady-state error constants for polynomial inputs for discrete systems were established in Section 7.2 and are given by

$$K_p = \lim_{z \rightarrow 1} D(z)G(z)$$

for a Type 0 system, and by

$$K_v = \lim_{z \rightarrow 1} \frac{(z-1)D(z)G(z)}{Tz}, \quad (7.52)$$

for a Type 1 system. In the frequency domain, for a Type 0 system, the procedure is identical to the continuous case. Since $z = e^{j\omega T}$, $z \rightarrow 1$ implies that $\omega T \rightarrow 0$, and the magnitude frequency-response plot will show a constant value on the low-frequency asymptote which is equal to K_p . For a Type 1 system, the procedure is again identical to the continuous case in that the magnitude of $D(z)G(z)$ at $\omega = 1$ on the low-frequency asymptote is equal to K_v . This can be seen from Eq. (7.52) if we note that for $\omega T \rightarrow 0$, $e^{j\omega T} \cong 1 + j\omega T$. Therefore

$$\lim_{z \rightarrow 1} \frac{(z-1)}{Tz} = \lim_{j\omega \rightarrow 0} \omega,$$

thus establishing the fact that evaluation of the low-frequency asymptote of $D(z)G(z)$ at $\omega = 1$ yields K_v . This fact is most easily used if the frequency-response magnitude is plotted versus ω in units of rad/sec so that $\omega = 1$ rad/sec is readily found. If the magnitude is plotted versus ω in units of Hz or versus ωT , one would need to perform a calculation to find the $\omega = 1$ rad/sec point. However, the error constants could be calculated directly with good software tools; therefore the issues in their calculation are of passing interest only. But no matter how the constants are found, the fact remains for discrete and continuous frequency response alike, the higher the magnitude curve at low frequency, the lower the steady-state errors.

◆ Example 7.14 Finding Velocity Constant on a Bode Plot

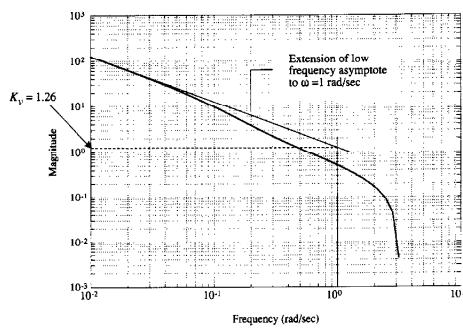
Use the discrete Bode plot to determine the K_v for the antenna system of Example 7.6 with the compensation given by Eq. (7.19).

Solution. The open-loop discrete transfer function is

$$G(z)D(z) = (0.0484) \frac{z + 0.9672}{(z - 1)(z - 0.9048)} (6) \frac{z - 0.80}{z - 0.05},$$

which yields the magnitude versus frequency in Fig. 7.32. Evaluation of the magnitude of the low-frequency asymptote at $\omega = 1$ indicates that $K_v = 1.26$. Also note in the figure that the

Figure 7.32
Determination of K_v from frequency response



extension of the low-frequency asymptote reaches crossover at $\omega = 1.26$, thus indicating also that $K_v = 1.26$ since this Type 1 system has a low-frequency slope of -1 .

7.4.4 Compensator Design

The amplitude and phase curves can be used to determine the stability margins based on the Nyquist stability criterion for either continuous or discrete systems. In the continuous case with minimum-phase transfer functions, Bode showed that the phase is uniquely determined by an integral of the slope of the magnitude curve on a log-log plot as expressed by Eq. (7.47). If the function is rational, these slopes are readily and adequately approximated by constants! As a consequence the amplitude curve must cross unity gain (zero on a log scale) at a slope of -1 for a reasonable phase margin. The ability to predict stability from the amplitude curve alone in minimum phase systems is an important contributor to the ease with which designers can evaluate changes in compensator parameters in those cases.

Bode's relationship between the amplitude and phase curve is lost for discrete systems because the variable z takes on values around the unit circle in contrast to s traversing the imaginary axis as in continuous systems. Figure 7.16 illustrates the degree to which the relationship is lost and indicates that the error would be small for frequencies lower than 1/20th of the sample frequency. However, it is typically

necessary to determine both magnitude and phase for discrete z -plane systems and not depend on magnitude alone for an accurate assessment of the stability. In carrying out direct digital designs, some intuition from continuous design can be used if the z -plane poles and zeros on the real axis are measured by their distance from $+1$. For example, the equivalent idea in the z -plane for the "breakpoint" in Bode's hand-plotting rules is that the magnitude will change slope at a frequency when ωT , the angular position on the unit circle in radians, has the same value as the fractional distance of the singularity on the real axis to $z = +1$. For example, a pole at $z = 0.9$ will produce a slope change at $\omega T = 0.1$ rad. This equivalence is very accurate for low angular values ($\omega T \leq 0.1$ rad, i.e., sampling at more than 60 times the frequency) and is a reasonable approximation for angular values less than 0.8 rad (i.e., sampling at more than 8 times the frequency). In order to arrive at trial compensations with potential for better *PM*, *GM*, steady-state errors, or crossover frequency, it is useful to understand how a pole or zero placement will affect the magnitude and phase curves. Because of the equivalence of the break-point concept between the continuous and discrete cases, this is accomplished for discrete systems using the ideas from the continuous Bode hand-plotting techniques, keeping in mind that their fidelity degrades as frequency approaches the Nyquist frequency. It is easiest to select compensator break points if the frequency-response magnitude and phase are plotted versus ωT so that the correspondence between those curves and the location of the compensation parameters is retained.

◆ Example 7.15 Design of the Antenna Servo Control

Design a discrete controller for the antenna system with $T = 1$ using the frequency response. The specifications are as before: overshoot less than 16%, settling time less than 10 sec and $K_v \geq 1$.

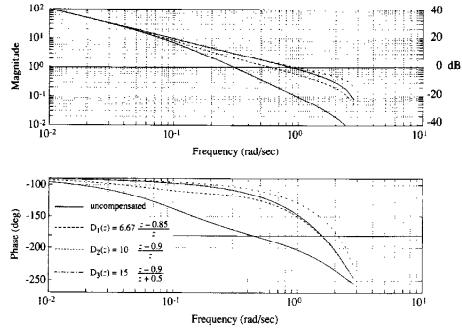
Solution. The system transfer function is

$$G(z) = 0.0484 \frac{z + 0.9672}{(z - 1)(z - 0.9048)} \quad (7.53)$$

The magnitude and phase of the uncompensated system $|G(z)|$ shown in Fig. 7.35 indicate that with a compensator gain of $K = 1$ the system has a *PM* of 8° and a gain crossover frequency (ω_c) of 0.3 rad/sec. The 16% overshoot requirement translates into $\zeta \geq 0.5$, which translates in turn into the requirement that the *PM* be $\geq 30^\circ$ from Fig. 7.24. The specification for settling time translates into the requirement that $\omega_n \geq 0.92$.

Because the gain of $(z - 1)G(z)$ at $z = 1$ is 1, and $T = 1$, the compensated system will also have $K_v = 1$ provided the DC gain of $D(z) = 1$. In terms of the frequency response, this means that the extension of the low-frequency-magnitude asymptote should pass through the value 1 at $\omega = 1$ for the uncompensated case (in Fig. 7.33), and the gain of this low-frequency asymptote should not be decreased with any candidate compensation. To maintain an acceptable K_v , we will evaluate only $D(z)$'s with a DC gain of 1. The uncompensated

Figure 7.33
Frequency responses with D_1 , D_2 , and D_3 for Example 7.15



system's $PM = 8^\circ$ indicates poor damping, and the ω_{eg} of 0.3 rad/sec indicates that it will be too slow. Just as for continuous systems, ω_{eg} occurs approximately at the system bandwidth and dominant natural frequency; therefore, we should try to change the design so that it has a ω_{eg} of about 0.9 rad/sec in order to meet the $t_s \leq 10$ sec. Once we find a compensation that meets the guidelines of $PM = 50^\circ$ and $\omega_{eg} = 0.9$ rad/sec, we will need to check whether the settling time and overshoot specifications are actually met, because the design guidelines followed are only approximate.

Figure 7.33 shows several attempts to produce a design. The breakpoint of the first attempt ($D_1(z)$ in Fig. 7.33) was at 0.15 rad/sec⁴ and did not increase the slope of the magnitude curve at a low enough frequency to bring about the desired ω_{eg} . This was remedied in $D_3(z)$, where the breakpoint was lowered to 0.1 rad/sec (zero at $z = 0.9$) causing a ω_{eg} of 0.9 rad/sec, but the resulting $PM = 40^\circ$ was still lower than desired. By moving the compensator pole out to $z = -0.5$ in $D_3(z)$, we had very little effect on the ω_{eg} but achieved an increase in the PM to 50°. Because both goals are met, $D_3(z)$ has a reasonable chance to meet the specifications; in fact, the calculation of a time history of the system response to a step input shows that the t_s is 7 sec, but, alas, the overshoot is 27%. The guidelines were not successful in meeting the specifications because the system is third order with a zero, whereas the rules were derived assuming a second-order system without a zero.

The necessary revisions to our design guidelines are clear: we want more than a 50° PM and do not require a 0.9 rad/sec ω_{eg} . Figure 7.34 shows the system frequency response using $D_3(z)$ along with two revisions of $D_3(z)$ that satisfy our revised goals. $D_3(z)$ has a 60° PM and a 0.6 rad/sec ω_{eg} , and $D_4(z)$ has a 58° PM and a 0.8 rad/sec ω_{eg} . The time history of the system

4. The zero at $z = 0.85$ translates into a 0.15 rad/sec breakpoint only because the sample period, T , is 1 sec. For $T = 0.1$ sec, a zero at $z = 0.85$ would translate into a 1.5 rad/sec breakpoint, etc.

Figure 7.34
Bode plots of designs with controllers D_3 , D_4 , and D_5 for Example 7.15

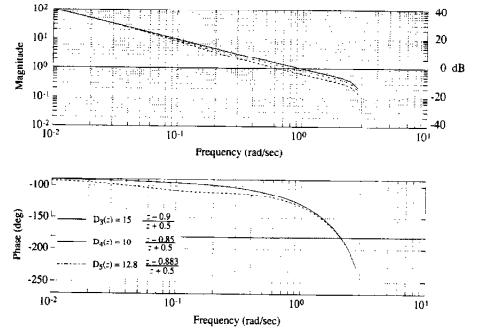
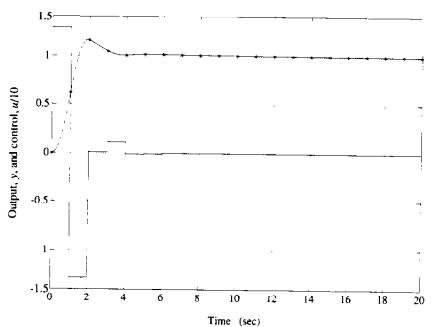
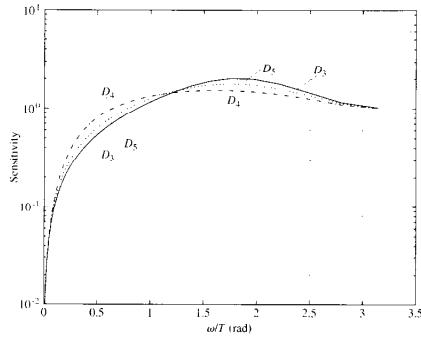


Figure 7.35
Step response of the system with controller D_5



response to a step using $D_5(z)$ in Fig. 7.35 shows that it exactly meets the requirements for 16% overshoot and $t_s = 10$ sec. Furthermore, the design of the system imposed the constraint that $K_v = 1$ and the design is complete.

Figure 7.36
Sensitivity plots of designs with controllers D_3 , D_4 , and D_5 for Example 7.15



It is interesting to look at the sensitivity functions for these designs, plotted in Fig. 7.36 as log of the sensitivity versus a linear frequency scale to illustrate the balance between positive and negative areas for such plots for stable systems. On this plot, one can see that controller D_5 results in the highest bandwidth and also the highest maximum of the sensitivity or lowest vector gain margin. Controller D_4 has improved robustness (lower maximum of the sensitivity) but also lower bandwidth. Finally, the design given by D_3 splits the difference and meets each specification.

7.5 Direct Design Method of Ragazzini

Much of the style of the transform design techniques we have been discussing in this chapter grew out of the limitations of technology that was available for realization of continuous-time compensators with pneumatic components or electric networks and amplifiers. In particular, many constraints were imposed in order to assure the realization of electric compensator networks $D(s)$ consisting only of resistors and capacitors.⁵ With controllers realized by digital computer, such limitations on realization are, of course, not relevant and one can ignore these particular constraints. An alternative design method that ignores constraints of

⁵ In the book by Truxal (1955), where much of this theory is collected at about the height of its first stage of development, a chapter is devoted to RC network synthesis.

technology has been found to be useful in adaptive controls. Suppose we are given a discrete transfer function $G(z)$ of the plant (plus hold) and a desired transfer function $H(z)$ between reference R and output Y . The structure is assumed to be a unity feedback system and the design is to select the computer transfer function $D(z)$ to realize $H(z)$. The overall transfer function is given by the formula

$$H(z) = \frac{DG}{1 + DG},$$

from which we get the direct design formula

$$D(z) = \frac{1}{G(z)} \frac{H(z)}{1 - H(z)}. \quad (7.54)$$

From Eq. (7.54) we can see that this design calls for a $D(z)$ that will cancel the plant effects and add whatever is necessary to give the desired result. The problem is to discover and implement constraints on $H(z)$ so that we do not ask for the impossible.

First, the design must be causal. From z -transform theory we know that if $D(z)$ is to be causal, then as $z \rightarrow \infty$ its transfer function is bounded and does not have a pole at infinity. Looking at Eq. (7.54), we see that if $G(z)$ were to have a zero at infinity, then $D(z)$ would have a pole there unless we require an $H(z)$ that cancels it. Thus we have the constraint that for $D(z)$ to be causal

$H(z)$ must have a zero at infinity of the same order
as the zero of $G(z)$ at infinity. (7.55)

This requirement has an elementary interpretation in the time domain: A zero of order k at infinity in $G(z)$ corresponds to a delay of k samples in the pulse response of the plant. The causality requirement on $H(z)$ is that the closed-loop system must have at least as long a delay as the plant.

Considerations of stability add a second constraint. The roots of the characteristic equation of the closed-loop system are the roots of the equation

$$1 + D(z)G(z) = 0. \quad (7.56)$$

We can express Eq. (7.56) as a polynomial if D and G are rational and we identify $D = c(z)/d(z)$ and $G = b(z)/a(z)$ where a, b, c , and d are polynomials. Then the characteristic polynomial is

$$ad + bc = 0. \quad (7.57)$$

Now suppose there is a common factor in DG , as would result if $D(z)$ were called upon to cancel a pole or zero of $G(z)$. Let this factor be $z - \alpha$ and suppose it is a pole of $G(z)$, so we can write $a(z) = (z - \alpha)\tilde{a}(z)$, and to cancel it we have $c(z) = (z - \alpha)\tilde{c}(z)$. Then Eq. (7.57) becomes

$$\begin{aligned} (z - \alpha)\tilde{a}(z)d(z) + b(z)(z - \alpha)\tilde{c}(z) &= 0. \\ (z - \alpha)[\tilde{a}d + \tilde{b}\tilde{c}] &= 0. \end{aligned} \quad (7.58)$$

In other words—perhaps it was obvious from the start—a common factor *remains a factor of the characteristic polynomial*. If this factor is outside the unit circle, the system is unstable! How do we avoid such cancellation? Considering again Eq. (7.54), we see that if $D(z)$ is not to cancel a pole of $G(z)$, then that factor of $a(z)$ must also be a factor of $1 - H(z)$. Likewise, if $D(z)$ is not to cancel a zero of $G(z)$, such zeros must be factors of $H(z)$. Thus we write the constraints⁶

$$1 - H(z) \text{ must contain as zeros all the poles of } G(z) \text{ that are outside the unit circle,} \quad (7.59)$$

$$H(z) \text{ must contain as zeros all the zeros of } G(z) \text{ that are outside the unit circle.} \quad (7.60)$$

Consider finally the constraint of steady-state accuracy. Because $H(z)$ is the overall transfer function, the error transform is given by

$$E(z) = R(z)(1 - H(z)). \quad (7.61)$$

Thus if the system is to be Type 1 with velocity constant K_v , we must have zero steady-state error to a step and $1/K_v$ error to a unit ramp. The first requirement is

$$e(\infty) = \lim_{z \rightarrow 1} (z-1) \frac{Tz}{z-1} [1 - H(z)] = 0, \quad (7.62)$$

which implies

$$H(1) = 1. \quad (7.63)$$

The velocity constant requirement is that

$$e(\infty) = \lim_{z \rightarrow 1} (z-1) \frac{Tz}{(z-1)^2} [1 - H(z)] = \frac{1}{K_v}. \quad (7.64)$$

From Eq. (7.63) we know that $1 - H(z)$ is zero at $z = 1$, so that to evaluate the limit in Eq. (7.64), it is necessary to use L'Hôpital's rule with the result

$$-T \frac{dH}{dz} \Big|_{z=1} = \frac{1}{K_v}. \quad (7.65)$$

◆ Example 7.16 Design by the Direct Method

Consider again the plant described by the transfer function of Eq. (7.53) and suppose we ask for a digital design that has the characteristic equation that is the discrete equivalent of the continuous characteristic equation

$$s^2 + s + 1 = 0.$$

with a sampling period $T = 1$ sec.

⁶ Roots on the unit circle are also unstable by some definitions, and good practice indicates that we should not cancel singularities outside the radius of desired settling time.

Solution. The discrete characteristic equation according to the specifications is

$$z^2 - 0.7859z + 0.36788 = 0. \quad (7.66)$$

Let us therefore ask for a design that is stable, has $K_v = 1$, and has poles at the roots of Eq. (7.66) plus, if necessary, additional poles at $z = 0$, where the transient is as short as possible. The form of $H(z)$ is thus

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + \dots}{1 - 0.7859z^{-1} + 0.3679z^{-2}}. \quad (7.67)$$

The causality design constraint, using Eq. (7.55) requires that

$$H(z)|_{z=\infty} = 0$$

or

$$b_0 = 0. \quad (7.68)$$

Equations (7.59) and (7.60) add no constraints because $G(z)$ has all poles and zeros inside the unit circle except for the single zero at ∞ , which is taken care of by Eq. (7.68). The steady-state error requirement leads to

$$\begin{aligned} H(1) &= 1 \\ &= \frac{b_1 + b_2 + b_3 + \dots}{1 - 0.7859 + 0.3679} = 1. \end{aligned} \quad (7.69)$$

Therefore

$$b_1 + b_2 + b_3 + \dots = 0.5820,$$

and

$$-T \frac{dH}{dz} \Big|_{z=1} = \frac{1}{K_v}.$$

Because in this case both T and K_v are 1, we use Eq. (7.69) and the derivative with respect to z^{-1} to obtain

$$\begin{aligned} 1 &= \frac{1}{K_v} = \frac{dH}{dz} \Big|_{z=1} \\ &= \frac{(0.5820)[b_1 + 2b_2 + 3b_3 + \dots] - [0.5820] - 0.7859 + 0.3679(2)}{(0.5820)(0.5820)} \end{aligned}$$

or

$$\frac{b_1 + 2b_2 + 3b_3 + \dots - [-0.05014]}{0.5820} = 1. \quad (7.70)$$

Because we have only two equations to satisfy, we need only two unknowns and we can truncate $H(z)$ at b_2 . The resulting equations are

$$b_1 + b_2 = 0.5820,$$

$$b_1 + 2b_2 = 0.5318,$$

which have the solution

$$b_1 = 0.6321. \quad (7.71)$$

$$b_2 = -0.05014. \quad (7.72)$$

Thus the final design gives an overall transfer function

$$H(z) = \frac{0.6321z - 0.05014}{z^2 - 0.7859z + 0.3679}. \quad (7.73)$$

We shall also need

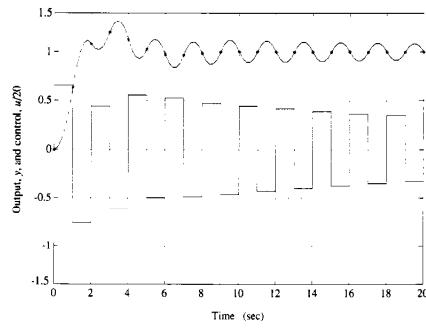
$$1 - H(z) = \frac{(z - 1)(z - 0.4180)}{z^2 - 0.7859z + 0.3679}. \quad (7.74)$$

We know that $H(1) = 1$ so that $1 - H(z)$ must have a zero at $z = 1$. Now, turning to the basic design formula, Eq. (7.54), we compute

$$\begin{aligned} D(z) &= \frac{(z - 1)(z - 0.9048)(0.6321)}{(0.04837)(z + 0.9672)} \frac{(z - 0.07932)}{(z - 1)(z - 0.4180)} \\ &= 13.07 \frac{(z - 0.9048)}{(z + 0.9672)} \frac{(z - 0.07932)}{(z - 0.4180)}. \end{aligned}$$

A plot of the step response of the resulting design for this example is provided in Fig. 7.37 and verifies that the response samples behave as specified by $H(z)$. However, as can be seen also from the figure, large oscillations occur in the

Figure 7.37
Step response of antenna system from direct design



control that cause a large ripple in the system response between samples. How can this be for a system response transfer function,

$$\frac{Y(z)}{R(z)} = H(z) = \frac{DG}{1 + DG},$$

that is designed to have only two well-damped roots? The answer lies in the fact that the control response is determined from

$$\frac{U(z)}{R(z)} = \frac{D}{1 + DG} = \frac{H(z)}{G(z)},$$

which for this example is

$$\frac{U(z)}{R(z)} = 13.06 \frac{z - 0.0793}{z^2 - 0.7859z + 0.3679} \frac{(z - 1)(z - 0.9048)}{z + 0.9672}.$$

The pole at $z = -0.9672$, very near to the unit circle, is the source of the oscillation in the control response. The poor transient due to the pole did not show up in the output response because it was exactly canceled by a zero in the plant transfer function. The large control oscillation in turn causes the ripple in the output response. This pole was brought about because we allowed the controller to have a pole to cancel a plant zero at this position. The poor response that resulted could have been avoided if this nearly unstable zero had been included in the stability constraint list. In that case we would introduce another term in $H(z)$, $b_2 z^{-2}$, and require that $H(z)$ be zero at $z = -0.9672$, so this zero of $G(z)$ is not canceled by $D(z)$. The result will be a simpler $D(z)$ with a slightly more complicated $H(z)$.

7.6 Summary

In this chapter we have reviewed the philosophy and specifications of the design of control systems by transform techniques and discussed three such methods.

- Discrete controllers can be designed by emulation, root locus, or frequency response methods.
- Successful design by emulation typically requires a sampling frequency at least 30 times the expected closed-loop bandwidth.
- Expressions for steady-state error constants for discrete systems have been given in Eq. (7.12) and Eq. (7.14) in terms of open-loop transfer functions and in Eq. (7.18) in terms of closed-loop poles and zeros.
- Root locus rules for discrete system characteristic equations are shown to be the same as the rules for continuous system characteristic equations.
- Step response characteristics such as rise time and overshoot can be correlated with regions of acceptable pole locations in the z -plane as sketched in Fig. 7.10.

- Asymptotes as used in continuous system frequency response plots *do not apply* for discrete frequency response.
- Nyquist's stability criterion and gain and phase margins were developed for discrete systems.
- The sensitivity function was shown to be useful to develop specifications on performance robustness as expressed in Eq. (7.34).
- Stability robustness in terms of the overall transfer function, the complementary sensitivity function, is expressed in Eq. (7.47).
- Limitations on the frequency response of closed-loop discrete designs are made more severe by poles and zeros outside the unit circle as expressed by Eq. (7.50) and Eq. (7.51).
- Lead and lag compensation can be used to improve the steady-state and transient response of discrete systems.
- The direct design method of Ragazzini can be used to realize a closed-loop transfer function limited only by causality and stability constraints.

7.7 Problems

- 7.1 Use the $z = e^{j\theta}$ mapping function and prove that the curve of constant ζ in s is a logarithmic spiral in z .
- 7.2 A servomechanism system is expected to have a rise-time of no more than 10 milliseconds and an overshoot of no more than 5%.
- Plot in the s -plane the corresponding region of acceptable closed-loop pole locations.
 - What is the estimated Bode gain crossover frequency (rad/sec)?
 - What is the estimated phase margin in degrees?
 - What is the sample period, T , if the estimated phase shift due to the sample and hold is to be no more than 10° at the gain crossover?
 - What is the sample period T if there are to be 8 samples per rise time?
- 7.3 *Root locus review.* The following root loci illustrate important features of the root locus technique. All are capable of being sketched by hand, and it is recommended that they be done that way in order to develop skills in verifying a computer's output. Once sketched roughly by hand, it is useful to fill in the details with a computer.
- The locus for

$$1 + K \frac{s+1}{s^2(s+p_1)} = 0$$

is typical of the behavior near $s = 0$ of a double integrator with lead compensation or a single integration with a lag network and one additional real pole. Sketch the locus versus K for values of p_1 of 5, 9, and 20. Pay close attention to the real axis break-in and break-away points.

- (b) The locus for

$$1 + K \frac{1}{s(s+1)((s+a)^2+4)} = 0$$

is a typical locus that includes complex poles and shows the value of departure angles. Plot the locus for $a = 0, 1$, and 2 . Be sure to note the departure angles from the complex poles in each case.

- (c) The locus for

$$1 + K \frac{(s+1)^2 + \omega_n^2}{s(s^2+4)} = 0$$

illustrates the use of complex zeros to compensate for the presence of complex poles due to structural flexibility. Be sure to estimate the angles of departure and arrival. Sketch the loci for $\omega_n = 1$ and $\omega_n = 3$. Which case is unconditionally stable (stable for all positive K less than the design value)?

- (d) For

$$1 + K \frac{s}{(s-p_1)(s-p_2)} = 0$$

show that the locus is a circle of radius $\sqrt{p_1 p_2}$ centered at the origin (location of the zero). Can this result be translated to the case of two poles and a zero on the negative real axis?

- 7.4 The basic transfer function of a satellite attitude control is $G(s) = \frac{1}{s^3}$.

- Design a continuous lead network compensation so as to give closed-loop poles corresponding to $\zeta = 0.5$ and natural frequency $\omega_n = 1.0$. The ratio of pole to zero of the lead is to be no more than 10.
 - Plot the step response of the design and note the rise time and the percent overshoot.
 - What is the system type and corresponding error constant?
- Select a sampling period to give 10 samples in a rise time and compute the discrete equivalent to the lead using the Tustin bilinear transformation. Plot the step response of the discrete system and compare the rise time and overshoot to those of the continuous design.
- Select a sampling period that will give 5 samples per rise time, compute the discrete equivalent using Tustin's method, and compare rise time and overshoot of this design with the continuous case.

- 7.5 Repeat the design for the satellite attitude control of Problem 4, including method of choosing sampling periods but using the matched pole-zero method to obtain the discrete compensations.

- 7.6 Repeat the design of the satellite attitude control of Problem 4 including method of choice of sampling period but using the triangle hold equivalent (noncausal first-order hold) to design the discrete compensations.

- 7.7** Repeat the design for the satellite attitude control of Problem 4 but augment the plant with a Padé approximation to the delay of $T/2$ which is to say, multiply the plant transfer function by

$$P(s) = \frac{1 - \frac{sT}{2}}{1 + \frac{sT}{2}}$$

before doing the continuous design. Once the design of the lead compensation is completed, continue with the discrete equivalents as in Problem 4, including the method of choosing sampling periods. Use the matched pole-zero method to obtain the discrete compensations. Compare the design with the continuous case.

- 7.8** Design a discrete compensation for the antenna control system as specified in Example 7.2 with a sample period of $T = 0.1$ using a matched pole-zero equivalent for the discrete compensation. Plot the step response and compare rise time and overshoot with those of the continuous design.

- 7.9** Design the antenna control system as specified in Example 7.2 with a sample period of $T = 0.5$ sec.

- (a) Use the zero-pole mapping equivalent emulation method.
 (b) Augment the plant model with an approximation of the sample-hold delay consisting of

$$P(s) = \frac{2/T}{s + 2/T},$$

then redesign $D(s)$ and find the discrete equivalent with the matched pole-zero equivalent emulation method. Plot the step response and compare with the continuous design done on the unaugmented plant.

- (c) Compare the degradation of the equivalent damping ratio ζ due to sampling for both design methods.

- 7.10** For the satellite with transfer function $1/s^2$, design a lead compensation to give closed-loop poles with damping $\zeta = 0.5$ and natural frequency $\omega_n = 1.0$. The pole-to-zero ratio of the compensation should not be more than 10. Plot the step response of the closed loop and note the rise time and overshoot.

- (a) Let sampling period be $T_s = 0.5$ sec and compute the discrete model of the plant with a sample and zero-order hold. Using this model, design a discrete lead compensation with pole at $\zeta = -0.5$ and a zero so as to give closed-loop poles at the mapped place from the continuous poles. $\zeta = 0.5$ and $\omega_n = 1.0$. What is the ratio of ω/ω_n for this problem? How many samples do you expect to find per rise time? Plot the step response and compare result with expectation. Compare the discrete design with the continuous design.

- (b) Repeat the discrete design with sampling period $T_s = 1.2$ sec, plot the step response, and compare rise time and overshoot with the continuous case.

- 7.11** Sketch the region in the z -plane of discrete pole locations corresponding to $\zeta \geq 0.5$ and

- (a) $\omega_n \leq \omega_s/30$.
 (b) $\omega_n \leq \omega_s/10$.
 (c) $\omega_n \leq \omega_s/5$.

- 7.12** The plant transfer function

$$G(s) = \frac{1}{(s + 0.1)(s + 3)}$$

is to be controlled with a digital controller using a sample period of $T = 0.1$ sec.

- (a) Design compensation using the z -plane root locus that will respond to a step with a rise time of ≤ 1 sec and an overshoot $\leq 5\%$. Plot the step response and verify that the response meets the specifications.
 (b) What is the system type and corresponding error constant? What can be done to reduce the steady-state error to a step input?
 (c) Design a discrete lag compensation that will cut the steady-state error in half. Plot the step response and compare the complete response to the transient and steady-state error specifications.

- 7.13** It is possible to suspend a steel ball bearing by means of an electromagnet whose current is controlled by the position of the mass [Woodson and Melcher (1968)]. A schematic of a possible setup is shown in Fig. 7.38. The equations of motion are

$$m\ddot{x} = -mg + f(X, I),$$

where the force on the ball due to the electromagnet is given by $f(X, I)$. It is found that the magnet force balances the gravity force when the magnet current is I_0 , and the ball at X_0 . If we write $I = I_0 + i$ and $X = X_0 + x$ and expand f about $X = X_0$ and $I = I_0$, and then neglect higher-order terms, we obtain a linear approximation

$$m\ddot{x} = k_1 x + k_2 i,$$

Values measured for a particular device in the Stanford Controls Laboratory are $m = 0.02$ kg, $k_1 = 20$ N/m, $k_2 = 0.4$ N/A.

- (a) Compute the transfer function from i to x and draw the (continuous) root locus for proportional feedback $i = -Kx$.
 (b) Let the sample period be $T = 0.02$ sec and compute the plant discrete transfer function when used with a sample and zero-order hold.
 (c) Design a digital control for the magnetic levitation to meet the specifications $t_r \leq 0.1$ sec, $t_c \leq 0.4$ sec, and overshoot $\leq 20\%$.

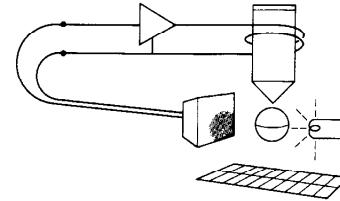


Figure 7.38
A steel ball balanced by an electromagnet

- (d) Plot a root locus of your design versus m and discuss the possibility of balancing balls of various masses.
- (e) Plot a step response of your design to an initial disturbance displacement on the ball and show both x and the control current i . If the sensor can measure x on a range of only $\pm \frac{1}{2}$ cm, and if the amplifier can provide a maximum current of 1 A, what is the maximum initial displacement, $x(0)_{\max}$, that will keep the variables within these limits, using $m = 0.02$ kg?

7.14 A discrete transfer function for approximate derivative control is

$$D(z) = K_p T_p \frac{z - 1}{Tz}.$$

where the pole at $z = 0$ adds some destabilizing phase lag. It therefore seems that it would be advantageous to remove it and to use derivative control of the form

$$D(z) = K_p T_p \frac{(z - 1)}{T}.$$

Can this be done? Support your answer with the difference equation that would be required and discuss the requirements to implement it.

7.15 For the automotive cruise-control system shown in Fig. 7.39, the sample period is $T = 0.5$ sec.

- (a) Design a PD controller to achieve a t_r of 5 sec with no overshoot.
- (b) Determine the speed error on a 3% grade (i.e., $G_r = 3$ in Fig. 7.39).
- (c) Design a PID controller to meet the same specifications as in part (a) and that has zero steady-state error on constant grades. What is the velocity constant of your design?

7.16 For the disk drive read/write head assembly described in Fig. 7.40, you are to design a compensation that will result in a closed-loop settling time $t_s = 20$ msec and with overshoot to a step input $M_p \leq 20\%$.

- (a) Assume no sampling and use a continuous compensation. Plot the step response and verify that your design meets the specifications.
- (b) Assume a sampling period $T = 1$ msec and use matched pole-zero emulation. If you wish, you can include a Pade approximation to the delay and do a redesign of the continuous compensation before computing the discrete equivalent. Plot the step response and compare it with the continuous design's response.

Figure 7.39

An automotive cruise-control system

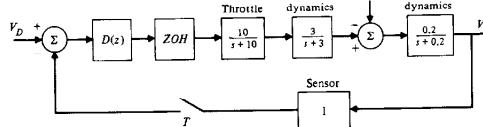
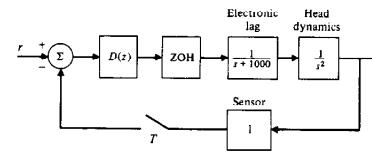


Figure 7.40
A disk drive read/write head assembly



- (c) Do a z -plane design for the same specifications and plot its step response. Compare these three designs with respect to meeting the transient performance.

7.17 The tethered satellite system shown in Fig. 7.41 has a moveable tether attachment point so that torques can be produced for attitude control. The block diagram of the system is shown in Fig. 7.42. Note that the integrator in the actuator block indicates that a constant-voltage command to the servomotor will produce a constant velocity of the attachment point.

- (a) Is it possible to stabilize this system with θ feedback to a PID controller? Support your answer with a root locus argument.
- (b) Suppose it is possible to measure δ , and θ as well as $\dot{\theta}$. Select the variable(s) on which you would like to place a sensor(s) to augment the θ feedback.
- (c) Design compensation for the system using the sensor(s) that you selected in part (b) so that it has a 2-sec rise time and equivalent closed loop damping of $\zeta = 0.5$.

Figure 7.41
A tethered satellite system

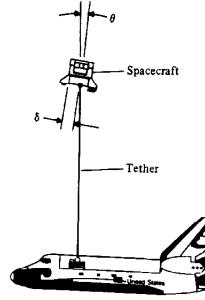
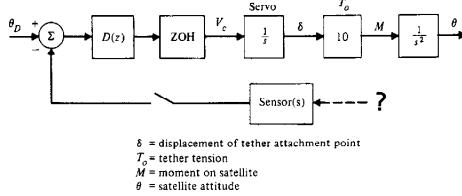
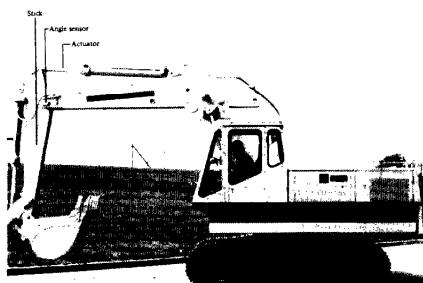


Figure 7.42

Block diagram for the tethered satellite system

**Figure 7.43**

An excavator with an automatic control system



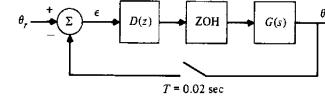
7.18 The excavator shown in Fig. 7.43 has a sensor measuring the angle of the stick as part of a control system to control automatically the motion of the bucket through the earth. The sensed stick angle is to be used to determine the control signal to the hydraulic actuator moving the stick. The schematic diagram for this control system is shown in Fig. 7.44, where $G(s)$ is the system transfer function given by

$$G(s) = \frac{1000}{s(s+10)(s^2 + 1.2s + 144)}.$$

The compensation is implemented in a control computer sampling at $f_s = 50$ Hz and is of the form $D(z) = K(1 - z^{-1}) + K_1(1 - 2z^{-1} + z^{-2})$. The oscillatory roots in $G(s)$ arise from the compressibility of the hydraulic fluid (with some entrained air) and is often referred to as the *oil-mass resonance*.

Figure 7.44

Schematic diagram for the control system of the excavator



- (a) Show that the steady-state error, $\epsilon (= \theta_r - \theta)$, is

$$\epsilon(\infty) = \frac{1.44}{K}.$$

when θ_r is a unit ramp.

- (b) Determine the highest K possible (i.e., at the stability boundary) for proportional control ($K_1 = K_2 = 0$).
- (c) Determine the highest K possible (i.e., at the stability boundary) for PD ($K_2 = 0$) control.
- (d) Determine the highest K possible (i.e., at the stability boundary) for PD plus acceleration ($K_2 \neq 0$) control.⁷

- 7.19** For the excavator described in Problem 7.18 with transfer function

$$G(s) = \frac{1000}{s(s+10)(s^2 + 1.2s + 144)},$$

plot the Bode plot and measure the gain and phase margins with just unity feedback.

- (a) Design a compensation that will give a phase margin of 50°, a gain margin as measured at the resonance peak of at least 2 and a crossover of at least $\omega_c \geq 1.0$. Plot the step response of the resulting design and note the rise time and the overshoot.
- (b) With sample frequency $f_s = 50$ Hz, design a discrete controller to meet the same specifications given for the continuous case. Plot the Bode plot of the design and verify that the specifications are met.

- 7.20** A simple model of a satellite attitude control has the transfer function $1/s^2$.

- (a) Plot the Bode plot for this system and design a lead compensation to have a phase margin of 50° and a crossover ω_{cg} of 1.0 rad/sec. Plot the step response and note the rise time and overshoot.
- (b) With sample period of $T = 0.2$ sec, design a discrete compensation to give 50° phase margin and crossover $\omega_{cg} = 1.0$. Plot the step response and compare rise time and overshoot with the continuous design.
- (c) With sample period $T = 0.5$ sec, design a discrete compensation to give 50° phase margin and crossover $\omega_{cg} = 1.0$. Plot the step response and compare rise time and overshoot with the continuous design.

- 7.21** For a system given by

$$G(s) = \frac{a}{s(s+a)},$$

⁷ For further reading on damping the oil-mass resonance of hydraulic systems, see Viersma (1980).

determining the conditions under which the K_c of the continuous system is approximately equal to the K_d of the system preceded by a ZOH and represented by its discrete transfer function.

- 7.22 Design a digital controller for

$$G(s) = \frac{1}{s(s + 0.4)}$$

preceded by a ZOH so that the response has a rise time of approximately 0.5 sec, overshoot < 25%, and zero steady-state error to a step command. [Hint: Cancel the plant pole at $s = 0.4$ with a compensator zero; a second-order closed-loop system will result, making the transient response comparison between experiment and theory much easier.]

- (a) Determine a $D(z)$ using emulation with the matched pole-zero mapping technique. Do two designs, one for $T = 100$ msec and one for $T = 250$ msec.
- (b) Repeat part (a) using the z -plane root locus method for the two sample periods.
- (c) Simulate the closed-loop system response to a unit step with the $D(z)$'s obtained in parts (a) and (b). Use the discrete equivalent of the plant in your calculations. Compare the four digitally controlled responses with the original specifications. Explain any differences that you find.

- 7.23 A generic mechanical control problem is that of two masses coupled by a lightly damped flexible structure. With amplitude and time scaling, the model can be reduced to

$$G(s) = K \frac{Bs + 1}{s^2(s^2 + Bs + 1)}$$

For the parameters $K = 0.05$ and $B = 0.1$, plot the Bode plot and indicate the gain and phase margins. For performance tracking, the open loop gain for frequencies below $\omega = 0.005$ must be above 50 and the gain for frequencies above $\omega = 1$ must be below 0.5. The phase margin must be at least 35°.

- (a) Design a continuous compensation to meet the specifications. If possible, keep the phase above -180° so the system will not be conditionally stable.
- (b) Design a discrete compensation for the system with the lowest sampling frequency possible. A prize will be given to the student who obtains the lowest sampling frequency with a design that meets the specifications.
- (c) Plot the sensitivity of the digital design and compute the vector gain margin.

• 8 •

Design Using State-Space Methods

A Perspective on State-Space Design Methods

In Chapter 7, we discussed how to design digital controllers using transform techniques, methods now commonly designated as "classical design." The goal of this chapter is to solve the identical problem using the state-space formulation. The difference in the two approaches is entirely in the design method; the end result, a set of difference equations providing control, is identical.

Advantages of the state-space formulation are especially apparent when designing controllers for multi-input, multi-output (MIMO) systems, that is, those with more than one control input and/or sensed output. However, state-space methods are also an aid in the design of controllers for single-input, single-output (SISO) systems because of the widespread use of computer-aided control system design (CACSD) tools, which often rely heavily on this system representation. Chapters 4 and 5 have already demonstrated the advantages of the state-space formulation in using CACSD packages for the computation of discrete equivalents. State-space methods also offer advantages in the structure for command inputs and disturbance estimation. In this chapter, we will limit our state-space design efforts to SISO controllers, similar to those found in Chapter 7 with classical methods. Techniques for MIMO design are discussed in Chapter 9.

In Chapter 7, two basic methods were described: *emulation* and *direct digital design*. The same two methods apply to the state-space formulation as well. Using emulation, one would design a continuous controller using state-space methods, then transform the controller to a discrete form by using one of the discrete equivalents from Chapter 6. The discussion of the method and its accuracy in Chapter 7 applies equally well here. Furthermore, the development in Chapter 6 used both classical and state-space system descriptions in the computation of the equivalents. Therefore, no further discussion of emulation is required, and we will concentrate solely on the direct digital design method.

Chapter Overview

As for the continuous case reviewed in Section 2.6, design using state-space involves separate design of the control assuming all state elements are available, then design of an estimator to reconstruct the state given a partial measurement of it. Section 8.1 covers the discrete control design while Section 8.2 covers the discrete estimator design. Section 8.3 puts it together into what is called the **regulator** and Section 8.4 discusses the relative merits of the various ways of introducing the reference input command. Section 8.5 presents how a designer builds an integral control feature or disturbance estimation, and how they accomplish similar goals. Section 8.6 discusses the limitations imposed by delays in the system and how to minimize their effect. The chapter concludes in Section 8.7 with a discussion of observability and controllability, the required conditions for the design to be possible.

8.1 Control Law Design

In Chapter 2, we saw that the state-space description of a continuous system is given by Eq. (2.1)

$$\dot{x} = Fx + Gu, \quad (8.1)$$

and Eq. (2.2)

$$y = Hx + Ju. \quad (8.2)$$

We assume the control is applied from the computer through a ZOH as shown in Fig. 1.1. Therefore, Eqs. (8.1) and (8.2) have an exact discrete representation as given by Eq. (4.59)

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k), \\ y(k) &= Hx(k) + Ju(k). \end{aligned} \quad (8.3)$$

where

$$\begin{aligned} \Phi &= e^{F\tau}, \\ \Gamma &= \int_0^{\tau} e^{Ft} d\eta G. \end{aligned} \quad (8.4)$$

We can easily transform between the classical transfer function of a continuous system, $G(s)$ (represented by sysTF in MATLAB), to the state-space continuous description by the MATLAB script¹

`sysC = ss(sysTF)`

¹ In MATLAB 4, $G(s)$ would be represented by [num,den] and the tf2ss function would be invoked. See Appendix E.

estimator

where sysC contains F , G , H , J . Provided the continuous system is preceded by a ZOH, we can transform to the discrete model, Φ , Γ , H , J , (or sysD) with a sample period T , using the MATLAB script

`sysD = c2d(sysC,T)`.

One of the attractive features of state-space design methods is that the procedure consists of two independent steps. The first step *assumes* that we have all the state elements at our disposal for feedback purposes. In general, of course, this would not be a good assumption: a practical engineer would not, as a rule, find it necessary to purchase such a large number of sensors, especially because he or she knows that they would not be needed using classical design methods. The assumption that all states are available merely allows us to proceed with the first design step, namely, the control law. The remaining step is to design an "estimator" (or "observer"), which estimates the entire state vector, given measurements of the portion of the state provided by Eq. (8.2). The final control algorithm will consist of a combination of the control law and the estimator with the control-law calculations based on the estimated states rather than on the actual states. In Section 8.3 we show that this substitution is reasonable and that the combined control law and estimator can give closed-loop dynamic characteristics that are unchanged from those assumed in designing the control law and estimator separately. The dynamic system we obtain from the combined control law and estimator is the same that has been previously referred to as compensation.

As for the continuous case, the control law is simply the feedback of a linear combination of all the state elements, that is

$$u = -Kx = -[K_1 \ K_2 \ \dots] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix}. \quad (8.5)$$

Note that this structure does not allow for a reference input to the system. The topology that we used all through Chapter 7 (Fig. 7.5) always included a reference input, r . The control law, Eq. (8.5), assumes that $r = 0$ and is, therefore, usually referred to as a **regulator**. Section 8.4 will discuss how one introduces reference inputs.

Substituting Eq. (8.5) in Eq. (8.3), we have

$$x(k+1) = \Phi x(k) - \Gamma Kx(k). \quad (8.6)$$

Therefore the z -transform of Eq. (8.6) is

$$(zI - \Phi + \Gamma K)X(z) = 0,$$

² The literature [Luenberger (1960)] commonly refers to these devices as "observers"; however, we feel that the term "estimator" is much more descriptive of their function because "observe" implies a direct measurement. In this book the term "estimator" is used but the reader can think of the terms interchangeably.

and the characteristic equation of the system with the hypothetical control law is

$$|zI - \Phi + \Gamma K| = 0. \quad (8.7)$$

8.1.1 Pole Placement

The approach we wish to take at this point is pole placement; that is, having picked a control law with enough parameters to influence all the closed-loop poles, we will arbitrarily select the desired pole locations of the closed-loop system and see if the approach will work. Although this approach can often lead to trouble in the design of complex systems (see the discussion in Sections 2.6 and 8.3), we use it here to illustrate the power of full state feedback. In Chapter 9, we will build on this idea to arrive at a more practical design methodology.

The control-law design, then, consists of finding the elements of \mathbf{K} so that the roots of Eq. (8.7), that is, the poles of the closed-loop system, are in the desired locations. Unlike classical design, where we iterate on parameters in the compensator (hoping) to find acceptable closed-loop root locations, the full state feedback, pole-placement approach guarantees success and allows us to arbitrarily pick any pole location, providing that n poles are specified for an n th-order system.

Given desired pole locations,³ say

$$z_i = \beta_1, \beta_2, \beta_3, \dots, \beta_n.$$

the desired control-characteristic equation is

$$\alpha_c(z) = (z - \beta_1)(z - \beta_2) \cdots (z - \beta_n) = 0. \quad (8.8)$$

Equations (8.7) and (8.8) are both the characteristic equation of the controlled system; therefore, they must be identical, term by term. Thus we see that the required elements of \mathbf{K} are obtained by matching the coefficients of each power of z in Eq. (8.7) and Eq. (8.8), and there will be n equations for an n th-order system.

◆ Example 8.1 Pole Placement for Satellite Attitude Control

Design a control law for the satellite attitude-control system described by Eq. (4.47). Pick the z -plane roots of the closed-loop characteristic equation so that the equivalent s -plane roots have a damping ratio of $\xi = 0.5$ and real part of $s = -1.8$ rad/sec (i.e., $s = -1.8 \pm j3.12$ rad/sec). Use a sample period of $T = 0.1$ sec.

³ Discussion of how one selects pole locations was reviewed in Section 2.6 for the continuous case and will occur through the following examples and in Section 8.3 for the discrete case. The results of the specification discussion in Chapter 7 can also be used to specify poles.

Solution. Example 4.11 showed that the discrete model for this system is

$$\Phi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \Gamma = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}. \quad (8.9)$$

Using $z = e^{jT}$ with a sample period of $T = 0.1$ sec, we find that $s = -1.8 \pm j3.12$ rad/sec translates to $z = 0.8 \pm j0.75$, as shown in Fig. 8.1. The desired characteristic equation is then

$$z^2 - 1.6z + 0.70 = 0. \quad (8.9)$$

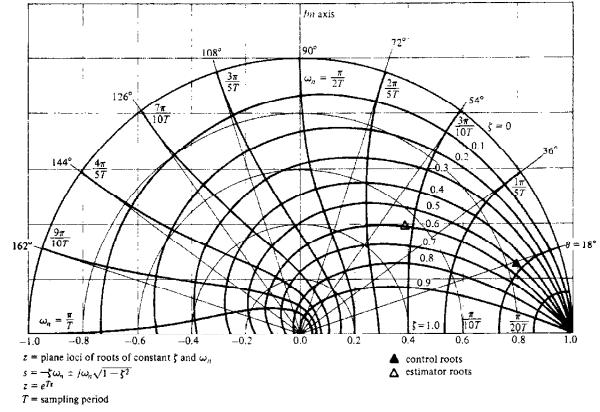
and the evaluation of Eq. (8.7) for any control law \mathbf{K} leads to

$$\left| z \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} [K_1 \ K_2] \right| = 0$$

or

$$z^2 + (TK_2 - (T^2/2)K_1 - 2)z + (T^2/2)K_1 - TK_2 + 1 = 0. \quad (8.10)$$

Figure 8.1 Desired root locations for satellite attitude-control system of Examples 8.1 and 8.4



Equating coefficients in Eqs. (8.9) and (8.10) with like powers of z , we obtain two simultaneous equations in the two unknown elements of \mathbf{K} :

$$\begin{aligned} T K_2 + (T^2/2) K_1 - 2 &= -1.6, \\ (T^2/2) K_1 - T K_2 + 1 &= 0.70, \end{aligned}$$

which are easily solved for the coefficients and evaluated for $T = 0.1$ sec:

$$K_1 = \frac{0.10}{T^2} = 10, \quad K_2 = \frac{0.35}{T} = 3.5.$$

control canonical form

The calculation of the gains using the method illustrated in the previous example becomes rather tedious when the order of the system (and therefore the order of the determinant to be evaluated) is greater than 2. A computer does not solve the tedium unless it is used to perform the algebraic manipulations necessary in expanding the determinant in Eq. (8.7) to obtain the characteristic equation. Therefore, other approaches have been developed to provide convenient computer-based solutions to this problem.

The algebra for finding the specific value of \mathbf{K} is especially simple if the system matrices happen to be in the form associated with the block diagram of Fig. 4.8(c). This structure is called "control canonical form" because it is so useful in control law design. Referring to that figure and taking the state elements as the outputs of the delays (z^{-1} blocks), numbered from the left, we get (assuming $b_0 = 0$ for this case)

$$\Phi_c = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \Gamma_c = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{H}_c = [b_1 \ b_2 \ b_3]. \quad (8.11)$$

Note that from Eq. (4.15), the characteristic polynomial of this system is $a(z) = z^3 + a_1 z^2 + a_2 z + a_3$. The key idea here is that the elements of the first row of Φ_c are exactly the coefficients of the characteristic polynomial of the system. If we now form the closed-loop system matrix $\Phi_c - \Gamma_c \mathbf{K}$, we find

$$\Phi_c - \Gamma_c \mathbf{K} = \begin{bmatrix} -a_1 - K_1 & -a_2 - K_2 & -a_3 - K_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \quad (8.12)$$

By inspection, we find that the characteristic equation of Eq. (8.12) is

$$z^3 + (a_1 + K_1)z^2 + (a_2 + K_2)z + (a_3 + K_3) = 0.$$

Thus, if the desired pole locations result in the characteristic equation

$$z^3 + \alpha_1 z^2 + \alpha_2 z + \alpha_3 = 0,$$

then the necessary values for control gains are

$$K_1 = \alpha_1 - a_1, \quad K_2 = \alpha_2 - a_2, \quad K_3 = \alpha_3 - a_3. \quad (8.13)$$

Conceptually, then, we have the canonical-form design method: Given an arbitrary (Φ, Γ) and a desired characteristic equation $a(z) = 0$, we convert (by redefinition of the state) (Φ, Γ) to control form (Φ_c, Γ_c) and solve for the gain by Eq. (8.13). Because this gain is for the state in the control form, we must finally express the result back in terms of the original state elements. This method is sometimes used by CACSD packages because of the numerical advantages; however, the transformation is transparent to the designer, who generally prefers to use a state definition that is related to the physical system's characteristics.

8.1.2 Controllability

The first question this process raises is existence: Is it always possible to find an equivalent (Φ_c, Γ_c) for arbitrary (Φ, Γ) ? The answer is almost always "yes." The exception occurs in certain pathological systems, dubbed "uncontrollable," for which no control will give arbitrary pole locations. These systems have certain modes or subsystems that are unaffected by the control. Uncontrollability is best exhibited by a realization (selection of state elements) where each state element represents a natural mode of the system. If all the roots of the open-loop characteristic equation

$$|z\mathbf{I} - \Phi| = 0$$

are distinct, then Eq. (8.3) written in this way (normal mode or "Jordan canonical form") becomes

$$\mathbf{x}(k+1) = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_n \end{bmatrix} u(k), \quad (8.14)$$

and explicitly exhibits the criterion for controllability: No element in Γ can be zero. If any Γ element was zero, the control would not influence that normal mode, and the associated state would remain uncontrolled. A good physical understanding of the system being controlled usually prevents any attempt to design a controller for an uncontrollable system; however, there is a mathematical test for controllability applicable to any system description, which may be an additional aid in discovering this condition: a discussion of this test is contained in Section 8.7.

8.1.3 Pole Placement Using CACSD

Ackermann's formula

MATLAB has two functions that perform the calculation of \mathbf{K} : `place.m` and `acker.m`. `Acker` is based on Ackermann's formula [Ackermann (1972)] and is satisfactory for SISO systems of order less than 10 and can handle systems with repeated roots. The relation is

$$\mathbf{K} = [0 \dots 1] [\Gamma \Phi\Gamma \Phi^2\Gamma \dots \Phi^{n-1}\Gamma]^{-1} \alpha_c(\Phi) \quad (8.15)$$

controllability matrix

where $\mathcal{C} = [\Gamma \Phi\Gamma \dots]$ is called the **controllability matrix**, n is the order of the system or number of state elements, and we substitute Φ for z in $\alpha_c(z)$ to form

$$\alpha_c(\Phi) = \Phi^n + \alpha_1\Phi^{n-1} + \alpha_2\Phi^{n-2} + \dots + \alpha_n\mathbf{I}, \quad (8.16)$$

where the α_i 's are the coefficients of the desired characteristic equation, that is,

$$\alpha_c(z) = |z\mathbf{I} - \Phi + \Gamma\mathbf{K}| = z^n + \alpha_1z^{n-1} + \dots + \alpha_n. \quad (8.17)$$

The controllability matrix, \mathcal{C} , must be full rank for the matrix to be invertible and for the system to be controllable.

`place` [Kautsky, Nichols, and Van Dooren (1985)] is best for higher order systems and can handle MIMO systems, but will not handle systems where the desired roots are repeated.

Note that these functions are used for both the continuous and discrete cases because they solve the same mathematical problems given by Eqs. (2.37) and (8.7). The only difference is that the desired root locations are in substantially different locations for the s -plane and z -plane and that \mathbf{F} , \mathbf{G} have been replaced by Φ and Γ .

◆ Example 8.2 Control Law Pole Placement with MATLAB

Design a control law for the satellite attitude-control system as in Example 8.1. Place the z -plane closed-loop poles at $z = 0.8 \pm j0.25$.

Solution. The MATLAB statements

```
T = .1
Phi = [1 T;0 1]
Gam = [T^2/2;T]
p = [8+1*.25,-8-1*.25]
K = acker(Phi,Gam,p)
```

result in $\mathbf{K} = [10.25 \quad 3.4875]$. The difference between these values and those shown in Example 8.1 are due to round-off error in the hand calculation. `place` would have given the same answer. ◆

A more complex system demonstrates the kind of difficulty you might encounter in using the pole placement approach. The specific difficulty is brought about by the necessity to pick n desired pole locations. Where should the higher frequency poles be picked? The system specifications typically help the designer pick only two of the desired poles. As discussed in Section 2.6, it is helpful to move poles as little as possible in order to minimize the required amount of control effort and to avoid exciting the system any more than necessary. The following example specifically illustrates how a poor choice of desired poles can cause an undesirable response, and how a wise choice of desired poles can drastically improve the situation.

◆ Example 8.3 Pole Placement for a 4th-Order System with MATLAB

Design a control law for the double mass spring system in Appendix A.4 using d as the measurement. This system is representative of many systems where there is some flexibility between the measured output and control input. Assume the resonant mode has a frequency $\omega_n = 1$ rad/sec and damping $\zeta = 0.02$ and select a 10:1 ratio of the two masses. The parameters that provide these characteristics are: $M = 1$ kg, $m = 0.1$ kg, $b = 0.0036$ N·sec/m, and $k = 0.091$ N/m. Pick the sample rate to be 15 times faster than the resonance and show the free response to an initial condition of $d = 1$ m for two cases:

- (a) Pick all the poles at $z = 0.9$.
- (b) Pick the poles at $z = 0.9 \pm j0.05, 0.8 \pm j0.4$.

Discuss why the response is better for case (b).

Solution. From Eq. (A.17) we can write the state-space description as

$$x = [d \quad \dot{d} \quad y \quad \dot{y}]^T \quad \mathbf{H} = [1 \quad 0 \quad 0 \quad 0]^T$$

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -0.91 & -0.036 & 0.91 & 0.036 \\ 0 & 0 & 0 & 1 \\ 0.091 & 0.0036 & -0.091 & -0.0036 \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad J = 0.$$

The sample rate should be 15 rad/sec which translates to approximately $T = 0.4$ secs.

- (a) All poles at $z = 0.9$, the MATLAB script

```
sysC = ss(F,G,H,J)
```

```

sysD = c2d(sysC,T,'zoh')
p = [ .9; .9; .9]
[phi,gam,H,J] = ssdata(sysD)
K=acker(phi,gam,p)

results in the feedback gain
K = [0.650 -0.651 -0.645 -0.718].          (8.18)

place cannot be used for case (a) because of the repeated roots. For the response to an
initial condition, the script

```

```

sysCL = feedback(K*sysD,1)
X0 = [1;0;0;0]
y = initall(sysCL,X0)

```

produces the closed-loop response for $d(0) = 1$ m, shown in Fig. 8.2(a). It exhibits a response that is *much* larger than that of the initial condition, but the time characteristics are consistent with the selected poles.

- (b) For the desired poles at $z = 0.9 \pm j0.05, 0.8 \pm j0.4$, we modify the script above with

$$p = [.9+j*.05; .9-j*.05; 8+j*.4; 8-j*.4]$$

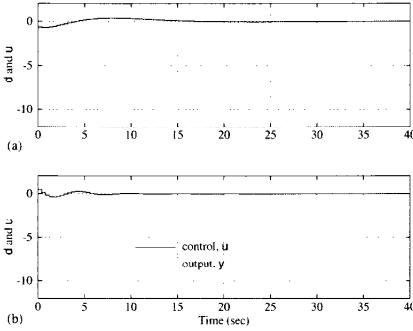


Figure 8.2
Initial condition response
for Example 8.3;
(a) desired poles all at
 $z = 0.9$, and (b) desired
poles at $z =$
 $0.9 \pm j0.05, 0.8 \pm j0.4$

and this results in the feedback gain

$$K = [-0.458 -0.249 0.568 0.968]. \quad (8.19)$$

which produces the response to an initial condition, $d = 1$ m, shown in Fig. 8.2(b). It exhibits *much* less response of d with no increase in control effort, although the resonant mode oscillations did influence the response with damping consistent with the poles selected. The primary reason for the superior response is that the oscillatory mode was not changed substantially by the control. Two of the selected poles ($z = 0.8 \pm j0.4$) have a natural frequency $\omega_n = 1$ rad/sec with a damping $\zeta = 0.2$. Therefore, the control is not attempting to change the natural frequency of the resonant mode at all; rather, the only task for the control in modifying this pole is to increase its damping from $\zeta = 0.02$ to $\zeta \geq 0.2$. Since this mode remains lightly damped, its oscillations are still visible on the output. The selected poles at $z = 0.9 \pm j0.05$ affect the overall motion of the system and their placement is less critical. Generally, pole selections with a damping $\zeta \geq 0.7$ give a better balance between system response and control usage. The control is clearly much more effective with these desired pole locations. ◆

controller pole selection

So we see that the mechanics of computing the control law are easy, once the desired pole locations are known. The trick is to pick a good set of poles! The designer should iterate between pole selections and some other system evaluation to determine when the design is complete. System evaluation might consist of an initial-condition time response as shown in the example, a step response, steady-state errors, gain and phase margins, or the entire frequency-response shape. Pole placement by itself leaves something to be desired. But it is useful as a design tool to be used in conjunction with the other design methods discussed in Chapter 7 or as a part of an optimal design process that will be discussed in Chapter 9.

8.2 Estimator Design

The control law designed in the last section assumed that all state elements were available for feedback. Typically, not all elements are measured; therefore, the missing portion of the state needs to be reconstructed for use in the control law. We will first discuss methods to obtain an estimate of the entire state given a measurement of one of the state elements. This will provide the missing elements as well as provide a smoothed value of the measurement, which is often contaminated with random errors or "noise." There are two basic kinds of estimates of the state, $\mathbf{x}(k)$: We call it the **current estimate**, $\hat{\mathbf{x}}(k)$, if based on measurements $y(k)$ up to and including the k th instant; and we call it the **predictor estimate**, $\tilde{\mathbf{x}}(k)$, if based on measurements up to $y(k-1)$. The idea eventually will be to let $u = -\mathbf{K}\hat{\mathbf{x}}$ or $u = -\mathbf{K}\tilde{\mathbf{x}}$, replacing the true state used in Eq. (8.5) by its estimate.

8.2.1 Prediction Estimators

One method of estimating the state vector which might come to mind is to construct a model of the plant dynamics,

$$\tilde{x}(k+1) = \Phi\tilde{x}(k) + \Gamma u(k). \quad (8.20)$$

We know Φ , Γ , and $u(k)$, and hence this estimator should work if we can obtain the correct $x(0)$ and set $\tilde{x}(0)$ equal to it. Figure 8.3 depicts this "open-loop" estimator. If we define the error in the estimate as

$$\tilde{x} = x - \tilde{x}, \quad (8.21)$$

and substitute Eqs. (8.3) and (8.20) into Eq. (8.21), we find that the dynamics of the resulting system are described by the estimator-error equation

$$\tilde{x}(k+1) = \Phi\tilde{x}(k) + \Gamma u(k). \quad (8.22)$$

Thus, if the initial value of \tilde{x} is off, the dynamics of the estimate error are those of the uncompensated plant, Φ . For a marginally stable or unstable plant, the error will never decrease from the initial value. For an asymptotically stable plant, an initial error will decrease only because the plant and estimate will both approach zero. Basically, the estimator is running open loop and not utilizing any continuing measurements of the system's behavior, and we would expect that it would diverge from the truth. However, if we feed back the difference between the measured output and the estimated output and constantly correct the model with this error signal, the divergence should be minimized. The idea is to construct a feedback system around the open-loop estimator with the estimated output error as the feedback. This scheme is shown in Fig. 8.4; the equation for it is

$$\tilde{x}(k+1) = \Phi\tilde{x}(k) + \Gamma u(k) + L_p[y(k) - H\tilde{x}(k)]. \quad (8.23)$$

Figure 8.3
Open-loop estimator

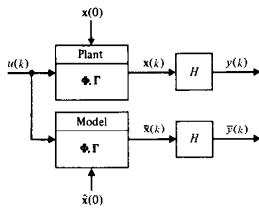
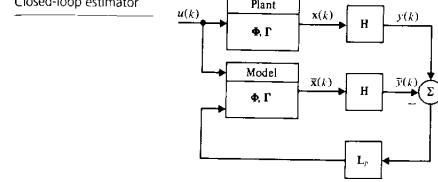


Figure 8.4
Closed-loop estimator



estimator error equation

where L_p is the feedback gain matrix. We call this a **prediction estimator** because a measurement at time k results in an estimate of the state vector that is valid at time $k+1$; that is, the estimate has been predicted one cycle in the future.

A difference equation describing the behavior of the estimation errors is obtained by subtracting Eq. (8.23) from Eq. (8.3). The result is

$$\tilde{x}(k+1) = [\Phi - L_p H]\tilde{x}(k). \quad (8.24)$$

This is a homogeneous equation, but the dynamics are given by $[\Phi - L_p H]$; and if this system matrix represents an asymptotically stable system, \tilde{x} will converge to zero for any value of $\tilde{x}(0)$. In other words, $\tilde{x}(k)$ will converge toward $x(k)$ regardless of the value of $\tilde{x}(0)$ and could do so faster than the normal (open-loop) motion of $x(k)$ if the estimator gain, L_p , were large enough so that the roots of $\Phi - L_p H$ are sufficiently fast. In an actual implementation, $\tilde{x}(k)$ will not equal $x(k)$ because the model is not perfect, there are unmodelled disturbances, and the sensor has some errors and added noise. However, typically the sensed quantity and L_p can be chosen so that the system is stable and the error is acceptably small.

To find the value of L_p , we take the same approach that we did when designing the control law. First, specify the desired estimator pole locations in the z -plane to obtain the desired estimator characteristic equation,

$$(z - \beta_1)(z - \beta_2) \cdots (z - \beta_n) = 0, \quad (8.25)$$

where the β 's are the desired estimator pole locations⁴ and represent how fast the estimator state vector converges toward the plant state vector. Then form the characteristic equation from the estimator-error equation (8.24),

$$|zI - \Phi + L_p H| = 0. \quad (8.26)$$

⁴ The following sections discuss how one should select these poles in relation to the control poles and how both sets of poles appear in the combined system. The issue was also discussed for the continuous case in Section 2.6.2.

Equations (8.25) and (8.26) must be identical. Therefore, the coefficient of each power of z must be the same, and, just as in the control case, we obtain n equations in n unknown elements of \mathbf{L}_p for an n th-order system.

◆ Example 8.4 Estimator Design for Satellite Attitude

Construct an estimator for the same case as in Example 8.1, where the measurement is the position state element, x_1 , so that $\mathbf{H} = [1 \ 0]$ as given by Eq. (4.47). Pick the desired poles of the estimator to be at $\zeta = 0.4 \pm j0.4$ so that the s -plane poles have $\zeta \geq 0.6$ and ω_n is about three times faster than the selected control poles (see Fig. 8.1).

Solution. The desired characteristic equation is then (approximately)

$$z^2 - 0.8z + 0.32 = 0. \quad (8.27)$$

and the evaluation of Eq. (8.26) for any estimator gain \mathbf{L}_p leads to

$$z \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] - \left[\begin{array}{cc} 1 & T \\ 0 & 1 \end{array} \right] + \left[\begin{array}{c} L_{p1} \\ L_{p2} \end{array} \right] \left[\begin{array}{cc} 1 & 0 \end{array} \right] = 0$$

or

$$z^2 + (L_{p1} - 2)z + TL_{p2} + 1 - L_{p1} = 0. \quad (8.28)$$

Equating coefficients in Eqs. (8.27) and (8.28) with like powers of z , we obtain two simultaneous equations in the two unknown elements of \mathbf{L}_p

$$\begin{aligned} L_{p1} - 2 &= -0.8 \\ TL_{p2} + 1 - L_{p1} &= 0.32. \end{aligned}$$

which are easily solved for the coefficients and evaluated for $T = 0.1$ sec

$$L_{p1} = 1.2, \quad L_{p2} = \frac{0.52}{T} = 5.2. \quad (8.29)$$

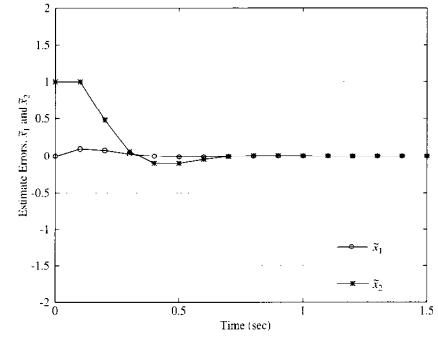
Thus the estimator algorithm would be Eq. (8.23) with L_p given by Eq. (8.29), and the equations to be coded in the computer are

$$\begin{aligned} \hat{x}_1(k+1) &= \hat{x}_1(k) + 0.1\ddot{x}_1(k) + 0.005u(k) + 1.2(y(k) - \hat{x}_1(k)) \\ \hat{x}_2(k+1) &= \hat{x}_2(k) + 0.1u(k) + 5.2(y(k) - \hat{x}_1(k)). \end{aligned}$$

Figure 8.5 shows the time history of the estimator error from Eq. (8.24) for the gains in Eq. (8.29) and with an initial error of 0 for the position estimate and 1 rad/sec for the velocity estimate.

The transient settling in \hat{x}_2 could be hastened by higher values of the gains, \mathbf{L}_p , that would result by selecting faster estimator poles, but this would occur at the expense of more response of both \hat{x}_1 and \hat{x}_2 to measurement noise.

Figure 8.5
Time history of the prediction estimator error



It is important to note that an initial estimator transient or, equivalently, the occurrence of an unmodelled input to the plant, can be a rare event. If the problem is one of regulation, the initial transient might be unimportant compared to the long-term performance of the estimator in the presence of noisy measurements. In the regulator case with very small plant disturbances, very slow poles (maybe even slower than the control poles) and their associated low estimator gains would give smaller estimate errors. Optimal selection of estimator gains based on the system's noise characteristics will be discussed in Chapter 9.

8.2.2 Observability

Given a desired set of estimator poles, is \mathbf{L}_p uniquely determined? It is, provided y is a scalar and the system is "observable." We might have an unobservable system if some of its modes do not appear at the given measurement. For example, if only derivatives of certain states are measured and these states do not affect the dynamics, a constant of integration is obscured. This situation occurs with a $1/s^2$ plant if only velocity is measured, for then it is impossible to deduce the initial condition of the position. For an oscillator, a velocity measurement is sufficient to estimate position because the acceleration, and consequently the velocity, observed are affected by position. A system with cycle delays can also be unobservable because the state elements representing the delays have no influence on the measurement and can therefore not be reconstructed by the measurements. A mathematical test for observability is stated in the next section.

as a necessary condition for the solution of Ackermann's formula; its proof is given in Section 8.7.

8.2.3 Pole Placement Using CACSD

If we take the transpose of the error-equation system matrix from Eq. (8.24)

$$[\Phi - \mathbf{L}_p \mathbf{H}]^T = \Phi^T - \mathbf{H}^T \mathbf{L}_p^T,$$

we see that the result is the same form as the system matrix $\Phi - \Gamma \mathbf{K}$ of the control problem from Eq. (8.6), and the mathematics of the solution is the same. Therefore, to solve the problem, we substitute Φ^T for Φ , \mathbf{H}^T for Γ and \mathbf{L}_p^T for \mathbf{K} , and use the control-design results. Making the substitutions in Eq. (8.15) results in Ackermann's estimator formula

$$\text{observability matrix } \mathbf{L}_p = \alpha_c(\Phi) \begin{bmatrix} \mathbf{H} \\ \mathbf{H}\Phi \\ \mathbf{H}\Phi^2 \\ \vdots \\ \mathbf{H}\Phi^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad (8.30)$$

where

$$\alpha_c(\Phi) = \Phi^n + \alpha_1 \Phi^{n-1} + \alpha_2 \Phi^{n-2} + \cdots + \alpha_n \mathbf{I}. \quad (8.31)$$

and the α_i 's are the coefficients of the desired characteristic equation, that is

$$\alpha_c(z) = (z - \beta_1)(z - \beta_2) \cdots (z - \beta_n) = z^n + \alpha_1 z^{n-1} + \cdots + \alpha_n. \quad (8.32)$$

The coefficient matrix with rows $\mathbf{H}\Phi^i$ is called the **observability matrix** and must be full rank for the matrix to be invertible and for the system to be observable.

For calculation of \mathbf{L}_p with MATLAB, either `acker` or `place` can be used by invoking the substitutions above. The same restrictions apply that existed for the control problem.

◆ Example 8.5 Predictor Estimator Pole Placement with MATLAB

Design an estimator for the satellite attitude-control system as in Example 8.4. Place the z -plane poles at $z = 0.4 \pm j0.4$.

Solution. The MATLAB statements

```
T = .1
Phi = [1 T; 0 1]
Gam = [T^2/2; T]
p = [.4+i*.4, .4-i*.4]
```

`[p = acker(Phi', H', p)'`

result in

$$\mathbf{L}_p = \begin{bmatrix} 1.2 \\ 5.2 \end{bmatrix}.$$

`place` would have given the same answer. ◆

8.2.4 Current Estimators

As was already noted, the previous form of the estimator equation (8.23) arrives at the state vector estimate $\tilde{\mathbf{x}}$ after receiving measurements up through $y(k-1)$. This means that the current value of control⁵ does not depend on the most current value of the observation and thus might not be as accurate as it could be. For high-order systems controlled with a slow computer or any time the sample periods are comparable to the computation time, this delay between the observation instant and the validity time of the control output can be a blessing because it allows time for the computer to complete the calculations. In many systems, however, the computation time required to evaluate Eq. (8.23) is quite short compared to the sample period, and the delay of almost a cycle between the measurement and the proper time to apply the resulting control calculation represents an unnecessary waste. Therefore, it is useful to construct an alternative estimator formulation that provides a current estimate $\hat{\mathbf{x}}$ based on the current measurement $y(k)$.⁶ Modifying Eq. (8.23) to yield this feature, we obtain

$$\hat{\mathbf{x}}(k) = \tilde{\mathbf{x}}(k) + \mathbf{L}_c [y(k) - \mathbf{H}\tilde{\mathbf{x}}(k)], \quad (8.33)$$

where $\tilde{\mathbf{x}}(k)$ is the predicted estimate based on a model prediction from the previous time estimate, that is

$$\tilde{\mathbf{x}}(k) = \Phi\tilde{\mathbf{x}}(k-1) + \Gamma u(k-1). \quad (8.34)$$

Control from this estimator cannot be implemented exactly because it is impossible to sample, perform calculations, and output with absolutely no time elapsed. However, the calculation of $u(k)$ based on Eq. (8.33) can be arranged to minimize computational delays by performing all calculations before the sample instant that do not directly depend on the $y(k)$ measurement. In Chapter 3, we also had a $u(k)$ that was dependent on $y(k)$ (see Table 3.1); and here, too, we organized the calculations to minimize the delay. If this latency in the implementation causes significant errors in the performance of the system compared to the analysis,

⁵ We plan to use $u(k) = -\mathbf{K}\tilde{\mathbf{x}}$ in place of $u(k) = -\mathbf{K}\mathbf{x}$.

⁶ This form of the equations is used in the Kalman filter, which is discussed in Chapter 9.

it could be modeled and accounted for in the estimator equations by using the results of Section 4.3.

To help understand the difference between the prediction and current form of estimation—that is, Equations (8.23) and (8.33)—it is useful to substitute Eq. (8.33) into Eq. (8.34). This results in

$$\tilde{x}(k+1) = [\Phi - \Phi L_c] \tilde{x}(k) + \Gamma u(k) + \Phi L_c [y(k) - H \tilde{x}(k)]. \quad (8.35)$$

Furthermore, the estimation-error equation for $\tilde{x}(k)$, obtained by subtracting Eq. (8.3) from Eq. (8.35), is

$$\tilde{x}(k+1) = [\Phi - \Phi L_c, H \tilde{x}(k)]. \quad (8.36)$$

By comparing Eqs. (8.35) with (8.23) and (8.36) with (8.24), we can conclude that \tilde{x} in the current estimator equation, (8.33), is the same quantity as \tilde{x} in the predictor estimator equation, (8.23), and that the estimator gain matrices are related by

$$L_p = \Phi L_c. \quad (8.37)$$

The relationship between the two estimates is further illuminated by writing Eqs. (8.33) and (8.34) as a block diagram, as in Fig. 8.6. It shows that \hat{x} and \tilde{x} represent different outputs of the same estimator system.

We can also determine the estimator-error equation for \tilde{x} by subtracting Eq. (8.33) from (8.3). The result is⁷

$$\tilde{x}(k+1) = [\Phi - L_c H \Phi] \tilde{x}(k). \quad (8.38)$$

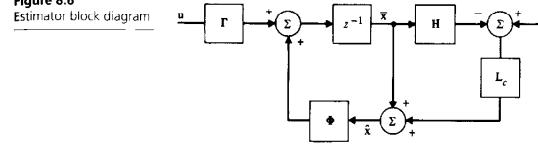
The two error equations, (8.36) and (8.38), can be shown to have the same roots, as should be the case because they simply represent the dynamics of different outputs of the same system. Therefore, we could use either form as the basis for computing the estimator gain, L_c . Using Eq. (8.38), we note that it is similar to Eq. (8.24) except that $H\Phi$ appears instead of H . To use Ackermann's formula for L_c , we use Eq. (8.30) with H replaced by $H\Phi$ and find

$$L_c = \alpha_c(\Phi) \begin{bmatrix} H\Phi \\ H\Phi^2 \\ H\Phi^3 \\ \vdots \\ H\Phi^n \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \quad (8.39)$$

where $\alpha_c(\Phi)$ is based on the desired root locations and is given by Eqs. (8.31) and (8.32). To use the control form of Ackermann's formula to perform the calculations, we take the transpose of $\Phi - L_c H\Phi$ and get $\Phi^T - \Phi^T H^T L_c^T$, which is the same form as the system matrix $\Phi - \Gamma K$ of the control problem. Therefore

⁷ Equation (8.21) defines x to be $x - \hat{x}$; however, we use x here to be $x - \tilde{x}$. In both cases, \hat{x} refers to the estimator error.

Figure 8.6
Estimator block diagram



substitutions Φ^T for Φ and $\Phi^T H^T$ for H yield L_c instead of K . Alternatively, we could compute L_p using Eq. (8.30) and then compute L_c using Eq. (8.37); that is

$$L_c = \Phi^{-1} L_p. \quad (8.40)$$

◆ Example 8.6 Current Estimator Pole Placement with MATLAB

Design a current estimator for the satellite attitude-control system as in Examples 8.4 and 8.5. Place the z -plane poles at $z = 0.4 + j0.4$ and compare the error response to an initial error in the velocity with that obtained in Example 8.4.

Solution. The MATLAB statements

```
T = .1
Phi = [1 1;0 1]
Gam = [T^2/2;T]
p = [.4+i*.4;4-i*.4]
Lc = acker(Phi',Phi'*H',p)'
```

result in

$$L_c = \begin{bmatrix} 0.68 \\ 5.2 \end{bmatrix}$$

Therefore, the estimator implementation using Eq. (8.33) in a way that reduces the computation delay as much as possible is, before sampling

$$\begin{aligned} \hat{x}_1(k) &= \hat{x}_1(k-1) + 0.008 u(k-1) + 0.1 \hat{x}_2(k-1), \\ \hat{x}_2(k) &= \hat{x}_2(k-1) + 0.1 u(k-1), \\ x'_1 &= (1 - 0.68) \hat{x}_1(k), \\ x'_2 &= (1 - 5.2) \hat{x}_1(k) + \hat{x}_2, \end{aligned}$$

and after sampling $y(k)$

$$\begin{aligned} \hat{x}_1(k) &= x'_1 + 0.68 y(k), \\ \hat{x}_2(k) &= x'_2 + 5.2 y(k). \end{aligned}$$

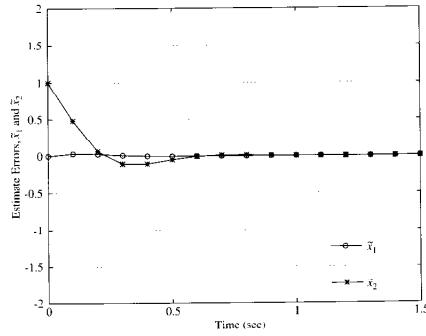
at which time the state vector estimate is available for the calculation of the control $u(k)$.

Figure 8.7 shows the time history of the estimator error equation from Eq. (8.38), again with an initial error of 0 for the position estimate and 1 rad/sec for the velocity estimate. The figure shows very similar results compared to the prediction estimator in Fig. 8.5; however, the current estimator implementation exhibits a response which is about a cycle faster.

If Φ is singular, as can happen with systems having time delays, neither Eqs. (8.39) nor (8.40) can be used. However, estimators can be designed for these systems as discussed in Section 8.6.

Note that we now have two estimates that could be used for control purposes, the predicted estimate $\hat{x}(k)$ from Eq. (8.23) and the current estimate $\hat{x}(k)$ from Eq. (8.33). The current estimate is the obvious choice because it is based on the most current value of the measurement, y . Its disadvantage is that it is out of date before the computer can complete the computation of Eqs. (8.33) and (8.5), thus creating a delay that is not accounted for in the design process, which will cause less damping in the implemented system than specified by the desired poles. The use of the predicted estimate for control eliminates the modeling error from the latency because it can be calculated using the measurement, $y(k-1)$, thus providing an entire sample period to complete the necessary calculations of $u(k)$ before its value is required. Generally, however, one should use the

Figure 8.7
Time history of current estimator error



current estimate because it provides the fastest response to unknown disturbances or measurement errors and thus better regulation of the desired output. Any deficiencies in the system response due to the latency from the computation lag that is found by simulation or experiment can be patched up with additional iterations on the desired pole locations or accounted for exactly by including computation delay in the plant model.

8.2.5 Reduced-Order Estimators

The estimators discussed so far are designed to reconstruct the entire state vector, given measurements of some of the state elements.⁸ One might therefore ask: Why bother to reconstruct the state elements that are measured directly? The answer is: You don't have to, although, when there is significant noise on the measurements, the estimator for the full state vector provides smoothing of the measured elements as well as reconstruction of the unmeasured state elements.

To pursue an estimator for only the unmeasured part of the state vector, let us partition the state vector into two parts: x_a is the portion directly measured, which is y , and x_b is the remaining portion to be estimated. The complete system description, like Eq. (8.3), becomes

$$\begin{bmatrix} x_a(k+1) \\ x_b(k+1) \end{bmatrix} = \begin{bmatrix} \Phi_{aa} & \Phi_{ab} \\ \Phi_{ba} & \Phi_{bb} \end{bmatrix} \begin{bmatrix} x_a(k) \\ x_b(k) \end{bmatrix} + \begin{bmatrix} \Gamma_a \\ \Gamma_b \end{bmatrix} u(k), \quad (8.41)$$

$$y(k) = [I \quad 0] \begin{bmatrix} x_a(k) \\ x_b(k) \end{bmatrix} \quad (8.42)$$

and the portion describing the dynamics of the unmeasured state elements is

$$x_b(k+1) = \Phi_{bb}x_b(k) + \underbrace{\Phi_{ba}x_a(k) + \Gamma_b u(k)}_{\text{known "input"}}, \quad (8.43)$$

where the right-hand two terms are known and can be considered as an input into the x_b dynamics. If we reorder the x_a portion of Eq. (8.41), we obtain

$$\underbrace{x_a(k+1) - \Phi_{aa}x_a(k) - \Gamma_a u(k)}_{\text{known "measurement"}!} = \Phi_{ab}x_b(k). \quad (8.44)$$

Note that this is a relationship between a measured quantity on the left and the unknown state vector on the right. Therefore, Eqs. (8.43) and (8.44) have the same relationship to the state vector x_b that the original equation, (8.3), had to the

⁸ Reduced-order estimators (or observers) were originally proposed by Luenberger (1964). This development follows Gopinath (1971).

entire state vector \mathbf{x} . Following this reasoning, we arrive at the desired estimator by making the following substitutions

$$\begin{aligned}\mathbf{x} &\leftarrow \mathbf{x}_a, \\ \Phi &\leftarrow \Phi_{ab}, \\ \Gamma u(k) &\leftarrow \Phi_{ba}x_a(k) + \Gamma_b u(k), \\ y(k) &\leftarrow x_a(k+1) - \Phi_{aa}x_a(k) - \Gamma_a u(k), \\ \mathbf{H} &\leftarrow \Phi_{ab},\end{aligned}$$

into the prediction estimator equations (8.23). Thus the reduced order estimator equations are

$$\begin{aligned}\hat{\mathbf{x}}_b(k+1) &= \Phi_{bb}\hat{\mathbf{x}}_b(k) + \Phi_{ba}x_a(k) + \Gamma_b u(k) \\ &\quad + \mathbf{L}_r[x_a(k+1) - \Phi_{aa}x_a(k) - \Gamma_a u(k) - \Phi_{ab}\hat{\mathbf{x}}_b(k)].\end{aligned}\quad (8.45)$$

Subtracting Eq. (8.45) from (8.43) yields the error equation

$$\hat{\mathbf{x}}_b(k+1) = [\Phi_{bb} - \mathbf{L}_r\Phi_{ab}]\hat{\mathbf{x}}_b(k), \quad (8.46)$$

and therefore \mathbf{L}_r is selected exactly as before, that is, (a) by picking roots of

$$|z\mathbf{I} - \Phi_{bb} + \mathbf{L}_r\Phi_{ab}| = \alpha_r(z) = 0 \quad (8.47)$$

to be in desirable locations, or (b) using Ackermann's formula

$$\mathbf{L}_r = \alpha_r(\Phi_{bb}) \begin{bmatrix} \Phi_{bb} \\ \Phi_{ba}\Phi_{bb} \\ \Phi_{ab}\Phi_{bb} \\ \vdots \\ \Phi_{ab}\Phi_{bb}^{n-2} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \quad (8.48)$$

We note here that Gopinath (1971) proved that if a full-order estimator as given by Eq. (8.23) exists, then the reduced-order estimator given by Eq. (8.45) also exists; that is, we can place the roots of Eq. (8.47) anywhere we choose by choice of \mathbf{L}_r .

◆ Example 8.7 Reduced-Order Estimator for Satellite Attitude

Determine a reduced-order estimator for the same case as in Examples 8.4 and 8.5.

Solution. We start out by partitioning the plant equations to fit the mold of Eqs. (8.41) and (8.42). This results in

$$\begin{bmatrix} \Phi_{aa} & \Phi_{ab} \\ \Phi_{ba} & \Phi_{bb} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} \Gamma_a \\ \Gamma_b \end{bmatrix} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} = \begin{bmatrix} 0.005 \\ 0.1 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \text{the measured position state } y \\ \text{the velocity to be estimated} \end{bmatrix}. \quad (8.49)$$

where Φ_{ab} , and so on, are all scalars. Therefore L_r is a scalar also, and there is only one estimator pole to pick, the pole corresponding to the speed at which the estimate of scalar velocity converges. From Eq. (8.47) we pick L_r from

$$z - 1 + L_r T = 0,$$

For this estimator to be about the same speed as the two previous estimator examples, which had two poles at $z = 0.4 \pm j0.4$, we will pick the pole at $z = 0.5$; therefore $L_r T - 1 = -0.5$ and $L_r = 5$. The estimator equation, (8.45), is

$$\begin{aligned}\hat{x}_b(k) &= \hat{x}_b(k-1) + 0.1 u(k-1) \\ &\quad + 5.0[y(k) - y(k-1) - 0.005 u(k-1) - (0.1)\hat{x}_b(k-1)].\end{aligned}\quad (8.50)$$

The implementation in a control computer would, before sampling, look something like

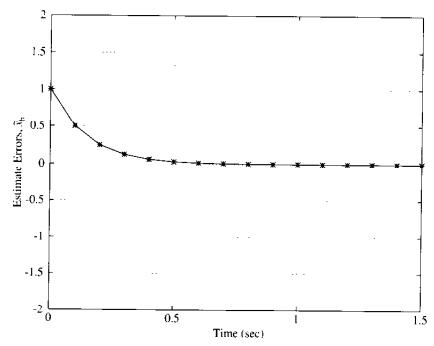
$$x' = 0.5 \hat{x}_b(k-1) + 0.075 u(k-1) - 5 y(k-1),$$

and after sampling

$$\hat{x}_b(k) = x' + 5 y(k).$$

Figure 8.8 shows the time history of the estimator-error equation (8.46) with an initial (velocity) estimate error of 1 rad/sec. The figure shows very similar results compared to the velocity element estimates in Figs. 8.5 and 8.7. Of course, there is no position estimate because this formulation assumes that the measurement is used directly without smoothing. ◆

Figure 8.8
Time history of
reduced-order estimator
error



8.3 Regulator Design: Combined Control Law and Estimator

If we take the control law (Section 8.1) and implement it, using an estimated state vector (Section 8.2), the control system can be completed. A schematic of such a system is shown in Fig. 8.9. However, because we designed the control law assuming that the true state, \mathbf{x} , was fed back instead of $\hat{\mathbf{x}}$ or $\tilde{\mathbf{x}}$, it is of interest to examine what effect this has on the system dynamics. We will see that it has no effect! The poles of the complete system consisting of the estimator feeding the control law will have the same poles as the two cases analyzed separately.

8.3.1 The Separation Principle

The control is now

$$u(k) = -\mathbf{K}\tilde{\mathbf{x}}(k)$$

and the controlled plant equation (8.6) becomes

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) - \Gamma\mathbf{K}\tilde{\mathbf{x}}(k), \quad (8.51)$$

which can also be written in terms of the estimator error using Eq. (8.21)

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) - \Gamma\mathbf{K}(\mathbf{x}(k) - \tilde{\mathbf{x}}(k)). \quad (8.52)$$

Combining this with the estimator-error equation (8.24)⁹ we obtain two coupled equations that describe the behavior of the complete system¹⁰

$$\begin{bmatrix} \tilde{\mathbf{x}}(k+1) \\ \mathbf{x}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi - \mathbf{L}_p\mathbf{H} & 0 \\ \Gamma\mathbf{K} & \Phi - \Gamma\mathbf{K} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}(k) \\ \mathbf{x}(k) \end{bmatrix}. \quad (8.53)$$

The characteristic equation is

$$\begin{vmatrix} z\mathbf{I} - \Phi + \mathbf{L}_p\mathbf{H} & 0 \\ \Gamma\mathbf{K} & z\mathbf{I} - \Phi + \Gamma\mathbf{K} \end{vmatrix} = 0 \quad (8.54)$$

which, because of the zero matrix in the upper right, can be written as

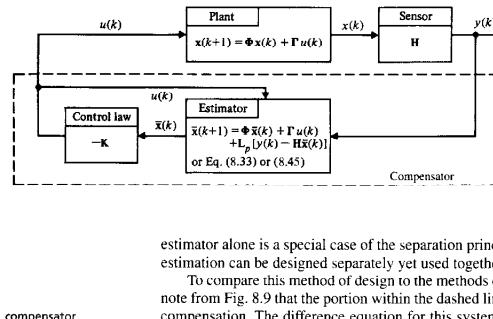
$$\begin{vmatrix} z\mathbf{I} - \Phi + \mathbf{L}_p\mathbf{H} & z\mathbf{I} - \Phi + \Gamma\mathbf{K} \end{vmatrix} = \alpha_c(z)\alpha_r(z) = 0. \quad (8.55)$$

In other words, the characteristic poles of the complete system consist of the combination of the estimator poles and the control poles that are unchanged from those obtained assuming actual state feedback. The fact that the combined control-estimator system has the same poles as those of the control alone and the

⁹ We show only the prediction estimator case. The other estimators lead to identical conclusions.

¹⁰ This description of the entire system does not apply if the estimator model is imperfect; see Section 11.5 for an analysis of that case.

Figure 8.9
Estimator and controller mechanization



estimator alone is a special case of the separation principle by which control and estimation can be designed separately yet used together.

To compare this method of design to the methods discussed in Chapter 7, we note from Fig. 8.9 that the portion within the dashed line corresponds to classical compensation. The difference equation for this system or "state-space designed compensator" is obtained by including the control feedback (because it is part of the compensation) in the estimator equations. Using Eq. (8.23) yields for the prediction estimator

$$\begin{aligned} \tilde{\mathbf{x}}(k) &= [\Phi - \Gamma\mathbf{K} - \mathbf{L}_p\mathbf{H}]\tilde{\mathbf{x}}(k-1) + \mathbf{L}_p\mathbf{y}(k-1), \\ u(k) &= -\mathbf{K}\tilde{\mathbf{x}}(k). \end{aligned} \quad (8.56)$$

and using Eq. (8.33) yields for the current estimator

$$\begin{aligned} \tilde{\mathbf{x}}(k) &= [\Phi - \Gamma\mathbf{K} - \mathbf{L}_c\mathbf{H}\Phi + \mathbf{L}_c\mathbf{H}\Gamma\mathbf{K}]\tilde{\mathbf{x}}(k-1) + \mathbf{L}_c\mathbf{y}(k), \\ u(k) &= -\mathbf{K}\tilde{\mathbf{x}}(k). \end{aligned} \quad (8.57)$$

The poles of the compensators above are obtained from, for Eq. (8.56),

$$|z\mathbf{I} - \Phi + \Gamma\mathbf{K} + \mathbf{L}_p\mathbf{H}| = 0 \quad (8.58)$$

and, for Eq. (8.57),

$$|z\mathbf{I} - \Phi + \Gamma\mathbf{K} + \mathbf{L}_c\mathbf{H}\Phi - \mathbf{L}_c\mathbf{H}\Gamma\mathbf{K}| = 0. \quad (8.59)$$

and are neither the control law poles, Eq. (8.8), nor the estimator poles, Eq. (8.25). These poles need not be determined during a state-space design effort, but can be of interest for comparison with compensators designed using the transform methods of Chapter 7.

If desired, Eq. (8.56) can be converted to transfer function-form using the same steps that were used in arriving at Eq. (4.64). This results in what was called

estimator pole selection

compensation in Chapter 7 and usually referred to as $D(z)$. For the prediction estimator, we find

$$\frac{U(z)}{Y(z)} = D_p(z) = -\mathbf{K}[z\mathbf{I} - \Phi + \Gamma\mathbf{K} + \mathbf{L}_p\mathbf{H}]^{-1}\mathbf{L}_p. \quad (8.60)$$

The $D(z)$ could also be found from Eq. (8.56) by using `tf` in MATLAB or, for the current estimator, from Eq. (8.57). Likewise, the transfer function for the reduced-order compensator is found by using the measured part of the state, x_a , directly in the control law and the estimated part, $\hat{x}_b(k)$, for the remainder. Thus, the control gain \mathbf{K} needs to be partitioned, so that

$$u(k) = [K_a \ K_b] \begin{bmatrix} x_a \\ x_b \end{bmatrix} \quad \text{where } \mathbf{K} = [K_a \ K_b]. \quad (8.61)$$

In the previous sections, we developed techniques to compute \mathbf{K} and \mathbf{L} , (which define the compensation), given the desired locations of the roots of the characteristic equations of the control and the estimator. We now know that these desired root locations will be the closed-loop system poles. The same meter sticks that applied to the classical design and were discussed in Section 7.2 also apply to picking these poles. In practice, when measurement noise is not an issue, it is convenient to pick the control poles to satisfy the performance specifications and actuator limitations, and then to pick the estimator poles somewhat faster (by a factor of 2 to 4) so that the total response is dominated by the response due to the slower control poles. It does not cost anything in terms of actuator hardware to increase the estimator gains (and hence speed of response) because they appear only in the computer. The upper limit to estimator speed of response is based on the behavior of sensor-noise rejection, which is the subject of Chapter 9.

In order to evaluate the full system response with the estimator in the loop, it is necessary to simulate both the real system and the estimator system as was formulated in Eq. (8.53). However, it is easier to see what is going on by using \hat{x} or $\dot{\hat{x}}$ in place of \hat{x} . The result for the predictor case using Eq. (8.56) is

$$\begin{bmatrix} x(k+1) \\ \hat{x}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & -\Gamma\mathbf{K} \\ \mathbf{L}_p\mathbf{H} & \Phi - \Gamma\mathbf{K} - \mathbf{L}_p\mathbf{H} \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix}, \quad (8.62)$$

the result for the current estimator using Eq. (8.57) is

$$\begin{bmatrix} x(k+1) \\ \hat{x}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & -\Gamma\mathbf{K} \\ \mathbf{L}_c\mathbf{H}\Phi & \Phi - \Gamma\mathbf{K} - \mathbf{L}_c\mathbf{H}\Phi \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix}, \quad (8.63)$$

and the result for the reduced-order estimator using Eq. (8.45) is

$$\begin{bmatrix} x(k+1) \\ \hat{x}_b(k+1) \end{bmatrix} = \begin{bmatrix} \Phi - \Gamma \begin{bmatrix} K_a & 0 \end{bmatrix} & -\Gamma\mathbf{K}_b \\ \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}_b(k) \end{bmatrix}. \quad (8.64)$$

where

$$\mathbf{A} = \mathbf{L}_r\mathbf{H}\Phi + \Phi_{ba}\mathbf{H} - \Gamma_b\mathbf{K}\mathbf{H} - \mathbf{L}_r\Phi_{aa}\mathbf{H}$$

and

$$\mathbf{B} = \Phi_{bb} - \Gamma_b\mathbf{K}_b - \mathbf{L}_r\Phi_{ab}$$

and where K_a and \mathbf{K}_b are partitions of \mathbf{K} according to the dimensions of x_a and x_b .

◆ Example 8.8 Compensation Based on the Predictor Estimator

Put together the full feedback control system based on the calculations already done in Examples 8.1 and 8.4, that is, using the prediction estimator. Use the control gain, $\mathbf{K} = [10 \ 3.5]$, and the estimator gain, $\mathbf{L}_p^T = [1.2 \ 5.2]$. Determine the $D(z)$ for comparison with a classical lead compensation. Plot the response of the system variables for an initial plant velocity of -1 rad/sec and zero for all other initial conditions. Comment on whether the responses are consistent with your expectations.

Solution. The compensation equations consist of Eq. (8.56) with the values of \mathbf{K} and \mathbf{L}_p plugged in and, being in difference-equation form, can be coded directly in a control computer. To find the transfer function form, we use `zp, m` (or Eq. (8.60)), to find¹¹

$$D_p(z) = -30.4 \frac{z - 0.825}{z - 0.2 \pm j0.557}. \quad (8.65)$$

There is a compensator zero near the two plant poles at $z = +1$ and there are two compensator poles considerably to the left. This is very similar to a classical lead compensator except that it has two poles instead of one. State-space design using a full-order estimator will always produce compensation that is the same order as the plant. Note that the difference equation that results from this $D(z)$ will have a one cycle delay between the input and output.

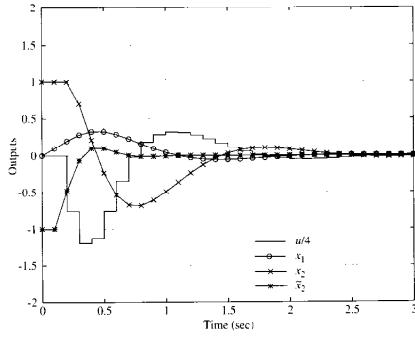
Figure 8.10 shows the response of the system and controller to the initial conditions. This could be thought of as a condition that would result from a sudden disturbance on the system. Note the estimator error at the beginning which decays in about 0.7 sec, consistent with the estimator poles. The overall system response is slower and has a settling time of about 2.5 sec, consistent with the control poles.

◆ Example 8.9 Compensation Based on the Current Estimator

Put together the full feedback control system based on the calculations already done in Examples 8.1 and 8.5, that is, using the current estimator. Use the control gain, $\mathbf{K} = [10 \ 3.5]$, and the estimator gain, $\mathbf{L}_c^T = [0.68 \ 5.2]$. Determine the $D(z)$ for comparison with a classical lead compensation. Plot the response of the system variables for an initial plant velocity of -1 rad/sec and zero for all other initial conditions. Comment on whether the responses are consistent with your expectations.

¹¹ Equation (8.60) includes a minus sign because it is the transfer function from $Y(s)$ to $U(s)$ rather than from $E(s)$, as is the normal convention used in Chapter 7.

Figure 8.10
Time histories of controlled system with prediction estimator, Example 8.8



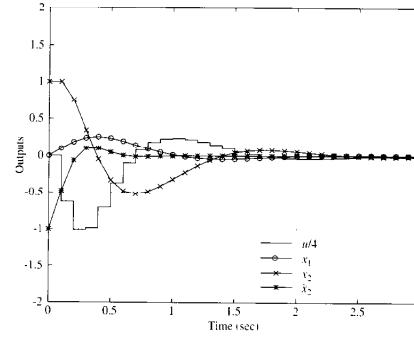
Solution. The compensation equations consist of Eq. (8.57) and, with the values of \mathbf{K} and \mathbf{L}_e plugged in, we find that

$$D_c(z) = -25.1 \frac{z(z - 0.792)}{z - 0.265 \pm j0.394}. \quad (8.66)$$

This compensation is very much like that from the prediction estimator; however, because of the extra z in the numerator, there is no 1 cycle delay between input and output. This faster cycle response required less lead from the compensation, as exhibited by the zero being further from $z = 1$.

Figure 8.11 shows the response of the controlled system to the initial conditions. Note the somewhat faster response as compared to Fig. 8.10 due to the more immediate use of the measured signal.

Figure 8.11
Time histories of controlled system with current estimator, Example 8.9



Solution. The compensation equations consist of Eqs. (8.50) and (8.61). With the values of \mathbf{K} and \mathbf{L}_e plugged in, we find after much algebra that

$$D_c(z) = -27.7 \frac{z - 0.8182}{z - 0.2375}. \quad (8.67)$$

Figure 8.12 shows the response of the system to the initial conditions. It is very similar to that of Example 8.9; the only notable difference is the first-order response of the estimator error, which slightly reduced the control usage.

This compensation now looks exactly like the classic lead compensation that was used often in Chapter 7 and would typically be used for a $1/s^2$ plant. A sketch of a root locus vs. K is given in Fig. 8.13. For this design a gain of 27.7 is now the variable K .

The closed-loop root locations corresponding to $K = 27.7$ are indicated by the triangles and lie on the two control roots at $z = 0.8 \pm j0.25$ and on the one estimator root at $z = 0.5$, as they should.

◆ Example 8.10 Compensation Based on the Reduced-Order Estimator

Put together the full feedback control system based on the calculations already done in Examples 8.1 and 8.6, that is, using the reduced-order estimator. Use the control gain, $\mathbf{K} = [10 \ 3.5]$, and the estimator gain, $\mathbf{L}_e^T = 5$. Determine the $D(z)$ for comparison with a classical lead compensation. Plot the response of the system variables for an initial plant velocity of -1 rad/sec and zero for all other initial conditions. Comment on whether the responses are consistent with your expectations.

Also, use the $D(z)$ to construct a root locus for this system and show where the desired root locations for the control and estimation lie on the locus.

The higher order compensators that resulted in Examples 8.8 and 8.9 have the benefit of more attenuation at high frequencies, thus reducing the sensitivity to measurement noise. This also follows from the fact that they provided a smoothed value of the measured output as well as reconstructing the velocity state variable. Full order estimators are also easier to implement because of the simpler matrix equations that result using $u = -\mathbf{K}\hat{\mathbf{x}}$ rather than the partitioned Eq. (8.61). As a result, reduced-order estimation is not often used in practice.

Figure 8.12
Time histories of system with reduced-order estimator, Example 8.10

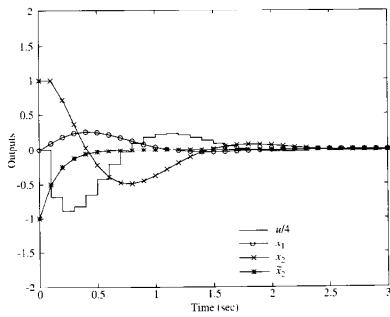
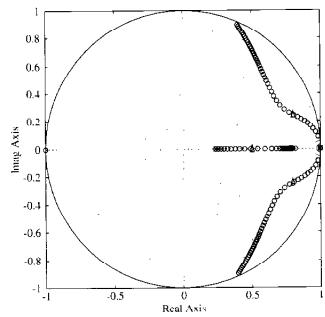


Figure 8.13
Sketch of the root locus for Example 8.10



8.3.2 Guidelines for Pole Placement

The selection criteria of the closed-loop control and estimator poles (or roots) have been encountered throughout the examples in Chapter 8 to this point. Also see the review of state-space design for continuous systems in Section 2.6 as well

as Franklin, Powell, and Emami-Naeini (1994), Sections 7.4 and 7.5.3. The key idea for control-pole selection is that one needs to pick the poles so that the design specifications are met while the use of control is kept to a level that is no more than needed to meet the specifications. This pole-selection criterion will keep the actuator sizes to a minimum, which helps to minimize the cost and weight of the control system. The relationships between various system specifications developed in Section 7.1 can be used as an aid in the pole-selection process. For high-order systems, it is sometimes helpful to use the ITAE or Bessel prototype design root locations as discussed in Section 7.4 in Franklin, Powell, and Emami-Naeini (1994). For the case where there is a lightly damped open-loop mode, a technique that minimizes control usage is simply to add damping with little or no change in frequency, a technique called **radial projection** that was demonstrated in Example 8.3.

The optimal design methods discussed in Chapter 9 can also be used to select pole locations. They are based on minimizing a cost function that consists of the weighted sum of squares of the state errors and control. The relative weightings between the state errors and control are varied by the designer in order to meet all the system specifications with the minimum control. Optimal methods can be applied to the SISO systems, which are the subject of this chapter, or to MIMO systems.

Estimator-error pole selection is a similar kind of design process to the control-pole selection process; however, the design trade-off is somewhat different. Fast poles in an estimator do not carry the penalty of a large actuator like they do in the control case because the large signals exist only in the computer. The penalty associated with fast estimator poles is that they create an increased sensitivity between sensor errors and estimation errors.

The key idea for estimator-error pole selection is that the estimation errors should be minimized with respect to the prevailing system disturbances and sensor noise. It is also convenient to keep the estimator poles faster than the control poles in order that the total system response is dominated by the control poles. Typically, we select well-damped estimator poles that are two to six times faster than the control poles in order to provide a response dominated by the control poles. For cases where this criterion produces estimation errors due to sensor noise that are unacceptable large, the poles can be slowed down to be less than two times the control poles; however, in this case the total response could be strongly influenced by the location of the estimator poles, thus coupling the estimator design with the control design and complicating the process.

In the optimal estimation discussion in Chapter 9 we will see that the optimal estimator error poles are proportional to the ratio between the plant model errors and the sensor errors. For an accurate plant model with small disturbances but large sensor errors, the optimal estimation is achieved with very low estimator gains (slow response) because the estimator is best served by relying primarily on the plant model. On the other hand, a system with a plant model that includes the possibility of large disturbances but with an accurate sensor achieves the best

estimation by using a large estimator gain (fast response) in order to use the sensed information to correct the model errors as quickly as possible.

8.4 Introduction of the Reference Input

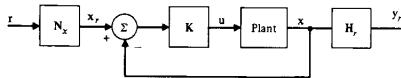
The compensation obtained by combining the control law studied in Section 8.1 with any of the estimators of Section 8.2 is a regulator design in that the goal was to drive all states to zero. We designed the characteristic equations of the control and the estimator to give satisfactory natural mode transients to initial conditions or disturbances, but no mention was made of how to structure a reference input or of the considerations necessary to obtain good transient response to reference inputs. To study these matters we will consider first how one introduces a reference input to the full-state feedback case, and then we will proceed to the case with estimators. We then turn to a discussion of output error command, a structure that occurs when the sensor is capable of providing only an error signal, for example, an attitude error from a gyro or the pointing error from a radar signal. The output error structure is also the one that results if one is designing the compensation using transfer-function methods and the reference input is structured according to Fig. 7.5, the typical case. It is, therefore, of interest to study this structure in order to understand the impact of its use on the dynamic response of the system. In conclusion, we will discuss the implications of this section's results and compare the relative advantages of the structure made possible by the state-space control/estimation approach with the classical approach.

8.4.1 Reference Inputs for Full-State Feedback

Let us first consider a reference input for a full-state feedback system as in Eq. (8.5). The structure is shown in Fig. 8.14 and consists of a state command matrix \mathbf{N}_r that defines the desired value of the state, \mathbf{x}_r . We wish to find \mathbf{N}_r so that some system output, $\mathbf{y}_r = \mathbf{H} \mathbf{x}$, is at a desired reference value. This desired output, \mathbf{y}_r , might not be the same quantity that we sense and feed to an estimator that has been called \mathbf{y} and determined by \mathbf{H} in the previous two sections.

Although so far in this book we have only considered systems with a single control input and single output (SISO), Chapter 9 considers the case of more

Figure 8.14
Block diagram for full-state feedback with reference input.



than one input and output (MIMO); and, therefore, we will allow for this in the development here and in the following subsection. We will, however, require that the number of inputs in \mathbf{u} and desired outputs in \mathbf{y}_r be the same.¹² The basic idea in determining \mathbf{N}_r is that it should transform the reference input, \mathbf{r} , to a reference state that is an equilibrium one for that \mathbf{r} . For example, for a step command to Example 8.1, $\mathbf{N}_r' = [1 \ 0]$; that is, we wish to command the position state element, but the velocity state element will be zero in steady state. For the double mass-spring system of Example 8.3, if we desire that $d = r$ —that is, that $\mathbf{H} = [1 \ 0 \ 0 \ 0]$ —then we should set $\mathbf{N}_r' = [1 \ 0 \ 1 \ 0]$ because that provides a state reference, \mathbf{x}_r , that, if matched by the actual state, is at equilibrium for the desired output.

More specifically, we have defined \mathbf{N}_r so that

$$\mathbf{N}_r \mathbf{r} = \mathbf{x}_r \quad \text{and} \quad \mathbf{u} = -\mathbf{K}(\mathbf{x} - \mathbf{x}_r). \quad (8.68)$$

If the system is Type 1 or higher and \mathbf{r} is a step, there will be no steady-state error, and the final state

$$\mathbf{x}(\infty) = \mathbf{x}_{ss} = \mathbf{x}_r.$$

For Type 0 systems, there will be an error because some control is required to maintain the system at the desired \mathbf{x}_r .

Often the designer has sufficient knowledge of the plant to know what the equilibrium state is for the desired output, in which case the determination of \mathbf{N}_r is complete. For complex plants, however, this can be difficult. In these cases, it is useful to solve for the equilibrium condition that satisfies $\mathbf{y}_r = \mathbf{r}$.¹³

In order for the solution to be valid for all system types, whether they require a steady-state control input or not, we will include the possibility of a steady-state control term that is proportional to the reference input step, that is,

$$\mathbf{u}_s = \mathbf{N}_s \mathbf{r}. \quad (8.69)$$

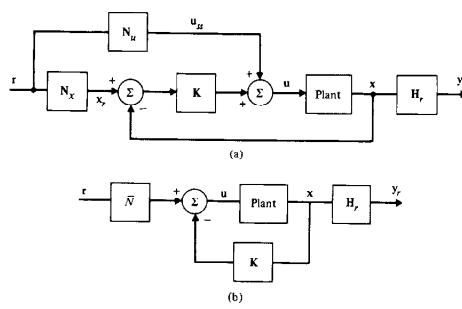
as shown in Fig. 8.15(a). The proportionality constant, \mathbf{N}_s , will be solved for in the formulation.

If the resulting \mathbf{u}_s is actually computed and implemented in the reference input structure, we refer to it as "feedforward," but the feedforward component of the input is often not used. Instead, the preferred method of providing for zero steady-state error is through integral control or bias estimation, which essentially replaces the \mathbf{u}_s in Fig. 8.15(a) with an integral that is an estimate of the steady-state control, a topic that is the subject of Section 8.5. In some cases, it is difficult to achieve a high enough bandwidth when replacing feedforward with integral control; therefore, feedforward is sometimes used to reduce the demands on the integral so that it need only provide the error in the feedforward control, thus

¹² This is the only case that has a unique and exact answer, although other situations have been studied.

¹³ See Franklin and Bryson (1978) for a more complete discussion and extensions of the ideas to inputs other than steps, often called "model following."

Figure 8.15
 (a) Block diagram and
 (b) modified block
 diagram for full-state
 feedback with reference
 input and feedforward



speeding up the system response. On the other hand, if the system is Type 1 or higher, the steady-state value of control for a step will be zero, and the solution will simply give us $N_u = 0$ and N_x , which defines the desired value of the state, x_r .

Continuing on then, the steady state requirements for the system are that

$$\begin{aligned} N_x r &= x_r = x_{ss}, \\ H_r x_{ss} &= y_r = r. \end{aligned} \quad (8.70)$$

which reduce to

$$H_r N_x r = r \quad \text{and} \quad H_r N_x = I. \quad (8.71)$$

Furthermore, we are assuming the system is at steady state; therefore,

$$x(k+1) = \Phi x(k) + \Gamma u(k) \Rightarrow x_{ss} = \Phi x_{ss} + \Gamma u_{ss},$$

or

$$(\Phi - I)x_{ss} + \Gamma u_{ss} = 0;$$

and, from Eqs. (8.69) and (8.70)

$$(\Phi - I)N_x r + \Gamma N_u r = 0.$$

which reduces to

$$(\Phi - I)N_x + \Gamma N_u = 0. \quad (8.72)$$

Collecting Eqs. (8.71) and (8.72) into one matrix equation

$$\begin{bmatrix} \Phi - I & \Gamma \\ H_r & 0 \end{bmatrix} \begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

yields the desired result¹⁴

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} \Phi - I & \Gamma \\ H_r & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix}. \quad (8.73)$$

It is also possible to enter the reference input after the gain multiplication according to Fig. 8.15(b) by combining N_x and N_u according to

$$\bar{N} = N_u + KN_x. \quad (8.74)$$

Calculation of N_x and N_u can be carried out by the MATLAB function refi.m contained in the Digital Control Toolbox.

◆ Example 8.11 Reference Input for the Mass-Spring System

Compute the reference input quantities for the system of Example 8.3 where it is desired to command d to a new value. Compare the two structures (a) and (b) in Fig. 8.15.

Solution. The state was defined to be $x = [d \quad \dot{d} \quad y \quad \dot{y}]^T$. Therefore, to command a desired value of d , $H_r = [1 \quad 0 \quad 0 \quad 0]$ and the evaluation of Eq. (8.73) leads to $N_x^T = [1 \quad 0 \quad 1 \quad 0]$ and $N_u = 0$, as expected by inspecting the physical system shown in Fig. A.8. The fact that $N_u = 0$ makes sense because this system is in equilibrium without the need of any control as long as $d = y$, which is ensured by the resulting value of N_x . In fact, N_u will be zero for any system of Type 1 or higher, as already discussed. The elements of N_x do not depend on specific values of any parameters in the plant and are therefore not sensitive to modeling errors of the plant.

Use of Eq. (8.74) leads to $\bar{N} = 0.005$ using the K from Eq. (8.18) and $\bar{N} = 0.011$ using the K from Eq. (8.19). However, note that this input structure can be very sensitive to errors in K . In this particular example, \bar{N} is the result of a difference of two numbers (K_1 and K_2) that are close to each other in absolute value, extremely close for the first case (which also exhibited poor response), thus producing an extreme sensitivity. Specifically, if one of the elements of K in Eq. (8.18) were in error by 1%, the resulting error in \bar{N} would be 120%! To avoid the high sensitivity for cases like this, it is advisable to structure the reference input as in Fig. 8.15(a).

This example shows that there are some systems where it is better to use the structure of Fig. 8.15(a). However, most cases do not exhibit this sensitivity and Fig. 8.15(b) is preferred due to its simplicity.

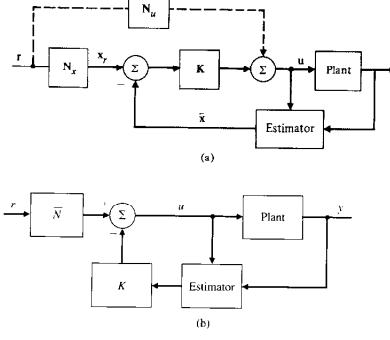
¹⁴ This cannot be solved if the plant has a zero at $z = 1$.

If the system in the example had been Type 0, we would have found that N_u was nonzero and that its value was inversely proportional to the plant gain. However, the plant gain can vary considerably in practice; therefore, designers usually choose not to use any feedforward for Type 0 systems, relying instead on the feedback to keep errors acceptably small or by implementing integral control as discussed in Section 8.5. If this is the desired action for a control design using state space, the designer can simply ignore the computed value of N_u and solely rely on N_x for guidance on how to command the state vector to the desired values.

8.4.2 Reference Inputs with Estimators: The State-Command Structure

The same ideas can be applied to the case where the estimator is used to supply a state estimate for control in place of the actual state feedback. However, it is important to structure the system so that the control u that is applied to the plant is also applied to the estimator as shown in Fig. 8.16. This means that Eqs. (8.56) and (8.57) should *not* be used in the estimator box in Fig. 8.16 because they were based on control feedback that did not include the reference input. The basic idea of the estimator that resulted in the structure of Fig. 8.4 is to drive the plant model in the estimator with the *same* inputs that are applied to the actual plant.

Figure 8.16
Block diagrams for the best reference input structure with estimators: the state command structure. (a) as derived, (b) simplified



thus minimizing estimation errors. Therefore, the form of the estimator given by Eq. (8.23) should be used with $u(k)$ as shown by Fig. 8.16, that is

$$\mathbf{u}(k) = -\mathbf{K}(\hat{\mathbf{x}}(k) - \mathbf{x}_r) + \mathbf{N}_x \mathbf{r} = -\mathbf{K}\hat{\mathbf{x}}(k) + \bar{\mathbf{N}}\mathbf{r}, \quad (8.75)$$

or, with the current estimator, Eq. (8.33) is used with

$$\mathbf{u}(k) = -\mathbf{K}(\hat{\mathbf{x}}(k) - \mathbf{x}_r) + \mathbf{N}_x \mathbf{r} = -\mathbf{K}\hat{\mathbf{x}}(k) + \bar{\mathbf{N}}\mathbf{r}, \quad (8.76)$$

or, for the reduced-order estimator, Eq. (8.45) should be used with

$$\mathbf{u}(k) = -[\mathbf{K}_u \quad \mathbf{K}_p] \begin{bmatrix} \frac{\mathbf{x}_u}{\hat{\mathbf{x}}_u} \\ \frac{\mathbf{x}_p}{\hat{\mathbf{x}}_p} \end{bmatrix} - \mathbf{x}_r + [\mathbf{N}_u \quad \mathbf{N}_p] \begin{bmatrix} \frac{\mathbf{x}_u}{\hat{\mathbf{x}}_u} \\ \frac{\mathbf{x}_p}{\hat{\mathbf{x}}_p} \end{bmatrix} + \bar{\mathbf{N}}\mathbf{r}. \quad (8.77)$$

Under ideal conditions where the model in the estimator is perfect and the input \mathbf{u} applied to plant and estimator is identical, no estimator error will be excited. We use the feedback to the estimator through y only to correct for imperfections in the estimator model, input scale factor errors, and unknown plant disturbances.

To analyze the response of a system, we must combine the estimator equations with the model of the system to be controlled. It is often useful to analyze the effect of disturbances, so the system equations (8.3) are augmented to include the disturbance, w , as

$$\begin{aligned} \mathbf{x}(k+1) &= \Phi \mathbf{x}(k) + \Gamma \mathbf{u}(k) + \Gamma_1 w(k), \\ \mathbf{y}(k) &= \mathbf{H} \mathbf{x}(k) + J w(k). \end{aligned} \quad (8.78)$$

For the predictor estimator, Eq. (8.62) is augmented with the input command and disturbance as follows:

$$\begin{bmatrix} \mathbf{x} \\ \hat{\mathbf{x}} \end{bmatrix}_{k+1} = \begin{bmatrix} \Phi & -\Gamma \mathbf{K} \\ \mathbf{L}_p \mathbf{H} & \Phi - \Gamma \mathbf{K} - \mathbf{L}_p \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \hat{\mathbf{x}} \end{bmatrix}_k + \begin{bmatrix} \Gamma \bar{\mathbf{N}} \\ \mathbf{F} \mathbf{N} \end{bmatrix} r(k) + \begin{bmatrix} \Gamma_1 \\ 0 \end{bmatrix} w(k). \quad (8.79)$$

Note that the term on the right with r introduces the command input in an identical way to both the plant and estimator equations. The term on the right with w introduces the disturbance into the plant only; the estimator is unaware of it.

It may be useful to inspect the performance of the system in terms of the desired output y_r , the control u , and the estimator error $\hat{\mathbf{x}}$. This can be accomplished with the output equation

$$\begin{bmatrix} y_r(k) \\ u(k) \\ \hat{\mathbf{x}}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{H}_r & 0 \\ 0 & -\mathbf{K} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \hat{\mathbf{x}}(k) \end{bmatrix}. \quad (8.80)$$

For the current estimator, the system equations are found by combining Eqs. (8.57) and (8.78), which yields

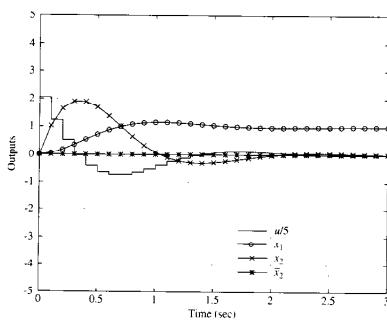
$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix}_{k+1} = \begin{bmatrix} \Phi & -\Gamma K \\ L_c H \Phi & \Phi - \Gamma K - L_c H \Phi \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}_k + \begin{bmatrix} \Gamma \bar{N} \\ \Gamma \bar{N} + L_c H \Gamma \bar{N} \end{bmatrix} r(k) + \begin{bmatrix} \Gamma_1 \\ L_c H \Gamma_1 \end{bmatrix} w(k). \quad (8.81)$$

◆ Example 8.12 Reference Input Command to Satellite Attitude

Determine the state-command structure for the system whose regulator was found in Example 8.8 and verify that its step response does not excite an estimator error.

Solution. Evaluation of Eq. (8.73) yields $N_e = [1 \ 0]^T$ and $N_w = 0$. Therefore, $\bar{N} = K_1$. The desired response of the system is obtained from Eqs. (8.79) and (8.80). Using step in MATLAB yields the unit step responses shown in Fig. 8.17. Note that the estimator error remains zero; thus the response is exactly the same as if no estimator were present. Note also that the structure shown in Fig. 8.16(b) does not allow us to represent the system in a simple, classical manner with the $D_p(z)$ from Example 8.8 placed in the upper path as in Fig. 7.5, nor does it allow us to place $D_p(z)$ in the lower feedback path. Rather, it is best to stay with the state-space description and enter the equations in the control computer based on Fig. 8.16. The response in Fig. 8.17 would also have resulted if using no estimator, a current estimator, or a reduced-order estimator.

Figure 8.17
Step-response time histories for Example 8.12



It is worthwhile to reflect on the fact that the combined system has poles that consist of the control *and* the estimator poles, as given by Eq. (8.55). The fact that the system response to an input structured as in Fig. 8.16(b) did not excite the estimator response means that the transfer function of this system had zeros that canceled the estimator poles. The determination of the structure in which the reference command is entered into a system can be viewed as one of "zero placement" and, in fact, it has been shown that it is possible to place the zeros of the closed loop transfer function at any arbitrary location [(Emami-Naeini and Franklin (1982)].

8.4.3 Output Error Command

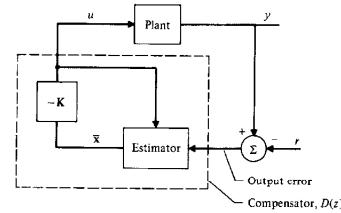
Another approach to the reference-input structure is to introduce the command only at the measured estimator input, as shown in Fig. 8.18. This solution is sometimes forced on the control designer because the sensor measures only the error. For example, many thermostats have an output that is the difference between the temperature to be controlled and the reference, or set-point, temperature. No absolute indication of the reference temperature is available to the controller. Likewise, some radar tracking systems have a reading that is proportional to the pointing error, and this signal alone must be used for control. In this case, the estimator equation (8.56) becomes

$$\dot{\bar{x}}(k+1) = (\Phi - \Gamma K - L_p H) \bar{x} + L_p(y - r),$$

and the system response can be determined by solving

$$\begin{bmatrix} \dot{x} \\ \dot{\bar{x}} \end{bmatrix}_{k+1} = \begin{bmatrix} \Phi & -\Gamma K \\ L_p H & \Phi - \Gamma K - L_p H \end{bmatrix} \begin{bmatrix} x \\ \bar{x} \end{bmatrix}_k + \begin{bmatrix} 0 \\ -L_p \end{bmatrix} r(k) + \begin{bmatrix} \Gamma_1 \\ 0 \end{bmatrix} w(k). \quad (8.82)$$

Figure 8.18
Reference input as an output-error command



Note that the command input r only enters the estimator; therefore, the plant and estimator do not see the same command and an estimator error will be excited.

◆ **Example 8.13 Output Command Structure with a Predictor Estimator**

Analyze the performance of Example 8.12 when using the output-command structure, specifically looking at the step response of the estimator error.

Solution. The system is analyzed using Eq. (8.82) and the output Eq. (8.80). The result is shown in Fig. 8.19. Note the estimator error response and that it causes a substantial increase in overshoot as compared to that of Fig. 8.17 (about 70% rather than 20%) and an increased use of control. Although this degradation could be reduced with faster estimator-error roots, there are limits due to the adverse effect on the estimator's sensitivity to measurement noise. Some of the degradation can be reduced by using a current or reduced order estimator because of their quicker response from the immediate use of the measured signal.

◆ **Example 8.14 Output Command Structure with a Reduced-Order Estimator**

Analyze the performance of Example 8.13 when using the output-command structure and a reduced-order estimator.

Figure 8.19
Output time histories for prediction estimator with output command, Example 8.13

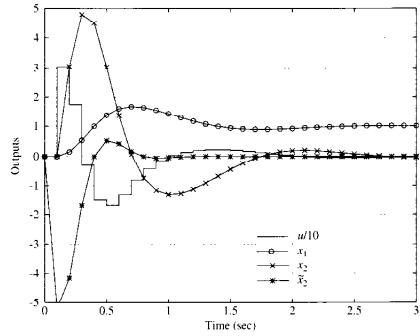
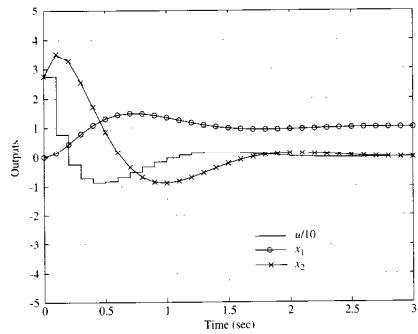


Figure 8.20
Output time histories for reduced-order estimator with output command, Example 8.14



Solution. The easiest way to analyze this system is by transform methods. The plant transfer function was analyzed in Section 4.3.1 and found to be

$$G(z) = \frac{T^2}{2} \frac{(z+1)}{(z-1)^2},$$

and the transfer function that arose from the reduced-order estimator is given by Eq. (8.67). The output time histories using these transfer functions are shown in Fig. 8.20. Note again that there is more overshoot (about 50%) than shown in Fig. 8.17 but not as much as the prediction estimator. The control usage for this case is also less than in Fig. 8.19, and it starts at the first cycle rather than being delayed by one cycle, as was the case in Fig. 8.19. This result produced a settling time for the reduced-order case that is about one sample period faster than the prediction estimator. A current estimator would produce results similar to this case because it, too, shares the feature of an immediate response to the measured output.

8.4.4 A Comparison of the Estimator Structure and Classical Methods

This section has demonstrated a key result: The controller structure based on state-space design of the control and estimator exposes a methodology for the introduction of the reference input in a way that produces a better response than that typically used with transfer-function design. The state-command input shown in Fig. 8.16 provides a faster response with less control usage than the

output-error-command scheme shown in Fig. 8.18, which is the typical transfer function structure. The reason for the advantage is that the state-command structure provides an immediate input to the plant and does not excite any modes of the compensator that degrade the response. Although this result has been specifically shown only for a simple second-order system, it applies to more complicated systems as well. In fact, for higher-order plants, higher-order compensators are often required with more modes for potential excitation.

It is not mandatory that the state-space representation and the state-estimator design approach be used in order to determine a structure that does not excite compensator modes. However, the determination is difficult using the transfer-function approach, especially in the MIMO case.

The advantages of using the transfer-function representation are that high-order systems are easier to trouble shoot, the designs are made robust with less effort, and experimental frequency-response data can be used to close a loop quickly without a time-consuming modeling effort. These advantages might not always warrant transforming to the state-space form in order to achieve easily the better reference input structure.

Although not discussed in this section using linear systems, the state-command structure allows for superior response of systems with nonlinear control characteristics, for example, saturation or on-off actuators. Whatever nonlinearity is present in the plant can usually be modeled to some degree in the estimator as well, thus reducing errors that would otherwise be excited by the nonlinearity.

◆ Example 8.15 Compensation Design for a System with a Resonance

A computer disk drive has a control system that commands the read head to move to specific tracks on the disk. One such system is described in Chapter 14. Although the dynamics between the torque and the head motion are primarily $G(s) = 1/s^2$, there are typically several resonances due to the arm's flexibility that limit the performance of the servo system. Here we wish to limit the flexibility to one resonance mode for simplicity. The transfer function is the same as the mass-spring system shown in Appendix A.4, although the derivation of it would involve rotational dynamics rather than the linear dynamics in Appendix A.4. (See Franklin, Powell, and Emami-Naeini (1994), Chapter 2, for more details.)

The system transfer function is

$$G(s) = \frac{1 \times 10^8}{s^2(s^2 + 2\xi_r\omega_r s + \omega_r^2)}$$

where the resonance frequency $\omega_r = 1 \text{ kHz}$ and the damping $\xi_r = 0.05$. Use a sample rate of 6 kHz and design control systems that have a rise time of 10 msec with an overshoot less than 15%.

- (a) Do the design using a state estimator and the state command structure,
- (b) evaluate the K and L from (a) using the output error structure whether or not they meet the specifications, and

(c) do a classical design with a lead compensator.

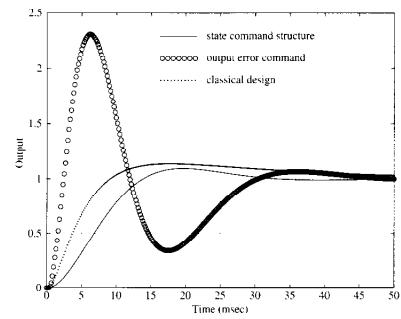
Iterate on whatever design parameters are appropriate for (a) and (c) to meet the design specifications.

Solution. Equation (2.16) indicates that a $t_r < 10 \text{ msec}$ would be met if $\omega_n > 180 \text{ rad/sec}$ and the $M_p < 15\%$ would be met if $\zeta > 0.5$.

(a) For the state space design, a good starting point would be with two of the desired poles at $\omega_n = 200 \text{ rad/sec}$ and $\zeta = 0.6$. Two more poles also need to be selected for this 4th order system; so let's pick them at $\omega = 1 \text{ kHz}$ and $\zeta = 0.06$. As discussed in Section 2.6, it is not wise to move poles any more than you have to. In this case, we have retained the same natural frequency and increased the damping slightly. For the estimator, a good starting point is to pick the natural frequencies twice those of the controller, that is, at 400 rad/sec and 2 kHz. The damping was selected to be 0.7 for both sets of poles in the estimator. The selected poles were converted to their discrete equivalents and used with acker to find K and L . Evaluation of the response for the state command structure is found using Eq. (8.79) in step and is shown in Fig. 8.21. We see that the rise time and overshoot specifications were met for this case and that the system settles to its final value at about 35 msec.

(b) Using Eq. (8.82) shows the response of the system with the same K and L for the output error command case and it is shown in the figure also. Note that the estimator excitation for this case caused an overshoot of 130% rather than the desired 15% and the system appears to be lightly damped even though its poles are in precisely the same place as case (a). It is interesting that the large overshoot was, in part, caused by using desired poles that increased the resonant mode damping from $\zeta = 0.05$ to 0.06. Had the damping

Figure 8.21
Disk read head response,
Example 8.15



been kept at 0.05, the overshoot would have been about 50% which is still excessive but significantly less. This illustrates the sensitivity of the system to the pole placement when using the output error command structure.

- (c) For the classical design, we note that the sample rate of 6 kHz is over 100 times faster than the desired closed loop natural frequency (200 rad/sec \approx 30 Hz). Therefore, the most expedient design method is to use the s -plane and convert the result to the discrete case when through. Furthermore, since the resonance frequency is significantly faster than the desired bandwidth (\approx 200 rad/sec), we can ignore the resonance for the first cut at the design and simply find a compensation for a $1/s^2$ system. Using frequency response ideas, we know that a lead compensation with a ratio of 25 between the zero and pole will yield a maximum increase in phase of about 60° (Fig. 2.17). We also know that a 60° PM (Section 2.4.4) will translate to a damping ratio of about 0.6 which will meet the overshoot specification. For a $1/s^2$ plant, the phase is 180° everywhere; therefore, the desired PM will be obtained if we place the lead compensation so that the maximum phase lead is at the desired crossover point (\approx 200 rad/sec). This is accomplished by placing the zero a factor of 5 below 200 (at 40 rad/sec) and the pole a factor of 5 above 200 (at 1000 rad/sec), thus producing

$$D(s) = K \frac{s + 40}{s + 1000}.$$

Using this with the $G(s)$ above in a Bode plot shows that the resonant mode does not affect the design significantly, the PM is met, and the desired crossover is achieved when $K = 8000$. To convert to the digital form, we invoke $c2d$ using the matched pole-zero approach and find that

$$D(z) = 7394 \frac{z + 0.9934}{z + 0.8465}.$$

The closed loop step response with $D(z)$ in the forward path is found using $step$ and shown in Fig. 8.21. It also meets the specifications, but a slow compensator mode was excited and the settling time of this system is considerably longer than the state command structure. The advantage of this approach is that the compensator is first order while the estimator approach (a) required a 4th order compensation.

8.5 Integral Control and Disturbance Estimation

Integral control is useful in eliminating the steady-state errors due to constant disturbances or reference input commands. Furthermore, most actual control systems are nonlinear and the input gain Γ and the state matrix Φ vary with time and/or the set point. The linear analysis which is the subject of this book pertains to perturbations about a set point of the nonlinear plant and the control u is a perturbation from a nominal value. The use of integral control eliminates the need to catalog nominal values or to reset the control. Rather, the integral term can be thought of as constantly calculating the value of the control required at the

set point to cause the error to go to zero. For these reasons, some form of integral control is typically included in most control systems. More generally, the external signals frequently include persistent deterministic components and the control engineer is required to design a controller which will force the steady-state error to be zero in the presence of such signals. A particular case is that of the disk drive servo required to follow a data track that is slightly off center so the reference signal is sinusoidal.

In the state-space design methods discussed so far, no mention has been made of integral control; nor have any of the design examples produced a compensation with an integral kind of behavior. In fact, state-space designs will not produce an integral action unless special steps are taken. There are two basic methods to force zero steady-state error in the presence of persistent signals: state augmentation also called internal signal model control and disturbance estimation. The idea of state augmentation for constant commands or disturbances was discussed for continuous systems in Section 2.6.5 and is essentially the same as the addition of an integral term that was discussed for transform design in Section 2.2.3. The generalization augments the state in a way that achieves zero steady-state error for a general class of reference and disturbance signals. Disturbance estimation provides the same effect based on estimation of the state of a model which could generate the external signals. We begin with the heuristic introduction of integral control.

8.5.1 Integral Control by State Augmentation

The idea is to add an integrator so as to obtain an integral of the error signal. This integrator will be physically implemented as part of the controller equations. We then feed back that integral along with the estimated or measured state as shown in Figs. 8.22 and 8.23 in a similar manner as before. To accomplish the design of the feedback gains for both the integral and the original state vector, we augment

Figure 8.22
Block diagram for integral control with full-state feedback

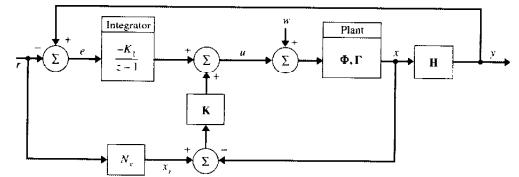
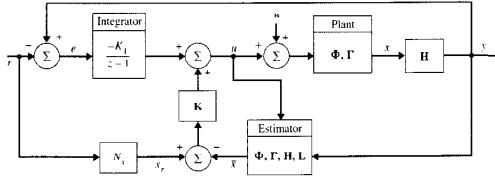


Figure 8.23
Block diagram for integral control with state estimation



the model of the plant with an integrator, thus adding an error integral output to the existing plant state output. This augmented model is then used as before to calculate the feedback control gains for the augmented state. More specifically, to the standard system

$$\begin{aligned} \mathbf{x}(k+1) &= \Phi\mathbf{x}(k) + \Gamma u(k) + \Gamma_1 w(k), \\ y(k) &= \mathbf{H}\mathbf{x}(k). \end{aligned}$$

we augment the state with x_i , the integral of the error, $e = y - r$. The discrete integral is simply a summation of all past values of $e(k)$ (Eq. 3.15), which results in the difference equation

$$x_i(k+1) = x_i(k) + e(k) = x_i(k) + \mathbf{H}\mathbf{x}(k) - r(k). \quad (8.83)$$

therefore arriving at the augmented plant model

$$\begin{bmatrix} x_i(k+1) \\ \mathbf{x}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{H} \\ 0 & \Phi \end{bmatrix} \begin{bmatrix} x_i(k) \\ \mathbf{x}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \Gamma \end{bmatrix} u(k) - \begin{bmatrix} 1 \\ 0 \end{bmatrix} r(k). \quad (8.84)$$

The control law, following Eq. (8.75), is

$$u(k) = -[K_i \quad \mathbf{K}] \begin{bmatrix} x_i(k) \\ \mathbf{x}(k) \end{bmatrix} + \mathbf{K} \mathbf{N}_1 r(k).$$

With this revised definition of the system, the design techniques already developed can be used directly for the control law design. Following Figs. 8.15(a) and 8.16(a), it would be implemented as shown in Fig. 8.22 for the full-state feedback case and as shown in Fig. 8.23 for the case where an estimator is used to provide $\hat{\mathbf{x}}$. The integral is replacing the feedforward term, N_1 , and has the additional role of eliminating errors due to w .

The estimator is based on the unaugmented model and is used to reconstruct the unaugmented state. It will be n th-order, where n is the order of the original system and requires the placement of n poles. On the other hand, the design of

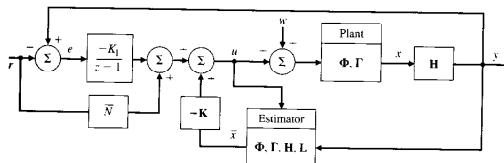
$[K_i \quad \mathbf{K}]$ requires the augmented system matrices (Eq. 8.84); therefore, there will be $n+1$ poles to be selected for this portion of the design.

If implemented as in Fig. 8.23, the addition of the extra pole for the integrator state element will typically lead to a deteriorated command input response compared to that obtained without integral control. While it is possible to iterate on the $n+1$ selected control poles until a satisfactory response is obtained, it is also possible to retain the reference response obtained from a non-integral-control design by placing a zero in the controller as shown in Fig. 8.24 so that it cancels the extra pole from the integrator. Note that the feedforward N_1 term in Fig. 8.22 has been replaced by \bar{N} which introduces a zero at $z_j = 1 - \frac{K_i}{K}$.¹⁵ Using the zero to cancel the closed-loop pole that was added for the integral state element cancels the excitation of that pole by command inputs. Note that this does not cancel the integral action, it merely eliminates the excitation of the extra root by command inputs. A change in the disturbance, w , will also excite the integral dynamics and the steady-state errors due to either constant disturbances or constant command inputs are eliminated. As always, the integrator output changes until its input, which is constructed to be the system error, is zero. The configuration of Fig. 8.24 can be changed to replace the feedforward of r to additional feedforward of e and modified feedback of r .

◆ Example 8.16 Integral Control for the Satellite Attitude Case

Determine the integral control structure and gains for the satellite attitude control problem using full state feedback. Place the control poles at $z = 0.8 \pm j0.25, 0.9$ and use a sample

Figure 8.24
Block diagram for integral control with full-state feedback and an added zero



¹⁵ In fact, the system of Fig. 8.22 has a zero at $1 - \frac{K_i}{K \lambda_i}$, so the selection of the zero location corresponds to a particular selection of N_1 .

period of $T = 0.1$ sec. Compute the time responses for a unit step in r at $t = 0$ sec and a step disturbance of 5 deg/sec^2 at $t = 2$ sec for (a) no integral control, (b) integral control as in Fig. 8.22, and (c) integral control as in Fig. 8.24 with the added zero at $z = +0.9$.

Solution.

- (a) This case is the same controller as used in Example 8.12. The only difference is that there is a step in the disturbance at 2 sec. Therefore, lsm must be used in order to allow the multiple inputs, the step in r at $t = 0$ and the disturbance step in w starting at 2 sec. The result is shown in Fig. 8.25. We see that the system responds identically to Fig. 8.17 for the first 2 sec, then there is a steady-state error in the output, x_1 , after the transient from the disturbance die out. The output error can be shown via the final value theorem to be $w/K_1 = 0.5$, thus the final value of x_1 is 1.5 instead of the commanded value of 1.
- (b) A steady-state error resulting from a disturbance is a classic motivation for the addition of integral control. The system model from Example 8.1 is augmented according to Eq. (8.84) and used with acker to obtain the augmented feedback gain matrix, $[K_f \ K] = [1.025 \ 13.74 \ 4.313]$ by asking for control roots at $z = 0.8 \pm j0.25, 0.9$. We saw from Example 8.12 that $N_d = [1 \ 0]^T$; therefore, the system design is complete and can be implemented according to Fig. 8.22. Use of lsm produces the response in Fig. 8.26(a). Note that the desired result has been obtained in that there is no longer a steady-state error in the output, x_1 . However, also note that it has come with a price: the behavior before the disturbance step has been degraded. More control was used, and the initial overshoot has increased from the original 20% to about 40% because of the additional root at $z = 0.9$.
- (c) The implementation is structured as shown in Fig. 8.24 which produces a zero at $z = 0.9$. All other parameters are the same as (b). Note that the resulting response in Fig. 8.26(b)

Figure 8.25
Response of satellite example to a unit reference input at $t = 0$ and a step disturbance at $t = 2$ sec with no integral control action, Example 8.16

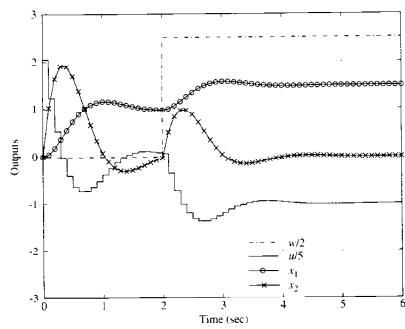
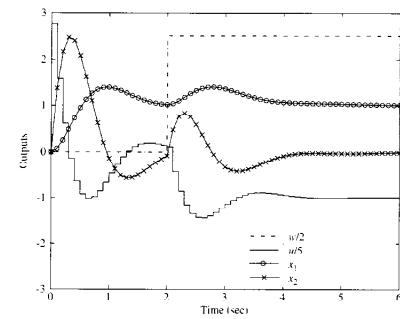
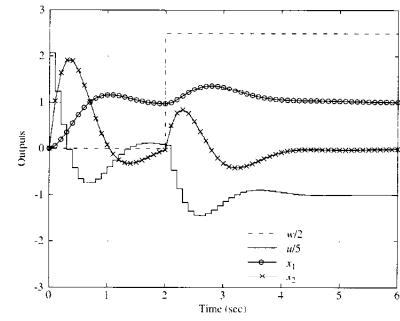


Figure 8.26
Response of satellite attitude to a unit step input at $t = 0$ and a step disturbance at $t = 2$ sec with integral control, Example 8.16. (a) As in Fig. 8.22, (b) with an added zero as in Fig. 8.24



(a)



(b)

before 2 sec is now identical to the case in Example 8.12 with no integral control; yet, the integral action successfully eliminates the steady-state error. \blacklozenge

It should be clear from the discussion and the example that the preferred implementation of integral control is given by Fig. 8.24 where the zero cancels the integrator closed-loop root.

8.5.2 Disturbance Estimation

An alternate approach to state augmentation is to estimate the disturbance signal in the estimator and then to use that estimate in the control law so as to force the error to zero as shown in Fig. 8.27. This is called **disturbance rejection**.

disturbance rejection

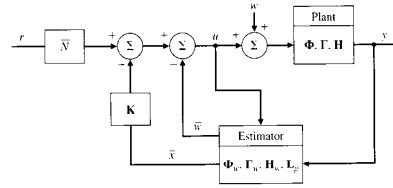
This approach yields results that are equivalent to integral control when the disturbance is a constant. After the estimate, \hat{w} , converges, the feedback of its value as shown in Fig. 8.27 will cancel the actual disturbance, w , and the system will behave in the steady state as if no disturbance were present. Therefore, the system will have no steady-state error assuming, of course, that the steady-state error was due to a disturbance described by the assumed equation used in the estimator. It is important to notice that while a disturbance may, in general, appear at any point in the plant equations, the control can apply a signal to cancel it only at the control input. To reconcile these facts, we introduce the “input equivalent” disturbance. This is a virtual signal applied at the control input which would produce the *same steady state output* at y as the actual disturbance does. Then, when the control applies the negative of the virtual disturbance, the effect of the real disturbance at the output is cancelled and the error is driven to zero. To obtain an estimate of the virtual disturbance, we build the estimator with the equations of the virtual disturbance included.

disturbance modeling

Disturbances other than constant biases can be modeled, included in the estimator equations, estimated along with the plant state, and their effect on errors eliminated in steady-state. If we assume the disturbance is a constant, the continuous model is quite simple:

$$\dot{w} = 0.$$

Figure 8.27
Block diagram for input disturbance rejection



A sinusoidal disturbance would have the model

$$\ddot{w} = -\omega_o^2 w,$$

or, in general, we could say that the disturbance obeys

$$\dot{x}_d = \mathbf{F}_d x_d$$

$$w(k) = \mathbf{H}_d x_d(k)$$

and the discrete model is given by

$$\dot{x}_d(k+1) = \Phi_d x_d(k) \quad (8.85)$$

$$w(k) = \mathbf{H}_d x_d(k) \quad (8.86)$$

where $\Phi_d = e^{\mathbf{F}_d T}$. For purposes of disturbance estimation, we augment the system model with the disturbance model, so Eqs. (8.85) and (8.86) become

$$\begin{bmatrix} x(k+1) \\ x_d(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma_d \mathbf{H}_d \\ 0 & \Phi_d \end{bmatrix} \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u(k), \quad (8.87)$$

$$y = [\mathbf{H} \ 0] \begin{bmatrix} x \\ x_d \end{bmatrix}, \quad (8.88)$$

which can be written as

$$\begin{bmatrix} x(k+1) \\ x_d(k+1) \end{bmatrix} = \Phi_x \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix} + \Gamma_u u(k)$$

$$y(k) = \mathbf{H}_w \begin{bmatrix} x(k) \\ x_d(k) \end{bmatrix}$$

In the particular case where the disturbance is a constant, these equations reduce to

$$\begin{bmatrix} x(k+1) \\ w(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma_d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ w(k) \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u(k), \quad (8.89)$$

$$y(k) = [\mathbf{H} \ 0] \begin{bmatrix} x(k) \\ w(k) \end{bmatrix}. \quad (8.90)$$

All the ideas of state estimation in Section 8.2 still apply, and any of the estimation methods can be used to reconstruct the state consisting of x and x_d , provided the system is observable.¹⁶ The computation of the required estimator gains is exactly as given in Section 8.2, the only change being that the system model is the augmented one given above by Φ_x and \mathbf{H}_w . Note from Fig. 8.27, however, that the control gain matrix, \mathbf{K} , is *not* obtained using the augmented \mathbf{F} and \mathbf{G} . In fact, the augmented system described by $[\Phi_w, \mathbf{H}_w]$ will always be uncontrollable! We have no influence over the value of w by means of the control

¹⁶ Observability requires that the virtual disturbance is “seen” at the plant output. Thus, if the disturbance is a constant, then the plant cannot have a zero from u to y at $z = 1$.

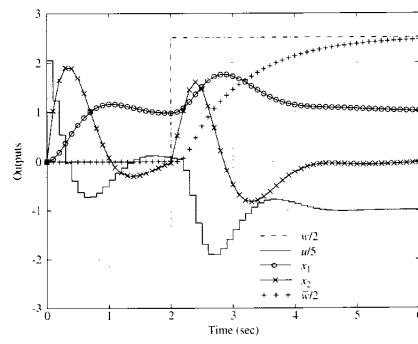
input, u , and must live with whatever value nature deals us; hence, the augmented system is uncontrollable. Our plan is not to control w , but to use the estimated value of w in a feedforward control scheme to eliminate its effect on steady-state errors. This basic idea works if w is a constant, a sinusoid, or any combination of functions that can be generated by a linear model. It works regardless of where the actual disturbance acts since the design is based on the virtual disturbance. The only constraint is that the disturbance state, x_d , be observable.

◆ Example 8.17 Bias Estimation and Rejection for the Satellite Attitude Case

Determine the bias rejection control structure and gains for the satellite attitude control problem. Place the control poles at $z = 0.8 \pm j0.25$ and use a sample period of $T = 0.1$ sec. Use a predictor estimator and place the estimator poles at $z = 0.4 \pm j0.4, 0.9$. Compute the time responses for a unit step in r at $t = 0$ sec and a step disturbance of 5 Deg/sec^2 at $t = 2$ sec and compare the results with the integral control in Example 8.16.

Solution. For purposes of finding the control gain, we use the unaugmented model as in Example 8.1 and, therefore, find the same value of $\mathbf{K} = [10.25 \quad 3.49]$. For purposes of designing the estimator, we augment the plant model according to Eqs. (8.89) and (8.90) and find that the desired poles yield $\mathbf{L}_p^T = [1.3 \quad 6.14 \quad 5.2]$. Structuring the control as in Fig. 8.27 and applying the inputs as specified above, lsrn yields the response as shown in Fig. 8.27. Note the similarity to Example 8.16 shown in Figs. 8.26(a) and (b). The disturbance rejection approach also eliminates the steady-state error. But also note the early response to the reference

Figure 8.28
Response of satellite example to a unit reference input at $t = 0$ and a step disturbance at $t = 2$ sec with bias estimation as in Fig. 8.27, Example 8.17



input. There is no increased overshoot as in Fig. 8.26(a); in fact, the response is identical to Fig. 8.26(b) up until the disturbance enters at $t = 2$ sec. Note further that the disturbance estimate, \hat{w} , approaches the actual disturbance value asymptotically. Notice that in this case the steady state error due to the reference input is made to be zero by the calculation of \bar{N} and is not robust to small parameter changes in the way provided by integral control. ◆

Example 8.17 shows that disturbance estimation can be used to estimate a constant disturbance input and then to use that estimate so as to reject the effect of the disturbance on steady-state errors. When the disturbance is a constant, this approach essentially duplicates the function of integral control. The following example shows how disturbance estimation can be used to estimate the value of the disturbance when it is a sinusoid. The estimate is used to cancel the effect of the disturbance, thus creating disturbance rejection.

◆ Example 8.18 Disturbance Rejection for a Spinning Satellite

For a spinning satellite, a disturbance torque from solar pressure acts on the system as a sinusoid at the spin frequency. The attitude dynamics become 4th order as the two axes are now coupled; however, for slow spin rates, the dynamics may be approximated to be $1/s^2$ as in Example 8.17.

Determine the disturbance rejection control structure and gains for the attitude control with a disturbance torque from solar pressure of 2 deg/sec^2 where the satellite is spinning at 15 rpm. Place the control poles at $z = 0.8 \pm j0.25$ and use a sample period of $T = 0.1$ sec, as before. Use a predictor estimator and place the estimator poles at $z = 0.4 \pm j0.4, 0.9 \pm j0.1$. The location of the estimator poles corresponding to the disturbance can be selected at a relatively slow frequency as above if the sinusoidal disturbance is known to be relatively stable in magnitude and phase. By picking those estimator poles at a slow frequency, the disturbance estimate will not respond much to other higher frequency disturbances.

Plot the time history of the disturbance, the estimate of the disturbance, and the system output to verify that, in steady-state, there is no error remaining from the disturbance. Put in a step command of 1° at $t = 5$ sec to verify that the input will not excite any unwanted estimator errors. Examine the roots of the 8th order system and explain what each of them represent.

Solution. The feedback for the unaugmented system is computed as in Example 8.17 to be

$$\mathbf{K} = [10.25 \quad 3.49].$$

The disturbance acts at the control input so there is no need for the concept of the virtual disturbance. It is modeled by choosing

$$\mathbf{F}_d = \begin{bmatrix} 0 & 1 \\ -\omega_p^2 & 0 \end{bmatrix}, \quad \mathbf{H}_d = \mathbf{I} \quad \text{and} \quad \mathbf{H}_d = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

in Eqs. (8.87) and (8.88). Use of acker with Φ_a and \mathbf{H}_a and the desired poles results in

$$\mathbf{L}_p^T = [1.375 \quad 6.807 \quad 8.391 \quad -6.374].$$

The time response of the system described by Fig. 8.27 is found by use of lsim where the state of the complete system consists of the augmented state as well as the estimate of the augmented state, an 8th order system. The feedback of \hat{w} can be accomplished by using an augmented feedback gain $\mathbf{K} = [\mathbf{K} \quad 1 \quad 0]$. Figure 8.29 shows the results. Note in the figure that the estimate takes about 4 sec to converge to the correct value and that there is a noticeable error in the output due to the disturbance until that time. The step at 5 sec has no effect on the estimate quality and therefore the response to the step is precisely as it was originally designed. Without the disturbance rejection, there would have been a steady sinusoidal error of about 0.2° superimposed on the output.

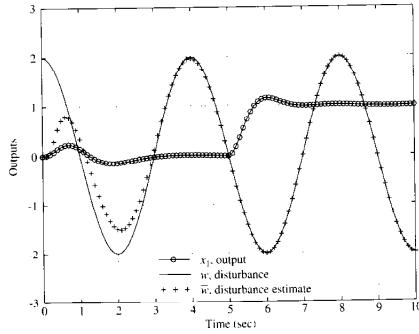
The roots of the closed loop 8th order system are:

$$z = 0.8 \pm 0.25j, \quad 0.4 \pm 0.4j, \quad 0.9 \pm 0.1j, \quad 0.988 \pm 0.156j.$$

The first 6 roots represent those selected in the control and estimation design. The last two represent the discrete equivalent of the pure oscillation at 15 rpm, which are unchanged by the feedback.

The previous example had a sinusoidal disturbance torque acting on the input to the system. It is also possible to have a measurement or sensor error that is sinusoidal in nature that one would like to eliminate as a source of error to the

Figure 8.29
Response of satellite to a sinusoidal input disturbance with disturbance rejection as in Fig. 8.27, Example 8.18



◆ Example 8.19 Satellite Attitude Control with Sinusoidal Sensor Disturbance Rejection

control system. The role of the estimator in this case is to use the contaminated measurement to reconstruct the error-free state for use in the control. The estimate of the sensor error is ignored by the controller.

◆ Example 8.19 Satellite Attitude Control with Sinusoidal Sensor Disturbance Rejection

When an attitude sensor on a spinning satellite is misaligned from a principal axis of inertia, the sensed attitude will have a sinusoidal component as the satellite naturally spins about its principal axis. Typically, it is desirable to spin about the principal axis; therefore, it is useful to estimate the magnitude and phase of the misalignment and to reject that component of the measurement.

Repeat the design from Example 8.18, but replace the sinusoidal disturbance torque with a sinusoidal measurement error of 0.3° at 15 rpm. Again include a step command of 1° at $t = 5$ sec. Place the control poles at $z = 0.8 \pm j0.25$, as in Example 8.18, and use a sample period of $T = 0.1$ sec. Use a predictor estimator and place the estimator poles at $z = 0.8 \pm j0.2, 0.95 \pm j0.05$.

Solution. As in Example 8.18, the feedback for the unaugmented system is $\mathbf{K} = [10.25 \quad 3.49]$. The output disturbance is modeled by augmenting the continuous plant with the matrices

$$\mathbf{F}_d = \begin{bmatrix} 0 & 1 \\ -\omega_p^2 & 0 \end{bmatrix}, \quad \mathbf{G}_d = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{H}_d = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

and the augmented continuous system is

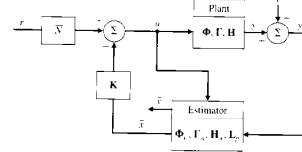
$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{F}\mathbf{x} + \mathbf{G}u \\ \dot{\mathbf{x}}_d &= \mathbf{F}_d\mathbf{x}_d \\ y &= [\mathbf{H} \quad \mathbf{H}_d]\mathbf{x} \end{aligned} \quad (8.91)$$

From these matrices, use of c2d will compute the discrete matrices Φ_a and \mathbf{H}_a from which, with the desired poles the estimator gain is computed

$$\mathbf{L}_p^T = [0.3899 \quad 0.1624 \quad 0.0855 \quad 0.7378].$$

The time response of the system described by Fig. 8.30 is found by use of lsim where the state of the complete system consists of the augmented state as well as the estimate of the

Figure 8.30
Block diagram for sensor disturbance rejection



augmented state, an 8th order system. The estimate of \bar{v} is ignored. Figure 8.31 shows the results. Note in the figure that the estimate takes about 3 sec to converge to the correct value and that there is a noticeable error in the output due to the measurement error until that time. The step at 5 sec has no effect on the estimate quality and therefore the response to the step is precisely as it was originally designed. Without the disturbance rejection, there would have been a steady sinusoidal error of about 0.2 as the controller attempted to follow the sinusoidal measurement.

reference following

A final example of the use of the estimator to achieve zero steady state error arises when it is desirable to track a reference signal with as little error as possible and it is known that the signal follows some persistent pattern. Since it would usually take some control effort to follow such a signal, the system would normally exhibit a following error of sufficient magnitude to produce the required control effort. This following error can be eliminated if the systematic pattern can be modeled and estimated, then used in a feedforward manner to produce the desired control effort. This is called **reference following**. The idea is the same as with disturbance rejection except that the error is now not the output only but the difference between the reference and the output. The idea again is to construct a virtual reference, ρ , at the control input which would produce the system error at the plant output, as shown in Figure 8.32(a). The feedback gain K is designed

Figure 8.31
Response of satellite to a sinusoidal sensor disturbance with disturbance rejection as in Fig. 8.30, Example 8.19

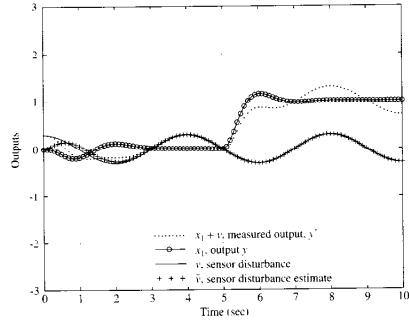
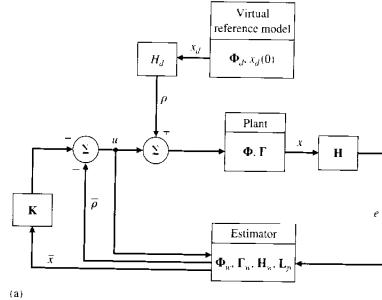
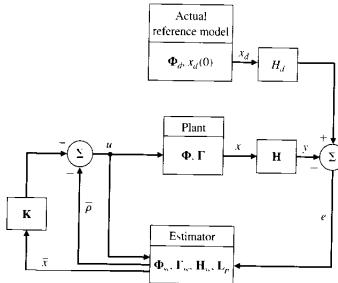


Figure 8.32
Block diagram for sensor disturbance following.
(a) The pretend model, and (b) the implementation model



(a)



(b)

using the unaugmented plant described by Φ and Γ , but the feedback signal is the system error e .

By estimating ρ and feeding that estimate into the plant with the control as shown in Fig. 8.32(a), the effect of ρ is eliminated in this virtual model and $e = 0$ in steady state. The actual situation that is implemented is shown in Fig. 8.32(b) where the reference is subtracted from the output to form the error, e , and the

estimate of the virtual reference, $\bar{\rho}$, is subtracted from the control. Therefore, e will behave as if $\bar{\rho}$ was canceled, so that in steady state, $e \equiv 0$, implying that y is following r .

◆ Example 8.20 Sinusoidal Sensor Disturbance Following of a Disk Drive

A computer disk read head must follow tracks on the disk. Typically, there is a small offset between the center of the tracks and the rotational center of the disk, thus producing a wobble in the tracks that the read head must follow. Assume the transfer function between the input command and the read head response is

$$G(s) = \frac{1000}{s^2}$$

and the disk spins at 5000 rpm. Design a control system that follows tracks on the disk with no error even though they may be off center by some small amount. Pick the gains so that the system has a rise time better than 3 msec and no more than a 15% overshoot. Use a sample rate of 2 kHz.

Solution. The disk head system is described by

$$\mathbf{F}_d = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 0 \\ 1000 \end{bmatrix}, \quad \text{and} \quad \mathbf{H}_d = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

We pretend that the disturbance is the same as in Example 8.18, that is

$$\mathbf{F}_u = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{H}_u = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

where $\omega_n = 5000$ rpm or 523.6 rad/sec. The difference here compared to Example 8.18 is that the actual reference is being subtracted from the output and we wish to eliminate its effect on e . In other words, we want y to follow r in Fig. 8.32(b).

Using \mathbf{F} and \mathbf{G} with c2d yields Φ and Γ for the disk head dynamics. To meet the rise time and overshoot specifications, a few iterations show that poles with a natural frequency of 1000 rad/sec and a 0.6 damping ratio, converted to the digital domain, produce the desired response. The resulting feedback gain is

$$\mathbf{K} = [736.54 \quad 1.0865].$$

But we wish to include the virtual reference in the control as

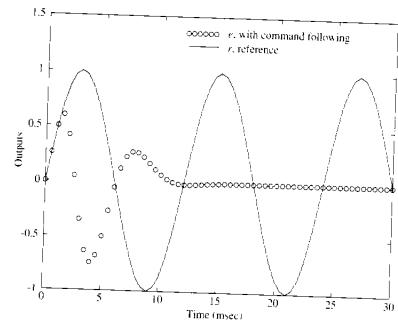
$$u = \mathbf{K}x - \bar{\rho}.$$

The estimator is constructed using the augmented system. A satisfactory design was found with equivalent s-plane poles that had natural frequencies of 1000 and 1200 rad/sec, both sets with $\zeta = 0.9$. Use of acker with Φ_u and \mathbf{H}_u results in $L_p^T = [3.9 \quad 1012.9 \quad -2.4 \quad 553.0]$.

The time response of the system described by Fig. 8.32(b) is found by use of isfm where the state consists of the augmented state as well as the estimate of the augmented state, an 8th order system.

The purpose of the feedforward is to provide a control that will produce $y = r$ and thus $e_{ss} \equiv 0$. Since we have an estimate of an input, $\bar{\rho}$, that will produce the track wobble which is the tracking reference, r , feedforward of $-\bar{\rho}$ causes $y = -r$. This produces the time response

Figure 8.33
Response of disk drive to sinusoidal reference following as in Fig. 8.32, Example 8.20



marked by small circles in Fig. 8.33. Note that, once $\bar{\rho}$ has converged, there is no noticeable following error. ◆

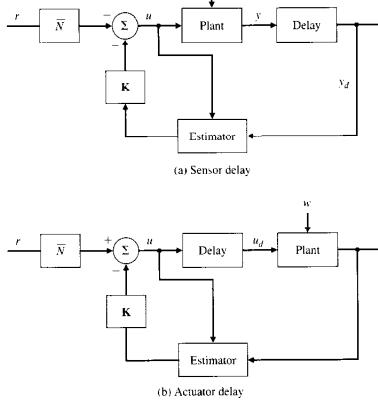
8.6 Effect of Delays

Many feedback control systems have a pure time delay, λ , imbedded in some part of the loop. A one cycle delay was analyzed in Section 7.3.2 by adding a z^{-1} to the system model and using the root locus method to show that the stability of the system was decreased when no change to the compensation was made. We can also analyze the effect of delays with frequency response methods as in Section 7.4 by reducing the phase by $\omega\lambda$. This analysis easily shows that, if no changes are made to the compensation, the phase margin and hence stability will be reduced. Either design method can be used to modify the compensation so the response is improved.

For state-space design, we saw in Section 4.3.4 that, for an actuator delay, one state element must be added to the model for each cycle of delay or fraction thereof. In other words, a delay of $0 < \lambda \leq T$ will require an increase in the order of the system model by 1, a delay of $T < \lambda \leq 2T$ will increase the order by 2, and so on. Using the pole placement approach, we can assign any desired pole locations to the system. Therefore, we are able to achieve the same closed-loop poles in a system with delays as one without delays; however, there are extra

delay model

Figure 8.34
System with delays,
(a) sensor delay,
(b) actuator delay



8.6.1 Sensor Delays

For the sensor delay, Fig. 8.34(a), an estimator can be used to reconstruct the entire state; therefore, the undelayed state is available for control. The system can be made to respond to command inputs, r , in exactly the same way that a system would respond without a sensor delay because the estimator sees the command through the feedforward of u and does not depend on the delayed output to detect it. Therefore, no estimator error is excited and the undelayed state estimate is accurate. On the other hand, a disturbance input, w , will not usually be seen by the estimator until the output, y_d , responds; therefore, the estimator error will be increased by the delay.

The model of a one cycle delay of a quantity y is

$$y_{1d}(k+1) = y(k), \quad (8.92)$$

where y_{1d} is the delayed version of y and is an additional state element that is added to the system model. The model for more than one cycle can be obtained by adding more similar equations and state elements. So, for two cycles of delay, we would add

$$y_{2d}(k+1) = y_{1d}(k), \quad (8.93)$$

to Eq. (8.92), where y_{2d} is one more state element and is the value of y that is delayed by two cycles. Therefore, for a system given by

$$\begin{aligned} \mathbf{x}(k+1) &= \Phi\mathbf{x}(k) + \Gamma u(k) \\ y(k) &= \mathbf{H}\mathbf{x}(k), \end{aligned}$$

the system model including a two-cycle sensor delay is

$$\begin{bmatrix} \mathbf{x}(k+1) \\ y_{1d}(k+1) \\ y_{2d}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \mathbf{0} & \mathbf{0} \\ \mathbf{H} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ y_{1d}(k) \\ y_{2d}(k) \end{bmatrix} + \begin{bmatrix} \Gamma \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} u(k), \quad (8.94)$$

$$y_d(k) = \begin{bmatrix} \mathbf{x}(k) \\ y_{1d}(k) \\ y_{2d}(k) \end{bmatrix}, \quad (8.95)$$

where y_d is the output y delayed by two cycles. Any number of cycles of delay can be achieved easily by using this scheme.

If a sensor had delay that was not an integer number of cycles, it would not influence the sampled value until the next sample instance. Therefore, sensor delays must be an integer number of samples.

Note that, due to the column of zeros, the augmented system matrix, Φ , in Eq. (8.94) will always be singular, a fact that will cause difficulties when calculating gains for a current estimator using MATLAB.¹⁷

◆ Example 8.21 Effect of a Sensor Delay on the Satellite Attitude Response

Examine the step response to a command input r of the satellite attitude control with one cycle of delay at the sensor. Place two of the control poles at $z = 0.8 \pm 0.25j$, as was the case for Examples 8.2 and 8.12, and the additional pole for the delay state at $z = 0$. Place the poles for a prediction estimator at $z = 0.4 \pm 0.4j$, as was the case for Examples 8.5 and 8.12, and the additional pole for the delay state at $z = 0$. Compare the results with Example 8.12, the same system step response without a delay.

¹⁷ We will see in Chapter 9 that a singular Φ matrix will also cause difficulty with dltqr.

Solution. The state-space equations for this system are (from Eqs. 8.94 and 8.95)

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ y_{1d}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ y_{1d}(k) \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \\ 0 \end{bmatrix} u(k),$$

$$y_d(k) = [0 \ 0 \ 1] \begin{bmatrix} x_1(k) \\ x_2(k) \\ y_{1d}(k) \end{bmatrix}.$$

Use of `acker` in MATLAB with the augmented Φ and Γ matrices as defined above with the desired control pole locations yields

$$K = [10.25 \ -3.4875 \ 0].$$

The first two elements are the exact same values that were obtained in Examples 8.2 and 8.12, and the zero third element means that the delayed state element is ignored. Thus the sensor delay has no effect on the controller.

Use of `acker` in MATLAB with the augmented Φ and Π matrices as defined above with the desired estimator pole locations yields

$$L_p = \begin{bmatrix} 1.72 \\ 5.2 \\ 1.2 \end{bmatrix}.$$

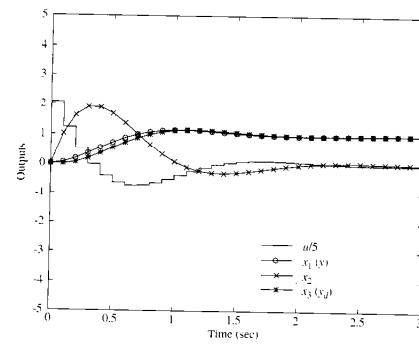
In evaluating the step response according to Eq. (8.79), \bar{N} needs to be evaluated with the new values of N_x and N_u . In this case, Eq. (8.73) shows that

$$N_x = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad N_u = 0, \quad \text{and} \quad \bar{N} = 10.25.$$

The use of `step` with the system defined by Eq. (8.79) produces the step response shown in Fig. 8.35. It includes the output y and the delayed output y_d . Note that the response of y is precisely the same as the response for Example 8.12 in Fig. 8.17 even though the only quantity fed back was the delayed output. This result follows because the current value of the state was estimated correctly due to the feedforward from the command input and the current value of the state was used by the controller. ◆

It is remarkable that a system with a sensor delay can respond to a command input in precisely the same way that the undelayed system would respond. This implies that the system closed-loop transfer function was such that the pole at $z = 0$ was not excited. The ability to design such a control system is enhanced by the state-space controller/estimator approach. It is important to note, however, that if the system had a disturbance, w , that was not fed into the estimator in the feedforward path, the delayed output measurement would cause a delayed estimator response and, thus, a degraded response of the system to that disturbance.

Figure 8.35
Step response for input command for Example 8.21



application to engine control

An example of a system in common use with a sensor delay is the fuel injection control for an automobile engine. Here, the sensed value of the fuel-air ratio in the exhaust is delayed by the piston motion and the time for the exhaust stream to reach the sensor. The primary disturbance is the motion of the throttle by the driver's foot; however, this motion can be sensed and used as feedforward to the estimator. Thus the estimator structure is capable of instantaneous rejection of the throttle disturbance in spite of the significant delay of the exhaust sensor. (See Fekete, 1995.)

8.6.2 Actuator Delays

The model for an actuator delay is derived in Section 4.3.4. The result is Eq. (4.79), which allows for any length delay, fractional or whole cycles. The poles of a system with actuator delays can be placed arbitrarily, just like the sensor delay case, and it makes sense to add a pole at $z = 0$ for each added delay state element. However, for input commands or disturbance inputs, the extra poles at $z = 0$ will be excited and the system response will be delayed by the actuator delay. This fact can be understood by examining Fig. 8.34(b). There is no way that the input r can influence the plant without passing through the delay. A disturbance w will influence the plant without delay; however, the feedback through the estimator will be delayed before it can counteract the disturbance, thus causing an increased sensitivity to disturbances.

◆ Example 8.22 Effect of an Actuator Delay on an Engine Speed Governor

An engine speed governor consists of an rpm sensor, usually a device that counts the time between the passage of teeth on a gear, and logic to determine the input to the fuel injector (diesel engines) or to the throttle actuator or spark advance (gasoline engines). Figure 8.36 shows the block diagram including the engine model, which is a simple first order lag with a time constant of around 3 sec so that $a = 0.3$. Fig. 8.36(a) shows the system without any delay; the ideal case. The actuator delay shown in Fig. 8.36(b) varies from engine to engine. For a gasoline engine with a throttle actuator, it is typically around a half engine cycle due to the piston motion (100 msec at 600 rpm). For diesel engines the actuator can be made to act more quickly; however, some manufacturers use injectors that have a full engine cycle delay (200 msec at 600 rpm). For this example, we will use a sample time of $T = 200$ msec and a one cycle actuator delay.

Investigate the response of the system for step input commands, r , and impulsive disturbances, w . Compare the results for control systems structured as in the three cases in Fig. 8.36. Pick the control gain so that the poles are at $z = 0.4$ for the ideal case and the estimator gain (if there is one) so that its pole is at $z = 0.2$. Select \bar{N} so that there is no steady-state error.

Solution. The step responses are shown in Fig. 8.37. For the ideal case (Fig. 8.36(a)), the gains were found to be

$$K = 9.303 \quad \text{and} \quad \bar{N} = 10.303.$$

The step response for $r = 1$ is shown in Fig. 8.37(a) and was obtained using step.

The model of the system with the delay in Fig. 8.36(b) can be accomplished with the aid of Eq. (4.79). For this system it reduces to

$$\begin{bmatrix} x(k+1) \\ u_d(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u_d(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} w(k).$$

$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u_d(k) \end{bmatrix}.$$

where Φ and Γ can be found using MATLAB's c2d, but can also be computed easily for this first order case to be

$$\Phi = e^{-aT} \quad \text{and} \quad \Gamma = 1 - e^{-aT}$$

Figure 8.37(b) shows the response of this system with the classical feedback as shown in Fig. 8.36(b) using the same gains as in (a). Not only is the response delayed by a cycle, but it has become oscillatory. A z-plane analysis of this system explains why: the roots are at $z = 0.47 \pm 0.57j$, which yield an equivalent s-plane damping of $\zeta = 0.35$.

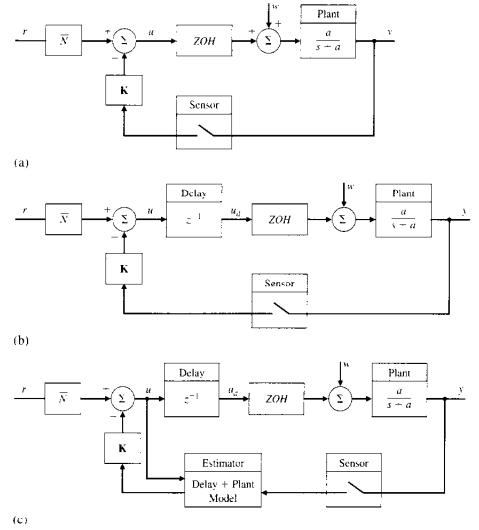
Figure 8.37(c) shows the response of the system with the estimator structure as shown in Fig. 8.36(c) using the desired control pole locations of $z = 0.4$. This yielded gains of

$$K = [8.76 \quad 0.54] \quad \text{and} \quad \bar{N} = 10.303.$$

The estimator poles were placed at $z = 0.2$, which yielded an estimator gain of

$$L_p = \begin{bmatrix} 0.742 \\ 0 \end{bmatrix}.$$

Figure 8.36
Engine speed control block diagrams, (a) the ideal feedback system without a delay and without an estimator, (b) the classical feedback with an actuator delay, (c) the estimator structure with an actuator delay



The response was obtained using Eq. (8.79) and is identical to the ideal case; however, it is delayed by one cycle. As previously discussed, this delay cannot be reduced if its source is due to the actuator.

The oscillatory response shown in Fig. 8.37(b) also can be eliminated by reducing the gain in the classical structure of Fig. 8.36(b), an easier solution than adding an estimator. Reducing the gain, K , from 9.3 to 4.0 produced the response in Fig. 8.37(d). While the oscillations have been removed, the response has slowed down. There is no value of gain in this simple structure that will duplicate the quality of the response obtained from the estimator system, although the difference is not large for this example.

Three of the cases were analyzed for an impulsive disturbance, w and shown in Fig. 8.38(a), b, and c). Qualitatively, the same kind of responses were obtained; however, the predictor estimator had an extra cycle of delay compared to the classical feedback system; therefore, there is little difference between the responses in Figs. 8.38(b) and 8.38(c). A current estimator

Figure 8.37
Responses to a step input, r , for Example 8.22. (a) Ideal case, no delay ($K = 9.3$), (b) classical feedback with delay ($K = 9.3$), (c) predictor estimator with delay, (d) classical feedback with delay ($K = 4.0$)

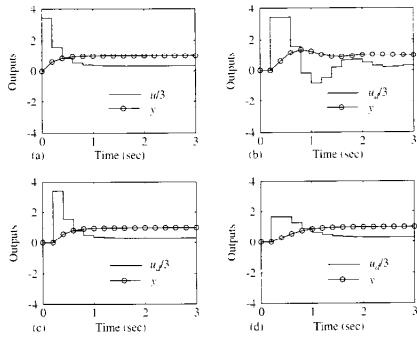
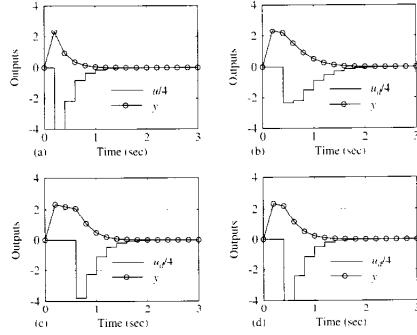


Figure 8.38
Responses to impulsive disturbance, w , for Example 8.22. (a) Ideal case, no delay ($K = 9.3$), (b) classical feedback with delay ($K = 4$), (c) predictor estimator w/delay ($K = 9.3$), (d) current estimator w/delay ($K = 9.3$)



formulation eliminates the extra delay and improves the response, which was obtained by evaluating Eq. (8.81) and is shown in Fig. 8.38(d).

Example 8.22 shows that an actuator delay will effect the response to a command input or a disturbance regardless of the control implementation. This is an inherent characteristic of an actuator delay for any system. However, use of an estimator including the delay minimizes its effect. Use of a current estimator provides the best response to a disturbance.

8.7 *Controllability and Observability

Controllability and observability are properties that describe structural features of a dynamic system.¹⁸ These concepts were explicitly identified and studied by Kalman (1960) and Kalman, Ho, and Narendra (1961). We will discuss only a few of the known results for linear, constant systems that have one input and one output.

We have encountered these concepts already in connection with design of control laws and estimator gains. We suggested in Section 8.1.3 that if the matrix C given by

$$C = [\Gamma : \Phi\Gamma : \dots : \Phi^{n-1}\Gamma]$$

is nonsingular, then by a transformation of the state we can convert the given description into the control canonical form and construct a control law such that the closed-loop characteristic equation can be given arbitrary (real) coefficients. We begin our discussion of controllability by making the definition (the first of three):

- I. The system (Φ, Γ) is controllable if for every n th-order polynomial $\alpha_i(z)$, there exists a control law $u = -Kx$ such that the characteristic polynomial of $\Phi - \Gamma K$ is $\alpha_i(z)$.

And, from the results of Section 8.1.3, we have the test:

- The pair (Φ, Γ) is controllable if and only if the rank of $C = [\Gamma : \Phi\Gamma : \dots : \Phi^{n-1}\Gamma]$ is n .

The idea of pole placement that is used above to define controllability is essentially a z -transform concept. A time-domain definition is the following:

¹⁸ This section contains material that may be omitted without loss of continuity.

II. The system (Φ, Γ) is controllable if for every \mathbf{x}_0 and \mathbf{x}_1 there is a finite N and a sequence of controls $u(0), u(1), \dots, u(N)$ such that if the system has state \mathbf{x}_k at $k = 0$, it is forced to state \mathbf{x}_1 at $k = N$.

In this definition we are considering the direct action of the control u on the state \mathbf{x} and are not concerned explicitly with modes or characteristic equations. Let us develop a test for controllability for definition **II**. The system equations are

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma u(k).$$

and, solving for a few steps, we find that if $\mathbf{x}(0) = \mathbf{x}_0$, then

$$\begin{aligned} \mathbf{x}(1) &= \Phi\mathbf{x}_0 + \Gamma u(0), \\ \mathbf{x}(2) &= \Phi\mathbf{x}(1) + \Gamma u(1) \\ &= \Phi^2\mathbf{x}_0 + \Phi\Gamma u(0) + \Gamma u(1), \\ &\vdots \\ \mathbf{x}(N) &= \Phi^N\mathbf{x}_0 + \sum_{j=0}^{N-1} \Phi^{N-1-j} \Gamma u(j) \\ &= \Phi^N\mathbf{x}_0 + [\Gamma : \Phi\Gamma : \dots : \Phi^{N-1}\Gamma] \begin{bmatrix} u(N-1) \\ \vdots \\ u(0) \end{bmatrix}. \end{aligned}$$

If $\mathbf{x}(N)$ is to equal \mathbf{x}_1 , then we must be able to solve the equations

$$[\Gamma : \Phi\Gamma : \dots : \Phi^{N-1}\Gamma] \begin{bmatrix} u(N-1) \\ u(N-2) \\ \vdots \\ u(0) \end{bmatrix} = \mathbf{x}_1 - \Phi^N\mathbf{x}_0.$$

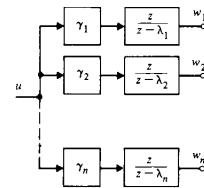
We have assumed that the dimension of the state, and hence the number of rows of the coefficient matrix of these equations, is n ; the number of columns is N . If N is less than n , we cannot possibly find a solution for every \mathbf{x}_1 . If, on the other hand, N is greater than n , we will add a column $\Phi^N\Gamma$, and so on. But, by the Cayley-Hamilton theorem, (see Appendix C), Φ^N is a linear combination of lower powers of Φ , and the new columns add no new rank. Therefore we have a solution, and our system is controllable by definition **II** if and only if the rank of C is n , exactly the same condition as we found for pole assignment!

Our final definition is closest to the structural character of controllability.

III. The system (Φ, Γ) is controllable if every mode in Φ is connected to the control input.

Because of the generality of modes, we will treat only the case of systems for which Φ can be transformed to diagonal form. (The double-integrator model for the satellite does not qualify.) Suppose we have a diagonal Φ_λ matrix and

Figure 8.39
Block diagram for a system with a diagonal Φ -matrix



corresponding input matrix Γ_i with elements γ_i . Then the structure is as shown in Fig. 8.39. By the definition, the input must be connected to each mode so that no γ_i is zero. However, this is not enough if the roots λ_i are not distinct. Suppose, for instance, $\lambda_1 = \lambda_2$. Then the equations in the first two states are

$$\begin{aligned} w_1(k+1) &= \lambda_1 w_1(k) + \gamma_1 u, \\ w_2(k+1) &= \lambda_1 w_2(k) + \gamma_2 u. \end{aligned}$$

If we now define $\xi = \gamma_2 w_1 - \gamma_1 w_2$, the equation in ξ is

$$\gamma_2 w_1(k+1) - \gamma_1 w_2(k+1) = \lambda_1 \gamma_2 w_1(k) - \lambda_1 \gamma_1 w_2(k) + \gamma_1 \gamma_2 u - \gamma_1 \gamma_2 u,$$

which is the same as

$$\xi(k+1) = \lambda_1 \xi(k).$$

The point is that if two characteristic roots are equal in a diagonal Φ , system with only one input, we effectively have a hidden mode that is not connected to the control, and the system is not controllable. Therefore, even in this simple case, we have two conditions for controllability:

1. All characteristic values of Φ_λ are distinct, and
2. No element of Γ_i is zero.

Now let us consider the controllability matrix of this diagonal system. By direct computation, we obtain

$$C = \begin{bmatrix} \gamma_1 & \gamma_1 \lambda_1 & \gamma_1 \lambda_1^{n-1} \\ \vdots & \gamma_2 \lambda_2 & \cdots & \vdots \\ \gamma_n & \gamma_n \lambda_n & \cdots & \gamma_n \lambda_n^{n-1} \end{bmatrix}$$

$$= \begin{bmatrix} \gamma_1 & 0 & 0 \\ 0 & \gamma_2 & \\ & \ddots & \\ 0 & & \gamma_n \end{bmatrix} \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \dots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \lambda_2^2 & \dots & \\ \vdots & \vdots & \vdots & & \\ 1 & \lambda_n & \lambda_n^2 & \dots & \lambda_n^{n-1} \end{bmatrix}.$$

The controllability matrix is a product of two terms, and \mathcal{C} is nonsingular if and only if each factor is nonsingular. The first term has a determinant that is the product of the γ_i , and the second term is nonsingular if and only if the λ_i are distinct! So once again we find that our definition of controllability leads to the same test: The matrix \mathcal{C} must be nonsingular. If \mathcal{C} is nonsingular, then we can assign the system poles by state feedback: we can drive the state to any part of the space in finite time, and we know that every mode is connected to the input.¹⁹

As our final remark on the topic of controllability we present a test that is an alternative to testing the rank (or determinant) of \mathcal{C} . This is the Rosenbrock-Hautus-Popov (RHP) test [see Rosenbrock (1970), Kailath (1979)]. The system (Φ, Γ) is controllable if the system of equations

$$\mathbf{v}[z\mathbf{I} - \Phi]^{-1}\Gamma = \mathbf{0}$$

has only the trivial solution $\mathbf{v}' = \mathbf{0}'$, or, equivalently

$$\text{rank}[z\mathbf{I} - \Phi]^{-1}\Gamma = n,$$

or there is no nonzero \mathbf{v}' such that

$$(i) \quad \mathbf{v}'\Phi = z\mathbf{v}', \quad (ii) \quad \mathbf{v}'\Gamma = 0.$$

This test is equivalent to the rank-of- \mathcal{C} test. It is easy to show that if such a \mathbf{v}' exists, then \mathcal{C} is singular. For if a nonzero \mathbf{v}' exists such that $\mathbf{v}'\Gamma = 0$ by (i), then, multiplying (i) by Γ on the right, we find

$$\mathbf{v}'\Phi\Gamma = z\mathbf{v}'\Gamma = 0.$$

Then, multiplying by $\Phi\Gamma$, we find

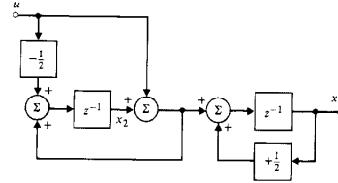
$$\mathbf{v}'\Phi^2\Gamma = z\mathbf{v}'\Phi\Gamma = 0,$$

and so on. Thus we derive $\mathbf{v}'\mathcal{C} = \mathbf{0}'$ has a nontrivial solution, \mathcal{C} is singular, and the system is not controllable. To show that a nontrivial \mathbf{v}' exists if \mathcal{C} is singular requires a bit more work and is omitted. See Kailath (1979).

We have given two pictures of an uncontrollable system. Either the input is not connected to a dynamic part physically or else two parallel parts have identical characteristic roots. The engineer should be aware of the existence of a third simple situation, illustrated in Fig. 8.40. Here the problem is that the mode at $z = \frac{1}{2}$ appears to be connected to the input but is masked by the zero in the preceding member; the result is an uncontrollable system. First we will confirm

¹⁹ Of course, we showed this only for Φ that can be made diagonal. The result is true for general Φ .

Figure 8.40
Block diagram of a simple uncontrollable system



this allegation by computing the determinant of the controllability matrix. The system matrices are

$$\Phi = \begin{bmatrix} +\frac{1}{2} & 1 \\ 0 & 1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix},$$

and

$$\mathcal{C} = [\Gamma \quad \Phi\Gamma] = \begin{bmatrix} 1 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix},$$

which is clearly singular. If we compute the transfer function from u to x_1 , we find

$$H(z) = \frac{z - \frac{1}{2}}{z - 1} \frac{1}{z - \frac{1}{2}} \quad (8.96)$$

$$= \frac{1}{z - 1}. \quad (8.97)$$

Because the natural mode at $z = \frac{1}{2}$ disappears, it is not connected to the input. Finally, if we consider the RHP test,

$$[z\mathbf{I} - \Phi \quad \Gamma] = \begin{bmatrix} z - \frac{1}{2} & -1 & 1 \\ 0 & z - 1 & \frac{1}{2} \end{bmatrix},$$

and let $z = \frac{1}{2}$, then we must test the rank of

$$\begin{bmatrix} 0 & -1 & 1 \\ 0 & -\frac{1}{2} & \frac{1}{2} \end{bmatrix},$$

which is clearly less than two, which means, again, uncontrollable. In conclusion, we have three definitions of controllability: pole assignment, state reachability, and mode coupling to the input. The definitions are equivalent, and the tests for any of these properties are found in the rank of the controllability matrix or in the rank of the input system matrix $[z\mathbf{I} - \Phi \quad \Gamma]$.

We have thus far discussed only controllability. The concept of observability

observability

observability definitions

is parallel to that of controllability, and most of the results thus far discussed can be transferred to statements about observability by the simple expedient of substituting the transposes Φ^T for Φ and \mathbf{H}^T for Γ . The result of these substitutions is a “dual” system. We have already seen an application of duality when we noticed that the conditions for the ability to select an observer gain \mathbf{L} to give the state-error dynamics an arbitrary characteristic equation were that (Φ^T, \mathbf{H}^T) must be controllable—and we were able to use the same Ackermann formula for estimator gain that we used for control gain. The other properties that are dual to controllability are

- OI.** The system (Φ, \mathbf{H}) is observable if for any n th-order polynomial $\sigma_c(z)$, there exists an estimator gain \mathbf{L} such that the characteristic equation of the state error of the estimator is $\sigma_c(z)$.
- OII.** The system (Φ, \mathbf{H}) is observable if for any $\mathbf{x}(0)$, there is a finite N such that $\mathbf{x}(0)$ can be computed from observation of $y(0), y(1), \dots, y(N-1)$.
- OIII.** The system (Φ, \mathbf{H}) is observable if every dynamic mode in Φ is connected to the output y via \mathbf{H} .

We will consider the development of a test for observability according to definition OII. The system is described by²⁰

$$\begin{aligned}\mathbf{x}(k+1) &= \Phi\mathbf{x}(k), & \mathbf{x}(0) &= \mathbf{x}_0, \\ y(k) &= \mathbf{H}\mathbf{x}(k);\end{aligned}$$

and successive outputs from $k = 0$ are

$$\begin{aligned}y(0) &= \mathbf{H}\mathbf{x}_0, \\ y(1) &= \mathbf{H}\mathbf{x}(1) = \mathbf{H}\Phi\mathbf{x}_0, \\ y(2) &= \mathbf{H}\mathbf{x}(2) = \mathbf{H}\Phi\mathbf{x}(1) = \mathbf{H}\Phi^2\mathbf{x}_0, \\ &\vdots \\ y(N-1) &= \mathbf{H}\Phi^{N-1}\mathbf{x}_0.\end{aligned}$$

In matrix form, these equations are

$$\begin{bmatrix} y(0) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}\Phi \\ \vdots \\ \mathbf{H}\Phi^{N-1} \end{bmatrix} \mathbf{x}_0.$$

20. Clearly the input is irrelevant here if we assume that all values of $u(k)$ are available in the computation of \mathbf{x}_0 . If some inputs, such as a disturbance w , are not available, we have a very different problem.

As we saw in the discussion of state controllability, new rows in these equations cannot be independent of previous rows if $N > n$ because of the Cayley-Hamilton theorem. Thus the test for observability is that the matrix

$$\mathcal{O} = \begin{bmatrix} \mathbf{H} \\ \mathbf{H}\Phi \\ \vdots \\ \mathbf{H}\Phi^{n-1} \end{bmatrix}$$

must be nonsingular. If we take the transpose of \mathcal{O} and let $\mathbf{H}^T = \Gamma$ and $\Phi^T = \Phi$, then we find the controllability matrix of (Φ, Γ) , another manifestation of duality.

8.8 Summary

- For any controllable system (Φ, Γ) of order n , there exists a discrete full state feedback control law (\mathbf{K}) that will place the n closed-loop poles at arbitrary locations. acker.m or place.m using Φ, Γ perform this function.
- $\mathcal{C} = [\Gamma \quad \Phi\Gamma \dots]$, the **controllability matrix**, must be of rank n , the order of the system, for the system to be controllable. ctrb.m performs this calculation.
- The general rule in selecting the desired pole locations is to move existing open-loop poles as little as possible in order to meet the system specifications.
- For any observable system (Φ, \mathbf{H}) of order n , there exists a discrete estimator with gain \mathbf{L} that will place the n estimator error equation poles at arbitrary locations. acker.m or place.m using Φ^T, \mathbf{H}^T calculates \mathcal{L} .
- $\mathcal{O} = [\mathbf{H} \quad \mathbf{H}\Phi \dots]$, the **observability matrix**, must be of rank n , the order of the system, for the system to be observable. obsf.m performs this calculation.
- Feedback via \mathbf{K} using the estimated state elements results in system poles that consist of the n control poles plus the n estimator poles.
- Estimator poles are usually selected to be approximately twice as fast as the controller poles in order for the response to be dominated by the control design. However, in order to smooth the effects of measurement noise, it is sometimes useful to select estimator poles as slow or slower than the control poles.
- Calculation of \mathbf{N}_c and \mathbf{N}_e via refi.m and their usage with the structure in Figs. 8.15 or 8.16, the **state command structure**, provides the best response to command inputs.
- **Integral control** is achieved by implementing the desired error integral and including this as part of an augmented plant model in calculating \mathbf{K} . The estimator is based on the non-augmented model. Integral control eliminates steady state errors due to command inputs and input disturbances.

- Disturbance estimation is accomplished by augmenting the model used in the estimator to include the unknown disturbance as a state element. The disturbance could be an unknown constant or a sinusoid with unknown magnitude and phase. The disturbance estimate can then be used in the control law to reject its effect, called **disturbance rejection** or to cause the system to track the disturbance with no error, called **disturbance following**.
- Delays in the actuator or sensor can be modeled in the estimator so that no estimation errors occur. **Sensor delays** will cause a delay in the response to a disturbance but there need not be any delay in the response to command inputs. However, **actuator delays** will cause a delayed response from command inputs and disturbances.

8.9 Problems

8.1 For the open-loop system

$$G(s) = \frac{y(s)}{u(s)} = \frac{1}{s^2 + 0.2s + 1}.$$

- Find the discrete state-space representation assuming there is a ZOH and the sample period $T = 0.5$ sec.
- Find the full state digital feedback that provides equivalent s -plane poles at $\omega_n = 7$ rad/sec with $\zeta = 0.5$.
- Examine the response of the closed-loop system to an initial value of y and verify that the response is consistent with the desired poles.

8.2 For the open-loop system

$$\Phi = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix},$$

compute \mathbf{K} by hand so that the poles of the closed-loop system with full state feedback are at $z = 0.9 \pm j0.1$.

8.3 For the open-loop system

$$G(s) = \frac{y(s)}{u(s)} = \frac{1}{s^2(s+2)}.$$

- Find the discrete state-space model, assuming there is a ZOH and the sample rate is $T = 100$ msec.
- Pick poles so that the settling time $t_s < 1$ sec and find the \mathbf{K} that will produce those poles with full state feedback.
- Verify that t_s is satisfied by plotting the response to an initial value of y .

8.4 For the system in Problem 8.2, find the estimator equations and the value of the gain \mathbf{L} , by hand so that $\zeta_{de} = 0.6 \pm j0.3$ for

- a predictor estimator,
- a current estimator, and
- a reduced-order estimator ($\zeta_{de} = 0.6$).

8.5 For the system in Problem 8.3,

- Find the predictor estimator equations and the value of the gain \mathbf{L}_p so that the estimator $t_c < 0.5$ sec.
- Verify that t_c is satisfied by plotting the response of $\tilde{x}_1 (= \tilde{y})$ to an initial value.

8.6 For the open-loop system

$$G(s) = \frac{y(s)}{u(s)} = \frac{1}{s^2}$$

preceded by a ZOH and a sample rate of 20 Hz.

- Find the feedback gain \mathbf{K} so that the control poles have an equivalent s -plane $\omega_n = 10$ rad/sec and $\zeta = 0.7$.
- Find the estimator gain \mathbf{L}_e so that the estimator poles have an equivalent s -plane $\omega_n = 20$ rad/sec and $\zeta = 0.7$.
- Determine the discrete transfer function of the compensation.
- Design a lead compensation using transform techniques so that the equivalent s -plane natural frequency $\omega_n \cong 10$ rad/sec and $\zeta \geq 0.7$. Use either root locus or frequency response.
- Compare the compensation transfer functions from (c) and (d) and discuss the differences.

8.7 For the system in Problem 8.6, design the controller and estimator so that the closed-loop unit step response to a command input has a rise time $t_r < 200$ msec and an overshoot $M_p < 15\%$ when using:

- the state command structure,
- the output error command structure.

For both cases, check that the specifications are met by plotting the step response.

8.8 For the open-loop system

$$G(s) = \frac{y(s)}{u(s)} = \frac{1}{s^2(s^2 + 400)}.$$

design the controller and estimator so that the closed-loop unit step response to a command input has a rise time $t_r < 200$ msec and an overshoot $M_p < 15\%$ when using the state command structure. Check that the specifications are met by plotting the step response.

8.9 Compute $G(z)$ from Eq. (4.64) for

$$\Phi = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{H} = [b_1 \ b_2 \ b_3].$$

Why is this form for Φ and Γ called control canonical form?

8.10 (a) For

$$\Phi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} T^2/2 \\ T \end{bmatrix},$$

find a transform matrix \mathbf{T} so that if $\mathbf{x} = \mathbf{T}\mathbf{w}$, then the equations in \mathbf{w} will be in control canonical form.

- (b) Compute \mathbf{K}_u , the gain, such that if $u = -\mathbf{K}_u w$, the characteristic equation will be $\alpha_r(z) = z^2 - 1.6z + 0.7$.
 (c) Use \mathbf{T} from part (a) to compute \mathbf{K}_x , the gain in the x -states.

8.11 (a) Show that the equations for the current estimator can be written in standard state form

$$\dot{\xi}_{k+1} = \mathbf{A}\xi_k + \mathbf{B}y_k, \quad u = \mathbf{C}\xi_k + Dy_k,$$

where $\xi_k = \hat{x}_k - \mathbf{L}_e y_k$, $\mathbf{A} = (\mathbf{I} - \mathbf{L}_e \mathbf{H})(\Phi - \Gamma \mathbf{K})$, $\mathbf{B} = \mathbf{A}\mathbf{L}_e$, $\mathbf{C} = -\mathbf{K}$, and $D = -\mathbf{K}\mathbf{L}_e$.

- (b) Use the results of Eq. (4.65) to show that the controller based on a current estimator always has a zero at $z = 0$ for any choice of control law \mathbf{K} or estimator law \mathbf{L}_e .

8.12 For the open-loop system

$$\Phi = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix},$$

check the observability for:

- (a) $\mathbf{H} = [0 \ 1]$.
 (b) $\mathbf{H} = [1 \ 0]$.
 (c) Rationalize your results to (a) and (b), stating why the observability or lack of it occurred.

8.13 Design the antenna in Appendix A.2 by state-variable pole assignment.

- (a) Write the equations in state form with $x_1 = y$ and $x_2 = \dot{y}$. Give the matrices \mathbf{F} , \mathbf{G} , and \mathbf{H} . Let $a = 0.1$.
 (b) Let $T = 1$ and design \mathbf{K} for equivalent poles at $s = -1/2 \pm j(\sqrt{3}/2)$. Plot the step response of the resulting design.
 (c) Design a prediction estimator with \mathbf{L}_p selected so that $\alpha_r(z) = z^2$; that is, both poles are at the origin.
 (d) Use the estimated states for computing the control and introduce the reference input so as to leave the state estimate undisturbed. Plot the unit step response from this reference input and from a wind gust (step) disturbance that acts on the antenna just like the control force (but not on the estimator).
 (e) Plot the root locus of the closed-loop system with respect to the plant gain and mark the locations of the closed-loop poles.

8.14 In Problem 7.8 we described an experiment in magnetic levitation described by the equations

$$\ddot{x} = 1000x + 20u.$$

Let the sampling time, T , be 10 msec.

- (a) Use pole placement to design this system to meet the specifications that settling time is less than 0.25 sec and overshoot to an initial offset in x is less than 20%.
 (b) Design a reduced-order estimator for \dot{x} for this system such that the error-settling time will be less than 0.08 sec.
 (c) Plot step responses of x , \dot{x} , and u for an initial x displacement.
 (d) Plot the root locus for changes in the plant gain and mark the design pole locations.

- (e) Introduce a command reference with feedforward so that the estimate of x is not forced by r . Measure or compute the frequency response from r to system error $r - x$ and give the highest frequency for which the error amplitude is less than 20% of the command amplitude.

8.15 Derive Eq. (8.63) from Eq. (8.57).

8.16 For the open-loop system

$$G(z) = \frac{y(z)}{u(z)} = \frac{y(z)}{w(z)} = \frac{0.1185(z + 0.9669)}{z^2 - 1.6718z - 0.9048}$$

- (a) Find the control feedback \mathbf{K} and estimator gain \mathbf{L}_p that will place control poles at $z = 0.8 \pm j0.2$ and estimator poles at $z = 0.6 \pm j0.3$.
 (b) Plot the response of $y(k)$ for a unit step input (w) command using the state command structure. Would there be a steady-state error if $\bar{N} = 0$?
 (c) Determine what the steady state value of $y(k)$ would be if there was an input disturbance, w .
 (d) Determine an integral control gain and show the system block diagram with the integral control term included. Set the extra control pole at $z = 0.9$.
 (e) Demonstrate that the system will have no steady state error for a disturbance, w , or an input command, u , even when $\bar{N} = 0$.

8.17 For the open-loop system

$$\Phi = \begin{bmatrix} 0.8815 & 0.4562 \\ -0.4562 & 0.7903 \end{bmatrix}, \quad \Gamma = \Gamma_1 = \begin{bmatrix} 0.1185 \\ 0.4562 \end{bmatrix}, \quad \mathbf{H} = [1 \ 0],$$

- (a) Find the control feedback \mathbf{K} and estimator gain \mathbf{L}_p that will place control poles at $z = 0.6 \pm j0.3$ and estimator poles at $z = 0.3 \pm j0.3$.
 (b) Plot the response of $y(k)$ for a unit step of the disturbance (w).
 (c) Determine an integral control gain and show the system block diagram with the integral control term included. Set the extra control pole at $z = 0.9$.
 (d) Demonstrate that the system will have no steady-state error for a disturbance, w .

8.18 For the open-loop system from Problem 8.17

- (a) Assuming w takes on some unknown but constant value, construct an estimator that includes an estimate of that disturbance. Place poles of the system as in Problem 8.17, except place the extra estimator pole at $z = 0.9$. Determine values of \mathbf{K} and \mathbf{L}_e and sketch the block diagram showing how the various quantities are used in the control. Include the command input r in the diagram using the state command structure.
 (b) Plot the response of y and \tilde{w} to a unit step in w with $r \equiv 0$. State whether the responses meet your expectations.
 (c) Plot the response of y and \tilde{w} to a unit step in r with $w \equiv 0$. State whether the responses meet your expectations.

8.19 A disk drive read head²¹ has the open-loop transfer function

$$G(s) = \frac{y(s)}{u(s)} = \frac{y(s)}{w(s)} = \frac{1000\omega_r^2}{s^2(s^2 + 2\xi_r\omega_r s + \omega_r^2)}$$

²¹ See Franklin, Powell, and Emami (1994), Example 2.4.

where $\omega_r = 6000 \text{ rad/sec}$ and $\zeta_r = 0.02$.

- (a) Design a digital compensation so that there is no steady state error to an input command nor to a constant disturbance w . The plant gain of 1000 is not known precisely, so it is not acceptable to assume the steady state error due to input commands can be eliminated via feedforward with an N_p term. The specifications are that the rise time t_r must be less than 2 msec and the overshoot $M_p < 10\%$. Use a sample rate of 3 kHz.
- (b) The disk spins at 3000 rpm. There is typically a small offset between the center of the circular tracks on the disk and the center of rotation, thus producing a wobble in the location of the tracks that should be followed by the read head. Determine the track following error due to this wobble. Express the answer as a percentage of the wobble magnitude.
- (c) Embellish your design from (a) so that the error due to the wobble is eliminated as best you can. Plot the frequency response of the tracking error (e in Fig. 8.32) where the input to the system is the track wobble. Mark the frequency that represents the spin rpm.

8.20 A pendulum with a torquer at its hinge is described by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u - \begin{bmatrix} 0 \\ 1 \end{bmatrix} w,$$

$$y = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + b$$

where $x^T = [\theta \ \dot{\theta}]$, $\theta = \text{angle of the pendulum from vertical}$, $u = \text{torque}$, $w = \text{torque bias}$, and $b = \text{measurement bias}$. Answer the questions below assuming the output is sampled with $T = 100 \text{ msec}$ and the control $(u + w)$ is applied through a ZOH.

- (a) With no torque bias ($w = 0$), augment the system model so that the measurement bias is a state element. Is this system observable?
- (b) With no measurement bias ($b = 0$), augment the system model so that the torque bias is a state element. Is this system observable?
- (c) Augment the system model so that both biases are state elements. Is this system observable?

8.21 Design a controller (control plus estimator) for the same system as in Example 8.12, except add a delay of two sample periods between the system output and when the measurement is available to the computer.

- (a) Compute K for $z_{de} = 0.8 \pm j0.25, 0, 0$.
- (b) Is the system observable? Check for predictor and current estimators.
- (c) Compute L_p with $z_{de} = 0.4 \pm j0.4, 0, 0$.
- (d) Compute L_e with z_{de} as in (c) and with $z_{de} = 0.4 \pm j0.4, 0.1, 0$.
- (e) Plot the unit step response to an input command using the predictor estimator showing the plant output as well as the delayed sensor output.

8.22 Determine the state-command input structure for a feedback system with the Type 0 plant

$$G(s) = \frac{10}{s^2 + 0.18s + 9}$$

- (a) Convert the system to discrete state-space form with $T = 0.1 \text{ sec}$.

- (b) Find K to obtain equivalent s-plane control poles at $\zeta = 0.5$ and $\omega_n = 4 \text{ rad/sec}$.
- (c) Find L_p to obtain equivalent s-plane estimator poles at $\zeta = 0.5$ and $\omega_n = 8 \text{ rad/sec}$.
- (d) Determine N_p and N_e , then sketch a block diagram specifying the controller equations including the reference input.
- (e) Do you expect there will be a steady-state error for this system for a step input?
- (f) Plot the step response and confirm your answer to (e).

- 8.23 Repeat the design of the controller (control and estimator) for Examples 8.4 and 8.8, but place all four desired poles at $z = 0$. (This is often referred to as a *finite settling time or deadbeat* design because the system will settle in a finite number of sample periods.)
- (a) Using your K and L_p , determine the time response of the system to a unit-step input using the state-command input structure.
- (b) Determine the compensation in transfer function form, $D_c(z)$, and construct a root locus vs. the dc gain of $D_c(z)$. Mark the roots where the gain corresponds to the values computed for K and L_p .

- 8.24 The double mass-spring device described in Appendix A.4 is representative of many devices that have some structural resonances. Placing the sensor so that it measures y is called the *collocated case*, whereas placing it so that it measures d is called the *noncollocated case*. Often, the designer is not aware initially that a resonance exists in the system, a situation that is addressed by this problem.

For $M = 20 \text{ kg}$, $m = 1 \text{ kg}$, $k = 25 \text{ N/m}$, and $b = 0.2 \text{ N-sec/m}$, we obtain a resonance frequency of 5 rad/sec with a damping ratio, $\zeta = 0.02$.

- (a) To represent the case where the designer did not know about the resonance, assume the coupling is rigid, that is, k is infinite. The transfer function is then

$$G_1(z) = \frac{d(z)}{u(z)} = \frac{v(z)}{u(z)} = \frac{1}{(m+M)z^2}.$$

Design a digital controller (K and L_p) with $T = 200 \text{ msec}$, control poles at $z = 0.75 \pm j0.2$, and estimator poles at $z = 0.3 \pm j0.3$. Verify by simulation that it provides a response to a unit-step command using the state-command input structure that is consistent with the selected poles.

- (b) Use the controller (K and L_p) obtained in part (a) in a simulation of the system where the infinite spring is replaced with the flexible one and the output is y , that is, a fourth-order plant with second-order controller. Examine the response and compare it qualitatively with an analysis of the closed-loop roots of this combined system.
- (c) Repeat part (b), but replace the plant output, d , with y .
- (d) Analyze where the roots of the system would be if you measured y and \dot{y} directly (no estimator) and fed them back using your K from part (a).
- (e) Design a fourth-order controller with control poles at $z = 0.75 \pm j0.2, 0.4 \pm j0.6$, and estimator poles at $z = 0.3 \pm j0.3, 0 \pm j0.4$ with d as the measurement. Again, verify by simulation that it provides the correct response to a unit-step command using the state-command input structure.
- (f) Plot the frequency response of the compensation (control plus estimator) from part (e). State why you think this kind of compensation is usually referred to as a *notch filter*.

- (g) Plot the z -plane root locus of the system (plant plus controller) and comment on the sensitivity of this design to changes in the overall loop gain.

8.25 A heat exchanger²² has the transfer function

$$G(s) = \frac{e^{-5s}}{(10s+1)(60s+1)},$$

where the delay is due to the sensor.

- (a) Write state equations for this system.
- (b) Compute the zero-order-hold model with a sample period of 5 sec.
- (c) Design the compensation including the command input with the control poles at $0.8 \pm j0.25$ and the estimator poles at $0.4 \pm j0.4$.
- (d) Compute the step response to a reference input step and to a disturbance input at the control. Verify that there is no delay for the command input.

²² See Franklin, Powell, and Emami (1994), Example 2.16.

• 9 •

Multivariable and Optimal Control

A Perspective on Multivariable and Optimal Control

The control-design procedures described in Chapters 7 and 8 were applied to systems with a single input and single output (SISO). The transfer-function approach in Chapter 7 is best suited to SISO systems; the state-space methods of Chapter 8 were limited to SISO in order to simplify the procedures. In fact, if we try to apply the pole-placement approach of Chapter 8 to a multivariable (multi-input, multi-output, or MIMO) system, we find that the gains, **K** or **L**, are not uniquely determined by the resulting equations. Therefore, a design approach is required which intelligently uses this extra freedom for MIMO systems. In addition, we saw in Example 8.3 that the selection of desired pole locations for SISO systems can be tricky business. Some sort of systematic guidance for the selection of control and estimator pole locations seems highly desirable. The material in this chapter provides a tool that meets both these needs.

The subject of this chapter is the use of optimal control techniques as a tool in the design of control systems. It is important that the designer have no illusions that some true “optimal” design is being achieved; rather, the idea is to transfer the designer’s iteration on pole locations as used in Chapter 8, or compensation parameters as used in Chapter 7, to iterations on elements in a cost function, \mathcal{J} . The method will determine the control law that minimizes \mathcal{J} , but because the parameters in \mathcal{J} are arbitrarily selected, the design is at best only partially optimal. However, these designs will achieve some compromise between the use of control effort and the speed of response and will guarantee a stable system, no small feat. Therefore, each iteration on the parameters in \mathcal{J} produces a candidate design that should be evaluated in the light of the design specifications.

Chapter Overview

The chapter starts out in Section 9.1 by discussing some of the steps that might be taken in order to convert a MIMO system into a SISO one. Although this cannot

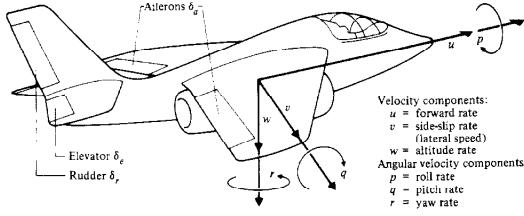
always be done, it can help to clarify the key control issues in a complex system so that the later optimization of the whole system will be better understood. Section 9.2 derives the time-varying optimal control solution that results directly from the optimal control problem statement. Section 9.3 shows how to find the steady-state value of the optimal feedback gain that is significantly easier to implement and is the one typically used in control implementations. Section 9.4 derives the companion optimal estimation problem. As for the control case, the time-varying gain solution is found first, then we find the steady-state gain case that is typically the one implemented. The final section, 9.5, shows how to use these results in the design of MIMO control systems.

9.1 Decoupling

The first step in any multivariable design should be an attempt either to find an approximate model consisting of two or more single input-output models or else to decouple the control gain matrix \mathbf{K} and the estimator gain matrix \mathbf{L} . This step will give better physical insight into the important feedback variables and can lead to a plant description that is substantially simpler for design purposes and yet yields no significant degradation from an analysis based on the full multivariable system.

For example, the linearized equations of motion of an aircraft (Fig. 9.1) are of eighth order but are almost always separated into two fourth-order sets representing longitudinal motion (w, u, q) and lateral motion (p, r, v). The elevator

Figure 9.1
Schematic of an aircraft showing variable definitions



control surfaces affect longitudinal motion; the aileron and rudder primarily affect lateral motion. Although there is a small amount of coupling of lateral motion into longitudinal motion, this is ignored with no serious consequences when the control, or "stability-augmentation," systems are designed independently for the two fourth-order systems.

Further decoupling of the equations is also possible.

◆ Example 9.1 Control Decoupling

Decouple the lateral aircraft fourth-order equations into two second-order equations and show how to design the control.

Solution. The aircraft lateral equations are multivariable and of the form

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k), \quad (9.1)$$

where

$$\mathbf{u} = \begin{bmatrix} \delta_a \\ \delta_e \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} v \\ r \\ \phi_p \\ p \end{bmatrix}, \quad \text{and} \quad p = \dot{\phi}_p.$$

A control law of the standard form

$$\begin{bmatrix} \delta_a \\ \delta_e \end{bmatrix} = - \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \end{bmatrix} \begin{bmatrix} v \\ r \\ \phi_p \\ p \end{bmatrix} \quad (9.2)$$

shows that there are eight elements in the gain matrix to be selected, and the specification of four closed-loop roots clearly causes the problem to be underdetermined and will leave many possible values of \mathbf{K} that will meet the specifications. This shows that the pole-placement approach to multivariable system design poses difficulties and additional criteria need to be introduced.

A decoupling that removes the ambiguity is to restrict the control law to

$$\begin{bmatrix} \delta_a \\ \delta_e \end{bmatrix} = - \begin{bmatrix} K_{11} & K_{12} & 0 & 0 \\ 0 & 0 & K_{22} & K_{23} \end{bmatrix} \begin{bmatrix} v \\ r \\ \phi_p \\ p \end{bmatrix}. \quad (9.3)$$

This makes good physical sense because the rudder primarily yaws the aircraft about a vertical axis (r motion), thus directly causing sideslip (v), and the ailerons primarily roll the aircraft about an axis through the nose, thus causing changes in the roll angle, ϕ_p , and the roll rate, p . Given an achievable set of desired pole locations, there are unique values of the four nonzero components of \mathbf{K} ; however, the governing equations cannot be cast in the same form as in Eq. (8.5) and therefore can be difficult to solve.

A further decoupling that would permit an easy gain calculation is to assume that Eq. (9.1) is of the form

$$\begin{bmatrix} v \\ r \\ \phi_p \\ p \end{bmatrix}_{k+1} = \begin{bmatrix} \phi_{11} & \phi_{12} & 0 & 0 \\ \phi_{21} & \phi_{22} & 0 & 0 \\ 0 & 0 & \phi_{33} & \phi_{34} \\ 0 & 0 & \phi_{43} & \phi_{44} \end{bmatrix} \begin{bmatrix} v \\ r \\ \phi_p \\ p \end{bmatrix}_k + \begin{bmatrix} \Gamma_{11} & 0 \\ \Gamma_{21} & 0 \\ 0 & \Gamma_{32} \\ 0 & \Gamma_{42} \end{bmatrix} \begin{bmatrix} \delta_r \\ \delta_s \end{bmatrix}_k \quad (9.4)$$

and that the control law is given by Eq. (9.3).

This makes some physical sense but ignores important coupling between the two aircraft modes. It does, however, decouple the system into second-order systems for which the methods of Chapter 8 can be applied directly to obtain the gains. The resulting closed-loop characteristic roots of the full lateral equations can be checked by calculating the eigenvalues of the closed loop matrix: (see eig.m in MATLAB)

$$\Phi_{\text{closed loop}} = \begin{bmatrix} \phi_{11} - \Gamma_{11}\mathbf{K}_{11} & \phi_{12} - \Gamma_{11}\mathbf{K}_{12} & \phi_{13} - \Gamma_{12}\mathbf{K}_{22} & \phi_{14} - \Gamma_{12}\mathbf{K}_{24} \\ \phi_{21} - \Gamma_{21}\mathbf{K}_{11} & \phi_{22} - \Gamma_{21}\mathbf{K}_{12} & \phi_{23} - \Gamma_{22}\mathbf{K}_{22} & \phi_{24} - \Gamma_{22}\mathbf{K}_{24} \\ \phi_{31} - \Gamma_{31}\mathbf{K}_{11} & \phi_{32} - \Gamma_{31}\mathbf{K}_{12} & \phi_{33} - \Gamma_{32}\mathbf{K}_{22} & \phi_{34} - \Gamma_{32}\mathbf{K}_{24} \\ \phi_{41} - \Gamma_{41}\mathbf{K}_{11} & \phi_{42} - \Gamma_{41}\mathbf{K}_{12} & \phi_{43} - \Gamma_{42}\mathbf{K}_{22} & \phi_{44} - \Gamma_{42}\mathbf{K}_{24} \end{bmatrix}. \quad (9.5)$$

which results from combining Eq. (9.3) and (9.4).

If the plane coupling that was ignored in the gain computation is important, the roots obtained from Eq. (9.5) will differ from those used to compute the gains using Eqs. (9.3) and (9.4). In many cases, the method will be accurate enough and one need look no further. In other cases, one could revise the "desired" root locations and iterate until the correct roots from Eq. (9.5) are satisfactory or else turn to the methods of optimal control to be described in the following sections.

The same ideas apply equally well to the decoupling of the estimator into SISO parts.

◆ Example 9.2 Estimator Decoupling

Decouple the estimator for the inverted pendulum on a motorized cart (Fig. 9.2).

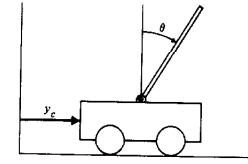
Solution. The equations of motion can be written as

$$\begin{bmatrix} \dot{x}_c \\ \ddot{x}_c \end{bmatrix}_{k+1} = \begin{bmatrix} \phi_{cc} & \phi_{ct} \\ \phi_{tc} & \phi_{tt} \end{bmatrix} \begin{bmatrix} \dot{x}_c \\ \ddot{x}_c \end{bmatrix}_k + \begin{bmatrix} \Gamma_c \\ \Gamma_t \end{bmatrix} u(k), \quad (9.6)$$

where

$$\begin{aligned} \dot{x}_c &= \begin{bmatrix} \dot{x}_c \\ \dot{\theta}_c \end{bmatrix} \text{ cart position and velocity.} \\ \ddot{x}_c &= \begin{bmatrix} \ddot{x}_c \\ \ddot{\theta}_c \end{bmatrix} \text{ stick angle and angular rate.} \end{aligned}$$

Figure 9.2
Hinged stick and
motorized cart



and the available measurements are

$$\begin{bmatrix} y \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ \dot{x}_c \\ \theta_c \\ \dot{\theta}_c \end{bmatrix}. \quad (9.7)$$

The stick pictured in Fig. 9.2 is substantially lighter than the cart. This means that stick motion has a small dynamic effect on cart motion, which in turn implies that $\dot{\theta}_c \approx 0$. This does not imply that $\dot{\theta}_c = 0$; in fact, cart motion is the mechanism for influencing stick motion and hence stabilizing it.

An estimator for the system described by Eq. (9.6) and Eq. (9.7) requires the determination of eight elements of an estimator gain matrix, \mathbf{L} . Hence, specifying the four estimator roots and using the methods of Chapter 8 would not determine this \mathbf{L} uniquely—another example of an underdetermined system caused by the multivariable nature of the problem.

But because we can assume that $\dot{\theta}_c = 0$, the cart equation in Eq. (9.6) uncouples from the stick equation, and we simply design an estimator for

$$\begin{aligned} \dot{x}_c(k+1) &= \phi_{cc}x_c(k) + \Gamma_c u, \\ y_c &= [1 \ 0] \begin{bmatrix} x_c \\ \dot{x}_c \end{bmatrix}, \end{aligned} \quad (9.8)$$

which can be done with the methods described in Chapter 8. There is one way coupling into the stick equation, but this just acts like an additional control input and can be ignored in the calculation of the stick estimator gain matrix, \mathbf{L}_s , using the pole-placement methods of Chapter 8. However, the ϕ_{ct} coupling should not be ignored in the estimator equations, and there is no reason to ignore the weak coupling, ϕ_{ct} . The final (predictor) estimator would be of the form

$$\begin{aligned} \dot{x}_c(k+1) &= \phi_{cc}\dot{x}_c(k) + \phi_{ct}\ddot{x}_c(k) + \Gamma_c u(k) + \mathbf{L}_s(y_c(k) - \hat{y}_c(k)), \\ \ddot{x}_c(k+1) &= \phi_{tc}\dot{x}_c(k) + \phi_{tt}\ddot{x}_c(k) + \Gamma_t u(k) + \mathbf{L}_t(\theta_c(k) - \hat{\theta}_c(k)), \end{aligned}$$

where \mathbf{L}_s and \mathbf{L}_t are both 2×1 matrices.

Even without the very weak one-way coupling in ϕ_{ct} , that was obvious for this example, one could assume this to be the case, then check the resulting full-system characteristic roots using a method similar to the previous airplane example. Note that ignoring the coupling only causes approximations in the gain-matrix calculation and thus the root locations. There is no

approximation in the system model used in the estimator; therefore, the estimation errors will still approach zero for stable estimator roots.

In short, it is often useful to apply your knowledge of the physical aspects of the system at hand to break the design into simpler and more tractable subsets. With luck, the whole job can be finished this way. At worst, insight will be gained that will aid in the design procedures to follow and in the implementation and checkout of the control system.

9.2 Time-Varying Optimal Control

Optimal control methods are attractive because they handle MIMO systems easily and aid in the selection of the desired pole locations for SISO systems. They also allow the designer to determine many good candidate values of the feedback gain, \mathbf{K} , using very efficient computation tools. We will develop the time-varying optimal control solution first and then reduce it to a steady-state solution in the following section. The result amounts to another method of computing \mathbf{K} in the control law Eq. (8.5)

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \quad (9.9)$$

that was used in Chapter 8 and illustrated by Eq. (9.2) in Example 9.1.

Given a discrete plant

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k), \quad (9.10)$$

cost function

we wish to pick $\mathbf{u}(k)$ so that a cost function

$$\mathcal{J} = \frac{1}{2} \sum_{k=0}^N [\mathbf{x}^T(k)\mathbf{Q}_1\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{Q}_2\mathbf{u}(k)] \quad (9.11)$$

is minimized. \mathbf{Q}_1 and \mathbf{Q}_2 are symmetric weighting matrices to be selected by the designer, who bases the choice on the relative importance of the various states and controls. Some weight will almost always be selected for the control ($[\mathbf{Q}_2] \neq 0$), otherwise the solution will include large components in the control gains, and the states would be driven to zero at a very fast rate, which could saturate the actuator device.¹ The \mathbf{Q} 's must also be **nonnegative definite**,² which is most

1 If the sampling rate, T_s , is long, however, a control that moves the state along as rapidly as possible might be feasible. Such controls are called "dead-beat" because they beat the state to a dead stop in most n steps. They correspond to placement of all poles at $z = 0$. See Problem 8.23.

2 Matrix equivalent of a nonnegative number; it ensures that $\mathbf{x}^T\mathbf{Q}_1\mathbf{x}$ and $\mathbf{u}^T\mathbf{Q}_2\mathbf{u}$ are nonnegative for all possible \mathbf{x} and \mathbf{u} .

Lagrange multipliers

easily accomplished by picking the \mathbf{Q} 's to be diagonal with all diagonal elements positive or zero.

Another way of stating the problem given by Eqs. (9.10) and (9.11) is that we wish to minimize

$$\mathcal{J} = \frac{1}{2} \sum_{k=0}^N [\mathbf{x}^T(k)\mathbf{Q}_1\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{Q}_2\mathbf{u}(k)] \quad (9.11)$$

subject to the constraint that

$$-\mathbf{x}(k+1) + \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k) = 0, \quad k = 0, 1, \dots, N. \quad (9.10)$$

adjoint equations

This is a standard constrained-minima problem which can be solved using the method of **Lagrange multipliers**. There will be one Lagrange multiplier vector, which we will call $\boldsymbol{\lambda}(k+1)$, for each value of k . The procedure is to rewrite Eqs. (9.10) and (9.11) as

$$\mathcal{J}' = \sum_{k=0}^N \left[\frac{1}{2} \mathbf{x}^T(k)\mathbf{Q}_1\mathbf{x}(k) + \frac{1}{2} \mathbf{u}^T(k)\mathbf{Q}_2\mathbf{u}(k) + \boldsymbol{\lambda}^T(k+1)(-\mathbf{x}(k+1) + \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k)) \right], \quad (9.12)$$

and find the minimum of \mathcal{J}' with respect to $\mathbf{x}(k)$, $\mathbf{u}(k)$, and $\boldsymbol{\lambda}(k)$. Note that for an optimal $\mathbf{u}(k)$ that obeys Eq. (9.10), the two cost functions, \mathcal{J}' and \mathcal{J} , are identical in magnitude. The index on $\boldsymbol{\lambda}$ is arbitrary conceptually, but we let it be $k+1$ because this choice will yield a particularly easy form of the equations later on.

Proceeding with the minimization leads to

$$\frac{\partial \mathcal{J}'}{\partial \mathbf{u}(k)} = \mathbf{u}^T(k)\mathbf{Q}_2 + \boldsymbol{\lambda}^T(k+1)\Gamma = 0, \quad \text{control equations}, \quad (9.13)$$

$$\frac{\partial \mathcal{J}'}{\partial \boldsymbol{\lambda}(k+1)} = -\mathbf{x}(k+1) + \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k) = 0, \quad \text{state equations, and} \quad (9.10)$$

$$\frac{\partial \mathcal{J}'}{\partial \mathbf{x}(k)} = \mathbf{x}^T(k)\mathbf{Q}_1 - \boldsymbol{\lambda}^T(k)\mathbf{Q}_2 + \boldsymbol{\lambda}^T(k+1)\Phi = 0, \quad \text{adjoint equations}. \quad (9.14)$$

The last set of the equations, the **adjoint equations**, can be written as the backward difference equation

$$\boldsymbol{\lambda}(k) = \Phi^T\boldsymbol{\lambda}(k+1) + \mathbf{Q}_1\mathbf{x}(k). \quad (9.15)$$

Restating the results in more convenient forms, we have from Eq. (9.10)

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k), \quad (9.16)$$

where, using Eq. (9.13)

$$\mathbf{u}(k) = -\mathbf{Q}_2^{-1}\Gamma^T\boldsymbol{\lambda}(k+1), \quad (9.17)$$

and from Eq. (9.15) we can describe $\lambda(k+1)$ in the forward difference equation form

$$\lambda(k+1) = \Phi^{-T}\lambda(k) - \Phi^{-T}Q_i x(k). \quad (9.18)$$

Equations (9.16), (9.17), and either (9.15) or (9.18) are a set of coupled difference equations defining the optimal solution of $x(k)$, $\lambda(k)$, and $u(k)$, provided the initial (or final) conditions are known. The initial conditions on $x(k)$ must be given; however, usually $\lambda(0)$ would not be known, and we are led to the endpoint to establish a boundary condition for λ . From Eq. (9.11) we see that $u(N)$ should be zero in order to minimize J because $u(N)$ has no effect on $x(N)$ [see Eq. (9.10)]. Thus Eq. (9.13) suggests that $\lambda(N+1) = \mathbf{0}$, and Eq. (9.14) thus shows that a suitable boundary condition is

$$\lambda(N) = Q_i x(N). \quad (9.19)$$

A set of equations describing the solution to the optimal control problem is now completely specified. It consists of the two difference equations (9.16) and (9.15) with u given by Eq. (9.17), the final condition on λ given by Eq. (9.19), and the initial condition on x to be given in the problem statement. The solution to this two-point boundary-value problem is not easy.

One method, called the **sweep method** by Bryson and Ho (1975), is to assume

$$\lambda(k) = S(k)x(k). \quad (9.20)$$

This definition allows the transformation of the two-point boundary-value problem in x and λ to one in S with a single-point boundary condition. With the definition Eq. (9.20), the control Eq. (9.13) becomes

$$\begin{aligned} Q_2 u(k) &= -\Gamma^T S(k+1)x(k+1) \\ &= -\Gamma^T S(k+1)(\Phi x(k) + \Gamma u(k)). \end{aligned}$$

Solving for $u(k)$, we obtain

$$\begin{aligned} u(k) &= -(Q_2 + \Gamma^T S(k+1)\Gamma)^{-1}\Gamma^T S(k+1)\Phi x(k) \\ &= -R^{-1}\Gamma^T S(k+1)\Phi x(k). \end{aligned} \quad (9.21)$$

In Eq. (9.21) we have defined

$$R = Q_2 + \Gamma^T S(k+1)\Gamma$$

for convenience. If we now substitute Eq. (9.20) into Eq. (9.15) for $\lambda(k)$ and $\lambda(k+1)$, we eliminate λ . Then we substitute Eq. (9.21) into Eq. (9.16) to eliminate $x(k+1)$ as follows. From Eq. (9.15), we have

$$\lambda(k) = \Phi^T \lambda(k+1) + Q_i x(k).$$

and substituting Eq. (9.20), we have

$$S(k)x(k) = \Phi^T S(k+1)x(k+1) + Q_i x(k).$$

Riccati equation

Now we use Eq. (9.16) for $x(k+1)$

$$S(k)x(k) = \Phi^T S(k+1)(\Phi x(k) + \Gamma u(k)) + Q_i x(k).$$

Next we use Eq. (9.21) for $u(k)$ in the above

$$[S(k) - \Phi^T S(k+1)\Phi + \Phi^T S(k+1)\Gamma R^{-1}\Gamma^T S(k+1)\Phi - Q_i]x(k) = 0. \quad (9.22)$$

Because Eq. (9.22) must hold for any $x(k)$, the coefficient matrix must be identically zero, from which follows a backward difference equation describing the solution of $S(k)$

$$S(k) = \Phi^T [S(k+1) - S(k+1)\Gamma R^{-1}\Gamma^T S(k+1)]\Phi + Q_i, \quad (9.23)$$

which is often rewritten as

$$S(k) = \Phi^T M(k+1)\Phi + Q_i, \quad (9.24)$$

where

$$M(k+1) = S(k+1) - S(k+1)\Gamma [Q_2 + \Gamma^T S(k+1)\Gamma]^{-1}\Gamma^T S(k+1). \quad (9.25)$$

Equation (9.23) is called the discrete **Riccati equation**. It is not easy to solve because it is nonlinear in S . But note that the matrix to be inverted in Eq. (9.23) R , has the same dimension as the number of controls, which is usually less than the number of states.

The boundary condition on the recursion relationship for $S(k+1)$ is obtained from Eq. (9.19) and Eq. (9.20); thus

$$S(N) = Q_i, \quad (9.26)$$

and we see now that the problem has been transformed so that the solution is described by the recursion relations Eq. (9.24) and Eq. (9.25) with the single boundary condition given by Eq. (9.26). The recursion equations must be solved backwards because the boundary condition is given at the endpoint. To solve for $u(k)$, we use Eq. (9.21) to obtain

$$u(k) = -K(k)x(k), \quad (9.27)$$

where

$$K(k) = [Q_2 + \Gamma^T S(k+1)\Gamma]^{-1}\Gamma^T S(k+1)\Phi \quad (9.28)$$

and is the desired "optimal" time-varying feedback gain.

Let us now summarize the entire procedure:

1. Let $S(N) = Q_i$ and $K(N) = \mathbf{0}$.
2. Let $k = N$.

3. Let $\mathbf{M}(k) = \mathbf{S}(k) - \mathbf{S}(k)\Gamma[\mathbf{Q}_2 + \Gamma^T\mathbf{S}(k)\Gamma]^{-1}\Gamma^T\mathbf{S}(k)$.
4. Let $\mathbf{K}(k-1) = [\mathbf{Q}_2 + \Gamma^T\mathbf{S}(k)\Gamma]^{-1}\Gamma^T\mathbf{S}(k)\Phi$.
5. Store $\mathbf{K}(k-1)$.
6. Let $\mathbf{S}(k-1) = \Phi^T\mathbf{M}(k)\Phi + \mathbf{Q}_1$.
7. Let $k = k - 1$.
8. Go to step 3.

For any given initial condition for \mathbf{x} , to apply the control, we use the stored gains $\mathbf{K}(k)$ and

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k). \quad [9.16]$$

where

$$\mathbf{u}(k) = -\mathbf{K}(k)\mathbf{x}(k). \quad [9.27]$$

Note that the optimal gain, $\mathbf{K}(k)$, changes at each time step but can be pre-computed and stored for later use as long as the length, N , of the problem is known. This is so because no knowledge of the initial state $\mathbf{x}(0)$ is required for computation of the control gain $\mathbf{K}(k)$.

◆ Example 9.3 Time-Varying Nature of Control Gains

Solve for the time history of \mathbf{K} for the satellite attitude-control example described in Appendix A.1. Choose the state weighting matrix to be

$$\mathbf{Q}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \quad [9.29]$$

which means that the angle state is weighted but not the angular velocity. Choose the control weighting matrix, a scalar in this case because there is a single control input, to have three values

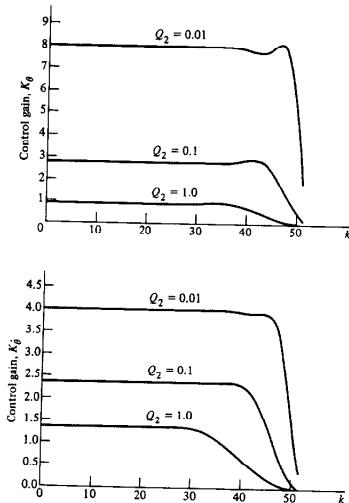
$$\mathbf{Q}_2 = 0.01, \quad 0.1, \quad \text{and} \quad 1.0. \quad [9.30]$$

and plot the resulting time histories of \mathbf{K} .

Solution. Equations (9.24) through (9.28) need to be solved for the system transfer function $G(s) = 1/s^2$. The problem length for purposes of defining J was chosen to be 51 steps, which, with the sample period of $T = 0.1$ sec, means that the total time was 5.1 sec. This time was chosen long enough so that it was apparent that the gains were essentially constant over the initial time period.

Figure 9.3 contains the resulting gain time histories plotted by the computer. We see from the figure that the problem length affects only the values of K near the end, and in fact, the first portions of all cases show constant values of the gains. If the problem length had been

Figure 9.3
Example of control gains versus time, Example 9.3



chosen to be longer, the end characteristics would have been identical, and the early constant gain portion would have existed for a longer time. ◆

The fact that the gain over the first portion of the example was constant is typical of the optimal gains for all constant coefficient systems, provided that the problem time is long enough. This means that the optimal controller over the early, constant-gain portion is identical to the constant-gain cases discussed in Chapter 8 and Section 9.1 except that the values of the constant gain, \mathbf{K} , are based on the minimization of a cost function rather than a computation based on specified root locations. We could also view this result as a method to find

a "good" set of pole locations because the constant, optimal \mathbf{K} during the early portion of the solution determines a set of closed-loop roots.

For MIMO problems, the time-varying gains act exactly as in the preceding example. The next section develops a method to compute the constant value of the optimal gains so that they can be used in place of the time-varying values, thus yielding a much simpler implementation. The only region where the gains are not optimal is during the transient region near the end. In fact, many control systems are turned on and left to run for very long times; for example, a satellite-attitude control system might run for years. This kind of problem is treated mathematically as if it runs for an infinite time, and therefore the constant-gain portion of the time-varying optimal solution is the true optimal.

Before we leave the time-varying case it is informative to evaluate the optimal cost function \mathcal{J} in terms of λ and \mathbf{S} . If we substitute Eqs. (9.13) and (9.14) for $\mathbf{x}^T(k+1)\Gamma$ and $\lambda^T(k+1)\Phi$ in Eq. (9.12), we find

$$\begin{aligned}\mathcal{J}' &= \frac{1}{2} \sum_{k=0}^N [\mathbf{x}^T(k)\mathbf{Q}_1\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{Q}_2\mathbf{u}(k) - \lambda^T(k+1)\mathbf{x}(k+1) \\ &\quad + (\lambda^T(k) - \mathbf{x}^T(k)\mathbf{Q}_1)\mathbf{x}(k) + (-\mathbf{u}^T(k)\mathbf{Q}_2)\mathbf{u}(k)] \\ &= \frac{1}{2} \sum_{k=0}^N [\lambda^T(k)\mathbf{x}(k) - \lambda^T(k+1)\mathbf{x}(k+1)] \\ &= \frac{1}{2} \lambda^T(0)\mathbf{x}(0) - \frac{1}{2} \lambda^T(N+1)\mathbf{x}(N+1).\end{aligned}$$

However, from Eq. (9.19), $\lambda(N+1) = \mathbf{0}$, and thus, using Eq. (9.20), we find

$$\begin{aligned}\mathcal{J}' &= \mathcal{J} = \frac{1}{2} \lambda^T(0)\mathbf{x}(0) \\ &= \frac{1}{2} \mathbf{x}^T(0)\mathbf{S}(0)\mathbf{x}(0).\end{aligned}\quad (9.31)$$

Thus we see that having computed \mathbf{S} , we can immediately evaluate the cost associated with the control. Although the cost could be used in evaluating different candidate designs, in fact it is not very useful because the weighting matrices, \mathbf{Q}_1 and \mathbf{Q}_2 , are arbitrary quantities that change with the different designs. Furthermore, the value of the discrete cost as defined by Eq. (9.11) is roughly proportional to the sample rate, thus eliminating any usefulness in evaluating the performance for different sample rates. Typically a designer would evaluate different designs by looking at the traditional measures that have been discussed in Chapter 7 or possibly by evaluating quadratic performance measures that are independent of the \mathbf{Q} 's and the sample rate. We will show in Section 9.3.4 how to remove the effect of sample rate on the value of the cost.

regulator

LQR

algebraic Riccati equation

Euler-Lagrange equations

9.3 LQR Steady-State Optimal Control

The previous section developed the optimal control gain that minimized the cost in Eq. (9.11). We saw that the result was a time-varying gain, $\mathbf{K}(k)$, but that there would usually be a portion of the solution that produced a constant gain, \mathbf{K}_{∞} , which would be much easier to implement in a control system. In fact, for the infinite time problem, called the **regulator** case, the constant-gain solution is the optimum. We call this solution the **linear quadratic regulator**, or LQR, because it applies to *linear* systems, the cost is *quadratic*, and it applies to the **regulator** case. This section will discuss how one finds the LQR solution and various properties of the solution.

One obvious method to compute the value of \mathbf{K} during the early, constant portion of a problem is to compute \mathbf{S} backward in time until it reaches a steady value, \mathbf{S}_{∞} , then use Eq. (9.28) to compute \mathbf{K}_{∞} . This has been done in some software packages and gives reliable answers. Its drawback is that it requires substantially more computation than the alternate methods.

Another method is to look for steady-state solutions of the Riccati equation. In steady state, $\mathbf{S}(k)$ becomes equal to $\mathbf{S}(k+1)$ (we'll call them both \mathbf{S}_{∞}) and the Riccati Eq. (9.23) reduces to

$$\mathbf{S}_{\infty} = \Phi'[\mathbf{S}_{\infty} - \mathbf{S}_{\infty}\Gamma\mathbf{R}^{-1}\Gamma^T\mathbf{S}_{\infty}]\Phi + \mathbf{Q}_1. \quad (9.32)$$

which is usually referred to as the **algebraic Riccati equation**. Because of the quadratic appearance of \mathbf{S}_{∞} , there is more than one solution, and one needs to know that \mathbf{S} must be positive definite to select the correct one. The fact that \mathbf{S} is positive definite follows by inspection of Eq. (9.31) and that \mathcal{J} must be positive. For extremely simple problems, one is sometimes able to use Eq. (9.32) to find an analytical solution for \mathbf{S}_{∞} , but in most cases this is impossible, and a numerical solution is required.

Most software packages use variations on a method called **eigenvector decomposition** due to its superior computational efficiency compared to the methods above. It is based on the linear description of the combined state and adjoint equations given by Eqs. (9.16), (9.17), and (9.18), which describe the time-varying solution. We can combine these equations into a set of difference equations in standard form in \mathbf{x} and λ if we assume that \mathbf{Q}_2 and Φ are nonsingular.³ These equations are called **Hamilton's equations** or the **Euler-Lagrange equations**

$$\begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix}_{k+1} = \begin{bmatrix} \Phi + \Gamma\mathbf{Q}_1^{-1}\Gamma^T\Phi^T\mathbf{Q}_1 & -\Gamma\mathbf{Q}_1^{-1}\Gamma^T\Phi^T \\ -\Phi^T\mathbf{Q}_1 & \Phi^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix}_k. \quad (9.33)$$

³ For systems with a pure time delay that is greater than the sampling period, \mathbf{T} , Φ is singular and the following development would fail. Software packages usually have features in their formulations that circumvent this difficulty.

and their system matrix is called the control Hamiltonian matrix

$$\mathcal{H}_c = \begin{bmatrix} \Phi + \Gamma Q_2^{-1} \Gamma^T \Phi^T Q_1 & -\Gamma Q_2^{-1} \Gamma^T \Phi^T \\ -\Phi^T Q_1 & \Phi^T \end{bmatrix}. \quad (9.34)$$

Because the system described by Eq. (9.33) is linear and the Hamiltonian matrix is constant, we can solve for the eigenvalues of Eq. (9.34) [or roots of Eq. (9.33)] using standard techniques (see eig.m in MATLAB). For an n th order system, there will be $2n$ eigenvalues. We will show in the next section that n of the roots are stable and the other n are unstable. In fact, the n unstable roots are the reciprocals of the n stable roots. Furthermore, the n stable roots are the roots of the optimal, constant-gain closed-loop system! If we were trying to find the optimal \mathbf{K} for a SISO system, the problem would now be complete, because knowing the optimal roots allows us to use Ackermann's formula to find \mathbf{K} . But we want the optimal \mathbf{K} for MIMO systems, too, so it's not that simple. We will return to the eigenvector decomposition solution after establishing the characteristics of the roots just stated.

9.3.1 Reciprocal Root Properties

Let us turn to the question of the reciprocal nature of the roots of Eq. (9.33). If we take the z -transforms of Eqs. (9.16), (9.17), and (9.15), we obtain

$$z\mathbf{X}(z) = \Phi\mathbf{X}(z) + \Gamma\mathbf{U}(z), \quad (9.35)$$

$$\mathbf{U}(z) = -zQ_2^{-1}\Gamma^T\Lambda(z), \quad (9.36)$$

$$\Lambda(z) = Q_1\mathbf{X}(z) + z\Phi^T\Lambda(z). \quad (9.37)$$

If we substitute Eq. (9.36) into Eq. (9.35) and write the remaining two equations in terms of the variables $\mathbf{X}(z)$ and $\Lambda(z)$, we find, in matrix form

$$\begin{bmatrix} z\mathbf{I} - \Phi & \Gamma Q_2^{-1}\Gamma^T \\ -Q_1 & z^{-1}\mathbf{I} - \Phi^T \end{bmatrix} \begin{bmatrix} \mathbf{X}(z) \\ z\Lambda(z) \end{bmatrix} = [\mathbf{0}].$$

Thus the roots of the Hamiltonian system are those values of z for which

$$\det \begin{bmatrix} z\mathbf{I} - \Phi & \Gamma Q_2^{-1}\Gamma^T \\ -Q_1 & z^{-1}\mathbf{I} - \Phi^T \end{bmatrix} = 0.$$

If we now reduce the term $-Q_1$ to zero by adding $Q_1(z\mathbf{I} - \Phi)^{-1}$ times the first row to the second row, we have

$$\det \begin{bmatrix} z\mathbf{I} - \Phi & \Gamma Q_2^{-1}\Gamma^T \\ 0 & z^{-1}\mathbf{I} - \Phi^T + Q_1(z\mathbf{I} - \Phi)^{-1}\Gamma Q_2^{-1}\Gamma^T \end{bmatrix} = 0.$$

Because this matrix is blockwise triangular, we have

$$\det(z\mathbf{I} - \Phi)\det(z^{-1}\mathbf{I} - \Phi^T + Q_1(z\mathbf{I} - \Phi)^{-1}\Gamma Q_2^{-1}\Gamma^T) = 0.$$

Now we factor the term $z^{-1}\mathbf{I} - \Phi^T$ from the second term to find

$$\det(z\mathbf{I} - \Phi)\det\{(z^{-1}\mathbf{I} - \Phi^T)(\mathbf{I} + (z^{-1}\mathbf{I} - \Phi^T)^{-1}Q_1(z\mathbf{I} - \Phi)^{-1}\Gamma Q_2^{-1}\Gamma^T)\} = 0.$$

To simplify the notation, we note that $\det(z\mathbf{I} - \Phi) = a(z)$, the plant characteristic polynomial, and $\det(z^{-1}\mathbf{I} - \Phi^T) = a(z^{-1})$. Thus, using the fact that $\det \mathbf{AB} = \det \mathbf{A} \det \mathbf{B}$, we find that the Hamiltonian characteristic equation is

$$a(z)a(z^{-1})\det(\mathbf{I} + \rho(z^{-1}\mathbf{I} - \Phi^T)^{-1}\mathbf{H}^T\mathbf{H}(z\mathbf{I} - \Phi)^{-1}\bar{\Gamma}\bar{\Gamma}^T) = 0. \quad (9.38)$$

where $Q_1 = \rho\mathbf{H}^T\mathbf{H}$ and $\Gamma Q_2^{-1}\Gamma^T = \bar{\Gamma}\bar{\Gamma}^T$. Now we use the result (Eq. C.3) from Appendix C for the determinant of a sum of \mathbf{I} and a matrix product \mathbf{AB} , choosing $\mathbf{A} = (z^{-1}\mathbf{I} - \Phi^T)^{-1}\mathbf{H}^T$ to write

$$a(z)a(z^{-1})\det[\mathbf{I} + \rho\mathbf{H}(z\mathbf{I} - \Phi)^{-1}\bar{\Gamma}\bar{\Gamma}^T(z^{-1}\mathbf{I} - \Phi^T)^{-1}\mathbf{H}^T] = 0. \quad (9.39)$$

If we replace z by z^{-1} in Eq. (9.39), the result is unchanged because $\det A^T = \det A$. Therefore, if z_i is a characteristic root of the optimal system, so is the reciprocal z_i^{-1} , and the desired relationship has been established.

These $2n$ roots are those of the coupled \mathbf{x}, \mathbf{A} system described by Eq. (9.33), which describes the solution of the time-varying gain case. But the time-varying gain solution includes the portion where the gains are constant. Furthermore, during the constant-gain portion, the system can be described by Eq. (9.16) with

$$\mathbf{u}(k) = -\mathbf{K}_{\infty}\mathbf{x}(k),$$

and the roots of this simplified n th order description must be n of the $2n$ roots of Eq. (9.33). But which ones are they? The answer must be the n stable ones, because if any unstable roots were included, the value of \mathcal{J} would be approaching infinity and would be far from optimal.

Therefore we see that once the roots of Eq. (9.33) are found, the n stable ones are the roots of the optimal *constant-gain* case.

9.3.2 Symmetric Root Locus

An interesting special case of Eq. (9.39) occurs for SISO systems. In this case, the cost function \mathcal{J} can be written as $\mathcal{J} = \rho y^2 + u^2$ where $y = \mathbf{Hx}$ and Q_2 was set equal to 1. Therefore, $\bar{\Gamma} = \Gamma$ and we see that $\mathbf{H}(z\mathbf{I} - \Phi)^{-1}\Gamma$ is the plant transfer function $G(z)$. Eq. (9.39) reduces to

$$1 + \rho G(z^{-1})G(z) = 0, \quad (9.40)$$

and is the characteristic equation for SISO optimal control. It is an equation in root-locus form with respect to ρ , the parameter that reflects the relative weighting on output error y and control u . If ρ is small, the optimal roots are near the poles of the plant (or the stable reflections of the poles if $G(z)$ is unstable), and as ρ gets large, the roots go toward the zeros of $G(z^{-1})G(z)$, which are inside the unit circle.

◆ **Example 9.4 Symmetric Root Locus for Satellite Attitude Control**

Draw the symmetric root locus for the satellite attitude-control problem in Example 4.11 for $T=1.4$ sec. Comment on the damping of the optimal controller vs. ρ .

Solution. The discrete transfer function from Example 4.11 is

$$G(z) = \frac{(z+1)}{(z-1)^2}$$

Replacing all the z 's with z^{-1} and multiplying top and bottom by z^2 results in

$$G(z^{-1}) = \frac{z(z+1)}{(z-1)^2}$$

Therefore, the locus of optimal root locations versus the parameter ρ is determined by substituting the two G 's into Eq. (9.40) to arrive at

$$1 + \rho \frac{z(z+1)^2}{(z-1)^2} = 0$$

The locus is drawn in Fig. 9.4. Note how the stable locus segments start from the open-loop poles and terminate at the zeros. Also note that, for each stable root, there is an unstable one that is its reciprocal.

The optimal damping ratio for any ρ is $\zeta \geq 0.7$. Designers have always known that picking $\zeta \geq 0.7$ produced a good compromise between speed of response, overshoot, and use of control; it also turns out, for this example, to be the *optimal* solution. This result makes sense because the optimal formulation is attempting to do the same thing that designers have always tried to do, that is, find the roots that achieve a good balance between the output error and the use of control.

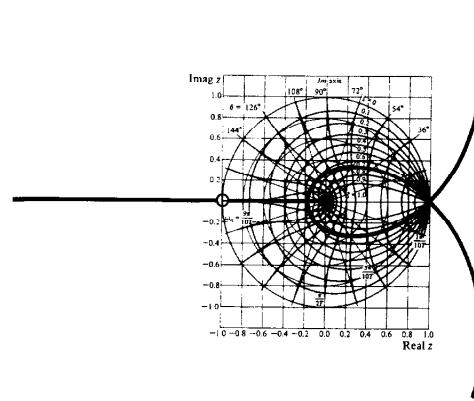
9.3.3 Eigenvector Decomposition

Now let us return to the optimal constant-gain solution for the general case. We can solve Eq. (9.33) by transforming to a new state that has a diagonal system matrix, and from this solution we can obtain the steady-state optimal control. Just as before, the eigenvalues of this matrix are such that the reciprocal of every eigenvalue is also an eigenvalue. Therefore, half the roots of the characteristic equation must be inside the unit circle and half must be outside. In this case, therefore, \mathcal{H}_c can be diagonalized to the form⁴

$$\mathcal{H}_c = \begin{bmatrix} \mathbf{E}^{-1} & 0 \\ 0 & \mathbf{E} \end{bmatrix}$$

⁴ In rare cases, \mathcal{H}_c will have repeated roots and cannot be made diagonal by a change of variables. In those cases, a small change in \mathbf{Q}_1 or \mathbf{Q}_2 will remove the problem, or else we must compute the Jordan form for \mathcal{H}_c^* [see Strang (1976)].

Figure 9.4
Symmetric root locus of Example 9.4



where \mathbf{E} is a diagonal matrix of the unstable roots ($|z| > 1$) and \mathbf{E}^{-1} is a diagonal matrix of the stable roots ($|z| < 1$). \mathcal{H}_c^* is obtained by the similarity transformation

$$\mathcal{H}_c^* = \mathbf{W}^{-1} \mathcal{H}_c \mathbf{W}$$

where \mathbf{W} is the matrix of eigenvectors of \mathcal{H}_c and can be written in block form as

$$\mathbf{W} = \begin{bmatrix} \mathbf{X}_f & \mathbf{X}_0 \\ \mathbf{A}_f & \mathbf{A}_0 \end{bmatrix}$$

where

$$\begin{bmatrix} \mathbf{X}_0 \\ \mathbf{A}_0 \end{bmatrix}$$

is the matrix of eigenvectors associated with the eigenvalues (roots) outside the unit circle and

$$\begin{bmatrix} \mathbf{x}_e \\ \mathbf{A}_e \end{bmatrix}$$

is the matrix of eigenvectors associated with the eigenvalues of \mathcal{H}_c that are inside the unit circle.

This same transformation matrix, \mathbf{W} , can be used to transform \mathbf{x} and λ to the normal modes of the system, that is,

$$\begin{bmatrix} \mathbf{x}^* \\ \lambda^* \end{bmatrix} = \mathbf{W}^{-1} \begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix},$$

where \mathbf{x}^* and λ^* are the normal modes. Conversely, we also have

$$\begin{bmatrix} \mathbf{x} \\ \lambda \end{bmatrix} = \mathbf{W} \begin{bmatrix} \mathbf{x}^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} \mathbf{X}_f & \mathbf{X}_0 \\ \mathbf{A}_f & \mathbf{A}_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \lambda^* \end{bmatrix}. \quad (9.41)$$

The solution to the coupled set of difference equations Eq. (9.33) can be simply stated in terms of the initial and final conditions and the normal modes, because the solution for the normal modes is given by

$$\begin{bmatrix} \mathbf{x}^* \\ \lambda^* \end{bmatrix}_y = \begin{bmatrix} \mathbf{E}^{-N} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^N \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \lambda^* \end{bmatrix}_0. \quad (9.42)$$

To obtain the steady state, we let N go to infinity; therefore $\mathbf{x}^*(N)$ goes to zero and, in general, $\lambda^*(N)$ would go to infinity because each element of \mathbf{E} is greater than one. So we see that the only sensible solution for the steady-state ($N \rightarrow \infty$) case is for $\mathbf{x}^*(0) = \mathbf{0}$ and therefore $\mathbf{x}^*(k) = \mathbf{0}$ for all k .⁵

From Eqs. (9.41) and (9.42) with $\lambda^*(k) \equiv \mathbf{0}$, we have

$$\mathbf{x}(k) = \mathbf{X}_f \mathbf{x}^*(k) = \mathbf{X}_f \mathbf{E}^{-k} \mathbf{x}^*(0), \quad (9.43)$$

$$\lambda(k) = \mathbf{A}_f \mathbf{x}^*(k) = \mathbf{A}_f \mathbf{E}^{-k} \mathbf{x}^*(0). \quad (9.44)$$

Therefore Eq. (9.43) leads to

$$\mathbf{x}^*(0) = \mathbf{E}^k \mathbf{X}_f^{-1} \mathbf{x}(k). \quad (9.45)$$

Thus, from Eqs. (9.44) and (9.45)

$$\lambda(k) = \mathbf{A}_f \mathbf{X}_f^{-1} \mathbf{x}(k) = \mathbf{S}_{\infty} \mathbf{x}(k),$$

which is the same form as our assumption Eq. (9.20) for the sweep method, so we conclude that

$$\mathbf{S}_{\infty} = \mathbf{A}_f \mathbf{X}_f^{-1} \quad (9.46)$$

⁵ From Eq. (9.41) we see that if λ^* is not zero then the state \mathbf{x} will grow in time and the system will be unstable. However, if the system is controllable we know that a control exists which will make the system stable and give a finite value to \mathcal{J} . Because we have the optimal control in Eq. (9.35), it must follow that the optimal system is stable and $\lambda^* \equiv \mathbf{0}$ if \mathbf{Q}_1 is such that all the states affect \mathcal{J} .

is the steady-state solution to (9.23), and that the control law for this system corresponding to \mathcal{J} with $N \rightarrow \infty$ is

$$\mathbf{u}(k) = -\mathbf{K}_{\infty} \mathbf{x}(k), \quad (9.47)$$

where, from Eqs. (9.46) and (9.28),

$$\mathbf{K}_{\infty} = (\mathbf{Q}_2 + \Gamma^T \mathbf{S}_{\infty} \Gamma)^{-1} \Gamma^T \mathbf{S}_{\infty} \Phi. \quad (9.48)$$

Furthermore, from Eq. (9.31), the cost associated with using this control law is

$$\mathcal{J}_{\infty} = \frac{1}{2} \mathbf{x}^T(0) \mathbf{S}_{\infty} \mathbf{x}(0).$$

In summary, the complete computational procedure is:

1. Compute eigenvalues of the system matrix \mathcal{H}_c defined by Eq. (9.34).
2. Compute eigenvectors associated with the stable ($|z| < 1$) eigenvalues of \mathcal{H}_c and call them

$$\begin{bmatrix} \mathbf{X}_f \\ \mathbf{A}_f \end{bmatrix}.$$

3. Compute control gain \mathbf{K}_{∞} from Eq. (9.48) with \mathbf{S}_{∞} given by Eq. (9.46).

We have already seen that the stable eigenvalues from Step 1 above are the resulting system closed-loop roots with constant gain \mathbf{K}_{∞} from Step 3. We can also show that the matrix \mathbf{X}_f of Eq. (9.41) is the matrix of eigenvectors of the optimal steady-state closed-loop system.

Most software packages (see MATLAB's dqr.m for the discrete case as developed here or lqr.m for the continuous case) use algorithms for these calculations that are closely related to the procedure above. In some cases the software gives the user a choice of the particular method to be used in the solution. Although it is possible to find the LQR gain \mathbf{K}_{∞} for a SISO system by picking optimal roots from a symmetric root locus and then using Ackermann's formula, it is easier to use the general lqr routines in MATLAB for either SISO or MIMO systems. If a locus of the optimal roots is desired, dqr.m can be used repetitively for varying values of elements in \mathbf{Q}_1 or \mathbf{Q}_2 .

◆ Example 9.5 Optimal Root Locus for the Double Mass-Spring System

Examine the optimal locus for the double mass-spring system of Appendix A.4 and comment on the relative merits of this approach vs. the pole-placement approach used in Example 8.3 for this system.

Solution. Using the same model and sample period as in Example 8.3, we find values for Φ , Γ which are then used with dqr.m to solve for the closed-loop roots for various values of

the weighting matrix \mathbf{Q}_1 . Because the output d is the only quantity of interest, it makes sense to use weighting on that state element only; that is

$$\mathbf{Q}_1 = \begin{bmatrix} q_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

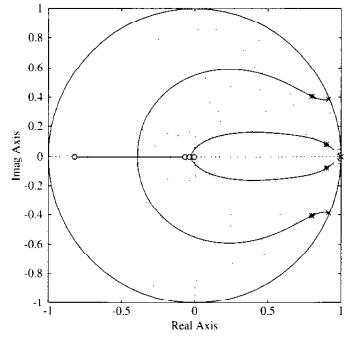
and \mathbf{Q}_2 is arbitrarily selected to be 1. The locus is determined by varying the value of q_{11} . The MATLAB script

```
q = logspace(-3,3,300); Q2=1
for i=1:100,
    Q1=diag([q(i);0;0;0]);
    [k,s,e]=dlqr(phi,gam,Q1,Q2);
end
```

produced a series of values of the closed-loop system roots (e above) which were plotted on the z -plane in Fig. 9.5. Note that the roots selected in Example 8.3 that gave the best results lie at the points marked by stars on the stable portion of the locus. So we see that the optimal solution provides guidance on where to pick the oscillatory poles. In fact, the results from this design led to the selection of the pole locations for the second case in Example 8.3 that yielded the superior response.

◆

Figure 9.5
Locus of optimal roots of Example 9.5



9.3.4 Cost Equivalents

It is sometimes useful to be able to find the discrete cost function defined by Eq. (9.11), which is the equivalent to an analog cost function of the form

$$\mathcal{J}_c = \frac{1}{2} \int_0^{N\tau} (\mathbf{x}^T \mathbf{Q}_{c1} \mathbf{x} + \mathbf{u}^T \mathbf{Q}_{c2} \mathbf{u}) d\tau. \quad (9.49)$$

Having this equivalence relationship will allow us to compute the \mathcal{J}_c of two discrete designs implemented with different sample periods and, therefore, will provide a fair basis for comparison. It will also provide a method for finding a discrete implementation of an optimal, continuous design—an “optimal” version of the emulation design method discussed in Section 7.2. In this section, we will develop the cost equivalence, and the use of it for an emulation design will be discussed in Section 9.3.5.

The integration of the cost in Eq. (9.49) can be broken into sample periods according to

$$\mathcal{J}_c = \sum_{k=0}^{N-1} \frac{1}{2} \int_{kT}^{(k+1)T} (\mathbf{x}^T \mathbf{Q}_{c1} \mathbf{x} + \mathbf{u}^T \mathbf{Q}_{c2} \mathbf{u}) d\tau, \quad (9.50)$$

and because

$$\mathbf{x}(kT + \tau) = \Phi(\tau)\mathbf{x}(kT) + \Gamma(\tau)\mathbf{u}(kT), \quad (9.51)$$

where

$$\Phi(\tau) = e^{\mathbf{F}\tau} \quad \Gamma(\tau) = \int_0^\tau e^{\mathbf{F}\eta} d\eta \mathbf{G}$$

as indicated by Eq. (4.58), substitution of Eq. (9.51) into (9.50) yields

$$\mathcal{J}_c = \frac{1}{2} \sum_{k=0}^{N-1} [(\mathbf{x}^T(k)\mathbf{u}^T(k))] \begin{bmatrix} \mathbf{Q}_{c1} & \mathbf{Q}_{c2} \\ \mathbf{Q}_{c2} & \mathbf{Q}_{c2} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{bmatrix}, \quad (9.52)$$

where

$$\begin{bmatrix} \mathbf{Q}_{c1} & \mathbf{Q}_{c2} \\ \mathbf{Q}_{c2} & \mathbf{Q}_{c2} \end{bmatrix} = \int_0^T \begin{bmatrix} \mathbf{F}^T(\tau) & \mathbf{0} \\ \mathbf{F}^T(\tau) & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{c1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{c2} \end{bmatrix} \begin{bmatrix} \mathbf{F}(\tau) & \Gamma(\tau) \\ \mathbf{0} & \mathbf{I} \end{bmatrix} d\tau. \quad (9.53)$$

Equation (9.53) is a relationship for the desired equivalent, discrete weighting matrices; however, we see that a complication has arisen in that there are now cross terms that weight the product of \mathbf{x} and \mathbf{u} . This can be circumvented by transforming the control to include a linear combination of the state, as we will show below; however, it is also possible to formulate the LQR solution so that it can account for the cross terms.

A method for computing the equivalent gains in Eq. (9.53), due to Van Loan (1978), is to form the matrix exponential⁶

$$\exp \begin{bmatrix} -\mathbf{F}^T & \mathbf{0} & \mathbf{Q}_{c1} & \mathbf{0} \\ -\mathbf{G}^T & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{c2} \\ \mathbf{0} & \mathbf{0} & \mathbf{F} & \mathbf{G} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} T \cong \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \mathbf{0} & \Phi_{22} \end{bmatrix}. \quad (9.54)$$

It will turn out that

$$\begin{aligned} \Phi_{22} &= \begin{bmatrix} \Phi & \Gamma \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\ \Phi_{11}^{-1} &= \Phi_{22}^T, \end{aligned}$$

however, one needs to calculate the matrix exponential Eq. (9.54) in order to find Φ_{12} . Because

$$\Phi_{12} = \Phi_{11} \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix},$$

we have the desired result

$$\begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix} = \Phi_{22}^T \Phi_{12}. \quad (9.55)$$

Routines to calculate the discrete equivalent of a continuous cost are available in some of the CAD control design packages (see `jdequiv.m` in the Digital Control Toolbox). Furthermore, MATLAB has an LQR routine (called `lqr.m`) that finds the discrete controller for a continuous cost and computes the necessary discrete cost in the process.

In summary, the continuous cost function J_c in Eq. (9.49) can be computed from discrete samples of the state and control by transforming the continuous weighting matrices, \mathbf{Q}_c 's, according to Eq. (9.55). The resulting discrete weighting matrices include cross terms that weight the product of \mathbf{x} and \mathbf{u} . The ability to compute the continuous cost from discrete samples of the state and control is useful for comparing digital controllers of a system with different sample rates and will also be useful in the emulation design method in the next section.

9.3.5 Emulation by Equivalent Cost

The emulation design method discussed in Section 7.2 took the approach that the design of the compensation be done in the continuous domain and the resulting $D(s)$ then be approximated using the digital filtering ideas of Chapter 6. This same approach can be applied when using optimal design methods [see Parsons (1982)]. First, an optimal design iteration is carried out in the continuous domain until the

⁶ Note that superscript $(\cdot)^T$ denotes transpose, whereas the entire matrix is multiplied by the sample period, T .

desired specifications are met. The discrete approximation is then obtained by calculating the discrete equivalent of the continuous cost function via Eq. (9.55), and then using that cost in the discrete LQR computation of Table 9.1, with modifications to account for the cross terms in the weighting matrices. These steps are all accomplished by MATLAB's `lqr.m`.

◆ Example 9.6 Design by Equivalent Cost

Examine the accuracy of the equivalent cost emulation method for the satellite attitude control example.

Solution. The continuous representation of the system is specified by \mathbf{F} and \mathbf{G} from Eq. (4.47). Use of a continuous LQR calculation (`lqr.m` in MATLAB) with

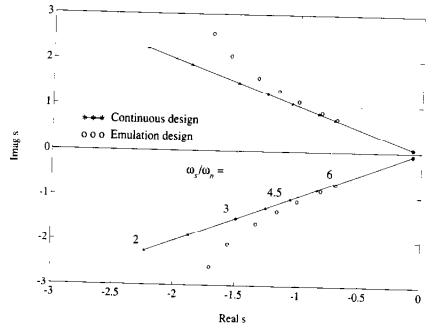
$$\mathbf{Q}_{c1} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

and

$$\mathbf{Q}_{c2} = [10000, 1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01]$$

results in s -plane roots exactly at $\zeta = 0.7$ as shown by the line in Fig. 9.6. Use of the `lqr.m` function computes the discrete controller that minimizes the same continuous cost, thus arriving at an emulation of the continuous design. The equivalent s -plane roots of these digital controllers are also plotted in Fig. 9.6. The figure shows that very little change in root locations occur by this emulation method. In fact the change in root locations is about 1% when

Figure 9.6
Degradations of s -plane root location for the optimal emulation method, Example 9.6



emulation advantages

sampling at six times the closed-loop natural frequency ($\omega_n/\omega_r = 6$) and increases to about 10% for $\omega_r/\omega_n = 3$. The accuracy of the zero-pole mapping emulation method was evaluated in Example 7.3; its calculations for a similar example showed that $\omega_r/\omega_n = 30$ resulted in a 10% change in root location, and $\omega_r/\omega_n = 6$ resulted in a 60% reduction in damping.

In general, use of the optimal emulation method will result in a digital controller whose performance will match the continuous design much closer than any of the emulation design methods discussed in Section 7.2 and Chapter 6. A requirement to use the method, however, is that the original continuous design be done using optimal methods so that the continuous weighting matrices Q_1 and Q_{r2} are available for the conversion because they are the parameters that define the design.

As discussed in Chapter 7, emulation design is attractive because it allows for the design process to be carried out before specifying the sample rate. Sampling degrades the performance of any system to varying degrees, and it is satisfying to be able to answer how good the control system can be in terms of disturbance rejection, steady-state errors, and so on, before the sampling degradation is introduced. Once the characteristics of a reasonable continuous design are known, the designer is better equipped to select a sample rate with full knowledge of how that selection will affect performance.

We acknowledge that the scenario just presented is not the reality of the typical digital design process. Usually, due to the pressure of schedules, the computer and sample rate are specified long before the controls engineers have a firm grasp on the controller algorithm. Given that reality, the most expedient path is to perform the design directly in the discrete domain and obtain the best possible with that constraint. Furthermore, many design exercises are relatively minor modifications to previous designs. In these cases, too, the most expedient path is to work directly in the discrete domain.

But we maintain that the most desirable design scenario is to gain knowledge of the effects of sampling by first performing the design in the continuous domain, then performing discrete designs. In this case, the emulation method described here is a useful tool to obtain quickly a controller to be implemented digitally or to use as a basis for further refinement in the discrete domain.

9.4 Optimal Estimation

Optimal estimation methods are attractive because they handle multi-output systems easily and allow a designer quickly to determine many good candidate designs of the estimator gain matrix, L . We will first develop the least squares estimation solution for the static case as it is the basis for optimal estimation, then

we will extend that to the time-varying optimal estimation solution (commonly known as the "Kalman filter"), and finally show the correspondence between the Kalman filter and the time-varying optimal control solution. Following the same route that we did for the optimal control solution, we will then develop the optimal estimation solution with a steady-state L -matrix. In the end, this amounts to another method of computing the L -matrix in the equations for the current estimator, Eqs. (8.33) and (8.34), which are

$$\hat{x}(k) = \tilde{x}(k) + L(y(k) - \tilde{y}(k)), \quad (9.56)$$

where

$$\tilde{x}(k+1) = \Phi\tilde{x}(k) + \Gamma u(k);$$

however, now L will be based on minimizing estimation errors rather than picking dynamic characteristics of the estimator error equation.

9.4.1 Least-Squares Estimation

Suppose we have a linear static process given by

$$y = Hx + v, \quad (9.57)$$

where y is a $p \times 1$ measurement vector, x is an $n \times 1$ unknown vector, v is a $p \times 1$ measurement error vector, and H is the matrix relating the measurements to the unknowns. We want to determine the best estimate of x given the measurements y . Often, the system is overdetermined; that is, there are more measurements in y than the unknown vector, x . A good way to find the best estimate of x is to minimize the sum of the squares of v , the fit error. This is called the **least squares** solution. This is both sensible and very convenient analytically. Proceeding, the sum of squares can be written as

$$\mathcal{J} = \frac{1}{2}v^T v = \frac{1}{2}(y - Hx)^T (y - Hx) \quad (9.58)$$

and, in order to minimize this expression, we take the derivative with respect to the unknown, that is

$$\frac{\partial \mathcal{J}}{\partial x} = (y - Hx)^T (-H) = 0 \quad (9.59)$$

which results in

$$H^T y = H^T H x,$$

so that

$$\hat{x} = [H^T H]^{-1} H^T y \quad (9.60)$$

where \hat{x} designates the "best estimate" of x . Note that the matrix to be inverted is $n \times n$ and that p must be $\geq n$ for it to be full rank and the inverse to exist. If there

least squares estimate

estimate accuracy

are fewer measurements than unknowns ($p < n$), there are too few measurements to determine a unique value of \mathbf{x} .

The difference between the estimate and the actual value of \mathbf{x} is

$$\hat{\mathbf{x}} - \mathbf{x} = [\mathbf{H}^T \mathbf{H}]^{-1} \mathbf{H}^T (\mathbf{H}\mathbf{x} + \mathbf{v}) - \mathbf{x} \\ = [\mathbf{H}^T \mathbf{H}]^{-1} \mathbf{H}^T \mathbf{v}. \quad (9.61)$$

Equation (9.61) shows that, if \mathbf{v} has zero mean, the error in the estimate, $\hat{\mathbf{x}} - \mathbf{x}$, will also be zero mean, sometimes referred to as an **unbiased estimate**.

The covariance of the estimate error, \mathbf{P} , is defined to be

$$\begin{aligned} \mathbf{P} &= \mathcal{E}((\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^T) \\ &= \mathcal{E}([\mathbf{H}^T \mathbf{H}]^{-1} \mathbf{H}^T \mathbf{v} \mathbf{v}^T \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1}) \\ &= (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathcal{E}(\mathbf{v} \mathbf{v}^T) \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1}. \end{aligned} \quad (9.62)$$

If the elements in the noise vector, \mathbf{v} , are uncorrelated with one another, $\mathcal{E}(\mathbf{v} \mathbf{v}^T)^7$ is a diagonal matrix, which we shall call \mathbf{R} . Furthermore, if all the elements of \mathbf{v} have the same uncertainty, then all the diagonal elements of \mathbf{R} are identical, and

$$\mathcal{E}(\mathbf{v} \mathbf{v}^T) = \mathbf{R} = \mathbf{I}\sigma^2. \quad (9.63)$$

where σ is the rms value of each element in \mathbf{v} . In this case, Eq. (9.62) can be written as

$$\mathbf{P} = (\mathbf{H}^T \mathbf{H})^{-1} \sigma^2 \quad (9.64)$$

and is a measure of how well we can estimate the unknown \mathbf{x} . The square root of the diagonal elements of \mathbf{P} represent the rms values of the errors in each element in \mathbf{x} .

◆ Example 9.7 Least-Squares Fit

The monthly sales (in thousands \$) for the first year of the Mewisham Co. are given by

$$\mathbf{y}^T = [0.2 \ 0.5 \ 1.1 \ 1.2 \ 1.1 \ 1.3 \ 1.1 \ 1.2 \ 2.0 \ 1.2 \ 2.2 \ 4.0].$$

Find the least-squares fit parabola to this data and use that to predict what the monthly sales will be during the second year. Also state what the predicted accuracy of the parabolic coefficients are, assuming that the rms accuracy of the data is \$700.

Solution. The solution is obtained using Eq. (9.60), where \mathbf{H} contains the parabolic function to be fit. Each month's sales obeys

$$y_i = a_0 + a_1 t_i + a_2 t_i^2 + v_i,$$

⁷ \mathcal{E} is called the **expectation** and effectively means the average of the quantity in $\{ \cdot \}$.

so that, in vector form

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \end{bmatrix},$$

and we see that

$$\mathbf{H} = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \\ \vdots & \vdots & \vdots \end{bmatrix} \quad t_i = i, \quad i = 1, 12$$

and

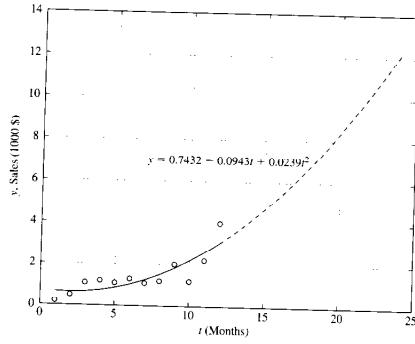
$$\mathbf{x} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}.$$

Evaluation of Eq. (9.60) produces an estimate of \mathbf{x}

$$\hat{\mathbf{x}} = \begin{bmatrix} 0.7432 \\ -0.0943 \\ 0.0239 \end{bmatrix}$$

which is used to plot the "best fit" parabola along with the raw data in Fig. 9.7. The data used to determine the parabola only occurred during the first 12 months, after that the parabola is extrapolated.

Figure 9.7
Least squares fit of parabola to data in Example 9.7



Equation (9.64), with \mathbf{H} as above and $\sigma^2 = 0.49$ (σ was given to be 0.7), shows that

$$\mathbf{P} = \begin{bmatrix} 0.5234 & -0.1670 & 0.0111 \\ -0.1670 & 0.0655 & -0.0048 \\ 0.0111 & -0.0048 & 0.0004 \end{bmatrix},$$

which means that the rms accuracy of the coefficients are

$$\sigma_{a_0} = \sqrt{0.5234} = 0.7235$$

$$\sigma_{a_1} = \sqrt{0.0655} = 0.2559$$

$$\sigma_{a_2} = \sqrt{0.0004} = 0.0192.$$

Weighted Least Squares

In many cases, we know *a priori* that some measurements are more accurate than others so that all the diagonal elements in \mathbf{R} are not the same. In this case, it makes sense to weight the measurement errors higher for those measurements known to be more accurate because that will cause those measurements to have a bigger influence on the cost minimization. In other words, the cost function in Eq. (9.58) needs to be modified

$$\mathcal{J} = \frac{1}{2} \mathbf{v}^T \mathbf{W} \mathbf{v}, \quad (9.65)$$

where \mathbf{W} is a diagonal weighting matrix whose elements are in some way inversely related to the uncertainty of the corresponding element of \mathbf{v} . Performing the same algebra as for the unweighted case above, we find that the best **weighted least squares** solution is given by

$$\hat{\mathbf{x}} = [\mathbf{H}^T \mathbf{W} \mathbf{H}]^{-1} \mathbf{H}^T \mathbf{W} \mathbf{y}. \quad (9.66)$$

The covariance of this estimate also directly follows the development of Eq. (9.62) and results in

$$\mathbf{P} = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \mathbb{E}[\mathbf{v} \mathbf{v}^T] \mathbf{W} \mathbf{H} (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1}. \quad (9.67)$$

A logical choice for \mathbf{W} is to let it be inversely proportional to \mathbf{R} , thus weighting the square of the measurement errors exactly in proportion to the inverse of their *a priori* mean square error, that is, let

$$\mathbf{W} = \mathbf{R}^{-1}. \quad (9.68)$$

This choice of weighting matrix is proven in Section 12.7 to minimize the trace of \mathbf{P} and is called the **best linear unbiased estimate**. With this choice, Eq. (9.66) becomes

$$\hat{\mathbf{x}} = [\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}]^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{y}. \quad (9.69)$$

and Eq. (9.67) reduces to

$$\mathbf{P} = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1}. \quad (9.70)$$

Recursive Least Squares

The two least-squares algorithms above, Eqs. (9.60) and (9.69), are both **batch** algorithms in that all the data is obtained and then processed in one calculation. For an estimation problem that runs for a long time, the measurement vector would become very large, and one would have to wait until the problem was complete in order to calculate the estimate. The recursive formulation solves both these difficulties by performing the calculation in small time steps. The ideas are precisely the same, that is, a weighted least squares calculation is being performed. But now we break the problem into old data, for which we already found $\hat{\mathbf{x}}$, and new data, for which we want a correction to $\hat{\mathbf{x}}$, so that the overall new $\hat{\mathbf{x}}$ is adjusted for the newly acquired data.

The problem is stated as

$$\begin{bmatrix} \mathbf{y}_o \\ \mathbf{y}_n \end{bmatrix} = \begin{bmatrix} \mathbf{H}_o \\ \mathbf{H}_n \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{v}_o \\ \mathbf{v}_n \end{bmatrix}, \quad (9.71)$$

where the subscript o represents old data and n represents new data. The best estimate of \mathbf{x} given all the data follows directly from Eq. (9.69) and can be written as

$$\begin{bmatrix} \mathbf{H}_o \\ \mathbf{H}_n \end{bmatrix}^T \begin{bmatrix} \mathbf{R}_o^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_n^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{H}_o \\ \mathbf{H}_n \end{bmatrix} \hat{\mathbf{x}} = \begin{bmatrix} \mathbf{H}_o \\ \mathbf{H}_n \end{bmatrix}^T \begin{bmatrix} \mathbf{R}_o^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_n^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{y}_o \\ \mathbf{y}_n \end{bmatrix}, \quad (9.72)$$

where $\hat{\mathbf{x}}$ is the best estimate of \mathbf{x} given all the data, old and new. Let's define $\bar{\mathbf{x}}$ as

$$\bar{\mathbf{x}} = \hat{\mathbf{x}}_o + \delta \hat{\mathbf{x}}, \quad (9.73)$$

where $\hat{\mathbf{x}}_o$ is the best estimate given only the old data, \mathbf{y}_o . We want to find an expression for the correction to this estimate, $\delta \hat{\mathbf{x}}$, given the new data. Since $\hat{\mathbf{x}}_o$ was the best estimate given the old data, it satisfies

$$\mathbf{H}_o^T \mathbf{R}_o^{-1} \mathbf{H}_o \hat{\mathbf{x}}_o = \mathbf{H}_o^T \mathbf{R}_o^{-1} \mathbf{y}_o. \quad (9.74)$$

Expanding out the terms in Eq. (9.72) and using Eqs. (9.73) and (9.74) yields

$$\mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{H}_n \hat{\mathbf{x}}_o + [\mathbf{H}_o^T \mathbf{R}_o^{-1} \mathbf{H}_o + \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{H}_n] \delta \hat{\mathbf{x}} = \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{y}_n, \quad (9.75)$$

which can be solved for the desired result

$$\delta \hat{\mathbf{x}} = [\mathbf{H}_o^T \mathbf{R}_o^{-1} \mathbf{H}_o + \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{H}_n]^{-1} \mathbf{H}_n^T \mathbf{R}_n^{-1} (\mathbf{y}_n - \mathbf{H}_n \hat{\mathbf{x}}_o). \quad (9.76)$$

Equation (9.70) defined the covariance of the estimate, and in terms of the old data would be written as

$$\mathbf{P}_o = (\mathbf{H}_o^T \mathbf{R}_o^{-1} \mathbf{H}_o)^{-1}, \quad (9.77)$$

so that Eq. (9.76) reduces to

$$\delta\hat{\mathbf{x}} = [\mathbf{P}_o^{-1} + \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{H}_n]^{-1} \mathbf{H}_n^T \mathbf{R}_n^{-1} (\mathbf{y}_n - \mathbf{H}_n \hat{\mathbf{x}}_o), \quad (9.78)$$

and we see that the correction to the old estimate can be determined if we simply know the old estimate and its covariance. Note that the correction to $\hat{\mathbf{x}}$ is proportional to the difference between the new data \mathbf{y}_n and the estimate of the new data $\mathbf{H}_n \hat{\mathbf{x}}_o$ based on the old \mathbf{x} .

By analogy with the weighted least squares, the covariance of the new estimate is

$$\mathbf{P}_n = [\mathbf{P}_o^{-1} + \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{H}_n]^{-1}. \quad (9.79)$$

Note here that it is no longer necessary for there to be more new measurements than the elements in \mathbf{x} . The only requirement is that \mathbf{P}_o be full rank, which could be satisfied by virtue of \mathbf{P}_o being full rank. In other words, in Example 9.7 it would have been possible to start the process with the first three months of sales, then recursively update the parabolic coefficients using one month's additional sales at a time. In fact, we will see in the Kalman filter that it is typical for the new \mathbf{y} to have fewer elements than \mathbf{x} .

To summarize the procedure, we start by assuming that \mathbf{x}_o and \mathbf{P}_o are available from previous calculations.

- Compute the new covariance from Eq. (9.79)
- $\mathbf{P}_n = [\mathbf{P}_o^{-1} + \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{H}_n]^{-1}.$
- Compute the new value of $\hat{\mathbf{x}}$ using Eqs. (9.73), (9.78), and (9.79)
- $\hat{\mathbf{x}} = \hat{\mathbf{x}}_o + \mathbf{P}_n \mathbf{H}_n^T \mathbf{R}_n^{-1} (\mathbf{y}_n - \mathbf{H}_n \hat{\mathbf{x}}_o).$ (9.80)
- Take new data and repeat the process.

This algorithm assigns relative weighting to the old $\hat{\mathbf{x}}$ vs. new data based on their relative accuracies, similarly to the weighted least squares. For example, if the old data produced an extremely accurate estimate so that \mathbf{P}_o was almost zero, then Eq. (9.79) shows that \mathbf{P}_n is $\cong 0$ and Eq. (9.80) shows that the new estimate will essentially ignore the new data. On the other hand, if the old estimates are very poor, that is, \mathbf{P}_o is very large, Eq. (9.79) shows that

$$\mathbf{P}_n \cong [\mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{H}_n]^{-1}.$$

and Eq. (9.80) shows that

$$\hat{\mathbf{x}} \cong [\mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{H}_n]^{-1} \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{y}_n,$$

which ignores the old data. Most cases are somewhere between these two extremes, but the fact remains that this recursive-weighted-least-squares algorithm weights the old and new data according to the associated covariance.

recursive least-square procedure

9.4.2 The Kalman Filter

Now consider a discrete dynamic plant

$$\mathbf{x}(k+1) = \Phi \mathbf{x}(k) + \Gamma \mathbf{u}(k) + \Gamma_v \mathbf{w}(k) \quad (9.81)$$

with measurements

$$\mathbf{y}(k) = \mathbf{H} \mathbf{x}(k) + \mathbf{v}(k), \quad (9.82)$$

where the **process noise** $\mathbf{w}(k)$ and **measurement noise** $\mathbf{v}(k)$ are random sequences with zero mean, that is

$$\mathcal{E}\{\mathbf{w}(k)\} = \mathcal{E}\{\mathbf{v}(k)\} = \mathbf{0},$$

have no time correlation or are "white" noise, that is

$$\mathcal{E}\{\mathbf{w}(i)\mathbf{w}^T(j)\} = \mathcal{E}\{\mathbf{v}(i)\mathbf{v}^T(j)\} = \mathbf{0} \quad \text{if } i \neq j,$$

and have covariances or mean square "noise levels" defined by

$$\mathcal{E}\{\mathbf{w}(k)\mathbf{w}^T(k)\} = \mathbf{R}_w, \quad \mathcal{E}\{\mathbf{v}(k)\mathbf{v}^T(k)\} = \mathbf{R}_v.$$

We allow \mathbf{L} in Eq. (9.56) to vary with the time step, and we wish to pick $\mathbf{L}(k)$ so that the estimate of $\mathbf{x}(k)$, given all the data up to and including time k , is optimal.

Let us pretend temporarily that without using the current measurement $\mathbf{y}(k)$, we already have a prior estimate of the state at the time of a measurement, which we will call $\tilde{\mathbf{x}}(k)$. The problem at this point is to update this old estimate based on the current new measurement.

Comparing this problem to the recursive least squares, we see that the estimation measurement equation Eq. (9.82) relates the new measurements to \mathbf{x} just as the lower row in Eq. (9.71) does; hence the optimal state estimation solution is given by Eq. (9.80), where \mathbf{x}_o takes the role of $\tilde{\mathbf{x}}(k)$, $\mathbf{P}_n = \mathbf{P}(k)$, $\mathbf{H}_n = \mathbf{H}$, and $\mathbf{R}_n = \mathbf{R}_v$. The solution equations are

$$\hat{\mathbf{x}}(k) = \tilde{\mathbf{x}}(k) + \mathbf{L}(k)(\mathbf{y}(k) - \mathbf{H}\tilde{\mathbf{x}}(k)), \quad (9.83)$$

where,

$$\mathbf{L}(k) = \mathbf{P}(k) \mathbf{H}^T \mathbf{R}_v^{-1}. \quad (9.84)$$

Equation (9.79) is used to find $\mathbf{P}(k)$, where we now call the old covariance $\mathbf{M}(k)$ instead of \mathbf{P}_o , thus

$$\mathbf{P}(k) = [\mathbf{M}^{-1} + \mathbf{H}^T \mathbf{R}_v^{-1} \mathbf{H}]^{-1}. \quad (9.85)$$

The size of the matrix to be inverted in Eq. (9.85) is $n \times n$, where n is the dimension of \mathbf{x} . For the Kalman filter, \mathbf{y} usually has fewer elements than \mathbf{x} , and it is more efficient to use the matrix inversion lemma (See Eq. (C.6) in Appendix C) to convert Eq. (9.85) to

$$\mathbf{P}(k) = \mathbf{M}(k) - \mathbf{M}(k) \mathbf{H}^T (\mathbf{H} \mathbf{M}(k) \mathbf{H}^T + \mathbf{R}_v)^{-1} \mathbf{H} \mathbf{M}(k). \quad (9.86)$$

$\mathbf{M}(k)$ is the covariance (or expected mean square error) of the state estimate, $\hat{\mathbf{x}}(k)$, before the measurement. The state estimate after the measurement, $\tilde{\mathbf{x}}(k)$, has an error covariance $\mathbf{P}(k)$.

The idea of combining the previous estimate with the current measurement based on the relative accuracy of the two quantities—the recursive least-squares concept—was the genesis for the relationships in Eq. (9.83) and Eq. (9.85) and is one of the basic ideas of the Kalman filter. The other key idea has to do with using the known dynamics of \mathbf{x} to predict its behavior between samples, an idea that was discussed in the development of the estimator in Chapter 8.

Use of dynamics in the propagation of the estimate of \mathbf{x} between $k - 1$ and k did not come up in the static least-squares estimation, but here this issue needs to be addressed. The state estimate at $k - 1$, given data up through $k - 1$, is called $\hat{\mathbf{x}}(k - 1)$, whereas we defined $\tilde{\mathbf{x}}(k)$ to be the estimate at k given the same data up through $k - 1$. In the static least squares, these two quantities were identical and both called \mathbf{x}_n because it was presumed to be a constant; but here the estimates differ due to the fact that the state will change according to the system dynamics as time passes. Specifically, the estimate $\tilde{\mathbf{x}}(k)$ is found from $\hat{\mathbf{x}}(k - 1)$ by using Eq. (9.81) with $\mathbf{w}(k - 1) = \mathbf{0}$, because we know that this is the expected value of $\mathbf{x}(k)$ since the expected value of the plant noise, $\mathcal{E}\{\mathbf{w}(k - 1)\}$ is zero. Thus

$$\tilde{\mathbf{x}}(k) = \Phi\hat{\mathbf{x}}(k - 1) + \Gamma\mathbf{u}(k - 1). \quad (9.87)$$

The change in estimate from $\hat{\mathbf{x}}(k - 1)$ to $\tilde{\mathbf{x}}(k)$ is called a “time update,” whereas the change in the estimate from $\hat{\mathbf{x}}(k)$ to $\tilde{\mathbf{x}}(k)$ as given by Eq. (9.83) is a “measurement update,” which occurs at the fixed time k but expresses the improvement in the estimate due to the measurement $\mathbf{y}(k)$. The same kind of time and measurement updates apply to the estimate covariances, \mathbf{P} and \mathbf{M} : \mathbf{P} represents the estimate accuracy immediately after a measurement, whereas \mathbf{M} is the propagated value of \mathbf{P} and is valid just before measurements. From Eq. (9.81) and Eq. (9.87), we see that

$$\mathbf{x}(k + 1) - \tilde{\mathbf{x}}(k + 1) = \Phi(\mathbf{x}(k) - \hat{\mathbf{x}}(k)) + \Gamma_1\mathbf{w}(k). \quad (9.88)$$

which we will use to find the covariance of the state at time $k + 1$ before taking $\mathbf{y}(k + 1)$ into account

$$\mathbf{M}(k + 1) = \mathcal{E}[(\mathbf{x}(k + 1) - \tilde{\mathbf{x}}(k + 1))(\mathbf{x}(k + 1) - \tilde{\mathbf{x}}(k + 1))^T].$$

If the measurement noise, \mathbf{v} , and the process noise, \mathbf{w} , are uncorrelated so that $\mathbf{x}(k)$ and $\mathbf{w}(k)$ are also uncorrelated, the cross product terms vanish and we find that

$$\mathbf{M}(k + 1) = \mathcal{E}[\Phi(\mathbf{x}(k) - \hat{\mathbf{x}}(k))(\mathbf{x}(k) - \hat{\mathbf{x}}(k))^T\Phi^T + \Gamma_1\mathbf{w}(k)\mathbf{w}^T(k)\Gamma_1^T]. \quad (9.89)$$

But because

$$\mathbf{P}(k) = \mathcal{E}\{(\mathbf{x}(k) - \hat{\mathbf{x}}(k))(\mathbf{x}(k) - \hat{\mathbf{x}}(k))^T\} \quad \text{and} \quad \mathbf{R}_w = \mathcal{E}\{\mathbf{w}(k)\mathbf{w}^T(k)\},$$

Eq. (9.89) reduces to

$$\mathbf{M}(k + 1) = \Phi\mathbf{P}(k)\Phi^T + \Gamma_1\mathbf{R}_w\Gamma_1^T. \quad (9.90)$$

This completes the required relations for the optimal, time-varying gain, state estimation, commonly referred to as the Kalman filter. A summary of the required relations is:

- At the measurement time (measurement update)

$$\tilde{\mathbf{x}}(k) = \hat{\mathbf{x}}(k) + \mathbf{P}(k)\mathbf{H}^T\mathbf{R}_v^{-1}(\mathbf{y}(k) - \mathbf{H}\hat{\mathbf{x}}(k)), \quad (9.91)$$

where

$$\mathbf{P}(k) = \mathbf{M}(k) - \mathbf{M}(k)\mathbf{H}^T(\mathbf{H}\mathbf{M}(k)\mathbf{H}^T + \mathbf{R}_v)^{-1}\mathbf{H}\mathbf{M}(k). \quad (9.92)$$

- Between measurements (time update)

$$\tilde{\mathbf{x}}(k + 1) = \Phi\tilde{\mathbf{x}}(k) + \Gamma\mathbf{u}(k) \quad (9.93)$$

and

$$\mathbf{M}(k + 1) = \Phi\mathbf{P}(k)\Phi^T + \Gamma_1\mathbf{R}_w\Gamma_1^T, \quad (9.94)$$

where the initial conditions for $\tilde{\mathbf{x}}(0)$ and $\mathbf{M}(0) = \mathcal{E}[\tilde{\mathbf{x}}(0)\tilde{\mathbf{x}}^T(0)]$ must be assumed to be some value for initialization.

Because \mathbf{M} is time-varying, so will be the estimator gain, \mathbf{L} , given by Eq. (9.84). Furthermore, we see that the structure of the estimation process is exactly the same as the current estimator given by Eq. (9.56), the difference being that \mathbf{L} is time varying and determined so as to provide the minimum estimation errors, given *a priori* knowledge of the process noise magnitude, \mathbf{R}_w , the measurement noise magnitude, \mathbf{R}_v , and the covariance initial condition, $\mathbf{M}(0)$.

◆ Example 9.8 Time-Varying Kalman Filter Gains

Solve for $\mathbf{L}(k)$ for the satellite attitude control problem in Example 9.3 assuming the angle, θ , is sensed with a measurement noise covariance

$$R_v = 0.1 \text{ deg}^2.$$

Assume the process noise is due to disturbance torques acting on the spacecraft with the disturbance-input distribution matrix

$$\mathbf{G}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

and assume several values for the mean square magnitude of this disturbance:

$$R_w = 0.001, 0.01, 0.1 (\text{deg}^2/\text{sec}^4).$$

Solution. Because only θ is directly sensed, we have from Eq. (4.47)

$$H = [1 \quad 0].$$

The time varying estimator gain, $L(k)$, is found by evaluating Eqs. (9.84), (9.92), and (9.94). In order to start these recursive equations, some value for $M(0)$ is required. Although somewhat arbitrary, a value of

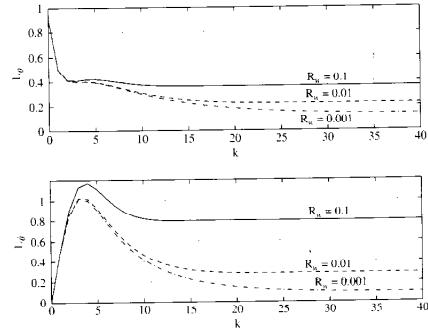
$$M(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

was selected in order to indicate that the initial rms uncertainty in θ was 1° and $\dot{\theta}$ was 1°/sec. The gain time histories are sensitive to this initial condition estimate during the initial transient, but the steady final values are not affected.

Figure 9.8 shows the time history of the L 's for the values given above. We see from the figure that after an initial settling time, the estimator gains essentially reach a steady state. Just as for the control problem, where the feedback gain reached a steady value early in the problem time, the eventual steady value for L occurs for all linear constant coefficient systems. The subject of the next section is a method to compute the value of this steady-state gain matrix so that it can be used in place of the time-varying one, thus eliminating the need to go through the rather lengthy recursive computation of Eq. (9.92) and Eq. (9.94).

Given an actual design problem, one can often assign a meaningful value to R_y , which is based on the sensor accuracy. The same cannot be said for R_u . The

Figure 9.8
Example of estimator gains versus time



divergent filters

assumption of white process noise is often a mathematical artifact that is used because of the ease of solving the resulting optimization problem. Physically, R_u is crudely accounting for unknown disturbances, whether they be steps, white noise, or somewhere in between, and for imperfections in the plant model.

If there is a random disturbance that is time correlated—that is, **colored noise**—it can be accurately modeled by augmenting Φ with a coloring filter that converts a white-noise input into time-correlated noise, thus Eq. (9.81) can also be made to describe nonwhite disturbances. In practice, however, this is often not done due to the complexity. Instead, the disturbances are assumed white, and the noise intensity is adjusted to give acceptable results in the presence of expected disturbances, whether time-correlated or not.

If R_u was chosen to be zero due to a lack of knowledge of a precise noise model, the estimator gain would eventually go to zero. This is so because the optimal thing to do in the idealistic situation of *no* disturbances and a *perfect* plant model is to estimate open loop after the initial condition errors have *completely* died out; after all, the very best filter for the noisy measurements is to totally ignore them. In practice, this will not work because there are always some disturbances and the plant model is never perfect; thus, the filter with zero gain will drift away from reality and is referred to as a **divergent filter**.⁸ If an estimator mode with zero gain was also naturally unstable, the estimator error would diverge from reality very quickly and likely result in saturation of the computer. We therefore are often forced to pick values of R_u and sometimes Γ , “out of a hat” in the design process in order to assure that no modes of the estimator are without feedback and that the estimator will track all modes of the actual system. The disturbance noise model should be selected to approximate that of the actual known disturbances when practical, but the designer often settles on acceptable values based on the quality of the estimation that results in subsequent simulations including all known disturbances, white and otherwise.

It is possible to include a nonzero process noise input in the plant model and yet still end up with a divergent filter [Bryson (1978)]. This can arise when the process noise is modeled so that it does not affect some of the modes of the system. Bryson showed that the filter will not be divergent if you select Γ so that the system, (Φ, Γ) , is **controllable** and all diagonal elements of R_u are nonzero.

For an implementation of a time-varying filter, initial conditions for M and \hat{x} are also required. Physically, they represent the *a priori* estimate of the accuracy of $\hat{x}(0)$, which in turn represents the *a priori* estimate of the state. In some cases, there might be test data of some kind to support their intelligent choice; however, that is not typical. In lieu of any better information, one could logically assume that the components of $\hat{x}(0)$ contained in y are equal to the first measurement, and the remaining components are equal to zero. Similarly, the components that

⁸ A filter with one or more modes that is running open loop is sometimes dubbed “oblivious” or “fat, dumb, and happy” because it ignores the measurements. It is somewhat analogous to the person whose mind is made up and not interested in facts.

represent a measured component could be logically set to \mathbf{R}_v and the remaining components set to a high value.

In obtaining the values of \mathbf{L} in Fig. 9.8, it was necessary to solve for the time history of \mathbf{P} . Because \mathbf{P} is the covariance of the estimation errors, we can sometimes use \mathbf{P} as an indicator of estimation accuracy, provided that the values of \mathbf{R}_w and \mathbf{R}_v are based on some knowledge of the actual noise characteristics and that \mathbf{w} and \mathbf{v} are approximately white.

The intent of this section and example is to give some insight into the nature of the solution so as to motivate and provide a basis for the following section. Readers interested in the application of Kalman filters are encouraged to review works devoted to that subject, such as Bryson and Ho (1975), Anderson and Moore (1979), and Stengel (1986).

9.4.3 Steady-State Optimal Estimation

As shown by Example 9.8, the estimator gains will eventually reach a steady-state value if enough time passes. This is so because the values of \mathbf{M} and \mathbf{P} reach a steady value. Because of the substantial simplification in the controller afforded by a constant estimator gain matrix, it is often desirable to determine the constant gain during the design process and to implement that constant value in the controller. As discussed in Section 9.2, many control systems run for very long times and can be treated mathematically as if they run for an infinite time. In this case the constant gain is the optimal because the early transient period has no significant effect. Whatever the motivation, a constant-gain Kalman filter is identical in structure to the estimator discussed in Chapter 8, the only difference being that the gain, \mathbf{L}_* , is determined so that the estimate errors are minimized for the assumed level of process and measurement noise. This approach replaces the pole-placement method of finding the estimator gain and has the highly desirable feature that it can be applied to MIMO systems.

The equations to be solved that determine \mathbf{M} and \mathbf{P} are Eq. (9.92) and Eq. (9.94). Repeated, they are

$$\mathbf{P}(k) = \mathbf{M}(k)\mathbf{H}^T(\mathbf{HM}(k)\mathbf{H}^T + \mathbf{R}_v)^{-1}\mathbf{HM}(k), \quad (9.92)$$

$$\mathbf{M}(k+1) = \Phi\mathbf{P}(k)\Phi^T + \Gamma_1\mathbf{R}_w\Gamma_1^T \quad (9.94)$$

Comparing Eqs. (9.92) and (9.94) to the optimal control recursion relationships, Eq. (9.24) and Eq. (9.25)

$$\mathbf{M}(k) = \mathbf{S}(k) - \mathbf{S}(k)\Gamma[\mathbf{Q}_1 + \Gamma^T\mathbf{S}(k)\Gamma]^{-1}\Gamma^T\mathbf{S}(k), \quad (9.25)$$

$$\mathbf{S}(k) = \Phi^T\mathbf{M}(k+1)\Phi + \mathbf{Q}_1. \quad (9.24)$$

we see that they are precisely of the same form! The only exception is that Eq. (9.94) goes forward instead of backward as Eq. (9.24) does. Therefore, we can simply change variables and directly use the steady-state solution of the

duality

control problem as the desired steady-state solution to the estimation problem, even though the equations are solved in opposite directions in time.

Table 9.1 lists the correspondences that result by direct comparison of the control and estimation recursion relations: Eq. (9.25) with Eq. (9.92) and Eq. (9.24) with Eq. (9.94).

By analogy with the control problem, Eqs. (9.92) and (9.94) must have arisen from two coupled equations with the same form as Eq. (9.33). Using the correspondences in Table 9.1, the control Hamiltonian in Eq. (9.34) becomes the estimation Hamiltonian

$$\mathcal{H}_e = \begin{bmatrix} \Phi^T + \mathbf{H}^T\mathbf{R}_v^{-1}\mathbf{H}\Phi^{-1}\Gamma_1\mathbf{R}_w\Gamma_1^T & -\mathbf{H}^T\mathbf{R}_v^{-1}\mathbf{H}\Phi^{-1} \\ -\Phi^{-1}\Gamma_1\mathbf{R}_w\Gamma_1^T & \Phi^{-1} \end{bmatrix}. \quad (9.95)$$

Therefore, the steady-state value of \mathbf{M} is deduced by comparison with Eq. (9.46) and is

$$\mathbf{M}_* = \mathbf{A}_f\mathbf{X}_f^{-1}, \quad (9.96)$$

where

$$\begin{bmatrix} \mathbf{X}_f \\ \mathbf{A}_f \end{bmatrix}$$

are the eigenvectors of \mathcal{H}_e associated with its stable eigenvalues. Hence, from Eqs. (9.92) and (9.84) after some manipulation we find the steady state Kalman-filter gain to be⁹

$$\mathbf{L}_* = \mathbf{M}_*\mathbf{H}^T(\mathbf{HM}_*\mathbf{H}^T + \mathbf{R}_v)^{-1}. \quad (9.97)$$

This is a standard calculation in MATLAB's `kalman.m`. Sometimes this solution is referred to as the **linear quadratic Gaussian (LQG)** problem because it is often assumed in the derivation that the noise has a Gaussian distribution. As can be seen from the development here, this assumption is not necessary. However, with this assumption, one can show that the estimate is not only the one that minimizes

Table 9.1

Control and Estimation Duality

Control	Estimation
Φ	Φ^T
\mathbf{M}	\mathbf{P}
\mathbf{S}	\mathbf{M}
\mathbf{Q}_1	$\Gamma_1\mathbf{R}_w\Gamma_1^T$
Γ	\mathbf{H}^T
\mathbf{Q}_2	\mathbf{R}_v

⁹ In the steady state, the filter has constant coefficients and, for the assumed model, is the same as the Wiener filter.

the error squared, but also the one that is the statistically "most likely." In this derivation, the result is referred to as the **maximum likelihood** estimate.

Because Eq. (9.34) and Eq. (9.95) are the same form, the eigenvalues have the same reciprocal properties in both cases. Furthermore, for systems with a single output and a single process noise input, the symmetric root locus follows by analogy with Eq. (9.40) and the use of Table 9.1. Specifically, the characteristic equation becomes

$$1 + qG_e(z^{-1})G_e(z) = 0, \quad (9.98)$$

where $q = R_w/R_i$ and

$$G_e(z) = H(zI - \Phi)^{-1}\Gamma_i.$$

Therefore, for systems where the process noise is additive with the control input—that is, $\Gamma = \Gamma_i$ —the control and estimation optimal root loci are identical, and the control and estimation roots could be selected from the same loci. For example, the loci in Figs. 9.4 and 9.5 could be used to select estimator roots as well as the control roots.

9.4.4 Noise Matrices and Discrete Equivalents

The quantities defining the noise magnitude, the covariance matrices R_w and R_v , were defined in Section 9.4.2 as

$$R_w = \mathcal{E}\{\mathbf{w}(k)\mathbf{w}^T(k)\} \quad \text{and} \quad R_v = \mathcal{E}\{\mathbf{v}(k)\mathbf{v}^T(k)\}.$$

Typically, if there is more than one process or measurement noise component, one has no information on the cross-correlation of the noise elements and therefore R_w and R_v are selected as diagonal matrices. The magnitudes of the diagonal elements are the variances of the noise components.

Process Noise, R_w

The process noise acts on the continuous portion of the system and, assuming that it is a white continuous process, varies widely throughout one sample period. Its effect over the sample period, therefore, cannot be determined as it was for Eq. (9.90); instead, it needs to be integrated. From Eqs. (4.55) and (4.56), we see that the effect of continuous noise input over one sample period is

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \int_0^T e^{\Phi(\eta)}\mathbf{G}_i\mathbf{w}(\eta)d\eta.$$

where \mathbf{G}_i is defined by Eq. (4.45). Repeating the derivation¹⁰ of Eq. (9.90) with the integral above replacing $\Gamma_i\mathbf{w}(k)$ in Eq. (9.88), we find that Eq. (9.90) becomes

$$\mathbf{M}(k+1) = \Phi P(k)\Phi^T + \int_0^T \int_0^\tau \Phi(\eta)\mathbf{G}_i\mathcal{E}\{\mathbf{w}(\eta)\mathbf{w}^T(\tau)\}\mathbf{G}_i^T\Phi^T(\tau)d\tau d\eta. \quad (9.99)$$

But the white noise model for \mathbf{w} means that

$$\mathcal{E}\{\mathbf{w}(\eta)\mathbf{w}^T(\tau)\} = R_{wpd}\delta(\eta - \tau),$$

where R_{wpd} is called the power spectral density, or mean-square spectral density, of the continuous white noise. Therefore, Eq. (9.99) reduces to

$$\mathbf{M}(k+1) = \Phi P(k)\Phi^T + C_d,$$

where

$$C_d = \int_0^T \Phi(\tau)\mathbf{G}_i R_{wpd} \mathbf{G}_i^T \Phi^T(\tau) d\tau. \quad (9.100)$$

Calculation of this integral (see `dswm.m` in the Digital Control Toolbox) can be carried out using a similar exponential form due to Van Loan (1978), as in Eq. (9.54). If T is very short compared to the system time constants, that is

$$\Phi \cong I \quad \text{and} \quad \Gamma_i \cong G_i T,$$

then the integral is approximately

$$C_d \cong T G_i R_{wpd} G_i^T,$$

which can also be written¹¹

$$C_d \cong \Gamma_i \frac{R_{wpd}}{T} \Gamma_i^T.$$

Therefore, one can apply the discrete covariance update Eq. (9.94) to the case where \mathbf{w} is continuous noise by using the approximation that

$$R_w \cong \frac{R_{wpd}}{T}. \quad (9.101)$$

as long as T is much shorter than the system time constants. If this assumption is not valid, one must revert to the integral in Eq. (9.100).

In reality, however, there is no such thing as white continuous noise. A pure white-noise disturbance would have equal magnitude content at all frequencies from 0 to ∞ . Translated, that means that the correlation time of the random signal is precisely zero. The only requirement for our use of the white-noise

¹⁰ See Stengel (1986), p. 327.

¹¹ By computing Γ_i exactly according to Eq. (4.58), the approximation that follows for C_d is substantially more accurate than the approximation using G_i .

model in discrete systems is that the disturbance have a correlation time that is short compared to the sample period. If the correlation time is on the same order or longer than the sample period, the correct methodology entails adding the colored-noise model to the plant model and estimating the random disturbance along with the original state. In fact, if the correlation time is extremely long, the disturbance acts much like a bias, and we have already discussed its estimation in Section 8.5.2. In practice, disturbances are often assumed to be either white or a bias, because the improved accuracy possible by modeling a random disturbance with a time constant on the same order as the sample period is not deemed worth the extra complexity.

The determination of the appropriate value of \mathbf{R}_{wpd} that represents a physical process is aided by the realization that pure white noise does not exist in nature. Disturbances all have a nonzero correlation time, the only question being: How short? Assuming that the time correlation is exponential with a correlation time τ_c , and that $\tau_c \ll$ system time constants, the relation between the power spectral density and the mean square of the signal is¹²

$$R_{wpd} \cong 2\tau_c \mathcal{E}\{w^2(t)\}. \quad (9.102)$$

Typically, one can measure the mean square value, $\mathcal{E}\{w^2(t)\}$, and can either measure or estimate its correlation time, τ_c , thus allowing the computation of each diagonal element of \mathbf{R}_{wpd} . However, the desired result for our purposes is the discrete equivalent noise, \mathbf{R}_w . It can be computed from Eq. (9.101) and Eq. (9.102), where each diagonal element, $[\mathbf{R}_w]_i$, is related to the mean square and correlation time of the i th disturbance according to

$$[\mathbf{R}_w]_i = \frac{2}{T} [\tau_c \mathcal{E}\{w^2(t)\}]_i. \quad (9.103)$$

Note from Eq. (9.103) that \mathbf{R}_w and $\mathcal{E}\{w^2(t)\}$ are not the same quantities. Specifically, the diagonal elements of \mathbf{R}_w are the mean square values of the discrete noise, $w(k)$, that produces a response from the discrete model given by Eq. (9.90) that matches the response of the continuous system acted on by a $w(t)$ with $\mathcal{E}\{w^2(t)\}$ and τ_c . Note also that it has been assumed that the noise is white compared to the sample period, that is, $\tau_c \ll T$ and that $T \ll$ any system time constants. Under these conditions the discrete equivalent mean square value is less than the continuous signal mean square because the continuous random signal acting on the continuous system is averaged over the sample period. If τ_c is not $\ll T$, then the noise is not white and Eq. (9.100) is not valid; thus calculation of \mathbf{R}_w is not relevant.

12 Bryson and Ho (1975), p. 331.

Sensor Noise, \mathbf{R}_v

The pertinent information given by the manufacturer of a sensor product would be the rms "jitter" error level (or some other similar name), which can usually be interpreted as the random component and assumed to be white, that is, uncorrelated from one sample to the next. The rms value is simply squared to arrive at the diagonal elements of $\mathcal{E}\{v^2(t)\}$. Unlike the process noise case these values are used directly, that is,

$$[\mathbf{R}_v]_i = [\mathcal{E}\{v^2(t)\}]_i. \quad (9.104)$$

The assumption of no time correlation is consistent with the development of the optimal estimator that was discussed in Section 9.4.2. If the correlation time of a sensor is longer than the sample period, the assumption is not correct, and an accurate treatment of the noise requires that its "coloring" model be included with the plant model and the measurement noise error estimated along with the rest of the state.¹³ One could also ignore this complication and proceed as if the noise were white, with the knowledge that the effect of the measurement noise is in error and that skepticism is required in the interpretation of estimate error predictions based on \mathbf{P} . Furthermore, one could no longer claim that the filter was optimal.

Sensor manufacturers typically list bias errors also. This component should at least be evaluated to determine the sensitivity of the system to the bias, and if the effect is not negligible, the bias should be modeled, augmented to the state, and estimated using the ideas of Section 8.5.2.

Note that neither the sample period nor the correlation time of the sensor error has an impact on \mathbf{R}_v if v is white. Although the rms value of v is not affected by the sampling, sampling at a higher rate will cause more measurements to be averaged in arriving at the state estimate, and the estimator accuracy will improve with sample rate.

In some cases, the designer wishes to know how the sensor noise will affect an estimator that is implemented with analog electronics. Although this can be done in principle, in practice it is rarely done because of the low cost of digital implementations. The value of analyzing the continuous case is that the knowledge can be useful in selecting a sample rate. Furthermore, the designer is sometimes interested in creating a digital implementation whose roots match that of a continuous design, and finding the discrete equivalent noise is a method to approximate that design goal. Whatever the reason, the continuous filter can be evaluated digitally, with the appropriate value of \mathbf{R}_v being the one that provides the discrete equivalent to the continuous process, the same situation that was

13 See Stengel (1986) or Bryson and Ho (1975).

examined for the process noise. Therefore, the proper relation in this special case is from Eq. (9.101) or (9.103)

$$\mathbf{R}_v = \frac{\mathbf{R}_{v,pd}}{T} \quad \text{or} \quad [\mathbf{R}_v]_i = \frac{2}{T} [\tau_i \mathcal{E}\{v^2(t)\}], \quad (9.105)$$

where $\mathcal{E}\{v^2(t)\}$ is the mean square value of the sensor noise and τ_i is its correlation time. Alternatively, if one desires only the continuous filter performance for a baseline, one can use a pure continuous analysis of the filter,¹⁴ which requires only $\mathbf{R}_{v,pd}$.

9.5 Multivariable Control Design

The elements of the design process of a MIMO system have been discussed in the preceding sections. This section discusses some of the issues in design and provides two examples of the process.

9.5.1 Selection of Weighting Matrices \mathbf{Q}_1 and \mathbf{Q}_2

As can be seen from the discussion in Sections 9.2 and 9.3, the selection of \mathbf{Q}_1 and \mathbf{Q}_2 is only weakly connected to the performance specifications, and a certain amount of trial and error is usually required with an interactive computer program before a satisfactory design results. There are, however, a few guidelines that can be employed. For example, Bryson and Ho (1975) and Kwakernaak and Sivan (1972) suggest essentially the same approach. This is to take $\mathbf{Q}_1 = \mathbf{H}^T \bar{\mathbf{Q}}_1 \mathbf{H}$ so that the states enter the cost via the important outputs (which may lead to $\mathbf{H} = \mathbf{I}$ if all states are to be kept under close regulation) and to select $\bar{\mathbf{Q}}_1$ and \mathbf{Q}_2 to be diagonal with entries so selected that a fixed percentage change of each variable makes an equal contribution to the cost.¹⁵ For example, suppose we have three outputs with maximum deviations m_1 , m_2 , and m_3 . The cost for diagonal $\bar{\mathbf{Q}}_1$ is

$$\bar{q}_{11}y_1^2 + \bar{q}_{22}y_2^2 + \bar{q}_{33}y_3^2,$$

The rule is that if $y_1 = \alpha m_1$, $y_2 = \alpha m_2$, and $y_3 = \alpha m_3$, then

$$\bar{q}_{11}y_1^2 = \bar{q}_{22}y_2^2 = \bar{q}_{33}y_3^2;$$

thus

$$\bar{q}_{11}\alpha^2m_1^2 = \bar{q}_{22}\alpha^2m_2^2 = \bar{q}_{33}\alpha^2m_3^2.$$

Bryson's rules

¹⁴ See Bryson and Ho (1975).

¹⁵ Kwakernaak and Sivan (1972) suggest using a percentage change from nominal values; Bryson and Ho use percentage change from the maximum values.

A satisfactory solution for elements of \mathbf{Q}_1 is then¹⁶

$$\bar{Q}_{1,11} = 1/m_1^2, \bar{Q}_{1,22} = 1/m_2^2, \bar{Q}_{1,33} = 1/m_3^2. \quad (9.106)$$

Similarly, for \mathbf{Q}_2 we select a matrix with diagonal elements

$$Q_{2,11} = 1/u_{1,\max}^2, Q_{2,22} = 1/u_{2,\max}^2. \quad (9.107)$$

There remains a scalar ratio between the state and the control terms, which we will call ρ . Thus the total cost is

$$\mathcal{J} = \rho x^T \mathbf{H}^T \bar{\mathbf{Q}}_1 \mathbf{H} x + u^T \mathbf{Q}_2 u, \quad (9.108)$$

where \mathbf{Q}_1 and \mathbf{Q}_2 are given by Eq. (9.106), and Eq. (9.107), and ρ is to be selected by trial and error. A computer-interactive procedure that allows examination of root locations and transient response for selected values of $\bar{\mathbf{Q}}_1$ and \mathbf{Q}_2 expedites this process considerably.

9.5.2 Pincer Procedure

The designer can introduce another degree of freedom into this problem by requiring that all the closed-loop poles be inside a circle of radius $1/\alpha$, where $\alpha \geq 1$. If we do this, then the magnitude of every transient in the closed loop will decay at least as fast as $1/\alpha^k$, which forms pincers around the transients and allows a degree of direct control over the settling time. We can introduce this effect in the following way.

Suppose that as a modification to the performance criterion of Eq. (9.11), we consider

$$\mathcal{J}_o = \sum_{k=0}^{\infty} [x^T \mathbf{Q}_1 x + u^T \mathbf{Q}_2 u] \alpha^{2k}. \quad (9.109)$$

We can distribute the scalar term α^{2k} in Eq. (9.109) as $\alpha^k \alpha^k$ and write it as

$$\begin{aligned} \mathcal{J}_o &= \sum_{k=0}^{\infty} [(\alpha^k x)^T \mathbf{Q}_1 (\alpha^k x) + (\alpha^k u)^T \mathbf{Q}_2 (\alpha^k u)] \\ &= \sum_{k=0}^{\infty} [z^T \mathbf{Q}_1 z + v^T \mathbf{Q}_2 v]. \end{aligned} \quad (9.110)$$

where

$$z(k) = \alpha^k x(k), \quad v(k) = \alpha^k u(k). \quad (9.111)$$

The equations in z and v are readily found. Consider

$$z(k+1) = \alpha^{k+1} x(k+1),$$

¹⁶ Sometimes called Bryson's rules.

From Eq. (9.10) we have the state equations for $\mathbf{x}(k+1)$, so that

$$\mathbf{z}(k+1) = \alpha^{k-1}[\Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k)].$$

If we multiply through by the α^{k-1} -term, we can write this as

$$\mathbf{z}(k+1) = \alpha\Phi\mathbf{z}(k) + \alpha\Gamma\mathbf{u}(k).$$

but from the definitions in Eq. (9.111), this is the same as

$$\mathbf{z}(k+1) = \alpha\Phi\mathbf{z}(k) + \alpha\Gamma\mathbf{v}(k). \quad (9.112)$$

The performance function Eq. (9.110) and the equations of motion Eq. (9.112) define a new problem in optimal control for which the solution is a control law

$$\mathbf{v} = -\mathbf{K}\mathbf{z},$$

which, if we work backward, is

$$\alpha^k\mathbf{u}(k) = -\mathbf{K}(\alpha^k\mathbf{x}(k))$$

or

$$\mathbf{u}(k) = -\mathbf{Kx}(k). \quad (9.113)$$

We conclude from all this that if we use the control law Eq. (9.113) in the state equations (9.10), then a trajectory results that is optimal for the performance J_α given by Eq. (9.109). Furthermore, the state trajectory satisfies Eq. (9.111), where $\mathbf{z}(k)$ is a stable vector so that $\mathbf{x}(k)$ must decay at least as fast as $1/\alpha^k$, or else $\mathbf{z}(k)$ could not be guaranteed to be stable.

To apply the pincers we need to relate the settling time to the value of α . Suppose we define settling time of x_i as that time t_s such that if $x_i(0) = 1$ and all other states are zero at $k = 0$, then the transients in x_i are less than 0.01 (1% of the maximum) for all times greater than t_s . If we approximate the transient in x_i as

$$x_i(k) \approx x_i(0)(1/\alpha)^k,$$

then when $kT = t_s$, we must have

$$x_i(kT) \leq 0.01x_i(0),$$

which will be satisfied if α is such that

$$(1/\alpha)^k \leq 0.01 = \frac{1}{100},$$

or

$$\alpha > 100^{1/k} = 100^{T/t_s}. \quad (9.114)$$

In summary, application of the pincer procedure requires that the designer select the settling time, t_s , within which all states should settle to less than 1%. Equation (9.114) is then used to compute α and, according to Eq. (9.112), the

revised system $\alpha\Phi$ and $\alpha\Gamma$ for use in an LQR computation for the feedback gain matrix \mathbf{K} . Use of \mathbf{K} with the original system, Φ , will produce a response of all states that settle within the prescribed t_s .

9.5.3 Paper-Machine Design Example

As an illustration of a multivariable control using optimal control techniques, we will consider control of the paper-machine head box described in Appendix A.5. The continuous equations of motion are given by

$$\dot{\mathbf{x}} = \begin{bmatrix} -0.2 & 0.1 & 1 \\ -0.05 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 1 \\ 0 & 0.7 \\ 1 & 0 \end{bmatrix} \mathbf{u}. \quad (9.115)$$

We assume the designer has the following specifications on the responses of this system:

1. The maximum sampling frequency is 5 Hz ($T = 0.2$ sec).
2. The 1% settling time to demands on x_1 (= total head) should be less than 2.4 sec (12 periods).
3. The settling time to demands on x_2 (= liquid level) should be less than 8 sec (40 periods).
4. The units on the states have been selected so that the maximum permissible deviation on total head, x_1 , is 2.0 units and the liquid level, x_2 , is 1.0.
5. The units on control have been selected so that the maximum permissible deviation on u_1 (air control) is 5 units and that of u_2 (stock control) is 10 units.

First let us apply the pincer procedure to ensure that the settling times are met. We have asked that the settling times be 2.4 sec for x_1 and 8 sec for x_2 . If, for purposes of illustration, we select the more stringent of these and in Eq. (9.114) set $t_s = 2.4$ for which $t_s/T = 12$, then

$$\alpha > 100^{1/12} = 1.47.$$

Now let us select the cost matrices. Based on the specifications and the discussion in Section 9.5.1, we can conclude that $m_1 = 2$ and $m_2 = 1$, and thus

$$\bar{\mathbf{Q}}_1 = \begin{bmatrix} 0.25 & 0 \\ 0 & 1 \end{bmatrix}.$$

Because we are interested only in x_1 and x_2 , the output matrix \mathbf{H} in Eq. (9.108) is

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

and

$$\mathbf{Q}_1 = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Furthermore, because $u_{1\max} = 5$ and $u_{2\max} = 10$

$$\mathbf{Q}_2 = \begin{bmatrix} 0.04 & 0 \\ 0 & 0.01 \end{bmatrix}.$$

Conversion of the continuous system in Eq. (9.115) to a discrete one ($T = 0.2$ sec) yields

$$\Phi = \begin{bmatrix} 0.9607 & 0.0196 & 0.1776 \\ -0.0098 & 0.9999 & -0.0009 \\ 0 & 0 & 0.8187 \end{bmatrix} \quad \text{and} \quad \Gamma = \begin{bmatrix} 0.0185 & 0.1974 \\ -0.0001 & 0.1390 \\ 0.1813 & 0 \end{bmatrix}.$$

These two matrices are then multiplied by the scalar 1.47 ($= \alpha$) and used in MATLAB's dlqr.m with the preceding \mathbf{Q}_1 and \mathbf{Q}_2 .

The LQR calculation gives a control gain

$$\mathbf{K} = \begin{bmatrix} 6.81 & -9.79 & 3.79 \\ 0.95 & 4.94 & 0.10 \end{bmatrix},$$

and closed-loop poles at

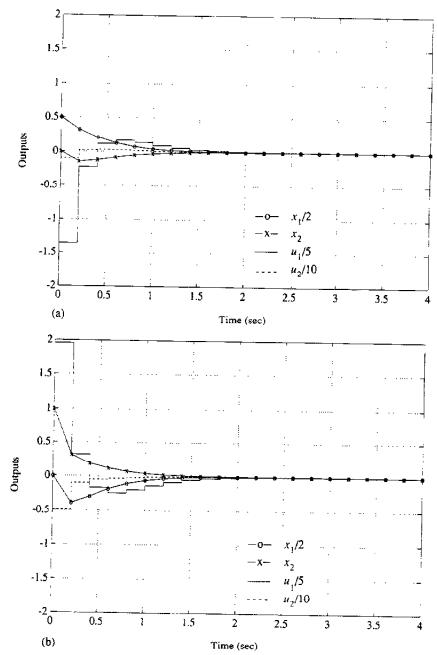
$$z = 0.108, 0.491 \pm j0.068,$$

which are well within $1/1.47 = 0.68$. Fig. 9.9(a) shows the transient response to unit-state initial conditions on x_1 , and Fig. 9.9(b) shows the same for initial conditions on x_2 . Examination of these results shows that the requirement on settling time has been substantially exceeded because x_1 settles to within 1% by 1.8 sec, and x_2 settles within 1% by 1.6 sec for both cases. However, the control effort on u_1 is larger than its specification for both initial condition cases, and further iteration on the design is required. To correct this situation, the designer should be led by the fast response to relax the demands on response time in order to lower the overall need for control. Figure 9.10 shows the response with t_f in Eq. (9.114) selected to be 5 sec for the case with $\mathbf{x}(0) = [0|10]^T$. The control u_1 is just within the specification, and the response time of x_1 is 2.3 sec and that of x_2 is 2.0 sec. All specifications are now met.

It is possible to improve the design still further by noting that the response of x_2 beats its specified maximum value and settling time by a substantial margin. To capitalize on this observation, let us try relaxing the cost on x_2 . After some iteration (see Problem 9.7), we find that no cost on x_2 , that is

$$\mathbf{Q}_1 = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Figure 9.9
Response of paper-machine closed-loop control to:
(a) initial state
 $\mathbf{x}'(0) = [1 \ 0 \ 0]$;
(b) initial state
 $\mathbf{x}''(0) = [0 \ 1 \ 0]$



results in a design that still meets all specifications and substantially reduces the use of both controls. Its response to $\mathbf{x}(0) = [0|10]^T$ is shown in Figure 9.11.

Figure 9.10
Response of
paper-machine
closed-loop control to
initial state
 $x^*(0) = [0 \ 1 \ 0]$ with
 t_s lengthened to 5 sec

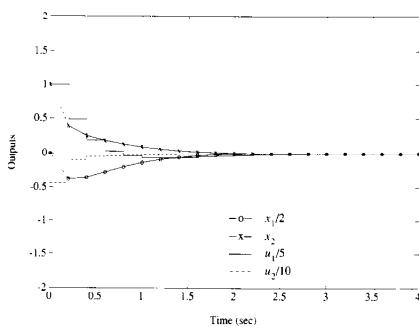
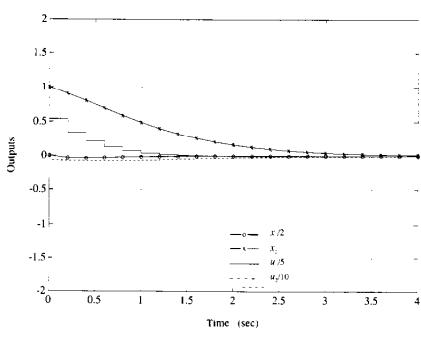


Figure 9.11
Response of
paper-machine
closed-loop control to
initial state
 $x^*(0) = [0 \ 1 \ 0]$ with
 t_s lengthened to 5 sec
and no weight on x_2



9.5.4 Magnetic-Tape-Drive Design Example

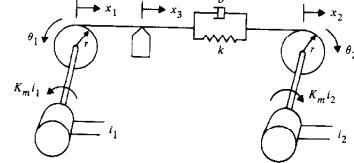
As a further illustration of MIMO design, we will now apply the ideas to an example that includes integral control and estimation. The system is shown in Fig. 9.12. There is an independently controllable drive motor on each end of the tape; therefore, it is possible to control the tape position over the read head, x_3 , as well as the tension in the tape. We have modeled the tape to be a linear spring with a small amount of viscous damping. Although the figure shows the tape head to the left of the spring, in fact, the springiness is distributed along the full length of the tape as shown by the equations below. The goal of the control system is to enable commanding the tape to specific positions over the read head while maintaining a specified tension in the tape at all times. We will carry out the design using MIMO techniques applied to the full system equations and conclude the discussion by illustrating how one could perform the design using a decoupled model and SISO techniques.

The specifications are that we wish to provide a step change in position of the tape head, x_3 , of 1 mm with a 1% settling time of 250 msec with overshoot less than 20%. Initial and final velocity of x_3 is to be zero. The tape tension, T_r , should be controlled to 2 N with the constraint that $0 < T_r < 4$ N. The current is limited to 1 A at each drive motor.

The equations of motion of the system are

$$\begin{aligned} J\ddot{\theta}_1 &= -T_r r + K_m i_1, \\ J\ddot{\theta}_2 &= -T_r r + K_m i_2, \\ T_r &= k(x_3 - x_1) + b(\dot{x}_3 - \dot{x}_1), \\ x_3 &= (x_1 + x_2)/2. \end{aligned} \quad (9.116)$$

Figure 9.12
Magnetic-tape-drive
design example



where

- i_1, i_2 = current into drive motors 1 and 2, respectively (A).
- T_e = tension in tape (N).
- θ_1, θ_2 = angular position of motor/capstan assembly (radians).
- x_1, x_2 = position of tape at capstan (mm).
- x_3 = position of tape over read head (mm).
- J = $0.006375 \text{ kg} \cdot \text{m}^2$, motor and capstan inertia.
- r = 0.1 m , capstan radius.
- K_m = $0.544 \text{ N} \cdot \text{m/A}$, motor torque constant.
- k = 2113 N/m , tape spring constant, and
- b = $3.75 \text{ N} \cdot \text{sec/m}$, tape damping constant.

In order to be able to simulate a control system design on inexpensive equipment where the system has time constants much faster than 1 sec, it is often useful to time-scale the equations so that the simulation runs slower in the laboratory than it would on the actual system.¹⁷ We have chosen to time-scale the equations above so they run a factor of 10 slower than the actual system. Therefore, the settling time specifications become 2.5 sec instead of 250 msec for the actual system. The numerical equations below and all the following discussion pertain to the time-scaled system. Incorporating the parameter values and writing the time-scaled equations in state form results in

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\omega}_1 \\ \dot{\omega}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 10 \\ 3.315 & -3.315 & -0.5882 & -0.5882 \\ 3.315 & -3.315 & -0.5882 & -0.5882 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \omega_1 \\ \omega_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 8.533 & 0 \\ 0 & 8.533 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix}, \quad (9.117)$$

where the desired outputs are

$$\begin{bmatrix} x_3 \\ T_e \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ -2.113 & 2.113 & 0.375 & 0.375 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \omega_1 \\ \omega_2 \end{bmatrix}. \quad (9.118)$$

where

ω_1, ω_2 = angular rates of motor/capstan assembly, $\dot{\theta}_i$ (rad/sec).

The eigenvalues of the system matrix in Eq. (9.117) are

$$s \cong 0, 0, \pm j8 \text{ rad/sec.}$$

¹⁷ See Franklin, Powell, and Emami-Naeini (1986) for a discussion of time-scaling.

where the oscillatory roots are from the spring-mass system consisting of the tape and the motor/capstan inertias. This open-loop resonance has a higher frequency than the required closed-loop roots to meet the settling time specifications; therefore, it would be wise to sample at $15 \times$ the open-loop resonance at 8 rad/sec. A sample rate of $T = 0.05$ sec results.

Full State Feedback

As a first step in the design, let's try state feedback of all four states in Eq. (9.117). We will address the state estimation in a few pages.

The output quantities to be controlled are x_3 and T_e so that it is logical to weight those quantities in the cost \mathbf{Q}_1 . According to the guidelines in Eq. (9.106), we should pick

$$\bar{\mathbf{Q}}_1 = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{4} \end{bmatrix},$$

and because we expect each control to be used equally

$$\mathbf{Q}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Although these weightings gave an acceptable result, a somewhat more desirable result was found by modifying these weightings

$$\bar{\mathbf{Q}}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{Q}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

where a third output has been added to that of Eq. (9.118), which represents \dot{T}_e in order to obtain better damping of the tension. Thus we obtain the weighting matrix from

$$\mathbf{Q}_1 = \mathbf{H}_v^T \bar{\mathbf{Q}}_1 \mathbf{H}_v, \quad (9.119)$$

where

$$\mathbf{H}_v = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ -2.113 & 2.113 & 0.375 & 0.375 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix}. \quad (9.120)$$

Use of the \mathbf{Q}_1 and \mathbf{Q}_2 with the discrete equivalent of Eq. (9.117) in MATLAB's dlqr.m results in the feedback gain matrix

$$\mathbf{K} = \begin{bmatrix} -0.823 & 0.286 & 1.441 & 0.311 \\ -0.286 & 0.823 & 0.311 & 1.441 \end{bmatrix}.$$

which can be used to form the closed-loop system for evaluation. Calculation of the eigenvalues of the discrete system and transformation of them to their discrete equivalent by $z = e^{sT}$ results in closed-loop roots at

$$s = -5.5 \pm j5.5, -10.4 \pm j12.4 \text{ sec}^{-1}.$$

To bring the system to the desired values of I_x and x_1 , we use the state command structure described in Section 8.4.2. Let us first try without the feedforward value of steady-state control, u_{ss} . Calculation of the reference state, \mathbf{x}_r , can be carried out from the physical relationships that must exist at the desired values of $x_2 = 1 \text{ mm}$ and $T_x = 2 \text{ N}$

$$\begin{aligned} x_1 + x_2 &= 2x_2 = 2, \\ x_2 - x_1 &= \frac{T_x}{k} = \frac{2}{2113} = 0.000947 \text{ m} = 0.947 \text{ mm}. \end{aligned}$$

Solving these two equations and adding the zero velocities results in

$$\mathbf{x}_r = \begin{bmatrix} 0.527 \\ 1.473 \\ 0 \\ 0 \end{bmatrix}.$$

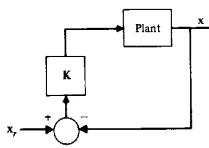
Evaluation of the time response using the structure in Fig. 9.13 (from Fig. 8.14) shows that the input current limits are violated and results in a settling time of 1 sec, which is significantly faster than required. In order to slow down the response and thus reduce the control usage, we should lower $\bar{\mathbf{Q}}_1$. Revising the weightings to

$$\bar{\mathbf{Q}}_1 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \quad \text{and} \quad \bar{\mathbf{Q}}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (9.121)$$

results in

$$\mathbf{K} = \begin{bmatrix} -0.210 & 0.018 & 0.744 & 0.074 \\ -0.018 & 0.210 & 0.074 & 0.744 \end{bmatrix}.$$

Figure 9.13
Reference input structure used for Fig. 9.14



which produces a closed-loop system whose roots transform to

$$s = -3.1 \pm j3.1, -4.5 \pm j9.2 \text{ sec}^{-1},$$

and produces the response shown in Fig. 9.14. The figure shows that x_1 has a settling time of about 2 sec with an overshoot less than 20% while the control currents are within their limits; however, the tension did not go to the desired 2 N. The steady-state tension error of the system is substantial, and so we shall proceed to correct it.

As discussed in Section 8.4.2, we can provide a feedforward of the required steady-state value of the control input that eliminates steady-state errors. Evaluation of Eq. (8.73) (see ref.m in the Digital Control Toolbox) using \mathbf{H} from Eq. (9.118) and Φ, Γ from Eq. (9.117) leads to

$$\mathbf{N}_r = \begin{bmatrix} 1 & -0.2366 \\ 1 & +0.2366 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{N}_{ss} = \begin{bmatrix} 0 & 0.1839 \\ 0 & 0.1839 \end{bmatrix},$$

which can be used to command the system as in Fig. 9.15(a) with $\mathbf{r} = [1 \text{ mm} \ 2 \text{ N}]^T$. Note that $\mathbf{N}_r \mathbf{r} = \mathbf{x}_r = [0.527 \ 1.473 \ 0 \ 0]^T$ as already computed above, and we now have the steady state control, $\mathbf{u}_{ss} = \mathbf{N}_{ss} \mathbf{r} = [0.368 \ 0.368]^T$, that should eliminate the tension error. The response of the system with this modification is shown in Fig. 9.15. It verifies the elimination of the tension error and that all other specifications are met.

Figure 9.14
Time response using Eq. (9.121)

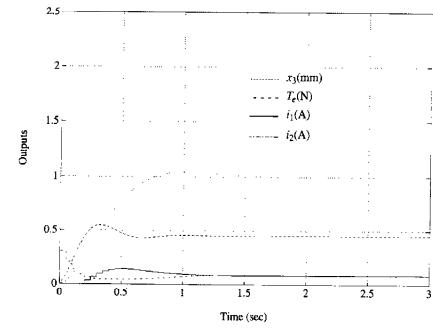
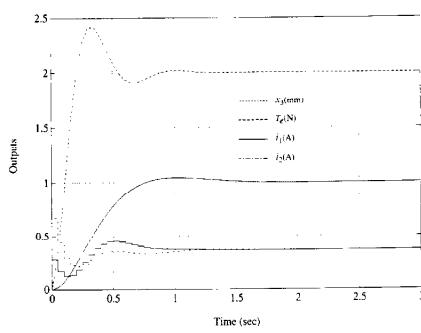


Figure 9.15
Time response using Eq. (9.121) and feedforward of u_{ss}



Normally, designers prefer to avoid the use of feedforward of u_{ss} alone because its value depends on accurate knowledge of the plant gain. The preferred method of reducing steady-state errors is to use integral control or possibly to combine integral control with feedforward. In the latter case, the integral's function is simply to provide a correction for a possible error in the feedforward and thus can be designed to have less of an impact on the system bandwidth. So let's design an integral control to replace the u_{ss} feedforward.

Because the tension was the only quantity with the error, the integral control need only correct that. Therefore, we will integrate a measurement of the tension and use it to augment the feedback. We start out by augmenting the state with the integral of T_d according to Eq. (8.83) so that

$$\begin{bmatrix} x_I \\ x \end{bmatrix}_{k+1} = \begin{bmatrix} I & H_u \\ 0 & \Phi \end{bmatrix} \begin{bmatrix} x_I \\ x \end{bmatrix}_k + \begin{bmatrix} 0 \\ \Gamma \end{bmatrix} u(k), \quad (9.122)$$

where, from Eq. (9.118)

$$H_u = [-2.113 \ 2.113 \ 0.375 \ 0.375].$$

We will use the augmented system matrices (Φ_u, Γ_u) from Eq. (9.122) in an LQR computation with a revised Q_1 that also weights the integral state variable. This is most easily accomplished by forming a revised H_e from Eq. (9.120) that includes

a fourth output consisting of the tension integral, x_I , and that is used with the augmented state defined by Eq. (9.122). The result is

$$H_e = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & -2.113 & 2.113 & 0.375 & 0.375 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (9.123)$$

The revised weighting matrices are then obtained by the use of Eq. (9.119) with an additional diagonal term in Q_1 corresponding to an equal weight on x_I , that is

$$\bar{Q}_1 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \quad \text{and} \quad Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (9.124)$$

Performing the calculations results in an augmented feedback gain matrix

$$K_f = \begin{bmatrix} 0.137 & -0.926 & 0.734 & 1.26 & 0.585 \\ 0.137 & -0.734 & 0.926 & 0.585 & 1.26 \end{bmatrix}, \quad (9.125)$$

which can be partitioned into two gains, one for the integral and one for the original state

$$K_I = \begin{bmatrix} 0.137 \\ 0.137 \end{bmatrix} \quad \text{and} \quad K = \begin{bmatrix} -0.926 & 0.734 & 1.26 & 0.585 \\ -0.734 & 0.926 & 0.585 & 1.26 \end{bmatrix}. \quad (9.126)$$

These feedback gain matrices are then used to control the system as shown in Fig. 9.16, which is based on the ideas from Fig. 8.23. Essentially, the only difference between this implementation and the previous one is that we have replaced the u_{ss} feedforward with an integral of the tension error.

The roots of the closed-loop system transformed to the s-plane are

$$s = -9.3, -3.1 \pm j3.1, -5.8 \pm j11.8 \text{ sec}^{-1},$$

and the time response of the system is shown in Fig. 9.17. Again, the tension goes to the desired 2 N, as it should, and all other specifications are met.

Add the Estimator

The only unfinished business now is to replace the full state feedback with an estimated state. We will assume there are two measurements, the tension, T_d , and tape position, x_1 . Although measurements of θ_a and θ_c would possibly be easier to make, it is typically best to measure directly the quantities that one is interested in controlling. Therefore, for purposes of estimation, our measurement matrix is

$$H_e = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ -2.113 & 2.113 & 0.375 & 0.375 \end{bmatrix}. \quad (9.127)$$

Figure 9.16
Integral control implementation with reference input and full state feedback

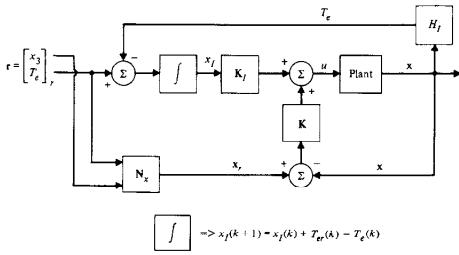
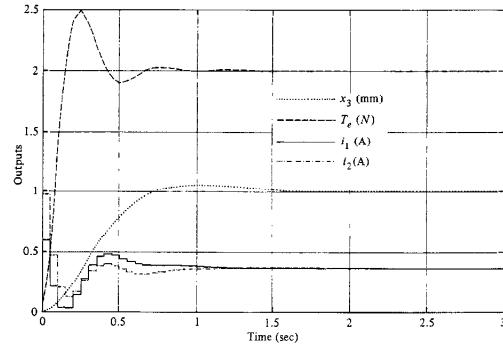


Figure 9.17
Time response using Eq. (9.126) and integral control as in Fig. 9.16



Let us suppose that the rms accuracy of the tape position measurement is 0.02 mm. According to Eq. (9.104), the first diagonal element of the \mathbf{R}_v matrix is therefore $0.02^2 = 0.0004$. Let us also suppose that the rms accuracy of the tension measurement is 0.01 N. This results in

$$\mathbf{R}_v = \begin{bmatrix} 0.0004 & 0 \\ 0 & 0.0001 \end{bmatrix}. \quad (9.128)$$

Determining values for the process-noise matrix, \mathbf{R}_w , is typically not based on the same degree of certainty as that possible for \mathbf{R}_v . We will assume that the process noise enters the system identically to i_1 and i_2 ; therefore, $\Gamma_w = \Gamma$ in Eq. (9.81). We could make measurements of the magnitude and spectrum of the input-current noise to determine \mathbf{R}_w via Eqs. (9.101) and (9.102). However, other disturbances and modeling errors typically also affect the system and would need to be quantified in order to arrive at a truly optimal estimator. Instead, we will somewhat arbitrarily pick

$$\mathbf{R}_w = \begin{bmatrix} 0.0001 & 0 \\ 0 & 0.0001 \end{bmatrix}. \quad (9.129)$$

compute the estimator gains and resulting roots, then modify \mathbf{R}_v based on the estimator performance in simulations including measurement noise and plausible disturbances. Proceeding, we use the unaugmented Φ and Γ , based on Eq. (9.117) in MATLAB's kalman.m with \mathbf{H}_f from Eq. (9.127) and the noise matrices from Eqs. (9.128) and (9.129) to find that

$$\mathbf{L} = \begin{bmatrix} 0.321 & -0.120 \\ 0.321 & 0.120 \\ -0.124 & 0.142 \\ 0.124 & 0.142 \end{bmatrix}, \quad (9.130)$$

which results in estimator error roots that transform to

$$s = -3.0 \pm j4.5, -4.1 \pm j15.4 \text{ sec}^{-1}. \quad (9.131)$$

If implemented using the state command structure as in Fig. 9.18, which has the same reference input structure as in Fig. 9.16, the response will be identical to Fig. 9.17. This follows because the estimator model receives the same control input that the plant does and thus no estimator error is excited. In order to evaluate the estimator, we need to add measurement noise and/or some plant disturbance that is not seen by the estimator. Figure 9.19 shows the system response to the same reference input as Fig. 9.17; but zero mean random noise with an rms of 0.02 mm has been added to the x_3 measurement, noise with an rms of 0.01 N has been added to the T_e measurement, and a step disturbance of 0.01 A has been added to the i_1 entering the plant¹⁸ at 1.5 sec. Note in the figure that there is

¹⁸ Only the plant, not the estimator.

Figure 9.18
Integral control implementation with reference input and estimated state feedback

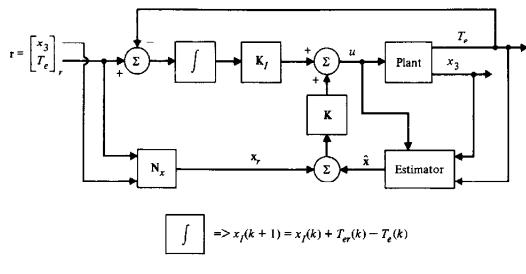
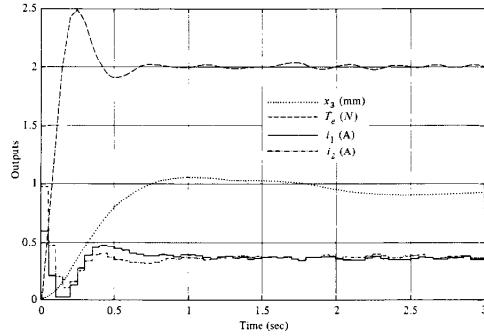


Figure 9.19
Time response with integral control and an estimator with L from Eq. (9.130) including measurement noise and a disturbance at 1.5 sec



some noise on T_e and the input currents, and that the disturbance on i_1 caused a steady-state error in x_3 .

This result can be modified by changing the estimator. Let us suppose that we wish to reduce the steady-state error on x_3 and we can tolerate increased noise effects.¹⁹ So let us increase the assumption for the process noise Eq. (9.129) to

$$\mathbf{R}_v = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}.$$

By increasing the process noise, we are essentially saying that the knowledge of the plant model is less precise; therefore, the optimal filter will pay more attention to the measurements by increasing the estimator gains and speeding up the estimator. Higher estimator gains should reduce the steady-state estimation errors and their effect on x_3 , but at the cost of increased sensitivity to measurement noise. Carrying out the calculations yields

$$\mathbf{L} = \begin{bmatrix} 0.497 & -0.143 \\ 0.497 & 0.143 \\ -0.338 & 0.329 \\ 0.338 & 0.329 \end{bmatrix}, \quad (9.132)$$

which is larger in every term than the previous gain in Eq. (9.130). Using the gain to form the estimator error equation and transforming its roots to the s-plane results in

$$s = -4.0 \pm j8.2, -2.1 \pm j20.6 \text{ sec}^{-1}.$$

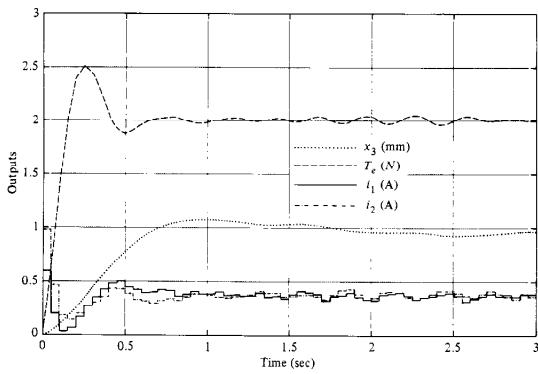
which are faster than the previous roots in Eq. (9.131). Evaluation of the same case as in Fig. 9.19 with everything identical except for the estimator gains results in Fig. 9.20. Note that the actual magnitude of the measurement noise and disturbance was not changed in generating the figure. The value of \mathbf{R}_v was changed in order to produce higher values in the gain matrix. The figure shows the expected result: larger sensitivity to the measurement noise but reduced sensitivity to the disturbance.

Decoupled Design

As a final comment, we note that the plant can be uncoupled and designed as two independent second-order systems, one for x_3 and one for T_e . The key idea to accomplish the uncoupling is to note that equal commands on the two current inputs will cause an increase in tension with no change in x_3 , whereas equal and

¹⁹ We could also add a second integral control loop on e_1 to kill the error, but it is preferable to reduce errors with high control and estimator gains in order to have a robust system for all kinds of errors, if the gains can be tolerated.

Figure 9.20
Time response with integral control and an estimator with L from Eq. (9.132) including measurement noise and a disturbance at 1.5 sec



opposite current commands will cause a change in x_3 with no effect on tension. We therefore transform the control accordingly

$$\mathbf{u}_d = \begin{bmatrix} u_3 \\ u_i \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} \Rightarrow \mathbf{T}_u = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}. \quad (9.133)$$

We also define a new state

$$\mathbf{x}_d = \begin{bmatrix} x_3 \\ \dot{x}_3 \\ T_c \\ \dot{T}_c \end{bmatrix} = \begin{bmatrix} \mathbf{H}_3 \\ \mathbf{H}_3 \mathbf{F} \\ \mathbf{H}_i \\ \mathbf{H}_i \mathbf{F} \end{bmatrix} \mathbf{x} - \mathbf{T}_d \mathbf{x}, \quad (9.134)$$

where \mathbf{H}_3 and \mathbf{H}_i are the partitions of \mathbf{H}_e in Eq. (9.127) and \mathbf{F} is the system matrix from Eq. (9.117). Following the state transformation ideas in Section 4.3.3, we can write that

$$\dot{\mathbf{x}}_d = \mathbf{F}_d \mathbf{x}_d + \mathbf{G}_d \mathbf{u}_d, \quad (9.135)$$

where

$$\mathbf{F}_d = \mathbf{T}_d \mathbf{F} \mathbf{T}_d^{-1} \quad \text{and} \quad \mathbf{G}_d = \mathbf{T}_d \mathbf{G} \mathbf{T}_u^{-1}. \quad (9.136)$$

Performing these calculations yields

$$\mathbf{F}_d = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -66.3 & -1.18 \end{bmatrix} \quad \text{and} \quad \mathbf{G}_d = \begin{bmatrix} 0 & 0 \\ 42.7 & 0 \\ 0 & 3.2 \\ 0 & 176.5 \end{bmatrix}. \quad (9.137)$$

Therefore, we see that there is a decoupling that results in two separate systems,

$$\begin{bmatrix} \dot{x}_3 \\ \ddot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_3 \\ \dot{x}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 42.7 \end{bmatrix} u_3, \quad (9.138)$$

and

$$\begin{bmatrix} \dot{T}_c \\ \ddot{T}_c \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -66.3 & -1.18 \end{bmatrix} \begin{bmatrix} T_c \\ \dot{T}_c \end{bmatrix} + \begin{bmatrix} 3.2 \\ 176.5 \end{bmatrix} u_i. \quad (9.139)$$

The two system equations, (9.138) and (9.139), can then be used to design two control systems using SISO methods. The resulting controls, u_3 and u_i , can then be "unscrambled" via \mathbf{T}_u to arrive at the desired i_1 and i_2 , that is,

$$\begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \mathbf{T}_u^{-1} \begin{bmatrix} u_3 \\ u_i \end{bmatrix}.$$

The perfect uncoupling was due to the perfect symmetry of the problem: motors and capstan inertias identical, motor constants identical, and tape read head exactly in the middle of the capstans. In an actual system, there would be differences and some coupling would be present. Due to the simplicity of the uncoupled design and its implementation, however, it can be worthwhile to evaluate the performance of a decoupled design to see if its errors warranted the full MIMO design.

9.6 Summary

- An optimal controller is one that minimizes a quadratic cost function where weighting matrices are selected by the designer to achieve a "good" design. Criteria for evaluation of the design are based on all the same factors that have been discussed throughout the book.
- Optimal control methods are useful in designing SISO systems because they help the designer place the pole locations.
- Optimal control methods are essential in designing MIMO systems as they reduce the degrees of freedom in the design iteration down to a manageable few.

- Generally, for linear, stationary systems, the steady-state optimal control solution is satisfactory and far more practical than the true optimal time-varying gain solution.
- The most accurate discrete emulation method is based on finding the discrete equivalent of a continuous cost function and solving for the discrete optimal control system with that cost.
- An optimal estimator is based on achieving the minimum mean square estimation error for a given level of process and measurement noise acting on the system.
- Generally, for linear, stationary systems, the steady-state optimal estimator is satisfactory and far more practical than the true optimal time-varying gain solution (Kalman filter).
- Optimal estimators are useful for SISO system design and essential for MIMO system design.
- The optimization procedures provide a technique for which many good design candidates can be generated which can then be evaluated to ascertain whether they meet all the design goals. The procedures do not eliminate trial-and-error methods; rather, they transfer iteration on classical compensation parameters to iteration on optimal cost-function parameters or assumed system noise characteristics.

9.7 Problems

9.1 Optimal control derivation:

- Derive Eq. (9.33) and hence verify the control Hamiltonian as given by Eq. (9.34).
 - Demonstrate that if \mathbf{W} is a transformation that brings \mathcal{H}_c to diagonal form, then the first n columns of \mathbf{W} are eigenvectors of \mathcal{H}_c associated with stable eigenvalues.
 - Demonstrate that if the optimal steady-state control, \mathbf{K}_c , given by Eq. (9.47) is used in the plant equation Eq. (9.10), then the matrix in the upper left corner of \mathbf{W} , which is \mathbf{X}_c , has columns that are eigenvectors of the closed-loop system.
- 9.2 Symmetric root locus:**
- Compute the closed-loop pole locations of the satellite design problem, using the steady-state gains given by Fig. 9.3.
 - Take the discrete transfer function from u to θ for the satellite design problem (Appendix A) and form the symmetric root locus. Plot the locus and locate the values of ρ corresponding to the selections of Q_1 given in Eq. (9.30).
 - Show that for a system with one control it is always possible to construct a symmetric root locus corresponding to the optimal steady-state control. [Hint: Show that if $Q_1 = \mathbf{H}\mathbf{H}^T$ and $\mathbf{G}(z)$ is a column-matrix transfer function given by $\mathbf{H}(\mathbf{I} - \Phi)^{-1}\Gamma$, then the roots of the optimal control are given by $1 + \rho\mathbf{G}'(z^{-1})\mathbf{G}(z) = 0$, which is a scalar symmetric root-locus problem and which can be put in the form $1 + \rho G_1(z^{-1})G_1(z) = 0$. Use Eq. (9.38).]

- (d) Give the equivalent scalar plant transfer function $G_1(z)$, if we have the satellite control problem and use $Q_1 = \mathbf{I}$, the 2×2 identity. Draw the symmetric root locus for this case.

- 9.3** Compute the location of the closed-loop poles of the optimal filter for the satellite problem for each of the values of R_{pd} given by the steady final values of the curves of Fig. 9.8.

- 9.4** For the antenna example in Appendix A.2,

- Design an optimal controller corresponding to the cost function $\mathcal{J}(y, u) = u^2 + \rho v^2$ for $\rho = 0.01, 0.1$, and 1 . Use $T = 0.2$ sec. Plot the step response of the closed loop system for initial errors in y and v in each case. Which value of ρ most nearly meets the step-response specification given in Chapter 7?
- Draw the symmetric root locus corresponding to the design of part (a).
- Design an optimal steady-state filter for the antenna. Assume that the receiver noise has a variance $R_v = 10^{-6}$ rad 2 and that the wind gust noise, w_{pd} , is white, and we want three cases corresponding to $R_{v,pd} = 10^{-2}, 10^{-4}$, and 10^{-6} .
- Plot the step response of the complete system with control law corresponding to $\rho = 0.1$ and filter corresponding to $R_{v,pd} = 10^{-4}$.

- 9.5** The lateral equations of motion for the B-767 in cruise are approximately

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\phi}_p \\ \dot{p} \end{bmatrix} = \begin{bmatrix} -0.0868 & -1 & -0.0391 & 0 \\ 2.14 & -0.228 & 0 & -0.0204 \\ 0 & 0 & 0 & 1 \\ -4.41 & 0.334 & 0 & -1.181 \end{bmatrix} \begin{bmatrix} v \\ r \\ \phi_p \\ p \end{bmatrix} + \begin{bmatrix} 0.0222 & 0 \\ -1.165 & -0.065 \\ 0 & 0 \\ 0.549 & -2.11 \end{bmatrix} \begin{bmatrix} \delta_r \\ \delta_{\phi_p} \end{bmatrix}$$

- Design two second-order controllers by ignoring the cross-coupling terms. Pick roots at $s = -1 \pm j1.5$ rad/sec for the yaw mode (v and r) and $s = -2, -0.1$ for the roll mode (ϕ_p and p). Use a sample period of $T = 0.05$ sec.
- Determine the root locations that result when the cross-coupling terms are included (Use eig.m in MATLAB.)
- Assuming one can measure r and ϕ_p , design two second-order estimators by ignoring the cross-coupling terms. Place all poles at $s = -2$ rad/sec.
- Determine the root locations that result from (c) when the cross-coupling terms are included.

- 9.6** The equations of motion for a stick balancer (Fig. 9.2) are given by

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \\ \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 21 & 0 & 0 & 0.8 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -0.4 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \\ x \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -2 \end{bmatrix} u.$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \\ x \\ v \end{bmatrix}.$$

- Design two second-order estimators, one for the stick ($\theta, \dot{\theta}$) and one for the cart (x, \dot{x}). Verify that the one-way coupling (cart to stick only) causes the full fourth-order estimator to have the same roots as the separate second-order designs.
- 9.7** Starting with the paper-machine design in Section 9.5.3 whose output is shown in Fig. 9.11, perform design iterations on Q_1, Q_2 , and γ to find the lowest possible value of u_i that still meets the state specifications on maximum value and settling time. [Hint: The routines FIG99.M, FIG910.M and FIG911.M from the Digital Control Toolbox will be useful to you.]
- 9.8** For the double mass-spring system in Example 9.5, investigate the effect of different Q 's on the closed-loop poles of the system. In particular, determine a Q that will yield an equivalent damping of the oscillatory poles that is greater than $\zeta = 0.5$. [Hint: FIG95.M from the Digital Control Toolbox will be useful to you.]
- 9.9** For the double mass-spring system in Example 9.5, design a controller and estimator that uses a measurement of d . It should respond to a command input for d with a rise time of 2 sec with an overshoot less than 20%.
- 9.10** For the double mass-spring system in Example 9.5, design a controller and estimator that uses a measurement of d . It should respond to a command input for d with a rise time of 3 sec with an overshoot less than 15%. Plot the frequency response of your compensation (control plus estimator) and qualitatively describe the features of the compensation.
- 9.11** A simplified model of a disk drive is

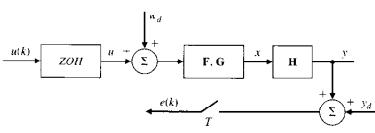
$$G(s) = \frac{10^6 \omega_r^2}{s^2(s^2 + 2\zeta\omega_r s + \omega_r^2)}$$

where $\omega_r = 19,000$ rad/sec (≈ 3 kHz) and $\zeta = 0.05$. This is the transfer function between the torque applied to the disk head arm and the motion of the read head on the end of the arm. Note this is also the same generic model as the double mass-spring system in Eq. (A.20). The system has the capability to measure where the heads are with respect to the data tracks on the disk. The two biggest disturbances are due to an actuator bias (u_d) and track wobble (y_d) from an off center disk mounting. A schematic diagram of the system to be controlled is shown in Fig. 9.21.

Since the disk is rotating at 3822 rpm (400 rad/sec), an offset causes a sinusoidal y_d to be added to the output of the transfer function above. The purpose of the servo is to follow the data tracks as closely as possible, whether the disk is slightly off center or not because that allows higher data densities on the disk. It is also desirable for the servo to settle on a new track as quickly as possible because the disk access time is directly related to this settling time and is a major selling point for disk drives.

Design a discrete compensator with a sample rate no faster than 8 kHz that has an overshoot to a step command no greater than 20%, does not allow steady state tracking

Figure 9.21
Disk drive closed-loop system for Problems 9.11 and 9.12



errors due to the actuator bias, and has a settling time (to within 5% of the final value) to a step input less than 5 msec. The system should be robust with respect to variations in the resonance frequency as this can change by $\pm 2\%$ due to temperature changes. Plot the closed-loop frequency response of the error, e , in order to evaluate how well your system attenuates the disturbance response in the vicinity of 400 rad/sec (3822 rpm).

- 9.12** Design a compensator for the disk drive servo specified in Problem 9.11 so that it provides the minimum possible error due to the disturbance at 3822 rpm. Since the spin rate drifts some, define the error to be the average of that from: the nominal spin rate of 3822 rpm, $+1\%$ of the nominal, and -1% of the nominal. [Hint: Review Section 8.5.]

9.13 For

$$G(s) = \frac{1}{s^2}$$

use optimal control and estimation to design compensation that provides a 10 msec rise time with less than a 10% overshoot to a command. Determine the closed-loop system bandwidth and the system phase margin.

- 9.14** The MATLAB code below will generate some noisy data from an oscillator driven by a square wave

```

F=[0 1;-1 0];G=[0 1];H=[1 0];J=0;T=0.5;
sysC = ss(F,G,H,J)
sysD=c2d(sysC,T)
Un=[ones(1,10) zeros(1,10) ones(1,10) zeros(1,11)];
U=Un+0.1*randn(size(Un));
Yn=lsimn(sysD,U);
Ym=Yn+0.2*randn(size(Yn));

```

The measurement noise is zero mean and normally distributed with $R_e = 0.2^2$ while the process noise is zero mean and normally distributed with $R_u = 0.1^2$. Assume the initial value of the state has a covariance matrix

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and use a time-varying Kalman filter to reconstruct the state. Compare the resulting time history of the optimal state estimate with that from a constant gain Kalman filter. [Hint: FIG98.M from the Digital Control Toolbox will be useful to you.]

- 9.15** Repeat Example 9.7, but fit the best straight line to the data. [Hint: FIG97.M from the Digital Control Toolbox will be useful to you.]

• 10 •

Quantization Effects

A Perspective on Quantization

In Chapter 13 we will consider the analysis of systems having general nonlinearities; but first, in this chapter, we will consider the special nonlinearity of digital control: Numbers in the computer must be forced to fit in digital words that are defined by a finite number of bits, usually 8, 16, 32, or 64 bits. Thus far we have considered the use of the digital computer as a linear, discrete-time device for implementing control designs. Now we turn to consideration of the fact that numbers in the computer are taken in, stored, calculated, and put out with finite accuracy. The technique of representing a real number by a digital value of finite accuracy affects digital control in two principal ways. First, the variable values such as e, u , and the internal state variables, x , used in the difference equations are not exact, and thus errors are introduced into the output of the computer. As we shall see, these errors can often be analyzed as if there were noise sources in the computer. Second, the coefficients such as a_i and b_i of the difference equations, which must be stored in the computer, cannot be arbitrary real numbers but must be realizable by a finite number of bits. This can cause the machine to solve a slightly different equation than it was designed to do and has the potential to result in instability. Our purpose in this chapter is to explore methods that can be used for the analysis of these two effects: quantization of variables and quantization of coefficient parameters. The effect of quantization when the microprocessor has 32 bits is typically not noticeable. However, some products are very cost sensitive and it is desirable to use an 8 bit computer, if possible. The material in this chapter is primarily directed toward the design of such systems.

Chapter Overview

The random error model for the round-off process is first developed in Section 10.1. This section also develops several analysis methods for such random processes that are useful for quantization and then applies the random process

analysis methods from Chapter 9 to the quantization case. Section 10.2 examines the deterministic problem of parameter storage quantization errors and demonstrates the sensitivity of the errors to the structure of the difference equation mechanization. The last section, 10.3, describes some of the consequences of round-off errors and what steps can be taken to alleviate them.

10.1 Analysis of Round-Off Error

fixed point

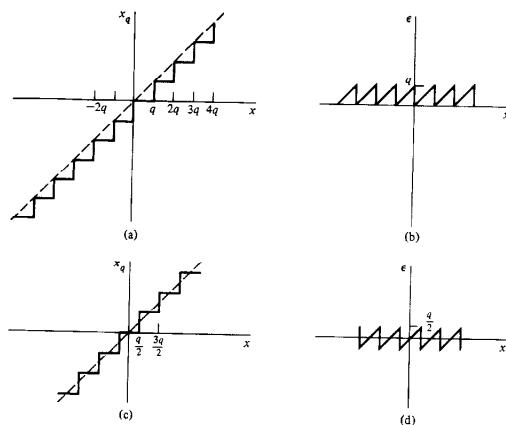
In our first analysis of finite accuracy we will assume that the computer represents each number with a fixed location for the equivalent of the decimal point, or **fixed point** representation. Although for any given number the point is fixed, considerations of amplitude scaling and dynamic range often require that different numbers in the same program have their decimal points in different locations. The consequence is that different magnitudes of errors are introduced at different locations as the results of calculations being fitted into the proper computer number format. Fixed point representation is typical in real-time control computers; however, in scientific calculations done in higher-level languages such as BASIC, MATLAB, C, **floating point** representations are mainly used, wherein the number representation has both a mantissa (magnitude and sign information) and an exponent that causes the location of the decimal point to float. With floating-point arithmetic, addition and subtraction are done without error except that the sum can overflow the limits of the representation. Overflow must be avoided by proper amplitude scaling or analyzed as a major nonlinearity. For control- or signal-processing applications, the logic is arranged so that the result of overflow is a saturation effect, a nonlinearity that will be treated in Chapter 13. In the case of multiplication, however, a double-length product is produced and the machine must reduce the number of bits in the product to fit it into the standard word size, a process we generally call **quantization**. One way to perform quantization is by ignoring the least significant half of the product, a process called **truncation**. If we assume that the numbers are represented with base 2 and that ℓ binary digits (bits) are to the right of the point (for the fractional part), the least significant bit kept represents the magnitude $2^{-\ell}$. The result of truncation will be accurate to this value (the part thrown away could be almost as large as $2^{-\ell}$). Thus a plot of the "true" value of a variable x versus the quantized value, x_q , would look like Fig. 10.1(a), where the error (shown in (b)) is decided by the quantum size, q (which is $2^{-\ell}$ under the conditions mentioned above). It is common practice in control computers to use **round-off** rather than truncation. With round-off, the result is the same as truncation if the first bit lost is a 0, but the result is increased by $2^{-\ell}$ if the first bit lost is a 1. The process is the same as is common with ordinary base 10 numbers where, for example, 5.125 is rounded to 5.13, but 5.124 becomes 5.12 to two (decimal) places of accuracy. An input-output plot of rounding is shown in Fig. 10.1(c), and the corresponding error is shown in

floating point

quantization

round-off

Figure 10.1
Plot of effects of number truncation. (a) Plot of variable versus truncated values. (b) Plot of error due to truncation. (c) Plot of variable versus rounded values. (d) Round-off error



10.1(d).¹ Notice that the maximum error with roundoff is half that resulting from truncation. The value of ϵ in a particular case depends on the word size of the particular computer in use. For control implementations the choice is based on the required accuracy and dynamic range and is limited by the expense. At this time (1997), microprocessors are readily available with word sizes of 8, 16, and 32 bits, with a cost premium on the larger sizes. One of the goals of this chapter is to analyze the effect of ℓ and thus of word size on the stability and on the errors due to quantization's effects so the designer can select the smallest word size consistent with the required performance.

We can represent either truncation or round-off by the equation

$$x = x_q + \epsilon. \quad (10.1)$$

¹ The plots in Fig. 10.1 assume that the numbers are represented in two's complement code.

quantization models

where ϵ is the error caused by the truncation or round-off of x into the digital representation x_q . Because the error due to truncation equals a constant plus round-off error, we will assume round-off in our analysis unless otherwise stated and assume that analysis of the effects of the additional constant bias when truncation is used (which is rare) can be treated separately. Clearly the process of analog-to-digital conversion also introduces a similar effect, although often with still another value for q than those resulting from round-off during arithmetic in the microprocessor.

The analysis of the effects of round-off depends on the model we take for ϵ . We will analyze three such models, which we can classify as (a) worst case, for which we will bound the error due to round-off; (b) steady-state worst case, for which we will compute the largest output possible if the system reaches a constant steady state; and (c) stochastic, for which we will present a model of the round-off error as a random process and compute the root-mean-square of the output error due to round-off.

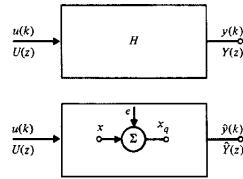
The Worst-Case Error Bound

The worst-case analysis is due to Bertram (1958). His analysis takes the pessimistic view that the round-off occurs in such a way as to cause the maximum harm, and his analysis bounds the maximum error that can possibly result from round-off. First we consider the case shown in Fig. 10.2, in which a single case of quantization is assumed to occur somewhere in an otherwise linear constant system.

There is a transfer function from the point of round-off to the output, which we will call $H_i(z)$, and we can describe the situation by the equations

$$\begin{aligned} Y(z) &= H(z)U(z), \\ \hat{Y}(z) &= H(z)U(z) - H_i(z)E_i(z; x), \\ Y - \hat{Y} &= \hat{Y} = H_i(z)E_i(z; x). \end{aligned} \quad (10.2)$$

Figure 10.2
A linear system and the introduction of one source of round-off errors



We write E_i as a function of the state variable x to emphasize that Eq. (10.2) is not linear because we do not know how to compute E_i until we have the values of x . However, we are not looking for the exact value of \hat{Y} but an upper bound on the time error, $\tilde{y}(k)$.

Because we wish to find a bound on the time signal $\tilde{y}(k)$, we need the time domain equivalent of Eq. (10.2), which is the convolution sum given by

$$\tilde{y}(n) = \sum_{k=0}^n h_i(k)\epsilon_i(n-k; x). \quad (10.3)$$

If we examine Fig. 10.1(d), we can see that whatever the exact values of ϵ_i may be, its magnitude is bounded by $q_i/2$, where q_i is the quantum value for the operation which introduces the quantization being analyzed. We can use this information to determine in short steps the bound on \tilde{y} as we did in Chapter 4 in the discussion of BIBO (bounded input, bounded output) stability. Taking the magnitudes of both sides on Eq. (10.3), we get

$$|\tilde{y}| = \left| \sum_{k=0}^n h_i(k)\epsilon_i \right|.$$

Because the sum is bounded by the sum of the magnitudes of each term, we have the inequality

$$\leq \sum_{k=0}^n |h_i| |\epsilon_i|,$$

which is the same as

$$\leq \sum_{k=0}^n |h_i| \frac{q_i}{2},$$

but by Fig. 10.1(d), the error magnitude is always less than $q_i/2$, so the output error is bounded by

$$\leq \sum_{k=0}^n |h_i| \frac{q_i}{2}.$$

Finally, the sum can only get larger if we increase the number of terms

$$|\tilde{y}(n)| < \sum_{k=0}^{\infty} |h_i| \frac{q_i}{2}. \quad (10.4)$$

Equation (10.4) is Bertram's worst-case bound. The function `owc.m`² computes it. By comparison with the condition for BIBO stability, we can conclude that if the linear system is BIBO stable in response to inputs applied at the point of the quantizer, then introduction of the quantization will not cause the system to be

² In the Digital Control Toolbox.

unstable in the BIBO sense.³ As we will see later, the system with quantization can have an output error that is nonzero either as a constant or as an oscillation; so the system may not be asymptotically stable but the output error will not grow beyond the bound given by Eq. (10.4).

◆ Example 10.1 Bertram Bound Application

For the first-order system

$$y(k+1) = \alpha y(k) + u(k), \quad (10.5)$$

compute Bertram's bound, assuming that the round-off occurs in the computation of the product αy .

Solution. Using Eq. (10.4), we have

$$\hat{y}(k+1) = \alpha \hat{y}(k) - \epsilon(k) + u(k), \quad (10.6)$$

and thus

$$\hat{y}(k+1) = \alpha \hat{y}(k) - \epsilon(k). \quad (10.7)$$

For the error system, the unit pulse response is $h_1(k) = \alpha^k$, and Eq. (10.4) is easily computed as follows for $|\alpha| < 1$

$$|\hat{y}| \leq \frac{q}{2} \sum_{k=0}^{\infty} |\alpha|^k \leq \frac{q}{2} \frac{1}{1 - |\alpha|}. \quad (10.8)$$

For this example, if $\alpha > 0$, the error due to quantization at any particular time is bounded by the DC gain from the quantizer to the output times $q/2$. This bound is only likely to be approached when a system has a constant input and has settled to its steady-state value.

The Steady-State Worst Case

The steady-state worst case was analyzed by Slaughter (1964) in the context of digital control, and by Blackman (1965) in the context of digital filters. For this analysis, we view the round-off to cause some stable, transient errors of no special concern, and we assume that all variables eventually, in the steady state, become constants. We wish to know how large this steady-state error can be as a result of round-off. We consider again the situation shown in Fig. 10.2 and thus Eq. (10.3). In this case, however, we assume that Eq. (10.3) reaches a steady state,

³ We assume that there is no saturation of the quantizer during the transient.

at which time ϵ is constant and in the range $-q/2 \leq \epsilon_{ss} \leq q/2$. Then Eq. (10.3) reduces to

$$\hat{y}_{ss}(\infty) = \sum_n h_1(n) \epsilon_{ss}.$$

The worst steady-state error is the magnitude of this signal with $\epsilon_{ss} = q/2$, which is

$$|\hat{y}_{ss}(\infty)| \leq \left| \sum_n h_1(n) \right| \frac{q}{2}. \quad (10.9)$$

There is one nice thing about Eq. (10.9): The sum is the value of the transfer function $H_1(z)$ at $z = 1$, and we can write⁴

$$|\hat{y}_{ss}(\infty)| \leq |H_1(1)| \frac{q}{2}. \quad (10.10)$$

◆ Example 10.2 Worst Case Steady-State Bound

Use the worst case analysis to compute the error bound for the first-order system given by Eq. (10.5).

Solution. In this case, the transfer function is

$$H_1(z) = 1/(z - \alpha).$$

From Eq. (10.10), we find that the bound is

$$|\hat{y}_{ss}| \leq \frac{q}{2} \frac{1}{1 - \alpha}. \quad (10.11)$$

This is the same as the Bertram worst-case bound if $\alpha > 0$.

So we see that this steady-state worst case yields the general result that the error is bounded by $q/2 \times$ dc gain from the quantization source to the output.

Equations (10.4) and (10.9) or (10.10) express the consequences of round-off from one source. For multiple sources, these equations can be extended in

⁴ A function to compute the quantization steady-state error is given by qss.m in the Digital Control Toolbox.

obvious way. For example, if we have K sources of round-off, each with possibly different quantization levels q_j , then Eq. (10.4) becomes

$$\begin{aligned} |\tilde{y}| &\leq \left\{ \sum_{n=0}^{\infty} |h_1(n)| \frac{q_1}{2} + \sum_{n=0}^{\infty} |h_2(n)| \frac{q_2}{2} + \dots \right\} \\ &\leq \sum_{j=1}^K \sum_{n=0}^{\infty} |h_j(n)| \frac{q_j}{2}. \end{aligned} \quad (10.12)$$

If all the quanta should be equal, the error bound is

$$\begin{aligned} |\tilde{y}| &\leq \left\{ \sum_{n=0}^{\infty} |h_1(n)| + \sum_{n=0}^{\infty} |h_2(n)| + \dots \right\} \frac{q}{2} \\ &\leq \sum_{j=1}^K \sum_{n=0}^{\infty} |h_j(n)| \frac{q}{2}. \end{aligned}$$

And likewise for multiple sources, Eq. (10.10) is extended to

$$|y_{ss}(\infty)| \leq [|H_1(1)| \frac{q_1}{2} + |H_2(1)| \frac{q_2}{2} + \dots + |H_K(1)| \frac{q_K}{2}]. \quad (10.13)$$

It is easily possible to express Eq. (10.10) in terms of a state-variable formulation of the equations of motion. Suppose the equations from the point of the quantization to the output are given by

$$\begin{aligned} \mathbf{x}(k+1) &= \Phi \mathbf{x}(k) + \Gamma_i \epsilon_i(k), \\ \tilde{y}(k) &= \mathbf{H} \mathbf{x}(k) + J \epsilon_i(k). \end{aligned} \quad (10.14)$$

The assumptions of the steady-state worst-case error are that $\mathbf{x}(k+1) = \mathbf{x}(k) = \mathbf{x}_{ss}$, and $\epsilon_i(k) = \epsilon_{i,ss}$. Then

$$\mathbf{x}_{ss} = \Phi \mathbf{x}_{ss} + \Gamma_i \epsilon_{i,ss}, \quad \tilde{y} = \mathbf{H} \mathbf{x}_{ss} + J \epsilon_{i,ss}. \quad (10.15)$$

Solving for \tilde{y} , we find

$$\tilde{y} = [\mathbf{H}(\mathbf{I} - \Phi)^{-1} \Gamma_i + J] \epsilon_{i,ss},$$

which is bounded by

$$|\tilde{y}| \leq \|[\mathbf{H}(\mathbf{I} - \Phi)^{-1} \Gamma_i + J]\| \frac{q_i}{2}. \quad (10.16)$$

The major advantage of the steady-state result is the vastly simpler forms of Eqs. (10.13) and (10.16) as compared to (10.12). Unfortunately, Eq. (10.13) does not always hold because of the assumption of a constant quantization error of $q/2$ in the steady state;⁵ and yet the worst-case upper bound given by Eq. (10.12) is often excessively pessimistic. However, in some cases there is great expense

⁵ Note that Eqs. (10.13) and (10.16) are *not* bounds on the error. They do, however, give a valuable estimate on error size, which is easy to compute for complex systems.

due to distortion of the data should a signal overflow, and the absolute bound can be used to select q so that no overflow is possible. An example of the use of Bertram's bound in the design of a spectrum analyzer is given in Schmidt (1978).

Stochastic Analysis of Round-Off Error

The third model for round-off error is that of a stochastic variable. The analysis will follow Widrow (1956) and has two parts: development of a stochastic model for $\epsilon(k)$ and analysis of the response of a linear system to a stochastic process that has the given characteristics. Because the development of the model requires use of somewhat sophisticated concepts of stochastic processes, and because the model can be given a very reasonable heuristic justification without this mathematical apparatus, we develop the model heuristically and proceed with an analysis of the response. A review of the necessary facts from the theory of probability and stochastic processes is to be found in Appendix D.

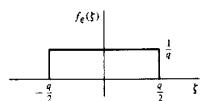
First, then, we give a heuristic argument for a stochastic model of round-off error. We begin with examination of Fig. 10.1, where we have a plot of the output versus the input of the round-off operation and a sketch of error versus amplitude of the input signal x . If we imagine that $x(n)$ is a random variable that takes values at successive sampling instants in a scattered way across the scale of values, then it seems reasonable to suppose that the sequence of errors $\epsilon(n)$ would be scattered over the entire range of possible values, which is to say, over the range from $-q/2$ to $q/2$. Furthermore, because the "teeth" in the saw-like plot of ϵ versus x are linear and contain no flat places that would signal a preference for one value of ϵ over another, it seems reasonable that the values of $\epsilon(n)$ are equally likely to be anywhere in the range $q/2 \leq \epsilon \leq q/2$. Furthermore, if the signal into the quantizer typically moves several quanta during one sample period, it seems reasonable to assume that the values of error at one sample time will not be correlated with errors at other times; that is, its spectrum would be expected to be flat, which we characterize as "white."⁶ The reflection of this argument in terms of stochastic processes is to assert that we can model $\epsilon(k)$ as a white random process having a uniform probability density from $-q/2$ to $q/2$. A plot of the uniform density is shown in Fig. 10.3.

From this density we can immediately compute the mean and variance of ϵ as follows

$$\mu_\epsilon = \mathcal{E}\{\epsilon\} = \int_{-\infty}^{\infty} \xi f_\epsilon(\xi) d\xi = \int_{-q/2}^{q/2} \xi \frac{1}{q} d\xi = \frac{1}{q} \left[\frac{\xi^2}{2} \right]_{-q/2}^{q/2} = 0. \quad (10.17)$$

⁶ If the input to the quantizer is a deterministic signal such as a square wave, a constant, or a sine wave, this argument is clearly wrong. It is also approximate if the input is a random signal with strong correlation over a time equal to a sampling period. A careful analysis is given in Clavier, et al., (1947). If the input is a sine wave, the error spectrum typically has a number of large spikes, far from flat or white.

Figure 10.3
Plot of the uniform density function



and

$$\begin{aligned}\sigma_\epsilon^2 &= \mathcal{E}[(\epsilon - \mu_\epsilon)^2] = \int_{-\infty}^{\infty} (\xi - 0)^2 f_\epsilon(\xi) d\xi = \int_{-q/2}^{q/2} (\xi)^2 \frac{1}{q} d\xi \\ &= \frac{1}{q} \left[\frac{(\xi)^3}{3} \right]_{-q/2}^{q/2} = \frac{1}{q} \left[\frac{1}{3} (q/2)^3 - \frac{1}{3} (-q/2)^3 \right] \\ &= \frac{1}{3q} \left[\frac{q^3}{8} + \frac{q^3}{8} \right] = \frac{q^2}{12}.\end{aligned}\quad (10.18)$$

Thus we assume the following white-noise model for $\epsilon(n)$ for the case of round-off error as sketched in Fig. 10.1(d)

$$\epsilon(n) = w(n), \quad (10.19)$$

where $w(n)$ has the autocorrelation function (defined in Appendix D)

$$\begin{aligned}R_w(n) &= \mathcal{E}\{w(k)w(k+n)\} = q^2/12 \quad (n = 0) \\ &= 0 \quad (n \neq 0).\end{aligned}\quad (10.20)$$

and the mean value μ_ϵ is zero.

With the model given by Eqs. (10.19) and (10.20) we can compute the mean and variance of the system error due to round-off using state-space methods. Again, we observe that analysis of the response to any constant nonzero mean μ_ϵ can be computed separately from the response to the white-noise component $w(n)$.

In order to analyze the effect of the random part of the quantization noise, let's review (see also Section 9.5) the general problem of finding the effect of zero-mean noise, w , on a linear system

$$\begin{aligned}\mathbf{x}(k+1) &= \Phi_i \mathbf{x}(k) + \Gamma_i w(k) \\ \tilde{y}(k) &= \mathbf{H}_i \mathbf{x}(k) + J_i w(k).\end{aligned}\quad (10.21)$$

We define the covariance of the state as

$$\mathbf{R}_x(k) = \mathcal{E}\{\mathbf{x}(k)\mathbf{x}^T(k)\}, \quad (10.22)$$

and, at $k+1$,

$$\begin{aligned}\mathbf{R}_x(k+1) &= \mathcal{E}\{\mathbf{x}(k+1)\mathbf{x}^T(k+1)\} \\ &= \mathcal{E}\{(\Phi_i \mathbf{x}(k) + \Gamma_i w(k))(\Phi_i \mathbf{x}(k) + \Gamma_i w(k))^T\} \\ &= \Phi_i \mathbf{R}_x(k) \Phi_i^T + \Gamma_i R_w(k) \Gamma_i^T.\end{aligned}\quad (10.23)$$

covariance propagation

Lyapunov equation

Equation (10.23) can be used with an initial value (typically zero) of \mathbf{R}_x to compute the transient development of state covariance toward the steady state. Typically, we are only interested in the steady state, which is obtained by letting $\mathbf{R}_x(k+1) = \mathbf{R}_x(k) = \mathbf{R}_x(\infty)$. In this case, Eq. (10.23) reduces to the equation (called the discrete Lyapunov equation for its occurrence in Lyapunov's stability studies)

$$\mathbf{R}_x(\infty) = \Phi_i \mathbf{R}_x(\infty) \Phi_i^T + \Gamma_i R_w \Gamma_i^T. \quad (10.24)$$

Several numerical methods for the solution of Eq. (10.24) have been developed, some based on solving Eq. (10.23) until \mathbf{R}_x no longer changes, and others based on converting Eq. (10.24) into a set of linear equations in the coefficients of \mathbf{R}_x and solving these equations by numerical linear algebra as done by dlyap.m in MATLAB.

◆ Example 10.3 Random Quantization Error for a First-Order System

Determine the output of the first-order system of Eq. (10.5) for the case where the multiplication, $\alpha x(k)$, is rounded with a quantum level q .

Solution. The first-order system of Eq. (10.5), with the rounding error represented by ϵ is

$$x(k+1) = \alpha x(k) + \epsilon(k),$$

and thus

$$\Phi = \alpha \quad \text{and} \quad \Gamma_1 = 1$$

and from Eq. (10.18), we see that

$$R_w = \frac{q^2}{12},$$

so that Eq. (10.21) is

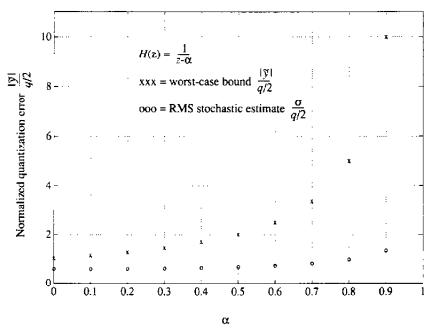
$$\begin{aligned}R_x &= \alpha R_x \alpha + (1) \frac{q^2}{12} (1) \\ R_x &= \frac{q^2}{12} \frac{1}{1-\alpha^2}.\end{aligned}\quad (10.25)$$

Taking the square root, we find that

$$\sigma_x = \sqrt{R_x} = \frac{q}{2\sqrt{3(1-\alpha^2)}}. \quad (10.26)$$

A plot of Eq. (10.26) along with the value of the worst-case bound from Eqs. (10.8) and (10.11) is shown in Fig. 10.4.

Figure 10.4
Error estimates and error bounds



In some cases, the output of a system is not one of the state elements. In other words,

$$\hat{y} = \mathbf{H}_i \mathbf{x} + J_i w,$$

and, therefore,

$$\mathcal{E}\{\hat{y}\hat{y}^T\} = \mathcal{E}\{(\mathbf{H}_i \mathbf{x} + J_i w)(\mathbf{H}_i \mathbf{x} + J_i w)^T\},$$

which is to say

$$R_y = \mathbf{H}_i \mathbf{R}_x \mathbf{H}_i^T + J_i R_w J_i^T. \quad (10.27)$$

Note that we can use Eq. (10.23) and (10.24) to compute the covariance of the state due to round-off error at several locations simply by taking \mathbf{w} to be a column matrix and \mathbf{R}_w to be a square diagonal matrix of covariances of the components of \mathbf{w} . In the multiple-source case, Γ_i is a matrix of dimension $n \times p$, where there are n states and p sources.

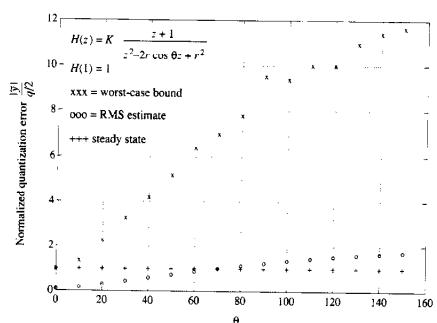
◆ Example 10.4 Random Quantization Error for a Second-Order System

Determine the quantization error of the second-order system

$$y(k+2) + a_1 y(k+1) + a_2 y(k) = \frac{1+a_1+a_2}{2} (u(k) + u(k-1)),$$

in terms of the random root-mean-square response and compare it with the worst-case bounds.

Figure 10.5
Error estimates and error bound with quantization input to a second-order system, where $r = 0.9$



Solution. The gain of the system has been set to be unity at frequency 0, and the parameters are related to the pole locations according to $a_1 = -2r \cos(\theta)$ and $a_2 = r^2$. In order to study the role of the frequency of the pole, given in normalized terms by the angle θ , we have computed the estimated errors according to Eqs. (10.4), (10.16), and (10.27) for several values of θ and plotted them in Fig 10.5. For the random-noise case, the square root of the result of Eq. (10.27) is plotted. From the plot, it is clear that the worst-case bound is quite large compared to the other two computations for most of the range of angles. Experiments in the laboratory with random inputs to the system give results that compare very favorably with the numbers given by the random model. ◆

10.2 Effects of Parameter Round-Off

We have thus far analyzed the effects of round-off on the variables such as y and u in Eq. (10.5). However, to do the calculations for a controller, the computer must also store the equation coefficients, and if the machine uses fixed point arithmetic, the parameter values must also be truncated or rounded off to the accuracy of the machine. This means that although we might design the program to solve

$$y(k+1) = \alpha y(k) + u(k),$$

it will actually solve

$$\hat{y}(k+1) = (\alpha + \delta\alpha)\hat{y}(k) + u(k) + \epsilon(k). \quad (10.28)$$

In this section, we give methods for the analysis of the effects of the parameter error $\delta\alpha$.

The principal concern with parameter variations is that the dynamic response and especially the stability of the system will be altered when the parameters are altered. One way to study this problem is to look at the characteristic equation and ask what effect a parameter change has on the characteristic roots. For example, for the first-order system described by Eq. (10.28), the perturbed characteristic equation is

$$z - (\alpha + \delta\alpha) = 0;$$

and it is immediately obvious that if we want a pole at $z = 0.995$, it will be necessary to store α to three decimal places and that the limit on $\delta\alpha$ for stability is 0.005 because a variation of this magnitude results in a pole on the unit circle. Note, however, that if we structure the α term as $\alpha = 1 - \beta$ and add the 1-term separately then $\beta = 1 - \alpha = 0.005$, and the relative accuracy requirements on β are much less than those on α . Thus we see that the details of the architecture of a realization can have a major impact on the robustness of the design to parameter variation. It is also the case that different canonical forms realized with a given parameter accuracy (word size) will be able to realize a different set of pole locations. It is often possible to select the realization structure in such a way that the desired dynamics are almost exactly realized with the available word length.

To study these matters, we consider the characteristic equation and ask how a particular root changes when a particular parameter changes. The general study of root variation with parameter change is the root locus; however, we can obtain results of some value by a linearized sensitivity analysis. We can compare the direct and the cascade realizations of Chapter 4, for example, to see which is to be preferred from a sensitivity point of view. In the direct realization, the characteristic equation is

$$z^n + \alpha_1 z^{n-1} + \cdots + \alpha_n = 0. \quad (10.29)$$

This equation has roots at $\lambda_1, \lambda_2, \dots, \lambda_n$, where $\lambda_j = r_j e^{j\theta_j}$. We assume that one of the α 's, say α_k , is subject to error $\delta\alpha_k$, and we wish to compute the effect this has on λ_j and especially the effect on r_j so that stability can be checked. For this purpose, we can conveniently write Eq. (10.29) as a polynomial $P(z, \alpha)$ that depends on z and α_i . At $z = \lambda_j$, the polynomial is zero, so we have

$$P(\lambda_j, \alpha_i) = 0. \quad (10.30)$$

If α_i is changed to $\alpha_i + \delta\alpha_i$, then λ_j also changes, and the new polynomial is

$$\begin{aligned} P(\lambda_j + \delta\lambda_j, \alpha_i + \delta\alpha_i) &= P(\lambda_j, \alpha_i) + \frac{\partial P}{\partial z} \Big|_{z=\lambda_j} \delta\lambda_j + \frac{\partial P}{\partial \alpha_i} \delta\alpha_i + \cdots \\ &= 0. \end{aligned} \quad (10.31)$$

root sensitivity

where the dots represent terms of higher order in $\delta\lambda$ and $\delta\alpha$. By Eq. (10.30) we see that the first term on the right-hand side of Eq. (10.31) is zero. If $\delta\lambda_j$ and $\delta\alpha_i$ are both small, then the higher-order terms are also negligible. Thus the change in λ_j is given to first order by

$$\delta\lambda_j \cong -\frac{\partial P/\partial\alpha_i}{\partial P/\partial z} \Big|_{z=\lambda_j} \delta\alpha_i. \quad (10.32)$$

We can evaluate the partial derivatives in Eq. (10.32) from Eq. (10.29) and the equivalent form

$$P(z, \alpha_i) = (z - \lambda_1)(z - \lambda_2) \cdots (z - \lambda_n). \quad (10.33)$$

First, using Eq. (10.29), we compute

$$\frac{\partial P}{\partial \alpha_i} \Big|_{z=\lambda_i} = \lambda_i^{n-1}. \quad (10.34)$$

and next, using Eq. (10.33), we compute⁷

$$\frac{\partial P}{\partial z} \Big|_{z=\lambda_j} = \prod_{i \neq j} (\lambda_i - \lambda_j). \quad (10.35)$$

Thus Eq. (10.32) reduces to

$$\delta\lambda_j = -\frac{\lambda_j^{n-1}}{\prod_{i \neq j} (\lambda_i - \lambda_j)} \delta\alpha_i. \quad (10.36)$$

We can say things about root sensitivity by examining Eq. (10.36). The numerator term varies with the index number of the parameter whose variation we are considering. Because we are dealing with a stable system, the magnitude of λ_j is less than one, so the larger the power of λ_j^{n-1} the smaller the variation. We conclude that the most sensitive parameter is α_n , the constant term in Eq. (10.29). However, for values of λ_j near the unit circle, the relative sensitivity decreases slowly as k gets smaller. The denominator of Eq. (10.36) is the product of vectors from the characteristic roots to λ_j . This means that if all the roots are in a cluster, then the sensitivity is high, and if possible, the roots should be kept far apart. For example, if we wish to construct a digital low-pass filter with a narrow-pass band and sharp cutoff, then the system will have many poles in a cluster near $z = 1$. If we implement such a filter in the control canonical form (Fig. 4.8c), then the sensitivity given by Eq. (10.36) will have many factors in the denominator, all small. However, if we implement the same filter in the cascade or parallel forms, then the sensitivity factor will have only one term. Mantey (1968) studies these issues and quotes an example of a narrow-bandpass filter of six poles for which

⁷ This is true if $P(z)$ has only one root at λ_j . At a multiple root, this derivative is zero and the coefficient of $\delta\alpha_i$ in Eq. (10.36) is not bounded.

the parallel realization was less sensitive by a factor of 10^{-5} . In other words, it would take 17 bits of additional accuracy to implement this example in direct form over that required for the parallel or cascade form! The papers collected by Rabiner and Rader (1972) contain many other interesting results in this area.

◆ Example 10.5 Illustration of Parameter Storage Errors

Compare the sensitivity of a fourth-order controller in cascade form

$$D(z) = \frac{(z+1)^3}{(z-0.9)(z-0.85)(z-0.8)(z-0.75)}$$

versus the same controller in direct form

$$D(z) = \frac{z^3 + 3z^2 + 3z + 1}{z^4 - 3.300z^3 + 4.0775z^2 - 2.2358z + 0.4590}.$$

Solution. If the transfer function is realized as a cascade of the first-order terms, then it is clear that the pole nearest to the unit circle is at $z = 0.9$, and a change in the coefficient by 0.1 will move this pole to the unit circle and lead to instability. This is a percent change of $(0.1/0.9)100 = 11.1\%$. On the other hand, if the controller is realized in one of the direct forms—either control or observer canonical form—then the coefficients used are those in the polynomial form. In this case, by numerical experimentation it is found that a change of a_4 from 0.4590 to 0.4580, a root is moved to the unit circle. This is a change of only $(0.001/0.4590)100 = 0.22\%$. Thus we see that the cascade form is more robust to parameter changes by a factor of almost 50 over the direct forms!

10.3 Limit Cycles and Dither

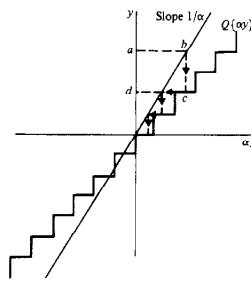
As a final study of the effects of finite word length in the realization of digital filters and compensators we present a view of the quantizer as a signal dependent gain and analyze more closely the motions permitted by this nonlinearity. One type of motion is an output that persists in spite of there being no input and that eventually becomes periodic: Such a motion is called a **limit cycle**.

To analyze a nonlinear system we must develop new tools because superposition and transforms do not apply. One such tool is to use graphic methods to solve the equations. For example, suppose we have the first-order equation

$$y(k+1) = Q(\alpha y(k)). \quad (10.37)$$

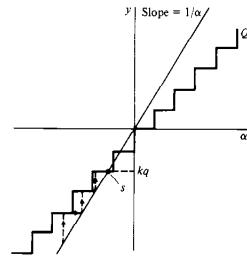
If we plot y versus αy as in Fig. 10.6, we can also plot the function $Q(\alpha y)$ and trace the trajectory of the solution beginning at the point a on the y -axis. Across from a at point b , we plot the value αy from the line with slope $1/\alpha$. Below b at c is the quantization of αy , and hence the next value of y , shown as d . We

Figure 10.6
Trajectory of a first-order system with truncation quantization



thus conclude that the trajectory can be found by starting with a point on the $(1/\alpha)$ -line, dropping to the $Q(\cdot)$ staircase, projecting left to the $(1/\alpha)$ -line again, dropping again and so on, as shown by the dashed lines and arrowheads. Note that the path will always end on the segment of zero amplitude where $\alpha y \leq q$. Now, however, suppose the initial value of y is negative. The trajectory is plotted in Fig. 10.7, where projection is up to Q and to the right to the $(1/\alpha)$ -line. Note that this time the trajectory gets stuck at the point s , and the response does *not* go to zero. The point s is an equilibrium or stationary point of the equation. If the initial value of y had a magnitude smaller than s , then the motion would have

Figure 10.7
Trajectory of a first-order system with truncation quantization and negative initial condition



moved down to the next equilibrium where the $(1/\alpha)$ -line and $Q(\cdot)$ intersect. It is clear that s represents the largest value of y that is at equilibrium and that this value depends on α . We should be able to find a relation between kq , the largest value of y at equilibrium, and α , the time constant of the filter. In fact, from inspection of Fig. 10.7 we see that the largest value of y at a stationary point is a value $y = -kq$ such that

$$-kq\alpha < -kq + q, \quad \text{or} \quad k\alpha > k - 1, \quad \text{or} \\ k < \frac{1}{1 - \alpha}, \quad \text{or} \quad |y| < q \frac{1}{1 - \alpha}. \quad (10.38)$$

The last result is the same as the worst-case bound that would be found from Eq. (10.8) with the maximum value of error q for truncation rather than $q/2$ as is the case for round-off. Thus we find in the first-order system that the worst case is rather likely to be realized.

In order to extrapolate these results to a higher-order system we must find a more subtle way to look at them. Such a viewpoint is provided by the observation that at the equilibrium points of Fig. 10.7 the gain of the quantizer is exactly $1/\alpha$. If we consider the quantizer as a variable gain, then the equilibrium occurs at those points where the combined gain of quantizer and parameter α are unity, which, for a linear system, would correspond to a pole at $z = 1$. The extrapolation of this idea to higher-order systems is to examine the range of possible **equivalent gains** of the quantizers and to conjecture that the limiting motion will be no larger than the largest signal for which the linear system with the resulting equivalent gain(s) has a pole on the unit circle.

We will illustrate the conjecture by means of the second-order control canonical form shown in Fig. 10.8(a) with the quantizer characteristic shown in Fig. 10.8(b) corresponding to round-off rather than truncation. Note from Fig. 10.8(b) that the staircase of this quantizer is centered about the line of slope 1.0 passing through the origin. The characteristic equation for this system, if quantization is ignored, is

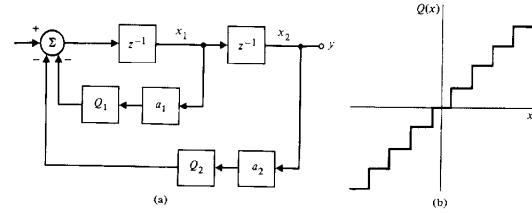
$$z^2 + a_1 z + a_2 = 0, \\ z^2 - 2r \cos \theta z + r^2 = 0,$$

where we assume that $0 < a_2 < 1$ and a_1 corresponds to a real angle. Thus we see that the system will have complex roots on the unit circle only if the action of the quantizer is to make the effective value of $a_1 = r^2 = 1.0$. Thus the condition for an oscillation is that $Q(a_1 y) \geq y$. Following an analysis similar to Eq. (10.38), the property of the quantizer is that $Q(y - q/2) \leq q/2$. If we let $a_2 y = kq - q/2$, then the oscillation condition becomes

$$kq \geq \frac{kq - q/2}{a_2}.$$

Figure 10.8

A second-order system in control canonical form and the quantizer characteristic corresponding to round-off



from which the amplitude is predicted to be less than

$$kq < \frac{1}{2} \frac{q}{1 - |a_2|}. \quad (10.39)$$

The effect of quantization on the a_1 -term influences only the frequency of the oscillations in this model. For this purpose, because the equivalent radius is 1, the equation for the digital frequency is $\theta = \cos^{-1}(a_1/2)$.

Another benefit of the view of quantization as a variable gain is the idea that a second signal of high frequency and constant amplitude added to the input of the quantizer can destroy the limit cycle. Such a signal is called a **dither**, and its purpose is to make the effective gain of the quantizer 1.0 rather than something greater than 1. Consider again the situation sketched in Fig. 10.7 with the signal stuck at s . If an outside high-frequency signal of amplitude $3q$ is added to y , then one can expect that although the output would contain a fluctuating component, the average value would drift toward zero rather than remain stuck at s . If the frequency of the dither is outside the desired pass band of the device, then the result is improved response; that is, a large constant bias or else a high-amplitude, low-frequency, self-sustained, limit-cycle oscillation can sometimes be removed this way at the cost of a low-amplitude, high-frequency noise that causes very low amplitude errors at the system output.

◆ Example 10.6 Quantization-Caused Limit Cycles and Effect of Dither

For the system in Fig. 10.8 with $a_1 = -1.78$ and $a_2 = 0.9$,

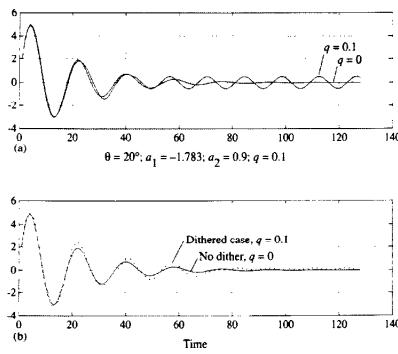
dither

- (a) determine the response to the initial condition of $x_1 = 2$ and $x_2 = 0$ with $q = 0$ and $q = 0.1$. Compare the results to that predicted by Eq. (10.39).
 (b) Determine a dither signal that improves the response.

Solution. A simulation of the system of Fig. 10.8 with and without quantization is shown in Fig. 10.9(a). The limit cycle is clearly visible for the $q = 0.1$ case. The system parameters correspond to roots at a radius of 0.95 at an angle of 20°. The quantization level was set at 0.1. The response with quantization clearly shows the limit cycle with an amplitude of 0.5, which is exactly the value given by Eq. (10.39) with $\alpha_1 = 0.9$. The period of the limit cycle can be seen from the figure to be approximately 14 samples. If we compute the frequency corresponding to the value of $a_1 = -1.78$ with the radius taken as 1 in order to reflect the fact that the system is oscillating, we find $\theta = \cos^{-1}(a_1/2) = 27^\circ$. This angle corresponds to an oscillation with a period of 13.2 samples, an estimate that is quite good, considering that the true period must be an integral number of sample periods.

In Fig. 10.9(b) are plotted the response with $q = 0$ and the response with $q = 0.1$ with dither added. In this case, after some experimentation, it was found that a square-wave dither at the Nyquist frequency worked quite well at an amplitude of $4q$. In that case the steady-state response has been reduced in amplitude from 0.5 to 0.1 and the frequency has been increased from that having a period of 14 to that having a period of 2, namely the Nyquist frequency. The experiments found that dither of less amplitude did not remove the natural limit cycle and dither of higher amplitude gave a larger steady-state error response. A random dither was

Figure 10.9
 (a) Second-order response with and without quantization showing limit cycle.
 (b) Second-order response with quantization $q = 0.1$ and dither $= 4q$



also tried, which was found to be much less effective than the Nyquist frequency square wave. Unfortunately, the selection of the amplitude and signal shape of an effective dither remains more a matter for experimentation than theory.

10.4 Summary

- Multiplication round-off errors are bounded, in steady-state, according to

$$|\tilde{y}_n(\infty)| \leq |H_i(1)| \frac{q}{2}, \quad (10.10)$$
 where H_i is the transfer function between the quantization error and the output; and $|H_i(1)|$ is the dc gain of H_i .
- Multiplication round-off errors can be analyzed under most conditions by considering them to be an additional *random* input at each multiplication. The distribution of the random input is flat, its mean is 0, its variance is $R_i = q^2/12$, and it can be considered to be white (no time correlation).
- The effect of the random multiplication round-off error is analyzed by using the discrete Lyapunov equation

$$R_x(\infty) = \Phi_i R_x(\infty) \Phi_i^T + \Gamma_i R_y \Gamma_i^T, \quad (7.30)$$

which is evaluated via MATLAB using `dyap.m` or `qrms.m`. For 16 and 32 bit computers, these errors are usually negligible. They could be significant using a computer with 8 bits or less, as will be shown in Fig. 11.6.

- The effect of parameter round-off (or parameter storage error) is systematic and has the capability to render an otherwise stable system unstable. It was shown that a parallel or cascade implementation of the difference equations significantly reduces the sensitivity to this type of error as does the use of a large number of bits to represent the parameters being stored.
- Under certain conditions (usually lightly damped systems), uncommanded oscillations will occur due to quantization called **limit cycles** which can be alleviated by the addition of low amplitude oscillations called **dither**.

10.5 Problems

- 10.1 For the system from Eq. (10.5) with $\alpha = 0.9$ and $q = .01$, what is the rms of the quantization error y that you expect for random input, $u(k)$?
 10.2 Verify Fig. 10.4 by simulating the response of the system from Eq. (10.5) to $u(k) = 1$ with and without quantization of the αy multiplication. Use $\alpha = 0.9$, $q = .02$, and compute the response from $k = 0$ to $k = 100$, repeating the simulation 20 or more times. Compute σ

of the difference between the case with and without quantization at $k = 5$ and $k = 95$ and compare with Fig. 10.4. Comment on the match you found and discuss any discrepancies.

10.3 A digital lowpass filter is described by the equation

$$y(k) = Q_2[\alpha y(k-1)] + Q_1[(1-\alpha)u(k-1)].$$

Assume that $\alpha > 0$ and that the input, $u(k)$, is a slowly varying signal that, for the purpose of this analysis, can be approximated by a constant. The magnitude of the input is restricted to be less than 0.99.

- (a) If both quantizers operate with ℓ bits to the right of the point, what is the value of the quantum q for this case?
- (b) Give an expression in terms of α for the minimum value that ℓ must have to guarantee that quantization error cannot cause the input to Q_1 to exceed 1.0.
- (c) Evaluate your expression in part (b) and give the necessary bit count if $\alpha = 0.9$, 0.98, 0.995.
- (d) Suppose the input A/D quantizer Q_1 is fixed at 12 bits with 11 bits to the right of the fixed point. At what value of α can the quantization error alone cause the input to Q_2 to equal unity?

10.4 A digital filter with the structure of Fig. 10.8 is preceded by an A/D converter having 12 bits of accuracy, with 11 bits to the right of the fixed point. The quantizer Q_1 is a 16-bit word length scaled to have 3 bits to the left of the point. The quantizer Q_2 is scaled to have only 1 (sign) bit to the left of the point in a 16-bit word length. Let $a_1 = -1.6$ and $a_2 = 0.81$.

- (a) Give the quantum sizes q_i for the converter and each internal round-off quantizer.
- (b) Give the relevant transfer functions necessary to compute the output quantization error.
- (c) Compute the steady-state worst error at the output due to quantization.
- (d) Use MATLAB to compute the worst-case bound on output quantization error.
- (e) Use MATLAB to compute the rms error due to quantization using the white-noise model.

10.5 For the second-order observer canonical form shown in Fig. 10.10,

- (a) compute the transfer functions from the quantizers to the output, y . Note carefully how many need to be computed.
- (b) If $b_1 = b_2 = 1$, $a_1 = -1.6$, $a_2 = 0.81$, what is the maximum steady-state error due to equal rounding quota of $+q/2$?
- (c) Show that the stochastic state-transition model to use Eq. (10.24) on this system has

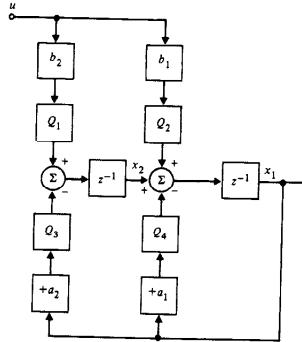
$$\Phi = \begin{bmatrix} 1.6 & 1 \\ -0.81 & 0 \end{bmatrix}, \quad \Gamma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_s(\infty) = \frac{q^2}{6}, \quad H = [1 \ 0].$$

- (d) Assume that

$$R = \begin{bmatrix} a & b \\ b & c \end{bmatrix}, \quad \vdots$$

solve Eq. (10.24) for this case, and compute $R_s(0)$ from Eq. (10.27).

Figure 10.10
A second-order system with quantization



10.6 Solve for the rms output due to round-off for the system of Fig. 10.8 using computer tools if $a_1 = -1.6$ and $a_2 = 0.81$.

10.7 The filter shown in Fig. 10.10 has input from an A/D converter that has 10 bits, 9 bits to the right of the fixed point. The input magnitude is restricted to 1. The four quantizers are based on 16-bit words scaled to have, respectively, 12, 12, 9, and 13 bits to the right of the points (i.e., Q_1 has 9 bits to the right). The parameters are $a_1 = -1.6$, $a_2 = 0.81$, and $b_1 = b_2 = 1$.

- (a) Give the quanta q_i for each quantization location.
- (b) Give the respective transfer functions from each quantizer to the output.
- (c) Use MATLAB to compute the worst-case error bound on the output quantization error.
- (d) Use MATLAB to compute the rms output error for this system based on the white-noise model of quantization error.

10.8 Consider the discrete compensation whose transfer function is

$$D(z) = \frac{z^3}{(z-0.5)(z-0.55)(z-0.6)(z-0.65)}.$$

- (a) If implemented as a cascade of first-order filters with coefficients stored to finite accuracy, what is the smallest gain perturbation that could cause instability? Which parameter would be most sensitive?
- (b) Draw an implementation of $D(z)$ in control canonical form with parameters a_1 , a_2 , a_3 , and a_4 . Compute the first-order sensitivities of these parameters according to Eq. (10.36). Which parameter is most sensitive? Compare the

sensitivities of root locations for the cascade and the control forms for these nominal root positions.

- (c) Using the root locus, find the maximum deviations possible for the parameters of the control canonical form and compare with the results of part (b).

10.9 For the equivalent-gain hypothesis of limit-cycle behavior:

- (a) Show that the equivalent gain of the round-off quantizer is given by

$$G_q \leq \frac{2k}{2k-1}, \quad k = 1, 2, \dots$$

for a signal of amplitude less than or equal to $kq - q/2$.

- (b) What is the largest-amplitude limit cycle you would expect for a second-order control canonical form system with complex roots at a radius of 0.9 and a quantization level of 0.01?

10.10 For the system of Fig. 10.8,

- (a) Use MATLAB to simulate the response to the initial conditions $x_1 = 2$ and $x_2 = 0$ with zero input. Use $a_{11} = -\sqrt{2}\alpha_1$ and $a_{12} = 0.8, 0.9, 0.95$, and 0.98. Compare the amplitudes and frequencies of the limit cycles (if any) with the values predicted.

- (b) Add dither at the Nyquist frequency to the quantizer of a_{21} with amplitude A and find the dither amplitude that minimizes the peak output in the steady state.

10.11 Repeat Example 10.6, but assume the design full scale is ± 20 units and the system is being implemented with an 8 bit cpu. Find the amplitude and frequency of the limit cycle, compare with the theory, and find a dither that improves the situation as much as possible. [Hint: FIG109.M in the Digital Control Toolbox will be useful.]

10.12 For the system of Example 10.6, assuming the design full scale is ± 20 units, find the number of bits required in the cpu to eliminate the limit cycle without any dither. [Hint: FIG109.M in the Digital Control Toolbox will be useful.]

10.13 For a system structured as in Example 10.6 with the same initial conditions, but with the coefficients selected so that the equivalent damping is $\zeta = 0.05$, assume the design full scale is ± 20 units and find the number of bits required in the cpu to eliminate the limit cycle without any dither. [Hint: FIG109.M in the Digital Control Toolbox will be useful.]

10.14 For the discrete system

$$D(z) = \frac{z^2 + 3z^2 + 3z - 1}{z^4 - 3.3000z^3 + 4.0775z^2 - 2.2358z - 0.4590}.$$

how many bits are required for parameter storage in order to keep errors in the pole locations to be less than 0.01 in the z plane in either the real or complex directions? Assume the system is to be implemented in the direct form.

• 11 •

Sample Rate Selection

A Perspective on Sample Rate Selection

The selection of the best sample rate (or rates) for a digital control system is a compromise. Generally, the performance of a digital controller improves with increasing sample rate, but cost may also increase with faster sampling. A decrease in sample rate means more time is available for the control calculations; hence slower computers are possible for a given control function, or more control capability is available for a given computer. Either result lowers the cost per function. For systems with A/D converters, slower sampling means less conversion speed is required, which will also lower cost. Furthermore, we will see that faster sampling can sometimes require a larger word size, which would also increase cost. All these arguments suggest that the best choice when considering the unit product cost is the slowest sample rate that meets all performance specifications.

On the other hand, digital controllers are often designed and built for systems where a very small number will be built. In this case, the cost of the design effort can be more than the unit product costs, and the savings in design time that are realized with very fast ($> 40 \times$ bandwidth) sampling dictates that a higher rate is the best choice.

This chapter will discuss the influence of the sample rate on system performance in order to give some insight into how to reconcile the issues for a given design.

Chapter Overview

Section 11.1 examines the fundamental limit on the sample rate imposed by the sampling theorem; Section 11.2 covers the effect on the time response, smoothness, and time lags; Section 11.3 examines the regulation effectiveness as measured by the response errors from random disturbances; Section 11.4 looks at the effect of sample rate on the sensitivity to plant parameter variations; and Section 11.5 examines how sample rate affects the error due to the measurement noise

and the influence of analog prefilters or antialiasing filters on this error. Since there are often conflicts in selecting the sample rate for different functions in a controller, some designers elect to have more than one sample rate (a "multirate system") and this topic is discussed in Section 11.6.

11.1 The Sampling Theorem's Limit

An absolute lower bound for the sample rate would be set if there is a specification to track certain command or reference input signals. This bound has a theoretical basis from the sampling theorem discussed in Section 5.2. Assuming we can represent our digital control system by a single-loop structure as depicted in Fig. 11.1, we can specify the tracking performance in terms of the frequency response from r to y . The sampling theorem states that in order to reconstruct an unknown band-limited continuous signal from samples of that signal, one must use a sample rate at least twice as fast as the highest frequency contained in the unknown signal. This theorem applies to a feedback controller like the one illustrated in Fig. 11.1 because r is an unknown signal that must be followed by the plant output y . If we want the system to track r up to a certain closed-loop bandwidth, ω_c , it follows that r will have spectral content up to that frequency. Based on the sampling theorem therefore, the sample rate must be at least twice the required closed-loop bandwidth of the system, that is

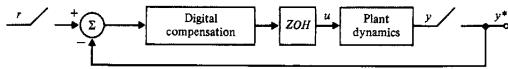
$$\frac{\omega_c}{\omega_b} > 2. \quad (11.1)$$

If a designer were to specify a certain value of the closed-loop bandwidth and then pick a sample rate that violated Eq. (11.1), the result would be that the sampled values of r would be aliased as discussed in Section 5.2, and the result would be a system response that was unstable or considerably slower than specified.

A similar argument can be made with regard to the closed-loop roots of a system, which typically would be slightly slower than the bandwidth. If a designer specified a closed-loop root by its s -plane location and failed to sample at twice its frequency, the actual root realized will be aliased and can have little

sample rate lower bound

Figure 11.1
Digital control system schematic



resemblance to that specified. In practice, no designer would consider such a low sample rate; we bring it up only because it marks the theoretical lower limit of possibilities.

11.2 Time Response and Smoothness

Equation (11.1) provides the fundamental lower bound on the sample rate. In practice, however, this theoretical lower bound would be judged far too slow for an acceptable time response. For a system with a rise time on the order of 1 sec (which translates to a closed-loop bandwidth on the order of 0.5 Hz), it would be typical to choose a sample rate of 10 to 20 Hz in order to provide some smoothness in the response and to limit the magnitude of the control steps. This means that the desired sampling multiple ($= \omega_c/\omega_b$) for a reasonably smooth time response is

$$20 < \frac{\omega_c}{\omega_b} < 40. \quad (11.2)$$

◆ Example 11.1 Double Integrator Control Smoothness vs. Sample Rate

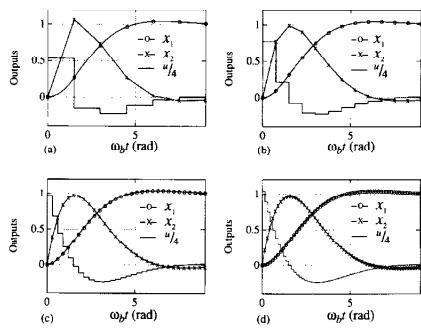
Compute the unit step response of the double integrator control problem as developed in Example 8.1 and plot the output x_1 , its rate x_2 , and control u time histories. Find the feedback gains for sample rates of 4, 8, 20, and 40 times the bandwidth, ω_b , so that the responses all have closed-loop roots in the z -plane that are equivalent to $s = 0.5\omega_b(1 \pm j)$. Discuss your results.

Solution. The four responses are shown in Fig. 11.2. The gains were computed using pole placement where the z -plane poles were computed using $z = e^{jT}$. Note that the relation $s = 0.5\omega_b(1 \pm j)$ is approximate in that the actual bandwidth of the closed-loop frequency response may be slightly different.

It is interesting to note that the x_1 response was smooth for all cases, including the one with $\omega_c/\omega_b = 4$; however, the acceleration had large discontinuities and would have had a strong tendency to excite any flexible modes and produce high stresses in the actuator and its surrounding structure. In turn, these acceleration steps produced noticeable changes of slope in the velocity. A sampling multiple of $\omega_c/\omega_b \geq 20$ appears necessary for a reasonable smoothness.

The degree of smoothness required in a particular design depends on the application and is highly subjective. The commands issued to an electric motor

Figure 11.2
Double integrator step response for the sampling multiple with ω_n/ω_s equal to (a) 4, (b) 8, (c) 20, and (d) 40



time delay

can have large discontinuities, whereas the commands issued to hydraulic actuators are best kept fairly smooth.¹ The tolerance to roughness in the response also depends on the application; for example, if a person is being affected by the controller, a smooth ride is likely desirable. An unmanned satellite controller can be rough; however, slow sampling contributes to increased pointing errors from disturbances, a topic that is discussed in the following section.

In addition to the smoothness issue, it is sometimes important to reduce the delay between a command input and the system response to the command input. A command input can occur at any time throughout a sample period; therefore, there can be a delay of up to a full sample period before the digital controller is aware of a change in the command input. All the responses in Fig. 11.2 assumed that the controller was aware of the command input at time zero and that, therefore, all the attitude responses, x_1 , responded in a similar fashion. For systems with human input commands where the system response is critical (such as fly-by-wire flight control), the time delay alone suggests that the sample period be kept to a small fraction of the rise time. A pilot flying an airplane with digital fly-by-wire flight control will complain if the sampling delay is on the order of a tenth of a second from input action to the beginning of the response. Assuming we wish to keep

¹ Sometimes lowpass filters are placed between the ZOH output and the actuators to soften the discontinuities, but the filters must be taken into account during design and compensated for.

the time delay to be 10% of the rise time, a 10-Hz sample frequency should be used for 1 sec rise time or, in terms of the nondimensional sampling multiple

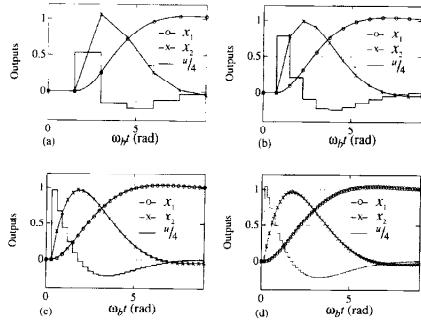
$$\frac{\omega_n}{\omega_s} \geq 20. \quad (11.3)$$

◆ Example 11.2 Double Integrator Response vs. Sample Rate

Repeat Example 11.1, but add a one cycle delay between the input command and the start of the control input in order to assess the worst case phasing of the input. Discuss the impact of this issue.

Solution. The result is shown in Fig. 11.3. It demonstrates the noticeable impact on the response for the two slower cases. The overshoot is unchanged because the controller was adjusted in each case to maintain the same z -plane roots, but the extra delay affected the rise time substantially as measured from the instant of the input command.

Figure 11.3
Double integrator step response with worst case phasing between command input and the sampler with ω_n/ω_s equal to (a) 4, (b) 8, (c) 20, (d) 40



11.3 Errors Due to Random Plant Disturbances

Disturbance rejection is an important aspect of any control system and, in many cases, is the most important one. In fact, Berman and Gran (1974) suggest that the sample rate for aircraft autopilots should be selected primarily on the basis of its effect on disturbance rejection. Indeed, this is the case for many control applications, although there are multitudes of applications with widely varying conditions where other factors are more important.

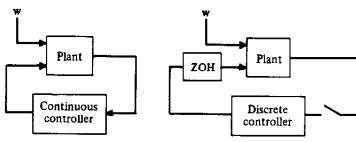
Disturbances enter a system with various characteristics ranging from steps to white noise. For determining the sample rate, the higher frequency random disturbances are the most influential; therefore, we will concentrate on their effect. In other words, we will look at disturbances that are fast compared to the plant and the sample rate, that is, where the plant noise can be considered to be white.

The ability of the control system to reject disturbances with a good continuous controller represents a lower bound on the magnitude of the error response that can be hoped for when implementing the controller digitally. In fact, some degradation over the continuous design must occur because the sampled values are slightly out of date at all times except at the very moment of sampling. In order to analyze the degradation of the digital controller as compared to the continuous controller, it is important to consider the effect of the noise [w in Eq. (4.45)] consistently with both the continuous and the digital controllers.

The block diagram in Fig. 11.4 shows the situation. The plant noise, generally a vector quantity, w , is continuous in nature and acts on the continuous part of the system independently of whether the controller is continuous or discrete. Furthermore, we are at liberty to analyze the effect of w with a certain power spectral density regardless of what values of \mathbf{R}_w or \mathbf{R}_{wpd} were used in computing the estimator gains, \mathbf{L} , and regardless of whether an estimator was used. Proceeding then, we have a continuous system represented by

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} + \mathbf{G}_1\mathbf{w}. \quad (11.4)$$

Figure 11.4
Block diagrams of the systems for disturbance analysis



where the power spectral density of w is \mathbf{R}_{wpd} (alternatively referred to as the "white-noise intensity" or "mean-square spectral density") so that the covariance of w is

$$E[w(t)w^T(t+\tau)] = \mathbf{R}_{wpd}\delta(\tau).$$

The steady-state value of the covariance of \mathbf{x} is given by the Lyapunov equation:²

$$\mathbf{FX} + \mathbf{XF}^T + \mathbf{G}_1\mathbf{R}_{wpd}\mathbf{G}_1^T = 0. \quad (11.5)$$

The solution to this equation, \mathbf{X} , ($= E[\mathbf{x}(t)\mathbf{x}(t)^T]$) represents the amplitude of the random response of the state due to the excitation from w . It will be used to establish a baseline against which discrete controllers are compared. Note that the system matrices, \mathbf{F} and \mathbf{G} , can represent a closed-loop system including a continuous controller. The solution is obtained by `lyap.m` in MATLAB.

It is also necessary to evaluate \mathbf{X} when the system has a digital controller for the identical excitation applied to the plant. In order to do this, the entire system must be transferred to its discrete equivalent as given by Eq. (9.81) and discussed in Section 4.3. The discrete equivalent of Eq. (11.5) is given by Eqs. (9.90), (10.24), and (D.34) in slightly different contexts. The desired result for our purposes here, called the discrete Lyapunov equation (see `dlyap.m` in MATLAB) is

$$\Phi\mathbf{X}\Phi^T + \mathbf{C}_d = \mathbf{X}, \quad (11.6)$$

where

$$\mathbf{C}_d = \int_0^T \Phi(\tau)\mathbf{G}_1\mathbf{R}_{wpd}\mathbf{G}_1^T\Phi^T(\tau)d\tau. \quad (11.7)$$

As discussed in Section 9.4.4, this integral can be approximated if T is shorter than all system time constants by

$$\mathbf{C}_d \cong \Gamma_1\mathbf{R}_w\Gamma_1^T, \quad \text{where} \quad \mathbf{R}_w = \frac{\mathbf{R}_{wpd}}{T}. \quad (11.8)$$

When the approximation is not valid, it is necessary to evaluate Eq. (11.7) exactly using Van Loan's (1978) algorithm, which can easily be done using `disrwr.m` in the Digital Control Toolbox. Therefore, in order to evaluate the effect of sample rate on the performance of a controller in the presence of white plant disturbances, we first evaluate Eq. (11.5) to find the baseline covariance (\mathbf{X}) and then repeatedly evaluate Eq. (11.6) with varying sample rates to establish the degradation versus sampling. It usually suffices to examine the diagonal elements for a performance measure and to compute their square roots to find the rms value, the quantity that is typically measured.

² See Kwakernaak and Sivan [1972].

◆ Example 11.3 Double Integrator Disturbance Response vs. Sample Rate

Examine the effect of sample rate on the performance of a digital control system compared to a continuous control system for the double integrator plant used in Examples 11.1 and 11.2.

- (a) Assume the plant is driven by white noise entering in the same way as the control input with $R_{w_{pd}} = 1$ and assume the use of full-state feedback.
- (b) Assume the same plant noise, but now use an estimator with only x_1 measured with additive noise with $R_e = 1$. Repeat this case assuming there is quantization in the estimator equivalent to 7, 8, and 9 bit word size.

Solution.

(a) The open loop \mathbf{F} , \mathbf{G} , and \mathbf{H} are given by Eq. (4.47). The control gain, \mathbf{K} , was determined by selecting optimal weighting matrices so that the closed-loop system had roots at $s = 0.5\omega_n i \pm j (c = 0.7)$ and we assume that $\omega_n = \sqrt{2}\omega_b$. With full-state feedback the closed-loop continuous system matrix is given by $\mathbf{F}_c = \mathbf{F} - \mathbf{GK}$. The plant disturbance noise enters the plant in precisely the same way that the control does, that is, $\mathbf{G}_d = \mathbf{G}$. Therefore the response of a hypothetical continuous controller is found by solving Eq. (11.5) with \mathbf{F} replaced with \mathbf{F}_c and \mathbf{G}_d as stated. Because we wish to illustrate only the degradation of the discrete controller compared to the continuous one, the choice of $R_{w_{pd}} = 1$ has no effect on the results.

The family of discrete controllers with different sample periods were all designed to the same continuous cost function according to the method of Section 9.3.5, the idea being that all the discrete controllers should be trying to do the same thing. Each discrete design resulted in a unique \mathbf{K} and system matrix $\Phi_d := \Phi - \mathbf{FK}$, which was used in Eq. (11.6) to evaluate \mathbf{X} . Because this example had no dynamic characteristics that were faster than the slowest sample period, the approximation for the plant noise given by Eq. (11.8) could have been used, although the exact calculation of Eq. (11.7) was actually used. The result shown in Fig. 11.5 is the ratio of the rms values for the discrete case to the continuous case. The specific curve shown is for the rms of the x_1 variable; however, the continuous case, the rms ratios are essentially identical.

If white plant disturbances were the dominant source of error in the system, one could conclude from this example that a sampling multiple of

$$\frac{\omega_1}{\omega_b} \approx 20 \quad (11.9)$$

would be a good choice. The relative errors grow quickly when sampling slower than this multiple, whereas the gain by sampling faster is simply to reduce the degradation from about 20% downward.

- (b) We can analyze the disturbance response of the system when the controller includes an estimator in a similar manner to that above. In this case, the system matrix Φ in Eq. (11.6) is defined by Eq. (8.62). Note that Φ now includes the plant dynamics as well as the estimator dynamics. The continuous control roots are in the same location as for part (a) and the continuous estimator roots were selected twice as fast as the control roots. The family of discrete controllers were found by using pole placement so that the discrete control and estimator roots were related to the continuous roots by $z = e^{j\theta}$. The lowest

Figure 11.5
Discrete controller degradation versus sample rate for full state feedback and driven by a white disturbance, Example 11.3

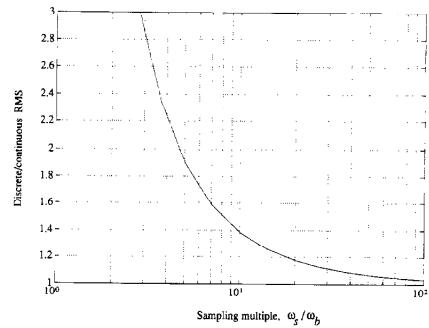
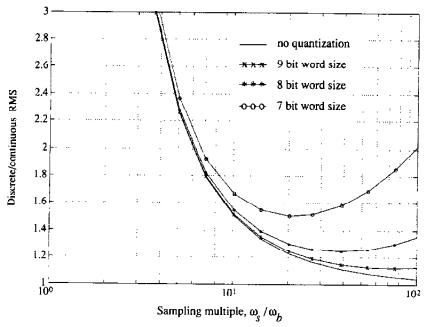


Figure 11.6
Discrete controller degradation versus sample rate for the case with an estimator with quantization errors and a white disturbance, Example 11.3



curve in Fig. 11.6 shows the discrete to continuous rms ratio and, compared to Fig. 11.5, is slightly higher. This is to be expected because the velocity information is now being estimated from a position measurement and the greater sensitivity to disturbances is the

result of the approximate differentiation that is occurring. Although the curve is generally higher at all sample rates, the conclusion concerning sample rate selection for the case where disturbances are the dominant consideration is the same: that is, sample at 20 times the bandwidth or higher.

quantization

The addition of the random noise from quantization shows that there are limits to the improving noise response as the sampling rate increases. Figure 11.6 also includes quantization noise added according to the discussion in Section 10.1. Quantization is usually important only with a fixed-point implementation of the control equations, and an assumption was therefore required as to the scaling of the signals for this analysis. In Fig. 11.6 it is assumed that the controller was scaled so that the continuous rms error due to the plant disturbance is 2% of full scale, a somewhat arbitrary assumption. But the point is not the specific magnitude as much as the notion that there is a limit, and that if a designer is dealing with a microprocessor with fewer than 12 bits, it can be useful to perform a similar analysis to determine whether the word-size errors are sufficiently large to impact selection of sample rate. With 16- and 32-bit microprocessors and a parallel or cascade realization, the increase in rms errors due to quantization at the fast sample rates is typically so small that word size is not an issue and no practical upper limit to the sample rate exists. However, if using an 8-bit microprocessor, it may be counterproductive to use too high a sample rate. ◆

resonances

The example, although on a very simple plant, shows the basic trend that usually results when a white disturbance acts on the plant: The degradation due to the discrete nature of the control over that possible with a continuous control is significant when sampling slower than 10 times the bandwidth. Except for controllers with small word sizes (8 bits or less), the performance continues to improve as the sample rate is increased, although diminishing returns tend to occur for sampling faster than 40 times the bandwidth.

Whether the rms errors due to plant disturbances are the primary criterion for selecting the sample rate is another matter. If cost was of primary importance and the errors in the system from all sources were acceptable with a sample rate at three times bandwidth, nothing encountered so far in this chapter would necessarily prevent the selection of such a slow sample rate.

On the other hand, resonances in the plant that are faster than the bandwidth sometimes can have a major impact on sample rate selection. Although they do not change the fundamental limit discussed in Section 11.1, they can introduce unacceptable sensitivities to plant disturbances. The analysis is identical to that used in the example above, except that it is mandatory in this case that an accurate evaluation of the integral in Eq. (11.7) be used because there are some plant dynamics that are faster than the sample period.

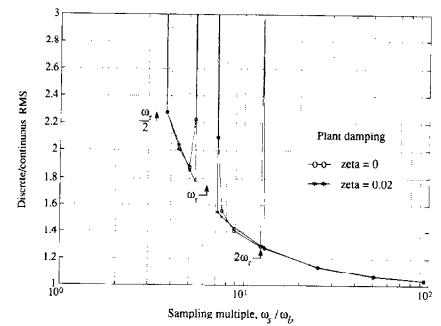
◆ Example 11.4 Double Mass-Spring Disturbance Response vs. Sample Rate

Repeat Example 11.3 for the double mass-spring system that was used in Examples 8.3 and 9.5 and described in Appendix A.4. Do part (a) for full-state feedback and part (b) for an estimator based on a measurement of d , that is, the noncollocated case where the resonance is between the sensor and the actuator.

Solution. The parameter values used are as given in Example 8.3, except that we will consider the case with no natural damping as well as the lightly damped ($\zeta = 0.02$) case used previously. The optimal continuous design was carried out in order to achieve a 6:1 ratio between the resonant mode frequency, ω_r , and the system bandwidth, ω_b . Weighting factors were applied to d and y , which provided good damping of the rigid body mode and increased slightly the damping of the resonance mode. The family of discrete designs were found by using pole placement so that the discrete roots were related to the continuous roots by $\zeta = e^{j\theta}$. The result for the full state feedback case is shown in Fig. 11.7. The general shape of the curves is amazingly similar to the previous example, but there are three highly sensitive sample rates at $2\omega_r$, ω_r , and $\omega_r/2$. Note that these are *not* resonances in the traditional sense. The curves represent the ratio of the response of the discrete system compared to the continuous, so that large spikes mean that the discrete controller at that sample rate exhibits poor disturbance rejection compared to the continuous controller. The excitation is broad band for both controllers at all sample rates. Also note that there is *no* peak at $2\omega_r$ for the case with $\zeta = 0.02$.

The origin of the sensitivity peak at exactly ω_r is that the controller has no information about the resonance at these critical sample rates. The sampled resonance value will be constant

Figure 11.7
Discrete controller degradation for the double mass-spring system using full state feedback, Example 11.4



no matter what the phase relationship between sampler and the resonant mode, thus producing an unobservable system. The additional peak at $2\omega_r$ arises from a similar unobservability, except that in this case the unobservability occurs only when the phasing is such that the samples are at the zero crossings. This peak is significantly less likely to occur and vanishes with a small amount of natural damping.

Figure 11.8 shows the sensitivity to the sampling rate ω_s .

controller than shown by the example, the sensitivity peaks would have been more pronounced because use of the mode information by the controllers would have been more critical to the performance. Furthermore, use of a colocated sensor and actuator that