

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

Compositor, 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 or 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		
3.5 Problems	69		
4 Discrete Systems Analysis	73	5 Sampled-Data Systems	155
4.1 Linear Difference Equations	73	5.1 Analysis of the Sample and Hold	156
4.2 The Discrete Transfer Function	78	5.2 Spectrum of a Sampled Signal	160
		5.3 Data Extrapolation	164
		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
5.7 Problems	183
5.8 Appendix	186

6 Discrete Equivalents 187

6.1 Design of Discrete Equivalents via Numerical Integration	189
6.2 Zero-Pole Matching Equivalents	200
6.3 Hold Equivalents	202
6.3.1 Zero-Order Hold Equivalent	203
6.3.2 A Non-Causal First-Order-Hold Equivalent: The Triangle-Hold Equivalent	204
6.4 Summary	208
6.5 Problems	209

7 Design Using Transform Techniques 211

7.1 System Specifications	212
7.2 Design by Emulation	214
7.2.1 Discrete Equivalent Controllers	215
7.2.2 Evaluation of the Design	218
7.3 Direct Design by Root Locus in the z -Plane	222
7.3.1 z -Plane Specifications	222
7.3.2 The Discrete Root Locus	227
7.4 Frequency Response Methods	234
7.4.1 Nyquist Stability Criterion	238
7.4.2 Design Specifications in the Frequency Domain	243
7.4.3 Low Frequency Gains and Error Coefficients	259
7.4.4 Compensator Design	260
7.5 Direct Design Method of Ragazzini	264
7.6 Summary	269
7.7 Problems	270

8 Design Using State-Space Methods 279

8.1 Control Law Design	280
------------------------	-----

8.1.1 Pole Placement	282
8.1.2 Controllability	285
8.1.3 Pole Placement Using CACSD	286
8.2 Estimator Design	289
8.2.1 Prediction Estimators	290
8.2.2 Observability	293
8.2.3 Pole Placement Using CACSD	294
8.2.4 Current Estimators	295
8.2.5 Reduced-Order Estimators	299
8.3 Regulator Design: Combined Control Law and Estimator	302
8.3.1 The Separation Principle	302
8.3.2 Guidelines for Pole Placement	308
8.4 Introduction of the Reference Input	310
8.4.1 Reference Inputs for Full-State Feedback	310
8.4.2 Reference Inputs with Estimators: The State-Command Structure	314
8.4.3 Output Error Command	317
8.4.4 A Comparison of the Estimator Structure and Classical Methods	319
8.5 Integral Control and Disturbance Estimation	322
8.5.1 Integral Control by State Augmentation	323
8.5.2 Disturbance Estimation	328
8.6 Effect of Delays	337
8.6.1 Sensor Delays	338
8.6.2 Actuator Delays	341
8.7 *Controllability and Observability	345
8.8 Summary	351
8.9 Problems	352

9 Multivariable and Optimal Control 359

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.

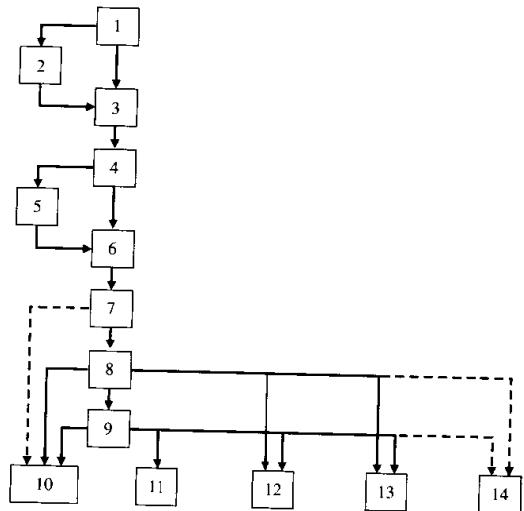
In Chapter 10 the nonlinear phenomenon of amplitude quantization and its effects on system error and system dynamic response are studied. Chapter 11 presents methods of analysis and design guidelines for the selection of the sampling period in a digital control system. It utilizes the design methods 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 Rahn, 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.

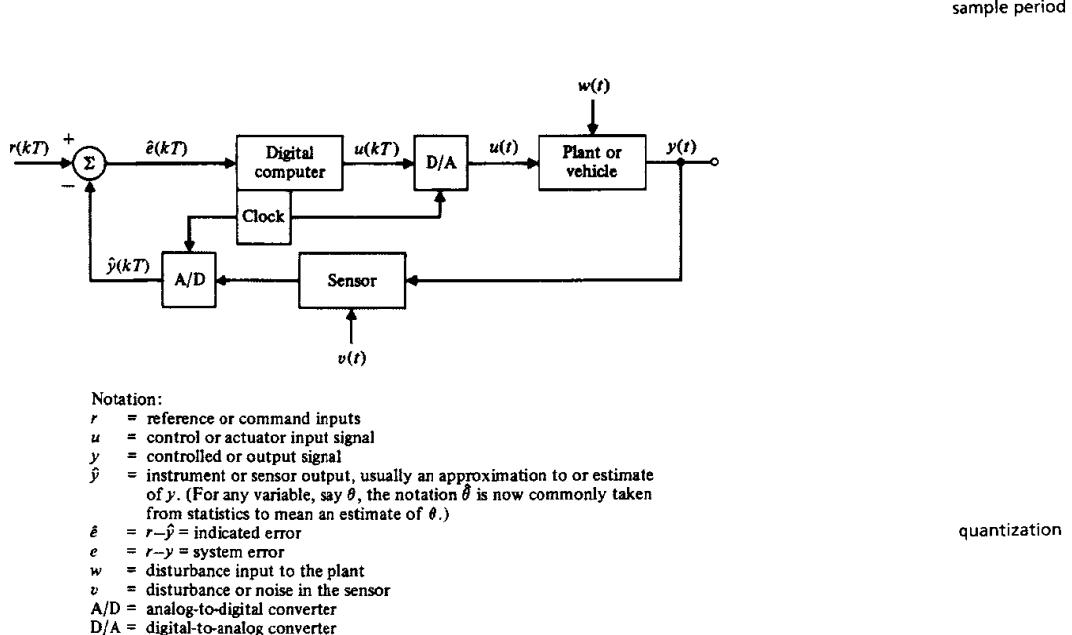
1.1 Problem Definition

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**.

Figure 1.1
Block diagram of a basic digital control system



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} , 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).

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



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 Matrix, 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. 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

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

can be written in the state-variable form as

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

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 **P**, **F**... 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,

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

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

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

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

$$G(s) = \frac{y(s)}{u(s)} = \mathbf{H}(s\mathbf{I} - \mathbf{F})^{-1}\mathbf{G} + J.$$

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

$$\text{sys} = \text{tf}(\text{ss}(\mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{J}))$$

or the zero-pole-gain form by

$$\text{sys} = \text{zpk}(\text{ss}(\mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{J})).$$

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

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

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

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

ramps

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

parabolas

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

and sinusoids

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

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

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

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

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

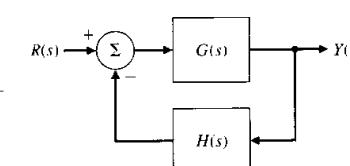


Figure 2.1
An elementary feedback system

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

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

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

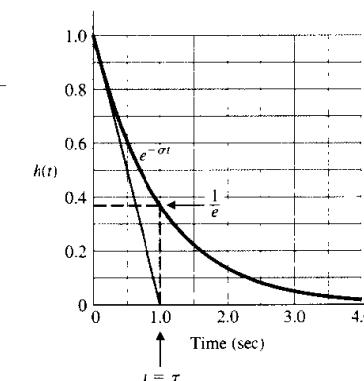
$$H(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}. \quad (2.13)$$

poles

zeros

impulse response

Figure 2.2
First-order system response



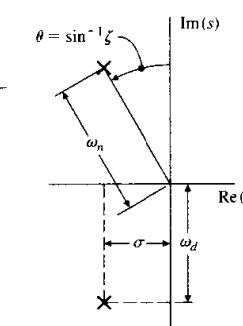
damping ratio

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 2*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

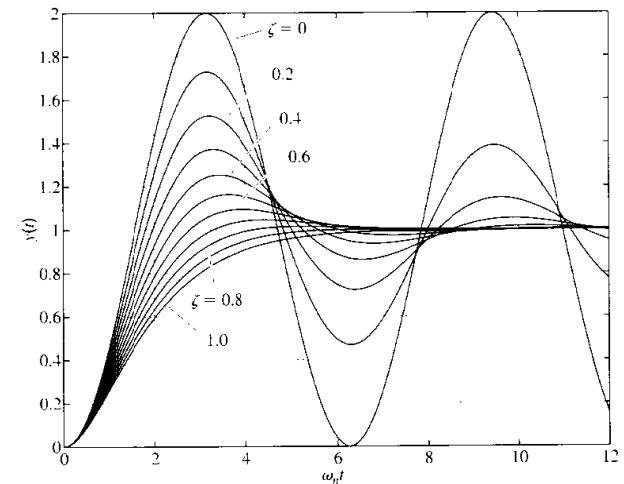
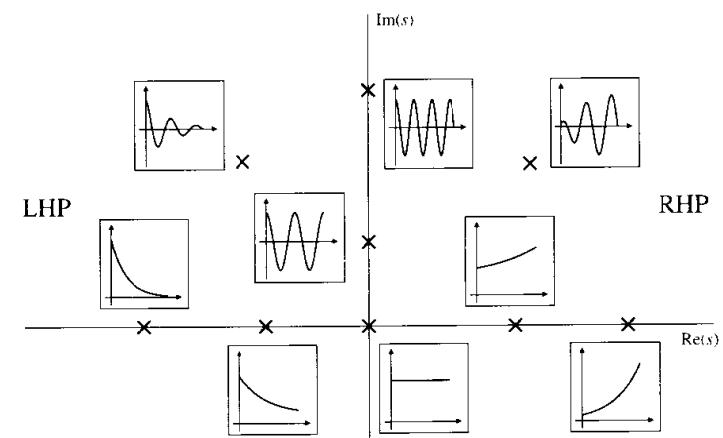


Figure 2.5
Time functions associated with points in the s -plane



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:

The **rise time** t_r is the time it takes the system to reach the vicinity of its new set point.

The **settling time** t_s is the time it takes the system transients to decay.

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).

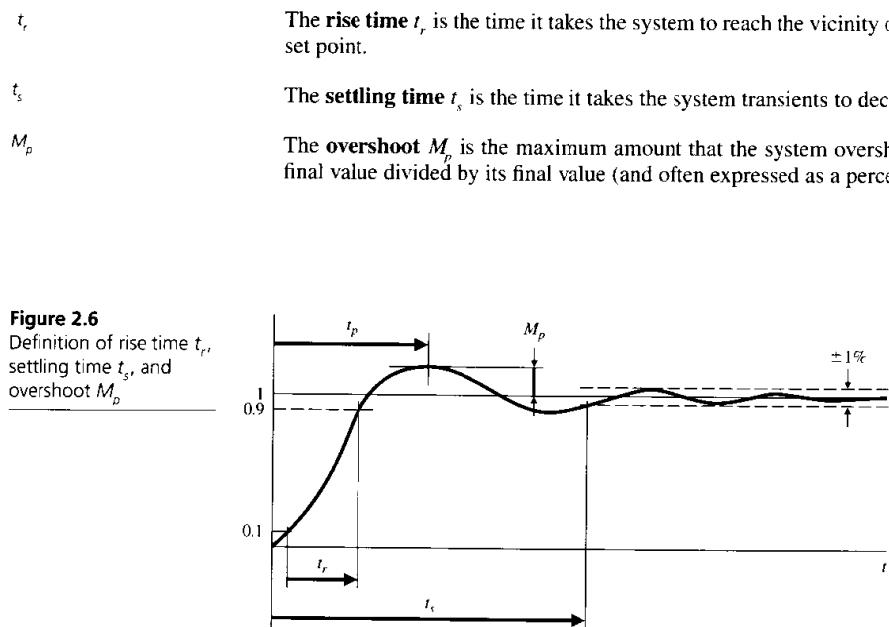


Figure 2.6
Definition of rise time t_r , settling time t_s , and overshoot M_p

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

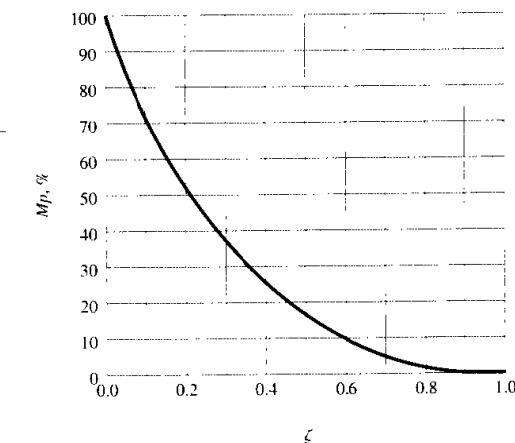


Figure 2.7
Plot of the peak overshoot M_p versus the damping ratio ζ for the second-order system

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

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

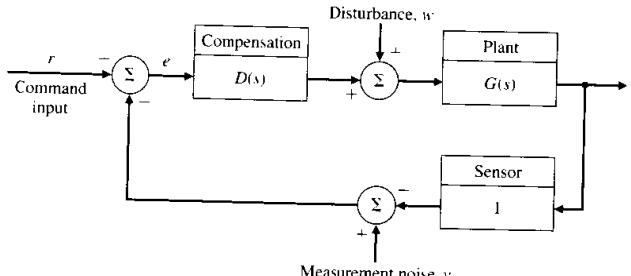
sometimes referred to as the **complementary sensitivity**. In this case, the dynamic characteristics and stability are determined by the poles of the closed-loop transfer function, that is, the roots of

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

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_p 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	$\frac{1}{K_v}$	∞
Type 2	0	0	$\frac{1}{K_a}$

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

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

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.

root locus definition

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 $()$.

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 $b(s)$) 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

$$n - m = \text{number of asymptotes}$$

$$n = \text{order of } a(s)$$

$$m = \text{order of } b(s)$$

$$\alpha = \frac{\sum p_i - \sum z_i}{n - m} = \frac{-a_1 + b_1}{n - m};$$

$$\phi_i = \frac{180^\circ + (l-1)360^\circ}{n - m}, \quad l = 1, 2 \dots n - m.$$

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

$$q\phi_{dep} = \sum \psi_i - \sum \phi_i - 180^\circ - l360^\circ$$

$$q\psi_{arr} = \sum \phi_i - \sum \psi_i + 180^\circ + l360^\circ$$

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

O 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

$$q\phi_{dep} = \sum \psi_i - \sum \phi_i - l360^\circ$$

$$q\psi_{arr} = \sum \phi_i - \sum \psi_i + l360^\circ.$$

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 $s = \pm j\omega$.

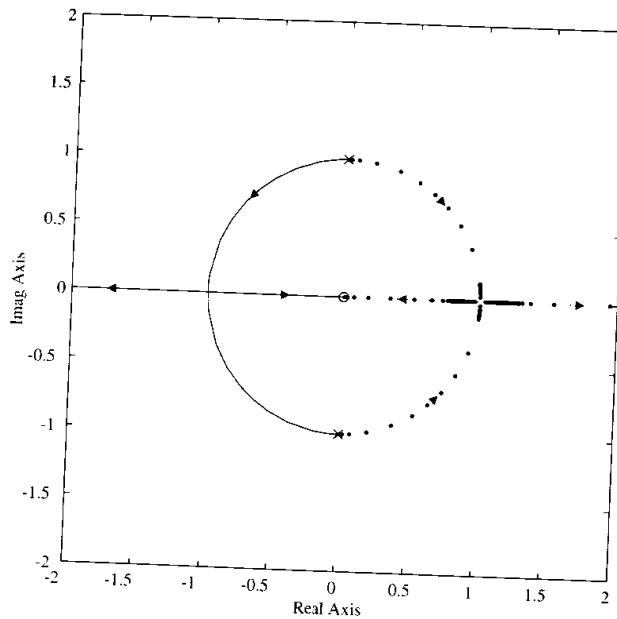
STEP 2: There is a locus on the entire negative real axis.

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

1. 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 $D(s) = K \frac{s+3}{s+10}$.
2. Find the gain K associated with the point of maximum damping and plot the step response with that value of K .
3. 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.

1. 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).
```

2. The statement

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

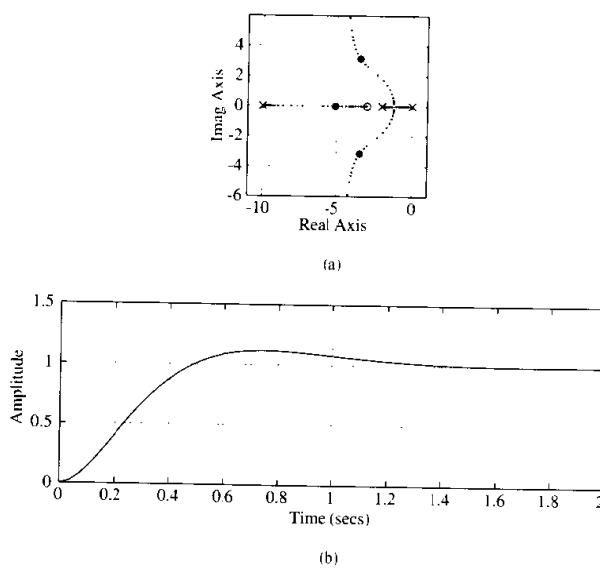
will place a cross-hair on the plot which can be moved with the mouse to the desired point on the locus in order to find the associated gain K and pole locations p . Given this value of K , ($\cong 3.7$) the script

```
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, $t_s \approx 1.4$ 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

$$t_r \approx \frac{1.8}{\omega_n} = 0.38 \text{ sec}$$

$$t_s \approx \frac{4.6}{\sigma} = 1.3 \text{ sec}$$

$$M_p \approx e^{-\pi\zeta\sqrt{1-\zeta^2}} = 4\% \quad (\text{Fig. 2.7}).$$

The rule-of-thumb values based on the second order system with no zeros predict a t_r and t_s 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 $s = 3$. ◆

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

$$u(t) = U_o \sin \omega_1 t,$$

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**, $A(\omega_1)$ of y with respect to the input, $= |G(j\omega_1)|$ and the **phase**, $\phi(\omega_1)$, is $\angle G(j\omega_1)$; 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)`.

bandwidth

2.4.1 Specifications

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_{bw} 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 $\omega = \omega_n$, which is $\frac{1}{2\xi}$ for $\xi < 0.5$.

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

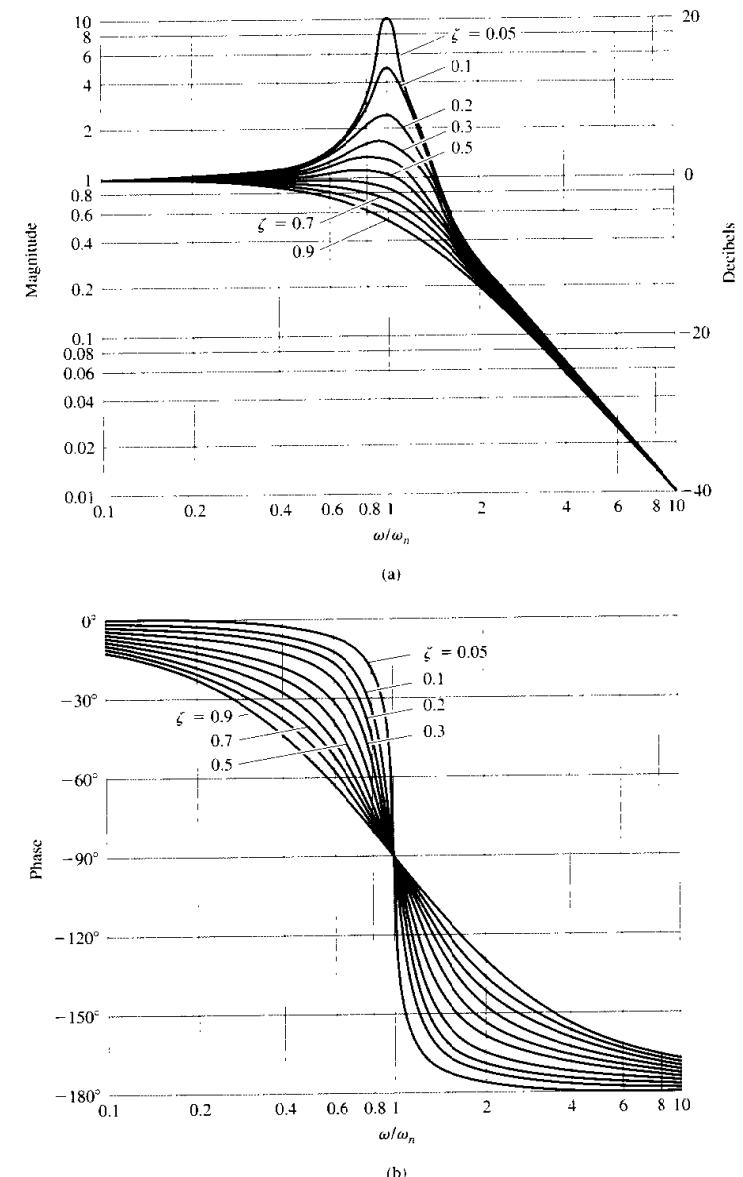
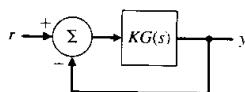
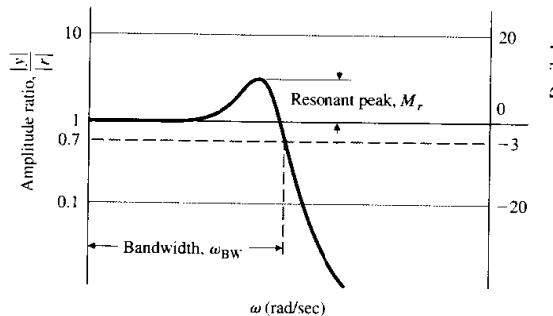


Figure 2.12

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 dB) at first order numerator breaks and decreasing it by a factor of 0.707 (-3 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 that $|G(j\omega)| = 1/(2\xi)$ 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.

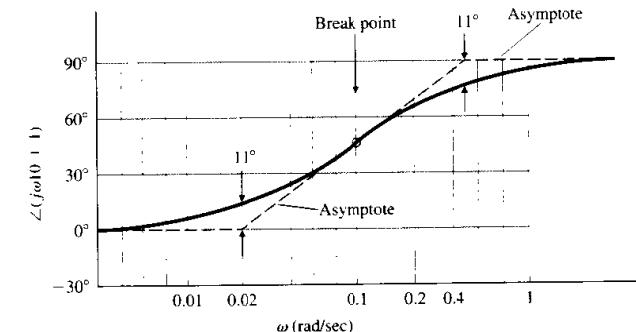
2.4.3 Steady-State Errors

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_o 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 $K_v = K_o$.

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 the 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

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

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

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.

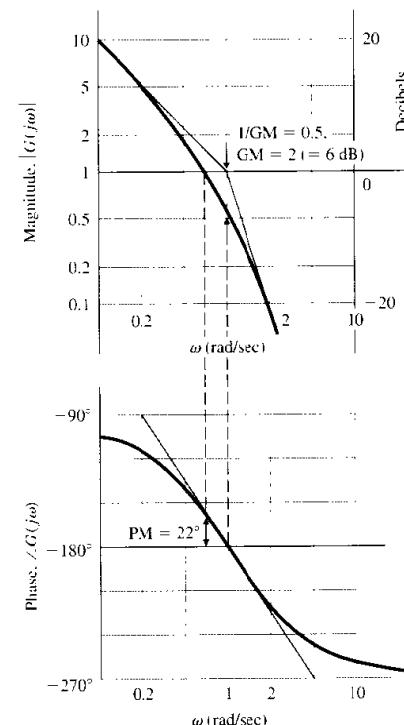
Using MATLAB⁴, the statements

```
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 $K = 0.2$

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



2.5 Compensation

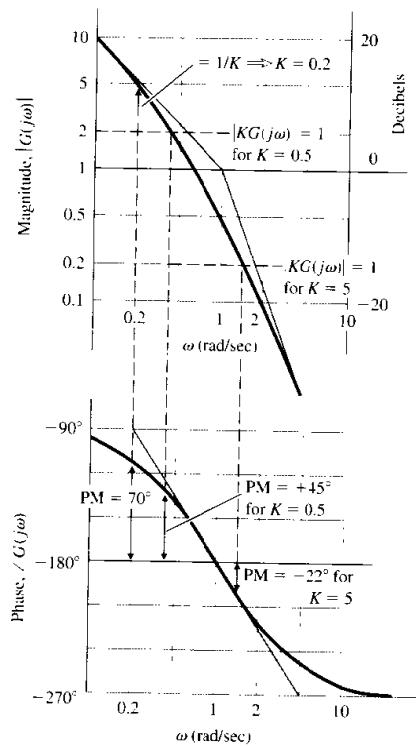
If the plant dynamics are of such a nature that a satisfactory design cannot be achieved by adjustment of the feedback gain alone, then some modification or **compensation** must be made in the feedback to achieve the desired specifications. Typically, it takes the form

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

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

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.

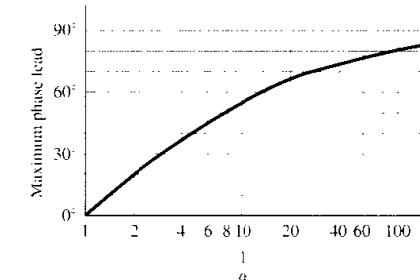
Figure 2.16PM versus K for
 $1/(s + 1)^2$ 

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 2.17Maximum 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,

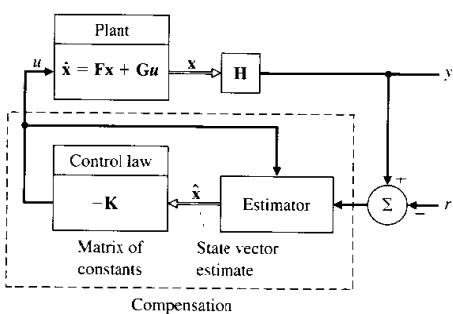
$$u = -Kx = -[K_1 \ K_2 \ \cdots \ K_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \quad (2.35)$$

pole placement

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_i .

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

$$\dot{x} = Fx - GKx. \quad (2.36)$$

The characteristic equation of this closed-loop system is

$$\det[sI - (F - GK)] = 0. \quad (2.37)$$

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 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 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 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 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.m`. 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, F and G , and a vector, p , of n desired pole locations. Their output is the feedback gain K . Thus the MATLAB statements

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

will provide the desired value of 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

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

and $R = 1$.

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

$$\mathbf{F} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{H} = [1 \ 0], \quad \mathbf{J} = 0.$$

(a) The desired characteristic equation is

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

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

$$\mathbf{K} = [9 \ -4.8].$$

(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

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

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 \mathbf{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, $\hat{\mathbf{x}}$, of the full state vector, \mathbf{x} , can be obtained based on measurements of the output, y , from

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

Here \mathbf{L} is a proportional gain defined as

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

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

$$\dot{\tilde{\mathbf{x}}} = (\mathbf{F} - \mathbf{L}\mathbf{H})\tilde{\mathbf{x}}. \quad (2.42)$$

The characteristic equation of the error is now given by

$$\det[s\mathbf{I} - (\mathbf{F} - \mathbf{L}\mathbf{H})] = 0. \quad (2.43)$$

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

Errors in the model of the plant ($\mathbf{F}, \mathbf{G}, \mathbf{H}$) cause additional errors to the state estimate from those predicted by Eq. (2.42). However, \mathbf{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 \mathbf{L} is approached in exactly the same fashion as \mathbf{K} is selected in the control-law design. If we specify the desired location of the estimator error poles as

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

then the desired estimator characteristic equation is

$$\alpha_e(s) \triangleq (s - \beta_1)(s - \beta_2) \cdots (s - \beta_n). \quad (2.44)$$

We can solve for \mathbf{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

$$\det[s\mathbf{I} - (\mathbf{F}^T - \mathbf{H}^T\mathbf{L}^T)] = 0. \quad (2.45)$$

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

$\mathbf{L} = \text{acker}(\mathbf{F}', \mathbf{H}', \mathbf{p})'$ or $\mathbf{L} = \text{place}(\mathbf{F}', \mathbf{H}', \mathbf{p})'$

observability

optimal estimation

where \mathbf{F}' is indicated in MATLAB as \mathbf{F}' , etc.

There will be a unique solution for \mathbf{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 \mathbf{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

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{G}u + \mathbf{G}_1w, \quad (2.46)$$

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

$$y = \mathbf{H}\hat{\mathbf{x}} + v. \quad (2.47)$$

The estimator error equation with these additional inputs is

$$\dot{\tilde{\mathbf{x}}} = (\mathbf{F} - \mathbf{L}\mathbf{H})\tilde{\mathbf{x}} + \mathbf{G}_1w - \mathbf{L}v. \quad (2.48)$$

In Eq. (2.48) the sensor noise is multiplied by \mathbf{L} and the process noise is not. If \mathbf{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 \mathbf{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 \mathbf{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

$\mathbf{L} = \text{kalman}(\text{sys}, \mathbf{R}_w, \mathbf{R}_v)$.

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

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

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 mechanization

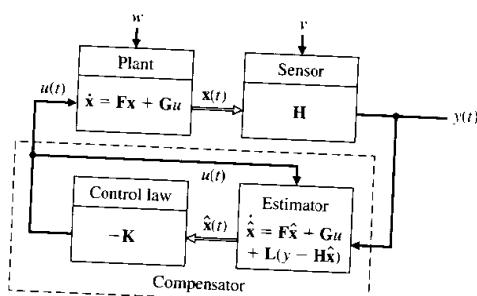
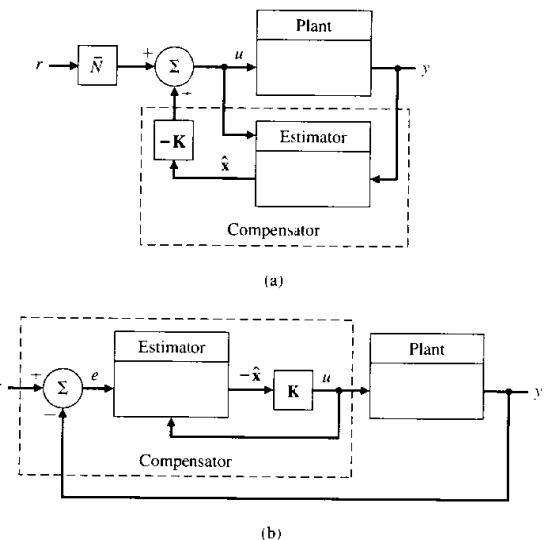


Figure 2.20
Possible locations for introducing the reference input:
(a) compensation in the feedback path,
(b) compensation in the feedforward path



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

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

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

$$u = -[K_1 \ K_0] \begin{bmatrix} x_I \\ \mathbf{x} \end{bmatrix},$$

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).

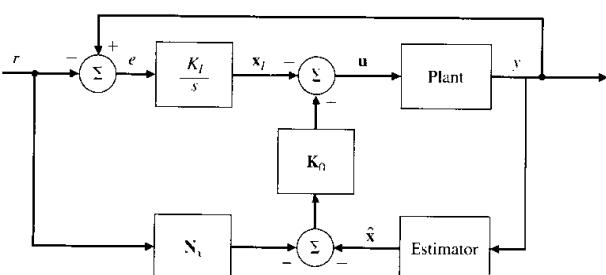


Figure 2.21

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

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

- (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 $s = -2 \pm 2j, -2 \pm 8j$.
- (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

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

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

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

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, $\omega_{BW} > 1$ 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

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

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

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

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

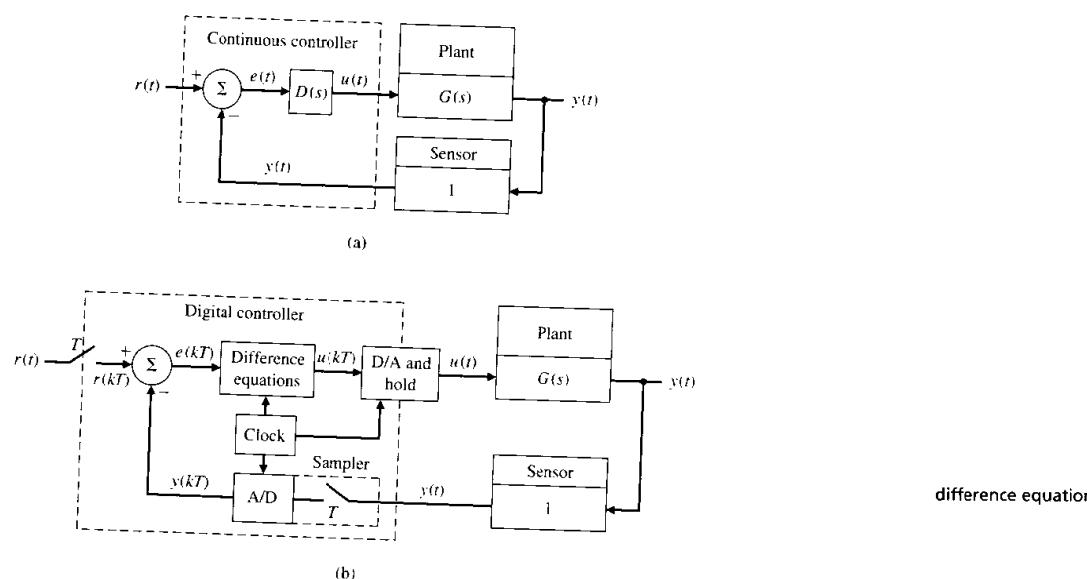
sampling on the performance of the system and a simple way to analyze that effect. Section 3.3 shows how to convert a continuous PID control law to the digital form.

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



sample period
sample rate

ZOH

Euler's method

difference equations

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

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

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:

$$\alpha_1 = 1 - bT$$

$$\alpha_2 = K_o(aT - 1)$$

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

$$x = \alpha_1 u + \alpha_2 e$$

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

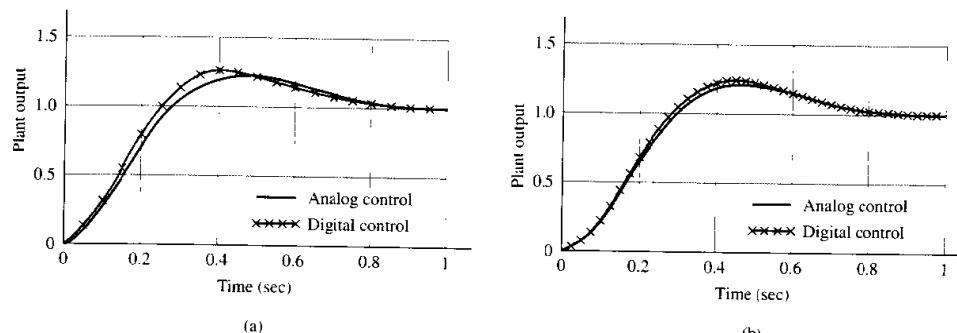
```
numD = 70*[1 2], denD = [1 10]
numG = 1, denG = [1 1 0]
sys1 = tf(numD,denD)*tf(numG,denG)
sysCL = feedback(sys1,1)
step(sysCL).
```

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

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

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

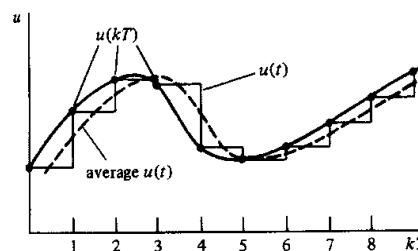
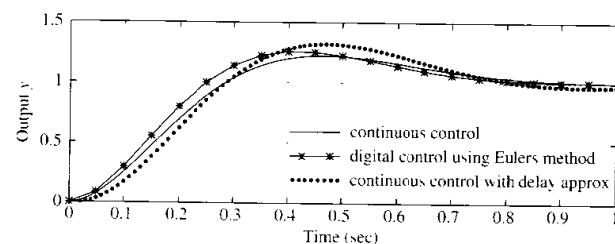


Figure 3.4

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

Figure 3.5
Frequency response for Example 3.3

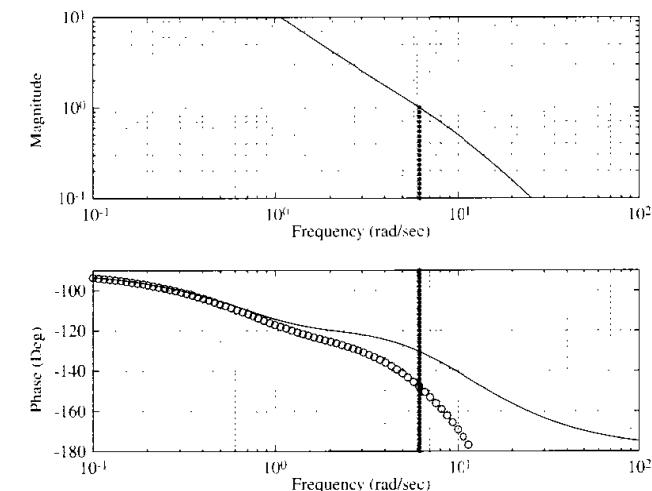
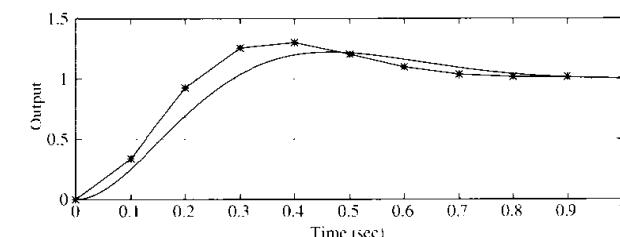


Figure 3.6
Continuous and digital responses for Example 3.3 (at 10 Hz sample rate)



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_o , 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.

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

$$\dot{u} = K \left(\dot{e} + \frac{1}{T_I} e + T_D \ddot{e} \right)$$

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 $\omega_n \cong 1800$ 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

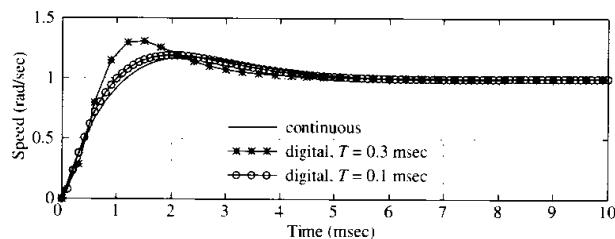
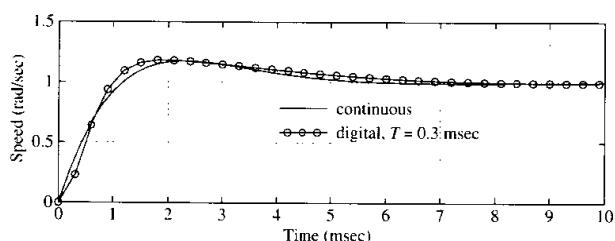


Figure 3.8
Effect of PID tuning on the digital response, Example 3.4



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

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

can be implemented digitally using Eq. (3.17)

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

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:

- 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.
- 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

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

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

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

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

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

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

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

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

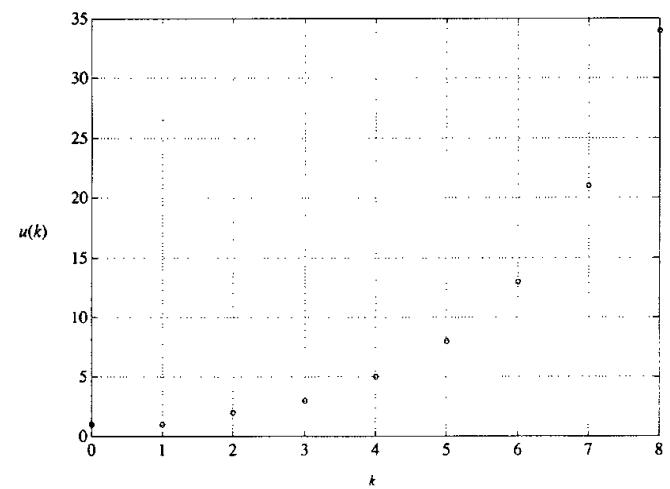
To solve a specific CCDE is an elementary matter. We need a starting time (k -value) and some initial conditions to characterize the contents of the computer memory at this time. For example, suppose we take the case

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

and start at $k = 2$. Here there are no input values, and to compute u_2 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, the output of the system represented by Eq. (4.4) would seem to be growing, to 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^{st} 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 $u(k) = Az^k$, we get the equation

$$Az^k = Az^{k-1} + Az^{k-2}.$$

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

$$A_1 = \frac{\sqrt{5} + 1}{2\sqrt{5}},$$

$$A_2 = \frac{\sqrt{5} - 1}{2\sqrt{5}}.$$

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

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

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

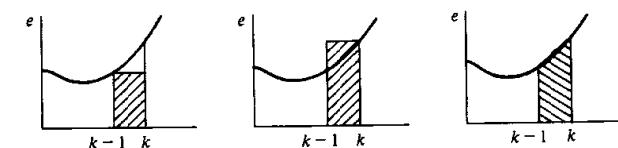


Figure 4.2
Plot of a function and alternative approximations to the area under the curve over a single time interval

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

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

Trapezoid rule

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 $e_k = kT$ 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}

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

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

$$\begin{aligned} \sum_{k=-\infty}^{\infty} e_k z^{-k} &= \sum_0^{\infty} e^{-akT} z^{-k} \\ &= \sum_0^{\infty} (e^{-aT} z^{-1})^k \\ &= \frac{1}{1 - e^{-aT} z^{-1}} \\ &= \frac{z}{z - e^{-aT}} \quad |z| > e^{-aT}. \end{aligned}$$

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 $\stackrel{\triangle}{=}$ 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 $\sum_0^{\infty} e_k z^{-k}$. 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_k and u_k 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

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

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_1 and we take b_1 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

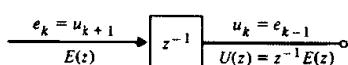
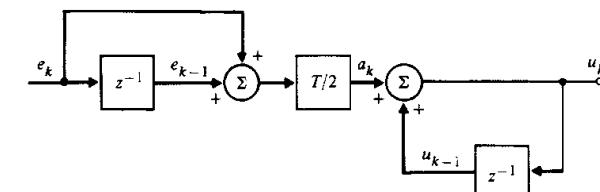


Figure 4.3
The unit delay

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

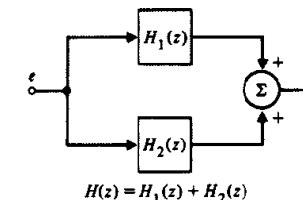
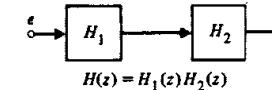


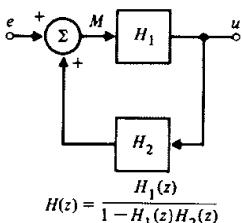
Figure 4.6
Block diagram of cascade blocks



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

control canonical form

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 $x_3(k+1) = x_2(k)$ and $x_2(k+1) = x_1(k)$. 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

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad (4.23)$$

$$\mathbf{A}_c = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

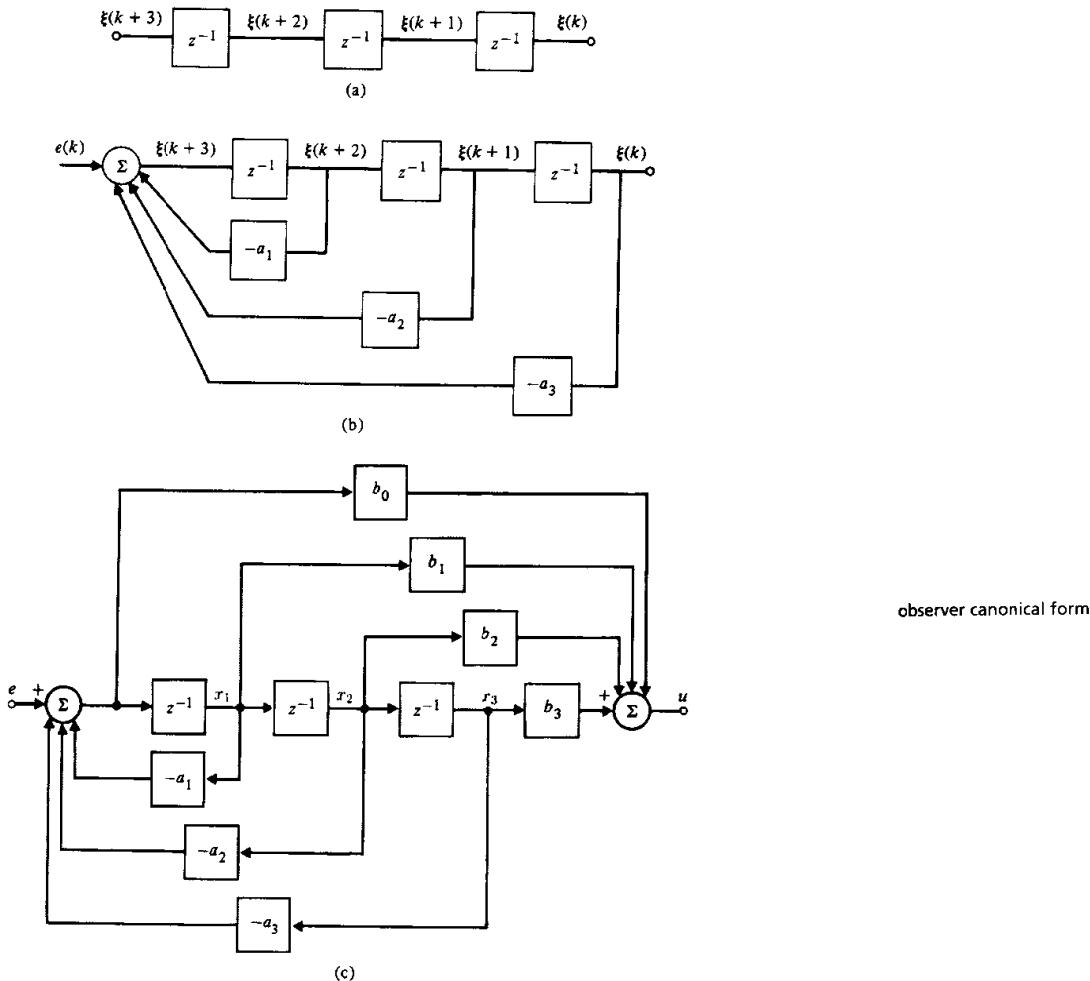
and

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

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



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_1 for example not only reaches the output through b_1 but also by the parallel path with gain $-b_0a_1$. The complete equation is

$$u = (b_1 - a_1b_0)x_1 + (b_2 - a_2b_0)x_2 + (b_3 - a_3b_0)x_3 + b_0e.$$

In vector/matrix notation, we have

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

where

$$\mathbf{C}_c = [b_1 - a_1b_0 \quad b_2 - a_2b_0 \quad b_3 - a_3b_0] \quad (4.25)$$

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

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

$$z^3u + a_1z^2u + a_2zu + a_3u = b_0z^3e + b_1z^2e + b_2ze + b_3e.$$

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

$$b_3e - a_3u = z^3u + a_1z^2u + a_2zu - b_0z^3e - b_1z^2e - b_2ze.$$

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_3u and $-b_2e$, 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 $u(k)$

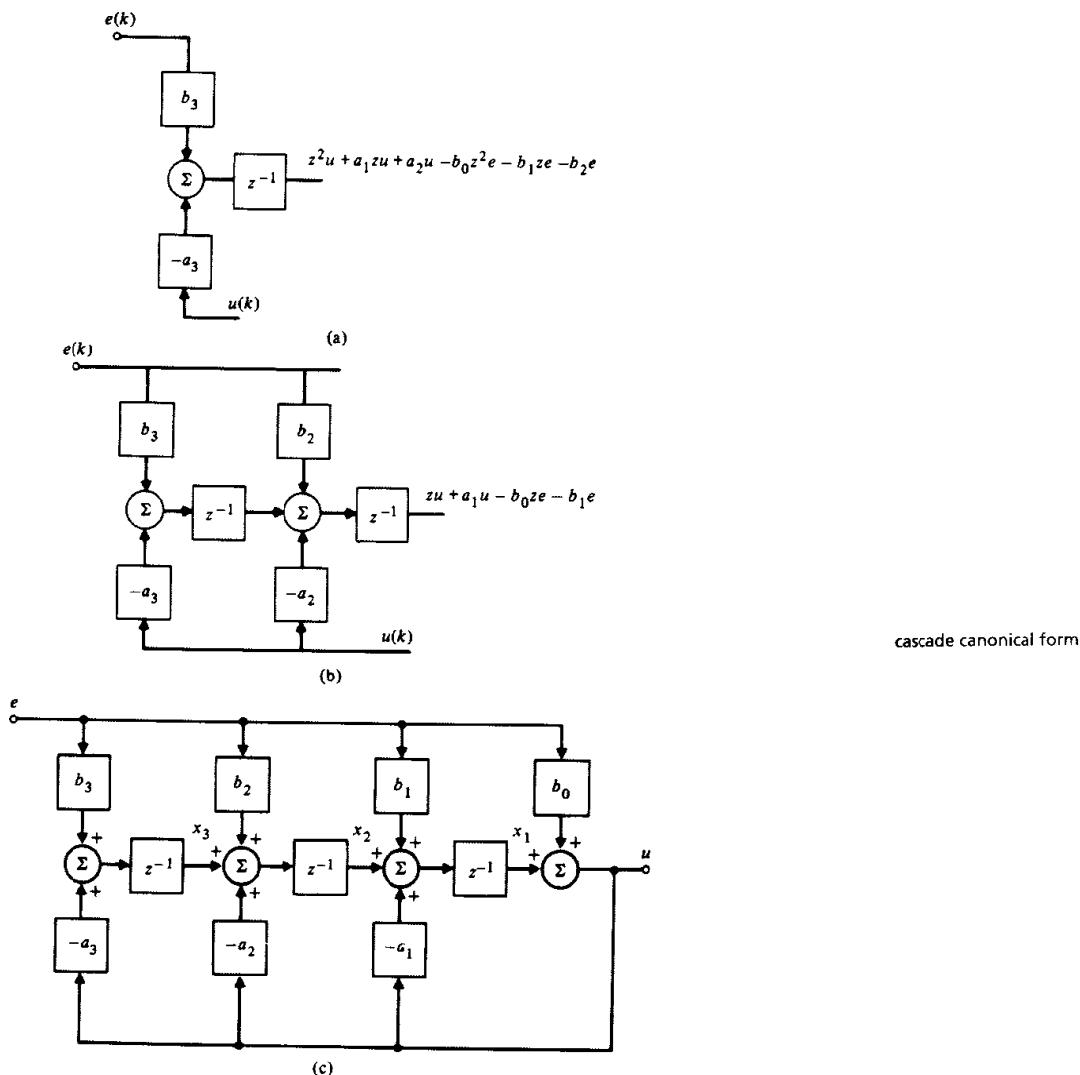
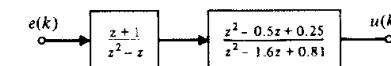


Figure 4.10
Block diagram of a cascade realization



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

$$\mathbf{A}_o = \begin{bmatrix} -a_1 & 1 & 0 \\ -a_2 & 0 & 1 \\ -a_3 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B}_o = \begin{bmatrix} b_1 - b_0 a_1 \\ b_2 - b_0 a_2 \\ b_3 - b_0 a_3 \end{bmatrix}$$

$$\mathbf{C}_o = [1 \ 0 \ 0]$$

$$\mathbf{D}_o = [b_0].$$

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

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

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_k -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

$$H(z) = \sum_{-\infty}^{\infty} u_k z^{-k} \triangleq \sum_{-\infty}^{\infty} h_k z^{-k}.$$

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}	$u_k \equiv h_k$
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

$$\begin{aligned} H(z) &= \sum_{k=0}^{\infty} T z^{-k} - \frac{T}{2} \\ &= \frac{T}{1-z^{-1}} - \frac{T}{2} \quad (1 < |z|) \\ &= \frac{2T - T(1-z^{-1})}{2(1-z^{-1})} \\ &= \frac{T + Tz^{-1}}{2(1-z^{-1})} \\ &= \frac{Tz+1}{2z-1} \quad (1 < |z|). \end{aligned} \quad (4.30)$$

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

$$\begin{aligned} &e_0 + e_1 z^{-1} + e_2 z^{-2} + e_3 z^{-3} + \dots \\ &h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} + \dots \\ &e_0 h_0 + e_1 h_0 z^{-1} + e_2 h_0 z^{-2} + e_3 h_0 z^{-3} \\ &\quad + e_0 h_1 z^{-1} + e_1 h_1 z^{-2} + e_2 h_1 z^{-3} \\ &\quad + e_0 h_2 z^{-2} + e_1 h_2 z^{-3} \\ &\quad + e_0 h_3 z^{-3} \\ &e_0 h_0 + (e_0 h_1 + e_1 h_0) z^{-1} + (e_0 h_2 + e_1 h_1 + e_2 h_0) z^{-2} + (e_0 h_3 + e_1 h_2 + e_2 h_1 + e_3 h_0) z^{-3} + \dots \end{aligned}$$

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

$$\begin{aligned} u_0 &= e_0 h_0 \\ u_1 &= e_0 h_1 + e_1 h_0 \\ u_2 &= e_0 h_2 + e_1 h_1 + e_2 h_0 \\ u_3 &= e_0 h_3 + e_1 h_2 + e_2 h_1 + e_3 h_0. \end{aligned}$$

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."

1. 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 $\alpha u_1 + \beta u_2$.

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

$$\begin{aligned} e &= e_j, \quad k = l \\ &= 0, \quad k \neq l. \end{aligned}$$

Finally, by linearity again, the total response at time k to a sequence of these pulses is the *sum* of the responses, namely,

$$u_k = e_0 h_k + e_1 h_{k-1} + \cdots + e_l h_{k-l} + \cdots + e_0 h_0,$$

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

$$\leq \sum_{-\infty}^{\infty} |e_l| |h_{k-l}|.$$

But, because we assume Eq. (4.33), this result is in turn bounded by

$$\leq M \sum_{-\infty}^{\infty} |h_{k-l}|. \quad (4.34)$$

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

$$\begin{aligned} u_0 &= b_0, \quad u_1 = a_1 b_0, \quad u_2 = a_1^2 b_0, \quad \dots \\ u_k &= b_0 a_1^k, \quad k \geq 0. \end{aligned} \quad (4.39)$$

Applying the test, we have

$$\begin{aligned} \sum_{-\infty}^{\infty} |h_l| &= \sum_{l=0}^{\infty} b_0 |a_l| = b_0 \frac{1}{1 - |a|} & (|a| < 1) \\ &= \text{unbounded} & (|a| \geq 1). \end{aligned}$$

Thus we conclude that the system described by this equation is BIBO stable if $|a| < 1$, 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

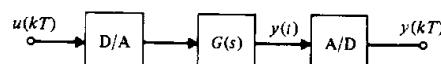


Figure 4.12
The prototype
sampled-data system

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 $y_1(t)$. 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

$$Y_1(s) = (1 - e^{-Ts}) \frac{G(s)}{s}, \quad (4.40)$$

and the required transfer function is the z -transform of the samples of the inverse of $Y_1(s)$, which can be expressed as

$$\begin{aligned} G(z) &= \mathcal{Z}\{Y_1(kT)\} \\ &= \mathcal{Z}\{\mathcal{L}^{-1}\{Y_1(s)\}\} \hat{=} \mathcal{Z}\{Y_1(s)\} \\ &= \mathcal{Z}\{(1 - e^{-Ts}) \frac{G(s)}{s}\}. \end{aligned}$$

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

$$G(z) = (1 - z^{-1}) \mathcal{Z}\left\{\frac{G(s)}{s}\right\}. \quad (4.41)$$

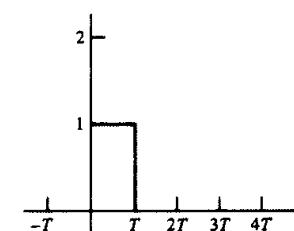


Figure 4.13
D/A output for
unit-pulse input

◆ 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 as

$$\begin{aligned} G(z) &= \frac{z-1}{z} \frac{z(1-e^{-aT})}{(z-1)(z-e^{-aT})} \\ &= \frac{1-e^{-aT}}{z-e^{-aT}}. \end{aligned} \quad (4.42)$$

◆ 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 $1/s^3$ 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

$$\frac{G(s)}{s} = e^{-\ell T s} \frac{e^{mT s}}{s} H(s).$$

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

$$G(z) = \frac{z-1}{z^{\ell+1}} \mathcal{Z} \left\{ \frac{e^{mT s}}{s} \right\} - \frac{e^{mT s}}{s+a}.$$

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.

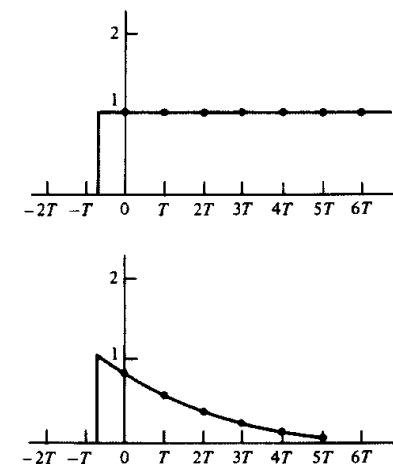
The samples are given by $1(kT)$ and $e^{-aT(\ell+m)} 1(kT)$. The corresponding z -transforms are $z/(z-1)$ and $ze^{-amT}/(z - e^{-aT})$. Consequently the final transfer function is

$$\begin{aligned} G(z) &= \frac{z-1}{z} \frac{1}{z^\ell} \left\{ \frac{z}{z-1} - \frac{ze^{-amT}}{z - e^{-aT}} \right\} \\ &= \frac{z-1}{z^\ell} \left\{ \frac{z[z - e^{-aT} - (z-1)e^{-amT}]}{(z-1)(z - e^{-aT})} \right\} \\ &= (1 - e^{-amT}) \frac{z + \alpha}{z^\ell (z - e^{-aT})} \end{aligned}$$

where the zero position is at $-\alpha = -(e^{-amT} - e^{-aT})/(1 - e^{-amT})$. 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

$$G(z) = \frac{z + 0.6025}{z^2(z - 0.3679)}. \quad (4.44)$$

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(a,[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

$$\dot{x} = Fx + Gu + G_1 w. \quad (4.45)$$

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

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_F \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_G u,$$

$$\theta = y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_H \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$
(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.

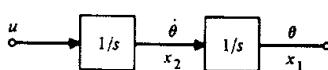
If we let $\xi = Tx$ in Eqs. (4.45) and (4.46), we find

$$\begin{aligned} \dot{\xi} &= T\dot{x} = T(Fx + Gu + G_1w) \\ &= TFx + TGu + TG_1w, \\ \dot{\xi} &= TFT^{-1}\xi + TGu + TG_1w, \\ y &= T^{-1}\xi + Ju. \end{aligned}$$

If we designate the system matrices for the new state ξ as A , B , C , and D , then

$$\dot{\xi} = A\xi + Bu + B_1w, \quad y = C\xi + Du,$$

Figure 4.15
Satellite attitude control
in classical
representation



where

$$A = TFT^{-1}, \quad B = TG, \quad B_1 = TG_1, \quad C = T^{-1}, \quad D = J.$$

◆ 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

$$A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = [0 \ 1].$$

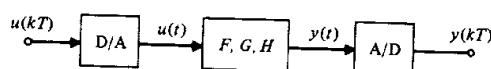
Most often, a change of state is made to bring the description matrices into a useful canonical form. We saw earlier how a single high-order difference equation could be represented by a state description in control or in observer canonical form. Also, there is a very useful state description corresponding 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 $y(k)$, 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

$$\dot{\mathbf{x}}_h = F\mathbf{x}_h(t), \quad \mathbf{x}_h(t_0) = \mathbf{x}_0. \quad (4.48)$$

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

$$\mathbf{x}_h(t) = \mathbf{A}_0 + \mathbf{A}_1(t - t_0) + \mathbf{A}_2(t - t_0)^2 + \dots \quad (4.49)$$

If we let $t = t_0$, we find immediately that $\mathbf{A}_0 = \mathbf{x}_0$. If we differentiate Eq. (4.49) and substitute into Eq. (4.48), we have

$$\mathbf{A}_1 + 2\mathbf{A}_2(t - t_0) + 3\mathbf{A}_3(t - t_0)^2 + \dots = \mathbf{F}\mathbf{x}_h$$

and, at $t = t_0$, $\mathbf{A}_1 = \mathbf{F}\mathbf{x}_0$. Now we continue to differentiate the series and the differential equation and equate them at t_0 to arrive at the series

$$\mathbf{x}_h(t) = \left[\mathbf{I} + \mathbf{F}(t - t_0) + \frac{\mathbf{F}^2(t - t_0)^2}{2!} + \frac{\mathbf{F}^3(t - t_0)^3}{3!} + \dots \right] \mathbf{x}_0.$$

This series is defined as the matrix exponential and written

$$\mathbf{x}_h(t) = e^{\mathbf{F}(t-t_0)} \mathbf{x}(t_0), \quad (4.50)$$

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

$$\mathbf{F}e^{\mathbf{F}(t-t_0)} v + e^{\mathbf{F}(t-t_0)} v = \mathbf{F}e^{\mathbf{F}(t-t_0)} v + \mathbf{G}u,$$

and, using the fact that the inverse is found by changing the sign of the exponent, we can solve for v as

$$\dot{v}(t) = e^{-\mathbf{F}(t-t_0)} \mathbf{G}u(t).$$

Assuming that the control $u(t)$ is zero for $t < t_0$, we can integrate \dot{v} from t_0 to t to obtain

$$v(t) = \int_{t_0}^t e^{-\mathbf{F}(\tau-t_0)} \mathbf{G}u(\tau) d\tau.$$

Hence, from Eq. (4.53), we get

$$\mathbf{x}_p(t) = e^{\mathbf{F}(t-t_0)} \int_{t_0}^t e^{-\mathbf{F}(\tau-t_0)} \mathbf{G}u(\tau) d\tau,$$

and simplifying, using the results of Eq. (4.52), we obtain the particular solution (convolution)

$$\mathbf{x}_p(t) = \int_{t_0}^t e^{\mathbf{F}(t-\tau)} \mathbf{G}u(\tau) d\tau. \quad (4.54)$$

The total solution for $w = 0$ and $u \neq 0$ is the sum of Eqs. (4.50) and (4.54):

$$\mathbf{x}(t) = e^{\mathbf{F}(t-t_0)} \mathbf{x}(t_0) + \int_{t_0}^t e^{\mathbf{F}(t-\tau)} \mathbf{G}u(\tau) d\tau. \quad (4.55)$$

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):

$$\mathbf{x}(kT + T) = e^{\mathbf{FT}} \mathbf{x}(kT) + \int_{kT}^{kT+T} e^{\mathbf{F}(kT+\tau-t)} \mathbf{G} u(\tau) d\tau. \quad (4.56)$$

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

$$\mathbf{x}(kT + T) = e^{\mathbf{FT}} \mathbf{x}(kT) + \int_0^T e^{\mathbf{F}\eta} d\eta \mathbf{G} u(kT). \quad (4.57)$$

If we define

$$\begin{aligned} \Phi &= e^{\mathbf{FT}} \\ \Gamma &= \int_0^T e^{\mathbf{F}\eta} d\eta \mathbf{G}, \end{aligned} \quad (4.58)$$

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

$$\Phi = e^{\mathbf{FT}} = \mathbf{I} + \mathbf{FT} + \frac{\mathbf{F}^2 T^2}{2!} + \frac{\mathbf{F}^3 T^3}{3!} + \dots$$

¹² If $w(\tau)$ 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

$$\Psi = \mathbf{I} + \frac{\mathbf{FT}}{2!} + \frac{\mathbf{F}^2 T^2}{3!} + \dots$$

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

$$\begin{aligned} [z\mathbf{I} - \Phi]\mathbf{X} &= \Gamma U(z), \\ Y(z) &= \mathbf{H}\mathbf{X}(z) \end{aligned} \quad (4.63)$$

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

$$\frac{Y(z)}{U(z)} = \mathbf{H}[z\mathbf{I} - \Phi]^{-1}\Gamma. \quad (4.64)$$

◆ 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 \mathbf{F} and \mathbf{G} defined in Eq. (4.47). Since $\mathbf{F}^2 = 0$ in this case, we have

$$\begin{aligned} \Phi &= \mathbf{I} + \mathbf{FT} + \frac{\mathbf{F}^2 T^2}{2!} + \dots \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} T = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \\ \Gamma &= \left[\mathbf{IT} + \mathbf{F} \frac{T^2}{2!} + \frac{\mathbf{F}^2 T^3}{3!} \right] \mathbf{G} \\ &= \left\{ \begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \frac{T^2}{2} \right\} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}. \end{aligned}$$

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 $\det(z\mathbf{I} - \Phi)$, 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

$$[z\mathbf{I} - \Phi]\mathbf{X}(z) = \mathbf{0}$$

have a nontrivial solution. From matrix algebra the well-known requirement for this is that $\det(z\mathbf{I} - \Phi) = 0$. Using the Φ from the previous example, we have

$$\begin{aligned} \det[z\mathbf{I} - \Phi] &= \det \left[\begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \right] \\ &= \det \begin{bmatrix} z-1 & -T \\ 0 & z-1 \end{bmatrix} \\ &= (z-1)^2 = 0, \end{aligned}$$

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

$$\begin{bmatrix} z\mathbf{I} - \Phi & -\Gamma \\ \mathbf{H} & 0 \end{bmatrix} \begin{bmatrix} X(z) \\ U(z) \end{bmatrix} = \mathbf{0}. \quad (4.65)$$

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.

$$\begin{aligned} &= +\frac{T^2}{2}z + \frac{T^2}{2} \\ &= +\frac{T^2}{2}(z+1). \end{aligned}$$

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

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{F}\mathbf{x}(t) + \mathbf{G}u(t - \lambda), \\ \mathbf{y} &= \mathbf{H}\mathbf{x}. \end{aligned} \quad (4.66)$$

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

$$\mathbf{x}(kT + T) = e^{\mathbf{FT}}\mathbf{x}(kT) + \int_{kT}^{kT+T} e^{\mathbf{F}(kT+\tau-\lambda)}\mathbf{G}u(\tau - \lambda) d\tau.$$

If we substitute $\eta = kT + T - \tau$ for τ in the integral, we find a modification of Eq. (4.57)

$$\begin{aligned} \mathbf{x}(kT + T) &= e^{\mathbf{FT}}\mathbf{x}(kT) + \int_T^0 e^{\mathbf{F}\eta}\mathbf{G}u(kT + T - \lambda - \eta)(-d\eta) \\ &= e^{\mathbf{FT}}\mathbf{x}(kT) + \int_0^T e^{\mathbf{F}\eta}\mathbf{G}u(kT + T - \lambda - \eta) d\eta. \end{aligned}$$

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

$$\begin{array}{ccl} \ell & \geq & 0, \\ 0 & \leq m & < 1. \end{array}$$

With this substitution, we find that the discrete system is described by

$$\mathbf{x}(kT + T) = e^{\mathbf{FT}}\mathbf{x}(kT) + \int_0^T e^{\mathbf{F}\eta}\mathbf{G}u(kT + T - \ell T + mT - \eta) d\eta. \quad (4.68)$$

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 $kT - \ell T + mT$ backward to $kT - \ell T + T$. 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

$$\begin{aligned} \mathbf{x}(kT + T) &= e^{\mathbf{FT}}\mathbf{x}(kT) + \int_0^{mT} e^{\mathbf{F}\eta}\mathbf{G}d\eta u(kT - \ell T + T) \\ &\quad + \int_{mT}^T e^{\mathbf{F}\eta}\mathbf{G}d\eta u(kT - \ell T) \\ &= \Phi\mathbf{x}(kT) + \Gamma_1 u(kT - \ell T) + \Gamma_2 u(kT - \ell T + T). \end{aligned} \quad (4.69)$$

In Eq. (4.69) we defined

$$\Phi = e^{\mathbf{FT}}, \quad \Gamma_1 = \int_{mT}^T e^{\mathbf{F}\eta}\mathbf{G}d\eta, \quad \text{and} \quad \Gamma_2 = \int_0^{mT} e^{\mathbf{F}\eta}\mathbf{G}d\eta. \quad (4.70)$$

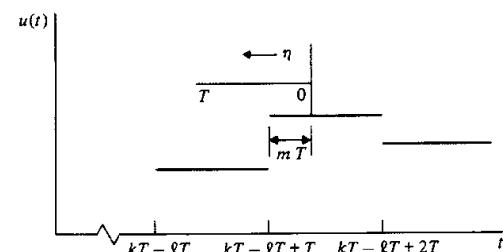


Figure 4.18
Sketch of a piecewise input and time axis for a system with time delay

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

$$\begin{aligned} \Psi(a) &= \frac{1}{a} \int_0^a \sum_{k=0}^{\infty} \frac{\mathbf{F}^k \sigma^k}{k!} d\sigma \\ &= \frac{1}{a} \sum_{k=0}^{\infty} \frac{\mathbf{F}^k}{k!} \frac{a^{k-1}}{k+1} \end{aligned}$$

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

$$\begin{aligned} y(k) &= \mathbf{H}\mathbf{x}(k) \\ &= \mathbf{H}[\xi(k) + \Gamma_2 u(k)] \\ &= \mathbf{H}\xi(k) + \mathbf{H}\Gamma_2 u(k) \\ &= \mathbf{H}_d \xi(k) + J_d u(k). \end{aligned} \quad (4.77)$$

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

$$\begin{bmatrix} \mathbf{x}(k+1) \\ x_{n+1}(k+1) \\ x_{n+2}(k+1) \\ \vdots \\ x_{n+\ell}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \Gamma_1 & \Gamma_2 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ x_{n+1}(k) \\ x_{n+2}(k) \\ \vdots \\ x_{n+\ell}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u(k)$$

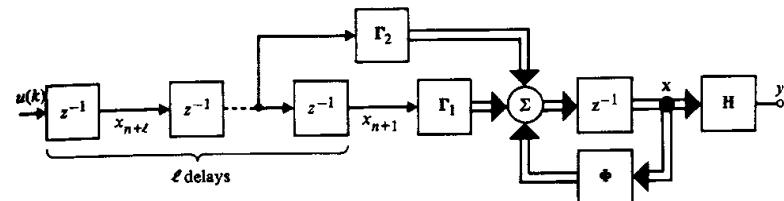
$$y(k) = [\mathbf{H} \ 0 \ \cdots \ 0] \begin{bmatrix} \mathbf{x}(k) \\ x_{n+1}(k) \\ \vdots \\ x_{n+\ell}(k) \end{bmatrix}. \quad (4.79)$$

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 \mathbf{FT} is large, then $(\mathbf{FT})^N/N!$ 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

$$(e^{\mathbf{FT}})^2 = e^{\mathbf{FT}} e^{\mathbf{FT}} = e^{2\mathbf{FT}}. \quad (4.80)$$

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

$$\Psi \left(\frac{T}{2^k} \right) = \sum_{j=0}^{N-1} \frac{(\mathbf{FT}/2^k)^j}{(j+1)!} + \sum_{j=N}^{\infty} \frac{(\mathbf{FT}/2^k)^j}{(j+1)!} = \hat{\Psi} + \mathbf{R}.$$

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 $\hat{\Psi}$, namely,

$$\hat{\mathbf{R}} \cong (\mathbf{FT})^N / (N+1)! 2^{Nk}.$$

A simpler method is to select k such that the size of \mathbf{FT} divided by 2^k is less than 1. In this case, the series for $\mathbf{FT}/2^k$ will surely converge. The rule is to select k such that

$$2^k > \|\mathbf{FT}\| = \max_j \sum_{i=1}^n |F_{ij}| T.$$

Taking the log of both sides, we find

$$k > \log_2 \|\mathbf{FT}\|,$$

from which we select

$$k = \max(\lceil \log_2 \|\mathbf{FT}\| \rceil, 0). \quad (4.81)$$

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 $\| \mathbf{F}T \|$ 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

$$\begin{aligned}\Phi(2T) &= \Phi(T)\Phi(T), \\ \mathbf{I} + 2T\mathbf{F}\Psi(2T) &= [\mathbf{I} + T\mathbf{F}\Psi(T)][\mathbf{I} + T\mathbf{F}\Psi(T)] \\ &= \mathbf{I} + 2T\mathbf{F}\Psi(T) + T^2\mathbf{F}^2\Psi^2(T);\end{aligned}$$

therefore

$$2T\mathbf{F}\Psi(2T) = 2T\mathbf{F}\Psi(T) + T^2\mathbf{F}^2\Psi^2(T).$$

This is equivalent to

$$\Psi(2T) = \left(\mathbf{I} + \frac{T\mathbf{F}}{2} \Psi(T) \right) \Psi(T),$$

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 \mathbf{F} and T .
2. Comment: Compute $\| \mathbf{F}T \|$.
3. $V \leftarrow \max_j \{ \sum_i | F_{ij} | \} \times T$
4. $k \leftarrow$ smallest nonnegative integer greater than $\log_2 V$.
5. Comment: compute $\Psi(T/2^k)$.
6. $T_1 \leftarrow T/2^k$
7. $\mathbf{I} \leftarrow$ Identity
8. $\Psi \leftarrow \mathbf{I}$
9. $j \leftarrow 11$
10. If $j = 1$, go to step 14.
11. $\Psi \leftarrow \mathbf{I} + \frac{\mathbf{F}T_1}{j} \Psi$
12. $j \leftarrow j - 1$
13. Go to step 10.
14. Comment: Now double Ψk times.
15. If $k = 0$, stop.
16. $\Psi \leftarrow (\mathbf{I} + \frac{\mathbf{F}T}{2^{k-1}} \Psi) \Psi$
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:

$$f_{i,\mathbf{x}} \triangleq \left(\frac{\partial f_1}{\partial x_1} \dots \frac{\partial f_1}{\partial x_n} \right). \quad (4.88)$$

The row vector in Eq. (4.88) is called the **gradient** of the scalar f_i 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 **Jacobeans**) 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

$$\dot{\mathbf{x}} = \mathbf{Fx} + \mathbf{Gu}, \quad \mathbf{y} = \mathbf{Hx} + \mathbf{Ju}. \quad (4.91)$$

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

$$\begin{aligned} \mathbf{x}(k+1) &= \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k), \\ \mathbf{y}(k) &= \mathbf{H}\mathbf{x}(k) + \mathbf{Ju}(k), \end{aligned} \quad (4.92)$$

from which the transfer function is

$$G(z) = Y(z)/U(z) = \mathbf{H}(z\mathbf{I} - \Phi)^{-1}\Gamma + J. \quad (4.93)$$

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 1.0.

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

$$\begin{aligned} e_2(k) &= 1 & (k \geq 0) \\ &= 0 & (k < 0) \\ &\equiv 1(k). \end{aligned}$$

In this case, the z -transform is

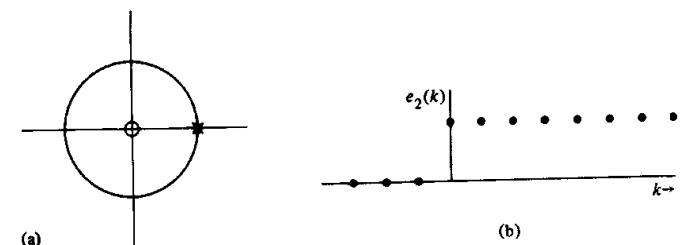
$$\begin{aligned} E_2(z) &= \sum_{k=-\infty}^{\infty} e_2(k) z^{-k} = \sum_{k=0}^{\infty} z^{-k} \\ &= \frac{1}{1 - z^{-1}} & (|z^{-1}| < 1) \\ &= \frac{z}{z - 1} & (|z| > 1). \end{aligned} \quad (4.95)$$

¹⁸ We have shifted notation here to use $e(k)$ rather than e_k 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 $E_2(z)$ marked \times and the zero marked \circ . Beside the z -plane, we sketch the time plot of $e_2(k)$.

Figure 4.21
(a) Pole and zero of $E_2(z)$ in the z -plane. The unit circle is shown for reference. (b) Plot of $e_2(k)$



4.4.3 Exponential

The one-sided exponential in time is

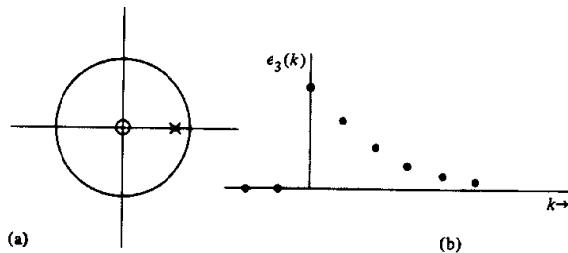
$$\begin{aligned} e_3(k) &= r^k & (k \geq 0) \\ &= 0 & (k < 0) \end{aligned} \quad (4.96)$$

which is the same as $r^k 1(k)$, using the symbol $1(k)$ for the unit step function. Now we get

$$\begin{aligned} E_3(z) &= \sum_{k=0}^{\infty} r^k z^{-k} \\ &= \sum_{k=0}^{\infty} (rz^{-1})^k \\ &= \frac{1}{1 - rz^{-1}} & (|rz^{-1}| < 1) \\ &= \frac{z}{z - r} & (|z| > |r|). \end{aligned} \quad (4.97)$$

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

$$e_4(k) = r^k \left(\frac{e^{jk\theta} + e^{-jk\theta}}{2} \right) 1(k),$$

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

$$e_5(k) = r^k e^{jk\theta} 1(k) \quad (4.98)$$

and compute

$$\begin{aligned} E_5(z) &= \sum_{k=0}^{\infty} r^k e^{jk\theta} z^{-k} \\ &= \sum_{k=0}^{\infty} (re^{j\theta} z^{-1})^k \end{aligned}$$

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 $e_5(k)$ 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).

$$\mathcal{Z}\{r^k e^{-jk\theta} 1(k)\} = \frac{z}{z - re^{-j\theta}} \quad (|z| > r), \quad (4.100)$$

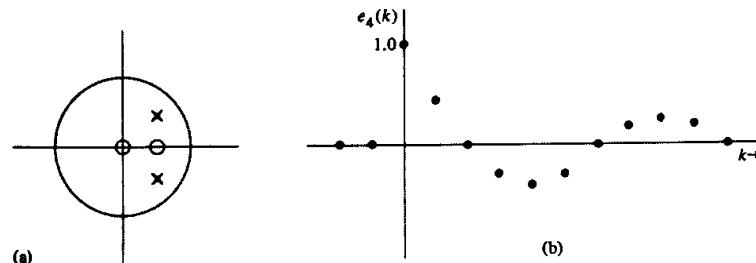
and thus that

$$\begin{aligned} E_4(z) &= \frac{1}{2} \left\{ \frac{z}{z - re^{j\theta}} + \frac{z}{z - re^{-j\theta}} \right\} \\ &= \frac{z(z - r \cos \theta)}{z^2 - 2r(\cos \theta)z + r^2} \quad (|z| > r). \end{aligned} \quad (4.101)$$

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_2 , 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 $e_4(k)$



1. 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.

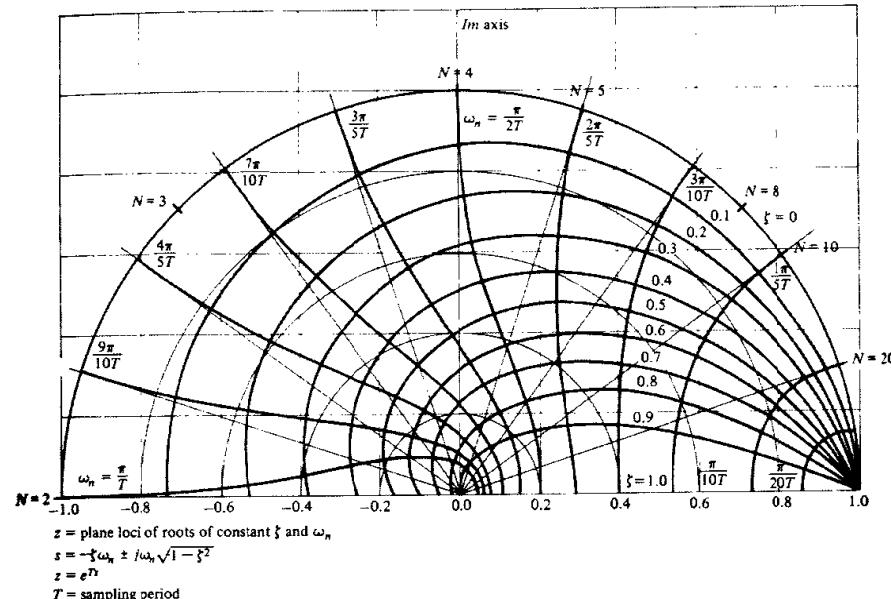
2. The number of samples per oscillation of a sinusoidal signal is determined by θ . If we require $\cos(\theta k) = \cos(\theta(k + N))$, we find that a period of 2π rad contains N samples, where

$$N = \frac{2\pi}{\theta} \Big|_{\text{rad}} = \frac{360}{\theta} \Big|_{\text{deg}} \quad \text{samples/cycle.}$$

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 $N = f_s/f$ 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

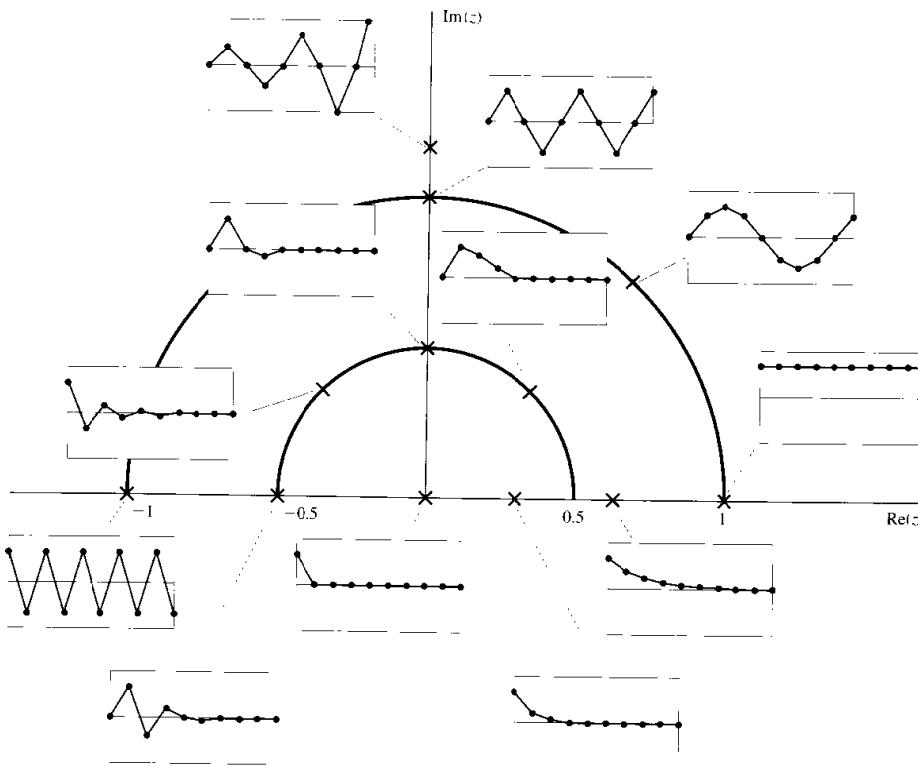
$$y(t) = e^{-at} \cos bt \quad 1(t) \quad (4.102)$$

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

$$r = e^{-aT}, \quad \theta = bT$$

$$z_{1,2} = e^{s_1 T}, e^{s_2 T}.$$

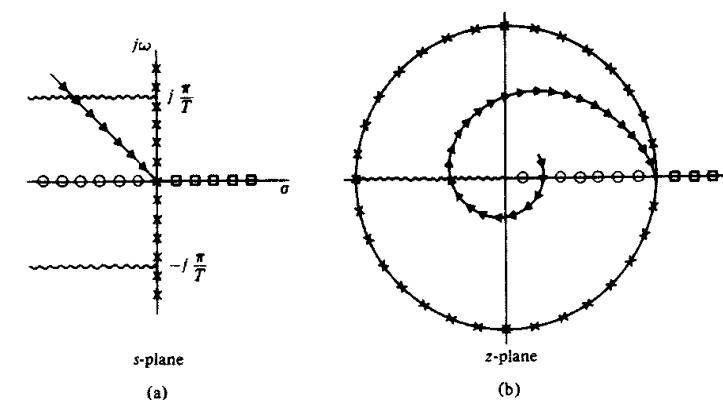
If $E(z)$ is a ratio of polynomials in z , which will be the case if $e(k)$ is generated by a linear difference equation with constant coefficients, then by partial fraction expansion, $E(z)$ can be expressed as a sum of elementary terms like E_4 and E_3 .²⁰ In all such cases, the discrete signal can be generated by samples from continuous signals where the relation between the s-plane poles and the corresponding z-plane poles is given by

$$z = e^{sT}. \quad (4.103)$$

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
 $z = e^{sT}$



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 $r^k e^{jk\theta}$.

Table 4.2

Description of corresponding lines in s -plane and z -plane		
s -plane	Symbol	z -plane
$\{ s = j\omega$	$\times \times \times$	$\{ z = 1$
Real frequency axis		Unit circle
$s = \sigma \geq 0$	$\square \square \square$	$z = r \geq 1$
$s = \sigma \leq 0$	$\circ \circ \circ$	$z = r, 0 \leq r \leq 1$
$\left\{ \begin{array}{l} s = -\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2}, \\ = -a + jb \end{array} \right.$	$\triangle \triangle \triangle$	$\left\{ \begin{array}{l} z = re^{j\theta} \text{ where } r = \exp(-\zeta\omega_n T) \\ \theta = \omega_n T \sqrt{1-\zeta^2} \\ \text{Logarithmic spiral} \end{array} \right.$
Constant damping ratio if ζ is fixed and ω_n varies		
$s = \pm j(\pi/T) + \sigma,$	$\sigma \leq 0$	$z = -r$

$z = e^{jT}$ of Eq. (4.103) is many-to-one. There are many values of s for each value of z . In fact, if

$$s_2 = s_1 + j \frac{2\pi}{T} N,$$

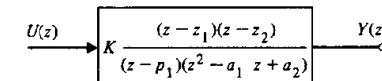
then $e^{s_1 T} = e^{s_2 T}$. The (great) significance of this fact will be explored in Chapter 5.

Lines of constant damping in the s -plane are mapped into the z -plane according to Eq. (4.103) for several values of ζ 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

$$Y(z) = \frac{z - r \cos \theta}{z^2 - 2r \cos \theta z + r^2}. \quad (4.104)$$

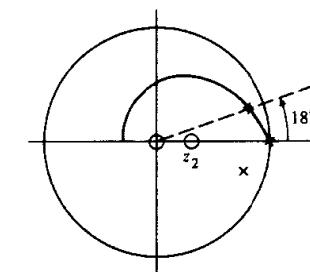
This transform, when compared with the transform $E_4(z)$ given in Eq. (4.101), is seen to be

$$Y(z) = z^{-1} E_4(z),$$

and we conclude that under these circumstances the system pulse response is a delayed version of $e_4(k)$, 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 $\zeta = 0.5$



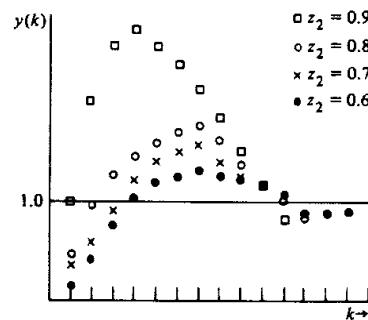
The major effect of the zero z_2 on the step response $y(k)$ is to change the percent overshoot, as can be seen from the four step responses for this case plotted in Fig. 4.29. To summarize all these data, we plot the percent overshoot versus zero location in Fig. 4.30 for $\zeta = 0.5$ and in Fig. 4.31, 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 $z_1 = z_2 = -1$ 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.

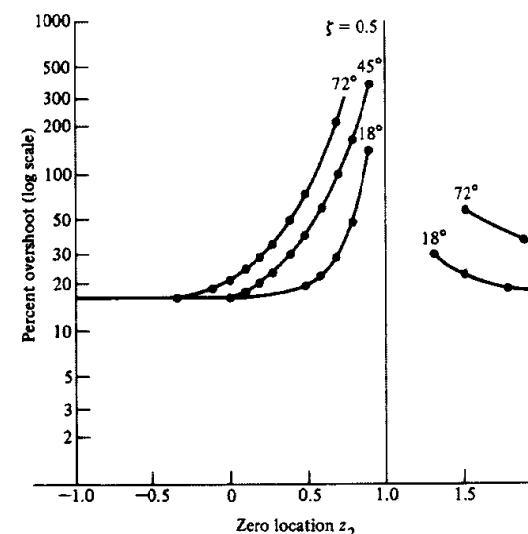
The understanding of how poles and zeros affect the time response is very useful for the control system designer. The knowledge helps guide the iterative

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

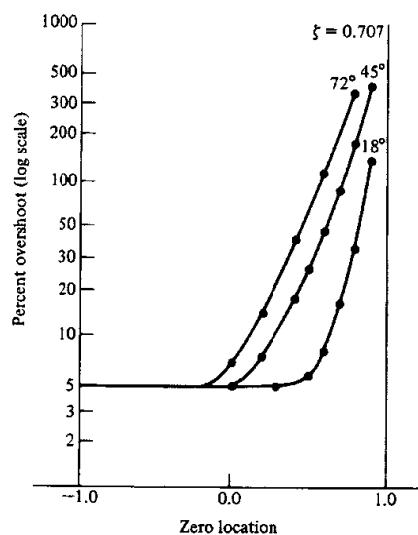
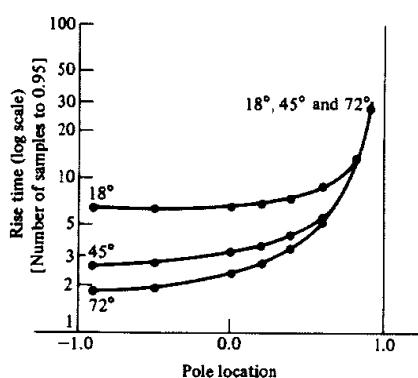


Figure 4.32
Effects of an extra pole
on rise time for a
discrete third-order
system, two zeros at -1 ,
one zero at ∞ : $\zeta =$
0.5; $\theta = 18^\circ, 45^\circ, 72^\circ$



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 $H(e^{j\omega T}) = A(\omega_o T)e^{j\psi(\omega_o T)}$. 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

$$\begin{aligned} U_{ss}(kT) &= \frac{A}{2} e^{j\psi} e^{j\omega_o T k} + \frac{A}{2} e^{-j\psi} e^{-j\omega_o T k} \\ &= A \cos(\omega_o T k + \psi), \end{aligned} \quad (4.109)$$

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

	Time	Frequency
Fourier Series	periodic	discrete
<i>z</i> -transform	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

$$\sum_{n=0}^{N-1} \left\{ \sum_{\ell=0}^{N-1} f_\ell e^{-j2\pi(n\ell)/N} \right\} e^{j2\pi(nk)/N}.$$

Interchanging the order of the summations gives

$$\sum_{\ell=0}^{N-1} f_\ell \left\{ \sum_{n=0}^{N-1} e^{j2\pi(n(k-\ell))/N} \right\}.$$

The sum in the braces is a finite geometric series, which we can evaluate as follows

$$\begin{aligned} \sum_{n=0}^{N-1} e^{j2\pi(n(k-\ell))/N} &= \frac{1 - e^{j2\pi(k-\ell)}}{1 - e^{j2\pi(k-\ell)/N}} \\ &= \begin{cases} N & k - \ell = 0 \\ 0 & k - \ell = 1, 2, \dots, N-1. \end{cases} \end{aligned}$$

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 $\omega_i = 2\pi\ell/NT$ so that $e(kT) = A \sin(2\pi\ell k T / NT)$. 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

$$\begin{aligned} E_n &= \sum_{k=0}^{N-1} A \sin\left(\frac{2\pi\ell k}{N}\right) e^{-j(2\pi n k)/N} \\ &= \sum_{k=0}^{N-1} \frac{A}{2j} [e^{j(2\pi\ell k)/N} - e^{-j(2\pi\ell k)/N}] e^{-j(2\pi n k)/N} \\ &= \begin{cases} 0, & \ell \neq n \\ \frac{NA}{2j}, & \ell = n. \end{cases} \end{aligned}$$

The DFT of the output is

$$\begin{aligned} U_n &= \sum_{k=0}^{N-1} B \sin\left(\frac{2\pi\ell k}{N} + \psi\right) e^{-j(2\pi n k)/N} \\ &= \sum_{k=0}^{N-1} \frac{B}{2j} [e^{j\psi} e^{j(2\pi\ell k)/N} - e^{-j\psi} e^{-j(2\pi\ell k)/N}] e^{-j(2\pi n k)/N} \end{aligned}$$

$$= \begin{cases} 0, & \ell \neq n \\ \frac{NB}{2j} e^{j\psi}, & \ell = n. \end{cases}$$

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

$$H(e^{j(2\pi\ell)/N}) = \frac{U_\ell}{E_\ell},$$

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 $F_i(z) = \mathcal{Z}\{f_i(kT)\}$.

- 1. 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 $re^{\pm j\theta}$, then the scaled polynomial $\alpha^2 z^2 + a_1 \alpha z + a_2$ has roots $(r/\alpha)e^{\pm j\theta}$. 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

$$\lim_{k \rightarrow \infty} f(k) = \lim_{z \rightarrow 1} (z-1)F(z). \quad (4.115)$$

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

$$U(z) = \frac{z}{z-0.5} \frac{T}{2} \frac{z+1}{z-1}, \quad |z| > 1.$$

Because $U(z)$ satisfies the conditions of Eq. (4.115), we have

$$\begin{aligned} \lim_{k \rightarrow \infty} u(k) &= \lim_{z \rightarrow 1} (z-1) \frac{z}{z-0.5} \frac{T}{2} \frac{z+1}{z-1} \\ &= \lim_{z \rightarrow 1} \frac{z}{z-0.5} \frac{T}{2} (z+1) \end{aligned}$$

$$= \frac{1}{1 - 0.5} \frac{T}{2} (1 + 1) \\ = 2T.$$

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}

$$U(z) = \frac{T}{2} \frac{1 + z^{-1}}{1 - 1.5z^{-1} + 0.5z^{-2}}, \quad (4.119)$$

and divide as follows

$$\begin{array}{r} \frac{\frac{T}{2}(1 + 2.5z^{-1} + 3.25z^{-2} + 3.625z^{-3} + \dots)}{1 - 1.5z^{-1} + 0.5z^{-2}} \\ \frac{1 + z^{-1}}{1 - 1.5z^{-1} + 0.5z^{-2}} \\ \hline 2.5z^{-1} - 0.5z^{-2} \\ \frac{2.5z^{-1} - 3.75z^{-2} + 1.25z^{-3}}{3.25z^{-2} - 1.25z^{-3}} \\ \frac{3.25z^{-2} - 4.875z^{-3} + 1.625z^{-4}}{3.625z^{-3} - 1.625z^{-4}} \\ \hline 3.625z^{-3} - \dots \end{array}$$

By direct comparison with $U(z) = \sum_k u(k)z^{-k}$, we conclude that

$$\begin{aligned} u_0 &= T/2, \\ u_1 &= (T/2)2.5, \\ u_2 &= (T/2)3.25. \end{aligned} \quad (4.120)$$

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

$$A = \frac{T}{2} \frac{2}{0.5} = 2T.$$

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).



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_1 , then $(z - z_1)^n F(z)$ is regular at z_1 and can be expanded in a Taylor series near z_1 as

$$(z - z_1)^n F(z) = A_{-n} + A_{-n+1}(z - z_1) + \dots + A_{-1}(z - z_1)^{-1} + A_0(z - z_1)^n + \dots \quad (4.124)$$

The residue of $F(z)$ at z_1 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

$$\mathcal{I} = \frac{1}{2\pi j} \oint_C \sum_{l=-\infty}^{\infty} f(l) z^{-l} z^k \frac{dz}{z}.$$

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

$$\mathcal{I} = \frac{1}{2\pi j} \sum_{l=-\infty}^{\infty} f(l) \oint_C z^{k-l} \frac{dz}{z}.$$

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)

$$\begin{aligned} z^{-k} \frac{z^k}{z-1} &= \frac{-1}{z-1} \\ &= -(1 + z^{-1} + z^{-2} + \cdots + z^{-k} + \cdots). \end{aligned}$$

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

$$F_3(z) = \sum_{k=-\infty}^{\infty} f_1(k)z^{-k} \frac{1}{2\pi j} \oint_{C_1} F_2(\xi)\xi^k \frac{d\xi}{\xi}.$$

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 $F_2(\xi)$ and $F_1(z/\xi)$. 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 $e_2(k) = z^{k-j} = z^j z^k$ for some fixed j . From Eq. (4.131), if we let $\alpha = z^j$, it must follow that

$$\begin{aligned} u_2 &= z^j u(k) \\ &= z^j H(z, k)z^k \\ &= H(z, k)z^{k+j}. \end{aligned} \quad (4.133)$$

From Eq. (4.132), we must have

$$\begin{aligned} u_2(k) &= u(k+j) \\ &= H(z, j+k)z^{k+j} \quad \text{for all } j. \end{aligned} \quad (4.134)$$

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

$$\sum_{l=-\infty}^{\infty} |h_{k-l}| < \infty. \quad (4.35)$$

- 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

$$\begin{aligned} \Phi &= e^{F\tau} \\ \Gamma &= \int_0^\tau e^{F\eta} d\eta G, \end{aligned} \quad (4.58)$$

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_o , given by an amplitude, A , and phase, ψ , as

$$A = |H(e^{j\omega_o T})| \quad \text{and} \quad \psi = \angle(H(e^{j\omega_o T})) \quad (4.110)$$

which can be evaluated 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:

- $u(k) = 0.5u(k-1) - 0.3u(k-2)$
- $u(k) = 1.6u(k-1) - u(k-2)$
- $u(k) = 0.8u(k-1) + 0.4u(k-2)$

- 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 $t = kT - \tau$ 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

$$u(k+2) = 0.25u(k).$$

- Assume a solution $u(k) = A_1 z^k$ and find the characteristic equation in z .
- Find the characteristic roots z_1 and z_2 and decide if the equation solutions are stable or unstable.
- 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

$$z_{1,2} = re^{\pm j\theta}.$$

- 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) $\frac{v}{s(s+3)}$

(c) $\frac{3}{(s+1)(s+3)}$

(d) $\frac{(s+1)}{s^2}$

(e) $\frac{e^{T/2}}{s^2}$

(f) $\frac{(1-s)}{s^2}$

(g) $\frac{3e^{-1.5T}s}{(s+1)(s+3)}$

- (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-colocated 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 $\omega_p = 5$, $\omega_i = 3$, $\zeta_p = \zeta_i = 0$.

- (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 , J .

- (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 $\mathbf{x} = T\xi$.

- (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

$$F(z) = \sum_0^{\infty} f(k)z^{-k}.$$

- (a) Show that the one-sided transform of $f(k+1)$ is

$$Z\{f(k+1)\} = zF(z) - zf(0).$$

- (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 $u = Az^k$ and $e = Bz^k$ 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. (4.27).
- 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)** $F(z) = \frac{1}{1+z^2}, \quad |z| > 1;$
- (b)** $F(z) = \frac{z(z-1)}{z^2-1.2z+0.25}, \quad |z| > 1;$
- (c)** $F(z) = \frac{z}{z^2-2z+1}, \quad |z| > 1;$
- (d)** $F(z) = \frac{z}{(z-\frac{1}{2})(z-2)}, \quad 1/2 < |z| < 2.$

- 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

$$y(k) - 3y(k-1) + 2y(k-2) = 2u(k-1) - 2u(k-2).$$

$$u(k) = \begin{cases} k, & k \geq 0 \\ 0, & k < 0 \end{cases}$$

$$y(k) = 0, \quad k < 0.$$

- 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 $G(s)$ 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 is 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

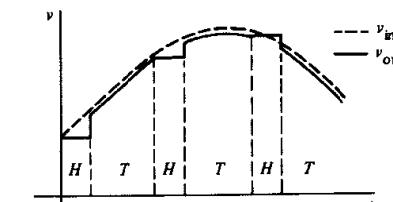
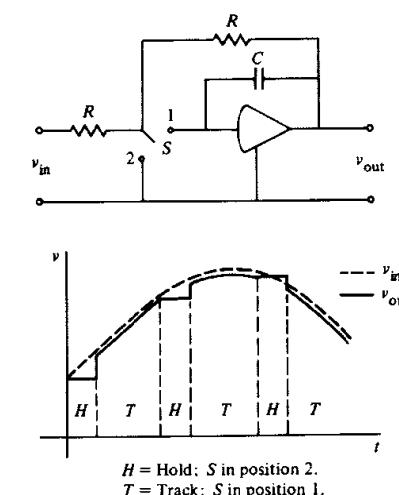
In this chapter, we introduce the analysis of the sampling process and describe both a time-domain and a frequency-domain representation. We also describe the companion process of data extrapolation or data holding to construct a continuous time signal from samples. As part of this analysis we show that 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



H = Hold; S in position 2.
 T = Track; S in position 1.

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

$$r(t) \xrightarrow{T} r^*(t) = \sum_{k=-\infty}^{\infty} r(kT) \delta(t - kT)$$

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

$$r^*(t) = \sum_{k=-\infty}^{\infty} r(kT) \delta(t - kT). \quad (5.1)$$

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

$$= \int_{-\infty}^{\infty} \sum_{k=-\infty}^{\infty} r(kT) \delta(\tau - kT) e^{-s\tau} d\tau,$$

and now, exchanging integration and summation and using Eq. (5.2), we have

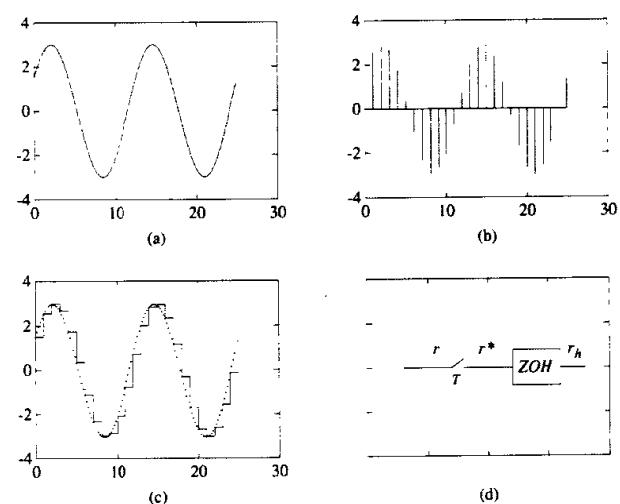
$$R^*(s) = \sum_{k=-\infty}^{\infty} r(kT) e^{-skT}. \quad (5.5)$$

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_i ;
(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

$$1^*(s) = \sum_{k=0}^{\infty} \delta(s - kT). \quad (a)$$

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,

$$\mathcal{L}\{r^*(t)\} = \int_{-\infty}^{\infty} r(t) \left\{ \frac{1}{T} \sum_{n=-\infty}^{\infty} e^{j\omega_s t} \right\} e^{-st} dt$$

and integrate the sum, term by term to obtain

$$R^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} r(t) e^{j\omega_s t} e^{-st} dt.$$

If we combine the exponentials in the integral, we get

$$R^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} r(t) e^{-(s-j\omega_s)t} dt.$$

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

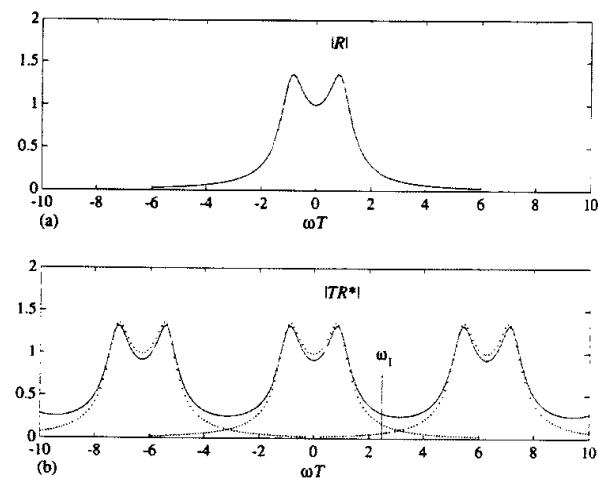
$$R^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} R(s - jn\omega_s), \quad (5.9)$$

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 $f_1 = \frac{1}{8}$, then

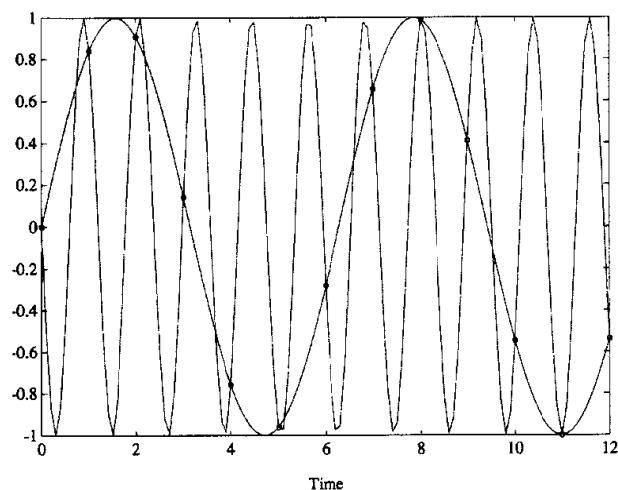
$$f_0 = \frac{1}{8} - 1 = -\frac{7}{8}.$$

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

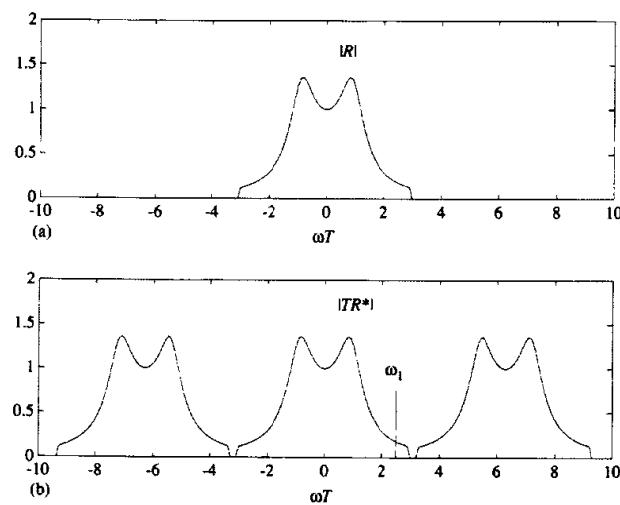
hidden oscillations

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

$$\begin{aligned} \ell(t) &= \frac{1}{2\pi} \int_{-\pi/T}^{\pi/T} T e^{j\omega t} d\omega \\ &= \frac{T}{2\pi} \frac{e^{j\omega t}}{jt} \Big|_{-\pi/T}^{\pi/T} \\ &= \frac{T}{2\pi jt} (e^{j(\pi t/T)} - e^{-j(\pi t/T)}) \\ &= \frac{\sin(\pi t/T)}{\pi t/T} \\ &\triangleq \text{sinc} \frac{\pi t}{T}. \end{aligned} \quad (5.11)$$

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 interpolators 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).

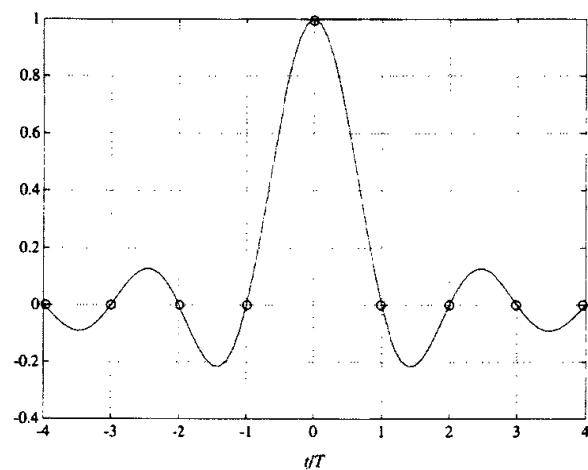
There is one serious drawback to the extrapolating signal given by Eq. (5.11). Because $\ell(t)$ is the impulse response of the ideal low-pass filter $L(j\omega)$, it follows that this filter is *noncausal* because $\ell(t)$ is nonzero for $t < 0$. $\ell(t)$ starts at $t = -\infty$ while the impulse 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, $e^{-j\omega\tau}$, 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,

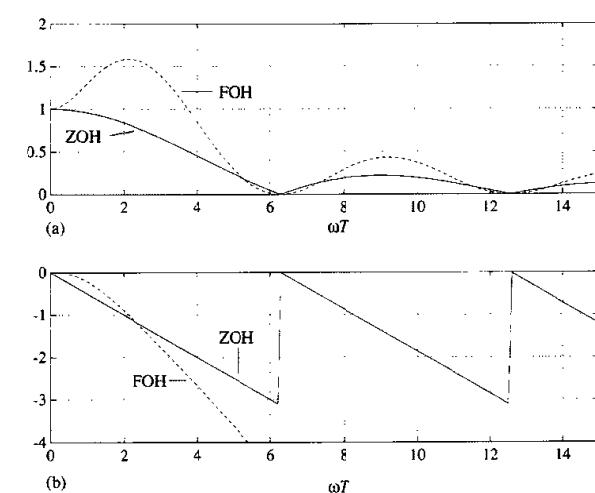
$$ZOH(j\omega) = e^{-j\omega T/2} T \text{sinc}(\omega T/2). \quad (5.14)$$

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



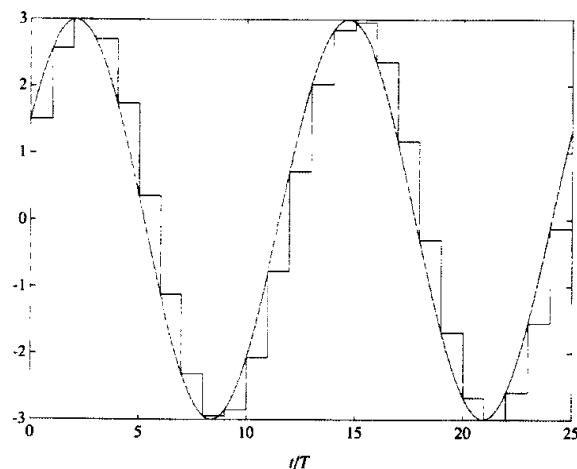
which slowly gets smaller as ω gets larger until it is zero for the first time at $\omega = \omega_s = 2\pi/T$. The phase is

$$\angle ZOH(j\omega) = \frac{-\omega T}{2}, \quad (5.16)$$

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 $\omega_s = 2\pi/T = 200\pi$ 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

$$R(j\omega) = \pi A [e^{j\phi} \delta(\omega - \omega_o) + e^{-j\phi} \delta(\omega + \omega_o)].$$

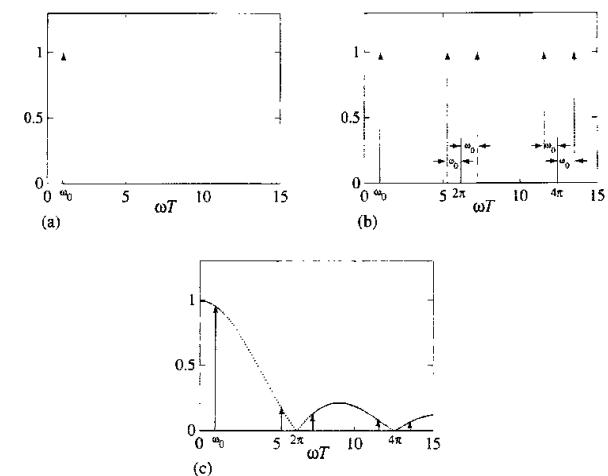
A sketch of this spectrum is shown in Fig. 5.10(a) for $A = 1/\pi$. We represent the impulses by arrows whose heights are proportional to the intensities of the impulses.

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



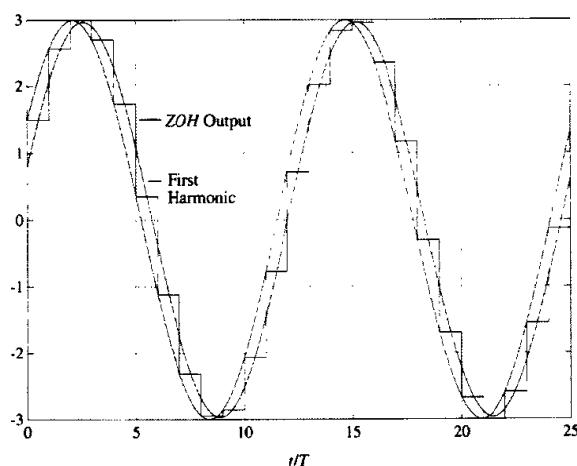
Thus the spectrum of R_h is also a sum of an infinite number of terms, but now with intensities modified by the sinc function and phases shifted by the delay function $\omega T/2$. These intensities are plotted in Fig. 5.10(c). Naturally, when all the harmonics included in R_h are converted to their 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

$$v_1(t) = A[\text{sinc}(\omega T/2)] \sin[\omega_o(t - \frac{T}{2})]. \quad (5.17)$$

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



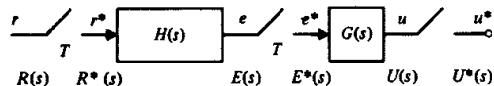
5.4 Block-Diagram Analysis of Sampled-Data Systems

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

$$U^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s - jn\omega_s)G(s - jn\omega_s); \quad (5.20)$$

but $E^*(s)$ is

$$E^*(s) = \frac{1}{T} \sum_{k=-\infty}^{\infty} E(s - jk\omega_s),$$

so that

$$E^*(s - jn\omega_s) = \frac{1}{T} \sum_{\ell=-\infty}^{\infty} E(s - jk\omega_s - jn\omega_s). \quad (5.21)$$

Now in Eq. (5.21) we can let $k = \ell - n$ to get

$$\begin{aligned} E^*(s - jn\omega_s) &= \frac{1}{T} \sum_{\ell=-\infty}^{\infty} E(s - j\ell\omega_s) \\ &= E^*(s). \end{aligned} \quad (5.22)$$

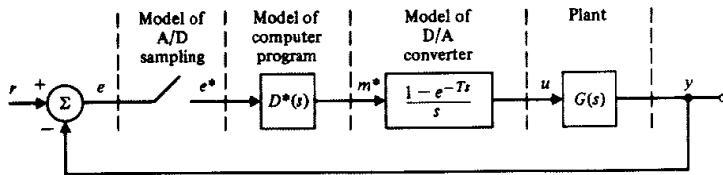
In other words, because E^* is already periodic, shifting it an integral number of periods leaves it unchanged. Substituting Eq. (5.22) into Eq. (5.20) yields

$$\begin{aligned} U^*(s) &= E^*(s) \frac{1}{T} \sum_{-\infty}^{\infty} G(s - jn\omega_s) \\ &= E^*(s)G^*(s). \quad \text{QED} \end{aligned} \quad (5.23)$$

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

$$\begin{aligned} (1 - e^{-Ts})(G(s)/s)^* &= (1 - e^{-Ts}) \left(\frac{a}{s(s+a)} \right)^* \\ &= (1 - e^{-Ts}) \left(\frac{1}{s} - \frac{1}{s+a} \right)^*. \end{aligned}$$

Using Eq. (5.5), we have

$$(1 - e^{-Ts})(G(s)/s)^* = (1 - e^{-Ts}) \left(\frac{1}{1 - e^{-Ts}} - \frac{1}{1 - e^{-aT} e^{-Ts}} \right).$$

Because $e^{-aT} = \frac{1}{2}$, this reduces to

$$\begin{aligned} (1 - e^{-Ts})(G(s)/s)^* &= \frac{(1/2)e^{-Ts}}{1 - (1/2)e^{-Ts}} \\ &= \frac{1/2}{e^{Ts} - 1/2}. \end{aligned} \quad (5.36)$$

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

$$Y(s) = R^* \frac{D^*}{1 + H^*} \frac{(1 - e^{-Ts})}{s} G(s). \quad (5.38)$$

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 $(1 - e^{-Ts})$ -term correspond to a train of impulses whose individual values can be computed by expanding in powers of e^{-Ts} . 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)

$$\begin{aligned} E &= R - Y, & (a) \\ U &= HE, & (b) \\ Y &= U^* G; & (c) \end{aligned} \quad (5.39)$$

and after sampling, the equations are

$$\begin{aligned} E^* &= R^* - Y^*, & (a) \\ U^* &= (HE)^*, & (b) \\ Y^* &= U^* G^*. & (c) \end{aligned} \quad (5.40)$$

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

$$U^* = (HR)^* - U^*(HG)^*.$$

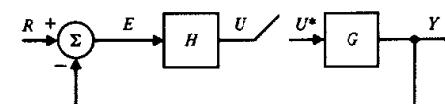


Figure 5.14
A simple system that does not have a transfer function

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 on 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.

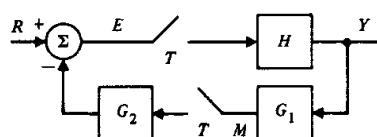
◆ Example 5.4 Another Block Diagram Analysis

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):

$$\begin{aligned} E^* &= R^* - M^* G_2^*, \\ M^* &= E^* (H G_1)^*. \end{aligned} \quad (5.44)$$

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

$$Y^* = \frac{R^* H^*}{1 + (H G_1)^* G_2^*}. \quad (5.47)$$

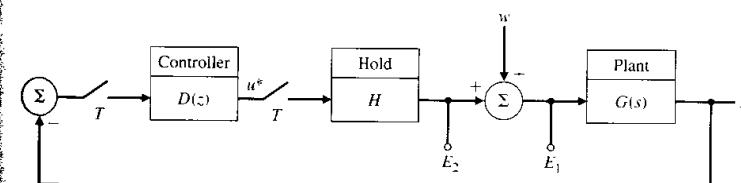
In this case we have a transfer function. Why? Because only the samples of the external input are used to cause the output. To obtain the z -transform of the samples of the output, we would let $e^{-T} = z$ in Eq. (5.47). From Eq. (5.46) we can solve for the continuous output, which consists of impulses 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

$$E_1 = W + U^*H \quad (5.48)$$

$$E_2 = U^*H \quad (5.49)$$

$$Y = WG - U^*HG \quad (5.50)$$

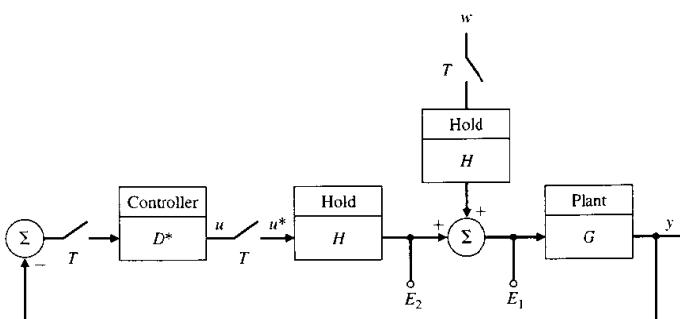
$$Y^* = (WG)^* + U^*(HG)^* \quad (5.51)$$

$$U^* = -D^*Y^* \quad (5.52)$$

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

$$(GW)^* = \frac{1}{T} \sum_{k=-\infty}^{k=\infty} G(j\omega - jk\frac{2\pi}{T}) 2\pi\delta(\omega - \omega_0 - \frac{2\pi k}{T}).$$

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.

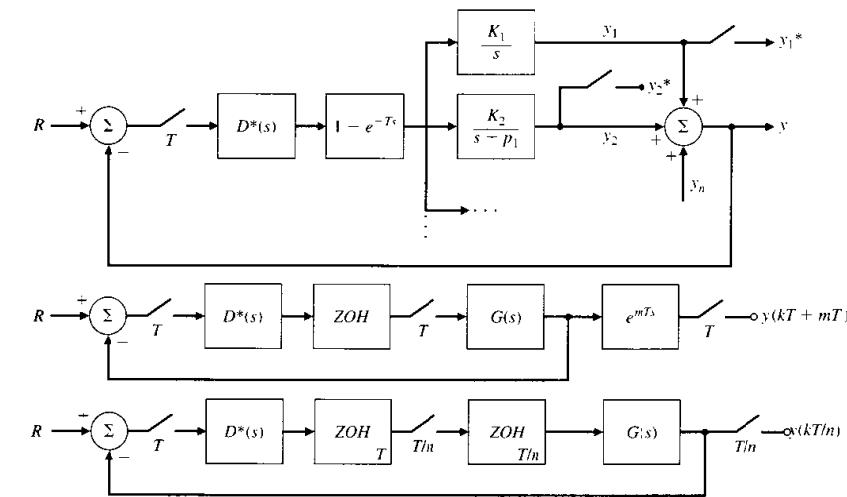
In the partial-fraction method, shown in Fig. 5.18(a), the outputs of the several fractions are sampled and the values of $y_1(kT)$, $y_2(kT)$, ... are computed in the regular way with z -transforms or MATLAB statements. These values at the instant kT represent initial conditions for the several partial 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

$$\text{SYS} = \text{ss}(F, G, H, J).$$

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

$$\text{SYSD} = \text{c2d}(\text{SYS}, T).$$

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

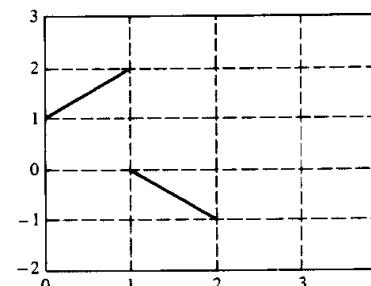


Figure 5.20
Block diagram of a sample and first-order hold

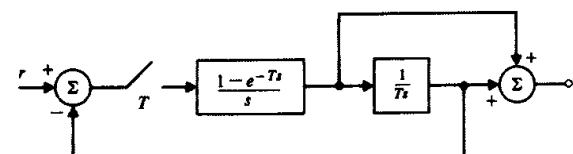


Figure 5.21
A sampled-data system

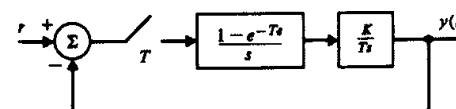
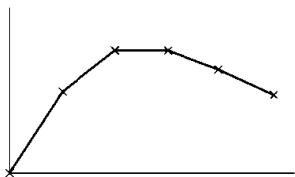
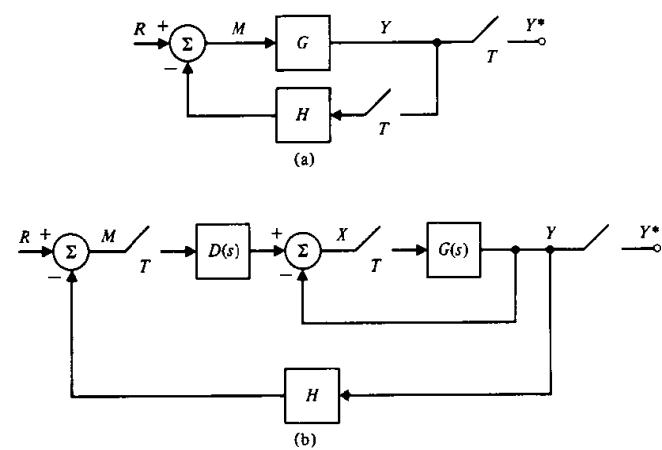


Figure 5.22
Response of a sample and triangle hold



- (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 $\omega_s/10$.
- 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 $\omega_s/10$.
- 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 $\omega_s/10$.
- 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 $\omega_s/10$.
- 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 $r(t) = \sin(2t) + \sin(15t)$ 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 $\frac{1}{(s+1)^2}$ 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

$$G(s) = \frac{1}{(s+1)}, \quad R(s) = \frac{1}{s}, \quad \text{and} \quad T = 1.$$

- (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

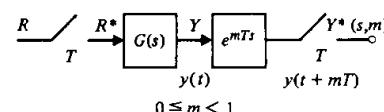


Figure 5.24
Block diagrams showing the modified z-transform

$$0 \leq m < 1$$

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 $v(t) = e^{j\omega_0 t + j\phi}$. For this we have

$$V(j\omega) = \int_{-\infty}^{\infty} e^{j\omega_0 t + j\phi} e^{-j\omega t} dt. \quad (5.59)$$

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

$$\begin{aligned} V(j\omega) &= \int_{-\infty}^{\infty} e^{j\omega_0 t - j\phi} e^{-j\omega t} dt \\ &= e^{j\phi} \int_{-\infty}^{\infty} e^{j(\omega_0 - \omega)t} dt \\ &= 2\pi e^{j\phi} \delta(\omega - \omega_0). \end{aligned}$$

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.

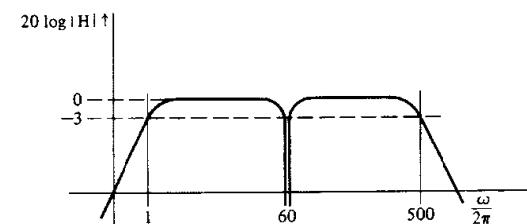
• 6 •

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



¹ Including microprocessors and special-purpose devices for digital signal processing, called DSP chips.

emulation

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 use 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 $s = s_o$ will, when sampled with period T , represent the response of a discrete filter with a pole at $z = e^{s_o T}$. 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

$$\begin{aligned} u(t) &= \int_0^t [-au(\tau) + ae(\tau)] d\tau, \\ u(kT) &= \int_0^{kT-T} [-au + ae] d\tau + \int_{kT-T}^{kT} [-au + ae] d\tau \\ &= u(kT - T) + \left\{ \begin{array}{l} \text{area of } -au + ae \\ \text{over } kT - T \leq \tau < kT. \end{array} \right. \end{aligned} \quad (6.3)$$

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 ,

$$\begin{aligned} u_1(kT) &= u_1(kT - T) + T[-au_1(kT - T) + ae(kT - T)] \\ &= (1 - aT)u_1(kT - T) + aTe(kT - T). \end{aligned} \quad (6.4)$$

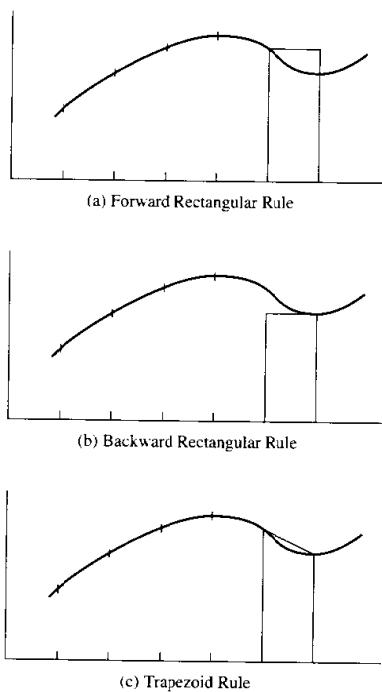
The transfer function corresponding to the forward rectangular rule in this case is

$$\begin{aligned} H_F(z) &= \frac{aTz^{-1}}{1 - (1 - aT)z^{-1}} \\ &= \frac{a}{(z - 1)/T + a} \quad (\text{forward rectangular rule}). \end{aligned} \quad (6.5)$$

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

$$\begin{aligned} u_2(kT) &= u_2(kT - T) + T[-au_2(kT) + ae(kT)] \\ &= \frac{u_2(kT - T)}{1 + aT} + \frac{aT}{1 + aT}e(kT). \end{aligned} \quad (6.6)$$

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	$s \leftarrow \frac{z-1}{Tz}$	(6.11)
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

$$H_T(z) = H(s)|_{s=\frac{z-1}{Tz+1}}. \quad (6.12)$$

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

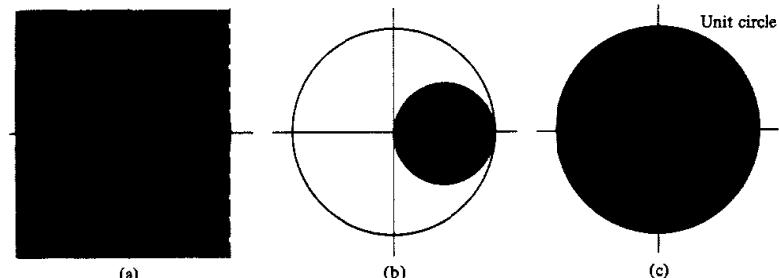
$$\begin{aligned} i) \quad z &= 1 + Ts, && \text{(forward rectangular rule).} \\ ii) \quad z &= \frac{1}{1 - Ts}, && \text{(backward rectangular rule).} \\ iii) \quad z &= \frac{1 + Ts/2}{1 - Ts/2} && \text{(bilinear rule).} \end{aligned} \quad (6.13)$$

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

$$\begin{aligned} z &= \frac{1}{2} + \left\{ \frac{1}{1 - Ts} - \frac{1}{2} \right\} \\ &= \frac{1}{2} - \frac{1}{2} \frac{1 + Ts}{1 - Ts}. \end{aligned} \quad (6.14)$$

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 $H_T(z)$ 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 $H_T(z)$ to a sinusoid of frequency ω_1 is $H_T(z_1)$. We consider now Eq. (6.8) for $H_T(z_1)$ and manipulate it into a more convenient form for our present purposes

$$\begin{aligned} H_T(z_1) &= a / \left(\frac{2 e^{j\omega_1 T} - 1}{T e^{j\omega_1 T} + 1} + a \right) \\ &= a / \left(\frac{2 e^{j\omega_1 T/2} - e^{-j\omega_1 T/2}}{T e^{j\omega_1 T/2} + e^{-j\omega_1 T/2}} + a \right) \\ &= a / \left(\frac{2}{T} j \tan \frac{\omega_1 T}{2} + a \right). \end{aligned} \quad (6.15)$$

The magnitude squared of H_T will be $\frac{1}{2}$ when

$$\frac{2}{T} \tan \frac{\omega_1 T}{2} = a$$

or

$$\tan \frac{\omega_1 T}{2} = \frac{aT}{2}. \quad (6.16)$$

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 $\tan^{-1}(aT/2) \approx aT/2$, that is, if $\omega_s (= 2\pi/T) \gg a$ 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 prewarped 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 pass bandwidth ($\omega_p = 1$) is

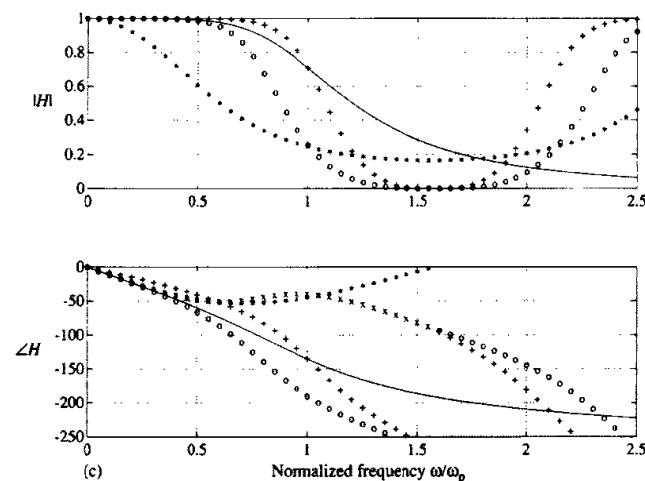
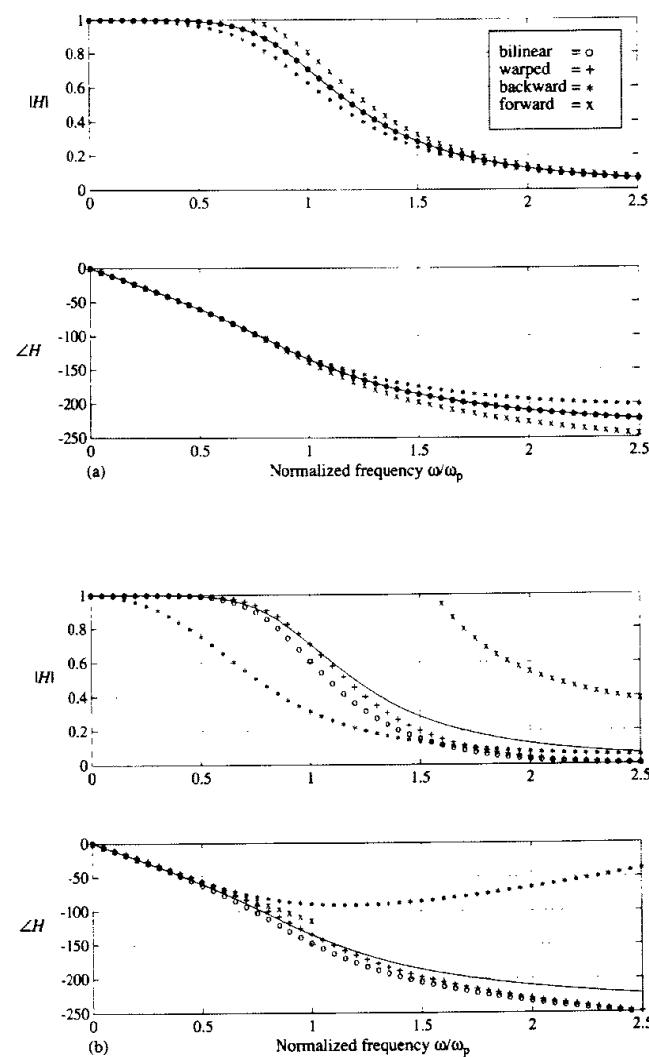
$$H(s) = \frac{1}{s^3 + 2s^2 + 2s + 1}.$$

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 close 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

$$\begin{aligned} \mathbf{x}(k+1) - T\mathbf{A}\mathbf{x}(k+1) - T\mathbf{B}e(k+1) &= \mathbf{x}(k) \\ &\stackrel{\triangle}{=} \mathbf{w}(k+1). \end{aligned} \quad (6.23)$$

From this equation, solving for \mathbf{x} in terms of \mathbf{w} and e

$$\begin{aligned} (\mathbf{I} - \mathbf{AT})\mathbf{x} &= \mathbf{w} + T\mathbf{Be} \\ \mathbf{x} &= (\mathbf{I} - \mathbf{AT})^{-1}\mathbf{w} + (\mathbf{I} - \mathbf{AT})^{-1}\mathbf{BT}e. \end{aligned} \quad (6.24)$$

With this expression for \mathbf{x} , Eq. (6.23) can be put in standard form as

$$\mathbf{w}(k+1) = (\mathbf{I} - \mathbf{AT})^{-1}\mathbf{w}(k) + (\mathbf{I} - \mathbf{AT})^{-1}\mathbf{BT}e(k). \quad (6.25)$$

and the output equation is now

$$u(k) = \mathbf{C}(\mathbf{I} - \mathbf{AT})^{-1}\mathbf{w} + \{\mathbf{D} + \mathbf{C}(\mathbf{I} - \mathbf{AT})^{-1}\mathbf{BT}\}e. \quad (6.26)$$

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

$$\begin{aligned} \frac{2(z-1)}{T(z+1)}\mathbf{X} &= \mathbf{AX} + \mathbf{BE} \\ (z-1)\mathbf{X} &= \frac{AT}{2}(z+1)\mathbf{X} + \frac{BT}{2}(z+1)e \\ U &= \mathbf{CX} + \mathbf{DE}, \end{aligned} \quad (6.27)$$

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

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{Ax}(t) + \mathbf{Be}(t), \\ u(t) &= \mathbf{Cx}(t) + \mathbf{De}(t). \end{aligned}$$

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	$\mathbf{I} + \mathbf{A}T$	$(\mathbf{I} - \mathbf{A}T)^{-1}\mathbf{B}T$	$(\mathbf{I} + \frac{\Delta T}{2})(\mathbf{I} - \frac{\Delta T}{2})^{-1}$
Γ	$\mathbf{B}T$	$(\mathbf{I} - \mathbf{A}T)^{-1}$	$(\mathbf{I} - \frac{\Delta T}{2})^{-1}\mathbf{B}\sqrt{T}$
\mathbf{H}		$\mathbf{C}(\mathbf{I} - \mathbf{A}T)^{-1}$	$\sqrt{T}\mathbf{C}(\mathbf{I} - \frac{\Delta T}{2})^{-1}$
\mathbf{J}	\mathbf{D}	$\mathbf{D} + \mathbf{C}(\mathbf{I} - \mathbf{A}T)^{-1}\mathbf{B}T$	$\mathbf{D} + \mathbf{C}(\mathbf{I} - \frac{\Delta T}{2})^{-1}\mathbf{B}T/2$

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:

1. 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 $\theta = bT$.
2. 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.
3. 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.

- (a) If no delay in the discrete response is desired, all zeros at $s = \infty$ are mapped to $z = -1$.
 - (b) 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.
4. 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 $\frac{1 - e^{-aT}}{2}$. 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

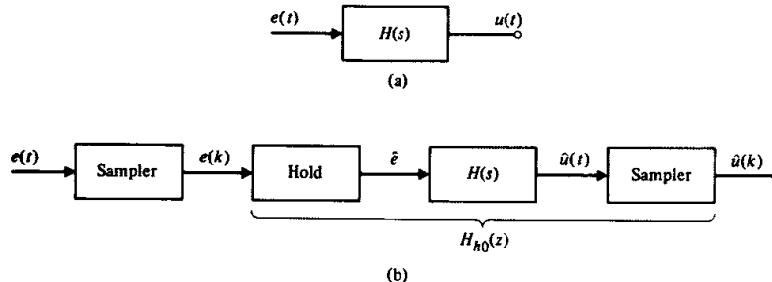
$$\text{SYSD} = c2d(\text{SYS}, T_s, \text{'matched'}).$$

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

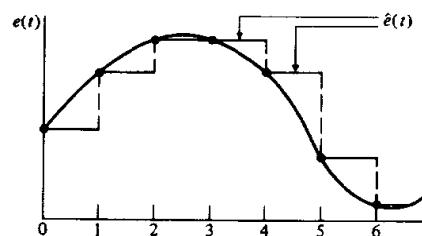
For this technique, we construct the situation sketched in Fig. 6.5. The samplers in Fig. 6.5(b) provide the samples at the input of $H_{h0}(z)$ and take samples at its output insuring that $H_{h0}(z)$ can be realized as a discrete transfer function. The philosophy of the design is the following. We 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 $\hat{e}(t)$ 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

$$H_{h0}(z) = (1 - z^{-1}) \mathcal{Z} \left\{ \frac{H(s)}{s} \right\}. \quad (6.35)$$

◆ 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)

$$\begin{aligned} \mathcal{Z}\left\{\frac{H(s)}{s}\right\} &= \sum_0^{\infty} z^{-k} - \sum_0^{\infty} z^{-k} e^{-akT} \\ &= \frac{1}{1-z^{-1}} - \frac{1}{1-e^{-aT}z^{-1}} \\ &= \frac{(1-e^{-aT}z^{-1}) - (1-z^{-1})}{(1-z^{-1})(1-e^{-aT}z^{-1})}. \end{aligned} \quad (6.37)$$

Substituting Eq. (6.37) in Eq. (6.35), the zero-order-hold equivalent of $H(s)$ is found as

$$H_{ho}(z) = \frac{(1-e^{-aT})}{z-e^{-aT}}. \quad (6.38)$$

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

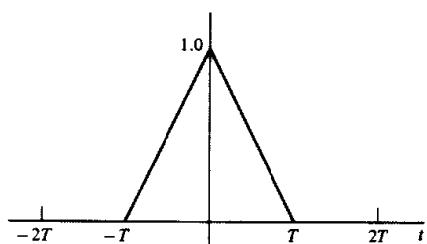
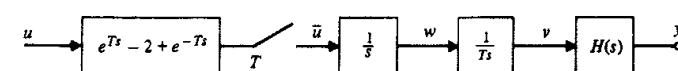


Figure 6.8
Block diagram of the triangle-hold equivalent



hold. The effect of the triangle hold is to extrapolate the samples so as to connect sample to sample in a straight line. Although the continuous system that does this is noncausal, the resulting discrete equivalent is causal.

The Laplace transform of the extrapolation filter that follows the impulse sampling is

$$\frac{e^{Ts} - 2 + e^{-Ts}}{Ts^2}.$$

Therefore the discrete equivalent that corresponds to Eq. (6.35) is

$$H_{tri}(z) = \frac{(z-1)^2}{Tz} \mathcal{Z}\left\{\frac{H(s)}{s^2}\right\}. \quad (6.39)$$

◆ 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

$$\begin{aligned} \mathcal{Z}\left\{\frac{H(s)}{s^2}\right\} &= \mathcal{Z}\left\{\frac{1}{s^4}\right\} \\ &= \frac{T^3}{6} \frac{(z^2 + 4z + 1)z}{(z-1)^4}, \end{aligned} \quad (6.40)$$

and direct substitution into Eq. (6.39) results in

$$\begin{aligned} H_{tri}(z) &= \frac{(z-1)^2}{Tz} \frac{T^3}{6} \frac{(z^2 + 4z + 1)z}{(z-1)^4} \\ &= \frac{T^2 z^2 + 4z + 1}{6(z-1)^2}. \end{aligned} \quad (6.41)$$

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

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{F}\mathbf{x} + \mathbf{G}v, \\ \dot{v} &= w/T, \\ \dot{w} &= u(t+T)\delta(t+T) - 2u(t)\delta(t) + u(t-T)\delta(t-T), \end{aligned} \quad (6.42)$$

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 \mathbf{F}_T , and the one-step solution to this equation is

$$\zeta(kT + 1) = e^{\mathbf{F}_T T} \zeta(kT)$$

because \bar{u} consists only of impulses at the sampling instants. If we define

$$\exp(\mathbf{F}_T T) = \begin{bmatrix} \Phi & \Gamma_1 & \Gamma_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.44)$$

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

$$\begin{aligned} y(k) &= \mathbf{H}\mathbf{x}(k) + \mathbf{J}u(k) \\ &= H(\xi(k) + \Gamma_2 u(k)) + Ju(k) \\ &= H\xi(k) + (J + H\Gamma_2)u(k). \end{aligned} \quad (6.46)$$

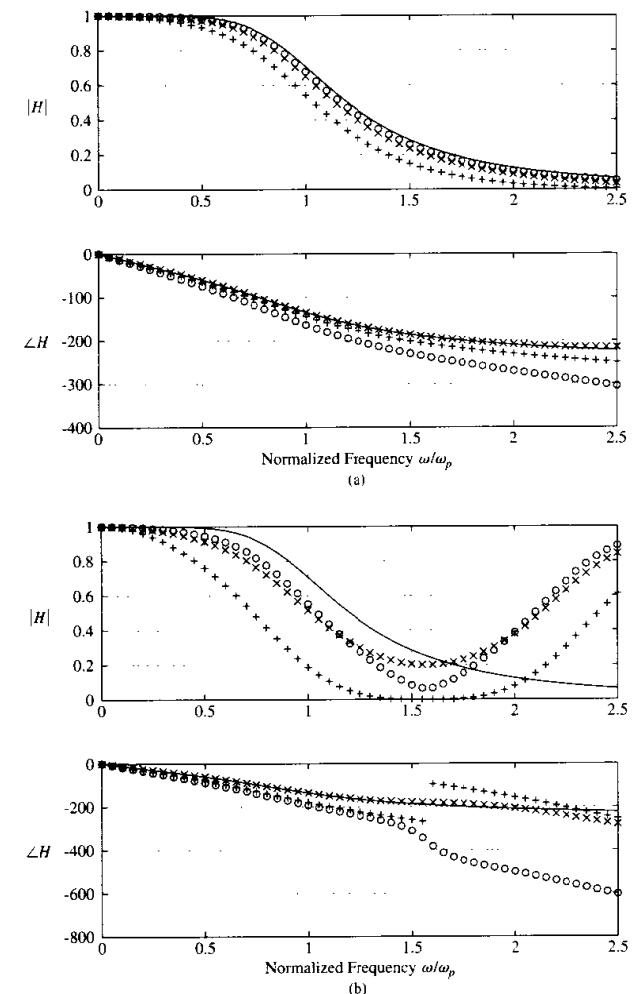
Thus the triangle equivalent of a continuous system described by $[\mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{J}]$ with sample period T is given by

$$\begin{aligned} \mathbf{A} &= \Phi, \\ \mathbf{B} &= \Gamma_1 + \Phi\Gamma_2 - \Gamma_2, \\ \mathbf{C} &= \mathbf{H}, \\ \mathbf{D} &= \mathbf{J} + \mathbf{H}\Gamma_2, \end{aligned} \quad (6.47)$$

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

$$\text{SYSD} = \text{c2d}(\text{SYS}, \text{Ts}, \text{'foh'}).$$

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 = o, zero-pole = +, and triangle = x



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 *Ts* as follows.

<i>SYSD</i>	=	<i>c2d(SYS, Ts, method)</i>	where
method	=	'zoh'	zero-order hold
method	=	'foh'	first-order hold (triangle hold)
method	=	'tustin'	Tustin's bilinear method
method	=	'prewarp'	bilinear with prewarping
method	=	'matched'	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 , 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

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 emulation

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.

design by root locus

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.

design by frequency response

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.

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:

- 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

$$|\operatorname{Re}\{s_i\}| = \sigma = \zeta \omega_n \geq 4.6/t_s. \quad (7.2)$$
- The fractional overshoot, M_p , is given in terms of the damping ratio, ζ , by the plot of Fig. 4.7 which can be very crudely approximated by

$$\zeta \approx 0.6(1 - M_p). \quad (7.3)$$

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 has a simple pole at $s = 0$.

velocity constant

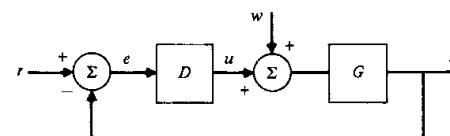
- The **velocity constant** is then given by

$$K_v = \frac{r_0}{e_{ss}} = \lim_{s \rightarrow 0} s D(s)G(s).$$

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

$$\omega_n = \frac{0.18\omega_s}{2\pi}$$

or

$$\omega_n \approx \omega_s/35.$$

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

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

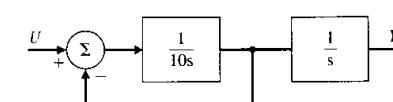
The specifications for this system are

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 $K_v \geq \frac{0.01}{0.00} = 1.0$. 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

$$D(s) = \frac{10s + 1}{s + 1}. \quad (7.4)$$

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

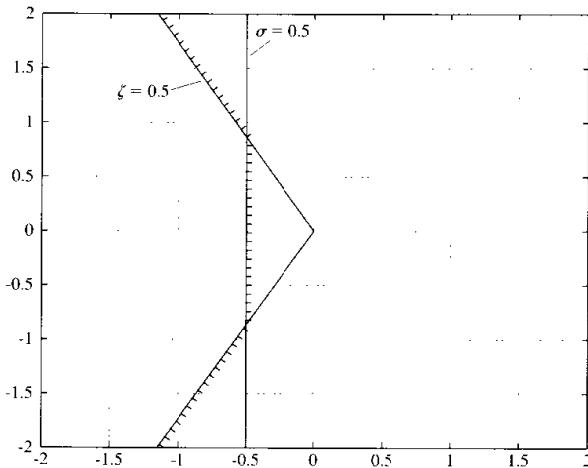
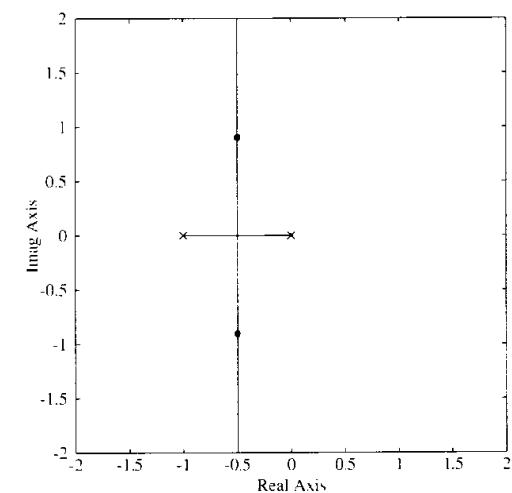


Figure 7.4
Root locus for compensated antenna model



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 $z = e^{-jT}$; 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

$$u(k) = 0.8187u(k-1) + 9.15(e(k) - 0.9802e(k-1)).$$

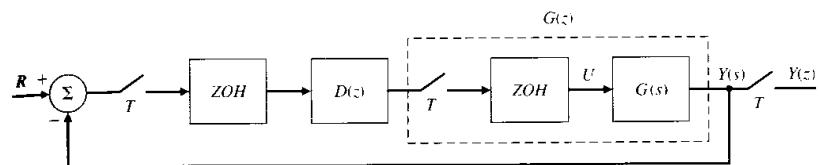
This equation can be directly evaluated by a computer.

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

$$\begin{aligned} G(z) &= \frac{z - 1}{z} \left\{ \frac{Tz}{(z-1)^2} - \frac{z}{a(z-1)} + \frac{1}{a} \frac{z}{z - e^{-aT}} \right\} \\ &= \frac{Az + B}{a(z-1)(z - e^{-aT})}, \end{aligned}$$

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

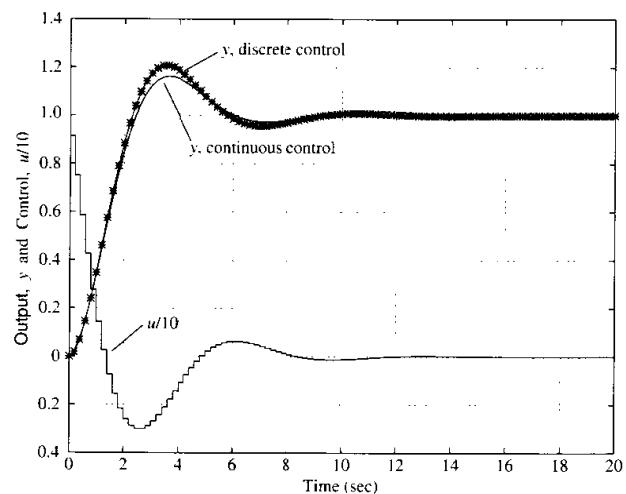
`antd = c2d(ant,0.2);`

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 = lead1d * 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 $\omega_{cp} = 0.8$ rad, and with sampling at $T = 0.2$, this corresponds only to $\phi = \omega_{cp}T = 4.5^\circ$. However, at $T = 1.0$, the sample-and-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

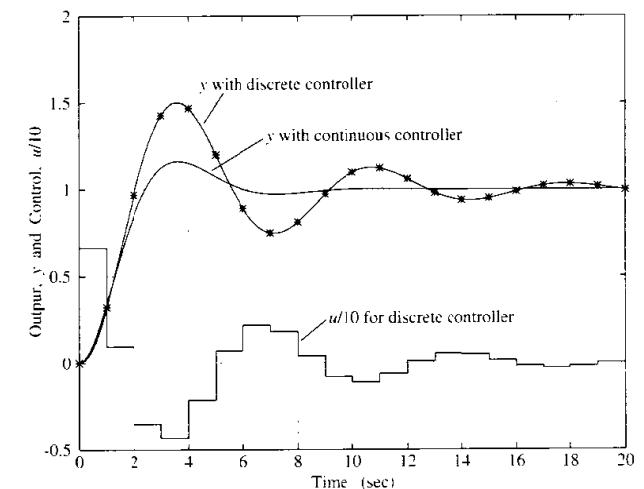
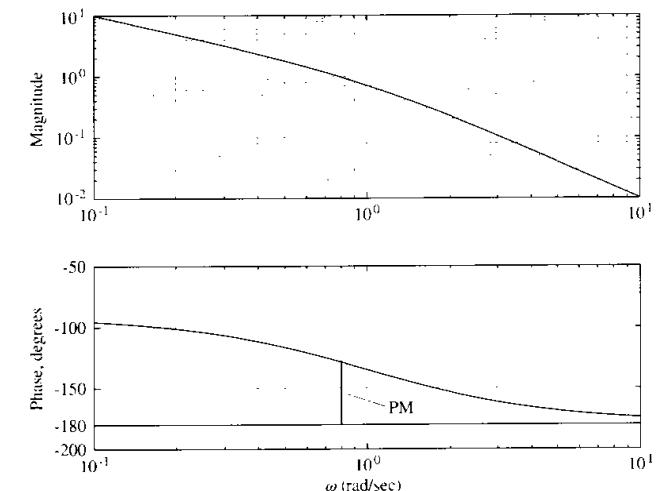


Figure 7.8
Bode plot of the continuous design for the antenna control



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-and-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 $z = e^{\sigma T}$, 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.

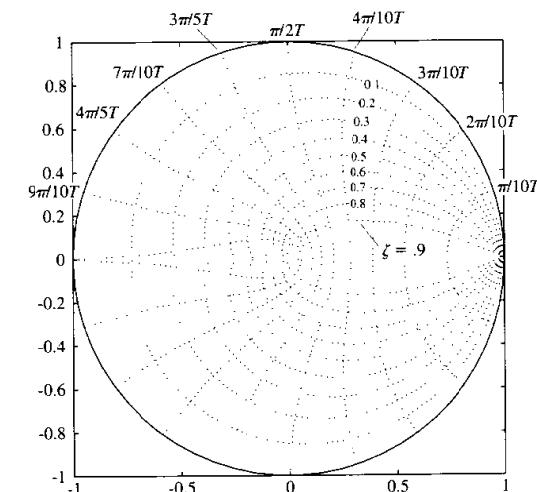


Figure 7.9
Lines of constant damping and natural frequency in the z -plane

◆ 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 $\zeta \geq 0.5$, 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 $\zeta = 0.5$. 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

$$G(z) = (1 - z^{-1}) \mathcal{Z} \left\{ \frac{G(s)}{s} \right\}. \quad (7.11)$$

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

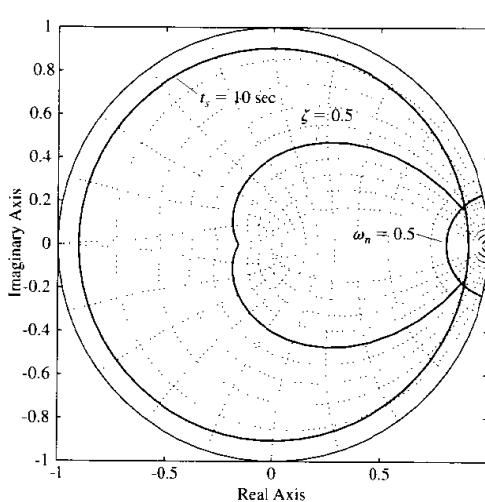


Figure 7.10
Plot of acceptable pole locations in the z -plane

discrete time final value

discrete system type

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

$$\begin{aligned} E(z) &= \frac{R(z)}{1 + D(z)G(z)} \\ &= \frac{z}{z - 1} \frac{1}{1 + D(z)G(z)}. \end{aligned}$$

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)

$$\begin{aligned} e(\infty) &= \lim_{z \rightarrow 1} (z - 1) \frac{z}{z - 1} \frac{1}{1 + D(z)G(z)} \\ &= \frac{1}{1 + D(1)G(1)} \\ &= \frac{1}{1 + K_p}. \end{aligned} \quad (7.12)$$

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

$$\begin{aligned} e(\infty) &= \lim_{z \rightarrow 1} (z - 1) \frac{Tz}{(z - 1)^2} \frac{1}{1 + DG} \\ &= \lim_{z \rightarrow 1} \frac{Tz}{(z - 1)(1 + DG)} \\ &\stackrel{\triangle}{=} \frac{1}{K_v}. \end{aligned} \quad (7.13)$$

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

$$\begin{aligned} E(z) &= \frac{R(z)(1 - H(z))}{Tz} \\ &= \frac{Tz}{(z-1)^2}(1 - H(z)), \end{aligned}$$

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.

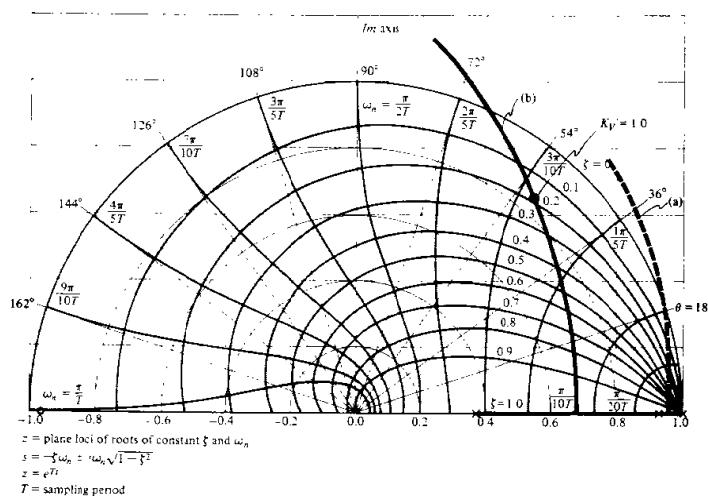
Solution. The exact discrete model of the plant plus hold is given by the $G(z)$ in Eq. (7.9). If the controller consisted simply of a proportional gain [$D(z) = K$], the locus of roots versus K can be found by solving the characteristic equation

$$1 + 0.0484K \frac{z + 0.9672}{(z - 1)(z - 0.9048)} = 0$$

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 wth $D(z)$ having the poles and zeros of Eq. (7.10)



radius of the roots never gets less than 0.95, preventing the t_s specification from being met. The system goes unstable at $K \geq 19$ [where $K_v = 0.92$, 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 [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 $p = \text{rlocus}(\text{sysold}, 6.64)$ 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 $K_v = 1$ 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 $K_v \geq 1$, $t_s \leq 10$ sec, and $\zeta \geq 0.5$. If we transform the specifications to the z -plane we compute that the t_s 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 $K_v \geq 1$ 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_v 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 $z \approx 0$ will produce acceptable z -plane pole locations, the gain K_v is not sufficiently high to meet the criterion for steady-state error. The only way to raise K_v 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

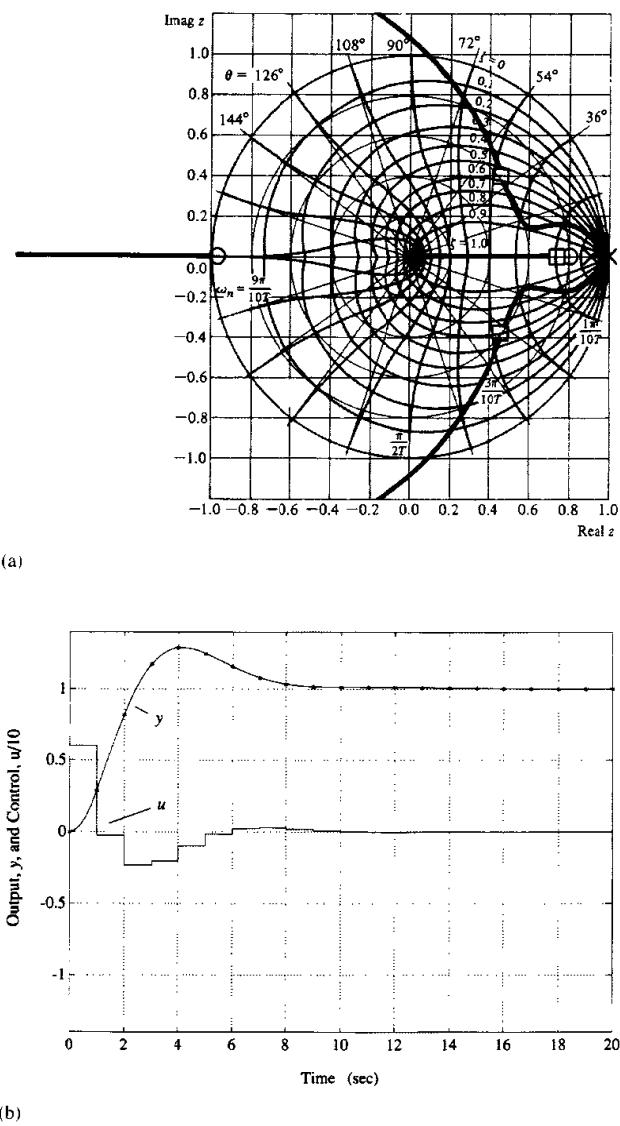
$$D(z) = 6 \frac{z - 0.80}{z - 0.05} \quad (7.19)$$

meets the required z -plane constraints for the complex roots and has a $K_v = 1.26$. 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 $K_c = 1.04$. 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

$$D(z) = 9 \frac{(z - 0.8)}{(z + 0.8)} \quad (7.21)$$

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 $\xi = 0.2$ 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 $K_c = 1$; 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.

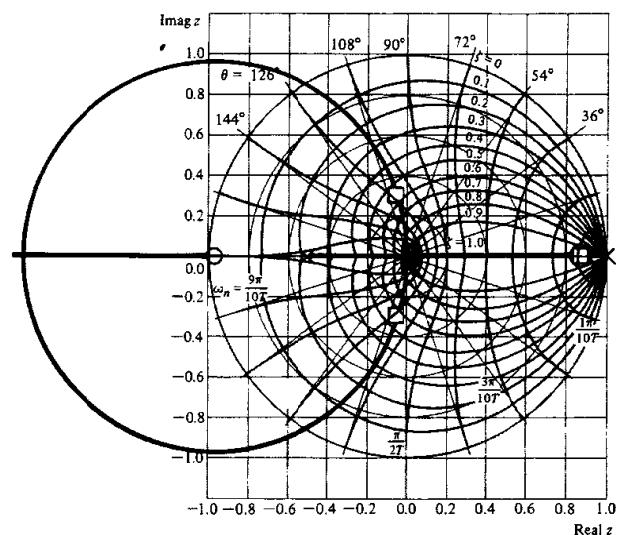
ripple

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.

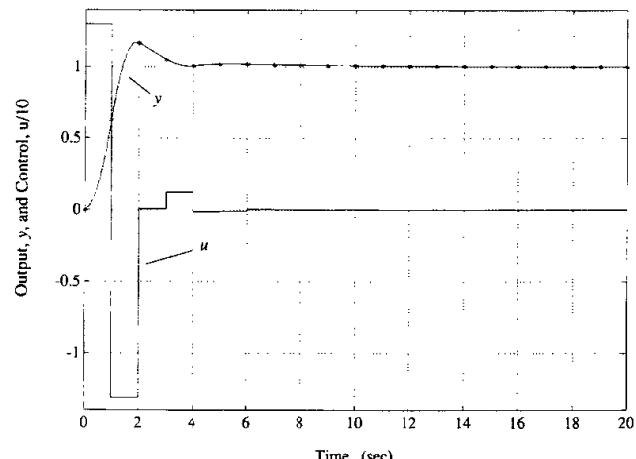
Because a one-cycle delay has a z -transform of z^{-1} , the effect of a full-cycle delay can be analyzed by adding z^{-1} to the numerator of the controller representation. This will result in an additional pole at the origin of the z -plane. If there is a delay of two cycles, two poles will be added 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



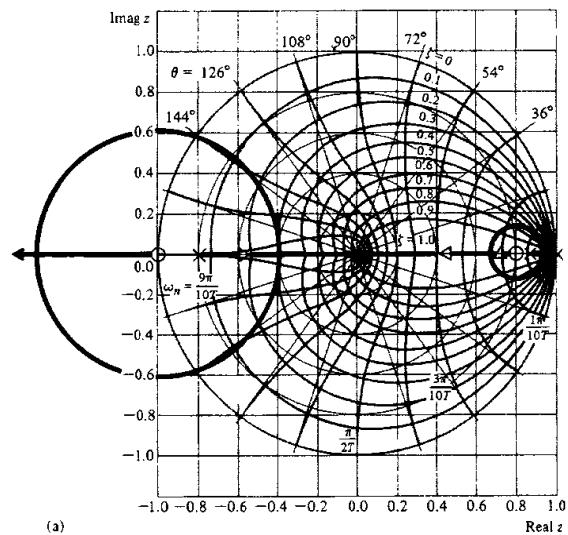
(a)



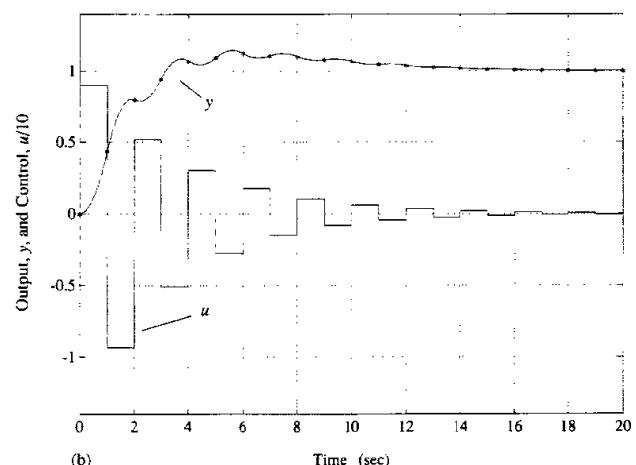
(b)

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.

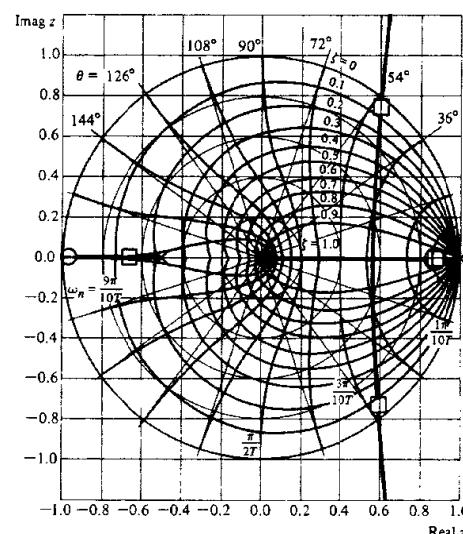
7.4 Frequency Response Methods

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.

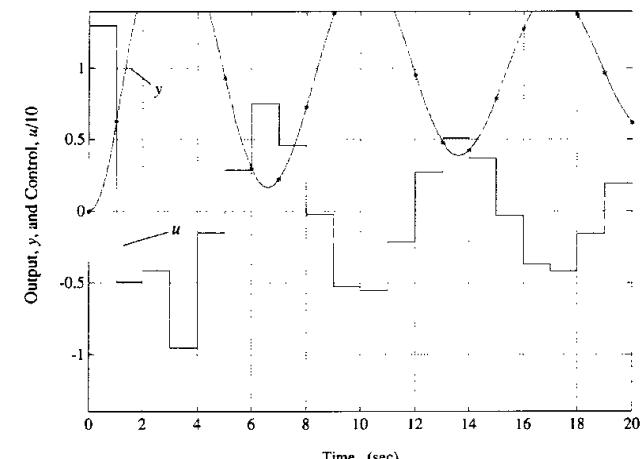
1. The gain and phase curves for a rational transfer function can be easily plotted by hand.
2. If a physical realization of the system is available, the frequency response can be measured experimentally without the necessity of having a mathematical model at all.
3. Nyquist's stability criterion can 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,

$$\begin{aligned} \text{magnitude} &= |H(z)|_{e^{j\omega T}}, \\ \text{phase} &= \angle H(z) \Big|_{e^{j\omega T}}. \end{aligned}$$

◆ 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

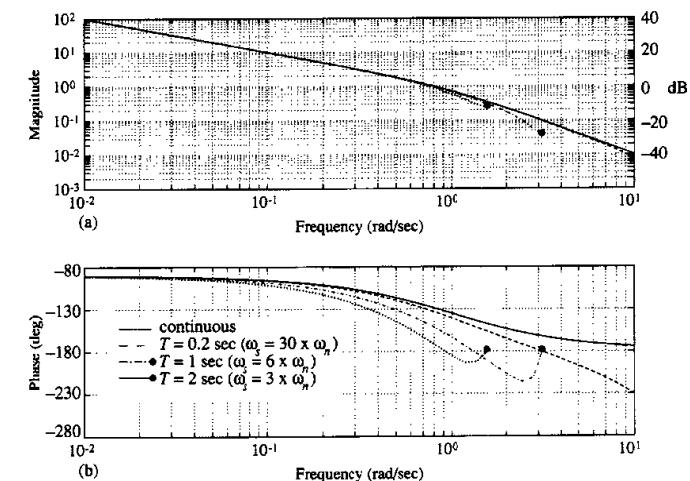
$$\begin{aligned} G_1(z) &= 0.0187 \frac{z + 0.9355}{(z - 1)(z - 0.8187)} \quad \text{for } T = 0.2 \text{ sec} \\ G_2(z) &= 0.368 \frac{z + 0.718}{(z - 1)(z - 0.368)} \quad \text{for } T = 1 \text{ sec} \\ G_3(z) &= 1.135 \frac{z + 0.523}{(z - 1)(z - 0.135)} \quad \text{for } T = 2 \text{ sec}. \end{aligned} \quad (7.24)$$

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

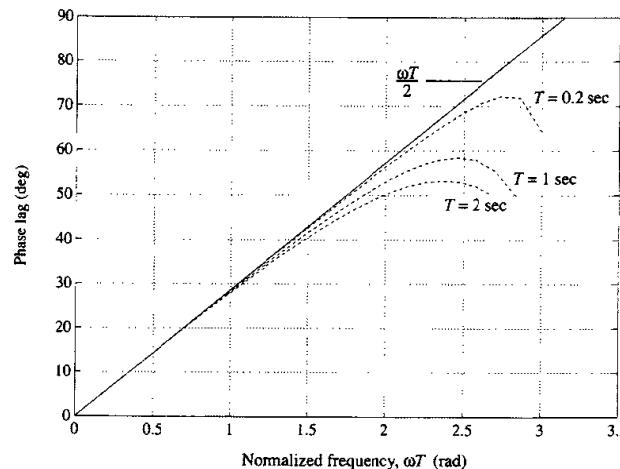
$$\Delta\phi = \frac{\omega T}{2} \quad (7.25)$$

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

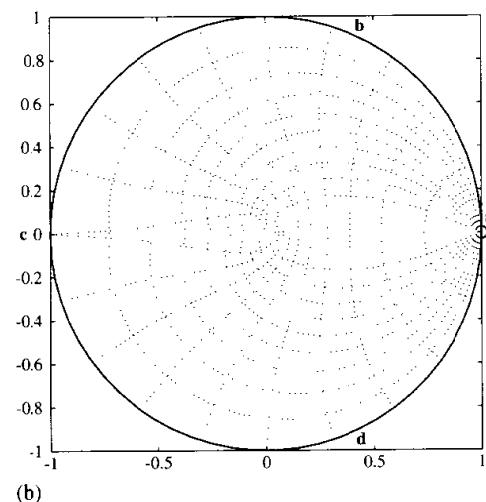
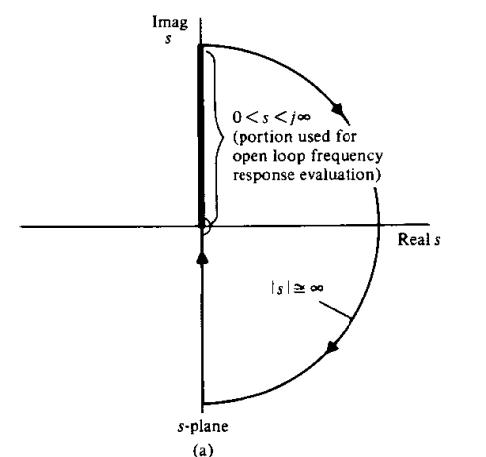
For continuous systems, the Nyquist stability criterion seeks to determine whether there are any zeros of the closed-loop characteristic equation

$$1 + KD(s)G(s) = 0 \quad (7.26)$$

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 $KD(s)G(s)$ 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 $KD(s)G(s)$ 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 KDG 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 $1 + KD(z)G(z)$ 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

$$G(z) = \frac{1}{s(s+1)}. \quad (7.30)$$

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.

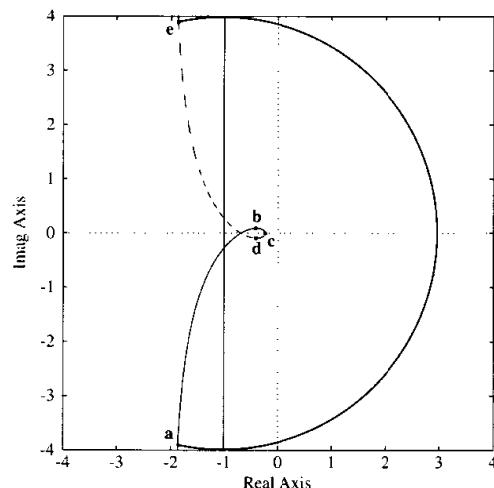


Figure 7.19
Nyquist plot of
Example 7.9

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

$$G(z) = 0.0012 \frac{(z - 3.38)(z + 0.242)}{(z - 1)(z - 0.8187)^2}.$$

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 $K = 1$ as 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

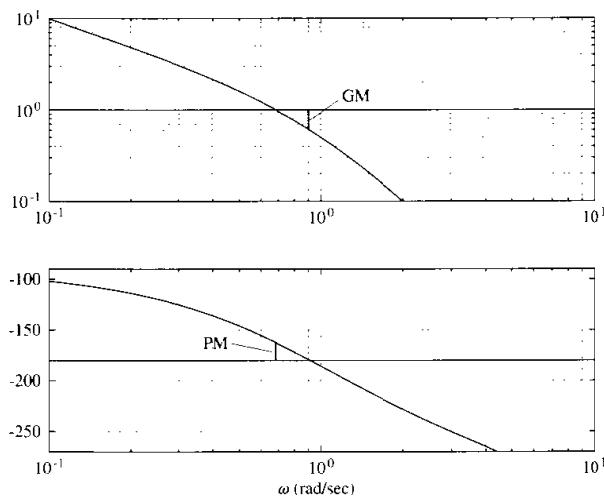
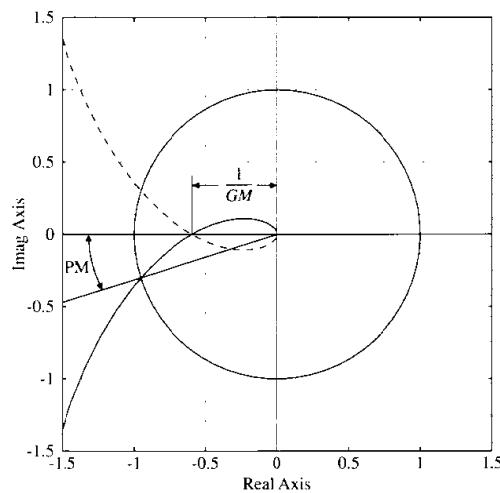
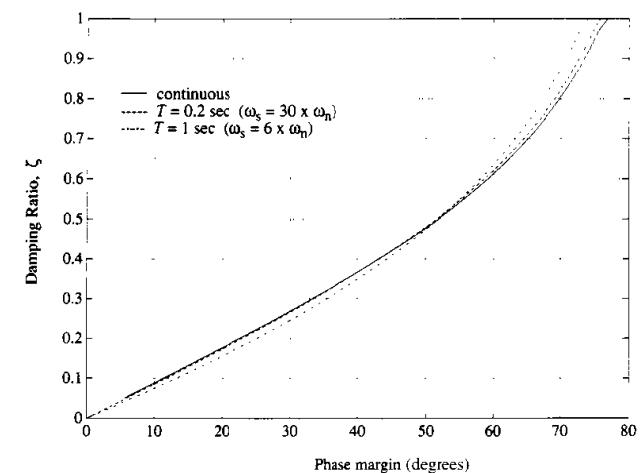


Figure 7.21
Gain and phase margins
on a Nyquist plot for
Example 7.10



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 $z = e^{j\omega T}$. The ζ 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 ζ and PM in the figure shows that the approximation of $\zeta \cong PM/100$ 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_1 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

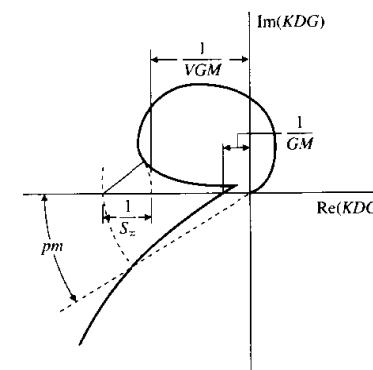
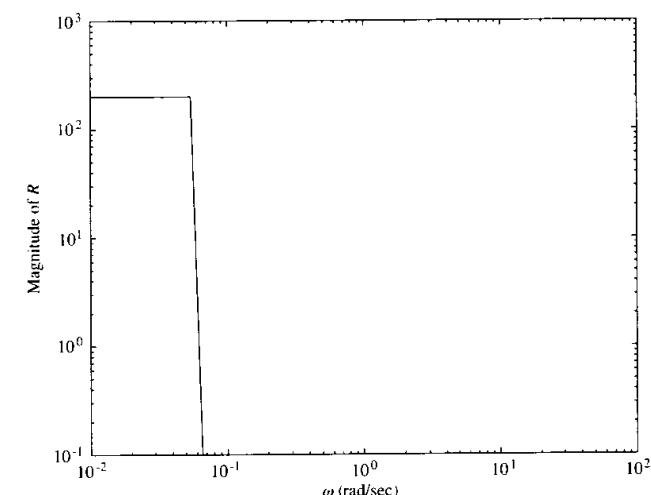


Figure 7.24
Sketch of typical specification of frequency content for reference input tracking



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

$$|\mathcal{S}| W_1 \leq 1. \quad (7.34)$$

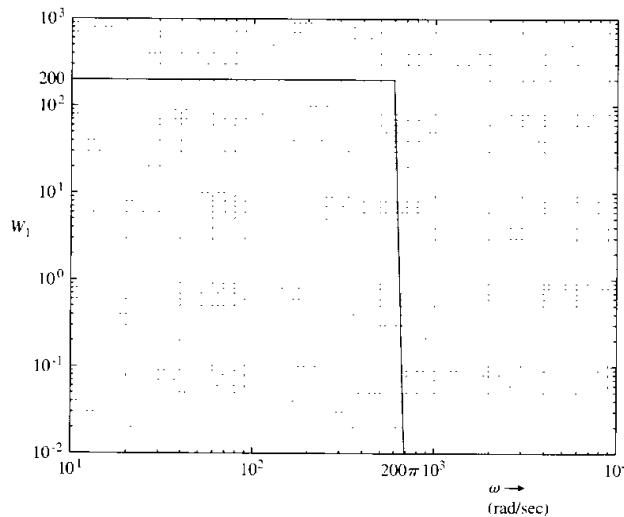
◆ Example 7.11 Performance Bound Function

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

$$\begin{aligned} \frac{W_1}{|DG|} &\leq 1 \\ |DG| &\geq W_1. \end{aligned} \quad (7.35)$$

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

$$G_o(s) = \frac{K}{s^2}. \quad (7.37)$$