

# پلتفرم اختصاصی مهندسی کنترل



<https://controlengineers.ir>



<https://t.me/controlengineers>



<https://www.instagram.com/controlengineers.ir>

# DIGITAL CONTROL SYSTEM ANALYSIS & DESIGN

*FOURTH EDITION*

*GLOBAL EDITION*

**Charles L. Phillips**

*Auburn University*

**H. Troy Nagle**

*North Carolina State University*

**Aranya Chakraborty**

*North Carolina State University*

**PEARSON**

**Pearson Education Limited**

Edinburgh Gate  
Harlow  
Essex CM20 2JE  
England

Visit us on the World Wide Web at:  
[www.pearsonglobaleditions.com](http://www.pearsonglobaleditions.com)

© Pearson Education Limited 2015

ISBN 10: 1-292-06122-7

ISBN 13: 978-1-292-06122-1 (Print)

ISBN 13: 978-1-292-06188-7 (PDF)

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

Typeset in 9/10 Times LT Std by Jouve India

Printed and bound by Courier Westford in The United States of America

# CONTENTS

Preface 9

## Chapter 1 INTRODUCTION 11

Overview	11
Digital Control System	12
The Control Problem	15
Satellite Model	16
Servomotor System Model	18
Antenna Pointing System	20
Robotic Control System	21
Temperature Control System	22
Single-Machine Infinite Bus Power System	24
Summary	27
References	27
Problems	27

## Chapter 2 DISCRETE-TIME SYSTEMS AND THE Z-TRANSFORM 35

Introduction	35
Discrete-Time Systems	35
Transform Methods	37
Properties of the z-Transform	40
Addition and Subtraction	40
Multiplication by a Constant	40
Real Translation	41
Complex Translation	43
Initial Value	44
Final Value	44
Finding z-Transforms	45
Solution of Difference Equations	48
The Inverse z-Transform	51
Power Series Method	51
Partial-Fraction Expansion Method	52
Inversion-Formula Method	56
Discrete Convolution	57
Simulation Diagrams and Flow Graphs	59
State Variables	63
Other State-Variable Formulations	71
Transfer Functions	80

<i>Solutions of the State Equations</i>	84
Recursive Solution	84
z-Transform Method	86
Numerical Method via Digital Computer	87
Properties of the State Transition Matrix	88
<i>Linear Time-Varying Systems</i>	89
<i>Summary</i>	90
References and Further Readings	90 • Problems 90

### **Chapter 3 SAMPLING AND RECONSTRUCTION 100**

<i>Introduction</i>	100
<i>Sampled-Data Control Systems</i>	100
<i>The Ideal Sampler</i>	103
<i>Evaluation of <math>E^*(S)</math></i>	105
<i>Results from the Fourier Transform</i>	108
<i>Properties of <math>E^*(S)</math></i>	110
<i>Data Reconstruction</i>	113
Zero-Order Hold	114
First-Order Hold	118
Fractional-Order Holds	119
<i>Summary</i>	121
References and Further Readings	121 • Problems 122

### **Chapter 4 OPEN-LOOP DISCRETE-TIME SYSTEMS 126**

<i>Introduction</i>	126
<i>The Relationship Between <math>E(Z)</math> and <math>E^*(S)</math></i>	126
<i>The Pulse Transfer Function</i>	127
<i>Open-Loop Systems Containing Digital Filters</i>	133
<i>The Modified z-Transform</i>	136
<i>Systems with Time Delays</i>	139
<i>Nonsynchronous Sampling</i>	142
<i>State-Variable Models</i>	145
<i>Review of Continuous-Time State Variables</i>	146
<i>Discrete-Time State Equations</i>	150
<i>Practical Calculations</i>	154
<i>Summary</i>	156
References and Further Readings	156 • Problems 156

### **Chapter 5 CLOSED-LOOP SYSTEMS 167**

<i>Introduction</i>	167
<i>Preliminary Concepts</i>	167

*Derivation Procedure* 171  
*State-Variable Models* 178  
*Summary* 187  
*References and Further Readings* 187 • *Problems* 188

## **Chapter 6 SYSTEM TIME-RESPONSE CHARACTERISTICS 198**

*Introduction* 198  
*System Time Response* 198  
*System Characteristic Equation* 207  
*Mapping the s-Plane into the z-Plane* 208  
*Steady-State Accuracy* 215  
*Simulation* 218  
*Control Software* 223  
*Summary* 223  
*References and Further Readings* 224 • *Problems* 224

## **Chapter 7 STABILITY ANALYSIS TECHNIQUES 230**

*Introduction* 230  
*Stability* 230  
*Bilinear Transformation* 234  
*The Routh-Hurwitz Criterion* 236  
*Jury's Stability Test* 239  
*Root Locus* 244  
*The Nyquist Criterion* 248  
*The Bode Diagram* 257  
*Interpretation of the Frequency Response* 259  
*Closed-Loop Frequency Response* 261  
*Summary* 270  
*References and Further Readings* 270 • *Problems* 270

## **Chapter 8 DIGITAL CONTROLLER DESIGN 279**

*Introduction* 279  
*Control System Specifications* 279  
     *Steady-State Accuracy* 280  
     *Transient Response* 280  
     *Relative Stability* 282  
     *Sensitivity* 283  
     *Disturbance Rejection* 284  
     *Control Effort* 285  
*Compensation* 285  
*Phase-Lag Compensation* 287

*Phase-Lead Compensation* 294  
*Phase-Lead Design Procedure* 295  
*Lag-Lead Compensation* 303  
*Integration and Differentiation Filters* 307  
*PID Controllers* 309  
*PID Controller Design* 313  
*Design by Root Locus* 321  
*Summary* 334  
*References and Further Readings* 334 • *Problems* 335

## **Chapter 9 POLE-ASSIGNMENT DESIGN AND STATE ESTIMATION 343**

*Introduction* 343  
*Pole Assignment* 343  
*State Estimation* 352  
     Observer Model 352  
     Errors in Estimation 354  
     Error Dynamics 354  
     Controller Transfer Function 359  
     Closed-Loop Characteristic Equation 362  
     Closed-Loop State Equations 363  
*Reduced-Order Observers* 364  
*Current Observers* 369  
*Controllability and Observability* 374  
*Systems with Inputs* 378  
*Summary* 383  
*References and Further Readings* 384 • *Problems* 384

## **Chapter 10 SYSTEM IDENTIFICATION OF DISCRETE-TIME SYSTEMS 390**

*Introduction* 390  
*Identification of Static Systems* 391  
*Identification of Dynamic Systems* 394  
*Black-Box Identification* 394  
*Least-Squares System Identification* 401  
*Estimating Transfer Functions with Partly Known Poles and Zeros* 407  
*Recursive Least-Squares System Identification* 409  
*Practical Factors for Identification* 412  
     Choice of Input 412  
     Choice of Sampling Frequency 413  
     Choice of Signal Scaling 413  
*Summary* 414  
*References and Further Readings* 414 • *Problems* 414

## Chapter 11 LINEAR QUADRATIC OPTIMAL CONTROL 418

<i>Introduction</i>	418
<i>The Quadratic Cost Function</i>	419
<i>The Principle of Optimality</i>	421
<i>Linear Quadratic Optimal Control</i>	424
<i>The Minimum Principle</i>	433
<i>Steady-State Optimal Control</i>	434
<i>Optimal State Estimation—Kalman Filters</i>	440
<i>Least-Squares Minimization</i>	446
<i>Summary</i>	446
<i>References and Further Readings</i>	447
<i>Problems</i>	448

## Chapter 12 CASE STUDIES 454

<i>Introduction</i>	454
<i>Servomotor System</i>	455
System Model	456
Design	459
<i>Environmental Chamber Control System</i>	461
Temperature Control System	463
<i>Aircraft Landing System</i>	467
Plant Model	468
Design	468
<i>Neonatal Fractional Inspired Oxygen</i>	474
Plant Transfer Function	474
Taube's PID Controller	476
MATLAB <i>pidtool</i> PIDF Controllers	477
<i>Topology Identification in Electric Power System Models</i>	484
References	488

<i>Appendix I</i>	<i>Design Equations</i>	490
<i>Appendix II</i>	<i>Mason's Gain Formula</i>	491
<i>Appendix III</i>	<i>Evaluation of <math>E^*(s)</math></i>	496
<i>Appendix IV</i>	<i>Review of Matrices</i>	501
<i>Appendix V</i>	<i>The Laplace Transform</i>	508
<i>Appendix VI</i>	<i>z-Transform Tables</i>	522
<i>Index</i>		525



# PREFACE

This book is intended to be used primarily as a text for a first course in discrete-time control systems at either the senior undergraduate or first-year graduate level. Furthermore, the text is suitable for self-study by the practicing control engineer.

This book is based on material taught at both Auburn University and North Carolina State University, and in intensive short courses taught in both the United States and Europe. The practicing engineers who attended these short courses have influenced both the content and the direction of this book, resulting in emphasis placed on the practical aspects of designing and implementing digital control systems.

Chapter 1 presents a brief introduction and an outline of the text. Chapters 2–11 cover the analysis and design of discrete-time linear control systems. Some previous knowledge of continuous-time control systems is helpful in understanding this material. The mathematics involved in the analysis and design of discrete-time control systems is the  $z$ -transform and vector-matrix difference equations; these topics are presented in Chapter 2. Chapter 3 is devoted to the very important topic of sampling signals and the mathematical model of the sampler and data hold. This model is basic to the remainder of the text. The implications and the limitations of this model are stressed.

The next four chapters, 4–7, are devoted to the application of the mathematics of Chapter 2 to the analysis of discrete-time systems, emphasis on digital control systems. Classical design techniques are covered in Chapter 8, with the frequency-response Bode technique emphasized. Modern design techniques are presented in Chapters 9–11. Chapter 12 summarizes some case studies in discrete-time control system design. Throughout these chapters, practical computer-aided analysis and design using MATLAB are stressed.

In this fourth edition, several changes have been made. We

- Added additional MATLAB examples throughout the chapters.
- Added a new chapter on system identification (Chapter 11).
- Added new problems in many of the chapters.
- Renumbered the end-of-chapter problems to reflect their corresponding textbook sections.
- Added the MATLAB *pidtool* design technique in Chapter 8.
- Added two new case studies in Chapter 12.
- Removed four chapters (formerly Chapters 11–14) and two appendices (formerly Appendices V and VI) on digital filter implementation to reduce the overall page count, thus placing more emphasis on control design.

Each end-of-chapter problem has been written to illustrate basic material in the chapter. Generally, short MATLAB programs are given with many of the textbook examples to illustrate the computer calculations of the results of the example. These programs are easily modified for the homework problems.

To further assist instructors using this book, a set of PowerPoint slides and a manual containing problem solutions has been developed. The authors feel that the problems at the end of the chapters are an indispensable part of the text, and should be fully utilized by all who study this book. Requests for both the problem solutions and PowerPoint slides can be sent directly to the publisher.

At Auburn University, three courses based on the controls portion of this text, Chapters 2–11, have been taught. Chapters 2–8 are covered in their entirety in a one-quarter four-credit-hour graduate course. Thus the material is also suitable for a three-semester-hour course and has been presented as such at North Carolina State University. These chapters have also been covered in twenty lecture hours of an undergraduate course, but with much of the material omitted. The topics not covered in this abbreviated presentation are state variables, the modified  $z$ -transform, nonsynchronous sampling, and closed-loop frequency response. A third course, which is a one-quarter three-credit course, requires one of the above courses as a prerequisite, and introduces the state variables of Chapter 2. Then the state-variable models of Chapter 4, and the modern design of Chapters 9–11, are covered in detail. In a recent offering at North Carolina State University, Chapters 2–11 were covered in a one-semester, three-credit-hour course using this new edition and the companion set of PowerPoint slides that are also available.

Finally, we gratefully acknowledge the many colleagues, graduate and undergraduate students, and staff members of the Electrical Engineering Department at Auburn University who contributed to the development of the first three editions of this book. In particular, we wish to thank Professor J. David Irwin, Electrical Engineering Department Head at Auburn University, for his aid and encouragement during those years. We would also like to acknowledge our colleagues and students in the Electrical and Computer Engineering Department at North Carolina State University for their contributions to and support for this fourth edition.

*Charles L. Phillips*  
Auburn University

*H. Troy Nagle*  
North Carolina State  
University

*Aranya Chakraborty*  
North Carolina State  
University

## GLOBAL ACKNOWLEDGMENTS

Pearson gratefully acknowledges and thanks the following people for their work on the Global Edition:

### Contributor

Nikhil Marriwala, *University Institute of Engineering and Technology, Kurukshetra*

### Reviewers

Lalu Seban, *National Institute of Technology, Silchar*

Tapas Kumar Saha, *National Institute of Technology, Durgapur*

Vitthal Shriray Bandal, *College of Engineering, Pune*

# 1

## Introduction

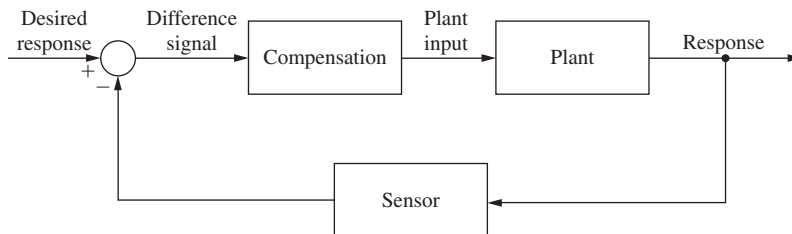
### 1.1 OVERVIEW

This book is concerned with the analysis and design of closed-loop physical systems that contain digital computers. The computers are placed within the system to modify the dynamics of the closed-loop system such that a *more satisfactory* system response is obtained.

A *closed-loop system* is one in which certain system forcing functions (inputs) are determined, at least in part, by the response (outputs) of the system (i.e., the input is a function of the output). A simple closed-loop system is illustrated in Fig. 1-1. The physical system (process) to be controlled is called the *plant*. Usually a system, called the *control actuator*, is required to drive the plant; in Fig. 1-1 the actuator has been included in the plant. The *sensor* (or sensors) measures the response of the plant, which is then compared to the desired response. This difference signal initiates actions that result in the actual response approaching the desired response, which drives the difference signal toward zero. Generally, an unacceptable closed-loop response occurs if the plant input is simply the difference between the desired response and actual response. Instead, this difference signal must be processed (filtered) by another physical system, which is called a *compensator*, a *controller*, or simply a *filter*. One problem of the control system designer is the design of the compensator.

An example of a closed-loop system is the case of a pilot landing an aircraft. For this example, in Fig. 1-1 the plant is the aircraft and the plant inputs are the pilot's manipulations of the various control surfaces and of the aircraft velocity. The pilot is the sensor, with his or her visual perceptions of position, velocity, instrument indications, and so on, and with his or her sense of balance, motion, and so on. The desired response is the pilot's concept of the desired flight path. The compensation is the pilot's manner of correcting perceived errors in flight path. Hence, for this example, the compensation, the sensor, and the generation of the desired response are all functions performed by the pilot. It is obvious from this example that the compensation must be a function of plant (aircraft) dynamics. A pilot trained only in a fighter aircraft is not qualified to land a large passenger aircraft, even if he or she can manipulate the controls.

We will consider systems of the type shown in Fig. 1-1, in which the sensor is an appropriate measuring instrument and the compensation function is performed by a digital computer.



**FIGURE 1-1** Closed-loop system.

The plant has dynamics; we will program the computer such that it has *dynamics* of the same nature as those of the plant. Furthermore, although generally we cannot choose the dynamics of the plant, we can choose those of the computer such that, in some sense, the dynamics of the closed-loop system are *satisfactory*. For example, if we are designing an automatic aircraft landing system, the landing must be safe, the ride must be acceptable to the pilot and to any passengers, and the aircraft cannot be unduly stressed.

Both classical and modern control techniques of analysis and design are developed in this book. Almost all control-system techniques developed are applicable to *linear time-invariant discrete-time* system models. A linear system is one for which the principle of superposition applies [1]. Suppose that the input of a system  $x_1(t)$  produces a response (output)  $y_1(t)$ , and the input  $x_2(t)$  produces the response  $y_2(t)$ . Then, if the system is linear, the principle of superposition applies and the input  $[a_1x_1(t) + a_2x_2(t)]$  will produce the output  $[a_1y_1(t) + a_2y_2(t)]$ , where  $a_1$  and  $a_2$  are any constants. All physical systems are inherently nonlinear; however, in many systems, if the system signals vary over a narrow range, the system responds in a linear manner. Even though the analysis and design techniques presented are applicable to linear systems only, certain nonlinear effects will be discussed.

When the parameters of a system are constant with respect to time, the system is called a *time-invariant* system. An example of a time-varying system is the booster stage of a space vehicle, in which fuel is consumed at a known rate; for this case, the mass of the vehicle decreases with time.

A *discrete-time* system has signals that can change values only at discrete instants of time. We will refer to systems in which all signals can change continuously with time as *continuous-time*, or *analog*, systems.

The compensator, or controller, in this book is a digital filter. The filter implements a transfer function. The design of transfer functions for digital controllers is the subject of Chapters 2 through 9 and 11. Once the transfer function is known, algorithms for its realization must be programmed on a digital computer. In Chapter 10 we introduce system identification methods to model the plant's dynamic behavior. In Chapter 12 we present several case studies in digital controls systems design.

Presented next in this chapter is an example digital control system. Then the equations describing three typical plants that appear in closed-loop systems are developed.

## 1.2 DIGITAL CONTROL SYSTEM

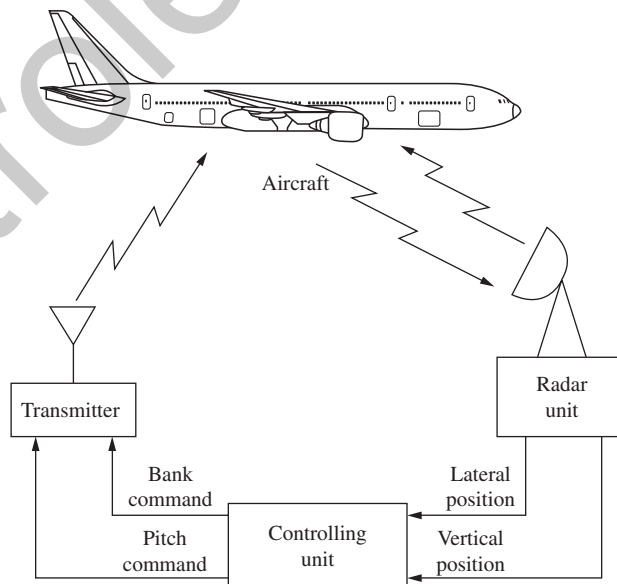
The basic structure of a digital control system will be introduced through the example of an automatic aircraft landing system. The system to be described is similar to the landing system that is currently operational on U.S. Navy aircraft carriers [2]. Only the simpler aspects of the system will be described.

The automatic aircraft landing system is depicted in Fig. 1-2. The system consists of three basic parts: the aircraft, the radar unit, and the controlling unit. During the operation of this control system, the radar unit measures the approximate vertical and lateral positions of the aircraft, which are then transmitted to the controlling unit. From these measurements, the controlling unit calculates appropriate pitch and bank commands. These commands are then transmitted to the aircraft autopilots, which in turn cause the aircraft to respond accordingly.

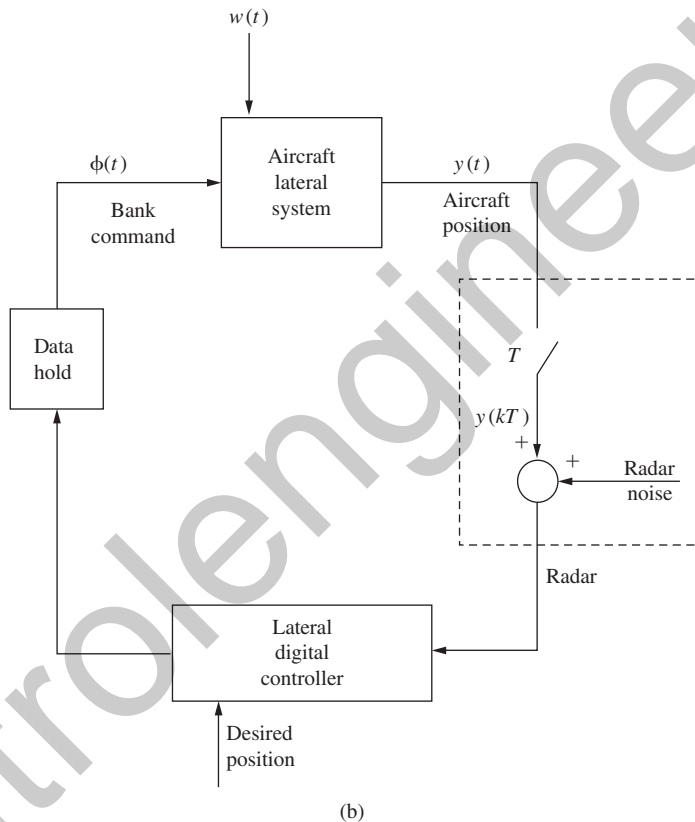
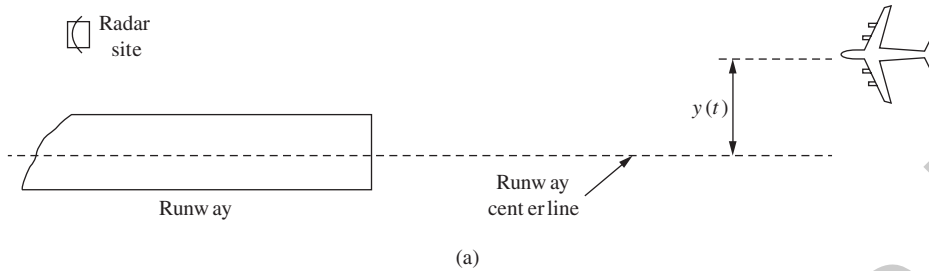
In Fig. 1-2 the controlling unit is a digital computer. The lateral control system, which controls the lateral position of the aircraft, and the vertical control system, which controls the altitude of the aircraft, are independent (decoupled). Thus the bank command input affects only the lateral position of the aircraft, and the pitch command input affects only the altitude of the aircraft. To simplify the treatment further, only the lateral control system will be discussed.

A block diagram of the lateral control system is given in Fig. 1-3. The aircraft lateral position,  $y(t)$ , is the lateral distance of the aircraft from the extended centerline of the runway. The control system attempts to force  $y(t)$  to zero. The radar unit measures  $y(t)$  every 0.05 s. Thus  $y(kT)$  is the sampled value of  $y(t)$ , with  $T = 0.05$  s and  $k = 0, 1, 2, 3, \dots$ . The digital controller processes these sampled values and generates the discrete bank commands  $\phi(kT)$ . The data hold, which is on board the aircraft, clamps the bank command  $\phi(t)$  constant at the last value received until the next value is received. Then the bank command is held constant at the new value until the following value is received. Thus the bank command is updated every  $T = 0.05$  s, which is called the *sample period*. The aircraft responds to the bank command, which changes the lateral position  $y(t)$ .

Two additional inputs are shown in Fig. 1-3. These are unwanted inputs, called *disturbances*, and we would prefer that they not exist. The first,  $w(t)$ , is the wind input, which certainly affects the position of the aircraft. The second disturbance input, labeled radar noise, is present since the radar cannot measure the exact position of the aircraft. This noise is the difference



**FIGURE 1-2** Automatic aircraft landing system.



**FIGURE 1-3** Aircraft lateral control system.

between the exact aircraft position and the measured position. Since no sensor is perfect, sensor noise is always present in a control system.

The design problem for this system is to maintain  $y(t)$  at a small level in the presence of the wind and radar-noise disturbances. In addition, the plane must respond in a manner that both is acceptable to the pilot and does not unduly stress the structure of the aircraft.

To effect the design, it is necessary to know the mathematical relationships between the wind input  $w(t)$ , the bank command input  $\phi(t)$ , and the lateral position  $y(t)$ . These mathematical relationships are referred to as the mathematical model, or simply the model, of the aircraft. For example, for the McDonnell-Douglas Corporation F4 aircraft, the model of lateral system is a ninth-order ordinary nonlinear differential equation [3]. For the case in which the bank command  $\phi(t)$  remains small in amplitude, the nonlinearities are not excited and the system model

described by this ninth-order ordinary nonlinear differential equation may be used for design purposes.

The task of the control system designer is to specify the processing to be accomplished in the digital controller. This processing will be a function of the ninth-order aircraft model, the expected wind input, the radar noise, the sample period  $T$ , and the desired response characteristics. Various methods of digital controller design are developed in Chapters 8, 9, and 11.

The development of the ninth-order model of the aircraft is beyond the scope of this book. In addition, this model is too complex to be used in an example in this book. Hence, to illustrate the development of models of physical systems, the mathematical models of four simple, but common, control-system plants will be developed later in this chapter. Two of the systems relate to the control of position, the third relates to temperature control, and the fourth one describes control of electrical power in single-machine infinite bus models of power systems. In addition, Chapter 10 presents procedures for determining the model of a physical system from input–output measurements of the system.

### 1.3 THE CONTROL PROBLEM

We may state the control problem as follows. A physical system or process is to be accurately controlled through closed-loop, or feedback, operation. An output variable (signal), called the response, is adjusted as required by an error signal. The error signal is a measure of the difference between the system response, as determined by a sensor, and the desired response.

Generally, a controller, or filter, is required to process the error signal in order that certain control criteria, or specifications, will be satisfied. The criteria may involve, but not be limited to:

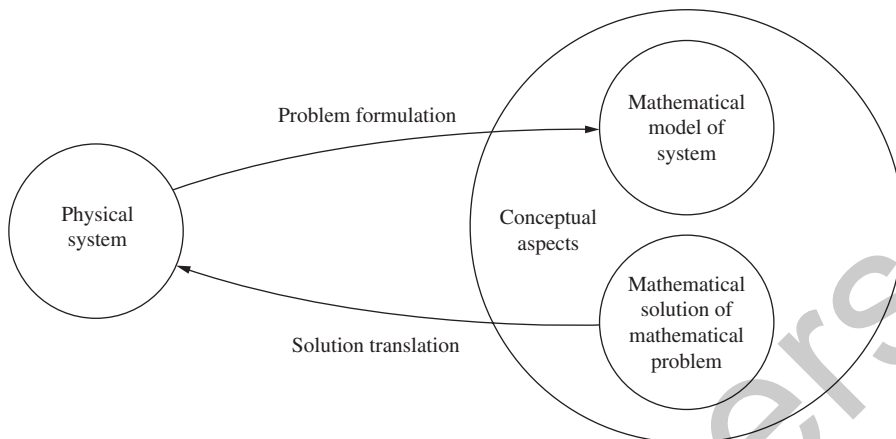
1. Disturbance rejection
2. Steady-state errors
3. Transient response
4. Sensitivity to parameter changes in the plant

Solving the control problem will generally involve:

1. Choosing sensors to measure the required feedback signals
2. Choosing actuators to drive the plant
3. Developing the plant, sensor, and actuator models (equations)
4. Designing the controller based on the developed models and the control criteria
5. Evaluating the design analytically, by simulation, and finally, by testing the physical system
6. Iterating this procedure until a satisfactory physical-system response results

Because of inaccuracies in the mathematical models, the initial tests on the physical system may not be satisfactory. The controls engineer must then iterate this design procedure, using *all* tools available, to improve the system. Intuition, developed while experimenting with the physical system, usually plays an important part in the design process.

Fig. 1-4 illustrates the relationship of mathematical analysis and design to physical-system design procedures [4]. In this book, all phases shown in the Fig. are discussed, but the emphasis is necessarily on the conceptual part of the procedures—the application of mathematical concepts to mathematical models. In practical design situations, however, the major difficulties are in formulating the problem mathematically and in translating the mathematical solution back to the physical world. Many iterations of the procedures shown in Fig. 1-4 are usually required in practical situations.



**FIGURE 1-4** Mathematical solutions for physical systems.

Depending on the system and the experience of the designer, some of the steps listed earlier may be omitted. In particular, many control systems are implemented by choosing standard forms of controllers and experimentally determining the parameters of the controller; a specified step-by-step procedure is applied directly to the physical system, and no mathematical models are developed. This type of procedure works very well for certain control systems. For other systems, it does not. For example, a control system for a space vehicle cannot be designed in this manner; this system must perform satisfactorily the first time it is activated.

In this book mathematical procedures are developed for the analysis and design of control systems. The techniques developed may or may not be of value in the design of a particular control system. However, standard controllers are utilized in the developments in this book. Thus the analytical procedures develop the concepts of control system design and indicate applications of each of the standard controllers.

## 1.4 SATELLITE MODEL

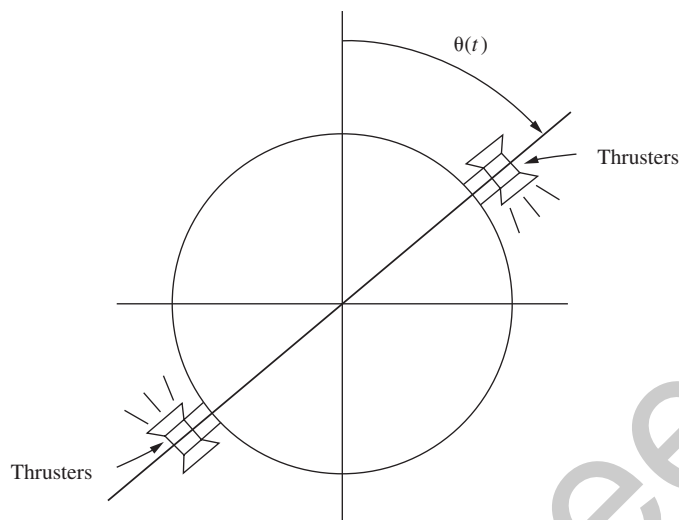
As the first example of the development of the mathematical model of a physical system, we will consider the attitude control system of a satellite. Assume that the satellite is spherical and has the thruster configuration shown in Fig. 1-5. Suppose that  $\theta(t)$  is the yaw angle of the satellite. In addition to the thrusters shown, thrusters will also control the pitch angle and the roll angle, giving complete three-axis control of the satellite. We will consider only the yaw-axis control systems, whose purpose is to control the angle  $\theta(t)$ .

For the satellite, the thrusters, when active, apply a torque  $\tau(t)$ . The torque of the two active thrusters shown in Fig. 1-5 tends to reduce  $\theta(t)$ . The other two thrusters shown tend to increase  $\theta(t)$ .

Since there is essentially no friction in the environment of a satellite, and assuming the satellite to be rigid, we can write

$$J \frac{d^2 \theta(t)}{dt^2} = \tau(t) \quad (1-1)$$





**FIGURE 1-5** Satellite.

where  $J$  is the satellite's moment of inertia about the yaw axis. We now derive the transfer function by taking the Laplace transform of (1-1):

$$Js^2\Theta(s) = T(s) = \mathcal{L}[\tau(t)] \quad (1-2)$$

Initial conditions are ignored when deriving transfer functions. Equation (1-2) can be expressed as

$$\frac{\Theta(s)}{T(s)} = G_p(s) = \frac{1}{Js^2} \quad (1-3)$$

The ratio of the Laplace transforms of the output variable  $[\theta(t)]$  to input variable  $[\tau(t)]$  is called the plant transfer function, and is denoted here as  $G_p(s)$ . A brief review of the Laplace transform is given in Appendix V.

The model of the satellite may be specified by either the second-order *differential equation* of (1-1) or the second-order *transfer function* of (1-3). A third model is the state-variable model, which we will now develop. Suppose that we define the variables  $x_1(t)$  and  $x_2(t)$  as

$$x_1(t) = \theta(t) \quad (1-4)$$

$$x_2(t) = \dot{x}_1(t) = \dot{\theta}(t) \quad (1-5)$$

where  $\dot{x}_1(t)$  denotes the derivative of  $x_1(t)$  with respect to time. Then, from (1-1) and (1-5),

$$\dot{x}_2(t) = \ddot{\theta}(t) = \frac{1}{J}\tau(t) \quad (1-6)$$

where  $\ddot{\theta}(t)$  is the second derivative of  $\theta(t)$  with respect to time.

We can now write (1-5) and (1-6) in vector-matrix form (see Appendix IV):

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{J} \end{bmatrix} \tau(t) \quad (1-7)$$

In this equation,  $x_1(t)$  and  $x_2(t)$  are called the state variables. Hence we may specify the model of the satellite in the form of (1-1), or (1-3), or (1-7). State-variable models of analog systems are considered in greater detail in Chapter 4.

## 1.5 SERVO MOTOR SYSTEM MODEL

In this section the model of a servo system (a positioning system) is derived. An example of this type of system is an antenna tracking system. In this system, an electric motor is utilized to rotate a radar antenna that tracks an aircraft automatically. The error signal, which is proportional to the difference between the pointing direction of the antenna and the line of sight to the aircraft, is amplified and drives the motor in the appropriate direction so as to reduce this error.

A dc motor system is shown in Fig. 1-6. The motor is armature controlled with a constant field. The armature resistance and inductance are  $R_a$  and  $L_a$ , respectively. We assume that the inductance  $L_a$  can be ignored, which is the case for many servomotors. The motor back emf  $e_m(t)$  is given by [5]

$$e_m(t) = K_b \omega(t) = K_b \frac{d\theta(t)}{dt} \quad (1-8)$$

where  $\theta(t)$  is the shaft position,  $\omega(t)$  is the shaft angular velocity, and  $K_b$  is a motor-dependent constant. The total moment of inertia connected to the motor shaft is  $J$ , and  $B$  is the total viscous friction. Letting  $\tau(t)$  be the torque developed by the motor, we write

$$\tau(t) = J \frac{d^2\theta(t)}{dt^2} + B \frac{d\theta(t)}{dt} \quad (1-9)$$

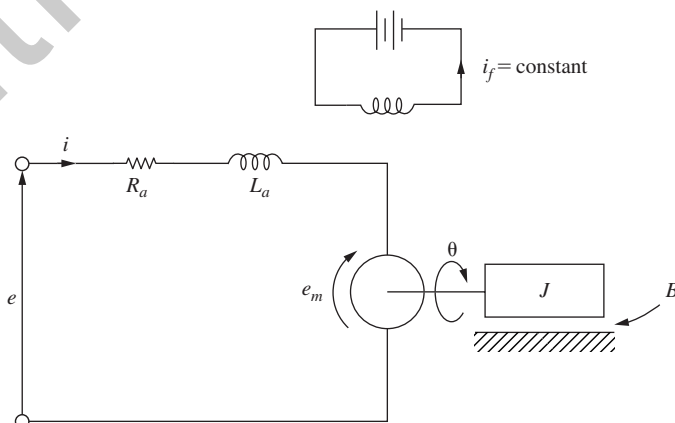


FIGURE 1-6 Servomotor system.

The developed torque for this motor is given by

$$\tau(t) = K_T i(t) \quad (1-10)$$

where  $i(t)$  is the armature current and  $K_T$  is a parameter of the motor. The final equation required is the voltage equation for the armature circuit:

$$e(t) = i(t)R_a + e_m(t) \quad (1-11)$$

These four equations may be solved for the output  $\theta(t)$  as a function of the input  $e(t)$ . First, from (1-11) and (1-8),

$$i(t) = \frac{e(t) - e_m(t)}{R_a} = \frac{e(t)}{R_a} - \frac{K_b}{R_a} \frac{d\theta(t)}{dt} \quad (1-12)$$

Then, from (1-9), (1-10), and (1-12),

$$\tau(t) = K_T i(t) = \frac{K_T}{R_a} e(t) - \frac{K_T K_b}{R_a} \frac{d\theta(t)}{dt} = J \frac{d^2\theta(t)}{dt^2} + B \frac{d\theta(t)}{dt} \quad (1-13)$$

This equation may be written as

$$J \frac{d^2\theta(t)}{dt^2} + \frac{BR_a + K_T K_b}{R_a} \frac{d\theta(t)}{dt} = \frac{K_T}{R_a} e(t) \quad (1-14)$$

which is the desired model. This model is second order; if the armature inductance cannot be neglected, the model is third order [6].

Next we take the Laplace transform of (1-14) and solve for the transfer function:

$$\frac{\Theta(s)}{E(s)} = G_p(s) = \frac{K_T/R_a}{Js^2 + \frac{BR_a + K_T K_b}{R_a}s} = \frac{K_T/JR_a}{s\left(s + \frac{BR_a + K_T K_b}{JR_a}\right)} \quad (1-15)$$

Many of the examples of this book are based on this transfer function.

The state-variable model of this system is derived as in the preceding section.

Let

$$\begin{aligned} x_1(t) &= \theta(t) \\ x_2(t) &= \dot{\theta}(t) = \dot{x}_1(t) \end{aligned} \quad (1-16)$$

Then, from (1-14),

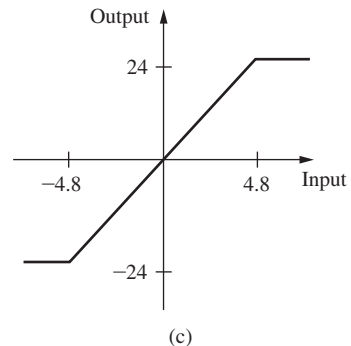
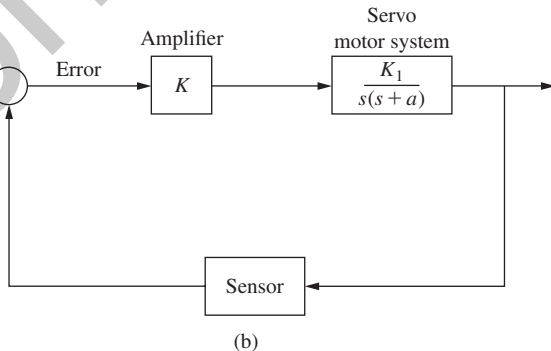
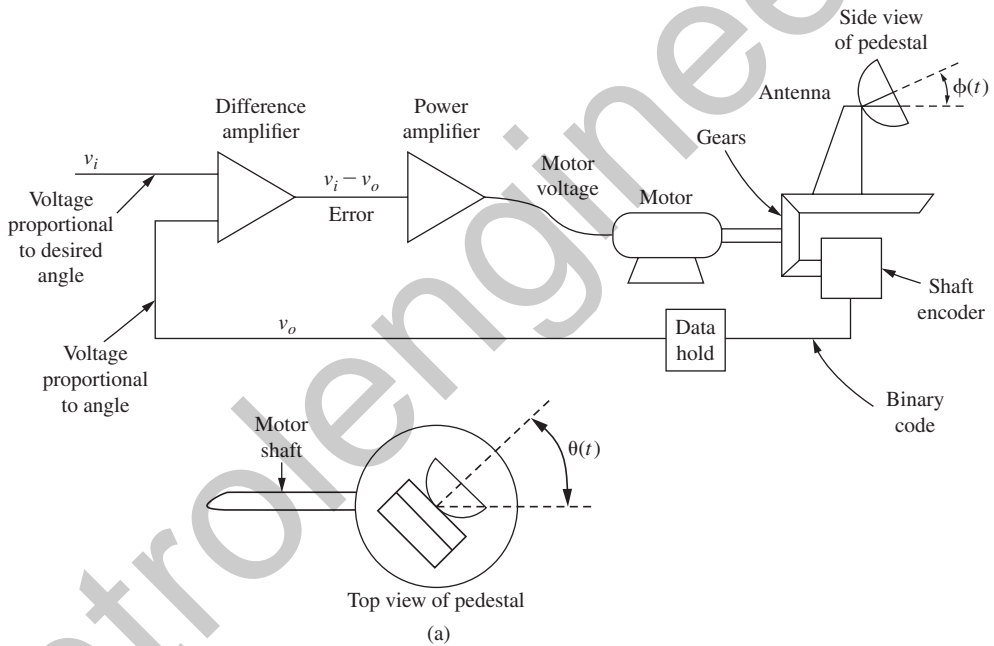
$$\dot{x}_2(t) = \ddot{\theta}(t) = -\frac{BR_a + K_T K_b}{JR_a} x_2(t) + \frac{K_T}{JR_a} e(t) \quad (1-17)$$

Hence the state equations may be written as

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{BR_a + K_T K_b}{JR_a} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_T}{JR_a} \end{bmatrix} e(t) \quad (1-18)$$

### Antenna Pointing System

We define a *servomechanism*, or more simply, a *servo*, as a system in which mechanical position is controlled. Two servo systems, which in this case form an antenna pointing system, are illustrated in Fig. 1-7. The top view of the pedestal illustrates the *yaw-axis* control system. The yaw angle,  $\theta(t)$ , is controlled by the electric motor and gear system (the control actuator) shown in the side view of the pedestal.



**FIGURE 1-7** Servo control system.

The pitch angle,  $\phi(t)$ , is shown in the side view. This angle is controlled by a motor and gear system within the pedestal; this actuator is not shown.

We consider only the yaw-axis control system. The electric motor rotates the antenna and the sensor, which is a digital shaft encoder [7]. The output of the encoder is a binary number that is proportional to the angle of the shaft. For this example, a digital-to-analog converter is used to convert the binary number to a voltage  $v_o(t)$  that is proportional to the angle of rotation of the shaft. Later we consider examples in which the binary number is transmitted directly to a digital controller.

In Fig. 1-7(a) the voltage  $v_o(t)$  is directly proportional to the yaw angle of the antenna, and the voltage  $v_i(t)$  is directly proportional (same proportionality constant) to the desired yaw angle. If the actual yaw angle and the desired yaw angle are different, the error voltage  $e(t)$  is nonzero. This voltage is amplified and applied to the motor to cause rotation of the motor shaft in the direction that reduces the error voltage.

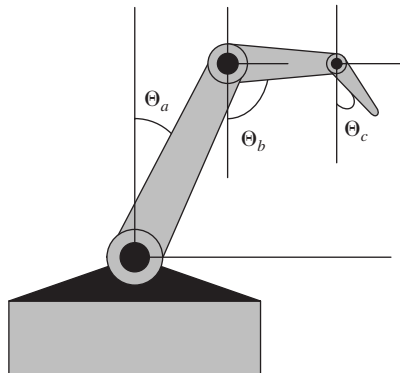
The system *block diagram* is given in Fig. 1-7(b). Since the error signal is normally a low-power signal, a power amplifier is required to drive the motor. However, this amplifier introduces a nonlinearity, since an amplifier has a maximum output voltage and can be saturated at this value. Suppose that the amplifier has a gain of 5 and a maximum output of 24 V. Then the amplifier input–output characteristic is as shown in Fig. 1-7(c). The amplifier saturates at an input of 4.8 V; hence, for an error signal larger than 4.8 V in magnitude, the system is nonlinear.

In many control systems, we go to great lengths to ensure that the system operation is confined to linear regions. In other systems, we purposely design for nonlinear operation. For example, in this servo system, we must apply maximum voltage to the motor to achieve maximum speed of response. Thus for large error signals we would have the amplifier saturated in order to achieve a fast response.

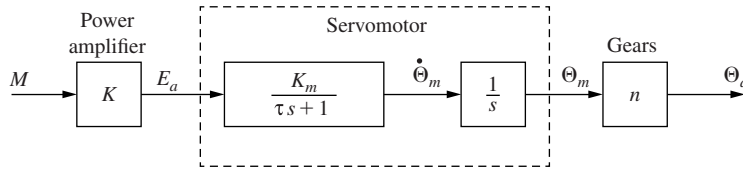
The analysis and design of nonlinear systems is beyond the scope of this book; we will always assume that the system under consideration is operating in a linear mode.

## Robotic Control System

A line drawing of an industrial robot is shown in Fig. 1-8. The basic element of the control system for each joint of many robots is a servomotor. We take the usual approach of considering each joint of the robot as a simple servomechanism, and ignore the movements of the other joints in the arm. Although this approach is simple in terms of analysis and design, the result is often less than desirable control of the joint [8].



**FIGURE 1-8** Schematic diagram of a robotic arm with three angles of motion.

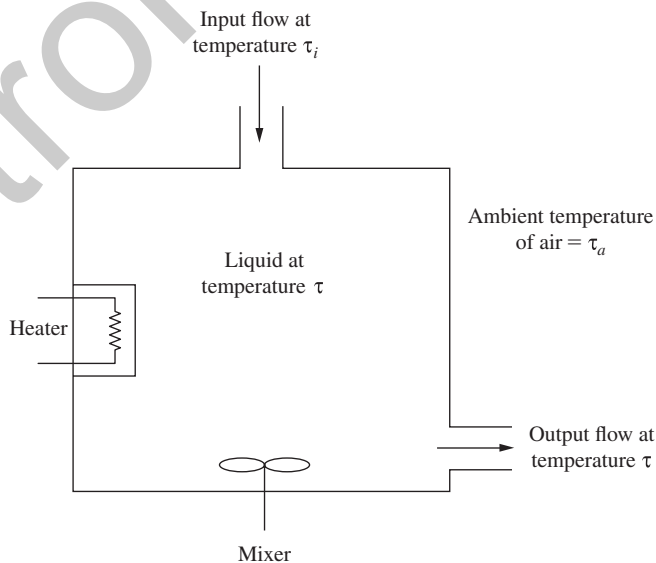


**FIGURE 1-9** Model of robot arm joint.

The model of a single robot arm joint is given in Fig. 1-9, where the second-order model of the servomotor is assumed. In addition, it is assumed that the arm is attached to the motor through gears, with a gear ratio of  $n$ . If the armature inductance of the motor cannot be ignored, the model is third order [8]. In this model,  $E_a(s)$  is the armature voltage, and is used to control the position of the arm. The input signal  $M(s)$  is assumed to be from a digital computer, and the power amplifier  $K$  is required since a computer output signal cannot drive the motor. The angle of the motor shaft is  $\Theta_m(s)$ , and the angle of the arm is  $\Theta_a(s)$ . Same holds for the two other arm angles  $\Theta_b(s)$  and  $\Theta_c(s)$ . As described above, the inertia and friction of both the gears and the arm are included in the servomotor model, and hence the model shown is the complete model of the robot joint. This model will be used in several problems that appear at the ends of the chapters.

### 1.6 TEMPERATURE CONTROL SYSTEM

As a third example of modeling, a thermal system will be considered. It is desired to control the temperature of a liquid in a tank. Liquid is flowing out at some rate, being replaced by liquid at temperature  $\tau_i(t)$  as shown in Fig. 1-10. A mixer agitates the liquid such that the liquid temperature can be assumed uniform at a value  $\tau(t)$  throughout the tank. The liquid is heated by an electric heater.



**FIGURE 1-10** Thermal system.

We first make the following definitions:

$q_e(t)$  = heat flow supplied by the electric heater

$q_i(t)$  = heat flow via liquid entering the tank

$q_l(t)$  = heat flow into the liquid

$q_o(t)$  = heat flow via liquid leaving the tank

$q_s(t)$  = heat flow through the tank surface

By the conservation of energy, heat added to the tank must equal that stored in the tank plus that lost from the tank. Thus

$$q_e(t) + q_i(t) = q_l(t) + q_o(t) + q_s(t) \quad (1-19)$$

Now [9]

$$q_l(t) = C \frac{d\tau(t)}{dt} \quad (1-20)$$

where  $C$  is the thermal capacity of the liquid in the tank. Letting  $v(t)$  equal the flow into and out of the tank (assumed equal) and  $H$  equal the specific heat of the liquid, we can write

$$q_i(t) = v(t)H\tau_i(t) \quad (1-21)$$

and

$$q_o(t) = v(t)H\tau(t) \quad (1-22)$$

Let  $\tau_a(t)$  be the ambient temperature outside the tank and  $R$  be the thermal resistance to heat flow through the tank surface. Then

$$q_s(t) = \frac{\tau(t) - \tau_a(t)}{R} \quad (1-23)$$

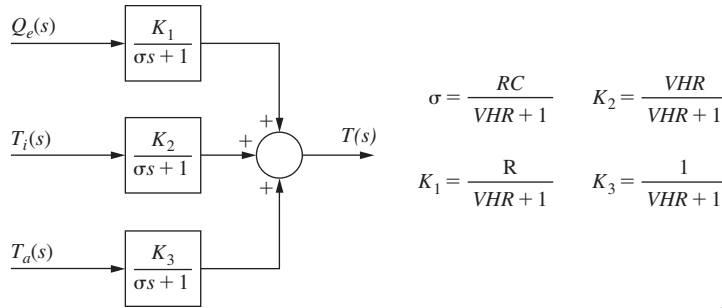
Substituting (1-20) through (1-23) into (1-19) yields

$$q_e(t) + v(t)H\tau_i(t) = C \frac{d\tau(t)}{dt} + v(t)H\tau(t) + \frac{\tau(t) - \tau_a(t)}{R}$$

We now make the assumption that the flow  $v(t)$  is constant with the value  $V$ ; otherwise, the last differential equation is time-varying. Then

$$q_e(t) + VH\tau_i(t) = C \frac{d\tau(t)}{dt} + VH\tau(t) + \frac{\tau(t) - \tau_a(t)}{R} \quad (1-24)$$

This model is a first-order linear differential equation with constant coefficients. In terms of a control system,  $q_e(t)$  is the control input signal,  $\tau_i(t)$  and  $\tau_a(t)$  are disturbance input signals, and  $\tau(t)$  is the output signal.



**FIGURE 1-11** Block diagram of a thermal system.

Taking the Laplace of (1-24) and solving for  $T(s) = \mathcal{L}[\tau(t)]$  yields

$$T(s) = \frac{Q_e(s)}{Cs + VH + (1/R)} + \frac{VHT_i(s)}{Cs + VH + (1/R)} + \frac{(1/R)T_a(s)}{Cs + VH + (1/R)} \quad (1-25)$$

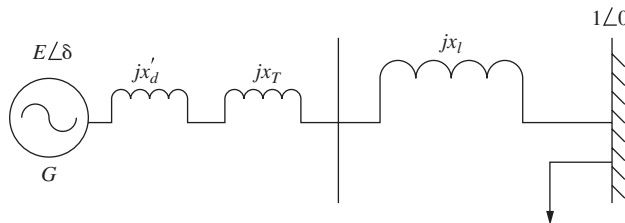
Different configurations may be used to express (1-25) as a block diagram; one is given in Fig. 1-11.

If we ignore the disturbance inputs, the transfer-function model of the system is simple and first order. However, at some step in the control system design the disturbances must be considered. Quite often a major specification in a control system design is the minimization of system response to disturbance inputs.

The model developed in this section also applies directly to the control of the air temperature in an oven or a test chamber. For many of these systems, no air is introduced from the outside; hence the disturbance input  $q_i(t)$  is zero. Of course, the parameters for the liquid in (1-25) are replaced with those for air.

### 1.7 SINGLE-MACHINE INFINITE BUS POWER SYSTEM

A single-machine infinite bus (SMIB) power system, as shown in Fig. 1-12, is often used as the starting model for understanding dynamics and stability of large power grids. The system consists of a synchronous generator  $G$ , which in many cases may represent the equivalent model of a larger area containing multiple synchronous machines inside it, supplying electrical power across a lossless transmission line to a load connected to a fixed or stationary point, commonly referred to as the infinite bus. The relevance of the term “infinite” is that this bus can also be viewed as a generator with theoretically infinite inertia, implying that the voltage and phase



**FIGURE 1-12** Single-machine infinite bus power system.



angle at this bus always remain static or constant, and thereby serve as a reference for quantitative analysis of the phase angle oscillations of the other generator(s) in the system. Therefore, for convenience, it is always assumed that the voltage at the infinite bus is 1 per unit, while the phase angle is 0 degrees. To derive the model of the SMIB power system, we next introduce the following set of symbols:

- $\delta$  phase angle of the synchronous generator (radians)
- $\omega$  angular speed (radian/sec)
- $\omega_s$  synchronous speed, equal to  $120\pi$  rad/s for a 60 Hz system
- $E$  internal constant voltage of the generator (per unit)
- $x'_d$  direct-axis salient reactance (per unit)
- $x_T$  transformer reactance (per unit)
- $x_l$  transmission line reactance (per unit)
- $d$  damping constant
- $M$  generator electro-mechanical inertia
- $P_m$  mechanical power input from turbine to generator (per unit)
- $P_e$  electrical power output from generator to infinite bus (per unit)

For details of the physical meanings of the above notations please see [10]. The dynamic model of the SMIB system can be written by applying Newton's second law of motion as

$$\begin{aligned} \dot{\delta} &= \omega - \omega_s \\ M\dot{\omega} &= P_m - P_e - d\omega \end{aligned} \quad (1-26)$$

which implies that the imbalance between input and output power flow causes the rotor of the synchronous generator to accelerate. From electric circuit law, the total complex power flowing from  $G$  to the infinite bus can be written as

$$P = (E\angle\delta)\tilde{I}^* \quad (1-27)$$

where,  $E\angle\delta = E(\cos\delta + j\sin\delta)$ ,  $\tilde{I}$  is the current phasor which is flowing out of the machine, and the superscript \* means complex conjugate. From Kirchoff's law this current can be written as

$$\tilde{I} = \frac{E\angle\delta - 1\angle 0}{j(x'_d + x_T + x_l)} \quad (1-28)$$

where  $1\angle 0$  is the voltage phasor at the infinite bus, while the expression in the denominator denotes the total reactance between the generator and the infinite bus. For simplicity of notation let us denote  $x = x'_d + x_T + x_l$ . Then the expression for the current phasor can be simplified as

$$\tilde{I} = \frac{E \cos \delta - 1 + jE \sin \delta}{jx} \quad (1-29)$$

from which  $P$ , after a few simple calculations, can be shown as

$$P = \frac{E}{x} (\sin \delta + j(E - \cos \delta)) \quad (1-30)$$

Since  $P_e$  in (1-26) represents only the real part of  $P$ , therefore another way to write (1-26) is

$$\begin{aligned}\dot{\delta} &= \omega - \omega_s \\ M\dot{\omega} &= P_m - \frac{E}{x} \sin \delta - d\omega\end{aligned}\quad (1-31)$$

Equation (1-31) gives the continuous-time state-variable model for the SMIB system. The model, however, is nonlinear because of the  $\sin \delta$  term in the RHS of the second equation. Hence, we linearize this model across an equilibrium of  $(\delta = \delta_0, \omega = \omega_s)$  to obtain a linear time-invariant model of the form

$$\begin{bmatrix} \Delta \dot{\delta} \\ \Delta \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{E \cos \delta_0}{Mx} & -\frac{d}{M} \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta \omega \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix} \Delta P_m \quad (1-32)$$

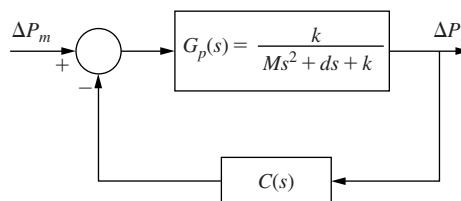
where  $\Delta$  stands for the small-signal perturbation of the corresponding states and inputs from their respective equilibrium values. The output can be considered as the change in electric power  $P_e$  as

$$\Delta P_e = \frac{E \cos \delta_0}{x} \Delta \delta = k \Delta \delta \quad (1-33)$$

The transfer function for the system (1-32) and (1-33) with input  $\Delta P_m$  and output  $\Delta P_e$  can be derived as

$$G_p(s) = \frac{k}{Ms^2 + ds + k} \quad (1-34)$$

$G_p(s)$  gives the open-loop transfer function between the small-signal mechanical power input and the electrical power output of the synchronous machine. It can be seen that the steady-state gain ( $s = 0$ ) of  $G_p(s)$  is 1, which means that in steady state the mechanical power input to the machine must exactly balance its electrical power output. The transient response of the output for a unit change in the input, however, may not be satisfactory to the user as it is. Therefore, one may design an output-feedback controller  $C(s)$  to control the transient response of the electrical power, as shown in the block diagram in Fig. 1-13.  $C(s)$  must be designed so that the steady-state gain of the closed-loop transfer function is 1. Depending on the values of  $M$ ,  $d$ , and  $k$ , the open-loop response may not be satisfactory in terms of damping, percent peak overshoot, settling time, etc. The controller  $C(s)$  can be designed to achieve these transient performance specifications.



**FIGURE 1-13** Block diagram of a closed-loop SMIB power system model.

## 1.8 SUMMARY

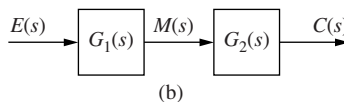
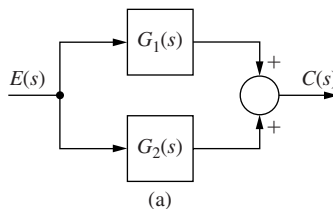
In this chapter we have introduced the concepts of a closed-loop control system. Next, models of four physical systems were discussed. First, a model of a satellite was derived. Next, the model of a servomotor was developed; then two examples, an antenna pointing system and a robot arm, were discussed. Next, a model was developed for control of the temperature of a tank of liquid. Finally, a model for a single-machine infinite bus (SMIB) power system was presented. These systems are continuous time, and generally, the Laplace transform is used in the analysis and design of these systems. In the next chapter we extend the concepts of this chapter to a system controlled by a digital computer and introduce some of the mathematics required to analyze and design this type of system.

## References

- [1] M. L. Dertouzos, M. Athans, R. N. Spann, and S. J. Mason, *Systems, Networks, and Computation: Basic Concepts*. Huntington, NY: R.E. Krieger Publishing Co., Inc., 1979.
- [2] R. F. Wigginton, "Evaluation of OPS-II Operational Program for the Automatic Carrier Landing System," Naval Electronic Systems Test and Evaluation Facility, Saint Inigoes, MD, 1971.
- [3] C. L. Phillips, E. R. Graf, and H. T. Nagle, Jr., "MATCALs Error and Stability Analysis," Report AU-EE-75-2080-1, Auburn University, Auburn, AL, 1975.
- [4] W. A. Gardner, *Introduction to Random Processes with Applications to Signals and Systems*, 2nd ed. New York: McGraw-Hill Publishing Company, 1990.
- [5] A. E. Fitzgerald, C. Kingsley, and S. D. Umans, *Electric Machinery*, 6th ed. New York: McGraw-Hill Publishing Company, 2003.
- [6] C. L. Phillips and J. Parr, *Feedback Control Systems*, 5th ed. Upper Saddle River, NJ: Prentice-Hall, 2011.
- [7] C. W. deSilva, *Control Sensors and Actuators*. Upper Saddle River, NJ: Prentice Hall, 1989.
- [8] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. New York: McGraw-Hill Publishing Company, 1987.
- [9] J. D. Trimmer, *Response of Physical Systems*. New York: John Wiley & Sons, Inc., 1950.
- [10] P. M. Anderson and A. A. Fouad, *Power System Stability and Control*, 2nd ed. Wiley, 2008.

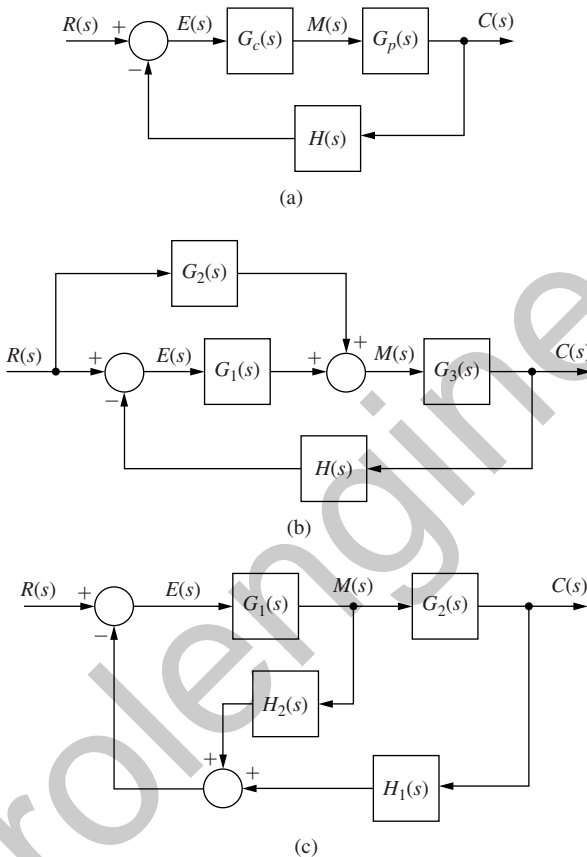
## Problems

- 1.1-1. (a) Show that the transfer function of two systems in parallel, as shown in Fig. P1.1-l(a), is equal to the sum of the transfer functions.
- (b) Show that the transfer function of two systems in series (cascade), as shown in Fig. P1.1-l(b), is equal to the product of the transfer functions.



**FIGURE P1.1-1** Systems for Problem 1.1-1.

- 1.1-2.** By writing algebraic equations and eliminating variables, calculate the transfer function  $C(s)/R(s)$  for the system of:
- (a) Figure P1.1-2(a).
  - (b) Figure P1.1-2(b).
  - (c) Figure P1.1-2(c).



**FIGURE P1.1-2** Systems for Problem 1.1-2.

- 1.1-3.** Use Mason's gain formula of Appendix II to verify the results of Problem 1.1-2 for the system of:
- (a) Figure P1.1-2(a).
  - (b) Figure P1.1-2(b).
  - (c) Figure P1.1-2(c).

**1.1-4.** A feedback control system is illustrated in Fig. P1.1-4. The plant transfer function is given by

$$G_p(s) = \frac{4}{0.3s + 1}$$

- (a) Write the differential equation of the plant. This equation relates  $c(t)$  and  $m(t)$ .
- (b) Modify the equation of part (a) to yield the system differential equation; this equation relates  $c(t)$  and  $r(t)$ . The compensator and sensor transfer functions are given by

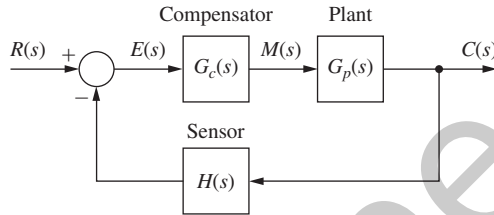
$$G_c(s) = 20, \quad H(s) = 1$$

- (c) Derive the system transfer function from the results of part (b).  
 (d) It is shown in Problem 1.1-2(a) that the closed-loop transfer function of the system of Fig. P1.1-4 is given by

$$\frac{C(s)}{R(s)} = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)H(s)}$$

Use this relationship to verify the results of part (c).

- (e) Recall that the transfer-function pole term  $(s + a)$  yields a time constant  $\tau = 1/a$ , where  $a$  is real. Find the time constants for both the open-loop and closed-loop systems.



**FIGURE P1.1-4** Feedback control system.

- 1.1-5.** Repeat Problem 1.1-4 with the transfer functions

$$G_c(s) = 4, \quad G_p(s) = \frac{3s + 5}{s^2 + 4s + 4}, \quad H(s) = 1$$

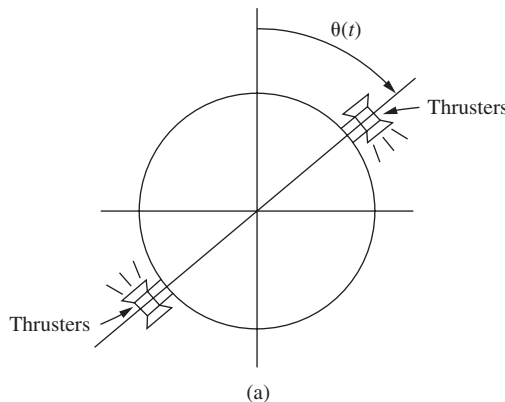
For part (e), recall that the transfer-function underdamped pole term  $[(s + a)^2 + b^2]$  yields a time constant,  $\tau = 1/a$ .

- 1.1-6.** Repeat Problem 1.1-4 with the transfer functions

$$G_c(s) = 4, \quad G_p(s) = \frac{4}{s^2 + 2s + 1}$$

- 1.4-1.** The satellite of Section 1.4 is connected in the closed-loop control system shown in Fig. P1.4-1. The torque is directly proportional to the error signal.

- (a) Derive the transfer function  $\Theta(s)/\Theta_c(s)$ , where  $\theta(t) = \mathcal{L}^{-1}[\Theta(s)]$  is the commanded attitude angle.  
 (b) The state equations for the satellite are derived in Section 1.4. Modify these equations to model the closed-loop system of Fig. P1.4-1.



**FIGURE P1.4-1** Satellite control system.

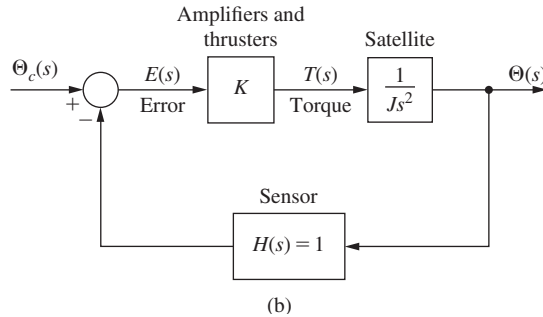


FIGURE P1.4-1 (continued)

- 1.4-2. (a) In the system of Problem 1.4-1,  $J = 0.6$  and  $K = 12.4$ , in appropriate units. The attitude of the satellite is initially at  $0^\circ$ . At  $t = 0$ , the attitude is commanded to  $40^\circ$ ; that is, a  $40^\circ$  step is applied at  $t = 0$ . Find the response  $\theta(t)$ .
- (b) Repeat part (a), with the initial conditions  $\theta(0) = 10^\circ$  and  $\dot{\theta}(0) = 30^\circ/\text{s}$ . Note that we have assumed that the units of time for the system is seconds.
- (c) Verify the solution in part (b) by first checking the initial conditions and then substituting the solution into the system differential equation.
- 1.4-3. The input to the satellite system of Fig. P1.4-1 is a step function  $\theta_c(t) = 4u(t)$  in degrees. As a result, the satellite angle  $\theta(t)$  varies sinusoidally at a frequency of 20 cycles per minute. Find the amplifier gain  $K$  and the moment of inertia  $J$  for the system, assuming that the units of time in the system differential equation are seconds.
- 1.4-4. The satellite control system of Fig. P1.4-1 is not usable, since the response to any excitation includes an undamped sinusoid. The usual compensation for this system involves measuring the angular velocity  $d\theta(t)/dt$ . The feedback signal is then a linear sum of the position signal  $\theta(t)$  and the velocity signal  $d\theta(t)/dt$ . This system is depicted in Fig. P1.4-4, and is said to have *rate feedback*.
- (a) Derive the transfer function  $\Theta(s)/\Theta_c(s)$  for this system.
- (b) The state equations for the satellite are derived in Section 1.4. Modify these equations to model the closed-loop system of Fig. P1.4-4.
- (c) The state equations in part (b) can be expressed as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\theta_c(t)$$

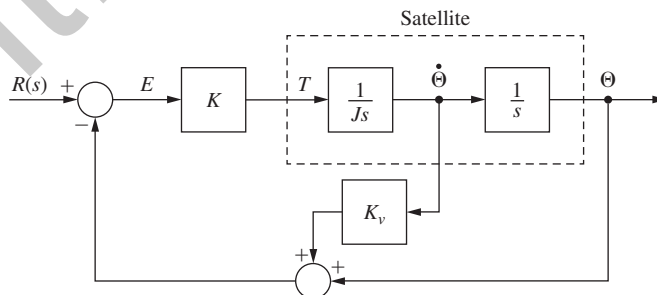


FIGURE P1.4-4 Satellite control system with rate feedback.

The system characteristic equation is

$$|s\mathbf{I} - \mathbf{A}| = 0$$

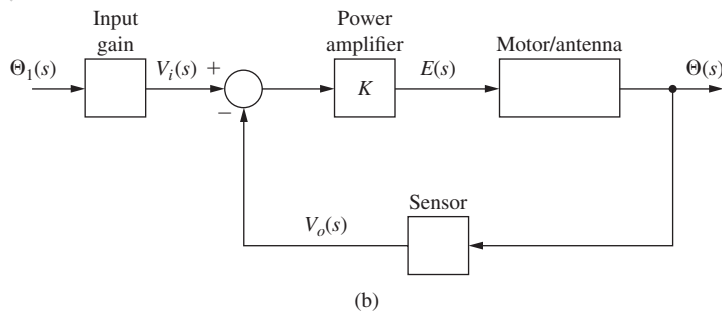
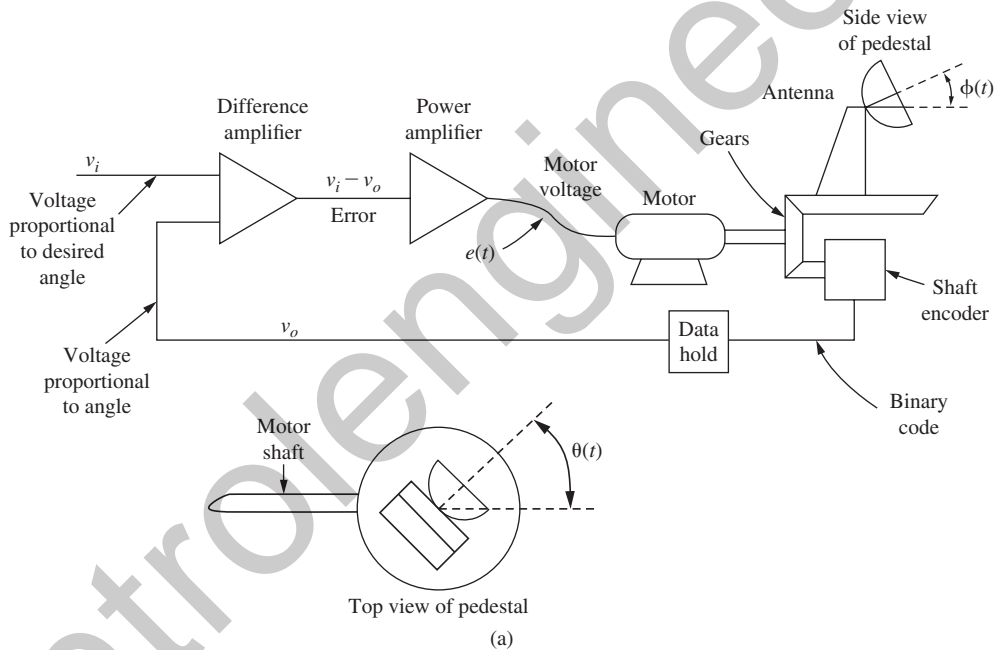
Show that  $|s\mathbf{I} - \mathbf{A}|$  in part (b) is equal to the transfer function denominator in part (a).

- 1.5-1.** The antenna positioning system described in Section 1.5 is shown in Fig. P1.5-1. In this problem we consider the yaw angle control system, where  $\theta(t)$  is the yaw angle. Suppose that the gain of the power amplifier is 5 V/V, and that the gear ratio and the angle sensor (the shaft encoder and the data hold) are such that

$$v_o(t) = 0.02\theta(t)$$

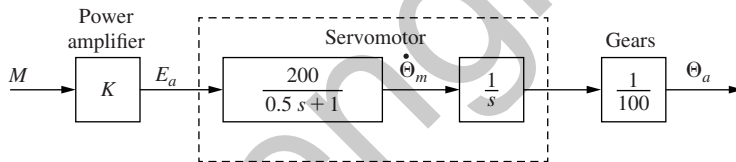
where the units of  $v_o(t)$  are volts and of  $\theta(t)$  are degrees. Let  $e(t)$  be the input voltage to the motor; the transfer function of the motor pedestal is given as

$$\frac{\Theta(s)}{E(s)} = \frac{20}{s(s + 4)}$$



**FIGURE P1.5-1** System for Problem 1.5-1.

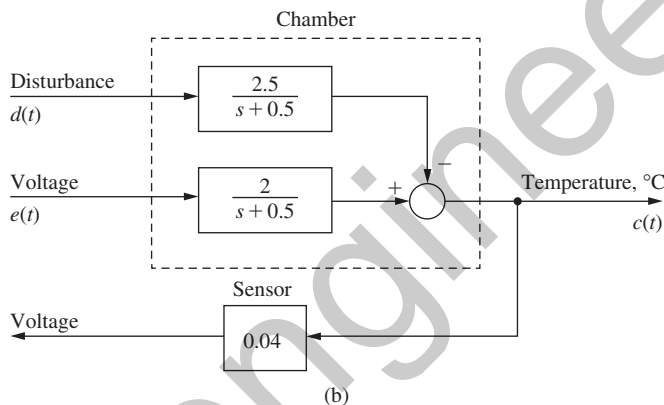
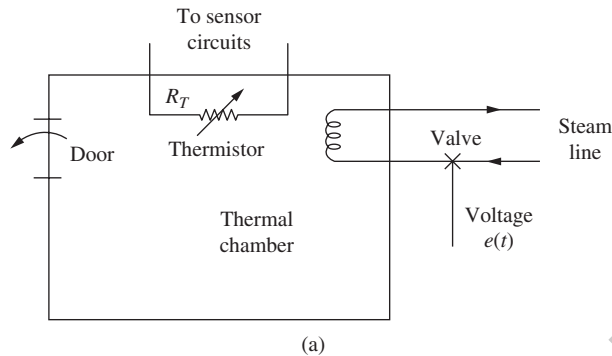
- (a) With the system open loop [ $v_o(t)$  is always zero], a unit step function of voltage is applied to the motor [ $E(s) = 1/s$ ]. Consider only the *steady-state response*. Find the output angle  $\theta(t)$  in degrees, and the angular velocity of the antenna pedestal,  $\dot{\theta}(t)$ , in both degrees per second and rpm.
- (b) The system block diagram is given in Fig. P1.5-1(b), with the angle signals shown in degrees and the voltages in volts. Add the required gains and the transfer functions to this block diagram.
- (c) Make the changes necessary in the gains in part (b) such that the units of  $\theta(t)$  are radians.
- (d) A step input of  $\theta_i(t) = 10^\circ$  is applied at the system input at  $t = 0$ . Find the response  $\theta(t)$ .
- (e) The response in part (d) reaches steady state in approximately how many seconds?
- 1.5-2.** The state-variable model of a servomotor is given in Section 1.5. Expand these state equations to model the antenna pointing system of Problem 1.5-1(b).
- 1.5-3.** (a) Find the transfer function  $\Theta(s)/\Theta_i(s)$  for the antenna pointing system of Problem 1.5-1(b). This transfer function yields the angle  $\theta(t)$  in degrees.  
 (b) Modify the transfer function in part (a) such that use of the modified transfer function yields  $\theta(t)$  in radians.  
 (c) Verify the results of part (b) using the block diagram of Problem 1.5-1(b).
- 1.5-4.** Shown in Fig. P1.5-4 is the block diagram of one joint of a robot arm. This system is described in Section 1.5. The input  $M(s)$  is the controlling signal,  $E_a(s)$  is the servomotor input voltage,  $\Theta_m(s)$  is the motor shaft angle, and the output  $\Theta_a(s)$  is the angle of the arm. The inductance of the armature of the servomotor has been neglected such that the servomotor transfer function is second order. The servomotor transfer function includes the inertia of both the gears and the robot arm. Derive the transfer functions  $\Theta_a(s)/M(s)$  and  $\Theta_a(s)/E_a(s)$ .



**FIGURE P1.5-4** A model of a robot arm.

- 1.5-5.** Consider the robot arm depicted in Fig. P1.5-4.
- (a) Suppose that the units of  $e_a(t)$  are volts, that the units of both  $\theta_m(t)$  and  $\theta_a(t)$  are degrees, and that the units of time are seconds. If the servomotor is rated at 24 V [the voltage  $e_a(t)$  should be less than or equal to 24 V], find the rated rpm of the motor (the motor rpm, in steady state, with 24 V applied).
- (b) Find the maximum rate of movement of the robot arm, in degrees per second, with a step voltage of  $e_a(t) = 24u(t)$  volts applied.
- (c) Assume that  $e_a(t)$  is a step function of 24 V. Give the time required for the arm to be moving at 99 percent of the maximum rate of movement found in part (b).
- (d) Suppose that the input  $m(t)$  is constrained by system hardware to be less than or equal to 10 V in magnitude. What value would you choose for the gain  $K$ . Why?
- 1.6-1.** A thermal test chamber is illustrated in Fig. P1.6-1(a). This chamber, which is a large room, is used to test large devices under various thermal stresses. The chamber is heated with steam, which is controlled by an electrically activated valve. The temperature of the chamber is measured by a sensor based on a thermistor, which is a semiconductor resistor whose resistance varies with temperature. Opening the door into the chamber affects the chamber temperature and thus must be considered as a disturbance.





**FIGURE P1.6-1** A thermal stress chamber.

A simplified model of the test chamber is shown in Fig. P1.6-1(b), with the units of time in minutes. The control input is the voltage  $e(t)$ , which controls the valve in the steam line, as shown. For the disturbance  $d(t)$ , a unit step function is used to model the opening of the door. With the door closed,  $d(t) = 0$ .

- Find the time constant of the chamber.
- With the controlling voltage  $e(t) = 4u(t)$  and the chamber door closed, find and plot the chamber temperature  $c(t)$ . In addition, give the steady-state temperature.
- A tacit assumption in part (a) is an initial chamber temperature of zero degrees Celsius. Repeat part (b), assuming that the initial chamber temperature is  $c(0) = 25^\circ\text{C}$ .
- Two minutes after the application of the voltage in part (c), the door is opened, and it remains open. Add the effects of this disturbance to the plot of part (c).
- The door in part (d) remains open for 12 min. and is then closed. Add the effects of this disturbance to the plot of part (d).

**1.6-2.** The thermal chamber transfer function  $C(s)/E(s) = 2.5/(s + 1)$  is based on the units of time being minutes.

- Modify this transfer function to yield the chamber temperature  $c(t)$  based on seconds.
- Verify the result in part (a) by solving for  $c(t)$  with the door closed and the input  $e(t) = 4u(t)$  volts, (i) using the chamber transfer function found in part (a), and (ii) using the transfer function of Fig. P1.6-1. Show that (i) and (ii) yield the same temperature at  $t = 1$  min.

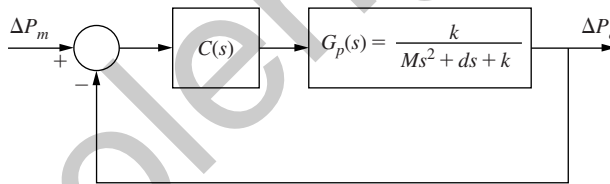
**1.7-1.** Consider the single-machine infinite bus power system of Fig. 1-13 with  $M = 0.5$ ,  $d = 0.1$ , and  $k = 10$ . Find the steady-state gain of the closed-loop transfer function when:

- (a)  $C(s) = 1$
- (b)  $C(s) = \frac{1}{1 + 2s}$
- (c)  $C(s) = \frac{10s}{s^2 + 2s + 8}$

**1.7-2.** Consider the single-machine infinite bus power system of Fig. 1-13 with  $M = 0.5$ ,  $d = 0.1$ ,  $k = 10$ , and  $C(s) = 1$ . Simulate the unit step response for this system, and compute the rise time of the output  $\Delta P_e$ . Repeat the same for  $M = 1$  and  $M = 10$ , and observe how the rise time is affected by increasing the inertia  $M$ . What is the steady-state value of  $\Delta P_e$  when  $k = 100$ ?

**1.7-3.** Consider a slightly different block-diagram for the closed-loop single-machine infinite-bus power system of Fig. 1-13 as shown below in Fig. P1.7-3. In this block diagram the controller  $C(s)$  is placed in cascade to the plant instead of in the feedback loop. The feedback gain is considered as unity.

- (a) For this set-up compute the steady-state error between a unit step input  $\Delta P_m$  and the output  $\Delta P_e$  for the following controller transfer functions:
  - (i)  $C(s) = a_0$  (proportional, or  $P$ -controller)
  - (ii)  $C(s) = a_0 + a_1s$  (proportional + derivative, or PD controller)
  - (iii)  $C(s) = a_0 + \frac{a_2}{s}$  (proportional + integral, or PI controller)
  - (iv)  $C(s) = a_0 + a_1s + \frac{a_2}{s}$  (proportional + integral + derivative, or PID controller)
- (b) Using  $M = 0.5$ ,  $d = 0.1$ , and  $k = 10$ , and a PID controller  $C(s) = 1 + 2s + \frac{10}{s}$ , compute the rise time of  $\Delta P_e$ . How does the rise time change when the derivative gain is doubled?



**FIGURE P1.7-3** System for Problem 1.7-3.

# 2

## Discrete-Time Systems and the $z$ -Transform

### 2.1 INTRODUCTION

In this chapter two important topics are introduced: discrete-time systems and the  $z$ -transform. In contrast to a continuous-time system, whose operation is described (modeled) by a set of differential equations, a discrete-time system is one whose operation is described by a set of difference equations. The transform method employed in the analysis of linear time-invariant continuous-time systems is the Laplace transform; in a similar manner, the transform used in the analysis of linear time-invariant discrete-time systems is the  $z$ -transform. The modeling of discrete-time systems by difference equations, transfer functions, and state equations is presented in this chapter.

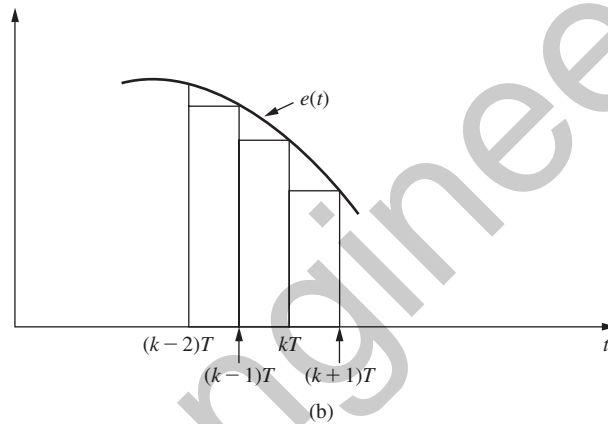
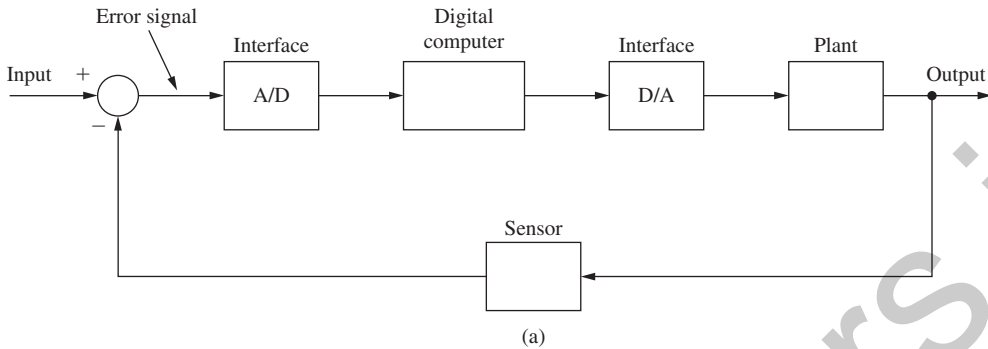
### 2.2 DISCRETE-TIME SYSTEMS

To illustrate the idea of a discrete-time system, consider the digital control system shown in Fig. 2-1(a). The digital computer performs the compensation function within the system. The interface at the input of the computer is an analog-to-digital (A/D) converter, and is required to convert the error signal, which is a continuous-time signal, into a form that can be readily processed by the computer. At the computer output a digital-to-analog (D/A) converter is required to convert the binary signals of the computer into a form necessary to drive the plant.

We will now consider the following example. Suppose that the A/D converter, the digital computer, and the D/A converter are to replace an analog, or continuous-time, proportional-integral (PI) compensator such that the digital control system response has essentially the same characteristics as the analog system. (The PI controller is discussed in Chapter 8.) The analog controller output is given by

$$m(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau \quad (2-1)$$

where  $e(t)$  is the controller input signal,  $m(t)$  is the controller output signal, and  $K_p$  and  $K_I$  are constant gains determined by the design process.



**FIGURE 2-1** Digital control system.

Since the digital computer can be programmed to multiply, add, and integrate numerically, the controller equation can be realized using the digital computer. For this example, the rectangular rule of numerical integration [1], illustrated in Fig. 2-1(b), will be employed. Of course, other algorithms of numerical integration may also be used.

For the rectangular rule, the area under the curve in Fig. 2-1(b) is approximated by the sum of the rectangular areas shown. Thus, letting  $x(t)$  be the numerical integral of  $e(t)$ , we can write

$$x(kT) = x[(k - 1)T] + Te(kT) \quad (2-2)$$

where  $T$  is the numerical algorithm step size, in seconds. Then (2-1) becomes, for the digital compensator,

$$m(kT) = K_p e(kT) + K_I x(kT)$$

Equation (2-2) is a first-order linear *difference equation*. The general form of a first-order linear time-invariant difference equation is (with the  $T$  omitted for convenience)

$$x(k) = b_1 e(k) + b_0 e(k - 1) - a_0 x(k - 1) \quad (2-3)$$

This equation is first order since the signals from only the last sampling instant appear explicitly in the equation. The general form of an  $n$ th-order linear difference equation is

$$x(k) = b_n e(k) + b_{n-1} e(k-1) + \dots + b_0 e(k-n) - a_{n-1} x(k-1) - \dots - a_0 x(k-n) \quad (2-4)$$

It will be shown in Chapter 5 that if the plant in Fig. 2-1 is also linear and time invariant, the entire system may be modeled by a difference equation of the form of (2-4), which is generally of higher order than that of the controller. Compare (2-4) to a linear differential equation describing an  $n$ th-order continuous-time system.

$$y(t) = \beta_n \frac{d^n e(t)}{dt^n} + \dots + \beta_1 \frac{de(t)}{dt} + \beta_0 e(t) - \alpha_n \frac{d^n y(t)}{dt^n} - \dots - \alpha_1 \frac{dy(t)}{dt} \quad (2-5)$$

Two approaches may be used in the design of digital compensators. First, an analog compensator may be designed and then converted by some approximate procedure to a digital compensator, as in the example above. Chapters 3 through 9 and 11 present exact methods of designing digital compensators, as compared to the approximate methods of converting analog compensators to digital compensators.

The describing equation of a linear, time-invariant analog (i.e., continuous-time) filter is also of the form of (2-5). The device that realizes this filter, usually a  $RC$  network with operational amplifiers, can be considered to be an analog computer programmed to solve the filter equation. In a similar manner, (2-4) is the describing equation of a linear, time-invariant discrete filter, which is usually called a *digital filter*. The device that realizes this filter is a digital computer programmed to solve (2-4). Thus the digital computer in Fig. 2-1 would be programmed to solve a difference equation of the form of (2-4), and the problem of the control system designer would be to determine (1)  $T$ , the sampling period; (2)  $n$ , the order of the difference equation; and (3)  $a_i$  and  $b_i$ , the filter coefficients, such that the control system has certain desired characteristics.

There are additional problems in the realization of the digital filter: for example, the computer word length required to keep system errors caused by round-off in the computer at an acceptable level. As an example, a digital filter (controller) was designed and implemented to land aircraft automatically on U.S. Navy aircraft carriers [2]. In this system, the sample rate is 25 Hz ( $T = 0.04$ s), and the controller is eleventh order. The minimum word length required for the computer was found to be 32 bits, in order that system errors caused by round-off in the computer remain at acceptable levels. As an additional point, this controller is a proportional-plus-integral-plus-derivative (PID) controller with extensive noise filtering required principally because of the differentiation in the D part of the filter. The integration and the differentiation are performed numerically, as is discussed in Chapter 8. In many applications other than control systems, digital filters have been designed to replace analog filters and the problems encountered are the same as those listed above.

## 2.3 TRANSFORM METHODS

In linear time-invariant continuous-time systems, the Laplace transform can be utilized in system analysis and design. For example, an alternative, but equally valid description of the operation of a system described by (2-5) is obtained by taking the Laplace transform of this equation and solving for the transfer function:

$$\frac{Y(s)}{E(s)} = \frac{\beta_n s^n + \cdots + \beta_1 s + \beta_0}{\alpha_n s^n + \cdots + \alpha_1 s + 1} \quad (2-6)$$

A transform will now be defined that can be utilized in the analysis of discrete-time systems modeled by difference equations of the form given in (2-4).

A transform is defined for number sequences as follows. The function  $E(z)$  is defined as a power series in  $z^{-k}$  with coefficients equal to the values of the number sequence  $\{e(k)\}$ . This transform, called the  $z$ -transform, is then expressed by the transform pair

$$\begin{aligned} E(z) &= \mathcal{Z}[\{e(k)\}] = e(0) + e(1)z^{-1} + e(2)z^{-2} + \cdots \\ e(k) &= \mathcal{Z}^{-1}[E(z)] = \frac{1}{2\pi j} \oint_{\Gamma} E(z)z^{k-1} dz, \quad j = \sqrt{-1} \end{aligned} \quad (2-7)$$

where  $\mathcal{Z}(\cdot)$  indicates the  $z$ -transform operation and  $\mathcal{Z}^{-1}(\cdot)$  indicates the inverse  $z$ -transform.  $E(z)$  in (2-7) can be written in more compact notation as

$$E(z) = \mathcal{Z}[\{e(k)\}] = \sum_{k=0}^{\infty} e(k)z^{-k} \quad (2-8)$$

For convenience, we often omit the braces and express  $\mathcal{Z}[\{e(k)\}]$  as  $\mathcal{Z}[e(k)]$ . However, it should be remembered that the  $z$ -transform applies to a sequence.

The  $z$ -transform is defined for any number sequence  $\{e(k)\}$ , and may be used in the analysis of any type of system described by linear time-invariant difference equations. For example, the  $z$ -transform is used in discrete probability problems, and for this case the numbers in the sequence  $\{e(k)\}$  are discrete probabilities [3].

Equation (2-7) is the definition of the single-sided  $z$ -transform. The double-sided  $z$ -transform, sometimes called the generating function [4], is defined as

$$G[\{f(k)\}] = \sum_{k=-\infty}^{\infty} f(k)z^{-k} \quad (2-9)$$

Throughout this book, only the single-sided  $z$ -transform as defined in (2-7) will be used, and this transform will be referred to as the ordinary  $z$ -transform. If the sequence  $e(k)$  is generated from a time function  $e(t)$  by sampling every  $T$  seconds,  $e(k)$  is understood to be  $e(kT)$  (i.e., the  $T$  is dropped for convenience).

Three examples will now be given to illustrate the  $z$ -transform.

### EXAMPLE 2.1

Given  $E(z)$  below, find  $\{e(k)\}$ .

$$E(z) = 1 + 3z^{-1} - 2z^{-2} + z^{-4} + \cdots$$

We know, then, from (2-7), that the values of the number sequence  $\{e(k)\}$  are

$$\begin{aligned} e(0) &= 1 & e(3) &= 0 \\ e(1) &= 3 & e(4) &= 1 \\ e(2) &= -2 & \cdots & \end{aligned}$$

Consider now the identity

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \dots, \quad |x| < 1 \quad (2-10)$$

This power series is useful, in some cases, in expressing  $E(z)$  in closed form, as will be illustrated in the next two examples.

### EXAMPLE 2.2

Given that  $e(k) = 1$  for all  $k$ , find  $E(z)$ . By definition  $E(z)$  is

$$E(z) = 1 + z^{-1} + z^{-2} + \dots$$

The closed form of  $E(z)$  is obtained from (2-10):

$$E(z) = \frac{1}{1-z^{-1}} = \frac{z}{z-1}, \quad |z^{-1}| < 1 \quad (2-11)$$

Note that  $\{e(k)\}$  may be generated by sampling a unit step function. However, there are many other time functions that have a value of unity every  $T$  seconds, and thus all have the same  $z$ -transform.

This same  $E(z)$  can be obtained using MATLAB as shown below. We also include the sampling interval  $T$  to illustrate that it has no impact on the result.

```
>>syms k
ek=1^k;
Ez=ztrans(ek)
```

```
Ez =
z/(z - 1)
```

```
>>syms k T
ek=1^(k*T);
Ez=ztrans(ek)
```

```
Ez =
z/(z - 1)
```

### EXAMPLE 2.3

Given that  $e(k) = \epsilon^{-akT}$ , find  $E(z)$ .  $E(z)$  can be written in power series form as

$$\begin{aligned} E(z) &= 1 + \epsilon^{-aT}z^{-1} + \epsilon^{-2aT}z^{-2} + \dots \\ &= 1 + (\epsilon^{-aT}z^{-1}) + (\epsilon^{-aT}z^{-1})^2 + \dots \end{aligned}$$

$E(z)$  can be put in closed form by applying (2-10), so that

$$E(z) = \frac{1}{1 - \varepsilon^{-aT}z^{-1}} = \frac{z}{z - \varepsilon^{-aT}}, \quad |\varepsilon^{-aT}z^{-1}| < 1$$

Note that, in this example,  $\{e(k)\}$  may be generated by sampling the function  $e(t) = \varepsilon^{-at}$ .

Again, this same  $E(z)$  can be obtained using MATLAB as follows:

```
>>syms k a T
ek=exp(-a*k*T);
Ez=ztrans(ek)

Ez =
z/(z - exp(-a*T))
```

As with the Laplace transform, each  $z$ -transform has a region of existence in the complex plane. This region is of importance if an integral is used to find the inverse  $z$ -transform [5]. However, if tables are used for both the forward transform and the inverse transform, the region of existence is not of direct importance. Hence we generally will not state the region of existence when a transform is employed.

## 2.4 PROPERTIES OF THE $z$ -TRANSFORM

Several properties of the  $z$ -transform will now be developed. These properties will prove to be useful in the analysis of discrete-time systems.

### Addition and Subtraction

**PROPERTY.** The  $z$ -transform of a sum of number sequences is equal to the sum of the  $z$ -transforms of the number sequences; that is,

$$\mathcal{Z}[e_1(k) \pm e_2(k)] = E_1(z) \pm E_2(z) \quad (2-12)$$

**Proof.** From the definition of the  $z$ -transform,

$$\begin{aligned} \mathcal{Z}[e_1(k) \pm e_2(k)] &= \sum_{k=0}^{\infty} [e_1(k) \pm e_2(k)]z^{-k} \\ &= \sum_{k=0}^{\infty} e_1(k)z^{-k} \pm \sum_{k=0}^{\infty} e_2(k)z^{-k} = E_1(z) \pm E_2(z) \end{aligned}$$

### Multiplication by a Constant

**PROPERTY.** The  $z$ -transform of a number sequence multiplied by a constant is equal to the constant multiplied by the  $z$ -transform of the number sequence:

$$\mathcal{Z}[ae(k)] = a\mathcal{Z}[e(k)] = aE(z) \quad (2-13)$$



**Proof.** From the definition of the z-transform,

$$\mathfrak{z}[ae(k)] = \sum_{k=0}^{\infty} ae(k)z^{-k} = a \sum_{k=0}^{\infty} e(k)z^{-k} = aE(z)$$

Properties (2-12) and (2-13) form the *linearity property* of the z-transform.

### EXAMPLE 2.4

The linearity property of the z-transform can also be proved as follows. Let

$$e(k) = ae_1(k) + be_2(k)$$

Then

$$\begin{aligned} \mathfrak{z}[e(k)] &= \mathfrak{z}[ae_1(k) + be_2(k)] \\ &= \sum_{k=0}^{\infty} [ae_1(k) + be_2(k)]z^{-k} \\ &= a \sum_{k=0}^{\infty} e_1(k)z^{-k} + b \sum_{k=0}^{\infty} e_2(k)z^{-k} \\ &= a\mathfrak{z}[e_1(k)] + b\mathfrak{z}[e_2(k)] \\ &= aE_1(z) + bE_2(z) \end{aligned}$$

### Real Translation

**PROPERTY.** Let  $n$  be a positive integer, and let  $E(z)$  be the z-transform of  $\{e(k)\}$ . Then

$$\mathfrak{z}[e(k-n)u(k-n)] = z^{-n}E(z) \quad (2-14)$$

and

$$\mathfrak{z}[e(k+n)u(k)] = z^n \left[ E(z) - \sum_{k=0}^{n-1} e(k)z^{-k} \right] \quad (2-15)$$

The inversion integral in the z-transform definition (2-7) yields  $e(k) = 0$  for all  $k < 0$ ; hence, the z-transform of  $e(k)$  can be expressed as

$$\mathfrak{z}[e(k)] = \mathfrak{z}[e(k)u(k)]$$

where  $u(k)$  is the *discrete unit step function* defined by

$$u(k) = \begin{cases} 0, & k < 0 \\ 1, & k \geq 0 \end{cases}$$

The time-delayed function in (2-14) is given by

$$e(k-n)u(k-n) = e(k)u(k) \Big|_{k \leftarrow k-n}$$

In (2-15), advancing  $e(k)$  in time results in

$$\begin{aligned} \mathfrak{z}[e(k)u(k) \Big|_{k \leftarrow k+n}] &= \mathfrak{z}[e(k+n)u(k+n)] \\ &= \mathfrak{z}[e(k+n)u(k)] \end{aligned}$$

**TABLE 2-1** Examples of Shifting

$k$	$e(k)$	$e(k - 2)$	$e(k + 2)$
0	2	0	1.3
1	1.6	0	1.1
2	1.3	2	1.0
3	1.1	1.6	...
4	1.0	1.3	...
...	...	...	...

We now prove (2-14) and (2-15).

**Proof.** From the definition of the  $z$ -transform,

$$\begin{aligned} \mathfrak{z}[e(k - n)u(k - n)] &= e(0)z^{-n} + e(1)z^{-(n+1)} + e(2)z^{-(n+2)} + \dots \\ &= z^{-n}[e(0) + e(1)z^{-1} + e(2)z^{-2} + \dots] = z^{-n}E(z) \end{aligned}$$

which proves (2-14). Also,

$$\mathfrak{z}[e(k + n)u(k)] = e(n) + e(n + 1)z^{-1} + e(n + 2)z^{-2} + \dots$$

By adding and subtracting terms and factoring  $z^n$ , we obtain

$$\begin{aligned} \mathfrak{z}[e(k + n)u(k)] &= z^n[e(0) + e(1)z^{-1} + e(2)z^{-2} + \dots + e(n - 1)z^{-(n-1)} \\ &\quad + e(n)z^{-n} + e(n + 1)z^{-(n+1)} + e(n + 2)z^{-(n+2)} + \dots \\ &\quad - e(0) - e(1)z^{-1} - e(2)z^{-2} - \dots - e(n - 1)z^{-(n-1)}] \end{aligned}$$

or

$$\mathfrak{z}[e(k + n)u(k)] = z^n \left[ E(z) - \sum_{k=0}^{n-1} e(k)z^{-k} \right]$$

which proves (2-15).

To illustrate these properties further, consider the number sequence shown in Table 2-1, which illustrates the effects of shifting by two sample periods. For the sequence  $e(k - 2)u(k - 2)$ , no numbers of the sequence  $e(k)$  are lost; thus the  $z$ -transform of  $e(k - 2)u(k - 2)$  can be expressed as a simple function of  $E(z)$ . However, in forming the sequence  $e(k + 2)u(k)$ , the first two values of  $e(k)$  have been lost, and the  $z$ -transform of  $e(k + 2)u(k)$  cannot be expressed as a simple function of  $E(z)$ .

### EXAMPLE 2.5

It was shown in Example 2.3 that

$$\mathfrak{z}[\varepsilon^{-akT}] = \frac{z}{z - \varepsilon^{-aT}}$$

Thus

$$\mathfrak{z}[\varepsilon^{-a(k-3)T}u[(k-3)T]] = z^{-3} \left[ \frac{z}{z - \varepsilon^{-aT}} \right] = \frac{1}{z^2(z - \varepsilon^{-aT})}$$

where  $u(kT)$  is the unit step. Also,

$$\mathfrak{z}[\varepsilon^{-a(k+2)T}u[(kT)]] = z^2 \left[ \frac{z}{z - \varepsilon^{-aT}} - 1 - \varepsilon^{-aT}z^{-1} \right]$$

We now define the *discrete unit impulse function*  $\delta(k - n)$ :

$$\delta(k - n) = \begin{cases} 1, & k = n \\ 0, & k \neq n \end{cases} \quad (2-16)$$

This function is also called the *unit sample function*. The  $z$ -transform of the unit impulse function is then, for  $n \geq 0$ ,

$$\mathfrak{z}[\delta(k - n)] = \sum_{k=0}^{\infty} \delta(k - n)z^{-k} = \delta(k - n)z^{-n} = z^{-n}$$

from (2-16).

### Complex Translation

**PROPERTY.** Given that the  $z$ -transform of  $e(k)$  is  $E(z)$ . Then

$$\mathfrak{z}[\varepsilon^{akT}e(kT)] = E(z\varepsilon^{-aT}) \quad (2-17)$$

**Proof.** From the definition of the  $z$ -transform,

$$\begin{aligned} \mathfrak{z}[\varepsilon^{akT}e(kT)] &= e(0) + \varepsilon^{aT}e(T)z^{-1} + \varepsilon^{2aT}e(2T)z^{-2} + \dots \\ &= e(0) + e(T)(z\varepsilon^{-aT})^{-1} + e(2T)(z\varepsilon^{-aT})^{-2} + \dots \end{aligned}$$

or

$$\mathfrak{z}[\varepsilon^{akT}e(kT)] = E(z)_{z \leftarrow z\varepsilon^{-aT}} = E(z\varepsilon^{-aT})$$

### EXAMPLE 2.6

Given that the  $z$ -transform of  $e(kT) = kT$  is  $E(z) = Tz/(z - 1)^2$ , then the  $z$ -transform of  $kT \varepsilon^{akT}$  is

$$\begin{aligned} E(z) \Big|_{z \leftarrow z\varepsilon^{-aT}} &= \frac{Tz}{(z - 1)^2} \Big|_{z \leftarrow z\varepsilon^{-aT}} \\ &= \frac{Tz\varepsilon^{-aT}}{(z\varepsilon^{-aT} - 1)^2} = \frac{T\varepsilon^{aT}z}{(z - \varepsilon^{aT})^2} \end{aligned}$$

Again, this same  $E(z)$  can be obtained using MATLAB as follows:

```
>>syms a k T
ekT=(k*T)*exp(a*k*T);
Ez=ztrans(ekT)

Ez =
(T*z*exp(T*a))/(z - exp(T*a))^2
```

### Initial Value

**PROPERTY.** Given that the  $z$ -transform of  $e(k)$  is  $E(z)$ . Then

$$e(0) = \lim_{z \rightarrow \infty} E(z) \quad (2-18)$$

**Proof.** Since

$$E(z) = e(0) + e(1)z^{-1} + e(2)z^{-2} + \dots$$

then (2-18) is seen by inspection.

### Final Value

**PROPERTY.** Given that the  $z$ -transform of  $e(k)$  is  $E(z)$ . Then

$$\lim_{n \rightarrow \infty} e(n) = \lim_{z \rightarrow 1} (z - 1)E(z)$$

provided that the left-side limit exists.

**Proof.** Consider the transform

$$\begin{aligned} \mathcal{Z}[e(k+1) - e(k)] &= \lim_{n \rightarrow \infty} \left[ \sum_{k=0}^n e(k+1)z^{-k} - \sum_{k=0}^n e(k)z^{-k} \right] \\ &= \lim_{n \rightarrow \infty} [-e(0) + e(1)(1 - z^{-1}) + e(2)(z^{-1} - z^{-2}) + \dots \\ &\quad + e(n)(z^{-n+1} - z^{-n}) + e(n+1)z^{-n}] \end{aligned}$$

Thus

$$\lim_{z \rightarrow 1} [\mathcal{Z}[e(k+1) - e(k)]] = \lim_{n \rightarrow \infty} [e(n+1) - e(0)]$$

Also, from the real translation property,

$$\begin{aligned} \mathcal{Z}[e(k+1) - e(k)] &= z[E(z) - e(0)] - E(z) \\ &= (z - 1)E(z) - ze(0) \end{aligned}$$

Equating the two expressions above, we obtain

$$\lim_{n \rightarrow \infty} e(n) = \lim_{z \rightarrow 1} (z - 1)E(z)$$

provided that the left-side limit exists. It is shown in Chapter 7 that this limit exists provided that all poles of  $E(z)$  are inside the unit circle, except for possibly a simple pole at  $z = 1$ .

### EXAMPLE 2.7

To illustrate the initial-value property and the final-value property, consider the  $z$ -transform of  $e(k) = 1, k = 0, 1, 2, \dots$ . We have shown, in Example 2.2, that

$$E(z) = \mathcal{Z}[1] = \frac{z}{z - 1}$$

Applying the initial-value property, we see that

$$e(0) = \lim_{z \rightarrow \infty} \frac{z}{z - 1} = \lim_{z \rightarrow \infty} \frac{1}{1 - 1/z} = 1$$

Since the final value of  $e(k)$  exists, we may apply the final-value property:

$$\lim_{k \rightarrow \infty} e(k) = \lim_{z \rightarrow 1} (z - 1)E(z) = \lim_{z \rightarrow 1} (z - 1) \left( \frac{z}{z - 1} \right) = \lim_{z \rightarrow 1} z = 1$$

The derived properties of the  $z$ -transform are listed in Table 2-2, which also includes additional properties. These additional properties are derived later, or else the derivations are given as problems. The notation  $e_1(k) * e_2(k)$  indicates convolution, and is discussed in Section 2.7.

## 2.5 FINDING z-TRANSFORMS

A table of  $z$ -transforms based on time sequences is given in Table 2-3. A more comprehensive table of  $z$ -transforms based on time-domain and Laplace-domain functions is given in Appendix VI. Note that the entries in Table 2-3 can be generated using MATLAB.

**TABLE 2-2 Properties of the z-Transform**

Sequence	Transform
$e(k)$	$E(z) = \sum_{k=0}^{\infty} e(k)z^{-k}$
$a_1e_1(k) + a_2e_2(k)$	$a_1E_1(z) + a_2E_2(z)$
$e(k - n)u(k - n); \quad n \geq 0$	$z^{-n}E(z)$
$e(k + n)u(k); \quad n \geq 1$	$z^n \left[ E(z) - \sum_{k=0}^{n-1} e(k)z^{-k} \right]$
$\mathcal{E}^{aT}e(k)$	$E(z)\mathcal{E}^{-aT}$
$ke(k)$	$-z \frac{dE(z)}{dz}$
$e_1(k) * e_2(k)$	$E_1(z)E_2(z)$
$e_1(k) = \sum_{n=0}^k e(n)$	$E_1(z) = \frac{z}{z - 1} E(z)$
Initial value: $e(0) = \lim_{z \rightarrow \infty} E(z)$	
Final value: $e(\infty) = \lim_{z \rightarrow 1} (z - 1)E(z)$ , if $e(\infty)$ exists	

TABLE 2-3 z-Transforms	
Sequence	Transform
$\delta(k - n)$	$z^{-n}$
1	$\frac{z}{z - 1}$
$k$	$\frac{z}{(z - 1)^2}$
$k^2$	$\frac{z(z + 1)}{(z - 1)^3}$
$a^k$	$\frac{z}{z - a}$
$ka^k$	$\frac{az}{(z - a)^2}$
$\sin ak$	$\frac{z \sin a}{z^2 - 2z \cos a + 1}$
$\cos ak$	$\frac{z(z - \cos a)}{z^2 - 2z \cos a + 1}$
$a^k \sin bk$	$\frac{az \sin b}{z^2 - 2az \cos b + a^2}$
$a^k \cos bk$	$\frac{z^2 - az \cos b}{z^2 - 2az \cos b + a^2}$

### EXAMPLE 2.8

In this example, we verify the last row in Table 2-3. We also include the sampling interval  $T$  to illustrate its impact on the result.

```
>> syms a b k
Ez = ztrans(a^k*cos(b*k));
pretty(Ez) % Make it easier to read
```

$$z \left( \begin{array}{c} z \\ \cos(b) - - \\ a \end{array} \right)$$


---


$$a \left( \begin{array}{c} / \quad 2 \\ z \quad 2 z \cos(b) \\ - - - - - + 1 \\ / \quad 2 \\ a \end{array} \right)$$

In Table 2-3, the sampling interval  $T$  is assumed to be 1. If  $T$  has some other value, the transforms must be altered appropriately. For the last line in the table, we now let  $T$  have an arbitrary value.

```

>> syms a b k T
Ez = ztrans(a^(k*T)*cos(b*k*T));
pretty(Ez)
    
```

$$\frac{z \left( \cos(T b) - \frac{z}{a} \right)}{z^2 - 2 \frac{z}{a} \cos(T b) + \frac{z}{a^2}} + 1$$

Thus the relationship is

$$\mathcal{Z}[a^{kT} \cos b k T] = \frac{z^2 - a^T z \cos b T}{z^2 - 2 a^T z \cos b T + a^{2T}}$$

In many digital control system design applications, system functions are conveniently expressed in the Laplace domain. Appendix VI may be used to find the z-transform of Laplace-domain functions. Using the Addition and Subtraction properties of z-transforms, one can first express a Laplace function in partial fraction form and then find the z-transform of each term.

### EXAMPLE 2.9

Find the z-transform of  $E(s) = \frac{s^2 + 4s + 3}{s^3 + 6s^2 + 8s}$

First find the partial fractions expansion:

$$\begin{aligned}
 E(s) &= \frac{s^2 + 4s + 3}{s^3 + 6s^2 + 8s} = \frac{s^2 + 4s + 3}{s(s + 2)(s + 4)} \\
 &= \frac{0.375}{s} + \frac{0.25}{s + 2} + \frac{0.375}{s + 4}
 \end{aligned}$$

Next find the z-transform of each term in the expansion:

$$E(z) = 0.375 \frac{z}{z - 1} + 0.025 \frac{z}{z - e^{-2T}} + 0.375 \frac{z}{z - e^{-4T}}$$

For  $T = 0.1$ , we can express  $E(z)$  as the ratio of two polynomials:

$$\begin{aligned}
 E(z) &= 0.375 \frac{z}{z - 1} + 0.025 \frac{z}{z - 0.8187} + 0.375 \frac{z}{z - 0.6703} \\
 &= \frac{z^3 - 1.658z^2 + 0.6804z}{z^3 - 2.489z^2 + 2.038z - 0.5488}
 \end{aligned}$$

MATLAB may also be used to find the z-transform of  $E(s)$ :

```

>>T=0.1;
num=[1 4 3];
    
```

```
denom=[1 6 8 0]; % no repeated poles allowed
n=length(denom);
Es=tf(num,denom)
[r,p,k]=residue(num,denom);
for i=1:1:n-1
pz(i)=exp(p(i)*T);
end
[numzz,denomz]=residue(r,pz,k);
numz=conv(numzz,[1 0]); % multiply by z
Ez=tf(numz,denomz,T)
```

$$E_s = \frac{s^2 + 4s + 3}{s^3 + 6s^2 + 8s}$$

$$E_z = \frac{z^3 - 1.658z^2 + 0.6804z}{z^3 - 2.489z^2 + 2.038z - 0.5488}$$

Note that the MATLAB *ztrans* function may also be used for this example by first finding  $e(t)$  and then finding  $E(z)$ :

```
>>T=0.1;
syms s t k z
Es=(s^2+4*s+3)/(s^3+6*s^2+8*s); % As given
et=ilaplace(Es); % Inverse Laplace to time domain
ekT=subs(et,t,k*T); % substitute t=kT
Ez=ztrans(ekT); % Take the z-Transform
pretty(Ez) % Make it easier to read
```

$$\frac{3z}{8(z-1)} + \frac{z}{4(z-\exp(-1/5))} + \frac{3z}{8(z-\exp(-2/5))}$$

## 2.6 SOLUTION OF DIFFERENCE EQUATIONS

There are three basic techniques for solving linear time-invariant difference equations. The first method, commonly referred to as the classical approach, consists of finding the complementary and the particular parts of the solution [6], in a manner similar to that used in the classical solution of linear differential equations. This technique will not be discussed here; however, a technique similar to it is presented later in this chapter in the discussion of state variables. The second technique, which is a sequential procedure, is the method used in the digital-computer solution of difference equations and is illustrated by the following example. The third technique will be considered later.

### EXAMPLE 2.10

It is desired to find  $m(k)$  for the equation

$$m(k) = e(k) - e(k-1) - m(k-1), k \geq 0$$



where

$$e(k) = \begin{cases} 1, & k \text{ even} \\ 0, & k \text{ odd} \end{cases}$$

and both  $e(-1)$  and  $m(-1)$  are zero. Then  $m(k)$  can be determined by solving the difference equation first for  $k = 0$ , then for  $k = 1$ ,  $k = 2$ , and so on. Thus

$$m(0) = e(0) - e(-1) - m(-1) = 1 - 0 - 0 = 1$$

$$m(1) = e(1) - e(0) - m(0) = 0 - 1 - 1 = -2$$

$$m(2) = e(2) - e(1) - m(1) = 1 - 0 + 2 = 3$$

$$m(3) = e(3) - e(2) - m(2) = 0 - 1 - 3 = -4$$

$$m(4) = e(4) - e(3) - m(3) = 1 - 0 + 4 = 5$$

Note the sequential nature of the solution process in Example 2.10. Using this approach, we can find  $m(k)$  for any value of  $k$ . This technique is not practical, however, for large values of  $k$ , except when implemented on a digital computer. For the example above, a program segment, in MATLAB, which solves the equation for  $0 \leq k \leq 20$  is

```

>>mkminus1 = 0;
ekminus1 = 0;
ek = 1;
for k=0:20
    mk = ek - ekminus1 - mkminus1;
    [k,mk]
    mkminus1 = mk;
    ekminus1 = ek;
    ek = 1 - ek;
end
    
```

In the coding,  $mkminus1$  represents  $m(k - 1)$ ,  $mk$  represents  $m(k)$ ,  $ekminus1$  represents  $e(k - 1)$ , and  $ek$  represents  $e(k)$ . The first three MATLAB statements initialize the sequential process, which begins with the fourth statement and extends to the last statement. A loop exists between the fourth and ninth statements and the first time through the loop  $mk = m(0)$ , the second time through  $mk = m(1)$ , and so on.

The third technique for solving linear time-invariant difference equations, which employs the use of the  $z$ -transform, will now be presented. Consider the following  $n$ th-order difference equation, where it is assumed that  $\{e(k)\}$  is known.

$$m(k) + a_{n-1}m(k - 1) + \cdots + a_0 m(k - n) = b_n e(k) + b_{n-1}e(k - 1) + \cdots + b_0 e(k - n) \quad (2-19)$$

The  $z$ -transform of (2-19), which results from the use of the real translation property (2-14), is

$$M(z) + a_{n-1}z^{-1}M(z) + \cdots + a_0 z^{-n}M(z) = b_n E(z) + b_{n-1}z^{-1}E(z) + \cdots + b_0 z^{-n}E(z) \quad (2-20)$$

Note that the  $z$ -transform has changed the difference equation in (2-19) to the algebraic equation in (2-20). Solving the expression above for  $M(z)$  yields

$$M(z) = \frac{b_n + b_{n-1}z^{-1} + \cdots + b_0 z^{-n}}{1 + a_{n-1}z^{-1} + \cdots + a_0 z^{-n}} E(z) \quad (2-21)$$

$m(k)$  can be found by taking the inverse  $z$ -transform of (2-21). General techniques for determining the inverse  $z$ -transform are discussed in the next section.

**EXAMPLE 2.11**

Consider the difference equation of Example 2.10.

$$m(k) = e(k) - e(k - 1) - m(k - 1)$$

The  $z$ -transform of this equation, obtained via the real translation property, is

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

or

$$M(z) = \left( \frac{z - 1}{z + 1} \right) E(z)$$

Since

$$e(k) = \begin{cases} 1, & k \text{ even} \\ 0, & k \text{ odd} \end{cases}$$

We see that

$$E(z) = 1 + z^{-2} + z^{-4} + \dots = \frac{1}{1 - z^{-2}} = \frac{z^2}{z^2 - 1} = \frac{z^2}{(z - 1)(z + 1)}$$

Thus

$$M(z) = \left( \frac{z - 1}{z + 1} \right) \frac{z^2}{(z - 1)(z + 1)} = \frac{z^2}{z^2 + 2z + 1}$$

We can expand  $M(z)$  into a power series by dividing the numerator of  $M(z)$  by its denominator so as to obtain

$$\begin{array}{r}
 \phantom{z^2 + 2z + 1} \overline{) 1 - 2z^{-1} + 3z^{-2} - 4z^{-3} + \dots} \\
 \underline{z^2 + 2z + 1} \phantom{z^2} \\
 - 2z - 1 \\
 \underline{- 2z - 4 - 2z^{-1}} \\
 3 + 2z^{-1} \\
 \underline{3 + 6z^{-1} + 3z^{-2}} \\
 - 4z^{-1} - 3z^{-2} \\
 \dots
 \end{array}$$

(We discuss this method for finding the inverse transform in the next section.) Therefore,

$$M(z) = 1 - 2z^{-1} + 3z^{-2} - 4z^{-3} + \dots$$

and the values of  $m(k)$  are seen to be the same as those found using the sequential technique in Example 2.10.

Thus far we have considered only difference equations for which the initial conditions are zero. The solution presented in (2-21) represents only the forced part of the response. In order to include initial conditions in the solution of (2-19),  $k$  is replaced with  $k + n$ . Then

$$\begin{aligned} m(k + n) + a_{n-1}m(k + n - 1) + \cdots + a_0m(k) \\ = b_n e(k + n) + b_{n-1}e(k + n - 1) + \cdots + b_0e(k) \end{aligned} \quad (2-22)$$

From the real translation property (2-15),

$$\mathcal{Z}[m(k + i)] = z^i [M(z) - m(0) - m(1)z^{-1} - \cdots - m(i - 1)z^{-(i-1)}] \quad (2-23)$$

Thus the  $z$ -transform of (2-22) can be found using (2-23), and all initial conditions are included in the solution. Note that if all initial conditions are zero, the  $z$ -transform of (2-22) yields the same results as given in (2-21). Note also that for the  $n$ th-order difference equation (2-22), the initial conditions are, from (2-23),  $m(0), m(1), \dots, m(n - 1)$ . Thus the term *initial conditions* has a different meaning than for a differential equation.

## 2.7 THE INVERSE z-TRANSFORM

In order for the  $z$ -transform technique to be a feasible approach in the solution of difference equations, methods for determining the inverse  $z$ -transform are required. Four such methods will be given here.

### Power Series Method

The power series method for finding the inverse  $z$ -transform of a function  $E(z)$ , which is expressed as the ratio of two polynomials in  $z$ , involves dividing the denominator of  $E(z)$  into the numerator such that a power series of the form

$$E(z) = e_0 + e_1 z^{-1} + e_2 z^{-2} + \cdots \quad (2-24)$$

is obtained. From the definition of the  $z$ -transform, it can be seen that the values of  $\{e(k)\}$  are simply the coefficients in the power series. This technique was illustrated in Example 2.11. Another example will now be given.

### EXAMPLE 2.12

It is desired to find the values of  $e(k)$  for  $E(z)$  given by the expression

$$E(z) = \frac{z}{z^2 - 3z + 2}$$

Using long division, we obtain

$$\begin{array}{r}
 z^{-1} + 3z^{-2} + 7z^{-3} + 15z^{-4} + \dots \\
 \hline
 z^2 - 3z + 2 \quad z \\
 \hline
 z - 3 + 2z^{-1} \\
 \hline
 3 - 2z^{-1} \\
 \hline
 3 - 9z^{-1} + 6z^{-2} \\
 \hline
 7z^{-1} - 6z^{-2} \\
 \hline
 7z^{-1} - 21z^{-2} + 14z^{-3} \\
 \hline
 15z^{-2} - 14z^{-3} + \dots \\
 \dots
 \end{array}$$

and therefore

$$\begin{array}{ll}
 e(0) = 0 & e(4) = 15 \\
 e(1) = 1 & \dots \\
 e(2) = 3 & e(k) = 2^k - 1 \\
 e(3) = 7 & \dots
 \end{array}$$

In this particular case, the general expression for  $e(k)$  as a function of  $k$  [i.e.,  $e(k) = 2^k - 1$ ] can be recognized. In general, this cannot be done using the power series method.

### Partial-Fraction Expansion Method

In a manner similar to that employed with the Laplace transform, a function  $E(z)$  can be expanded in partial fractions and then tables of known  $z$ -transform pairs can be used to determine the inverse  $z$ -transform. The  $z$ -transform pairs given in Table 2-3 and in Appendix VI can assist in these transformations. Before proceeding with an example of the partial-fraction expansion method, consider the function

$$E(z) = \frac{z}{z - a} = 1 + az^{-1} + a^2z^{-2} + a^3z^{-3} + \dots \quad (2-25)$$

Examination of the power series indicates that

$$\mathfrak{z}^{-1} \left[ \frac{z}{z - a} \right] = a^k \quad (2-26)$$

where  $\mathfrak{z}^{-1}[\cdot]$  indicates the inverse  $z$ -transform. This particular function is perhaps the most common  $z$ -transform encountered, since the sequence  $\{a^k\}$  is exponential in nature (see Problem 2.3-2).

It is seen from the transform table in Appendix VI that a factor of  $z$  appears in the numerator of the transforms given. Hence the partial-fraction expansion should be performed on  $E(z)/z$ , which will result in the terms of  $E(z)$  being of the same form as those in the tables.

### EXAMPLE 2.13

Consider the function  $E(z)$  given in Example 2.12:

$$E(z) = \frac{z}{(z-1)(z-2)}$$

Hence

$$\frac{E(z)}{z} = \frac{1}{(z-1)(z-2)} = \frac{-1}{z-1} + \frac{1}{z-2}$$

Then

$$\mathcal{Z}^{-1}[E(z)] = \mathcal{Z}^{-1}\left[\frac{-z}{z-1}\right] + \mathcal{Z}^{-1}\left[\frac{z}{z-2}\right]$$

From (2-26) or Table 2-3, the value of  $e(k)$  is given by

$$e(k) = -1 + 2^k$$

which is the same value as that found in Example 2.12.

We can employ MATLAB to perform this same transformation:

```
>>syms k z;
Ez=z/((z-1)*(z-2));
iztrans(Ez,k)
```

```
ans =
    2^k - 1
```

Consider next the function

$$E_1(z) = z^{-1}E(z) = \frac{1}{(z-1)(z-2)}$$

From the real translation property (2-14),  $e_1(k)$  is given by

$$\begin{aligned} e_1(k) &= \mathcal{Z}^{-1}[z^{-1}E(z)] = e(k-1)u(k-1) \\ &= [-1 + 2^{(k-1)}]u(k-1) \\ &= \begin{cases} 0 & k = 0 \\ -1 + 2^{(k-1)}, & k \geq 1 \end{cases} \end{aligned}$$

This inverse can also be found by partial-fraction expansion.

$$\frac{E_1(z)}{z} = \frac{1}{z(z-1)(z-2)} = \frac{\frac{1}{2}}{z} + \frac{-1}{z-1} + \frac{\frac{1}{2}}{z-2}$$

or,

$$E_1(z) = \frac{1}{2} + \frac{-z}{z-1} + \frac{(\frac{1}{2})z}{z-2}$$

Thus

$$e_1(k) = a - 1 + \left(\frac{1}{2}\right)(2)^k = a - 1 + (2)^{k-1}$$

where

$$a = \begin{cases} \frac{1}{2}, & k = 0 \\ 0, & k \geq 1 \end{cases}$$

since from Table 2-3,

$$\mathfrak{Z}^{-1}\left[\frac{1}{2}\right] = \frac{1}{2} \delta(k) = \begin{cases} \frac{1}{2}, & k = 0 \\ 0, & k \geq 1 \end{cases}$$

Hence the two procedures yield the same results for  $e_1(k)$ .

Again, we can employ MATLAB to perform the same transformation:

```
>>syms k z;
E1z=1/((z-1)*(z-2));
iztrans(E1z,k)

ans =
2^k/2 + kroneckerDelta(k, 0)/2 - 1
```

Thus far we have considered the inverse  $z$ -transform by partial fractions only for functions that have real poles. The same partial-fraction procedure applies for complex poles; however, the resulting inverse transform contains complex functions. Of course, the sum of these functions is real. We now develop a different procedure that expresses the inverse transforms as real functions.

First, consider the real function

$$\begin{aligned} y(k) &= A\epsilon^{akT} \cos(bkT + \theta) = \frac{A\epsilon^{akT}}{2} \left[ \epsilon^{jbkT} \epsilon^{j\theta} + \epsilon^{-jbkT} \epsilon^{-j\theta} \right] \\ &= \frac{A}{2} \left[ \epsilon^{(aT+jbT)k} \epsilon^{j\theta} + \epsilon^{(aT-jbT)k} \epsilon^{-j\theta} \right] \end{aligned} \quad (2-27)$$

where  $a$  and  $b$  are real. Euler's relation, given by

$$\cos x = \frac{\epsilon^{jx} + \epsilon^{-jx}}{2}$$

is used in (2-27). The  $z$ -transform of this function, from Appendix VI, is given by

$$\begin{aligned} Y(z) &= \frac{A}{2} \left[ \frac{\epsilon^{j\theta} z}{z - \epsilon^{aT+jbT}} + \frac{\epsilon^{-j\theta} z}{z - \epsilon^{aT-jbT}} \right] \\ &= \frac{(A\epsilon^{j\theta}/2)z}{z - \epsilon^{aT+jbT}} + \frac{(A\epsilon^{-j\theta}/2)z}{z - \epsilon^{aT-jbT}} = \frac{k_1 z}{z - p_1} + \frac{k_1^* z}{z - p_1^*} \end{aligned} \quad (2-28)$$

where the asterisk indicates the complex conjugate.

The usual partial-fraction expansion yields terms in the form of (2-28). Hence, given the partial-fraction coefficient  $k_1$  and the pole  $p_1$  in (2-28), we can solve for the discrete-time function of (2-27) using the following relationship from (2-28):

$$p_1 = \epsilon^{aT} \epsilon^{jbT} = \epsilon^{aT} \angle(bT) \Rightarrow aT = \ln |p_1|; \quad bT = \arg p_1 \quad (2-29)$$

and

$$k_1 = \frac{A\epsilon^{j\theta}}{2} = \frac{A}{2} \angle\theta \Rightarrow A = 2|k_1|; \quad \theta = \arg k_1 \quad (2-30)$$

Hence we calculate  $aT$  and  $bT$  from the poles, and  $A$  and  $\theta$  from the partial-fraction expansion. We can then express the inverse transform as the sinusoid of (2-27). An illustrative example is given next.

### EXAMPLE 2.14

We find the inverse  $z$ -transform of the function

$$Y(z) = \frac{-3.894z}{z^2 + 0.6065} = \frac{-3.894z}{(z - j0.7788)(z + j0.7788)}$$

$$= \frac{k_1 z}{z - j0.7788} + \frac{k_1^* z}{z + j0.7788}$$

Dividing both sides by  $z$ , we calculate  $k_1$ :

$$k_1 = (z - j0.7788) \left[ \frac{-3.894}{(z - j0.7788)(z + j0.7788)} \right]_{z=j0.7788}$$

$$= \frac{-3.894}{z + j0.7788} \Big|_{z=j0.7788} = \frac{-3.894}{2(j0.7788)} = 2.5 \angle 90^\circ.$$

A MATLAB program that calculates these coefficients is given by

```
>>num=[0,0,-3.894]; den=[1,0,.6065]; [r,p,k]=residue(num,den)

r =
    0 + 2.5001i
    0 - 2.5001i
p =
    0 + 0.7788i
    0 - 0.7788i
k =
    []
```

The values of  $r$  are the residues (the partial-fraction expansion coefficients) for the corresponding poles  $p$  of  $Y(z)$ , and  $k$  has values if the numerator of  $Y(z)$  is of higher order than the denominator. From (2-29) and (2-30), with  $p_1 = j0.7788$ ,

$$aT = \ln |p_1| = \ln (0.7788) = -0.250; \quad bT = \arg p_1 = \pi/2$$

$$A = 2|k_1| = 2(2.5) = 5; \quad \theta = \arg k_1 = \pi/2$$

Hence, from (2-27),

$$y(k) = A e^{a k T} \cos (b k T + \theta)$$

$$= 5 e^{-0.25 k} \cos \left( \frac{\pi}{2} k + \frac{\pi}{2} \right) = -5 e^{-0.25 k} \sin \frac{\pi}{2} k$$

This result can be verified by finding the  $z$ -transform of this function using the table in Appendix VI.

### Inversion-Formula Method

Perhaps the most general technique for obtaining the inverse of a  $z$ -transform is the inversion integral. This integral, derived via complex variable theory, is, from (2-7),

$$e(k) = \frac{1}{2\pi j} \oint_{\Gamma} E(z)z^{k-1} dz, \quad j = \sqrt{-1} \quad (2.31)$$

This expression is the line integral in the  $z$ -plane along the closed path  $\Gamma$ , where  $\Gamma$  is any path that encloses all the finite poles of  $E(z)z^{k-1}$  [5,7].

Using the theorem of residues [7], we can evaluate the integral in (2-31) via the expression

$$e(k) = \sum_{\substack{\text{at poles} \\ \text{of } [E(z)z^{k-1}]}} [\text{residues of } E(z)z^{k-1}] \quad (2.32)$$

If the function  $E(z)z^{k-1}$  has a simple pole at  $z = a$ , the residue is evaluated as

$$(\text{residue})_{z=a} = (z - a)E(z)z^{k-1} \Big|_{z=a} \quad (2.33)$$

For a pole of order  $m$  at  $z = a$ , the residue is calculated using the expression

$$(\text{residue})_{z=a} = \frac{1}{(m-1)!} \frac{d^{m-1}}{dz^{m-1}} [(z-a)^m E(z)z^{k-1}] \Big|_{z=a} \quad (2.34)$$

### EXAMPLE 2.15

Consider the function  $E(z)$  from Examples 2.12 and 2.13:

$$E(z) = \frac{z}{(z-1)(z-2)}$$

Substituting this expression into (2-32) and (2-33) yields

$$e(k) = \frac{z^k}{z-2} \Big|_{z=1} + \frac{z^k}{z-1} \Big|_{z=2} = -1 + 2^k$$

and the result is seen to be the same as that obtained in the previous examples. As in Example 2.13, let

$$E_1(z) = z^{-1}E(z) = \frac{1}{(z-1)(z-2)}$$

Then, from the real translation property,

$$e_1(k) = e(k-1)u(k-1) = [-1 + 2^{(k-1)}]u(k-1)$$



We can also find  $e_1(k)$  by the inversion formula. In (2-32),

$$E_1(z)z^{k-1} = \frac{z^{k-1}}{(z-1)(z-2)}$$

This function has a pole at  $z = 0$  only for  $k = 0$ , and thus

$$\begin{aligned} e_1(0) &= \sum_{\text{at } z=0,1,2} \left[ \text{residues of } \frac{1}{z(z-1)(z-2)} \right] \\ &= \frac{1}{2} - 1 + \frac{1}{2} = 0 \end{aligned}$$

The values of  $e_1(k)$  for  $k \geq 1$  is obtained directly from (2-32), and is left as an exercise for the reader.

The following example illustrates the inversion-formula technique for a multiple-order pole.

### EXAMPLE 2.16

The function  $E(z)$  below has a single pole of order 2 at  $z = 1$ .

$$E(z) = \frac{z}{(z-1)^2}$$

The inverse transform obtained using (2-34) is

$$\begin{aligned} e(k) &= \frac{1}{(2-1)!} \frac{d^{2-1}}{dz^{2-1}} \left[ (z-1)^2 \left( \frac{z}{(z-1)^2} \right) z^{k-1} \right] \Bigg|_{z=1} \\ &= \frac{d}{dz} (z^k) \Bigg|_{z=1} \\ &= k(z^{k-1}) \Bigg|_{z=1} \\ &= k \end{aligned}$$

Using MATLAB to perform this same inverse  $z$ -transform:

```
>>syms k z;
Ez=z/(z-1)^2;
iztrans(Ez,k)

ans =
k
```

### Discrete Convolution

The discrete convolution technique used for determining the inverse  $z$ -transform is analogous to the convolution integral employed in the use of Laplace transforms. Suppose that the function

$E(z)$  can be expressed as the product of two functions, each of which, in general, will be simpler than  $E(z)$ ; that is,

$$E(z) = E_1(z)E_2(z) \quad (2-35)$$

Further, let  $E_1(z)$  and  $E_2(z)$  be expressed as power series. Then

$$E(z) = [e_1(0) + e_1(1)z^{-1} + e_1(2)z^{-2} + \cdots][e_2(0) + e_2(1)z^{-1} + e_2(2)z^{-2} + \cdots] \quad (2-36)$$

Direct multiplication of the two power series yields

$$E(z) = e_1(0)e_2(0) + [e_1(0)e_2(1) + e_1(1)e_2(0)]z^{-1} + [e_1(0)e_2(2) + e_1(1)e_2(1) + e_1(2)e_2(0)]z^{-2} + \cdots \quad (2-37)$$

Thus the general relationship for  $e(k)$  is seen to be

$$\begin{aligned} e(k) &= e_1(0)e_2(k) + e_1(1)e_2(k-1) + \cdots + e_1(k)e_2(0) \\ &= \sum_{n=0}^k e_1(n)e_2(k-n) = \sum_{n=0}^k e_1(k-n)e_2(n) \end{aligned} \quad (2-38)$$

Equation (2-38) is the discrete convolution summation, and may be useful in determining the inverse  $z$ -transform of a function  $E(z) = E_1(z)E_2(z)$  if  $E_1(z)$  and  $E_2(z)$  are initially expressed as power series. Convolution is usually denoted as

$$e(k) = \mathcal{Z}^{-1}[E_1(z)E_2(z)] = e_1(k)*e_2(k) \quad (2-39)$$

### EXAMPLE 2.17

Once again consider the function  $E(z)$  from Example 2.12:

$$E(z) = \frac{z}{(z-1)(z-2)} = E_1(z)E_2(z)$$

where we define

$$E_1(z) = \frac{z}{z-1} = 1 + z^{-1} + z^{-2} + \cdots$$

and

$$E_2(z) = \frac{1}{z-2} = z^{-1} + 2z^{-2} + (2)^2z^{-3} + \cdots$$

These expansions are obtained by the power-series method. Then  $e(k)$  can be formed directly from (2-38). For example,  $e(3)$  is given by the expression

$$\begin{aligned}
 e(3) &= \sum_{n=0}^3 e_1(n)e_2(3-n) \\
 &= e_1(0)e_2(3) + e_1(1)e_2(2) + e_1(2)e_2(1) + e_1(3)e_2(0) \\
 &= (1)(2^2) + (1)(2) + (1)(1) + (1)(0) = 7
 \end{aligned}$$

The other values of  $e(k)$  can be obtained in a similar manner and a quick check of them will show that they agree with the values obtained via the other inversion techniques. A MATLAB program that performs a discrete convolution for this example is given by

```

>> e1 = [1 1 1 1 1 1];
e2 = [0 1 2 4 8 16];
e = conv(e1,e2)

e =
    0    1    3    7   15   31
    
```

This program correctly gives only the first six values of  $e(k)$ .

## 2.8 SIMULATION DIAGRAMS AND FLOW GRAPHS

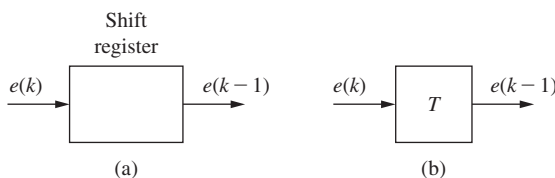
It has been shown that a linear time-variant discrete-time system may be represented by either a difference equation or a transfer function. A third representation commonly used is a simulation diagram (sometimes called a block diagram). Simulation diagrams for discrete-time systems are presented in this section.

First the basic elements used to construct simulation diagrams for a system described by a linear difference equation are developed. Let the block shown in Fig. 2-2(a) represent a shift register. Every  $T$  seconds, a number is shifted into the register, and at that instant, the number that was stored there is shifted out of the register. Therefore, if we let  $e(k)$  represent the number shifted into the register at  $t = kT$ , the number shifted out of the register is  $e(k - 1)$ . We let the symbolic representation of this memory device be as shown in Fig. 2-2(b). This symbol can represent any device that performs the foregoing operation.

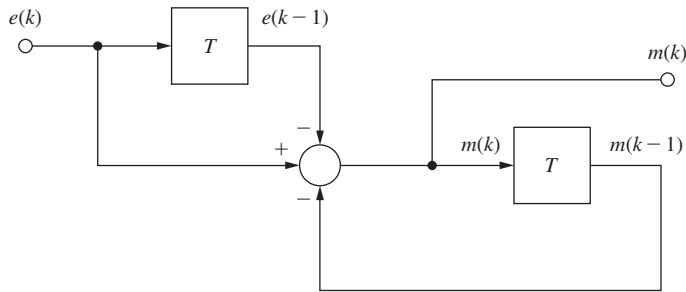
An interconnection of these devices, together with devices that perform multiplication by a constant and summation, can be used to represent a linear time-invariant difference equation. For example, consider the difference equation used in Example 2.10:

$$m(k) = e(k) - e(k - 1) - m(k - 1) \tag{2-40}$$

A simulation diagram of this equation is shown in Fig. 2-3; writing the equation for the output of the summing junction yields (2-40).



**FIGURE 2-2** Ideal time-delay element.



**FIGURE 2-3** Simulation diagram for the difference equation used in Example 2.10.

Electronic devices may be constructed to perform all the operations shown in the figure. Now suppose that such a construction exists. Then to solve Example 2.10 using this constructed machine, the numbers in both memory locations (shift registers) are set to zero and the input  $e(kT)$  is made equal to 1 at time instants  $kT = 0, 2T, 4T, \dots$  and  $e(kT)$  is made equal to 0 at time instants  $kT = T, 3T, 5T, \dots$ . The solution  $m(k)$  then appears at the output terminal at  $t = kT$ . (The reader may calculate the first few values of  $m(k)$  to illustrate the validity of the simulation diagram.) This machine would be a special-purpose computer, capable of solving only the difference equation (2-40). Recall that the computer program given below Example 2.10 also solves this difference equation, but in this case a general-purpose computer is used. The general-purpose computer software arranges the arithmetic registers, memory, and so on, to perform the operations depicted in Fig. 2-3.

To include nonzero initial conditions in our special-purpose computer described above, replace  $k$  with  $k + 1$  in (2-40)

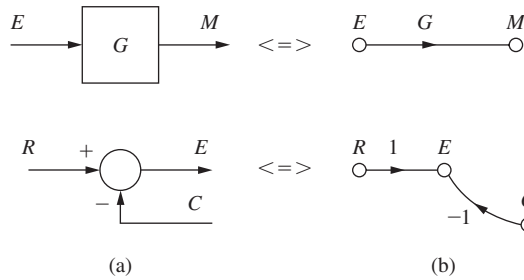
$$m(k + 1) = e(k + 1) - e(k) - m(k)$$

and in Fig. 2-3. Then the value of  $e(0)$  is placed in the register, in Fig. 2-3, whose output is now  $e(k)$ , and the value of  $m(0)$  is placed in the register whose output is now  $m(k)$ . The input values  $e(1), e(2), \dots$  are applied at the input terminal in Fig. 2-3, and the output values  $m(1), m(2), \dots$  appear at the output terminal.

Recall that in the analog simulation of continuous systems, the basic element is the integrator. In the simulation of discrete systems, the basic element is the time delay (or memory) of  $T$  seconds.

A somewhat different but equivalent graphical representation of a difference equation is the signal flow graph. A block diagram, such as that illustrated in Fig. 2-3, is simply a graphical representation of an equation, or a set of equations. The signal flow graph may also be used to graphically represent equations. The basic elements of a flow graph are the branches and the nodes. By definition, the signal out of a branch is equal to the branch gain (transfer function) times the signal into the branch. This is illustrated in Fig. 2-4, which shows both the block diagram representation and the flow graph representation of a branch. Also, by definition, the signal at a node in a flow graph is equal to sum of the signals from all branches into that node. This is also shown in Fig. 2-4. Thus a flow graph contains exactly the same information as a block diagram.

Once again, consider the time-delay device shown in Fig. 2-2(b). The  $z$ -transform of the input  $e(k)$  is  $E(z)$ , and the  $z$ -transform of the output  $e(k - 1)$  is  $z^{-1}E(z)$ . Thus the transfer function



**FIGURE 2-4** Equivalent block diagram (a) and flow graph (b) symbols.

of the time delay is  $z^{-1}$ . (Recall that the transfer function of an integrator is  $s^{-1}$ .) Consider again the system shown in Fig. 2-3. A flow graph representation of this system is as shown in Fig. 2-5. The transfer function of this system may be obtained either from Fig. 2-3 by block diagram reduction or from Fig. 2-5 by using Mason's gain formula. Those readers unfamiliar with Mason's gain-formula technique are referred to Appendix II. Basically, the gain formula is based on the geometry and signal flow directions of a flow graph. Since this architecture is exactly the same as that used in a block diagram, Mason's gain formula may also be applied directly to a block diagram.

The application of Mason's gain formula to Fig. 2-5 yields the transfer function

$$\frac{M(z)}{E(z)} = \frac{1 - z^{-1}}{1 + z^{-1}} = \frac{z - 1}{z + 1}$$

which is the same as that obtained in Example 2.11.

Consider now a general  $n$ th-order difference equation:

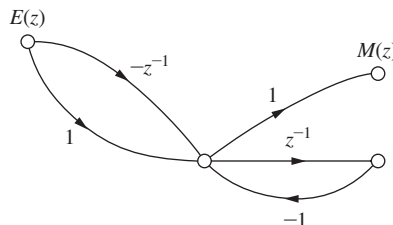
$$\begin{aligned} m(k) + a_{n-1}m(k-1) + \cdots + a_0m(k-n) \\ = b_n e(k) + b_{n-1}e(k-1) + \cdots + b_0e(k-n) \end{aligned} \quad (2-41)$$

Taking the  $z$ -transform of this equation yields

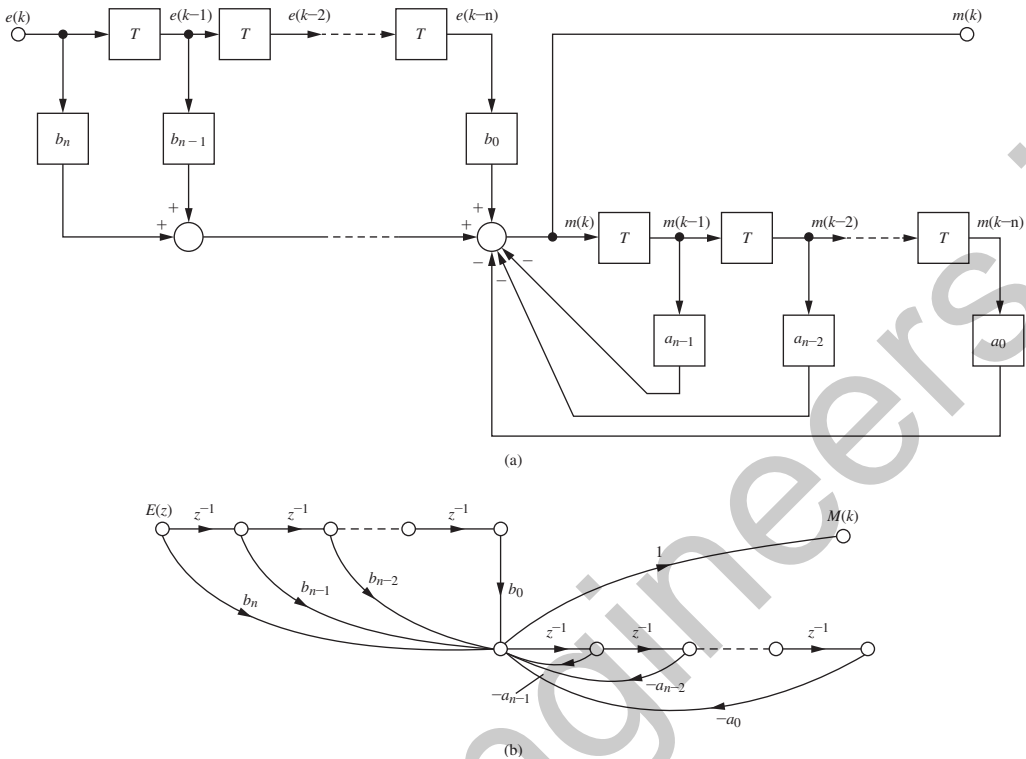
$$\begin{aligned} M(z) + a_{n-1}z^{-1}M(z) + \cdots + a_0z^{-n}M(z) \\ = b_n E(z) + b_{n-1}z^{-1}E(z) + \cdots + b_0z^{-n}E(z) \end{aligned} \quad (2-42)$$

This difference equation may then be represented by the transfer function

$$\frac{M(z)}{E(z)} = \frac{b_n + b_{n-1}z^{-1} + \cdots + b_0z^{-n}}{1 + a_{n-1}z^{-1} + \cdots + a_0z^{-n}} \quad (2-43)$$



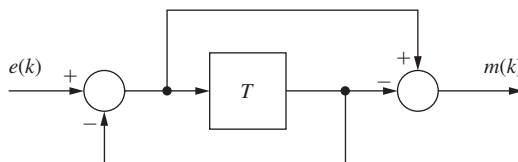
**FIGURE 2-5** Signal flow graph for the system shown in Fig. 2-3.



**FIGURE 2-6** (a) Simulation diagram for an  $n$ th-order difference equation; (b) flow graph for an  $n$ th-order difference equation.

The system of (2-41) may be represented by the simulation diagram shown in Fig. 2-6(a), since the signal out of the summing junction satisfies (2-41). The flow graph for this diagram is shown in Fig. 2-6(b). Application of Mason's gain formula to this flow graph yields (2-43).

The simulation diagram of Fig. 2-6 is only one of many that can be constructed to represent the transfer function of (2-43). The representation of Fig. 2-6 (and the example of Fig. 2-3) is non-minimal in the sense that the diagram contains  $2n$  delays, but an  $n$ th-order system can be represented with only  $n$  delays. A minimal representation of the system of Fig. 2-3 is given in Fig. 2-7. This representation may be verified by calculating the transfer function via Mason's gain formula. This brings us to the topic of state variables for discrete systems, which is closely related to simulation diagrams. The remainder of this chapter is devoted to the topic of state variables.



**FIGURE 2-7** A different simulation diagram for Example 2.10.

## 2.9 STATE VARIABLES

In preceding sections we defined a discrete-time system as one that can be described by a difference equation. If the discrete-time system is linear and time invariant, we can also represent the system by a transfer function. For a linear time-invariant discrete-time system with input  $E(z)$ , output  $M(z)$ , and transfer function  $G(z)$ , we can write

$$M(z) = G(z)E(z) \tag{2-44}$$

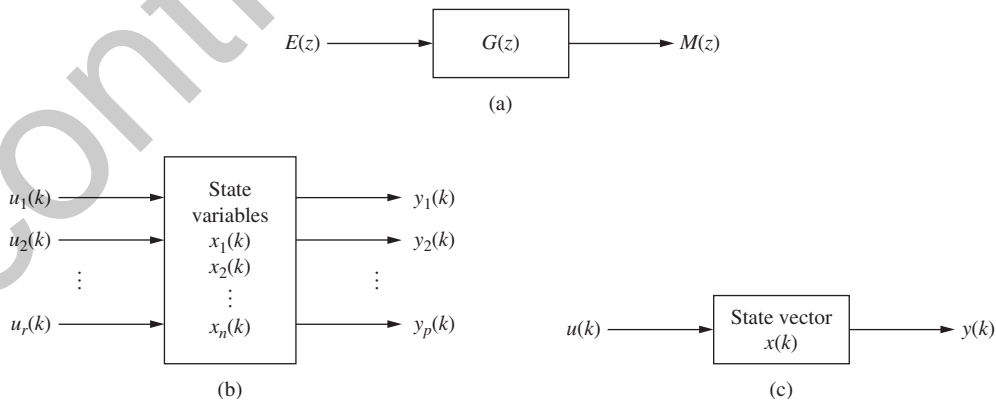
as illustrated in Section 2.8. Thus the discrete-time system may be represented by the block diagram of Fig. 2-8(a).

The more modern approach to the analysis and synthesis of discrete-time systems employs what is commonly called the state-variable method. In this approach the system is modeled as shown in Fig. 2-8(b). To be completely general, we must allow for the possibility of more than one input, and of more than one output. Thus, in Fig. 2-8(b), the variables  $u_i(k)$ ,  $i = 1, \dots, r$ , are the external inputs which drive the system; the variables  $y_i(k)$ ,  $i = 1, \dots, p$ , represent the system outputs or the system responses; and the variables  $x_i(k)$ ,  $i = 1, \dots, n$ , are the internal or state variables of the system. The state variables represent the minimum amount of information which is necessary to determine both the future states and the system outputs for given input functions; that is, given the system states, the system dynamics and the input functions, we can determine all subsequent states and outputs.

For convenience we represent the system shown in Fig. 2-8(b) by that shown in Fig. 2-8(c), where  $\mathbf{u}(k)$  is the input vector,  $\mathbf{y}(k)$  is the output vector, and  $\mathbf{x}(k)$  is the state vector:

$$\mathbf{u}(k) = \begin{bmatrix} u_1(k) \\ u_2(k) \\ \vdots \\ u_r(k) \end{bmatrix}, \quad \mathbf{y}(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \\ \vdots \\ y_p(k) \end{bmatrix}, \quad \mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix}$$

The set of values that the input vector  $\mathbf{u}(k)$  may assume is called the input space of the system. The output space and state space are defined in a similar manner.



**FIGURE 2-8** Representations of system dynamics: (a) z-transfer function representation; (b) state-variable representation; (c) state-vector representation.

In general, the equation that describes the state of the system at any time  $(k + 1)$  is given by the single-valued function relationship

$$\mathbf{x}(k + 1) = \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k)] \quad (2-45)$$

This equation simply states that the state  $\mathbf{x}$  at time  $(k + 1)$  is a function of the state and the input at the previous discrete-time increment  $k$ . The output response of the system is defined in a similar manner as

$$\mathbf{y}(k) = \mathbf{g}[\mathbf{x}(k), \mathbf{u}(k)] \quad (2-46)$$

As stated earlier, when we say time  $k$ , we actually mean time  $kT$ .

If the system is linear, then equations (2-45) and (2-46) reduce to

$$\mathbf{x}(k + 1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad (2-47)$$

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k) \quad (2-48)$$

where  $\mathbf{x}(k)$  is an  $n$ -vector,  $\mathbf{u}(k)$  is an  $r$ -vector,  $\mathbf{y}(k)$  is a  $p$ -vector (as shown in Fig. 2-8), and  $\mathbf{A}(k)$ ,  $\mathbf{B}(k)$ ,  $\mathbf{C}(k)$ , and  $\mathbf{D}(k)$  are time-varying matrices of dimensions  $n \times n$ ,  $n \times r$ ,  $p \times n$ , and  $p \times r$ , respectively. If the system is time invariant, then the matrices in (2-47) and (2-48) are constant, and hence the equations reduce to

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (2-49)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (2-50)$$

(See Appendix IV for a review of matrices.) State-variable modeling will now be illustrated by an example.

### EXAMPLE 2.18

It is desired to find a state-variable model of the system described by the difference equation

$$y(k + 2) = u(k) + 1.7y(k + 1) - 0.72y(k)$$

Let

$$x_1(k) = y(k)$$

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

Then

$$x_2(k + 1) = y(k + 2) = u(k) + 1.7x_2(k) - 0.72x_1(k)$$

or, from these equations, we write



$$\begin{aligned}
 x_1(k + 1) &= x_2(k) \\
 x_2(k + 1) &= -0.72x_1(k) + 1.7x_2(k) + u(k) \\
 y(k) &= x_1(k)
 \end{aligned}$$

We may express these equations in vector-matrix form of (2-49) and (2-50) as

$$\begin{aligned}
 \mathbf{x}(k + 1) &= \begin{bmatrix} 0 & 1 \\ -0.72 & 1.7 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} u(k) \\
 y(k) &= [1 \quad 0] \mathbf{x}(k)
 \end{aligned}$$

Equations (2-49) and (2-50) are the state equations for a linear time-invariant system and usually represent the starting point in the analysis or design of a discrete system by modern methods. However, let us first examine the connection between this approach and the  $z$ -transform method. To do this we will give one method for deriving a set of discrete state-variable equations from the  $z$ -transform transfer function.

Given the transfer function

$$G(z) = \frac{b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \cdots + b_1z + b_0}{z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0} \quad (2-51)$$

in which we have assumed that the order of the numerator is less than that of the denominator. We can write (2-51) as

$$\frac{Y(z)}{U(z)} = G(z) = \frac{b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \cdots + b_1z + b_0}{z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0} \frac{E(z)}{E(z)} \quad (2-52)$$

where the auxiliary variable  $E(z)$  has been introduced. We now let

$$Y(z) = (b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \cdots + b_1z + b_0)E(z) \quad (2-53)$$

$$U(z) = (z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0)E(z) \quad (2-54)$$

Recall from the real translation property the correspondence

$$\begin{aligned}
 E(z) &\rightarrow e(k) \\
 zE(z) &\rightarrow e(k + 1) \\
 z^2E(z) &\rightarrow e(k + 2) \\
 &\vdots
 \end{aligned}$$

Under this correspondence we define the state variables

$$\begin{aligned}
 x_1(k) &= e(k) \\
 x_2(k) &= x_1(k + 1) = e(k + 1) \\
 x_3(k) &= x_2(k + 1) = e(k + 2) \\
 &\vdots \\
 x_n(k) &= x_{n-1}(k + 1) = e(k + n - 1)
 \end{aligned} \quad (2-55)$$

From equations (2-54) and (2-55) we obtain the state equations

$$\begin{aligned}
 x_1(k+1) &= x_2(k) \\
 x_2(k+1) &= x_3(k) \\
 x_3(k+1) &= x_4(k) \\
 &\vdots \\
 x_n(k+1) &= -a_0x_1(k) - a_1x_2(k) - a_2x_3(k) - \cdots - a_{n-1}x_n(k) + u(k)
 \end{aligned} \tag{2-56}$$

which written in matrix form is

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\ & & & \vdots & & & \\ -a_0 & -a_1 & -a_2 & -a_3 & -a_4 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(k) \tag{2-57}$$

or simply

$$\mathbf{x}(k+1) = \mathbf{Ax}(k) + \mathbf{Bu}(k) \tag{2-58}$$

The output equation obtained from (2-53) is

$$y(k) = [b_0 \ b_1 \ \cdots \ b_{n-1}] \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} \tag{2-59}$$

or simply

$$y(k) = \mathbf{Cx}(k) \tag{2-60}$$

Hence equations (2-58) and (2-60) are a set of state equations for the discrete-time system described by equation (2-51).

Another convenient and useful representation of the discrete system is the signal flow graph or the equivalent simulation diagram. These two forms can be derived from equation (2-52)

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_{n-1}z^{-1} + b_{n-2}z^{-2} + \cdots + b_1z^{1-n} + b_0z^{-n}}{1 + a_{n-1}z^{-1} + \cdots + a_1z^{1-n} + a_0z^{-n}} \frac{E(z)}{E(z)} \tag{2-61}$$

From this expression we obtain the two equations

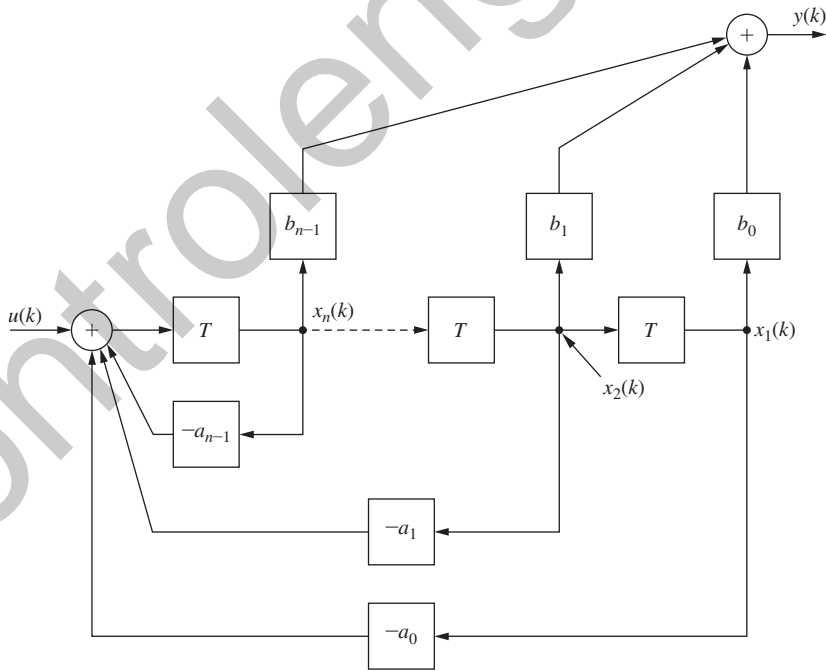
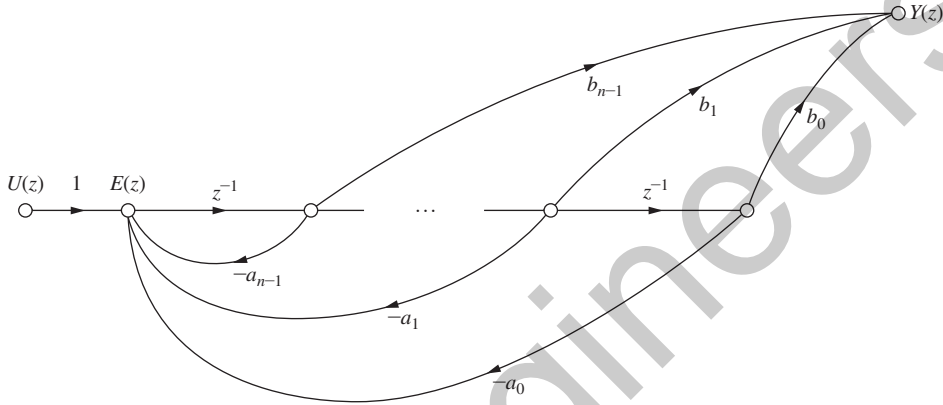
$$Y(z) = (b_{n-1}z^{-1} + b_{n-2}z^{-2} + \cdots + b_1z^{1-n} + b_0z^{-n})E(z) \tag{2-62}$$

$$U(z) = (1 + a_{n-1}z^{-1} + \cdots + a_1z^{1-n} + a_0z^{-n})E(z) \tag{2-63}$$

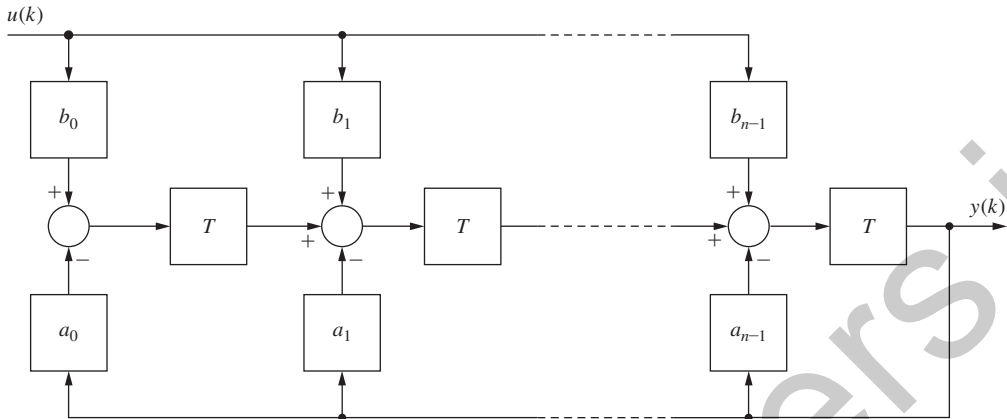
Equation (2-63) can be written as

$$E(z) = U(z) - a_{n-1}z^{-1}E(z) - \dots - a_1z^{1-n}E(z) - a_0z^{-n}E(z) \quad (2-64)$$

A signal flow graph representation of the system described by equation (2-51) can be easily derived from equations (2-62) and (2-64) and is shown in Fig. 2-9(a).



**FIGURE 2-9** Equivalent control canonical form (or phase variable canonical form) representations of equation (2-51): (a) signal flow graph representation; (b) simulation diagram.



**FIGURE 2-10** Observer canonical form.

Recall that the transfer function for a pure delay of  $T$  seconds is  $z^{-1}$ , where  $T$  is the discrete-time increment (i.e.,  $T$  is the interval between sampling instants  $k, k + 1, k + 2$ , etc.). Using this relationship, the signal flow graph of Fig. 2-9(a) can be immediately converted to the equivalent simulation diagram of Fig. 2-9(b). We see from (2-57) and (2-59) that the states are the register outputs, as noted in Fig. 2-9(b).

Note the relationships that exist between equations (2-57) and (2-59) and the diagrams in Fig. 2-9. One who is very familiar with the correspondence between the equations and diagrams can derive a flow graph or simulation diagram from the transfer function by inspection. From the flow graph, one may use Mason's gain formula to reconstruct the original transfer function. The structure of Fig. 2-9, together with equations (2-57) and (2-59), is called either the control canonical form or the phase variable canonical form, and is useful in the design procedures presented in Chapters 9 and 11.

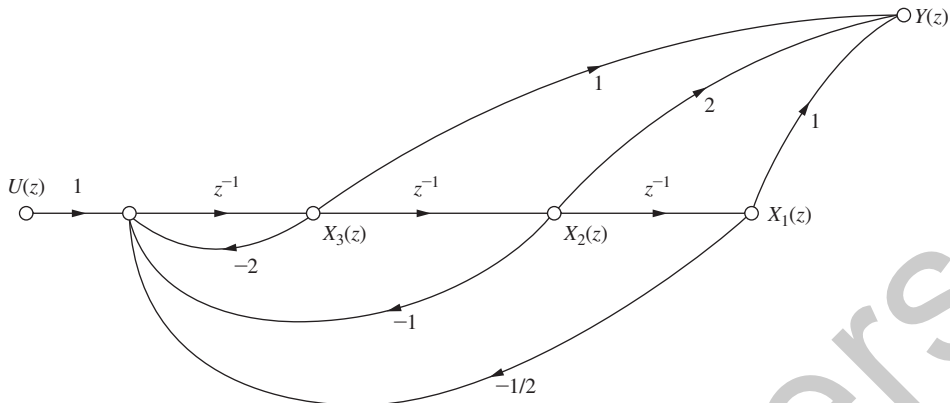
Another standard form that is useful in the design procedures of Chapter 9 is the observer canonical form, which is illustrated in Fig. 2-10. Note, using Mason's gain formula, that the transfer function for Fig. 2-10 is (2-61). It is left as an exercise for the reader to write the state equations for the observer canonical form (see Problem 2.9-2).

### EXAMPLE 2.19

Given the following function, we want to derive a signal flow graph and corresponding state equations:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{z^2 + 2z + 1}{z^3 + 2z^2 + z + \frac{1}{2}}$$

Comparing this expression with (2-51) and Fig. 2-9(a) indicates that the signal flow graph can be derived by inspection, as shown in Fig. 2-11. Note that the states are then chosen to be the delay outputs. The state equations can now be derived from Fig. 2-11 or from a comparison



**FIGURE 2-11** Flow graph for Example 2.19.

of the given transfer function with equations (2-51), (2-57), and (2-59). The resulting state equations are

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{1}{2} & -1 & -2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 2 \quad 1] \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}$$

The following example will demonstrate an important salient feature encountered in going from the transfer function to the state equations.

### EXAMPLE 2.20

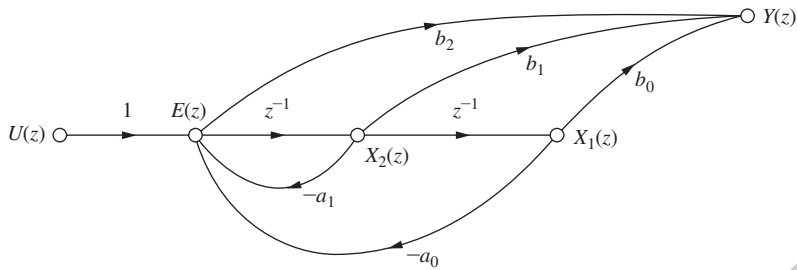
Given the following transfer function, we wish to derive the state equations:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_2 z^2 + b_1 z + b_0}{z^2 + a_1 z + a_0}$$

In this case, the order of the numerator is equal to that of the denominator. First we express the transfer function as

$$\frac{Y(z)}{U(z)} = \frac{b_2 + b_1 z^{-1} + b_0 z^{-2} E(z)}{1 + a_1 z^{-1} + a_0 z^{-2} E(z)}$$

We see from Mason's gain formula that the signal flow graph for this transfer function is as shown in Fig. 2-12. The states are shown on the flow graph, and the equations are seen to be



**FIGURE 2-12** Flow graph for Example 2.20.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

At this point it is important to note that the flow graph in Fig. 2-12 is slightly different from the form presented earlier; that is, there is a direct path from  $E(z)$  to  $Y(z)$ . From the flow graph the equation for the output is

$$Y(z) = b_0X_1(z) + b_1X_2(z) + b_2E(z)$$

But

$$E(z) = U(z) - a_0X_1(z) - a_1X_2(z)$$

From these equations we obtain

$$Y(z) = b_2U(z) + (b_0 - b_2a_0)X_1(z) + (b_1 - b_2a_1)X_2(z)$$

Hence the output equation is

$$y(k) = [(b_0 - b_2a_0) \quad (b_1 - b_2a_1)] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + b_2u(k)$$

Note the difference in this equation and (2-59).

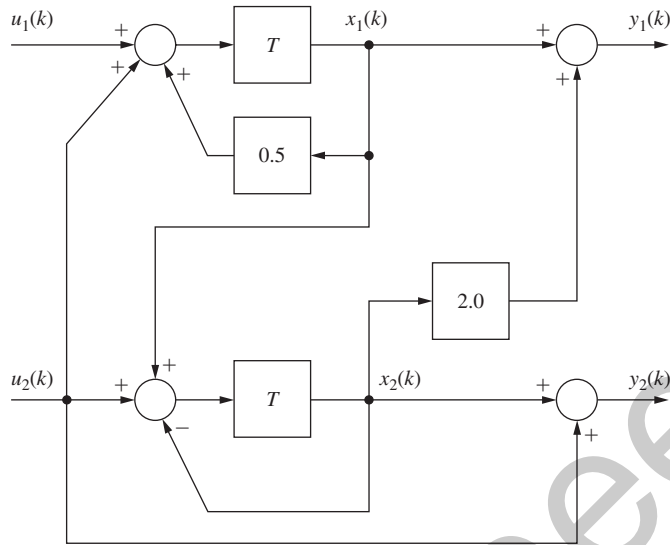
Whenever the numerator and denominator of the transfer function are of the same order, the output state equation must be handled as demonstrated above. Also, the output equation may be written by inspection from Fig. 2-12.

### EXAMPLE 2.21

We wish to derive the state equations for the multivariable discrete-time system shown in Fig. 2-13. The state equation can be obtained directly from the simulation diagram by inspection.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$



**FIGURE 2-13** Simulation diagram for a multivariable system.

In summary, state models may be derived from transfer functions by the procedure illustrated in (2-52) through (2-59). A second procedure is:

1. Draw a simulation diagram of the system, using any convenient method.
2. Assign a state variable to each delay output.
3. Write the equation for each delay input and each system output in terms of only the delay outputs and the system input.

The system transfer function gives an input–output description of the system, while the state model gives an internal description in addition to the input–output description. The transfer function for a given system is unique, while the internal model (state model) is not.

## 2.10 OTHER STATE-VARIABLE FORMULATIONS

We have seen in the preceding section the techniques for finding a state-variable formulation for a single-input, single-output discrete-time system, given either the system difference equation or the system transfer function. For a given system, there is no unique state-variable formulation. For a given transfer function, any simulation diagram with the system transfer function yields a valid state-variable model for the system. However, for certain analysis or design procedures, certain formulations present advantages with respect to calculations, as will be shown in later chapters. The following example illustrates some derivations of different formulations.

### EXAMPLE 2.22

Consider the difference equation of Example 2.18:

$$y(k + 2) = u(k) + 1.7y(k + 1) - 0.72y(k)$$

One state-variable model was derived in that example. Two additional models are derived here. Taking the z-transform of the difference equation yields

$$\frac{Y(z)}{U(z)} = \frac{1}{z^2 - 1.7z + 0.72}$$

This transfer function can be expressed as

$$\frac{Y(z)}{U(z)} = \left[ \frac{1}{z - 0.9} \right] \left[ \frac{1}{z - 0.8} \right]$$

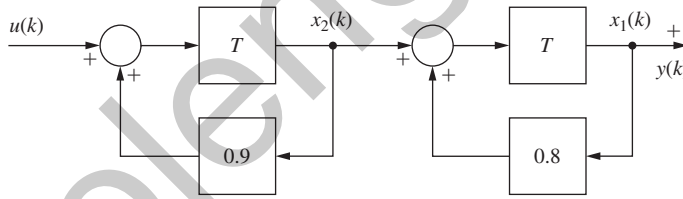
A simulation diagram for this transfer function is given in Fig. 2-14(a). From this figure we write the state equations

$$\mathbf{x}(k + 1) = \begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

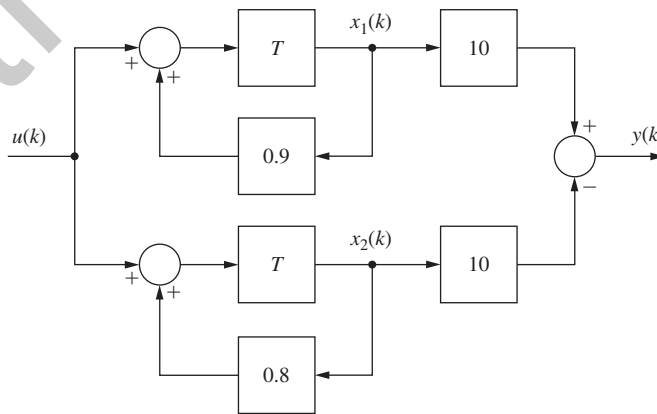
$$y(k) = [1 \quad 0] \mathbf{x}(k)$$

Also, the system transfer function can be expressed, through partial-fraction expansion, as

$$\frac{Y(z)}{U(z)} = \frac{10}{z - 0.9} + \frac{-10}{z - 0.8}$$



(a)



(b)

**FIGURE 2-14** Simulation diagrams for Example 2.22.



A simulation diagram for this transfer function is given in Fig. 2-14(b). From this figure we write the state equations

$$\mathbf{x}(k + 1) = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.8 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [10 \quad -10] \mathbf{x}(k)$$

In Example 2.22, we first represented a higher-order transfer function as a product of simpler transfer functions. For the general case, we would express the transfer function  $G(z)$  as

$$G(z) = G_1(z)G_2(z) \cdots G_n(z) \quad (2-65)$$

Each of the simpler transfer functions  $G_i(z)$  is realized by a technique of the preceding section, and the realizations are then connected in cascade to realize  $G(z)$ . If  $G(z)$  contains either complex poles or complex zeros, we may choose some of the  $G_i(z)$  in (2-65) to be second order, to avoid computational difficulties with complex elements in the state-variable matrices.

Next, in Example 2.22, we represented a higher-order transfer function as the sum of simpler transfer functions through partial-fraction expansion. For the general case, we would express the transfer function  $G(z)$  as

$$G(z) = G_{1p}(z) + G_{2p}(z) + \cdots + G_{np}(z) \quad (2-66)$$

Each of the simpler transfer functions  $G_{ip}(z)$  is realized by a technique of Section 2.9, and the realizations are then connected in parallel to realize  $G(z)$ .

We have just seen three methods for arriving at different state models for a discrete-time system, with these models derived from the system transfer function. In fact, we can derive any number of different state models, given one state model of the system, through *similarity transformations*. This procedure will now be developed. Consider the state equation in the form

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}u(k) \quad (2-67)$$

Now, we can apply the linear transformation

$$\mathbf{x}(k) = \mathbf{P}\mathbf{w}(k) \quad (2-68)$$

that is,

$$x_1(k) = p_{11}w_1(k) + p_{12}w_2(k) + \cdots + p_{1n}w_n(k)$$

and so on. Hence  $\mathbf{P}$  is a constant ( $n \times n$ ) matrix and  $\mathbf{w}(k)$  is the new state vector. Note that it is necessary that  $\mathbf{P}^{-1}$ , the inverse of  $\mathbf{P}$ , exists so that  $\mathbf{w}(k)$  can be determined from  $\mathbf{x}(k)$ . Substituting (2-68) into (2-67) yields the equations

$$\mathbf{w}(k + 1) = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\mathbf{w}(k) + \mathbf{P}^{-1}\mathbf{B}u(k)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{P}\mathbf{w}(k) + \mathbf{D}u(k) \quad (2-69)$$

These equations can be expressed as

$$\begin{aligned} \mathbf{w}(k + 1) &= \mathbf{A}_w \mathbf{w}(k) + \mathbf{B}_w \mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}_w \mathbf{w}(k) + \mathbf{D}_w \mathbf{u}(k) \end{aligned} \quad (2-70)$$

where

$$\begin{aligned} \mathbf{A}_w &= \mathbf{P}^{-1} \mathbf{A} \mathbf{P}, & \mathbf{B}_w &= \mathbf{P}^{-1} \mathbf{B} \\ \mathbf{C}_w &= \mathbf{C} \mathbf{P}, & \mathbf{D}_w &= \mathbf{D} \end{aligned} \quad (2-71)$$

Thus for each different  $\mathbf{P}$  for which  $\mathbf{P}^{-1}$  exists, a different state model of a given system can be found.

The *characteristic equation* of a matrix  $\mathbf{A}$  is defined by the determinant [8]

$$|z\mathbf{I} - \mathbf{A}| = 0 \quad (2-72)$$

and the *characteristic values* (or *eigenvalues*) of the matrix are the roots of the characteristic equation; that is,  $z_i$  are the characteristic values of  $\mathbf{A}$  if

$$|z\mathbf{I} - \mathbf{A}| = (z - z_1)(z - z_2) \cdots (z - z_n) = 0 \quad (2-73)$$

It will be shown in Chapter 7 that (2-73) determines system stability for the system of (2-67). Note that the characteristic equation of the system matrix is unchanged through the linear transformation,

$$\begin{aligned} |z\mathbf{I} - \mathbf{A}_w| &= |z\mathbf{I} - \mathbf{P}^{-1} \mathbf{A} \mathbf{P}| = |z\mathbf{P}^{-1} \mathbf{I} \mathbf{P} - \mathbf{P}^{-1} \mathbf{A} \mathbf{P}| \\ &= |\mathbf{P}^{-1}| |z\mathbf{I} - \mathbf{A}| |\mathbf{P}| \\ &= |z\mathbf{I} - \mathbf{A}| \end{aligned} \quad (2-74)$$

A linear transformation of the form

$$\mathbf{A}_w = \mathbf{P}^{-1} \mathbf{A} \mathbf{P} \quad (2-75)$$

which is called a *similarity transformation*, has the following properties:

1. As shown above, the characteristic values of the matrix are unchanged under the transformation.
2. The determinant of  $\mathbf{A}_w$  (which is the product of characteristic values) is equal to the determinant of  $\mathbf{A}$ :

$$|\mathbf{A}_w| = |\mathbf{P}^{-1} \mathbf{A} \mathbf{P}| = |\mathbf{P}^{-1}| |\mathbf{A}| |\mathbf{P}| = |\mathbf{A}| = z_1 z_2 \cdots z_n$$

3. The trace of  $\mathbf{A}_w$  is equal to the trace of  $\mathbf{A}$ :

$$\text{tr} \mathbf{A}_w = \text{tr} \mathbf{A} = z_1 + z_2 + \cdots + z_n$$

since the trace of a matrix, which is the sum of the diagonal elements, is equal to the sum of its characteristic values. The latter property follows immediately from the first property.

4. As is shown in Section 2.11,

$$\mathbf{C} [z\mathbf{I} - \mathbf{A}]^{-1} \mathbf{B} + \mathbf{D} = \mathbf{C}_w [z\mathbf{I} - \mathbf{A}_w]^{-1} \mathbf{B}_w + \mathbf{D}_w$$

### EXAMPLE 2.23

For the system of Example 2.22, one state-variable model is given as

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k)$$

We arbitrarily choose a linear-transformation matrix

$$\mathbf{P} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

The inverse of  $\mathbf{P}$  is given by

$$\mathbf{P}^{-1} = \frac{[\text{Cof}[\mathbf{P}]]^T}{|\mathbf{P}|}$$

where  $[\cdot]^T$  indicates the transpose, and  $\text{Cof}[\mathbf{P}]$  denotes the matrix of cofactors of  $\mathbf{P}$ .

Thus

$$\text{Cof}[\mathbf{P}] = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

and

$$|\mathbf{P}| = 2$$

Then

$$\mathbf{P}^{-1} = \begin{bmatrix} 0.5 & 0.5 \\ -0.5 & 0.5 \end{bmatrix}$$

Thus, from (2-71),

$$\begin{aligned} \mathbf{A}_w &= \mathbf{P}^{-1} \mathbf{A} \mathbf{P} = \begin{bmatrix} 0.5 & 0.5 \\ -0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0.5 & 0.5 \\ -0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 1.8 & 0.2 \\ 0.9 & 0.9 \end{bmatrix} = \begin{bmatrix} 1.35 & 0.55 \\ -0.45 & 0.35 \end{bmatrix} \end{aligned}$$

$$\mathbf{B}_w = \mathbf{P}^{-1} \mathbf{B} = \begin{bmatrix} 0.5 & 0.5 \\ -0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

$$\mathbf{C}_w = \mathbf{C} \mathbf{P} = [1 \quad 0] \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = [1 \quad -1]$$

Then the new state equations are

$$\mathbf{w}(k+1) = \begin{bmatrix} 1.35 & 0.55 \\ -0.45 & 0.35 \end{bmatrix} \mathbf{w}(k) + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad -1] \mathbf{w}(k)$$

Note that

$$|z\mathbf{I} - \mathbf{A}| = \begin{vmatrix} z - 0.8 & -1 \\ 0 & z - 0.9 \end{vmatrix} = z^2 - 1.7z + 0.72 = (z - 0.8)(z - 0.9)$$

and

$$|z\mathbf{I} - \mathbf{A}_w| = \begin{vmatrix} z - 1.35 & -0.55 \\ 0.45 & z - 0.35 \end{vmatrix} = z^2 - 1.7z + 0.72$$

Hence the characteristic values are  $z_1 = 0.8$  and  $z_2 = 0.9$ . Also,

$$|\mathbf{A}| = |\mathbf{A}_w| = 0.72 = z_1 z_2$$

and

$$\text{tr } \mathbf{A} = \text{tr } \mathbf{A}_w = 1.7 = z_1 + z_2$$

The results of this example can be verified using MATLAB.

MATLAB may be also used to find new state-space formulations using similarity transforms. Consider the third-order example below.

### EXAMPLE 2.24

Given

$$G(z) = \frac{5z^2 + 10.5z + 5.8}{z^3 + 1.1z^2 + 0.5z + 0.12}$$

we can write the state matrices for the control canonical form by inspection using (2-57) and (2-59):

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.12 & -0.5 & -1.1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{C} = [5.8 \quad 10.5 \quad 5]$$

Arbitrarily choosing matrix  $\mathbf{P}$

$$\mathbf{P} = \begin{bmatrix} 3 & 8 & 4 \\ 0 & 7 & 5 \\ 9 & 6 & 2 \end{bmatrix}$$

Using MATLAB, we can verify that  $\mathbf{P}$  has an inverse, and then find a new state-space formulation using matrix  $\mathbf{P}$  in (2-71):

```

>>A=[0 1 0;0 0 1;-.12 -.5 -1.1];
B=[0 0 1]';
C=[5.8,10.5,5];
D=0;
P=[3 8 4;0 7 5;9 6 2];
invP=inv(P)
    
```

$$\begin{aligned}
 A_w &= \text{inv}(P) * A * P \\
 B_w &= \text{inv}(P) * B \\
 C_w &= C * P \\
 D_w &= D
 \end{aligned}$$

$$\text{inv}P = \begin{bmatrix} -0.2667 & 0.1333 & 0.2000 \\ 0.7500 & -0.5000 & -0.2500 \\ -1.0500 & 0.9000 & 0.3500 \end{bmatrix}$$

$$A_w = \begin{bmatrix} -0.8520 & -3.2787 & -2.1027 \\ -1.9350 & 5.0150 & 4.0450 \\ 4.5090 & -5.8210 & -5.2630 \end{bmatrix}$$

$$B_w = \begin{bmatrix} 0.2000 \\ -0.2500 \\ 0.3500 \end{bmatrix}$$

$$C_w = \begin{bmatrix} 62.4000 & 149.9000 & 85.7000 \end{bmatrix}$$

$$D_w = \begin{bmatrix} 0 \end{bmatrix}$$

If a system has distinct characteristic values, we may derive a state-variable model in which the system matrix is diagonal. Consider a vector  $\mathbf{m}_i$  and a scalar  $z_i$  defined by the equation

$$\mathbf{A}\mathbf{m}_i = z_i\mathbf{m}_i \tag{2-76}$$

where  $\mathbf{m}_i = [m_{1i} \ m_{2i} \ \cdots \ m_{ni}]^T$ . We express (2-76) as

$$(z_i\mathbf{I} - \mathbf{A})\mathbf{m}_i = 0$$

For a nontrivial solution for this equation to exist, it is required that

$$|z_i\mathbf{I} - \mathbf{A}| = 0$$

Hence, from (2-72), it is seen that  $z_i$  is a characteristic value (also called an *eigenvalue*) of  $\mathbf{A}$ . The vector  $\mathbf{m}_i$  is called a *characteristic vector*, or *eigenvector*, of  $\mathbf{A}$ . From (2-76) we can construct the equation

$$\mathbf{A}[\mathbf{m}_1 \ \mathbf{m}_2 \ \cdots \ \mathbf{m}_n] = [\mathbf{m}_1 \ \mathbf{m}_2 \ \cdots \ \mathbf{m}_n] \begin{bmatrix} z_1 & 0 & \cdot & 0 \\ 0 & z_2 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & z_n \end{bmatrix}$$

or

$$\mathbf{A}\mathbf{M} = \mathbf{M}\mathbf{\Lambda}$$

where  $\mathbf{M}$ , called the *modal matrix*, is composed of the characteristic vectors as columns, and  $\mathbf{\Lambda}$  is a diagonal matrix with the characteristic values of  $\mathbf{A}$  as the diagonal elements. The characteristic

vectors are linearly independent provided the characteristic values are distinct. Hence we can write

$$\Lambda = \mathbf{M}^{-1}\mathbf{A}\mathbf{M} \quad (2-77)$$

Then, in the general similarity transformation,  $\mathbf{P}$  is equal to  $\mathbf{M}$ , the modal matrix. An example will be given to illustrate this procedure.

### EXAMPLE 2.25

For the system of Example 2.23

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k) \end{aligned}$$

From Example 2.23, the characteristic values are  $z_1 = 0.8$  and  $z_2 = 0.9$ . From (2-76),

$$\begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{21} \end{bmatrix} = 0.8 \begin{bmatrix} m_{11} \\ m_{21} \end{bmatrix}$$

Thus

$$\begin{aligned} 0.8m_{11} + m_{21} &= 0.8m_{11} \\ 0.9m_{21} &= 0.8m_{21} \end{aligned}$$

Hence  $m_{21} = 0$  and  $m_{11}$  is arbitrary. Let  $m_{11} = 1$ . Also,

$$\begin{bmatrix} 0.8 & 1 \\ 0 & 0.9 \end{bmatrix} \begin{bmatrix} m_{12} \\ m_{22} \end{bmatrix} = 0.9 \begin{bmatrix} m_{12} \\ m_{22} \end{bmatrix}$$

Then  $m_{22}$  is seen to be arbitrary; let  $m_{22} = 1$ . Then  $m_{12} = 10$ . The modal matrix and its inverse are

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} = \begin{bmatrix} 1 & 10 \\ 0 & 1 \end{bmatrix}$$

and

$$\mathbf{M}^{-1} = \begin{bmatrix} 1 & -10 \\ 0 & 1 \end{bmatrix}$$

From (2-71) and (2-77),

$$\Lambda = \mathbf{M}^{-1}\mathbf{A}\mathbf{M} = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.9 \end{bmatrix}$$

$$\mathbf{B}_w = \mathbf{M}^{-1}\mathbf{B} = \begin{bmatrix} -10 \\ 1 \end{bmatrix}$$

$$\mathbf{C}_w = \mathbf{C}\mathbf{M} = [1 \quad 10]$$

Hence the new state model is given by

$$\mathbf{w}(k + 1) = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.9 \end{bmatrix} \mathbf{w}(k) + \begin{bmatrix} -10 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 10] \mathbf{w}(k)$$

A MATLAB program that calculates the characteristic vectors and the characteristic values is given by

```

>> A = [0.8 1; 0 0.9];
[M,LAMBDA] = eig(A)

M =
    1.0000    0.9950
         0    0.0995

LAMBDA =
    0.8000    0
         0    0.9000
    
```

The matrix M gives the characteristic vectors as columns (each determined only within a constant factor), and LAMBDA gives the characteristic values on its diagonal.

Now let us demonstrate using MATLAB to generate a new state-space representation in which  $\mathbf{A}_w = \Lambda$ .

### EXAMPLE 2.26

Given

$$G(z) = \frac{10z - 5}{z^3 - 2.1z^2 + 1.4z - 0.288}$$

we can write the state matrices for this function from (2-57):

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.288 & -1.4 & 2.1 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{C} = [-5 \quad 10 \quad 0] \quad \mathbf{D} = [0]$$

Now we can employ MATLAB to compute (2-71) with  $\mathbf{P} = \mathbf{M}$ :

```

>>A=[0 1 0;0 0 1;0.288 -1.4 2.1];
B=[0 0 1]';
C=[-5 10 0];
D=0;
[M,LAMBDA]=eig(A);
P=M
Aw=inv(P)*A*P
Bw=inv(P)*B
    
```

$$C_w = C * P$$

$$D_w = D$$

$$P =$$

$$\begin{bmatrix} -0.9184 & -0.6985 & -0.6368 \\ -0.3674 & -0.5588 & -0.5731 \\ -0.1469 & -0.4470 & -0.5158 \end{bmatrix}$$

$$A_w =$$

$$\begin{bmatrix} 0.4000 & 0.0000 & 0.0000 \\ 0.0000 & 0.8000 & 0.0000 \\ -0.0000 & -0.0000 & 0.9000 \end{bmatrix}$$

$$B_w =$$

$$\begin{bmatrix} -5.4443 \\ 35.7911 \\ -31.4076 \end{bmatrix}$$

$$C_w =$$

$$[0.9184 \quad -2.0955 \quad -2.5472]$$

$$D_w =$$

$$0$$

## 2.11 TRANSFER FUNCTIONS

In the techniques described earlier in this text for obtaining a state-variable formulation, we derived the state model from the transfer function. In this section we demonstrate how one may obtain the transfer function, given the state model.

As a first approach, given the state equations of a discrete-time system, we can construct a simulation diagram. The transfer function can then be obtained from the simulation diagram, using Mason's gain formula.

### EXAMPLE 2.27

Consider the state model derived in Example 2.23.

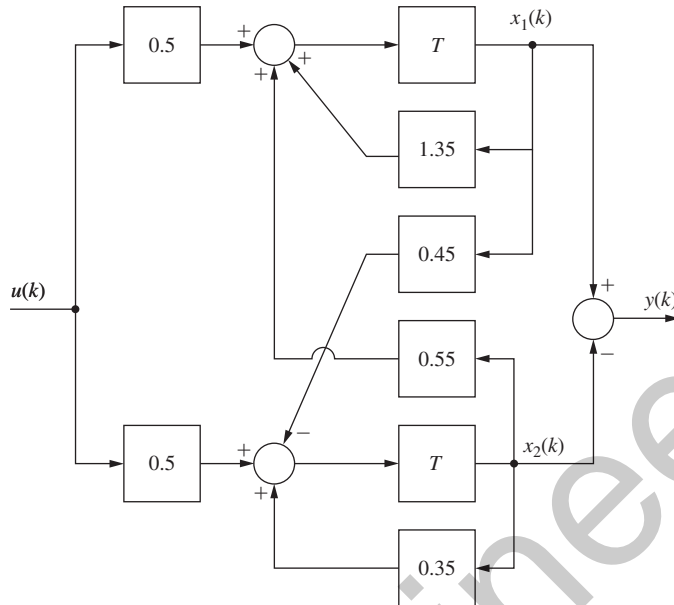
$$\mathbf{x}(k+1) = \begin{bmatrix} 1.35 & 0.55 \\ -0.45 & 0.35 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad -1] \mathbf{x}(k)$$

A simulation diagram, constructed from these equations, is shown in Fig. 2-15. Application of Mason's gain formula to this figure yields

$$\begin{aligned} \frac{Y(z)}{U(z)} &= \frac{0.5z^{-1}(1 - 0.35z^{-1}) - 0.5z^{-1}(1 - 1.35z^{-1}) + (0.5)(0.45)z^{-2} + (0.5)(0.55)z^{-2}}{1 - 1.35z^{-1} - 0.35z^{-1} + (0.45)(0.55)z^{-2} + (1.35z^{-1})(0.35z^{-1})} \\ &= \frac{z^{-2}}{1 - 1.7z^{-1} + 0.72z^{-2}} = \frac{1}{z^2 - 1.7z + 0.72} \end{aligned}$$





**FIGURE 2-15** System for Example 2.27.

Since this is the system transfer function, we see that the derived state model is valid.

Another approach for obtaining the system transfer function from the state equations is to take the  $z$ -transform of the state equations and eliminate  $\mathbf{X}(z)$ . Since

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \quad (2-78)$$

taking the  $z$ -transform yields

$$z\mathbf{X}(z) - z\mathbf{x}(0) = \mathbf{A}\mathbf{X}(z) + \mathbf{B}U(z) \quad (2-79)$$

Since, in deriving transfer functions, we ignore initial conditions, (2-79) can be expressed as

$$[z\mathbf{I} - \mathbf{A}]\mathbf{X}(z) = \mathbf{B}U(z) \quad (2-80)$$

Solving for  $\mathbf{X}(z)$ , we obtain

$$\mathbf{X}(z) = [z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B}U(z) \quad (2-81)$$

Also, since

$$y(k) = \mathbf{C}\mathbf{x}(k) + Du(k) \quad (2-82)$$

then

$$Y(z) = \mathbf{C}\mathbf{X}(z) + DU(z) \quad (2-83)$$

Substituting (2-81) into (2-83) yields

$$Y(z) = [C[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B} + D]U(z)$$

The system transfer function is then seen to be

$$G(z) = \mathbf{C}[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B} + D \quad (2-84)$$

This technique will now be illustrated by an example.

### EXAMPLE 2.28

Consider again the state equations of Example 2.24.

$$\mathbf{x}(k + 1) = \begin{bmatrix} 1.35 & 0.55 \\ -0.45 & 0.35 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad -1] \mathbf{x}(k)$$

Now

$$[z\mathbf{I} - \mathbf{A}] = \begin{bmatrix} z - 1.35 & -0.55 \\ 0.45 & z - 0.35 \end{bmatrix}$$

Thus

$$|z\mathbf{I} - \mathbf{A}| = z^2 - 1.7z + 0.72$$

Also,

$$\text{Cof}[z\mathbf{I} - \mathbf{A}] = \begin{bmatrix} z - 0.35 & -0.45 \\ 0.55 & z - 0.35 \end{bmatrix}$$

Then

$$[z\mathbf{I} - \mathbf{A}]^{-1} = \frac{[\text{Cof}[z\mathbf{I} - \mathbf{A}]]^T}{|z\mathbf{I} - \mathbf{A}|} = \frac{1}{z^2 - 1.7z + 0.72} \begin{bmatrix} z - 0.35 & 0.55 \\ -0.45 & z - 1.35 \end{bmatrix}$$

From (2-84), since  $D = 0$ ,

$$\begin{aligned}
 G(z) &= \mathbf{C}[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B} \\
 &= \frac{1}{z^2 - 1.7z + 0.72} [1 \quad -1] \begin{bmatrix} z - 0.35 & 0.55 \\ -0.45 & z - 1.35 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \\
 &= \frac{1}{z^2 - 1.7z + 0.72} [1 \quad -1] \begin{bmatrix} 0.5z + 0.1 \\ 0.5z - 0.9 \end{bmatrix} \\
 &= \frac{1}{z^2 - 1.7z + 0.72}
 \end{aligned}$$

This, of course, has been shown to be the transfer function. We can also perform the computations of (2-84) in MATLAB;

```

>> syms z
A = [1.35 0.55; -0.45 0.35];
B = [0.5; 0.5];
C = [1 -1];
D = 0;
Gz = C*inv(z*eye(2)-A)*B+D;
simplify(Gz)

50/((5*z - 4)*(10*z - 9))

% save this m-file
function [nr2,dr2] = nice(old,z)
% NICE takes the original expression and makes it look nice
% Input: (a*z^2 + b*z + c)/(d*z^2 + e*z + f)
% Output: (a1*z^2 + b1*z +c1)/(z^2 + e1*z + f1)
% old contains symbol of z
% rearrange the expression to collect coefficients
new = collect(old,z);
% obtain the coefficients of numerator and denominator
dr = flipplr(coeffs(feval(symengine,'denom',new)));
nr = flipplr(coeffs(feval(symengine,'numer',new)));
% normalize the coefficients with the coeff of dr z^2 term
dr2 = double(dr/dr(1)); nr2 = double(nr/dr(1));
end

>> [num,den]=nice(Gz,z); T=1;
Gz=tf(num,den,T) % Requires a value for T

Gz =

      1
-----
z^2 - 1.7 z + 0.72
    
```

which simplifies to the same result. MATLAB functions *ss2tf* and *tf* may also be employed as shown below to generate  $G(z)$  in reduced form:

```

>> A = [1.35 0.55; -0.45 0.35];
B = [0.5; 0.5];
C = [1 -1];
D = 0;
T = 1; %Sampling period should be specified
[num,denom]=ss2tf(A,B,C,D);
Gz = tf(num, denom, T)

Gz =

      1
-----
z^2 - 1.7 z + 0.72
    
```

Another property of the similarity transformation, applied to state models as given in (2-70), is seen from the example above. Since the transfer function of a system is invariant under a similarity transformation, then, from (2-70) and (2-84),

$$C[zI - A]^{-1}B + D = C_w[zI - A_w]^{-1}B_w + D_w \quad (2-85)$$

The proof of this property is left as an exercise for the reader (see Problem 2.11-5).

In this section two techniques were presented for obtaining the transfer function from the state equations. For high-order systems, the second method is more attractive.

## 2.12 SOLUTIONS OF THE STATE EQUATIONS

In this section we develop the general solution of linear time-invariant state equations. It will be seen that the key to the solution of the state equations is the calculation of the state transition matrix. Two related techniques, based on the z-transform, for calculating the state transition matrix are presented. Then the solution of state equations via the digital computer is mentioned.

### Recursive Solution

We will first assume that the system is time invariant (fixed) and that  $\mathbf{x}(0)$  and  $\mathbf{u}(j)$ ,  $j = 0, 1, 2, \dots$ , are known. Now the system equations are

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (2-86)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \quad (2-87)$$

In a recursive manner it is obvious that

$$\mathbf{x}(1) = \mathbf{A}\mathbf{x}(0) + \mathbf{B}\mathbf{u}(0)$$

and

$$\mathbf{x}(2) = \mathbf{A}\mathbf{x}(1) + \mathbf{B}\mathbf{u}(1)$$

and hence

$$\begin{aligned} \mathbf{x}(2) &= \mathbf{A}(\mathbf{A}\mathbf{x}(0) + \mathbf{B}\mathbf{u}(0)) + \mathbf{B}\mathbf{u}(1) \\ &= \mathbf{A}^2\mathbf{x}(0) + \mathbf{A}\mathbf{B}\mathbf{u}(0) + \mathbf{B}\mathbf{u}(1) \end{aligned}$$

In a similar manner we can show that

$$\mathbf{x}(3) = \mathbf{A}^3\mathbf{x}(0) + \mathbf{A}^2\mathbf{B}\mathbf{u}(0) + \mathbf{A}\mathbf{B}\mathbf{u}(1) + \mathbf{B}\mathbf{u}(2)$$

It is seen, then, that the general solution is given by

$$\mathbf{x}(k) = \mathbf{A}^k\mathbf{x}(0) + \sum_{j=0}^{k-1} \mathbf{A}^{(k-1-j)}\mathbf{B}\mathbf{u}(j) \quad (2-88)$$

If we define

$$\Phi(k) = \mathbf{A}^k$$

then

$$\mathbf{x}(k) = \Phi(k)\mathbf{x}(0) + \sum_{j=0}^{k-1} \Phi(k - 1 - j)\mathbf{B}\mathbf{u}(j) \quad (2-89)$$

This equation is the general solution to (2-86). From (2-87) and (2-89),

$$\mathbf{y}(k) = \mathbf{C}\Phi(k)\mathbf{x}(0) + \sum_{j=0}^{k-1} \mathbf{C}\Phi(k - 1 - j)\mathbf{B}\mathbf{u}(j) + \mathbf{D}\mathbf{u}(k) \quad (2-90)$$

$\Phi(k)$  is called the *state transition matrix* or the *fundamental matrix*. An example will now be given to illustrate the recursive nature of the solution.

### EXAMPLE 2.29

Consider the transfer function

$$G(z) = \frac{(z + 3)}{(z + 1)(z + 2)}$$

Using the technique of Section 2.9, we write

$$\begin{aligned} \mathbf{x}(k + 1) &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \\ y(k) &= [3 \quad 1] \mathbf{x}(k) \end{aligned}$$

Assume that the system is initially at rest so that  $\mathbf{x}(0) = \mathbf{0}$ ,  $y(0) = \mathbf{C}\mathbf{x}(0) = 0$ , and the input is a unit step; that is,

$$u(k) = 1, \quad k = 0, 1, 2, \dots$$

The recursive solution is obtained as follows:

$$\begin{aligned} \mathbf{x}(1) &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \mathbf{x}(0) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ y(1) &= [3 \quad 1] \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1 \end{aligned}$$

Then

$$\begin{aligned} \mathbf{x}(2) &= \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \mathbf{x}(1) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(1) = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (1) \\ &= \begin{bmatrix} 1 \\ -2 \end{bmatrix} \\ y(2) &= [3 \quad 1] \begin{bmatrix} 1 \\ -2 \end{bmatrix} = 3 - 2 = 1 \end{aligned}$$

In a similar manner it can be shown that

$$\mathbf{x}(3) = \begin{bmatrix} -2 \\ 5 \end{bmatrix}, \quad y(3) = -1$$

and

$$\mathbf{x}(4) = \begin{bmatrix} 5 \\ -10 \end{bmatrix}, \quad y(4) = 5, \text{ etc.}$$

Hence one can recursively determine the states and the output at successive time instants.

A MATLAB program for this example is given by

```
>> A = [0 1; -2 -3];
B = [0; 1];
C = [3 1];
x = [0; 0];
u = 1;
for k = 0:5
    x1 = A*x + B*u;
    k
    y = C*x
    x = x1; %time delay
end

k = 0 1 2 3 4 5
y = 0 1 1 -1 5 -9
```

### **z-Transform Method**

The general solution to the state equations

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (2-91)$$

was developed above, and is given by (2-89)

$$\mathbf{x}(k) = \Phi(k)\mathbf{x}(0) + \sum_{j=0}^{k-1} \Phi(k-1-j)\mathbf{B}\mathbf{u}(j)$$

where  $\Phi(k)$ , the state transition matrix, is given by

$$\Phi(k) = \mathbf{A}^k \quad (2-92)$$

One technique for evaluating  $\Phi(k)$  as a function of  $k$  is through the use of the  $z$ -transform. This technique will now be presented.

In (2-91), let  $\mathbf{u}(k) = \mathbf{0}$ . Then the  $z$ -transform of this equation yields

$$z\mathbf{X}(z) - z\mathbf{x}(0) = \mathbf{A}\mathbf{X}(z) \quad (2-93)$$

Solving for  $\mathbf{X}(z)$ , we see that

$$\mathbf{X}(z) = z[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{x}(0) \quad (2-94)$$

Then

$$\mathbf{x}(k) = \mathcal{Z}^{-1}[\mathbf{X}(z)] = \mathcal{Z}^{-1}[z[z\mathbf{I} - \mathbf{A}]^{-1}]\mathbf{x}(0) \quad (2-95)$$

Comparing (2-95) with (2-89), we see that

$$\Phi(k) = \mathcal{Z}^{-1}[z[z\mathbf{I} - \mathbf{A}]^{-1}] \quad (2-96)$$

To illustrate this method, consider the following example.

### EXAMPLE 2.30

For the state equations of Example 2.29,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$$

Then

$$[z\mathbf{I} - \mathbf{A}] = \begin{bmatrix} z & -1 \\ 2 & z + 3 \end{bmatrix}$$

and

$$|z\mathbf{I} - \mathbf{A}| = z^2 + 3z + 2 = (z + 1)(z + 2)$$

Evaluating the inverse matrix in (2-96) and multiplying by  $z$ , we obtain

$$\begin{aligned} z[z\mathbf{I} - \mathbf{A}]^{-1} &= \begin{bmatrix} \frac{z(z+3)}{(z+1)(z+2)} & \frac{z}{(z+1)(z+2)} \\ \frac{-2z}{(z+1)(z+2)} & \frac{z^2}{(z+1)(z+2)} \end{bmatrix} \\ &= \begin{bmatrix} \frac{2z}{z+1} + \frac{-z}{z+2} & \frac{z}{z+1} + \frac{-z}{z+2} \\ \frac{-2z}{z+1} + \frac{2z}{z+2} & \frac{-z}{z+1} + \frac{2z}{z+2} \end{bmatrix} = \mathcal{Z}[\Phi(k)] \end{aligned}$$

Thus

$$\Phi(k) = \mathcal{Z}^{-1}[z[z\mathbf{I} - \mathbf{A}]^{-1}] = \begin{bmatrix} 2(-1)^k - (-2)^k & (-1)^k - (-2)^k \\ -2(-1)^k + 2(-2)^k & -(-1)^k + 2(-2)^k \end{bmatrix}$$

### Numerical Method via Digital Computer

The digital computer is ideally suited for evaluating equations of the type

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (2-97)$$

These equations can be solved recursively by the computer without actually solving for the state transition matrix.

Another computer method for finding the state transition matrix is to evaluate the expression

$$\Phi(k) = \mathbf{A}^k \quad (2-98)$$

The disadvantage of this procedure is that  $\Phi(k)$  is not found as a general function of  $k$ . However, for high-order systems, the evaluation of  $\Phi(k)$  as a function of  $k$  is difficult using any method.

### Properties of the State Transition Matrix

Three properties of the state transition matrix will now be derived. Since

$$\mathbf{x}(k) = \Phi(k)\mathbf{x}(0) \quad (2-99)$$

then evaluating this expression for  $k = 0$  yields the first property:

$$\Phi(0) = \mathbf{I} \quad (2-100)$$

where  $\mathbf{I}$  is the identity matrix. Next, since

$$\Phi(k) = \mathbf{A}^k$$

then the second property is given by

$$\Phi(k_1 + k_2) = \mathbf{A}^{k_1+k_2} = \mathbf{A}^{k_1}\mathbf{A}^{k_2} = \Phi(k_1)\Phi(k_2) \quad (2-101)$$

The third property is seen from the relationships

$$\Phi(-k) = \mathbf{A}^{-k} = [\mathbf{A}^k]^{-1} = \Phi^{-1}(k) \quad (2-102)$$

or, taking the inverse of this expression, we obtain an equivalent expression

$$\Phi(k) = \Phi^{-1}(-k) \quad (2-103)$$

#### EXAMPLE 2.31

We will illustrate the properties of the state transition matrix using the system described by

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \mathbf{x}(k)$$

Now

$$\Phi(k) = \mathbf{A}^k = \begin{bmatrix} 1^k & 0 \\ 0 & 0.5^k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0.5^k \end{bmatrix}$$

Then, in (2-100),

$$\Phi(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \mathbf{I}$$

In (2-101),

$$\Phi(k_1 + k_2) = \begin{bmatrix} 1 & 0 \\ 0 & 0.5^{k_1+k_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0.5^{k_1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0.5^{k_2} \end{bmatrix} = \Phi(k_1)\Phi(k_2)$$

Also, in (2-103),

$$\Phi^{-1}(-k) = \begin{bmatrix} 1 & 0 \\ 0 & 0.5^{-k} \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 0.5^k \end{bmatrix} = \Phi(k)$$



## 2.13 LINEAR TIME-VARYING SYSTEMS

The state equations for a linear time-varying discrete system were given in (2-47) and (2-48), and are repeated here.

$$\mathbf{x}(k + 1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \quad (2-104)$$

$$\mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k) \quad (2-105)$$

We can find the solution to these equations in a recursive manner. If we denote initial time by  $k_0$  and assume that  $\mathbf{x}(k_0)$  is known and that  $\mathbf{u}(k)$  is known for  $k \geq k_0$ , then

$$\mathbf{x}(1 + k_0) = \mathbf{A}(k_0)\mathbf{x}(k_0) + \mathbf{B}(k_0)\mathbf{u}(k_0)$$

Thus

$$\begin{aligned} \mathbf{x}(2 + k_0) &= \mathbf{A}(1 + k_0)\mathbf{x}(1 + k_0) + \mathbf{B}(1 + k_0)\mathbf{u}(1 + k_0) \\ &= \mathbf{A}(1 + k_0)[\mathbf{A}(k_0)\mathbf{x}(k_0) + \mathbf{B}(k_0)\mathbf{u}(k_0)] + \mathbf{B}(1 + k_0)\mathbf{u}(1 + k_0) \\ &= \mathbf{A}(1 + k_0)\mathbf{A}(k_0)\mathbf{x}(k_0) + \mathbf{A}(1 + k_0)\mathbf{B}(k_0)\mathbf{u}(k_0) + \mathbf{B}(1 + k_0)\mathbf{u}(1 + k_0) \end{aligned}$$

In a like manner, we see that

$$\begin{aligned} \mathbf{x}(3 + k_0) &= \mathbf{A}(2 + k_0)\mathbf{x}(2 + k_0) + \mathbf{B}(2 + k_0)\mathbf{u}(2 + k_0) \\ &= \mathbf{A}(2 + k_0)\mathbf{A}(1 + k_0)\mathbf{A}(k_0)\mathbf{x}(k_0) + \mathbf{A}(2 + k_0)\mathbf{A}(1 + k_0)\mathbf{B}(k_0)\mathbf{u}(k_0) \\ &\quad + \mathbf{A}(2 + k_0)\mathbf{B}(1 + k_0)\mathbf{u}(1 + k_0) + \mathbf{B}(2 + k_0)\mathbf{u}(2 + k_0) \end{aligned}$$

Now, if we define

$$\Phi(k, k_0) = \mathbf{A}(k - 1)\mathbf{A}(k - 2) \cdots \mathbf{A}(k_0) = \begin{cases} \prod_{j=k_0}^{k-1} \mathbf{A}(j), & k > k_0 \\ \mathbf{I}, & k = k_0 \end{cases} \quad (2-106)$$

where  $\mathbf{I}$  is the identity matrix, then the equation above for  $\mathbf{x}(3 + k_0)$  can be written as

$$\mathbf{x}(3 + k_0) = \Phi(3 + k_0, k_0)\mathbf{x}(k_0) + \sum_{j=k_0}^{2+k_0} \Phi(3 + k_0, j + 1)\mathbf{B}(j)\mathbf{u}(j)$$

Thus it can be shown in general that

$$\mathbf{x}(k) = \Phi(k, k_0)\mathbf{x}(k_0) + \sum_{j=k_0}^{k-1} \Phi(k, j + 1)\mathbf{B}(j)\mathbf{u}(j) \quad (2-107)$$

and then

$$\mathbf{y}(k) = \mathbf{C}(k)\Phi(k, k_0)\mathbf{x}(k_0) + \sum_{j=k_0}^{k-1} \mathbf{C}(k)\Phi(k, j + 1)\mathbf{B}(j)\mathbf{u}(j) + \mathbf{D}(k)\mathbf{u}(k) \quad (2-108)$$

As before,  $\Phi(k, k_0)$  is called the state transition matrix. Note that since the matrices are time varying, they must be reevaluated at each time instant.

Using (2-106) and (2-107), we can derive the following important properties of the state transition matrix  $\Phi(k, k_0)$ :

$$\Phi(k_0, k_0) = \mathbf{I}$$

$$\begin{aligned}\Phi(k_2, k_1)\Phi(k_1, k_0) &= \Phi(k_2, k_0) \\ \Phi(k_1, k_2) &= \Phi^{-1}(k_2, k_1)\end{aligned}\tag{2-109}$$

Note that if the system is time invariant, then  $\mathbf{A}$  is not a function of  $k$ . The equation for the state transition matrix, (2-106), then reduces to that for the time-invariant system, (2-92). It is seen that the time-invariant system is a special case of the time-varying system.

## 2.14 SUMMARY

In this chapter we have introduced the concepts of discrete-time systems and the modeling of these systems by difference equations. The  $z$ -transform was defined and was shown to be applicable to the solution of linear time-invariant difference equations. Next, four methods for determining inverse  $z$ -transforms were presented. Finally, the representation of discrete-time systems by simulation diagrams and flow graphs was introduced, which leads naturally to state-variable modeling. Techniques for the solution of linear state equations were then presented. The foundations for the modeling and the analysis of discrete-time systems were presented in this chapter, and will serve for much of the mathematical basis for the following chapters.

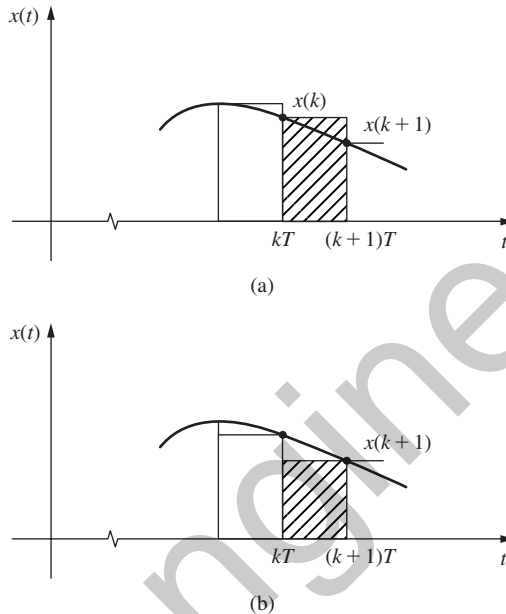
## References and Further Readings

- [1] C. F. Gerald, *Applied Numerical Analysis*, 7th ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 2007.
- [2] "Software Implementation ALS Computer Program," Contract N00421-75-C-0058, Bell Aerospace Corporation, Buffalo, NY, Mar. 1975.
- [3] A. W. Drake, *Fundamentals of Applied Probability Theory*. New York: McGraw-Hill Book Company, 1967.
- [4] A. N. Oppenheim, A. S. Willsky, and N.H. Nawab, *Signals and System*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.
- [5] G. Doetsch, *Guide to the Applications of the Laplace and  $z$ -Transforms*, 2nd ed. New York: Van Nostrand-Reinhold, 1971.
- [6] M. M. Guterman and Z. H. Nitecki, *Differential Equations: A First Course*, 3rd ed. Delhi: Indo American Book Co., 2006.
- [7] C. R. Wylie and L.C. Barrett, *Advanced Engineering Mathematics*, 6th ed. New York: McGraw-Hill Book Company, 1995.
- [8] P. M. De Russo, R. J. Roy, C. M. Close, and A. A. Desrochers, *State Variables for Engineers*, 2nd ed. New York: John Wiley & Sons, Inc., 1998.
- [9] B. Friedland, *Control System Design*. Mineola, NY: Dover Publications, 2005.
- [10] J. A. Cadzow and H. R. Martens, *Discrete-Time and Computer Control Systems*. Upper Saddle River, NJ: Prentice-Hall, 1970.
- [11] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Half Moon Bay, CA: Ellis-Kagle Press, 2006.
- [12] E. I. Jury, *Theory and Application of the  $z$ -Transform Method*. Huntington, NY: R.E. Krieger Publishing Co., Inc., 1973.
- [13] B. C. Kuo, *Digital Control Systems*, 2nd ed. New York: Oxford University Press, 1995.
- [14] M. S. Fadali and A. Visioli, *Digital Control Engineering*. Waltham, MA: Academic Press, 2009.

## Problems

- 2.2-1.** The rectangular rules for numerical integration are illustrated in Fig. P2.2-1. The left-side rule is depicted in Fig. P2.2-1(a), and the right-side rule is depicted in Fig. P2.2-1(b). The integral of  $x(t)$  is approximated by the sum of the rectangular areas shown for each rule. Let  $y(kT)$  be the numerical integral of  $x(t)$ ,  $0 \leq t \leq kT$ .

- Write the difference equation relating  $y(k + 1)$ ,  $y(k)$ , and  $x(k)$  for the left-side rule.
- Find the transfer function  $Y(z)/X(z)$  for part (a).
- Write the difference equation relating  $y(k + 1)$ ,  $y(k)$ , and  $x(k + 1)$  for the right-side rule.
- Find the transfer function  $Y(z)/X(z)$  for part (c).
- Express  $y(k)$  as a summation on  $x(k)$  for the left-side rule.
- Express  $y(k)$  as a summation on  $x(k)$  for the right-side rule.

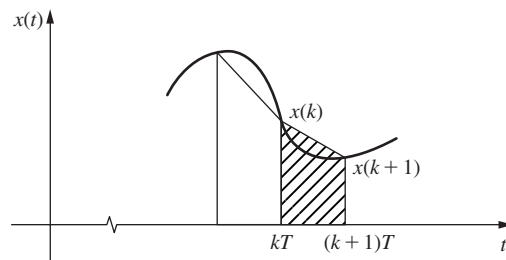


**FIGURE P2.2-1** Rectangular rules for integration: (a) left side; (b) right side.

**2.2-2.** The trapezoidal rule (modified Euler method) for numerical integration approximates the integral of a function  $x(t)$  by summing trapezoid areas as shown in Fig. P2.2-2. Let  $y(t)$  be the integral of  $x(t)$ .

- Write the difference equation relating  $y[(k + 1)T]$ ,  $y(kT)$ ,  $x[(k + 1)T]$ , and  $x(kT)$  for this rule.
- Show that the transfer function for this integrator is given by

$$\frac{Y(z)}{X(z)} = \frac{(T/2)(z + 1)}{z - 1}$$



**FIGURE P2.2-2** Trapezoidal rule for numerical integration.

- The transfer function for the right-side rectangular-rule integrator was found in Problem 2.2-1 to be  $Y(z)/X(z) = Tz/(z - 1)$ . We would suspect that the reciprocal of this transfer function should yield an approximation to a differentiator. That is, if  $w(kT)$  is a numerical derivative of  $x(t)$  at  $t = kT$ ,

$$\frac{W(z)}{X(z)} = \frac{z - 1}{Tz}$$

Write the difference equation describing this differentiator.

(b) Draw a figure similar to those in Fig. P2.2-1 illustrating the approximate differentiation.

(c) Repeat part (a) for the left-side rule, where  $W(z)/X(z) = T/(z - 1)$ .

(d) Repeat part (b) for the differentiator of part (c).

**2.3-1.** Find the  $z$ -transform of the number sequence generated by sampling the time function  $e(t) = 2t$  every  $T$  seconds, beginning at  $t = 0$ . Can you express this transform in closed form?

**2.3-2.** (a) Write, as a series, the  $z$ -transform of the number sequence generated by sampling the time function  $e(t) = e^{-2t}$  every  $T$  seconds, beginning at  $t = 0$ . Can you express this transform in closed form?

(b) Evaluate the coefficients in the series of part (a) for the case that  $T = 0.1$  s.

(c) The exponential  $e(t) = e^{-bt}$  is sampled every  $T = 0.2$  s, yielding the  $z$ -transform

$$E(z) = 1 + \left(\frac{1}{2}\right)z^{-1} + \left(\frac{1}{2}\right)^2 z^{-2} + \left(\frac{1}{2}\right)^3 z^{-3} + \dots$$

Evaluate  $b$ .

**2.3-3.** Find the  $z$ -transforms of the number sequences generated by sampling the following time functions every  $T$  seconds, beginning at  $t = 0$ . Express these transforms in closed form.

(a)  $e(t) = e^{-at}$

(b)  $e(t) = e^{-(t-T)}u(t - T)$

(c)  $e(t) = e^{-(t-5T)}u(t - 5T)$

**2.4-1.** A function  $e(t)$  is sampled, and the resultant sequence has the  $z$ -transform

$$E(z) = \frac{z^3 - 4z}{z^4 - 0.9z^2 + 0.8}$$

Solve this problem using  $E(z)$  and the properties of the  $z$ -transform.

(a) Find the  $z$ -transform of  $e(t - 2T)u(t - 2T)$ .

(b) Find the  $z$ -transform of  $e(t + 2T)u(t)$ .

(c) Find the  $z$ -transform of  $e(t - 3T)u(t - 4T)$ .

**2.4-2.** A function  $e(t)$  is sampled, and the resultant sequence has the  $z$ -transform

$$E(z) = \frac{z - b}{z^2 - cz^2 + d}$$

Find the  $z$ -transform of  $\epsilon^{akT}e(kT)$ . Solve this problem using  $E(z)$  and the properties of the  $z$ -transform.

**2.5-1.** From Table 2-3,

$$\mathcal{Z}[\cos akT] = \frac{z(z - \cos aT)}{z^2 - 2z \cos aT + 1}$$

(a) Find the conditions on the parameter  $a$  such that  $\mathcal{Z}[\cos akT]$  is first order (pole-zero cancellation occurs).

(b) Give the first-order transfer function in part (a).

(c) Find  $a$  such that  $\mathcal{Z}[\cos akT] = \mathcal{Z}[u(kT)]$ , where  $u(kT)$  is the unit step function.

**2.5-2.** Find the  $z$ -transform, in closed form, of the number sequence generated by sampling the time function  $e(t)$  every  $T$  seconds beginning at  $t = 0$ . The function  $e(t)$  is specified by its Laplace transform,

$$E(s) = \frac{2(1 - \epsilon^{-6s})}{s(s + 4)}, \quad T = 1 \text{ s}$$

**2.5-3.** Use the residue method to find the  $z$ -transform of the following functions. Check your results by computer.

(a)  $E(s) = \frac{20}{(s + 2)(s + 5)}$

(b)  $E(s) = \frac{5}{s(s + 1)}$

(c)  $E(s) = \frac{s + 2}{s(s + 1)}$

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

(e)  $E(s) = \frac{s^2 + 5s + 6}{s(s + 4)(s + 5)}$

(f)  $E(s) = \frac{2}{s^2 + 2s + 5}$

**2.6-1.** Solve the given difference equation for  $x(k)$  using:

- The sequential technique.
- The  $z$ -transform.
- Will the final-value theorem give the correct value of  $x(k)$  as  $k \rightarrow \infty$ ?

$$x(k) - 5x(k-1) + 6x(k-2) = e(k)$$

where

$$e(k) = \begin{cases} 1, & k = 0, 1 \\ 0, & k \geq 2 \end{cases}$$

$$x(-1) = x(-2) = 0$$

**2.6-2.** Given the difference equation

$$y(k+2) - \frac{3}{4}y(k+1) + \frac{1}{8}y(k) = e(k)$$

where  $y(0) = y(1) = 0$ ,  $e(0) = 0$ , and  $e(k) = 1$ ,  $k = 1, 2, \dots$

- Solve for  $y(k)$  as a function of  $k$ , and give the numerical values of  $y(k)$ ,  $0 \leq k \leq 4$ .
  - Solve the difference equation directly for  $y(k)$ ,  $0 \leq k \leq 4$ , to verify the results of part (a).
  - Repeat parts (a) and (b) for  $e(k) = 0$  for all  $k$ , and  $y(0) = 1$ ,  $y(1) = -2$ .
- 2.6-3.** Given the difference equation

$$x(k) - x(k-1) + x(k-2) = e(k)$$

where  $e(k) = 1$  for  $k \geq 0$ .

- Solve for  $x(k)$  as a function of  $k$ , using the  $z$ -transform. Give the values of  $x(0)$ ,  $x(1)$ , and  $x(2)$ .
  - Verify the values  $x(0)$ ,  $x(1)$ , and  $x(2)$ , using the power-series method.
  - Verify the values  $x(0)$ ,  $x(1)$ , and  $x(2)$  by solving the difference equation directly.
  - Will the final-value property give the correct value for  $x(\infty)$ ?
- 2.6-4.** Given the difference equation

$$x(k+2) + 5x(k+1) + 4x(k) = e(k)$$

where

$$e(k) = \begin{cases} 1, & k = 0 \\ 0, & \text{otherwise} \end{cases}$$

$$x(0) = 0$$

$$x(1) = -1$$

- Solve for  $x(k)$  as a function of  $k$ .
  - Evaluate  $x(0)$ ,  $x(1)$ ,  $x(2)$ , and  $x(3)$  in part (a).
  - Verify the results in part (b) using the power-series method.
  - Verify the results in part (b) by solving the difference equation directly.
- 2.6-5.** Given the difference equation

$$x(k+3) - 2.2x(k+2) + 1.57x(k+1) - 0.36x(k) = e(k)$$

where  $e(k) = 1$  for all  $k \geq 0$ , and  $x(0) = x(1) = x(2) = 0$ .

- Write a digital computer program that will calculate  $x(k)$ . Run this program solving for  $x(3)$ ,  $x(4)$ ,  $\dots$ ,  $x(25)$ .
  - Using the sequential technique, check the values of  $x(k)$ ,  $0 \leq k \leq 5$ .
  - Use the  $z$ -transform and the power-series method to verify the values  $x(k)$ ,  $0 \leq k \leq 5$ .
- 2.6-6.** Given the MATLAB program

```

>>s1 = 0;
e = 0;
for k = 0:5

```

```

s2 = e - s1;
m = 0.5*s2 - s1;
s1 = s2;
[k, m]
e = e + 1;
    
```

end

that solves the difference equation of a digital controller.

- Find the transfer function of the controller.
- Find the  $z$ -transform of the controller input.
- Use the results of parts (a) and (b) to find the inverse  $z$ -transform of the controller output.
- Run the program to check the results of part (c).

**2.7-1.** (a) Find  $e(0)$ ,  $e(1)$ , and  $e(10)$  for

$$E(z) = \frac{0.8}{z(z - 0.6)}$$

using the inversion formula.

- Check the value of  $e(0)$  using the initial-value property.
- Check the values calculated in part (a) using partial fractions.
- Find  $e(k)$  for  $k = 0, 1, 2, 3$ , and 4 if  $\mathcal{Z}[e(k)]$  is given by

$$E(z) = \frac{1.98z}{(z^2 - 0.9z + 0.9)(z - 0.8)(z^2 - 1.2z + 0.27)}$$

- Find a function  $e(t)$  which, when sampled at a rate of 10 Hz ( $T = 0.1$ s), results in the transform  $E(z) = 2z/(z - 0.6)$ .
- Repeat part (e) for  $E(z) = 2z/(z + 0.6)$ .
- From parts (e) and (f), what is the effect on the inverse  $z$ -transform of changing the sign on a real pole?

**2.7-2.** For the number sequence  $\{e(k)\}$ ,

$$E(z) = \frac{z}{(z-1)^2}$$

- Apply the final-value theorem to  $E(z)$ .
- Check your result in part (a) by finding the inverse  $z$ -transform of  $E(z)$ .
- Repeat parts (a) and (b) with  $E(z) = z/(z + 1)^2$ .
- Repeat parts (a) and (b) with  $E(z) = z/(z + 0.9)^2$ .
- Repeat parts (a) and (b) with  $E(z) = z/(z + 0.1)^2$ .

**2.7-3.** Find the inverse  $z$ -transform of each  $E(z)$  below by the four methods given in the text. Compare the values of  $e(z)$ , for  $k = 0, 1, 2$ , and 3, obtained by the four methods.

(a)  $E(z) = \frac{z}{(z - 1)(z - 0.8)}$

(b)  $E(z) = \frac{0.5}{(z - 1)(z - 0.8)}$

(c)  $E(z) = \frac{0.5(z + 1)}{(z - 1)(z - 0.8)}$

(d)  $E(z) = \frac{z(z - 1)}{(z - 1)(z - 0.8)}$

- Use MATLAB to verify the partial-fraction expansions.

**2.7-4.** Find the inverse  $z$ -transform of each  $E(z)$  below by any method.

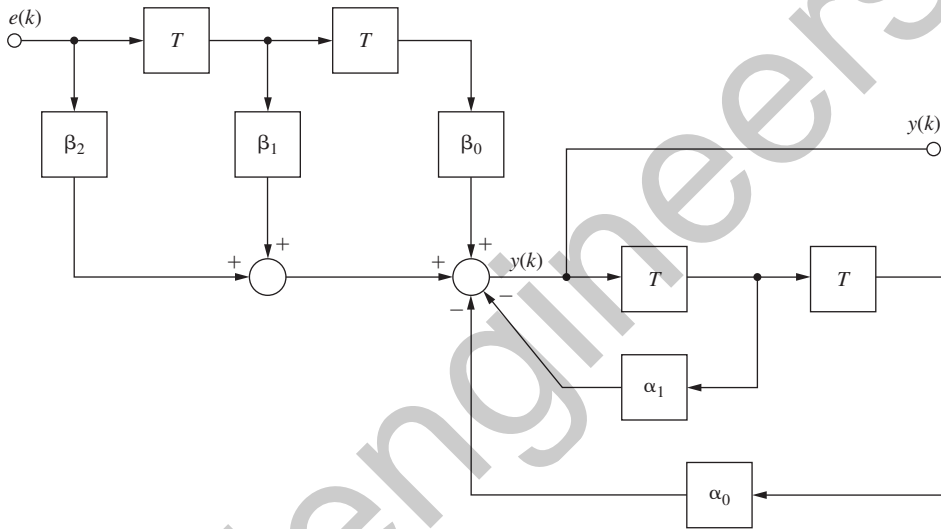
(a)  $E(z) = \frac{0.5z^2}{(z - 1)^2(z - 0.6)}$

(b)  $E(z) = \frac{0.5}{(z - 1)(z - 0.6)^2}$

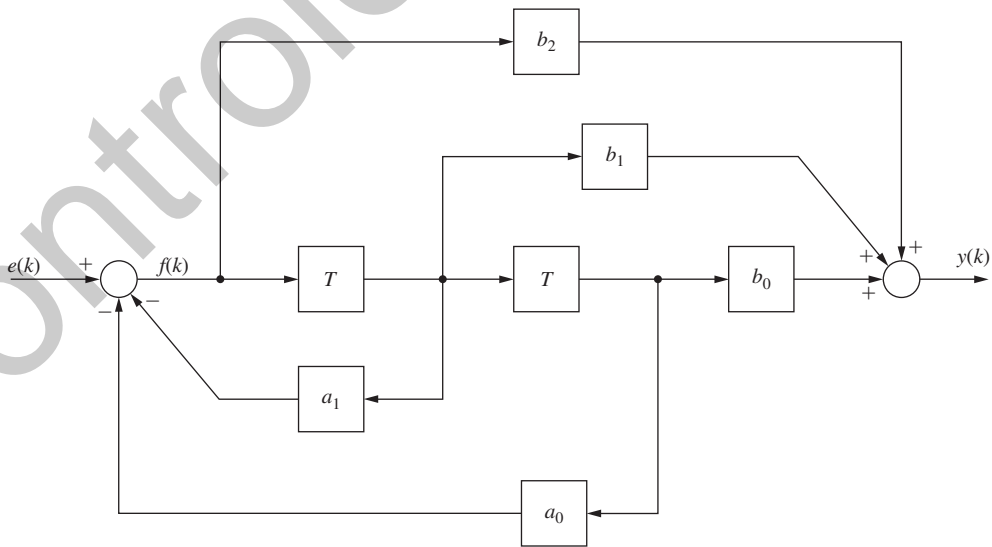
(c)  $E(z) = \frac{0.5z(z - 0.7)}{(z - 1)(z - 0.6)^2}$

(d)  $E(z) = \frac{0.5z(z - 0.7)}{(z - 1)(z - 0.59)(z - 0.61)}$

- 2.8-1.** Given in Fig. P2.8-1 are two digital-filter structures, or realizations, for second-order filters.
- Write the difference equation for the 3D structure of Fig. P2.8-1(a), expressing  $y(k)$  as a function of  $y(k - i)$  and  $e(k - i)$ .
  - Derive the filter transfer function  $Y(z)/E(z)$  for the 3D structure by taking the  $z$ -transform of the equation in part (a).
  - Write the difference equation for the 1D structure of Fig. P2.8-1(b). Two equations are required, with one for  $f(k)$  and one for  $y(k)$ .
  - Derive the filter transfer function  $Y(z)/E(z)$  for the 1D structure by taking the  $z$ -transform of the equations in part (c) and eliminating  $F(z)$ .



(a)



(b)

**FIGURE P2.8-1** Digital-filter structures: (a) 3D; (b) 1D.

- (e) From parts (b) and (d), relate the coefficients  $\alpha_i, \beta_i$  to  $a_i, b_i$  such that the two filters realize the same transfer function.
- (f) Write a computer-program segment that realizes the 3D structure. This program should be of the form used in Example 2.10.
- (g) Write a computer-program segment that realizes the 1D structure. This program should be of the form used in Example 2.10.

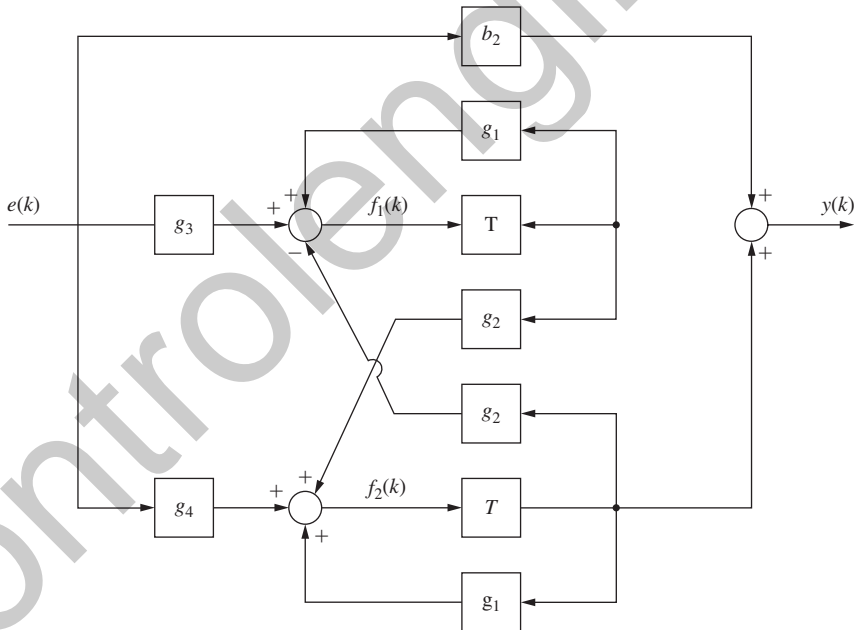
**2.8-2.** Shown in Fig. P2.8-2 is the second-order digital-filter structure 1X. This structure realizes the filter transfer function

$$D(z) = b_2 + \frac{A}{z - p} + \frac{A^*}{z - p^*}$$

where  $p$  and  $p^*$  (conjugate of  $p$ ) are complex. The relationships between the filter coefficients and the coefficients in Fig. P2.8-2 are given by

$$\begin{aligned} g_1 &= \operatorname{Re}(p) & g_3 &= -2 \operatorname{Im}(A) \\ g_2 &= \operatorname{Im}(p) & g_4 &= 2 \operatorname{Re}(A) \end{aligned}$$

- (a) To realize this filter, difference equations are required for  $f_1(k), f_2(k)$ , and  $y(k)$ . Write these equations.
- (b) Find the filter transfer function  $Y(z)/E(z)$  by taking the  $z$ -transform of the equations of part (a) and eliminating  $F_1(z)$  and  $F_2(z)$ .
- (c) Verify the results in part (b) using Mason's gain formula.
- (d) Write a computer-program segment that realizes the 1X structure. This program should be of the form of that is used in Example 2.10.



**FIGURE P2.8-2** Digital-filter structure 1X.

**2.8-3.** Given the second-order digital-filter transfer function

$$D(z) = \frac{2z^2 - 2.4z + 0.72}{z^2 - 1.4z + 0.98}$$

- (a) Find the coefficients of the 3D structure of Fig. P2.8-1 such that  $D(z)$  is realized.
- (b) Find the coefficients of the 1D structure of Fig. P2.8-1 such that  $D(z)$  is realized.



- (c) Find the coefficients of the IX structure of Fig. P2.8-2 such that  $D(z)$  is realized. The coefficients are identified in Problem 2.8-2.
- (d) Use MATLAB to verify the partial-fraction expansions in part (c).
- (e) Verify the results in part (c) using Mason's gain formula.
- 2.9-1.** Find two different state-variable formulations that model the system whose difference equation is given by:
- (a)  $y(k + 2) + 6y(k + 1) + 5y(k) = 2e(k)$
- (b)  $y(k + 2) + 6y(k + 1) + 5y(k) = e(k + 1) + 2e(k)$
- (c)  $y(k + 2) + 6y(k + 1) + 5y(k) = 3e(k + 2) + e(k + 1) + 2e(k)$
- 2.9-2.** Write the state equations for the observer canonical form of a system, shown in Fig. 2-10, which has the transfer function given in (2-51) and (2-61)

$$G(z) = \frac{b_{n-1}z^{n-1} + \cdots + b_1z + b_0}{z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0}$$

- 2.9-3.** Note the relationship between the control canonical form in Fig. 2-9 and the observer canonical form in Fig. 2-10. The diagram of Fig. 2-9(b) can be converted to that of Fig. 9-10 by changing all signal-summing points to signal-distribution points, all signal-distribution points to signal-summing points, and reversing the signal flow direction in each signal path. The coefficient gains remain the same and signals are reversed in the time delay elements. These "transverse" diagrams then have the same transfer function.
- (a) Write the state equations for the simulation diagram shown in Fig. 2-6.
- (b) Find the "transverse" simulation diagram for this structure.
- (c) Write the state equations for this transverse structure.
- (d) Show that the transfer function for this transverse structure is given by (2-43).
- 2.10-1.** Find a state-variable formulation for the system described by the coupled second-order difference equations given. The system output is  $y(k)$ , and  $e_1(k)$  and  $e_2(k)$  are the system inputs. *Hint:* Draw a simulation diagram first.

$$x(k + 2) + v(k + 1) = 4e_1(k) + e_2(k)$$

$$v(k + 2) - v(k) + x(k) = 2e_1(k)$$

$$y(k) = v(k + 2) - x(k + 1) + e_1(k)$$

- 2.10-2.** Consider the system described by

$$\mathbf{x}(k + 1) = \begin{bmatrix} 0 & 1 \\ 0 & 3 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [-2 \quad 1] \mathbf{x}(k)$$

- (a) Find the transfer function  $Y(z)/U(z)$ .
- (b) Using any similarity transformation, find a different state model for this system.
- (c) Find the transfer function of the system from the transformed state equations.
- (d) Verify that  $\mathbf{A}$  given and  $\mathbf{A}_w$  derived in part (b) satisfy the first three properties of similarity transformations. The fourth property was verified in part (c).
- 2.10-3.** Consider the system of Problem 2.10-2. A similarity transformation on these equations yields

$$\mathbf{w}(k + 1) = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} \mathbf{w}(k) + \mathbf{B}_w u(k)$$

$$y(k) = \mathbf{C}_w \mathbf{x}(k)$$

- (a) Find  $d_1$  and  $d_2$ .
- (b) Find a similarity transformation that results in the  $\mathbf{A}_w$  matrix given. Note that this matrix is diagonal.
- (c) Find  $\mathbf{B}_w$  and  $\mathbf{C}_w$ .
- (d) Find the transfer functions of both sets of state equations to verify the results of this problem.

**2.10-4.** Repeat Problem 2.10-2 for the system described by

$$\mathbf{x}(k + 1) = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 2] \mathbf{x}(k)$$

**2.11-1.** Consider a system with the transfer function

$$G(z) = \frac{Y(z)}{U(z)} = \frac{2}{z(z - 1)}$$

- Find three different state-variable models of this system.
- Verify the transfer function of each state model in part (a), using (2-84).

**2.11-2.** Consider a system described by the coupled difference equation

$$y(k + 2) - v(k) = 0$$

$$v(k + 1) + y(k + 1) = u(k)$$

where  $u(k)$  is the system input.

- Find a state-variable formulation for this system. Consider the outputs to be  $y(k + 1)$  and  $v(k)$ . *Hint:* Draw a simulation diagram first.
- Repeat part (a) with  $y(k)$  and  $v(k)$  as the outputs.
- Repeat part (a) with the single output  $v(k)$ .
- Use (2-84) to calculate the system transfer function with  $v(k)$  as the system output, as in part (c); that is, find  $V(z)/U(z)$ .
- Verify the transfer function  $V(z)/U(z)$  in part (d) by taking the  $z$ -transform of the given system difference equations and eliminating  $Y(z)$ .
- Verify the transfer function  $V(z)/U(z)$  in part (d) by using Mason's gain formula on the simulation diagram of part (a).

**2.11-3.** Given the system described by the state equations

$$\mathbf{x}(k + 1) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(k)$$

$$y(k) = [0 \quad 0 \quad 1] \mathbf{x}(k)$$

- Calculate the transfer function  $Y(z)/U(z)$ , using (2-84).
- Draw a simulation diagram for this system, from the state equations given.
- Use Mason's gain formula and the simulation diagram to verify the transfer function found in part (a).

**2.11-4.** Section 2.9 gives some standard forms for state equations (simulation diagrams for the control canonical and observer canonical forms). The MATLAB statement

$$[A,B,C,D] = \text{tf2ss}(\text{num},\text{den})$$

generates a standard set of state equations for the transfer function whose numerator coefficients are given in the vector *num* and denominator coefficients in the vector *den*.

- Use the MATLAB statement given to generate a set of state equations for the transfer function

$$G(z) = \frac{3z + 4}{z^2 + 5z + 6}$$

- Draw a simulation diagram for the state equations in part (a).
- Determine if the simulation diagram in part (b) is one of the standard forms in Section 2.9.

**2.11-5.** Show that for the similarity transformation of (2-71),

$$\mathbf{C}[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B} + \mathbf{D} = \mathbf{C}_w[z\mathbf{I} - \mathbf{A}_w]^{-1}\mathbf{B}_w + \mathbf{D}_w$$

**2.12-1.** Consider the system described in Problem 2.10-2.

- Find the transfer function of this system.
- Let  $u(k) = 1, k \geq 0$  (a unit step function) and  $\mathbf{x}(0) = 0$ . Use the transfer function of part (a) to find the system response.
- Find the state transition matrix  $\Phi(k)$  for this system.
- Use (2-90) to verify the step response calculated in part (b). This calculation results in the response expressed as a summation. Then check the values  $y(0), y(1),$  and  $y(2)$ .
- Verify the results of part (d) by the iterative solution of the state equations.

**2.12-2.** The system described by the equations

$$\mathbf{x}(k + 1) = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 2] \mathbf{x}(k)$$

is excited by the initial conditions  $\mathbf{x}(0) = [-1 \quad 2]^T$  with  $u(k) = 0$  for all  $k$ .

- Use (2-89) to solve for  $\mathbf{x}(k), k \geq 0$ .
- Find the output  $y(k)$ .
- Show that  $\Phi(k)$  in (a) satisfies the property  $\Phi(0) = \mathbf{I}$ .
- Show that the solution in part (a) satisfies the given initial conditions.
- Use an iterative solution of the state equations to show that the values  $y(k)$ , for  $k = 0, 1, 2,$  and  $3$ , in part (b) are correct.
- Verify the results in part (e) using MATLAB.

**2.12-3.** The system described by the equations

$$\mathbf{x}(k + 1) = \begin{bmatrix} 1.1 & 1 \\ -0.3 & 0 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad -1] \mathbf{x}(k)$$

is excited by the initial conditions  $\mathbf{x}(0) = [-1 \quad 2]^T$  with  $u(k) = 0$  for all  $k$ .

- Use (2-89) to solve for  $\mathbf{x}(k), k \geq 0$ .
- Find the output  $y(k)$ .
- Show that  $\Phi(k)$  in part (a) satisfies the property  $\Phi(0) = \mathbf{I}$ .
- Show that the solution in part (a) satisfies the given initial conditions.
- Use an iterative solution of the state equations to show that the values  $y(k)$ , for  $k = 0, 1, 2,$  and  $3$ , in part (b) are correct.

**2.12-4.** Let  $\Phi(k)$  be the state transition matrix for the equations

$$\mathbf{x}(k + 1) = \mathbf{A} \mathbf{x}(k)$$

Show that  $\Phi(k)$  satisfies the difference equation

$$\Phi(k + 1) = \mathbf{A} \Phi(k)$$

# 3

## Sampling and Reconstruction

### 3.1 INTRODUCTION

In Chapter 2 the concept of a discrete-time system was developed. We found that a discrete-time system is described (modeled) by a difference equation and that signals within the system are described by number sequences (e.g.,  $\{e(k)\}$ ). Some of these number sequences may be generated by sampling a continuous-time signal (e.g., in digital control systems). To provide a basis for thoroughly understanding the operation of digital control systems, it is necessary to determine the effects of sampling a continuous-time signal. These topics are investigated in this chapter.

### 3.2 SAMPLED-DATA CONTROL SYSTEMS

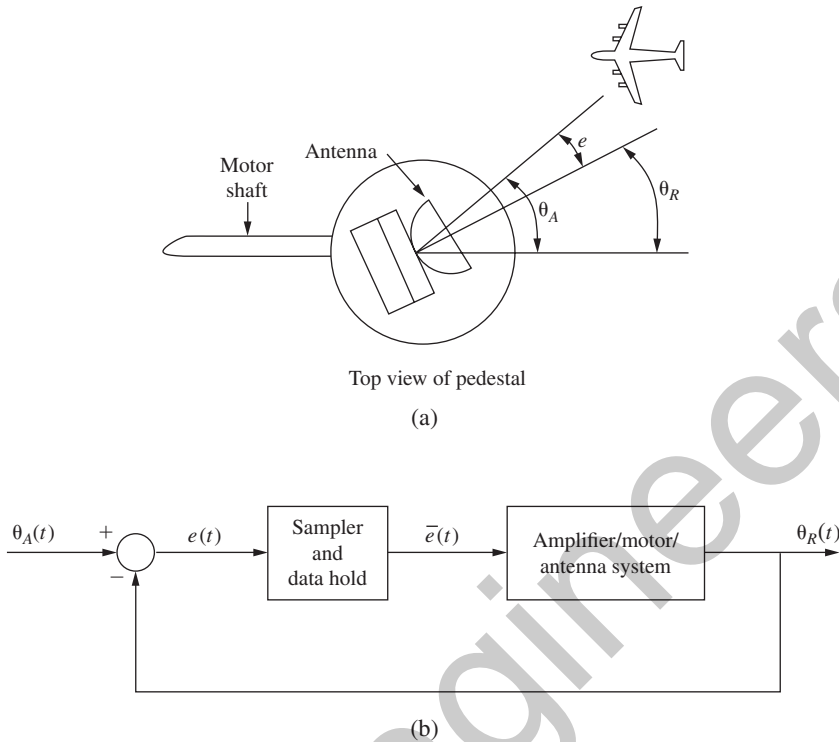
In this section, the type of sampling that generally occurs in sampled-data control systems and in digital control systems is introduced, and a mathematical model of this sampling is developed. From this model we may determine the effects of the sampling on the information content of the signal that is sampled.

To introduce sampled-data systems, we consider the radar tracking system of Fig. 3-1(a). This system is described in Section 1.5. We consider only the control of the yaw angle  $\theta_R(t)$ , shown in the top view of the pedestal. The closed-loop system is to track the aircraft shown automatically. In Fig. 3-1(a),  $\theta_R(t)$  is the yaw-axis pointing angle of the antenna and  $\theta_A(t)$  is the angle to the aircraft. Hence the tracking error is  $e(t)$ , given by

$$e(t) = \theta_A(t) - \theta_R(t)$$

Assume that the radar transmits every  $T$  seconds. Then the error  $e(t)$  is known only every  $T$  seconds. The block diagram of this system is shown in Fig. 3-1(b). The radar receiver must output a voltage at every instant of time to the power amplifier. Since only  $e(kT)$  is known, a decision must be made as to the form of power amplifier input,  $\bar{e}(t)$ , for  $t \neq kT$ .

In general, it is undesirable to apply a signal in sampled form, such as a train of narrow rectangular pulses, as shown in Fig. 3-2, to a plant, because of the high-frequency components

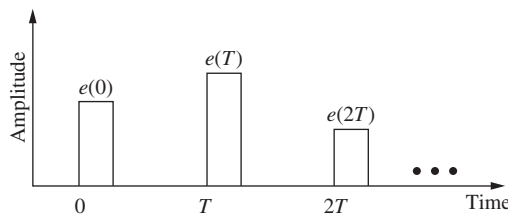


**FIGURE 3-1** Sampled-data control system.

inherently present in that signal. Therefore, a data-reconstruction device, called a data hold, is inserted into the system directly following the sampler. The purpose of the data hold is to reconstruct the sampled signal into a form that closely resembles the signal before sampling. The simplest data-reconstruction device, and by far the most common one, is the zero-order hold. The operation of a sampler/zero-order hold combination is described by the signals shown in Fig. 3-3. The zero-order hold clamps the output signal to a value equal to that of the input signal at the sampling instant.

The sampler and zero-order hold can be represented in block-diagram form as shown in Fig. 3-4. The signal  $\bar{e}(t)$  can be expressed as

$$\begin{aligned}
 \bar{e}(t) = & e(0)[u(t) - u(t - T)] + e(T)[u(t - T) - u(t - 2T)] \\
 & + e(2T)[u(t - 2T) - u(t - 3T)] + \dots
 \end{aligned}
 \tag{3-1}$$



**FIGURE 3-2** Sampled signal in pulse form.

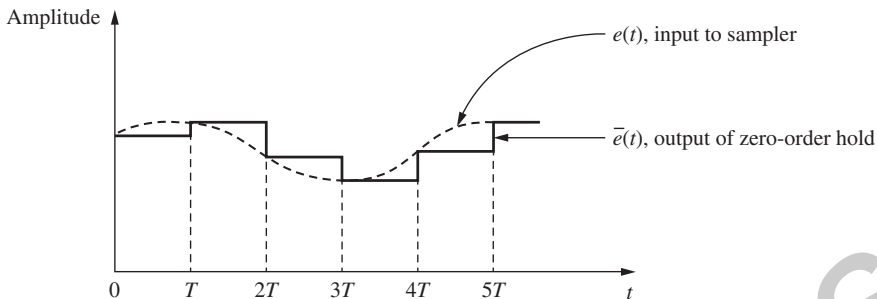


FIGURE 3-3 Input and output signals of sampler/data hold.

where  $u(t)$  is the unit-step function. The Laplace transform of  $\bar{e}(t)$  is  $\bar{E}(s)$ , given by

$$\begin{aligned} \bar{E}(s) &= e(0) \left[ \frac{1}{s} - \frac{\epsilon^{-Ts}}{s} \right] + e(T) \left[ \frac{\epsilon^{-Ts}}{s} - \frac{\epsilon^{-2Ts}}{s} \right] \\ &+ e(2T) \left[ \frac{\epsilon^{-2Ts}}{s} - \frac{\epsilon^{-3Ts}}{s} \right] + \dots \\ &= \left[ \frac{1 - \epsilon^{-Ts}}{s} \right] [e(0) + e(T)\epsilon^{-Ts} + e(2T)\epsilon^{-2Ts} + \dots] \\ &= \left[ \sum_{n=0}^{\infty} e(nT)\epsilon^{-nTs} \right] \left[ \frac{1 - \epsilon^{-Ts}}{s} \right] \end{aligned} \quad (3-2)$$

(The Laplace transform is reviewed in Appendix V, and Appendix VI gives a table of Laplace transforms.)

The first factor in the last expression in (3-2) is seen to be a function of the input signal  $e(t)$  and the sampling period  $T$ . The second factor is seen to be independent of  $e(t)$  and therefore can be considered to be a transfer function. Thus the sampler/hold operation can be represented as shown in Fig. 3-4. The function  $E^*(s)$ , called the starred transform, is defined as

$$E^*(s) = \sum_{n=0}^{\infty} e(nT)\epsilon^{-nTs} \quad (3-3)$$

Hence (3-2) is satisfied by the representation in Fig. 3-5. The operation denoted by the switch in Fig. 3-5 is defined by (3-3) and is called an ideal sampler; the operation denoted by the transfer function is called the data hold. It is to be emphasized that  $E^*(s)$  does not appear in the physical system but appears as a result of factoring (3-2). The sampler (switch) in Fig. 3-5 does not model a physical sampler, and the block does not model a physical data hold. However, the combination does accurately model the input–output characteristics of the sampler-data hold device, as demonstrated earlier.

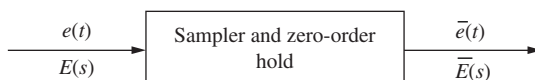
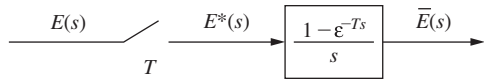


FIGURE 3-4 Sampler and data hold.



**FIGURE 3-5** Representation of sampler and data hold.

The operation symbolized by the sampler in Fig. 3-5 cannot be represented by a transfer function. From (3-3) we see that the output of the sampler is a function of  $e(t)$  only at  $t = kT$ ,  $k = 0, 1, 2, \dots$ . Hence many different input signals can result in the same output signal  $E^*(s)$ . However, the representation of a sampler as a transfer function would require each different  $E(s)$  to result in a different  $E^*(s)$ . Hence no transfer function exists for the sampler, and this property of the sampler complicates the analysis of systems of the type shown in Fig. 3-1(b).

As an aside, many of the systems that we consider will have unity gain in the feedback path, as shown in Fig. 3-1(b). In practical systems, the sensor transfer function will appear in the feedback path. However, the bandwidth of the sensor is usually much greater than that of the plant, and thus the sensor can be considered to be a constant gain for system analysis and design. For a discussion of the problems in converting practical control systems to systems with unity gain in the feedback path, see [1].

### 3.3 THE IDEAL SAMPLER

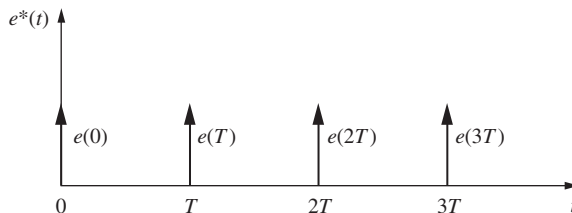
The inverse Laplace transform of  $E^*(s)$  is, from (3-3),

$$\begin{aligned}
 e^*(t) = \mathcal{L}^{-1}[E^*(s)] &= e(0)\delta(t) + e(T)\delta(t - T) \\
 &+ e(2T)\delta(t - 2T) + \dots
 \end{aligned}
 \tag{3-4}$$

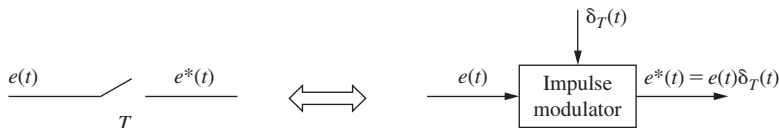
where  $\delta(t)$  is the unit impulse function occurring at  $t = 0$ . Then  $e^*(t)$  is a train of impulse functions whose weights are equal to the values of the signal at the instants of sampling. Thus  $e^*(t)$  can be represented as shown in Fig. 3-6, since the impulse function has infinite amplitude at the instant it occurs. Note again that  $e^*(t)$  is not a physical signal.

The sampler that appears in a sampler/hold model is usually referred to as an *ideal sampler*, since nonphysical signals (impulse functions) appear on its output. This sampler is also referred to as an *impulse modulator*. To demonstrate this modulation concept, we define

$$\delta_T(t) = \sum_{n=0}^{\infty} \delta(t - nT) = \delta(t) + \delta(t - T) + \dots
 \tag{3-5}$$



**FIGURE 3-6** Representation of  $e^*(t)$ .



**FIGURE 3-7** Representations of the ideal sampler.

Then  $e^*(t)$  can be expressed as

$$\begin{aligned} e^*(t) &= e(t)\delta_T(t) = e(t)\delta(t) + e(t)\delta(t - T) + \dots \\ &= e(0)\delta(t) + e(T)\delta(t - T) + \dots \end{aligned} \quad (3-6)$$

In this form it can be readily seen that  $\delta_T(t)$  is the carrier in the modulation process, and  $e(t)$  is the modulating signal [2]. The Laplace transform requires  $e(t)$  to be zero for  $t < 0$  [3]. For this reason the summation in (3-5) can be taken from  $n = -\infty$  to  $n = \infty$  with no change in (3-6). Two equivalent representations of the ideal sampler are given in Fig. 3-7.

A problem arises in the definition of the ideal sampler output in (3-4) if  $e(t)$  has a discontinuity at  $t = kT$ . For example, if  $e(t)$  is a unit-step function, what value is used for  $e(0)$  in (3-4)? In order to be consistent in the consideration of discontinuous signals, the output signal of an ideal sampler is *defined* as follows:

**DEFINITION.** The output signal of an ideal sampler is defined as the signal whose Laplace transform is

$$E^*(s) = \sum_{n=0}^{\infty} e(nT)\epsilon^{-nTs} \quad (3-7)$$

where  $e(t)$  is the input signal to the sampler. If  $e(t)$  is discontinuous at  $t = kT$ , where  $k$  is an integer, then  $e(kT)$  is taken to be  $e(kT^+)$ . The notation  $e(kT^+)$  indicates the value of  $e(t)$  as  $t$  approaches  $kT$  from the right (i.e., at  $t = kT + \Delta$ , where  $\Delta$  is made arbitrarily small).

It is important to note that the definition of the sampling operation as specified in (3-7) together with the zero-order-hold transfer function defined by

$$G_{ho}(s) = \frac{1 - \epsilon^{-Ts}}{s} \quad (3-8)$$

yield the correct mathematical description of the sampler/hold operation defined by (3-2). It should also be noted that if the signal to be sampled contains an impulse function at a sampling instant, the Laplace transform of the sampled signal does not exist; but since continuous signals in practical situations never contain impulse functions, this limitation is of no practical concern.

### EXAMPLE 3.1

Determine  $E^*(s)$  for  $e(t) = u(t)$ , the unit step. For the unit step,  $e(nT) = 1$ ,  $n = 0, 1, 2, \dots$ . Thus from (3-7),

$$E^*(s) = \sum_{n=0}^{\infty} e(nT)\epsilon^{-nTs} = e(0) + e(nT)\epsilon^{-Ts} + e(nT)\epsilon^{-2Ts} + \dots$$



or

$$E^*(s) = 1 + \epsilon^{-Ts} + \epsilon^{-2Ts} + \dots$$

$E^*(s)$  can be expressed in closed form using the following relationship. For  $|x| < 1$ ,

$$\frac{1}{1-x} = 1 + x + x^2 + \dots$$

The condition  $|x| < 1$  guarantees convergence of the series. Hence the expression for  $E^*(s)$  above can be written in closed form as

$$E^*(s) = \frac{1}{1 - \epsilon^{-Ts}}, \quad |\epsilon^{-Ts}| < 1$$

### EXAMPLE 3.2

Determine  $E^*(s)$  for  $e(t) = \epsilon^{-t}$ . (From 3-7),

$$\begin{aligned} E^*(s) &= \sum_{n=0}^{\infty} e(nT)\epsilon^{-nTs} \\ &= 1 + \epsilon^{-T}\epsilon^{-Ts} + \epsilon^{-2T}\epsilon^{-2Ts} + \dots \\ &= 1 + \epsilon^{-(1+s)T} + (\epsilon^{-(1+s)T})^2 + \dots \\ &= \frac{1}{1 - \epsilon^{-(1+s)T}}, \quad |\epsilon^{-(1+s)T}| < 1 \end{aligned}$$

## 3.4 EVALUATION OF $E^*(s)$

$E^*(s)$ , as defined by (3-7), has limited usefulness in analysis because it is expressed as an infinite series. However, for many useful time functions,  $E^*(s)$  can be expressed in closed form. In addition, there is a third form of  $E^*(s)$  that is useful. These two additional forms of  $E^*(s)$  will now be investigated.

As was seen in Section 3.3, we can express the inverse Laplace transform of  $E^*(s)$  as [see (3-6)]

$$e^*(t) = e(t)\delta_T(t) \quad (3-9)$$

If we then take the Laplace transform of  $e^*(t)$  using the complex convolution integral [3], we can derive two additional expressions for  $E^*(s)$ . These derivations are given in Appendix III. The resultant expressions are

$$E^*(s) = \sum_{\substack{\text{at poles} \\ \text{of } E(\lambda)}} \left[ \text{residues of } E(\lambda) \frac{1}{1 - \epsilon^{-T(s-\lambda)}} \right] \quad (3-10)$$

$$E^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) + \frac{e(0)}{2} \quad (3-11)$$

where  $\omega_s$  is the radian sampling frequency, that is,  $\omega_s = 2\pi/T$ . The residue of a function is defined in Section 2.7. The expression (3-10) is useful in generating tables for the starred transform  $E^*(s)$ , and the expression (3-11) will prove to be useful in analysis, as illustrated in the next section. Some examples that illustrate the use of (3-10) will now be given.

### EXAMPLE 3.3

Determine  $E^*(s)$  given that

$$E(s) = \frac{1}{(s + 1)(s + 2)}$$

From (3-10),

$$E(\lambda) \frac{1}{1 - \epsilon^{-T(s-\lambda)}} = \frac{1}{(\lambda + 1)(\lambda + 2)(1 - \epsilon^{-T(s-\lambda)})}$$

Then

$$\begin{aligned} E^*(s) &= \sum_{\text{poles of } E(\lambda)} \left[ \text{residues of } E(\lambda) \frac{1}{1 - \epsilon^{-T(s-\lambda)}} \right] \\ &= \frac{1}{(\lambda + 2)(1 - \epsilon^{-T(s-\lambda)})} \Big|_{\lambda=-1} + \frac{1}{(\lambda + 1)(1 - \epsilon^{-T(s-\lambda)})} \Big|_{\lambda=-2} \\ &= \frac{1}{1 - \epsilon^{-T(s+1)}} - \frac{1}{1 - \epsilon^{-T(s+2)}} \end{aligned}$$

### EXAMPLE 3.4

We wish to determine the starred transform of  $e(t) = \sin \beta t$ . The corresponding  $E(s)$  is

$$E(s) = \frac{\beta}{s^2 + \beta^2} = \frac{\beta}{(s - j\beta)(s + j\beta)}$$

$E^*(s)$  can be evaluated from the expression

$$\begin{aligned} E^*(s) &= \sum_{\text{poles of } E(\lambda)} \left[ \text{residues of } \frac{\beta}{(\lambda - j\beta)(\lambda + j\beta)(1 - \epsilon^{-T(s-\lambda)})} \right] \\ &= \frac{\beta}{(\lambda + j\beta)(1 - \epsilon^{-T(s-\lambda)})} \Big|_{\lambda=j\beta} + \frac{\beta}{(\lambda - j\beta)(1 - \epsilon^{-T(s-\lambda)})} \Big|_{\lambda=-j\beta} \end{aligned}$$

$$= \frac{1}{2j} \left[ \frac{1}{1 - \varepsilon^{-Ts} \varepsilon^{j\beta T}} - \frac{1}{1 - \varepsilon^{-Ts} \varepsilon^{-j\beta T}} \right]$$

$$= \frac{\varepsilon^{-Ts} \sin \beta T}{1 - 2\varepsilon^{-Ts} \cos \beta T + \varepsilon^{-2Ts}}$$

using the equations from Euler's relation:

$$\cos \beta T = \frac{\varepsilon^{j\beta T} + \varepsilon^{-j\beta T}}{2}; \quad \sin \beta T = \frac{\varepsilon^{j\beta T} - \varepsilon^{-j\beta T}}{2j}$$

### EXAMPLE 3.5

Given  $e(t) = 1 - \varepsilon^{-t}$ , determine  $E^*(s)$ , first using (3-3) and then (3-10). Now,

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

From (3-3),

$$E^*(s) = \sum_{n=0}^{\infty} e(nT) \varepsilon^{-nTs}$$

$$= \sum_{n=0}^{\infty} (1 - \varepsilon^{-nT}) \varepsilon^{-nTs}$$

$$= \sum_{n=0}^{\infty} \varepsilon^{-nTs} - \sum_{n=0}^{\infty} \varepsilon^{-(1+s)nT}$$

$$= \frac{1}{1 - \varepsilon^{-Ts}} - \frac{1}{1 - \varepsilon^{-(1+s)T}}$$

From (3-10),

$$E^*(s) = \sum_{\substack{\lambda=0 \\ \lambda=-1}} \left[ \text{residues of } \frac{1}{\lambda(\lambda+1)} \frac{1}{1 - \varepsilon^{-T(s-\lambda)}} \right]$$

$$= \frac{1}{1 - \varepsilon^{-Ts}} - \frac{1}{1 - \varepsilon^{-(1+s)T}}$$

It is interesting to consider the case in which the function  $e(t)$  contains a time delay. For example, consider a delayed signal of the type

$$e(t) = e_1(t - t_0)u(t - t_0)$$

From the shifting property of the Laplace transform (see Appendix V),

$$E(s) = \varepsilon^{-t_0 s} \mathcal{L}[e_1(t)] = \varepsilon^{-t_0 s} E_1(s) \quad (3-12)$$

For this case, (3-10) does not apply; instead, special techniques are required to find the starred transform of a delayed signal in closed form. These techniques will be presented in Chapter 4, where the modified  $z$ -transform is developed. However, for the special case in which the time signal is delayed a whole number of sampling periods, (3-10) can be applied in a slightly different form,

$$[\varepsilon^{-kTs}E_1(s)]^* = \varepsilon^{-kTs} \sum_{\text{at poles of } E_1(\lambda)} \left[ \text{residues of } E_1(\lambda) \frac{1}{1 - \varepsilon^{-T(s-\lambda)}} \right] \quad (3-13)$$

where  $k$  is a positive integer (see Problem 3.4-4).

### EXAMPLE 3.6

The starred transform of  $e(t) = [1 - \varepsilon^{-(t-1)}]u(t - 1)$ , with  $T = 0.5$  s, will now be found. First we find  $E(s)$ :

$$E(s) = \frac{\varepsilon^{-s}}{s} - \frac{\varepsilon^{-s}}{s+1} = \frac{\varepsilon^{-s}}{s(s+1)}$$

Therefore, in (3-13),  $k$  is equal to 2 and

$$E_1(s) = \frac{1}{s(s+1)}$$

Then, from (3-13),

$$\begin{aligned} \left[ \frac{\varepsilon^{-s}}{s(s+1)} \right]^* &= \varepsilon^{-s} \sum_{\lambda=0, -1} \left[ \text{residues of } \frac{1}{\lambda(\lambda+1)(1 - \varepsilon^{-0.5(s-\lambda)})} \right] \\ &= \varepsilon^{-s} \left[ \frac{1}{(\lambda+1)(1 - \varepsilon^{-0.5(s-\lambda)})} \Big|_{\lambda=0} + \frac{1}{\lambda(1 - \varepsilon^{-0.5(s-\lambda)})} \Big|_{\lambda=-1} \right] \\ &= \varepsilon^{-s} \left[ \frac{1}{1 - \varepsilon^{-0.5s}} + \frac{-1}{1 - \varepsilon^{-0.5(s+1)}} \right] = \frac{(1 - \varepsilon^{-0.5})\varepsilon^{-1.5s}}{(1 - \varepsilon^{-0.5s})(1 - \varepsilon^{-0.5(s+1)})} \end{aligned}$$

## 3.5 RESULTS FROM THE FOURIER TRANSFORM

In this section we present some results regarding the Fourier transform. These results are helpful in understanding the effects of sampling a signal.

The Fourier transform is defined by [4]

$$\mathcal{F}[e(t)] = E(j\omega) = \int_{-\infty}^{\infty} e(t)\varepsilon^{-j\omega t} dt \quad (3-14)$$

In many texts the notation  $E(\omega)$  is used instead of  $E(j\omega)$ ; we discuss this further below.

Recall the unilateral Laplace transform from Appendix V:

$$\mathcal{L}[e(t)] = E(s) = \int_0^{\infty} e(t)\epsilon^{-st} dt \quad (3-15)$$

If  $e(t)$  is zero for  $t < 0$ , its Fourier transform is given by

$$\begin{aligned} \mathcal{F}[e(t)] = E(j\omega) &= \int_{-\infty}^0 e(t)\epsilon^{-j\omega t} dt + \int_0^{\infty} e(t)\epsilon^{-j\omega t} dt \\ &= \int_0^{\infty} e(t)\epsilon^{-j\omega t} dt = \mathcal{L}[e(t)] \Big|_{s=j\omega} \end{aligned}$$

provided that both transforms exists. This result can be expressed in general as

$$\mathcal{F}[e(t)u(t)] = \mathcal{L}[e(t)u(t)] \Big|_{s=j\omega} \quad (3-16)$$

Hence, for the case that  $e(t)$  is zero for negative time, the Fourier transform of  $e(t)$  is equal to the Laplace transform of  $e(t)$  with  $s$  replaced with  $j\omega$ . We see then the reason for expressing the Fourier transform as  $E(j\omega)$ . In (3-16) it is assumed that both the Fourier transform and the Laplace transform exist. Recall that neither transform may exist for a given time function, or that the Laplace transform may exist while the Fourier transform does not.

A plot of the Fourier transform  $E(j\omega)$  is called the *frequency spectrum* of  $e(t)$ . A common procedure for showing the frequency spectrum is to express  $E(j\omega)$  as

$$E(j\omega) = |E(j\omega)|\epsilon^{j\theta(j\omega)} = |E(j\omega)|\angle\theta(j\omega)$$

and plot  $|E(j\omega)|$  versus  $\omega$  (the amplitude spectrum) and  $\theta(j\omega)$  versus  $\omega$  (the phase spectrum). We interpret the amplitude spectrum as giving the relative contents of a signal in different frequency bands. The meaning of the frequency spectrum of a signal is fundamental to understanding the transmission of signals through systems; those readers who do not have a good background in this area should review this material in a signals and systems book such as [4].

Consider now an analog system with the input signal  $e(t)$ , the output signal  $y(t)$ , and the transfer function  $G(s)$ ; that is,

$$Y(s) = G(s)E(s) \quad (3-17)$$

If  $e(t)$  is the unit impulse function  $\delta(t)$ ,  $E(s) = 1$  and  $Y(s) = G(s)$ . Hence  $g(t) = \mathcal{L}^{-1}[G(s)]$  is the unit impulse response of the system with the transfer function  $G(s)$ . A physical system cannot respond to an input signal before this signal is applied. For this case,  $g(t) = 0$  for  $t < 0$ . A system that satisfies this condition is called a *causal system*. Consequently, for a causal system, the Fourier transform of the unit impulse response is equal to the Laplace transform of that response with  $s$  replaced with  $j\omega$ , provided that the Fourier transform exists. Hence, for the system of (3-17), the Fourier transform yields

$$Y(j\omega) = G(j\omega)E(j\omega)$$

We see that  $G(j\omega)$  determines the manner that the frequency spectrum of the input  $e(t)$  is modified to yield the frequency spectrum of the output  $y(t)$ . The function  $G(j\omega)$  is called the *frequency response*.

### 3.6 PROPERTIES OF $E^*(s)$

Two  $s$ -plane properties of  $E^*(s)$  are now given. These properties are very important and will be used extensively in later derivations.

**PROPERTY 1.**  $E^*(s)$  is periodic in  $s$  with period  $j\omega_s$ .

This property can be proved using either (3-3), (3-10), or (3-11). From (3-3),

$$E^*(s + jm\omega_s) = \sum_{n=0}^{\infty} e(nT)\epsilon^{-nT(s+jm\omega_s)} \quad (3-18)$$

Since  $\omega_s T = (2\pi/T)T = 2\pi$ , and from Euler's relationship,

$$\epsilon^{j\theta} = \cos \theta + j \sin \theta$$

then

$$\epsilon^{-jnm\omega_s T} = \epsilon^{-jnm2\pi} = 1, \quad \text{for } m \text{ an integer} \quad (3-19)$$

Thus, in (3-18),

$$E^*(s + jm\omega_s) = \sum_{n=0}^{\infty} e(nT)\epsilon^{-nTs} = E^*(s) \quad (3-20)$$

**PROPERTY 2.** If  $E(s)$  has a pole at  $s = s_1$ , then  $E^*(s)$  must have poles at  $s = s_1 + jm\omega_s$ ,  $m = 0, \pm 1, \pm 2, \dots$

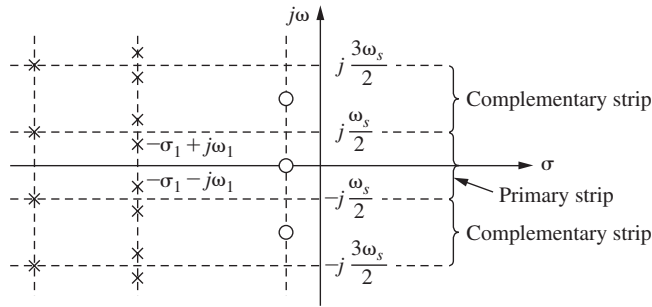
This property can be proved from (3-11). Consider  $e(t)$  to be continuous at all sampling instants. Then

$$E^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) = \frac{1}{T} [E(s) + E(s + j\omega_s) + E(s + 2j\omega_s) + \dots + E(s - j\omega_s) + E(s - 2j\omega_s) + \dots] \quad (3-21)$$

If  $E(s)$  has a pole at  $s = s_1$ , then each term of (3-21) will contribute a pole at  $s = s_1 + jm\omega_s$ , where  $m$  is an integer.

It is important to note that no equivalent statement can be made concerning the zeros of  $E^*(s)$ ; that is, the zero locations of  $E(s)$  do not uniquely determine the zero locations of  $E^*(s)$ . However, the zero locations are periodic with period  $j\omega_s$ , as indicated by the first property of  $E^*(s)$ .

An example of pole-zero locations of  $E^*(s)$  is given in Fig. 3-8. The primary strip in the  $s$ -plane is defined as the strip for which  $-\omega_s/2 \leq \omega \leq \omega_s/2$ , as shown in Fig. 3-8. Note that if the pole-zero locations are known for  $E^*(s)$  in the primary strip, then the pole-zero locations in the entire  $s$ -plane are also known.



**FIGURE 3-8** Pole-zero locations for  $E^*(s)$ .

In Fig. 3-8, if  $E(s)$  has a pole at  $-\sigma_1 + j\omega_1$ , the sampling operation will generate a pole in  $E^*(s)$  at  $-\sigma_1 + j(\omega_1 + \omega_s)$ . Conversely, if  $E(s)$  has a pole at  $-\sigma_1 + j(\omega_1 + \omega_s)$ , then  $E^*(s)$  will have a pole at  $-\sigma_1 + j\omega_1$ . In fact, a pole location in  $E(s)$  at  $-\sigma_1 + j(\omega_1 + k\omega_s)$ ,  $k$  an integer, will result in identical pole locations in  $E^*(s)$ , regardless of the integer value of  $k$ .

The conclusion above can also be seen from the example of Fig. 3-9. Note that both signals have the same starred transform, since the two signals have the same value at each sampling instant. Note also that  $\omega_1 = \omega_s/4$ , or  $\omega_s = 4\omega_1$ . Since

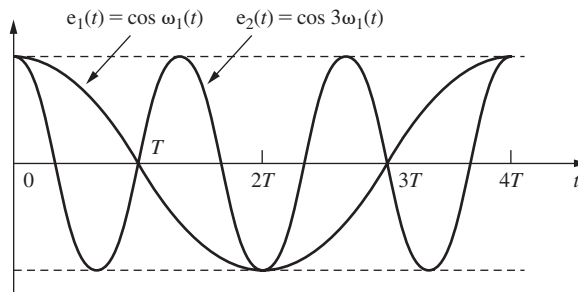
$$E_1(s) = \mathcal{L}[\cos \omega_1 t] = \frac{s}{s^2 + \omega_1^2} = \frac{s}{(s + j\omega_1)(s - j\omega_1)}$$

then  $E_1(s)$  has a pole at  $s = j\omega_1$ . And since

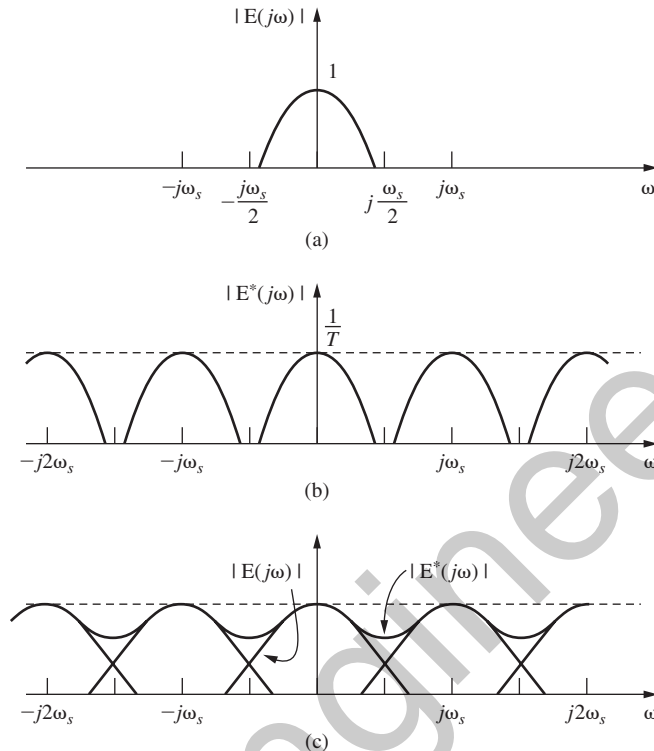
$$E_2(s) = \mathcal{L}[\cos 3\omega_1 t] = \frac{s}{(s + j3\omega_1)(s - j3\omega_1)}$$

then  $E_2(s)$  has a pole at  $s = -j3\omega_1 = j(\omega_1 - \omega_s)$ . The other pole of  $E_1(s)$  occurs at  $s = -j\omega_1$ , and the other pole of  $E_2(s)$  occurs at  $s = j3\omega_1 = j(-\omega_1 + \omega_s)$ .

The following discussion is based on the results of the Fourier transform given in the preceding section. Suppose that the signal  $e(t)$  has the amplitude spectrum  $|E(j\omega)|$  shown in Fig. 3-10(a), where  $E(j\omega)$  is the Fourier transform of  $e(t)$ . Then  $E^*(j\omega)$  has the amplitude spectrum shown in Fig. 3-10(b). This can be seen by evaluating (3-21) for  $s = j\omega$ .



**FIGURE 3-9** Two signals that have the same starred transform.



**FIGURE 3-10** Frequency spectra for  $E(j\omega)$  and  $E^*(j\omega)$ .

$$E^*(j\omega) = \frac{1}{T} [E(j\omega) + E(j\omega + j\omega_s) + E(j\omega + 2j\omega_s) + \dots + E(j\omega - j\omega_s) + E(j\omega - j2\omega_s) + \dots] + e(0)/2 \quad (3-22)$$

Hence the effect of ideal sampling is to replicate the original spectrum centered at  $\omega_s$ , at  $2\omega_s$ , at  $-\omega_s$ , at  $-2\omega_s$ , and so on.

An ideal filter is a filter with a unity gain in the passband and zero gain outside the passband. Of course, such a filter is not physically realizable [5]. It is seen from Fig. 3-10(b) that an ideal low-pass filter could completely recover  $E(j\omega)$ , and hence  $e(t)$ , if the bandwidth of the filter were  $\omega_s/2$ , for the case that the highest frequency present in  $E(j\omega)$  is less than  $\omega_s/2$ . This is, of course, essentially a statement of Shannon's sampling theorem [6].

**SHANNON'S SAMPLING THEOREM.** A time-varying signal  $e(t)$  whose Fourier transform contains no frequency components greater than  $f_0$  Hertz is uniquely determined by the values of  $e(t)$  at any set of sampling points spaced  $1/(2f_0)$  seconds apart.

Suppose, in Fig. 3-10(b), that  $\omega_s$  is decreased until the highest-frequency components present in  $E(j\omega)$  are greater than  $\omega_s/2$ . Then  $E^*(j\omega)$  has the amplitude spectrum shown in Fig. 3-10(c); and for this case, no filtering scheme, ideal or realizable, will recover  $e(t)$ . Thus, in choosing the sampling rate for a control system, the sampling frequency should be greater than twice the highest-frequency component of *significant amplitude* of the signal being sampled.



It is to be recalled that the ideal sampler is not a physical device, and thus the frequency spectrum, as shown in Fig. 3-10, is not the spectrum of a signal that appears in a physical system. The ideas above will be extended to signals that do appear in physical systems after an investigation of data holds.

### 3.7 DATA RECONSTRUCTION

In most feedback control systems employing sampled data, a continuous-time signal is reconstructed from the sampled signal. The block diagram of a simple sampled-data control system is repeated in Fig. 3-11. Suppose that the sampled signal is band limited in frequency, such that the highest-frequency component of  $e(t)$  is less than  $\omega_s/2$ . Then  $E^*(j\omega)$  would have the frequency spectrum shown in Fig. 3-10(b), and theoretically the signal could be reconstructed exactly by employing an ideal low-pass filter. However, since ideal filters do not exist in physically realizable systems, we must employ approximations. Practical data holds are devices that approximate, in some sense, an ideal low-pass filter.

The reader may ask: Why discuss a sampled-data system as illustrated in Fig. 3-11? The system contains a sampler to examine a continuous-time signal at discrete instants of time. Then a data hold is employed to try to reconstruct the original signal! Our reply is that many existing control systems actually operate in this manner due to hardware implementation techniques. More important, we will later add a digital compensator block between the sampler and data-hold device in order to improve system performance, and the sampler is required in this case. In either case, our discussion of data-hold devices in this chapter is prerequisite to closed-loop system analysis and synthesis.

A commonly used method of data reconstruction is polynomial extrapolation. Using a Taylor's series expansion about  $t = nT$ , we can express  $e(t)$  as

$$e(t) = e(nT) + e'(nT)(t - nT) + \frac{e''(nT)}{2!}(t - nT)^2 + \dots \quad (3-23)$$

where the prime denotes the derivative.  $e_n(t)$  is defined as the reconstructed version of  $e(t)$  for the  $n$ th sample period; that is,

$$e_n(t) \cong e(t) \quad \text{for} \quad nT \leq t < (n + 1)T \quad (3-24)$$

Thus  $e_n(t)$  denotes the output of the data hold. Since  $e(t)$  enters the data hold only in sampled form, the values of the derivatives are not known. However, the derivatives may be approximated by the backward difference

$$e'(nT) = \frac{1}{T}[e(nT) - e[(n - 1)T]] \quad (3-25)$$

$$e''(nT) = \frac{1}{T^2}[e'(nT) - e'[(n - 1)T]] \quad (3-26)$$

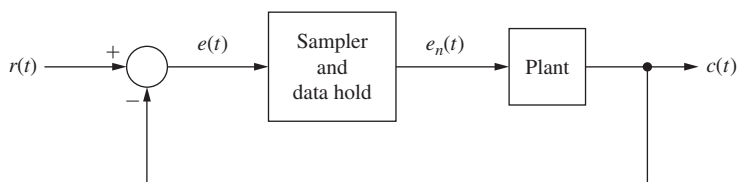


FIGURE 3-11 Sampled-data control system.

and so forth. Note that by substituting (3-25) into (3-26), we obtain

$$e''(nT) = \frac{1}{T} \left[ \frac{1}{T} [e(nT) - e[(n-1)T]] - \frac{1}{T} [e[(n-1)T] - e[(n-2)T]] \right]$$

or

$$e''(nT) = \frac{1}{T^2} [e(nT) - 2e[(n-1)T] + e[(n-2)T]] \quad (3-27)$$

Three types of data hold based on the foregoing relationships (the zero-, first-, and fractional-order holds) will now be discussed.

### Zero-Order Hold

If only the first term in the expansion of (3-23) is used, the data hold is called a zero-order hold. Here we assume that the function  $e(t)$  is approximately constant within the sampling interval at a value equal to that of the function at the preceding sampling instant. Therefore, for the zero-order hold,

$$e_n(t) = e(nT), \quad nT \leq t < (n+1)T \quad (3-28)$$

Recall that the  $e_n(t)$  for a zero-order hold has been defined as  $\bar{e}(t)$  in (3-1). Note that no memory is required, and therefore this data hold is the simplest to construct. The transfer function of the zero-order hold was derived in (3-2). However, this transfer function will be derived again using a simpler technique, one that may be easily used in the derivations of transfer functions for other data holds. Using the model of Fig. 3-5 for the sampler-data-hold device, we note that the input to the data hold will be only impulse functions. The signal  $e_o(t)$  shown in Fig. 3-12 then describes the data hold output if the input  $e_i(t)$  to the data hold is a unit impulse function. Thus

$$e_o(t) = u(t) - u(t - T)$$

and

$$E_o(s) = \frac{1}{s} - \frac{\epsilon^{-Ts}}{s}$$

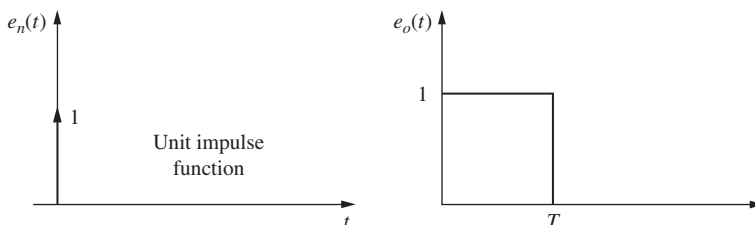


FIGURE 3-12 Input and output signals for the zero-order hold.

Since  $E_i(s) = 1$ , the transfer function of the zero-order hold is

$$G_{h0}(s) = \frac{E_o(s)}{E_i(s)} = \frac{1 - e^{-Ts}}{s} \quad (3-29)$$

as has already been shown in (3-2). In addition, recall that (3-29) is not the transfer function of a physical device since this equation was derived assuming that impulse functions occur at the input to the zero-order hold. Thus, as shown earlier in this chapter, if (3-7) is used to describe the sampling operation mathematically, and if (3-29) is used in conjunction with (3-7) to describe the data hold mathematically, a correct mathematical model of the overall sample-and-hold operation is obtained.

To obtain the frequency response of the zero-order hold, consider the following development:

$$\begin{aligned} G_{h0}(j\omega) &= \frac{1 - e^{-j\omega T}}{j\omega} e^{j(\omega T/2)} e^{-j(\omega T/2)} = \frac{2e^{-j(\omega T/2)}}{\omega} \left[ \frac{e^{j(\omega T/2)} - e^{-j(\omega T/2)}}{2j} \right] \\ &= T \frac{\sin(\omega T/2)}{\omega T/2} e^{-j(\omega T/2)} \end{aligned} \quad (3-30)$$

Since

$$\frac{\omega T}{2} = \frac{\omega}{2} \left( \frac{2\pi}{\omega_s} \right) = \frac{\pi\omega}{\omega_s}$$

The expression (3-30) can be rewritten as

$$G_{h0}(j\omega) = T \frac{\sin(\pi\omega/\omega_s)}{\pi\omega/\omega_s} e^{-j(\pi\omega/\omega_s)} \quad (3-31)$$

Thus

$$|G_{h0}(j\omega)| = T \left| \frac{\sin(\pi\omega/\omega_s)}{\pi\omega/\omega_s} \right| \quad (3-32)$$

and

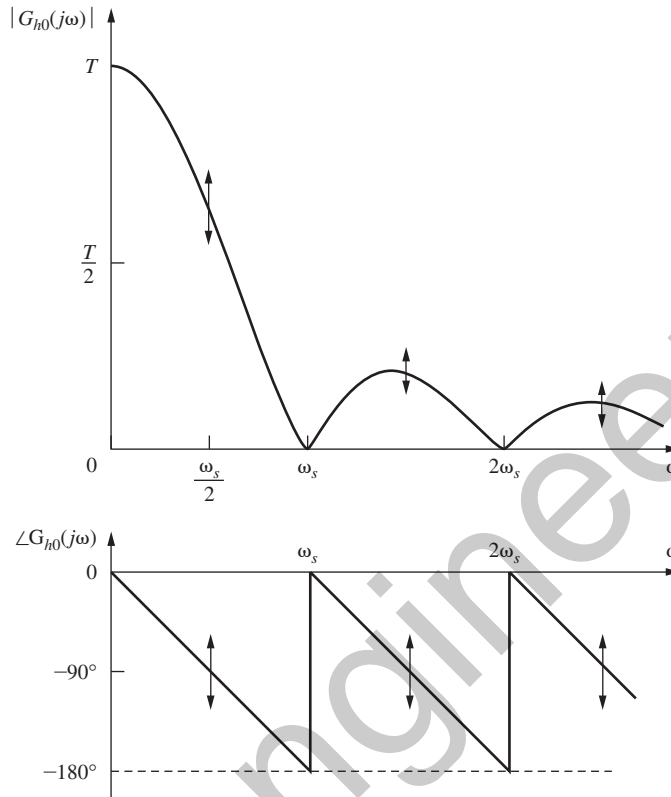
$$\angle G_{h0}(j\omega) = -\frac{\pi\omega}{\omega_s} + \theta, \quad \theta = \begin{cases} 0, & \sin(\pi\omega/\omega_s) > 0 \\ \pi, & \sin(\pi\omega/\omega_s) < 0 \end{cases} \quad (3-33)$$

The amplitude and phase plots for  $G_{h0}(j\omega)$  are shown in Fig. 3-13.

A word is in order concerning the interpretation of the frequency response of the zero-order hold. First, it must be remembered that the data hold must be preceded by an ideal sampler. Now, suppose that a sinusoid of frequency  $\omega_1$  is applied to the ideal sampler, where  $\omega_1 < \omega_s/2$ . Recall that the Fourier transform of  $e(t) = 2 \cos \omega_1 t$  is given by

$$E(j\omega) = \mathcal{F}[2 \cos \omega_1 t] = \delta(\omega - \omega_1) + \delta(\omega + \omega_1)$$

The frequency spectrum of this sinusoid is then two unit impulse functions, as shown in Fig. 3-14(a). In this Fig., the lengths of the arrows representing the impulse functions denote the



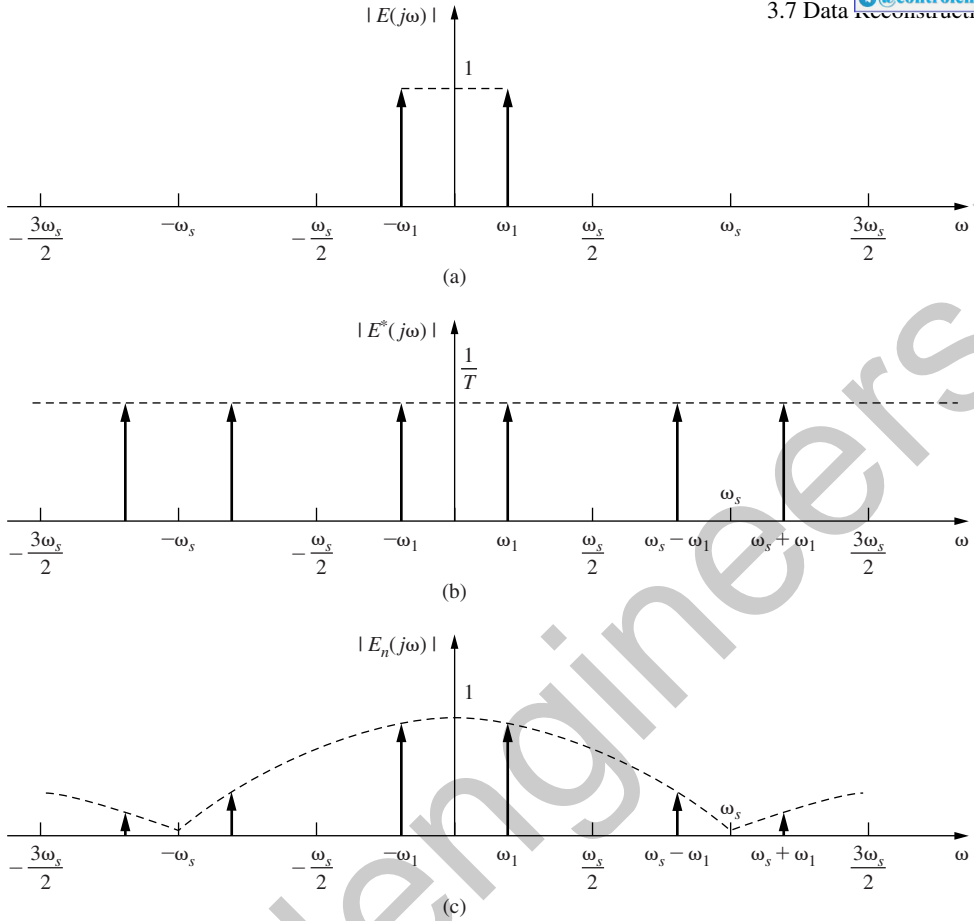
**FIGURE 3-13** Frequency response of the zero-order hold.

weights. Then, from (3-22), the output of the sampler contains the frequencies in the time domain represented by the impulse functions in the frequency domain, as shown in Fig. 3-14(b). Thus the frequency response of the zero-order hold may be used to determine the amplitude spectrum of the data-hold output signal. The output signal components are shown in Fig. 3-14(c). Hence a signal of the type shown in Fig. 3-15 has the frequency spectrum of Fig. 3-14(c).

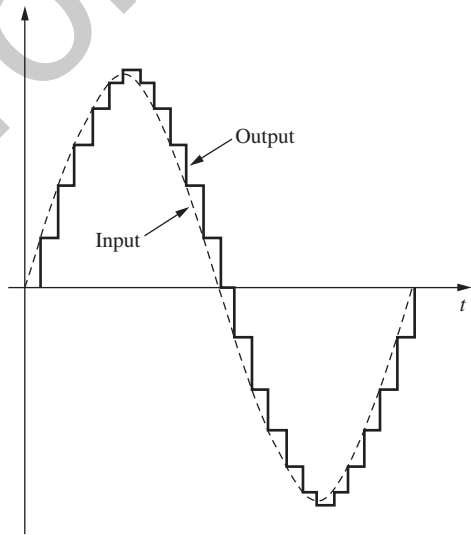
Note that the output signal amplitude spectrum will be the same as that shown in Fig. 3-14(c) if the input signal frequency is

$$\omega = k\omega_s \pm \omega_1, \quad k = 0, 1, 2, 3, \dots$$

Hence any frequencies  $\omega > \omega_s/2$  will reflect into the frequency range  $0 < \omega < \omega_s/2$ . This effect is called *frequency foldover*, or *frequency aliasing*. These reflected frequencies will be interpreted by the system as low-frequency information in  $e(t)$ , which generally cannot be tolerated. The frequency aliasing can be prevented either by increasing  $\omega_s$ , or by placing an analog *antialiasing* filter in front of the sampler. The antialiasing filter is a low-pass filter that removes any frequencies in  $e(t)$  that are greater than  $\omega_s/2$ . Since low-pass filters introduce phase lag, the cutoff frequency of the antialiasing filter cannot be made so low as to destabilize the control system. As a final point, note that, in Fig. 3-14, when  $\omega_1 \ll \omega_s/2$ , the high-frequency components of  $E^*(j\omega)$  will occur near zeros of  $G_{h0}(j\omega)$ ; hence the sampler and zero-order hold will have little effect on the signal.



**FIGURE 3-14** Sinusoidal response of the sampler and zero-order hold; (a) input signal to the ideal sampler; (b) output signal of the ideal sampler; (c) output signal from the zero-order hold.



**FIGURE 3-15** Output of a sampler/zero-order hold for a sinusoidal input.

Suppose that the frequency  $\omega_1$ , in Fig. 3-14, is equal to  $\omega_s/2$ . In this case the frequency components  $\omega_1$  and  $\omega_s - \omega_1$  are superimposed, and the data-hold output is a function of the phase of the input sinusoid. Note that the amplitude of the data-hold output can range from 0 to a value greater than the amplitude of the input signal. We emphasize this effect by showing arrows at  $k\omega_s + \omega_s/2, k = 0, 1, 2, \dots$ , in Fig. 3-13.

### First-Order Hold

The first two terms of (3-23) are used to realize the first-order hold. Therefore,

$$e_n(t) = e(nT) + e'(nT)(t - nT), \quad nT \leq t < (n + 1)T \quad (3-34)$$

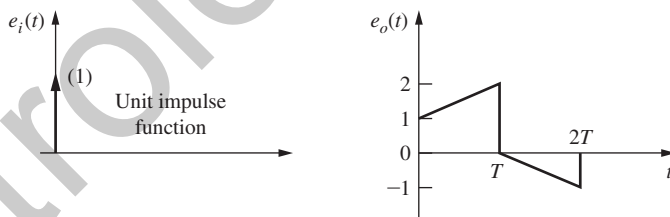
where, from (3-25),

$$e'(nT) = \frac{e(nT) - e[(n - 1)T]}{T} \quad (3-35)$$

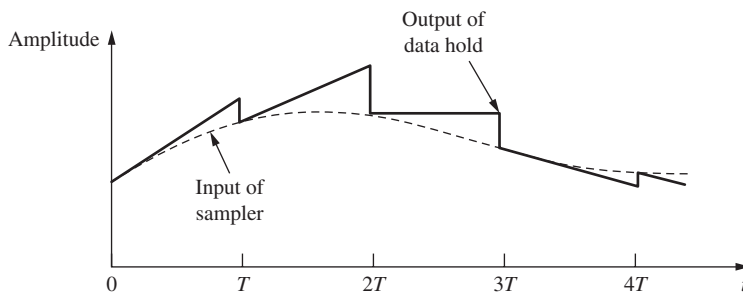
This expression indicates that the extrapolated function within a given interval is a straight line and that its slope is determined by the values of the function at the sampling instants in the previous interval. Note that memory is required in the realization of this data hold, since  $e[(n - 1)T]$  must be available at  $t = nT$ .

To determine the transfer function of a first-order hold, assume that the input is a unit impulse function. Then, from (3-34) and (3-35), the output, shown in Fig. 3-16(a), is

$$\begin{aligned}
 e_o(t) = & u(t) + \frac{1}{T}tu(t) - 2u(t - T) - \frac{2}{T}(t - T)u(t - T) + u(t - 2T) \\
 & + \frac{1}{T}(t - 2T)u(t - 2T)
 \end{aligned}$$



(a)



(b)

**FIGURE 3-16** Example of the output of a first-order hold.

Since  $E_i(s) = 1$ ,

$$G_{h1}(s) = \frac{E_o(s)}{E_i(s)} = \frac{1}{s} - \frac{2\varepsilon^{-Ts}}{s} + \frac{\varepsilon^{-2Ts}}{s} + \frac{1}{Ts^2}(1 - 2\varepsilon^{-Ts} + \varepsilon^{-2Ts})$$

or

$$G_{h1}(s) = \frac{1 + Ts}{T} \left[ \frac{1 - \varepsilon^{-Ts}}{s} \right]^2 \quad (3-36)$$

An example of the input–output waveforms of a sampler and first-order hold is shown in Fig. 3-16(b).

The frequency response of the first-order hold is obtained from (3-36).

$$G_{h1}(j\omega) = \frac{1 + j\omega T}{T} \left[ \frac{1 - \varepsilon^{-j\omega T}}{j\omega} \right]^2$$

$$|G_{h1}(j\omega)| = T \sqrt{1 + \frac{4\pi^2\omega^2}{\omega_s^2} \left[ \frac{\sin(\pi\omega/\omega_s)}{\pi\omega/\omega_s} \right]^2} \quad (3-37)$$

$$\angle G_{h1}(j\omega) = \tan^{-1} \left( \frac{2\pi\omega}{\omega_s} \right) - \frac{2\pi\omega}{\omega_s} \quad (3-38)$$

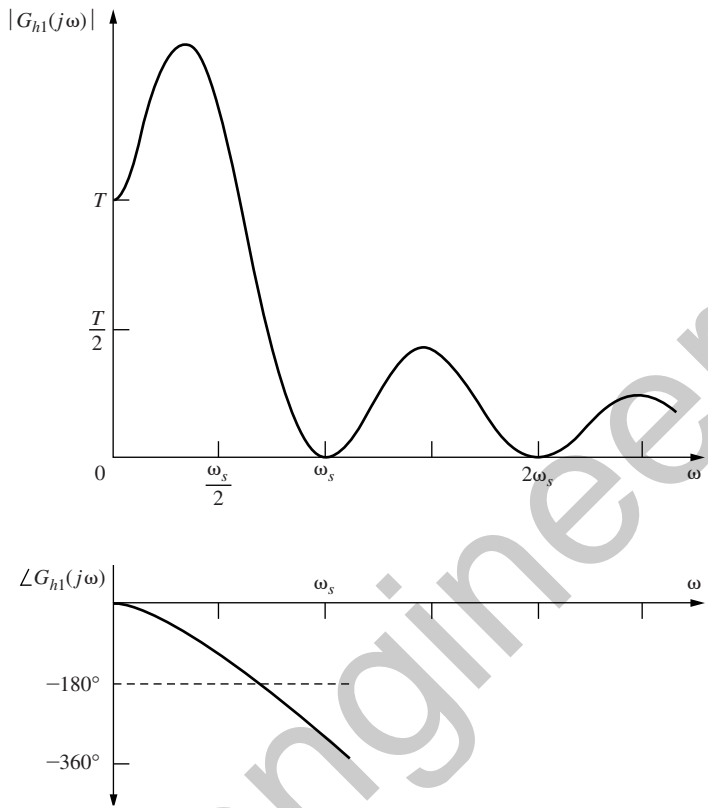
The amplitude and phase characteristics of the first-order hold are shown in Fig. 3-17. Note that the first-order hold provides a better approximation of the ideal low-pass filter in the vicinity of zero frequency than does the zero-order hold. However, for larger  $\omega$ , the zero-order hold yields a better approximation. Consider once again the sideband frequencies generated by the sampling process, as illustrated in Fig. 3-14(b). If  $\omega_1 \ll \omega_s/2$ , the first-order hold provides better reconstruction of the sampled signal than does the zero-order hold. However, if  $\omega_1$  is of the same order of magnitude as  $\omega_s/2$ , the zero-order hold may yield better results in the reconstruction process. Thus in some applications the zero-order hold is superior to the first-order hold. However, in any case, the zero-order hold is by far the most commonly used device, because of cost considerations.

### Fractional-Order Holds

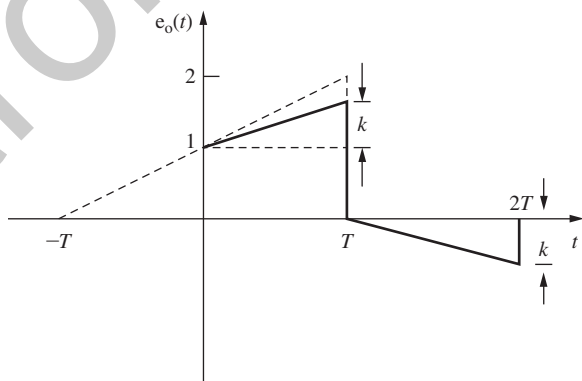
When using the first-order hold we essentially perform a linear extrapolation from one sampling interval to the next; that is, we assume that by using the approximate slope of the signal in the interval from  $(n - 1)T$  to  $nT$  we can obtain the value of the signal at  $(n + 1)T$ . The error generated in this process can be reduced by using only a fraction of the slope in the previous interval, as shown in Fig. 3-18. In this Fig. it is assumed that the input is a unit impulse and the value of  $k$  ranges from zero to unity. The frequency response of this data hold is shown in Fig. 3-19. Note that for  $k = 0$ , the hold is a zero-order hold; for  $k = 1$ , a first-order hold is obtained. Although it is difficult to determine the optimum value of  $k$  except in certain specific circumstances, the fractional-order hold may be used to match the data-hold frequency response to the sampled signal's frequency spectrum, thereby generating minimum error extrapolations. By using the techniques given above, the transfer function of the fractional-order hold can be shown to be

$$G_{hk}(s) = (1 - k\varepsilon^{-Ts}) \frac{1 - \varepsilon^{-Ts}}{s} + \frac{k}{Ts^2}(1 - \varepsilon^{-Ts})^2 \quad (3-39)$$

The derivation of this transfer function is given as an exercise in Problem 3.7-11.

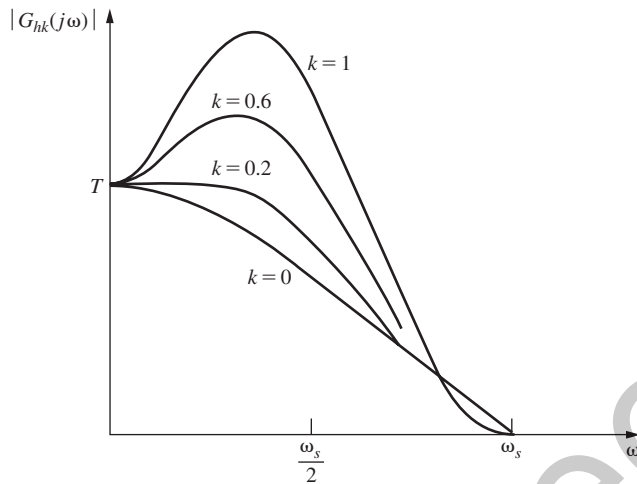


**FIGURE 3-17** Frequency response of the first-order hold.



**FIGURE 3-18** Impulse response of the fractional-order hold.





**FIGURE 3-19** Frequency response magnitudes for the fractional-order hold.

### 3.8 SUMMARY

The material presented in this chapter is fundamental to the study of sampled-data control systems. The sampling and reconstruction processes are an integral part of these systems, and it is important to have a good understanding of these topics before proceeding to the more advanced topics in analysis and design. The mathematical model of the sample-and-hold operation, Fig. 3-5, is basic to all analysis and design of digital control systems.

An important result of the topics developed in this chapter is the development of approximate rules for the choice of the sample period  $T$  for a given signal. Although this chapter emphasized the importance of the frequency contents of a signal in determining its sampling rate, later chapters will show the importance of a system's frequency response in determining the sample rate to be used in that system.

### References and Further Readings

- [1] C. L. Phillips and J. Parr, *Feedback Control Systems*, 5th ed. Upper Saddle River, NJ: Prentice-Hall, 2011.
- [2] F. G. Stremler, *Introduction to Communication Systems*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1990.
- [3] G. Doetsch, *Guide to the Applications of the Laplace and z-Transforms*, 2nd ed. New York: Van Nostrand-Reinhold, 1971.
- [4] C. L. Phillips, J. Parr, and E. Riskin, *Signals, Systems, and Transforms*, 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2007.
- [5] M. E. Van Valkenburg, *Network Analysis*. Upper Saddle River, NJ: Prentice-Hall, 1974.
- [6] R. M. Oliver, J. R. Pierce, and C. E. Shannon, "The Philosophy of Pulse Code Modulation," *Proc. IRE* 36(11), pp. 1324–1331, 1948.
- [7] J. W. Brown and R. V. Churchill, *Complex Variables and Applications*, 8th ed. New York: McGraw-Hill Book Company, 2009.
- [8] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Half Moon Bay, CA: Ellis-Kagle Press, 2006.
- [9] B. C. Kuo, *Digital Control Systems*, 2nd ed. New York: Oxford University Press, 1995.
- [10] C. L. Phillips, D. L. Chenoweth, and R. K. Cavin III, "z-Transform Analysis of Sampled Data Control Systems without Reference to Impulse Functions," *IEEE Trans. Educ.* E-11, pp. 141–144, 1968.
- [11] C. R. Wylie and L.C. Barrett, *Advanced Engineering Mathematics*, 6th ed. New York: McGraw-Hill Book Company, 1995.

## Problems

- 3.2-1.** (a) Give the definition of the starred transform.  
 (b) Give the definition of the  $z$ -transform.  
 (c) For a function  $e(t)$ , derive a relationship between its starred transform  $E^*(s)$  and its  $z$ -transform  $E(z)$ .
- 3.3-1.** A signal  $e(t)$  is sampled by the ideal sampler as specified by (3-3).  
 (a) List the conditions under which  $e(t)$  can be completely recovered from  $e^*(t)$ , that is, the conditions under which *no* loss of information by the sampling process occurs.  
 (b) State which of the conditions listed in part (a) can occur in a physical system. Recall that the ideal sampler itself is not physically realizable.  
 (c) Considering the answers in part (b), state why we can successfully employ systems that use sampling.
- 3.4-1.** Use the residue method of (3-10) to find the starred transform of the following functions.
- (a)  $E(s) = \frac{1}{(s+1)(s+2)(s+3)}$       (b)  $E(s) = \frac{10}{s(s+2)}$   
 (c)  $E(s) = \frac{s+2}{s(s+1)}$       (d)  $E(s) = \frac{s+2}{s^2(s+1)}$   
 (e)  $E(s) = \frac{s^2+5s+6}{s(s+4)(s+5)}$       (f)  $E(s) = \frac{2}{s^2+2s+5}$
- 3.4-2.** Find  $E^*(s)$  for each of the following functions. Express  $E^*(s)$  in closed form.
- (a)  $e(t) = e^{at}$       (b)  $E(s) = \frac{\epsilon^{-4Ts}}{s-2}$   
 (c)  $e(t) = \epsilon^{a(t-4T)}u(t-4T)$       (d)  $e(t) = \epsilon^{a(t-T/4)}u(t-T/4)$
- 3.4-3.** For  $e(t) = \epsilon^{-3t}$ .  
 (a) Express  $E^*(s)$  as a series.  
 (b) Express  $E^*(s)$  in closed form.  
 (c) Express  $E^*(s)$  as a series which is different from that in part (a).
- 3.4-4.** Express the starred transform of  $e(t-kT)u(t-kT)$ ,  $k$  an integer, in terms of  $E^*(s)$ , the starred transform of  $e(t)$ . Base your derivation on (3-3).
- 3.4-5.** Find  $E^*(s)$  for
- $$E(s) = \frac{1 - \epsilon^{-Ts}}{s(s+1)}$$
- 3.4-6.** Find  $E^*(s)$ , with  $T = 0.2$  s, for
- $$E(s) = \frac{\epsilon^{-2s}}{(s+1)(s+2)}$$
- 3.4-7.** Suppose that  $E^*(s) = [e(t)]^* = 1$ .  
 (a) Find  $e(kT)$  for all  $k$ .  
 (b) Can  $e(t)$  be found? Justify your answer.  
 (c) Sketch two different continuous-time functions that satisfy part (a).  
 (d) Write the equations for the two functions in part (c).
- 3.6-1.** (a) Find  $E^*(s)$ , for  $T = 0.1$  s, for the two functions below. Explain why the two transforms are equal, first from a time-function approach, and then from a pole-zero approach.  
 (i)  $e_1(t) = \cos(4\pi t)$   
 (ii)  $e_2(t) = \cos(16\pi t)$   
 (b) Give a third time function that has the same  $E^*(s)$ .

- 3.6-2.** Compare the pole-zero locations of  $E^*(s)$  in the  $s$ -plane with those of  $E(s)$ , for the functions given in Problem 3.4-1.
- 3.7-1.** A system is defined as linear if the principle of superposition applies. Is a sampler/zero-order-hold device linear? Prove your answer.
- 3.7-2.** Suppose that the signal  $e(t) = \cos[(\omega_s/2)t + \theta]$  is applied to an ideal sampler and zero-order hold.
- Show that the amplitude of the time function out of the zero-order hold is a function of the phase angle  $\theta$  by sketching this time function.
  - Show that the component of the signal out of the data hold, at the frequency  $\omega = \omega_s/2$ , is a function of the phase angle  $\theta$  by finding the Fourier series for the signal.
- 3.7-3.**
- A sinusoid with a frequency of 2 Hz is applied to a sampler/zero-order hold combination. The sampling rate is 10 Hz. List all the frequencies present in the output that are less than 50 Hz.
  - Repeat part (a) if the input sinusoid has a frequency of 8 Hz.
  - The results of parts (a) and (b) are identical. Give three other frequencies, which are greater than 50 Hz, that yield the same results as parts (a) and (b).
- 3.7-4.** Given the signal  $e(t) = 2 \sin 5t + 4 \sin 3t$ .
- List all frequencies less than  $\omega = 50$  rad/s that are present in  $e(t)$ .
  - The signal  $e(t)$  is sampled at the frequency  $\omega_s = 20$  rad/s. List all frequencies present in  $e^*(t)$  that are less than  $\omega = 50$  rad/s.
  - The signal  $e^*(t)$  is applied to a zero-order hold. List all frequencies present in the hold output that are less than  $\omega = 50$  rad/s.
  - Repeat part (c) for  $e^*(t)$  applied to a first-order hold.
- 3.7-5.** A signal  $e(t) = 2 \sin 3t$  is applied to a sampler/zero-order-hold device, with  $\omega = 6$  rad/s.
- What is the frequency component in the output that has the largest amplitude?
  - Find the amplitude and phase of that component.
  - Sketch the input signal and the component of part (b) versus time.
  - Find the ratio of the amplitude in part (b) to that of the frequency component in the output at  $\omega = 7$  rad/s (the input frequency).
- 3.7-6.** It is well known that the addition of phase lag to a closed-loop system is destabilizing. A sampler/data-hold device adds phase lag to a system, as described in Section 3.7. A certain analog control system has a bandwidth of 10 Hz. By this statement we mean that the system (approximately) will respond to frequencies less than 10 Hz, and (approximately) will not respond to frequencies greater than 10 Hz. A sampler/zero-order-hold device is to be added to this control system.
- It has been determined that the system can tolerate the addition of a maximum of  $10^\circ$  phase lag within the system bandwidth. Determine the approximate minimum sampling rate allowed, along with the sample period  $T$ .
  - Repeat part (a) for a maximum of  $5^\circ$  phase lag.
  - Repeat part (a) for a maximum of  $20^\circ$  phase lag.
- 3.7-7.** A sinusoid is applied to a sampler/zero-order-hold device, with a distorted sine wave appearing at the output, as shown in Fig. 3-15.
- With the sinusoid of unity amplitude and frequency 2 Hz, and with  $f_s = 12$  Hz, find the amplitude and phase of the component in the output at  $f_1 = 2$  Hz.
  - Repeat part (a) for the component in the output at  $(f_s - f_1) = 10$  Hz.
  - Repeat parts (a) and (b) for a sampler-first-order-hold device.
  - Comment on the distortion in the data-hold output for the cases considered in parts (a), (b), and (c).

3.7-8. A polygonal data hold is a device that reconstructs the sampled signal by the straight-line approximation shown in Fig. P3.7-8. Show that the transfer function of this data hold is

$$G(s) = \frac{\epsilon^{Ts}(1 - \epsilon^{-Ts})^2}{Ts^2}$$

Is this data hold physically realizable?

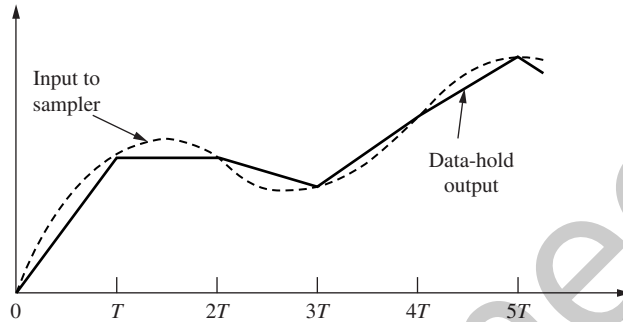


FIGURE P3.7-8 Response of a polygonal data hold.

3.7-9. A data hold is to be constructed that reconstructs the sampled signal by the straight-line approximation shown in Fig. P3.7-9. Note that this device is a polygonal data hold (see Problem 3.7-8) with a delay of  $T$  seconds. Derive the transfer function for this data hold. Is this data hold physically realizable?

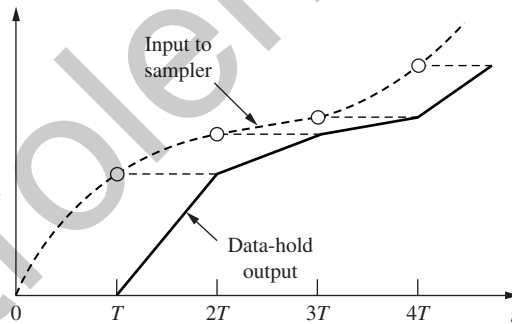
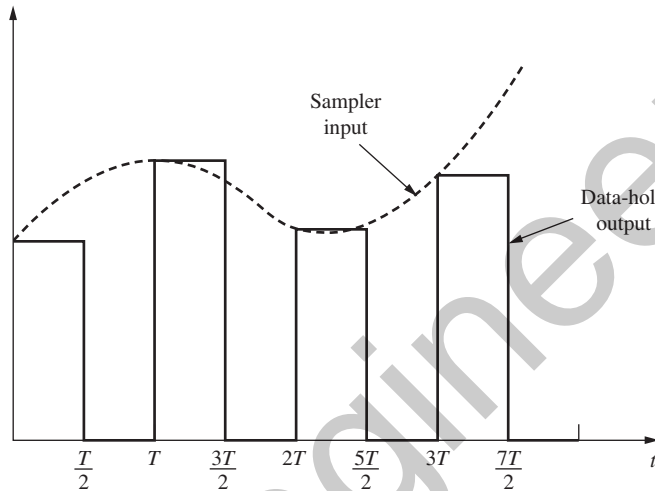


FIGURE P3.7-9 Response of a polygonal data hold with time delay.

3.7-10. Plot the ratio of the frequency responses (in decibels) and phase versus for the data holds of Problems 3.7-8 and 3.7-9. Note the effect on phase of making the data hold realizable.

3.7-11. Derive the transfer function of the fractional-order hold [see (3-39)].

- 3.7-12. Shown in Fig. P3.7-12 is the output of a data hold that clamps the output to the input for the first half of the sampling period, and returns the output to a value of zero for the last half of the sampling period.
- Find the transfer function of this data hold.
  - Plot the frequency response of this data hold.
  - Comparing this frequency response to that of the zero-order hold, comment on which would be better for data reconstruction.



**FIGURE P3.7-12** Data hold output for Problem 3.7-12.

# 4

## Open-Loop Discrete-Time Systems

### 4.1 INTRODUCTION

In the preceding chapters, the topics of discrete-time systems,  $z$ -transforms, and sampling and data reconstruction were developed. In this chapter these developments are utilized to derive analysis methods for open-loop discrete-time systems. (The derivations are necessary, since the ideal sampler does not have a transfer function.) These analysis techniques will then be extended to closed-loop systems in the chapters that follow.

### 4.2 THE RELATIONSHIP BETWEEN $E(z)$ AND $E^*(s)$

To provide the proper background for our analysis of open-loop systems, let us establish the relationship that exists between  $E(z)$  and  $E^*(s)$ . Recall that in Chapter 2 the  $z$ -transform of the number sequence  $\{e(k)\}$  was defined in equation (2-7) to be

$$\mathcal{Z}[\{e(k)\}] = E(z) = e(0) + e(1)z^{-1} + e(2)z^{-2} + \dots \quad (4-1)$$

In addition, the starred transform for the time function  $e(t)$  was defined in equation (3-7) as

$$E^*(s) = e(0) + e(T)\epsilon^{-Ts} + e(2T)\epsilon^{-2Ts} + \dots \quad (4-2)$$

The similarity between these two transforms is obvious. In fact, if we assume that the number sequence  $\{e(k)\}$  is obtained from sampling a time function  $e(t)$  [i.e., if  $e(k)$  of (4-1) is equal to  $e(kT)$  of (4-2)], and if  $\epsilon^{sT} = z$  in (4-2), then (4-2) becomes the  $z$ -transform. Hence in this case

$$E(z) = E^*(s) \Big|_{\epsilon^{sT} = z} \quad (4-3)$$

We see then that the  $z$ -transform can be considered to be a special case of the Laplace transform for our purposes. We will employ the change of variable in (4-3), and, in general, use the  $z$ -transform instead of the starred transform in our analysis of discrete-time systems. In addition, if the starred

transform is required, we will first use the  $z$ -transform tables to find  $E(z)$ , and then use the inverse of (4-3) to find  $E^*(s)$ . This approach is illustrated by the following example.

### EXAMPLE 4.1

Let

$$E(s) = \frac{1}{(s + 1)(s + 2)}$$

Then, from the  $z$ -transform tables in Appendix VI,

$$E(z) = E^*(s) \Big|_{\epsilon^{sT}=z} = \frac{z(\epsilon^{-T} - \epsilon^{-2T})}{(z - \epsilon^{-T})(z - \epsilon^{-2T})}$$

Hence

$$E^*(s) = \frac{\epsilon^{Ts}(\epsilon^{-T} - \epsilon^{-2T})}{(\epsilon^{Ts} - \epsilon^{-T})(\epsilon^{Ts} - \epsilon^{-2T})}$$

which checks that derived in Example 3.3 via (3-10).

Note that in Example 4.1,  $E^*(s)$  has an infinite number of poles and zeros in the  $s$ -plane. However,  $E(z)$  has only a single zero at  $z = 0$ , and two poles—one at  $\epsilon^{-T}$  and the other at  $\epsilon^{-2T}$ . Thus any analysis procedure that utilizes a pole-zero approach is greatly simplified through the use of the  $z$ -transform. Other advantages of this approach will become obvious as we develop analysis techniques for sampled-data systems.

$E(z)$  can now be calculated from (3-10) via the substitution in (4-3) as

$$E(z) = \sum_{\substack{\text{at poles} \\ \text{of } E(\lambda)}} \left[ \text{residues of } E(\lambda) \frac{1}{1 - z^{-1}\epsilon^{T\lambda}} \right] \quad (4-4)$$

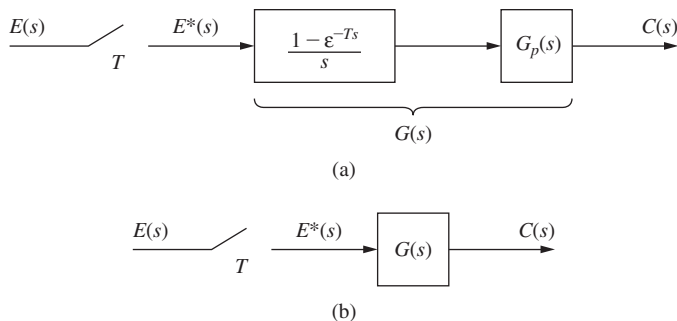
This expression is useful in generating  $z$ -transform tables. Because of the relation between  $E(z)$  and  $E^*(s)$ , the theorems developed in Chapter 2 for the  $z$ -transform also apply to the starred transform. In addition,  $z$ -transform tables become tables of starred transforms with the substitution  $z = \epsilon^{Ts}$ . For this reason, a separate table for starred transforms is usually not given.

## 4.3 THE PULSE TRANSFER FUNCTION

In this section we develop an expression for the  $z$ -transform of the output of open-loop sampled-data systems. This expression will be required later when we form closed-loop systems by feeding back this output signal.

Consider the open-loop system of Fig. 4-1(a), where  $G_p(s)$  is the plant transfer function. We denote the product of the plant transfer function and the zero-order hold transfer function as  $G(s)$ , as shown in the figure; that is,

$$G(s) = \frac{1 - \epsilon^{-Ts}}{s} G_p(s)$$



**FIGURE 4-1** Open-loop sampled-data system.

Hence this system can be represented as shown in Fig. 4-1(b). Note that when a representation of a system as shown in Fig. 4-1(b) is given,  $G(s)$  must contain the transfer function of a data hold. In general, we do not show the data-hold transfer function separately but combine it with the transfer function of that part of the system that follows the data hold.

In Fig. 4-1,

$$C(s) = G(s)E^*(s) \quad (4-5)$$

Assume that  $c(t)$  is continuous at all sampling instants. From (3-11),

$$C^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} C(s + jn\omega_s) = [G(s)E^*(s)]^* \quad (4-6)$$

where  $[\cdot]^*$  denotes the starred transform of the function in the brackets. Thus, from (4-5) and (4-6),

$$C^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} G(s + jn\omega_s)E^*(s + jn\omega_s) \quad (4-7)$$

Equation (3-20) gives the periodic property

$$E^*(s + jn\omega_s) = E^*(s)$$

Thus (4-7) becomes

$$C^*(s) = E^*(s) \frac{1}{T} \sum_{n=-\infty}^{\infty} G(s + jn\omega_s) = E^*(s)G^*(s) \quad (4-8)$$

and then from (4-3),

$$C(z) = E(z)G(z) \quad (4-9)$$

Now  $G(z)$  is called the *pulse transfer function* and is the transfer function between the sampled input and the output *at the sampling instants*. Note that the pulse transfer function gives no information on the nature of the output,  $c(t)$ , between sampling instants. This information is not



contained in either (4-8) or (4-9). However, we generally choose the sample frequency such that the response at sampling instants gives a very good indication of the response between sampling instants.

The derivation above is completely general. Thus given any function that can be expressed as

$$A(s) = B(s)F^*(s) \quad (4-10)$$

where  $F^*(s)$  must be expressible as

$$F^*(s) = f_0 + f_1\epsilon^{-Ts} + f_2\epsilon^{-2Ts} + \dots$$

then, from the preceding development,

$$A^*(s) = B^*(s)F^*(s) \quad (4-11)$$

Hence, from (4-3),

$$A(z) = B(z)F(z) \quad (4-12)$$

where  $B(s)$  is a function of  $s$  and  $F^*(s)$  is a function of  $\epsilon^{Ts}$ ; that is, in  $F^*(s)$ ,  $s$  appears only in the form  $\epsilon^{Ts}$ . Then, in (4-12),

$$B(z) = \mathfrak{z}[B(s)], \quad F(z) = F^*(s)|_{\epsilon^{Ts}=z} \quad (4-13)$$

The following examples illustrate this procedure.

### EXAMPLE 4.2

Suppose that we wish to find the  $z$ -transform of

$$A(s) = \frac{1 - \epsilon^{-Ts}}{s(s + 1)} = \frac{1}{s(s + 1)}(1 - \epsilon^{-Ts})$$

From (4-10), we consider

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

and

$$F^*(s) = 1 - \epsilon^{-Ts} \Rightarrow F(z) = 1 - z^{-1} = \frac{z - 1}{z}$$

Then, from the  $z$ -transform tables,

$$B(z) = \mathfrak{z}\left[\frac{1}{s(s + 1)}\right] = \frac{(1 - \epsilon^{-T})z}{(z - 1)(z - \epsilon^{-T})}$$

and

$$A(z) = B(z)F(z) = \frac{(1 - \epsilon^{-T})z}{(z - 1)(z - \epsilon^{-T})} \left[ \frac{z - 1}{z} \right] = \frac{1 - \epsilon^{-T}}{z - \epsilon^{-T}}$$

### EXAMPLE 4.3

Given the system shown in Fig. 4-2, with input  $e(t)$  a unit step function, let us determine the output function  $C(z)$ .

Now,

$$C(s) = \frac{1 - \epsilon^{-Ts}}{s(s + 1)} E^*(s) = G(s)E^*(s)$$

In Example 4.2 it was shown that

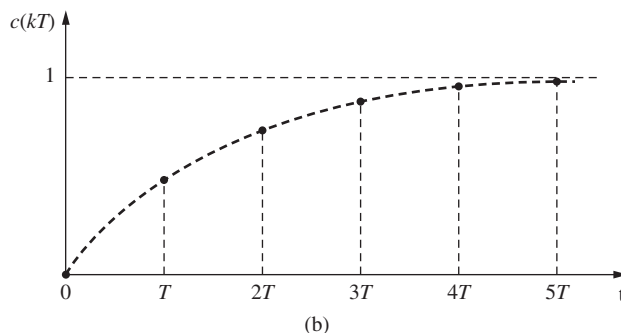
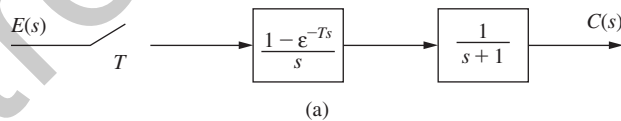
$$G(z) = \mathfrak{Z} \left[ \frac{1 - \epsilon^{-Ts}}{s(s + 1)} \right] = \frac{1 - \epsilon^{-T}}{z - \epsilon^{-T}}$$

In addition, from the table in Appendix VI,

$$E(z) = \mathfrak{Z}[u(t)] = \frac{z}{z - 1}$$

Thus

$$C(z) = G(z)E(z) = \frac{z(1 - \epsilon^{-T})}{(z - 1)(z - \epsilon^{-T})} = \frac{z}{z - 1} - \frac{z}{z - \epsilon^{-T}}$$



**FIGURE 4-2** Sampled-data system.

and the inverse  $z$ -transform of this function yields

$$c(kT) = 1 - \varepsilon^{-kT}$$

This response is plotted in Fig. 4-2(b).

Note that the output in Fig. 4-2(b) rises exponentially to a final value of unity at the sampling instants. Note also that the  $z$ -transform analysis yields the response *only* at the sampling instants. Based on the preceding analysis, we know nothing about the response of the system between sampling instants. If this information is needed (it normally is in a practical situation), we usually find the complete response by simulation.

If the input to the sampler/zero-order hold is a unit step signal, the output is also a unit step signal. Thus the zero-order hold reconstructs the sampled step function exactly. Because of this, the response of the system of Fig. 4-2 is simply the step response of a continuous-time system with a transfer function of  $1/(s + 1)$ . The reader can calculate this response to be

$$c(t) = 1 - \varepsilon^{-t}$$

The  $z$ -transform analysis of the last example is then seen to be correct. Given  $c(t)$ , we can find  $c(kT)$  by replacing  $t$  with  $kT$ . However, given  $c(kT)$  from a  $z$ -transform analysis, we *cannot* replace  $nT$  with  $t$  and have the correct expression for  $c(t)$ , in general.

A final point will be made relative to the example above. For many control systems, the steady-state gain for a constant input is important; we will call this gain the *dc gain*. For a system of the configuration of Fig. 4-1, the steady-state output,  $c_{ss}(k)$ , for an input of unity [ $E(z) = z/(z - 1)$ ], is, from the final-value property,

$$\begin{aligned} c_{ss}(k) &= \lim_{z \rightarrow 1} ((z - 1)C(z)) = \lim_{z \rightarrow 1} ((z - 1)G(z)E(z)) \\ &= \lim_{z \rightarrow 1} \left( (z - 1)G(z) \frac{z}{z - 1} \right) = \lim_{z \rightarrow 1} G(z) = G(1) \end{aligned}$$

assuming the  $c_{ss}(k)$  exists. Since the steady-state input is unity, we see that the dc gain is given by

$$\text{dc gain} = G(z)|_{z=1} = G(1)$$

For the example above,  $G(1) = 1$ , which checks the time-response calculation. Since, for a constant input, the gain of the sampler/zero-order-hold is unity, the dc gain of the system of Fig. 4-1 is also given by

$$\text{dc gain} = \lim_{s \rightarrow 0} G_p(s)$$

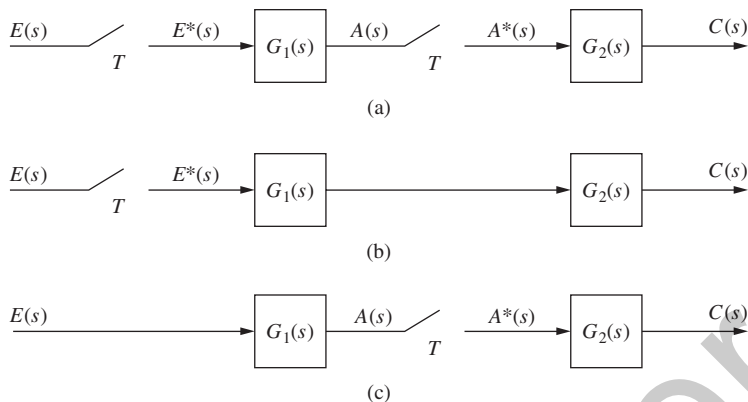
Thus

$$\text{dc gain} = \lim_{z \rightarrow 1} G(z) = \lim_{s \rightarrow 0} G_p(s) \quad (4-14)$$

This relationship gives us a relatively easy check on the calculation of  $G(z)$ , since each term in (4-14) is easily evaluated. For example, in Example 4.3,

$$\lim_{z \rightarrow 1} G(z) = \lim_{z \rightarrow 1} \left( \frac{1 - \varepsilon^{-T}}{z - \varepsilon^{-T}} \right) = 1; \quad \lim_{s \rightarrow 0} G_p(s) = \lim_{s \rightarrow 0} \left( \frac{1}{s + 1} \right) = 1$$

We will now investigate open-loop systems of other configurations. Consider the system of Fig. 4-3(a). In this system, there are two plants, and both  $G_1(s)$  and  $G_2(s)$  contain the transfer functions of the data holds.



**FIGURE 4-3** Open-loop sampled-data systems.

Now,

$$C(s) = G_2(s)A^*(s)$$

and thus

$$C(z) = G_2(z)A(z) \quad (4-15)$$

Also,

$$A(s) = G_1(s)E^*(s)$$

and thus

$$A(z) = G_1(z)E(z) \quad (4-16)$$

Then, from (4-15) and (4-16),

$$C(z) = G_1(z)G_2(z)E(z) \quad (4-17)$$

and the total transfer function is the product of the pulse transfer functions.

Consider now the system of Fig. 4-3(b). Of course, in this case,  $G_2(s)$  would not contain a data-hold transfer function. Then

$$C(s) = G_1(s)G_2(s)E^*(s)$$

and

$$C(z) = \overline{G_1G_2}(z)E(z)$$

where

$$\overline{G_1G_2}(z) = \mathcal{Z}[G_1(s)G_2(s)] \quad (4-18)$$

The bar above a product term indicates that the product must be performed in the  $s$ -domain before the  $z$ -transform is taken. In addition, note that

$$\overline{G_1 G_2(z)} \neq G_1(z)G_2(z) \quad (4-19)$$

that is, the  $z$ -transform of a product of functions is *not equal* to the product of the  $z$ -transforms of the functions.

For the system of Fig. 4-3(c),

$$C(s) = G_2(s)A^*(s) = G_2(s)\overline{G_1 E^*(s)}$$

Thus

$$C(z) = G_2(z)\overline{G_1 E(z)} \quad (4-20)$$

For this system a transfer function cannot be written; that is, we cannot factor  $E(z)$  from  $\overline{G_1 E(z)}$ .  $E(z)$  contains the values of  $e(t)$  only at  $t = kT$ . But the signal  $a(t)$  in Fig. 4-3(c) is a function of all previous values of  $e(t)$ , not just the values at sampling instants. Since

$$A(s) = G_1(s)E(s)$$

then, from the convolution property of the Laplace transform,

$$a(t) = \int_0^t g_1(t - \tau)e(\tau) d\tau \quad (4-21)$$

and the dependency of  $a(t)$  on all previous value of  $e(t)$  is seen. In general, if the input to a sample-data system is applied directly to a continuous-time part of the system before being sampled, the  $z$ -transform of the output of the system cannot be expressed as a function of the  $z$ -transform of the input signal. We will see later that this type of system presents no special problems in analysis and design.

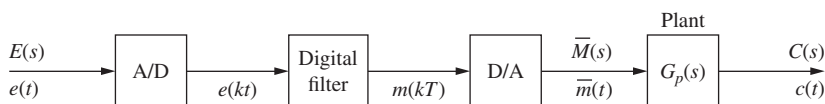
#### 4.4 OPEN-LOOP SYSTEMS CONTAINING DIGITAL FILTERS

In the preceding section a transfer-function technique was developed for open-loop sampled-data systems. In this section this technique is extended to cover the case in which the open-loop sampled-data system contains a digital filter.

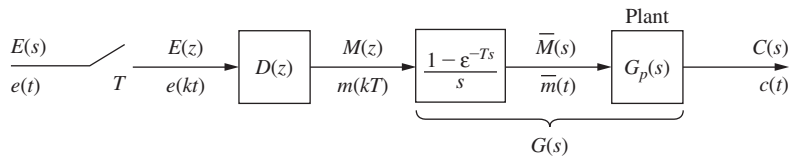
In the system shown in Fig. 4-4, the A/D converter on the filter input converts the continuous-time signal  $e(t)$  to a number sequence  $\{e(kT)\}$ ; the digital filter processes this number sequence  $\{e(kT)\}$  and generates the output number sequence  $\{m(kT)\}$ , which in turn is converted to the continuous-time signal  $\overline{m}(t)$  by the D/A converter.

As was shown in Chapter 2, a digital filter that solves a linear difference equation with constant coefficients can be represented by a transfer function  $D(z)$ , such that

$$M(z) = D(z)E(z) \quad (4-22)$$



**FIGURE 4-4** Open-loop system with a digital filter.



**FIGURE 4-5** Model for the open-loop system.

or, through the substitution  $z = \epsilon^{sT}$ ,

$$M^*(s) = D^*(s)E^*(s)$$

The D/A converter usually has an output data-hold register that gives the D/A the characteristics of a zero-order hold, and  $\bar{m}(t)$  is of the form of  $\bar{e}(t)$  in Fig. 3-3. Thus the Laplace transform of the signal  $\bar{m}(t)$  can be expressed as, from (3-2),

$$\bar{M}(s) = \frac{1 - \epsilon^{-Ts}}{s} M^*(s)$$

Hence

$$C(s) = G_p(s)\bar{M}(s) = G_p(s) \frac{1 - \epsilon^{-Ts}}{s} M^*(s)$$

Then, from (4-22),

$$C(s) = G_p(s) \frac{1 - \epsilon^{-Ts}}{s} D(z) \Big|_{z=\epsilon^{Ts}} E^*(s) \quad (4-23)$$

and we see that the filter and associated A/D and D/A converters can be represented in block-diagram form as shown in Fig. 4-5. Hence, from (4-23) or Fig. 4-5,

$$C(z) = \mathcal{Z} \left[ G_p(s) \frac{1 - \epsilon^{-Ts}}{s} \right] D(z) E(z) = G(z) D(z) E(z) \quad (4-24)$$

The digital computing device that implements the digital filter in Fig. 4-4 actually processes the values of the input data samples  $\{e(kT)\}$ . However, our model for the digital filter processes a sequence of *impulse functions* of weight  $\{e(kT)\}$ . Hence the complete model *must be used* as depicted in Fig. 4-5; that is, the combination of an ideal sampler,  $D(z)$ , and a zero-order hold does accurately model the combination of the A/D, digital filter, and D/A.

#### EXAMPLE 4.4

Let us determine the step response of the system shown in Fig. 4-5.

Suppose that the filter is described by the difference equation

$$m(kT) = 2e(kT) - e[(k - 1)T]$$

and thus

$$D(z) = \frac{M(z)}{E(z)} = 2 - z^{-1} = \frac{2z - 1}{z}$$

In addition, suppose that

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

Then, as shown in Example 4.3,

$$\mathfrak{Z} \left[ \frac{1 - \epsilon^{-Ts}}{s(s + 1)} \right] = \frac{1 - \epsilon^{-T}}{2 - \epsilon^{-T}}$$

Since  $E(z) = z/(z - 1)$ , from (4-24) we obtain

$$\begin{aligned} C(z) &= D(z)G(z)E(z) \\ &= \frac{2z - 1}{z} \left[ \frac{1 - \epsilon^{-T}}{z - \epsilon^{-T}} \right] \frac{z}{z - 1} = \frac{(2z - 1)(1 - \epsilon^{-T})}{(z - 1)(z - \epsilon^{-T})} \end{aligned}$$

By partial fractions,

$$\frac{C(z)}{z} = \frac{(2z - 1)(1 - \epsilon^{-T})}{z(z - 1)(z - \epsilon^{-T})} = \frac{1 - \epsilon^{-T}}{z} + \frac{1}{z - 1} + \frac{\epsilon^{-T} - 2}{z - \epsilon^{-T}}$$

and

$$C(z) = (1 - \epsilon^{-T}) + \frac{z}{z - 1} + \frac{(\epsilon^{-T} - 2)z}{z - \epsilon^{-T}}$$

Since

$$\mathfrak{Z}^{-1}[k_i z^{-i}] = \begin{cases} k_i, & n = i \\ 0, & n \neq i \end{cases}$$

the inverse  $z$ -transform of the first term in  $C(z)$  has a nonzero value only for  $n = 0$ , and thus

$$c(nT) = 1 + (\epsilon^{-T} - 2)\epsilon^{-nT}, \quad n = 1, 2, 3, \dots$$

and

$$c(0) = 1 - \epsilon^{-T} + 1 + \epsilon^{-T} - 2 = 0$$

Note that the value  $c(0) = 0$  is obvious from inspection of  $C(z)$ , since the order of the numerator is less than the order of the denominator. The verification of this example is considered in Problem 4-11.

We can partially verify the response of the system of Example 4.4. Note that the final value of  $c(nT)$  is unity. From the final value property of the  $z$ -transform, we verify this value:

$$\lim_{n \rightarrow \infty} c(nT) = \lim_{z \rightarrow 1} (z - 1)C(z) = \lim_{z \rightarrow 1} \frac{(2z - 1)(1 - \epsilon^{-T})}{z - \epsilon^{-T}} = 1$$

In addition, from (4-14), the dc gain of the system is given by

$$\text{dc gain} = D(z) \Big|_{z=1} G_p(s) \Big|_{s=0} = \frac{2z - 1}{z} \Big|_{z=1} \frac{1}{s + 1} \Big|_{s=0} = 1$$

Since in the steady state, the input is constant at a value of unity, the steady-state value of the output is

$$\lim_{n \rightarrow \infty} c(nT) = (\text{dc gain})(\text{constant input}) = (1)(1) = 1$$

which once again verifies the steady-state output.

As a final point in this example, note that the output of the digital filter is given by

$$M(z) = (2 - z^{-1})E(z) \Rightarrow m(kT) = 2u(kT) - u[(k - 1)T]$$

Hence in the steady state, the plant input is constant at a value of unity. It was shown in Example 4.3 that this condition results in a steady-state output of unity.

## 4.5 THE MODIFIED z-TRANSFORM

The analysis of open-loop systems, including those that contain digital filters, has been presented in preceding sections. However, this technique of analysis does not apply to systems containing ideal time delays. To analyze systems of this type, it is necessary to define the  $z$ -transform of a delayed time function. This transform is called the modified  $z$ -transform, which is developed in this section.

The modified  $z$ -transform can be developed by considering a time function  $e(t)$  that is delayed by an amount  $\Delta T$ ,  $0 < \Delta \leq 1$ , that is, by considering  $e(t - \Delta T)u(t - \Delta T)$ . The ordinary  $z$ -transform of the delayed time function is

$$\mathcal{Z}[e(t - \Delta T)u(t - \Delta T)] = \mathcal{Z}[E(s)\epsilon^{-\Delta Ts}] = \sum_{n=1}^{\infty} e(nT - \Delta T)z^{-n} \quad (4-25)$$

Note that the sampling is not delayed; that is, the sampling instants are  $t = 0, T, 2T, \dots$ . The  $z$ -transform in (4-25) is called the delayed  $z$ -transform, and thus, by definition, the delayed  $z$ -transform of  $e(t)$  is

$$E(z, \Delta) = \mathcal{Z}[e(t - \Delta T)u(t - \Delta T)] = \mathcal{Z}[E(s)\epsilon^{-\Delta Ts}] \quad (4-26)$$

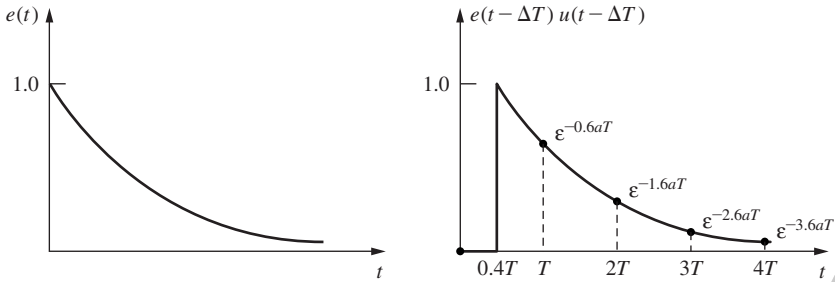
The delayed starred transform is also defined in (4-26), with the substitution  $z = \epsilon^{Ts}$ . The delayed  $z$ -transform will now be illustrated by an example.

### EXAMPLE 4.5

Find  $E(z, \Delta)$ , if  $\Delta = 0.4$ , for  $e(t) = \epsilon^{-at} u(t)$ . From (4-25),

$$\begin{aligned} E(z, \Delta) &= \epsilon^{-0.6aT} z^{-1} + \epsilon^{-1.6aT} z^{-2} + \epsilon^{-2.6aT} z^{-3} + \dots \\ &= \epsilon^{-0.6aT} z^{-1} [1 + \epsilon^{-aT} z^{-1} + \epsilon^{-2aT} z^{-2} + \dots] \\ &= \frac{\epsilon^{-0.6aT} z^{-1}}{1 - \epsilon^{-aT} z^{-1}} = \frac{\epsilon^{-0.6aT}}{z - \epsilon^{-aT}} \end{aligned}$$





**FIGURE 4-6** Example of the delayed z-transform.

The sketches in Fig. 4-6 show both  $e(t)$  and  $e(t - \Delta T)$ .

The modified z-transform is defined from the delayed z-transform. By definition, the modified z-transform of a function is equal to the delayed z-transform with  $\Delta$  replaced by  $1 - m$ . Thus if we let  $E(z, m)$  be the modified z-transform of  $E(s)$ , then, from (4-26),

$$E(z, m) = E(z, \Delta) \Big|_{\Delta=1-m} = \mathcal{Z}[E(s)\epsilon^{-\Delta Ts}] \Big|_{\Delta=1-m} \quad (4-27)$$

From (4-25) and (4-27),

$$\begin{aligned} E(z, m) &= [e(T - \Delta T)z^{-1} + e(2T - \Delta T)z^{-2} \\ &\quad + e(3T - \Delta T)z^{-3} + \dots]_{\Delta=1-m} \\ &= e(mT)z^{-1} + e[(1 + m)T]z^{-2} + e[(2 + m)T]z^{-3} + \dots \end{aligned} \quad (4-28)$$

with  $\Delta = 1 - m$ . Two properties of the modified z-transform are

$$E(z, 1) = E(z, m) \Big|_{m=1} = E(z) - e(0) \quad (4-29)$$

and

$$E(z, 0) = E(z, m) \Big|_{m=0} = z^{-1}E(z) \quad (4-30)$$

The value  $m = 1$  denotes no delay [but  $e(0)$  does not appear in (4-28)], and the value  $m = 0$  denotes a delay of one sample period.

### EXAMPLE 4.6

It is desired to find the modified z-transform of  $e(t) = \epsilon^{-t}$ . From (4-28),

$$\begin{aligned} E(z, m) &= \epsilon^{-mT}z^{-1} + \epsilon^{-(1+m)T}z^{-2} + \epsilon^{-(2+m)T}z^{-3} + \dots \\ &= \epsilon^{-mT}z^{-1}[1 + \epsilon^{-T}z^{-1} + \epsilon^{-2T}z^{-2} + \dots] \\ &= \frac{\epsilon^{-mT}z^{-1}}{1 - \epsilon^{-T}z^{-1}} = \frac{\epsilon^{-mT}}{z - \epsilon^{-T}} \end{aligned}$$

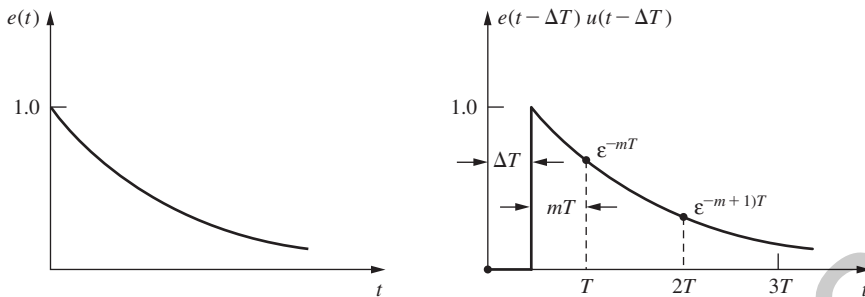


FIGURE 4-7 Example of the modified z-transform.

A sketch of the time functions is given in Fig. 4-7.

The ordinary  $z$ -transform tables do not apply to the modified  $z$ -transform. Instead, special tables must be derived. These tables are obtained by the following development. One can express the modified  $z$ -transform of a function  $E(s)$  in the following manner, from the development in Appendix III.

$$E(z, m) = \mathfrak{z}[E(s) \epsilon^{-\Delta Ts}]|_{\Delta=1-m} = \mathfrak{z}[E(s) \epsilon^{-(1-m)Ts}] = z^{-1} \mathfrak{z}[E(s) \epsilon^{mTs}] \quad (4-31)$$

Now  $\lim_{\lambda \rightarrow \infty} \lambda E(\lambda) \epsilon^{mT\lambda}$  is zero in the second integral in (A3-7) (see Fig. A3-2), and thus (3-10) applies. Then, from (3-10) and (4-31),

$$E(z, m) = z^{-1} \sum_{\text{poles of } E(\lambda)} \text{residues of } E(\lambda) \epsilon^{mT\lambda} \frac{1}{1 - z^{-1} \epsilon^{T\lambda}} \quad (4-32)$$

Also, from (3-11),

$$E^*(s, m) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) \epsilon^{-(1-m)(s+jn\omega_s)T} \quad (4-33)$$

provided that  $e(t - \Delta T)$  is continuous at all sampling instants. Modified  $z$ -transform tables are included with the ordinary  $z$ -transform tables in Appendix VI.

A useful property of the modified  $z$ -transform will now be stated. Since the modified  $z$ -transform is the ordinary  $z$ -transform of a shifted function, the theorems of the ordinary  $z$ -transform derived in Chapter 2 may be applied to the modified  $z$ -transform, if care is exercised. In particular, the shifting theorem applies directly.

Let  $\mathfrak{z}_m[\cdot]$  indicate the modified  $z$ -transform; that is,

$$\mathfrak{z}_m[E(s)] = E(z, m) = \mathfrak{z}_m[\epsilon^{-\Delta Ts} E(s)]|_{\Delta=1-m} \quad (4-34)$$

Then, by the shifting theorem, for  $k$  a positive integer,

$$\mathfrak{z}_m[\epsilon^{-kTs} E(s)] = z^{-k} \mathfrak{z}_m[E(s)] = z^{-k} E(z, m) \quad (4-35)$$

**EXAMPLE 4.7**

We wish to find the modified  $z$ -transform of the function  $e(t) = t$ . It is well known that  $E(s) = 1/s^2$ . This function has a pole of order 2 at  $s = 0$ . Therefore, the modified  $z$ -transform can be obtained from (4-32) as [see (2-34)]

$$\begin{aligned} E(z, m) &= z^{-1} \left[ \frac{d}{d\lambda} \left[ \frac{\epsilon^{mT\lambda}}{1 - z^{-1}\epsilon^{T\lambda}} \right]_{\lambda=0} \right] \\ &= z^{-1} \left[ \frac{(1 - z^{-1}\epsilon^{T\lambda})mT\epsilon^{mT\lambda} - \epsilon^{mT\lambda}(-Tz^{-1}\epsilon^{T\lambda})}{(1 - z^{-1}\epsilon^{T\lambda})^2} \right]_{\lambda=0} \\ &= z^{-1} \left[ \frac{mT(1 - z^{-1}) + Tz^{-1}}{(1 - z^{-1})^2} \right] \\ &= \frac{mT(z - 1) + T}{(z - 1)^2} \end{aligned}$$

which is verified in the table in Appendix VI.

**4.6 SYSTEMS WITH TIME DELAYS**

The modified  $z$ -transform may be used to determine the pulse transfer functions of discrete-time systems containing ideal time delays. To illustrate this, consider the system of Fig. 4-8, which has an ideal time delay of  $t_0$  seconds. For this system,

$$C(s) = G(s)\epsilon^{-t_0s}E^*(s) \tag{4-36}$$

Thus

$$C(z) = \mathfrak{z}[G(s)\epsilon^{-t_0s}]E(z) \tag{4-37}$$

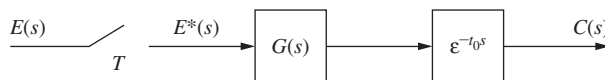
If we now let

$$t_0 = kT + \Delta T, \quad 0 < \Delta < 1 \tag{4-38}$$

where  $k$  is a positive integer, then from (4-35),

$$C(z) = z^{-k} \mathfrak{z}[G(s)\epsilon^{-\Delta Ts}]E(z) = z^{-k}G(z, m)E(z) \tag{4-39}$$

where  $m = 1 - \Delta$ . The foregoing development will now be illustrated with an example.



**FIGURE 4-8** System with ideal time delay.

**EXAMPLE 4.8**

In Fig. 4-8, let the input be a unit step,  $t_0 = 0.4T$ , and

$$G(s) = \frac{1 - \epsilon^{-Ts}}{s(s + 1)}$$

This system was considered in Example 4.3 with no delay. From (4-35) and the modified  $z$ -transform tables,

$$\begin{aligned} G(z, m) &= \mathfrak{Z}_m \left[ \frac{1 - \epsilon^{-Ts}}{s(s + 1)} \right] = (1 - z^{-1}) \mathfrak{Z}_m \left[ \frac{1}{s(s + 1)} \right] \\ &= \frac{z - 1}{z} \left[ \frac{z(1 - \epsilon^{-mT}) + \epsilon^{-mT} - \epsilon^{-T}}{(z - 1)(z - \epsilon^{-T})} \right] \end{aligned}$$

Thus, since  $mT = T - \Delta T = 0.6T$ ,

$$G(z, m) = \frac{z - 1}{z} \left[ \frac{z(1 - \epsilon^{-0.6T}) + \epsilon^{-0.6T} - \epsilon^{-T}}{(z - 1)(z - \epsilon^{-T})} \right]$$

Since  $k = 0$  in (4-39),  $C(z)$  is seen to be

$$C(z) = G(z, m) \frac{z}{z - 1} = \frac{z(1 - \epsilon^{-0.6T}) + \epsilon^{-0.6T} - \epsilon^{-T}}{(z - 1)(z - \epsilon^{-T})}$$

By the power-series method of Section 2.6,

$$C(z) = (1 - \epsilon^{-0.6T})z^{-1} + (1 - \epsilon^{-1.6T})z^{-2} + (1 - \epsilon^{-2.6T})z^{-3} + \dots$$

From Example 4.3, the response of this system with no delay is  $c(nT) = 1 - \epsilon^{-nT}$ . This response delayed by  $0.4T$  is then

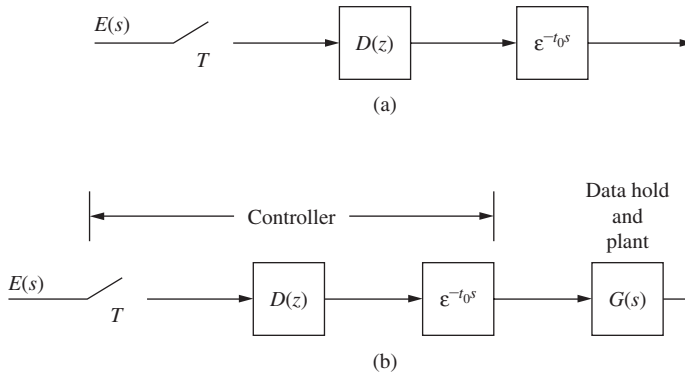
$$c(nT) |_{n \leftarrow (n-0.4)} = 1 - \epsilon^{-(n-0.4)T}, \quad n \geq 1 \tag{4-40}$$

which checks the results of this example.

The modified  $z$ -transform may also be used to determine the pulse transfer functions of digital control systems in which the computation time of the digital computer cannot be neglected. As given in (2-4), an  $n$ th-order linear digital controller solves the difference equation

$$\begin{aligned} m(k) &= b_n e(k) + b_{n-1} e(k - 1) + \dots + b_0 e(k - n) \\ &\quad - a_{n-1} m(k - 1) - \dots - a_0 m(k - n) \end{aligned} \tag{4-41}$$

every  $T$  seconds. Let the time required for the digital controller to computer (4-41) be  $t_0$  seconds. Thus an input at  $t = 0$  produces an output at  $t = t_0$ , an input at  $t = T$  produces an output at  $t = T + t_0$ , and so on. Hence the digital controller may be modeled as a digital controller



**FIGURE 4-9** Digital controller with nonzero computation time.

without time delay, followed by an ideal time delay of  $t_0$  seconds, as shown in Fig. 4-9(a). An open-loop system containing this controller may be modeled as shown in Fig. 4-9(b).

For this system,

$$C(z) = \mathcal{Z}[G(s)\epsilon^{-t_0 s}]D(z)E(z) \quad (4-42)$$

If we let

$$t_0 = kT + \Delta T, \quad 0 < \Delta < 1 \quad (4-43)$$

with  $k$  a positive integer, then from (4-39) and (4-42) we obtain

$$C(z) = z^{-k}G(z, m)D(z)E(z) \quad (4-44)$$

where  $m = 1 - \Delta$ .

### EXAMPLE 4.9

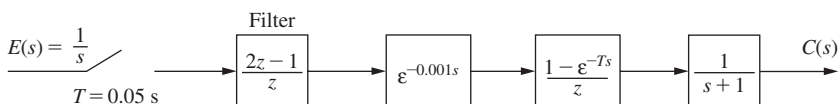
Consider the system of Fig. 4-10.

This system is that of Example 4.4, with a computational delay added for the filter. The delay is 1 ms ( $t_0 = 10^{-3}$  s) and  $T = 0.05$  s. Thus, for this system,

$$D(z) = \frac{2z - 1}{z}$$

Now

$$mT + \Delta T = T$$



**FIGURE 4-10** System for Example 4.9.

or

$$mT = T - \Delta T = 0.05 - 0.001 = 0.049$$

Then

$$G(z, m) = \mathfrak{Z}_m \left[ \frac{1 - \epsilon^{-Ts}}{s(s+1)} \right]_{mT=0.049} = \frac{z-1}{z} \mathfrak{Z}_m \left[ \frac{1}{s(s+1)} \right]_{mT=0.049}$$

From Example 4.8,

$$G(z, m) = \frac{z-1}{z} \left[ \frac{z(1 - \epsilon^{-0.049}) + (\epsilon^{-0.049} - \epsilon^{-0.05})}{(z-1)(z - \epsilon^{-0.05})} \right]$$

Since the input is a unit step, then, from (4-44),

$$\begin{aligned} C(z) &= G(z, m)D(z)E(z) \\ &= \frac{z-1}{z} \left[ \frac{z(1 - \epsilon^{-0.049}) + (\epsilon^{-0.049} - \epsilon^{-0.05})}{(z-1)(z - \epsilon^{-0.05})} \right] \frac{2z-1}{z} \left[ \frac{z}{z-1} \right] \\ &= \frac{(2z-1)[z(1 - \epsilon^{-0.049}) + (\epsilon^{-0.049} - \epsilon^{-0.05})]}{z(z-1)(z - \epsilon^{-0.05})} \end{aligned}$$

## 4.7 NONSYNCHRONOUS SAMPLING

In the preceding sections, simple open-loop systems and open-loop systems with digital filters and/or ideal time delays were considered. In this section open-loop systems with nonsynchronous sampling are analyzed. *Nonsynchronous sampling* can be defined by considering the system of Fig. 4-11. In this system both samplers operate at the same rate, but are not synchronous. We will now show that the output of this system can be derived using the modified  $z$ -transform.

To develop a method of analysis for systems with nonsynchronous sampling, consider the sampler and data hold in Fig. 4-12(a). Here the sampler operates at  $hT, T + hT, 2T + hT, 3T + hT, \dots$ , where  $0 < h < 1$ . The data-hold output is as shown in Fig. 4-12(b), and may be expressed as

$$\begin{aligned} \bar{e}(t) &= e(hT)[u(t - hT) - u(t - T - hT)] + e(T + hT)[u(t - T - hT) \\ &\quad - u(t - 2T - hT)] + e(2T + hT)[u(t - 2T - hT) \\ &\quad - u(t - 3T - hT)] + \dots \end{aligned} \quad (4-45)$$

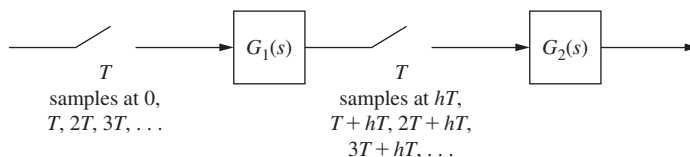
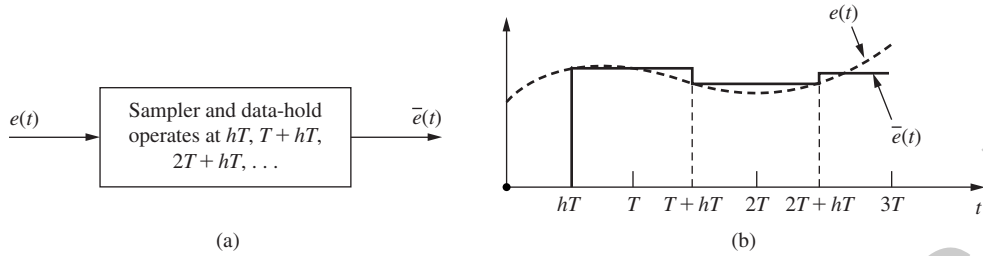


FIGURE 4-11 System with nonsynchronous sampling.



**FIGURE 4-12** Illustration of nonsynchronous sampling.

Thus

$$\begin{aligned} \bar{E}(s) = & e(hT) \left[ \frac{\epsilon^{-hTs}}{s} - \frac{\epsilon^{-(T+hT)s}}{s} \right] + e(T+hT) \left[ \frac{\epsilon^{-(T+hT)s}}{s} - \frac{\epsilon^{-(2T+hT)s}}{s} \right] \\ & + e(2T+hT) \left[ \frac{\epsilon^{-(2T+hT)s}}{s} - \frac{\epsilon^{-(3T+hT)s}}{s} \right] + \dots \end{aligned}$$

or

$$\begin{aligned} \bar{E}(s) = & \frac{1 - \epsilon^{-Ts}}{s} \epsilon^{-hTs} [e(hT) + e(T+hT)\epsilon^{-Ts} + e(2T+hT)\epsilon^{-2Ts} + \dots] \\ = & \frac{1 - \epsilon^{-Ts}}{s} \epsilon^{Ts} \epsilon^{-hTs} [e(hT)\epsilon^{-Ts} + e(T+hT)\epsilon^{-2Ts} + e(2T+hT)\epsilon^{-3Ts} + \dots] \end{aligned}$$

Then, from (4-28),

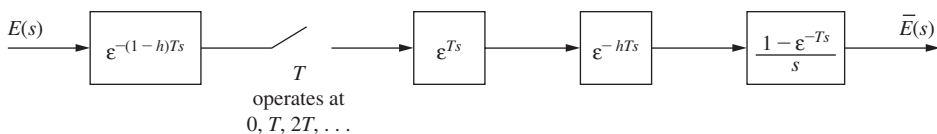
$$\bar{E}(s) = \frac{1 - \epsilon^{-Ts}}{s} \epsilon^{Ts} \epsilon^{-hTs} E(z, m) \Big|_{m=h, z=\epsilon^{Ts}} \quad (4-46)$$

Since

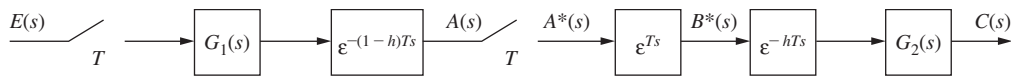
$$E(z, m) = \mathfrak{z}[E(s)\epsilon^{-\Delta Ts}] \Big|_{\Delta=1-m} \quad (4-47)$$

we see from (4-46) that the sampler and data hold of Fig. 4-12 can be modeled as shown in Fig. 4-13, where the sampler operates at  $t = 0, T, 2T, \dots$ . Note that this model delays the input signal, samples the delayed signal, and then advances the delayed sampled signal, such that the total delay in the signal is zero.

From the development above, we see that the system of Fig. 4-11, with nonsynchronous samplers, may be modeled as shown in Fig. 4-14. In Fig. 4-14, the samplers are synchronous and the system is of the form of that of Fig. 4-3(a).



**FIGURE 4-13** Model of a sampler and data hold.



Both samplers operate at  $0, T, 2T, \dots$

**FIGURE 4-14** Model of the system of Figure 4-11.

Now, in Fig. 4-14,

$$A(s) = \varepsilon^{-(1-h)Ts} G_1(s) E^*(s)$$

and thus

$$A(z) = G_1(z, m) \Big|_{m=h} E(z) \quad (4-48)$$

Also,

$$B^*(s) = \varepsilon^{Ts} A^*(s)$$

yielding

$$B(z) = zA(z) = zE(z)G_1(z, m) \Big|_{m=h} \quad (4-49)$$

Then

$$C(s) = \varepsilon^{-hTs} G_2(s) B^*(s)$$

yielding

$$C(z) = G_2(z, m) \Big|_{m=1-h} B(z) \quad (4-50)$$

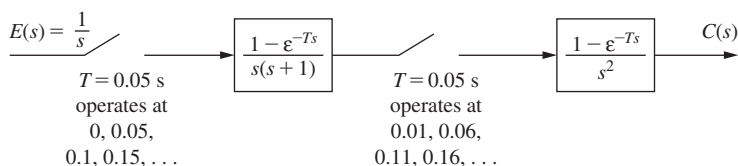
From (4-49) and (4-50), we find  $C(z)$  to be given by

$$C(z) = zE(z)G_1(z, m) \Big|_{m=h} G_2(z, m) \Big|_{m=1-h} \quad (4-51)$$

### EXAMPLE 4.10

We wish to find  $C(z)$  for the system of Fig. 4-15, which contains nonsynchronous sampling. Now

$$E(z) = \mathfrak{Z} \left[ \frac{1}{s} \right] = \frac{z}{z-1}$$



**FIGURE 4-15** System for Example 4.10.



For the system,  $T = 0.05$  and  $hT = 0.01$ . From Example 4.8,

$$G_1(z, m) = \check{z}_m \left[ \frac{1 - \epsilon^{-Ts}}{s(s + 1)} \right] = \frac{z - 1}{z} \left[ \frac{z(1 - \epsilon^{-mT}) + \epsilon^{-mT} - \epsilon^{-T}}{(z - 1)(z - \epsilon^{-T})} \right]$$

Then

$$G_1(z, m) \Big|_{mT=0.01} = \frac{z - 1}{z} \left[ \frac{z(1 - \epsilon^{-0.01}) + \epsilon^{-0.01} - \epsilon^{-0.05}}{(z - 1)(z - \epsilon^{-0.05})} \right]$$

Also, from the modified  $z$ -transform tables,

$$G_2(z, m) = \check{z}_m \left[ \frac{1 - \epsilon^{-Ts}}{s^2} \right] = \frac{z - 1}{z} \left[ \frac{mTz - mT + T}{(z - 1)^2} \right]$$

or

$$G_2(z, m) \Big|_{m=1-h} = \frac{0.04z + 0.01}{z(z - 1)}$$

Then, from the development above and (4-51),

$$\begin{aligned} C(z) &= z \left[ \frac{z}{z - 1} \right] \frac{z - 1}{z} \left[ \frac{z(1 - \epsilon^{-0.01}) + \epsilon^{-0.01} - \epsilon^{-0.05}}{(z - 1)(z - \epsilon^{-0.05})} \right] \frac{0.04z + 0.01}{z(z - 1)} \\ &= \frac{(0.04z + 0.01)[z(1 - \epsilon^{-0.01}) + \epsilon^{-0.01} - \epsilon^{-0.05}]}{(z - 1)^2(z - \epsilon^{-0.05})} \end{aligned}$$

## 4.8 STATE-VARIABLE MODELS

Thus far in this chapter we have discussed the analysis of open-loop sampled-data systems using the transfer-function approach. As was shown in Chapter 2, systems describable by a  $z$ -transform transfer function may also be modeled by discrete state equations. The state equations are of the form, from Section 2.8,

$$\begin{aligned} \mathbf{x}(k + 1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \end{aligned} \tag{4-52}$$

where  $\mathbf{x}(k)$  is the state vector,  $\mathbf{u}(k)$  is the input vector, and  $\mathbf{y}(k)$  is the output vector. The state-variable techniques used in Chapter 2 may be employed here to find the state-variable models of open-loop sampled-data systems of the type discussed in this chapter. To obtain a state-variable model:

1. Draw a simulation diagram from the  $z$ -transform transfer function.
2. Label each time-delay output as a state variable.
3. Write the state equations from the simulation diagram.

This technique will now be illustrated with an example.

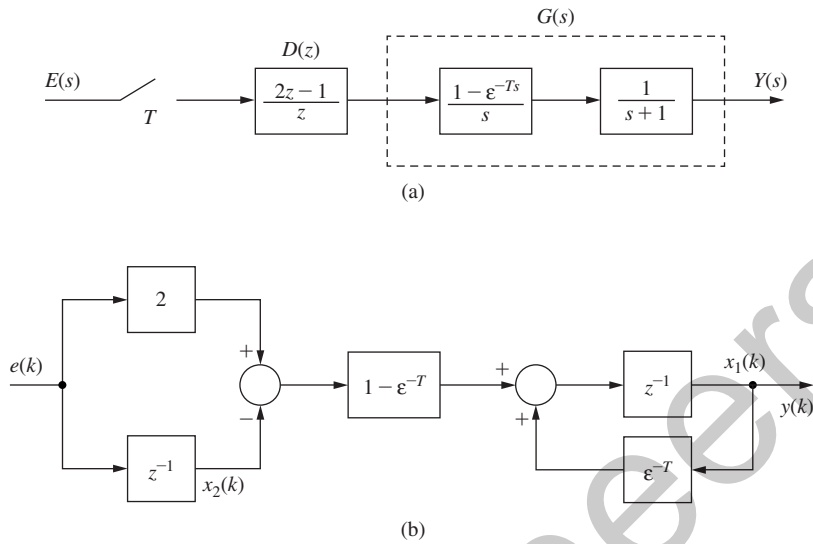


FIGURE 4-16 System for Example 4.11.

### EXAMPLE 4.11

Consider the system of Fig. 4-16(a), which is the system considered in Example 4.4.

Here we are denoting the output as  $Y(s)$  instead of  $C(s)$ , to prevent any notational confusion with the  $C$  matrix. From Example 4.4,

$$G(z) = \mathcal{Z} \left[ \frac{1 - e^{-Ts}}{s(s+1)} \right] = \frac{1 - e^{-T}}{z - e^{-T}}$$

and as is shown in Fig. 4-16(a),

$$D(z) = \frac{2z - 1}{z}$$

A simulation diagram for this system is given in Fig. 4-16(b). Next each delay output is labeled as a state variable. Then, from this Fig. we write

$$\mathbf{x}(k+1) = \begin{bmatrix} e^{-T} & -1 + e^{-T} \\ 0 & 0 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 2(1 - e^{-T}) \\ 1 \end{bmatrix} e(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k)$$

## 4.9 REVIEW OF CONTINUOUS-TIME STATE VARIABLES

Presented in the preceding section is a technique for obtaining a set of state equations describing a linear time-variant discrete-time system. This technique is based on transfer functions, and has two major disadvantages. One disadvantage can be illustrated by considering a simple mechanical system, the motion of a unit mass in a frictionless environment. For this system,

$$\frac{d^2x(t)}{dt^2} = \ddot{x}(t) = f(t) \quad (4-53)$$

where  $x(t)$  is the displacement, or position, of the mass, and  $f(t)$  is the applied force. To obtain a continuous-time state-variable model, choose as state variables

$$\begin{aligned} v_1(t) &= x(t) = \text{position} \\ v_2(t) &= \dot{v}_1(t) = \dot{x}(t) = \text{velocity} \end{aligned} \quad (4-54)$$

where  $\dot{x}(t) = dx(t)/dt$ . We denote the states of a continuous-time system as  $v_i(t)$ ,  $i = 1, 2$ .

Then the state equations are

$$\begin{bmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f(t) \quad (4-55)$$

Here we have chosen as state variables the position and the velocity of the mass. These variables may be considered to be the “natural” states (physical variables) of the system, and would be the desirable states to choose. If this simple system were a part of a sampled-data system, we can easily choose position as one of the states in the discrete state model, by letting position to be the output of the simple system. However, in taking the transfer-function approach to discrete state modeling, we would have difficulty in choosing velocity as the second state variable. Thus we lose the natural, and desirable, states of the system. Another disadvantage of the transfer-function approach is the difficulty in deriving the pulse transfer functions for high-order systems.

A different approach for obtaining the discrete-time state model of a system is presented in the following section. This approach is based on the use of continuous-time state variables. Hence a brief presentation of the requisite theory of continuous-time state variables will be made here.

As indicated in the example above of the motion of a mass, continuous-time state-variable equations for a linear-time-invariant system are of the form

$$\begin{aligned} \dot{\mathbf{v}}(t) &= \mathbf{A}_c \mathbf{v}(t) + \mathbf{B}_c \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}_c \mathbf{v}(t) + \mathbf{D}_c \mathbf{u}(t) \end{aligned} \quad (4-56)$$

In this equation,  $\mathbf{v}(t)$  are the states,  $\mathbf{u}(t)$  are the inputs,  $\mathbf{y}(t)$  are the outputs, and the matrices are subscripted to indicate continuous-time state equations. Nonsubscripted matrices will indicate discrete-time state equations. To illustrate continuous-time state equations, consider the following example.

### EXAMPLE 4.12

It is desired to find a state model for the mechanical system of Fig. 4-17(a).

Here  $M$  is mass,  $B$  is the damping factor for linear friction, and  $K$  is the stiffness factor for a linear spring. The equations for this system are [1]

$$\begin{aligned} \ddot{y}_1(t) + B_1 \dot{y}_1(t) + K y_1(t) + B_2 [\dot{y}_1(t) - \dot{y}_2(t)] &= 0 \\ \ddot{y}_2(t) + B_2 [\dot{y}_2(t) - \dot{y}_1(t)] &= u(t) \end{aligned}$$

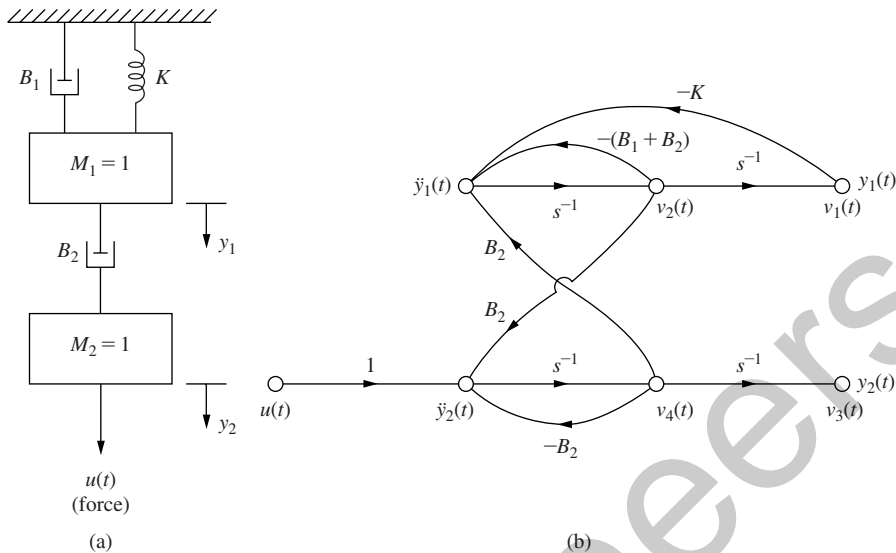


FIGURE 4-17 System for Example 4.12.

The state variables are chosen as

$$\begin{aligned} v_1(t) &= y_1(t); & v_3(t) &= y_2(t) \\ v_2(t) &= \dot{y}_1(t); & v_4(t) &= \dot{y}_2(t) \end{aligned}$$

A flow graph for these equations is shown in Fig. 4-17(b). The transfer functions  $s^{-1}$  represent integrators, and the output of each integrator is chosen as a state, as shown. Next the state equations are written from the flow graph.

$$\dot{\mathbf{v}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -K & -(B_1 + B_2) & 0 & B_2 \\ 0 & 0 & 0 & 1 \\ 0 & B_2 & 0 & -B_2 \end{bmatrix} \mathbf{v} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{u}$$

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

In Example 4.12, the state variables are positions and velocities. If the technique of drawing a flow graph from the system differential equations (written using physical laws) is used, the states chosen will be the natural states of the system.

Consider now the state equations for a single-input, single-output, continuous-time system.

$$\begin{aligned} \dot{\mathbf{v}}(t) &= \mathbf{A}_c \mathbf{v}(t) + \mathbf{B}_c u(t) \\ y(t) &= \mathbf{C}_c \mathbf{v}(t) + D_c u(t) \end{aligned} \tag{4-57}$$

To obtain the solution of these equations, we will use the Laplace transform. For (4-57) [1],

$$s\mathbf{V}(s) - \mathbf{v}(0) = \mathbf{A}_c \mathbf{V}(s) + \mathbf{B}_c U(s) \tag{4-58}$$

Solving for  $V(s)$  yields

$$V(s) = [I s - A_c]^{-1} v(0) + [I s - A_c]^{-1} B_c U(s) \quad (4-59)$$

Define  $\Phi_c(t)$  as

$$\Phi_c(t) = \mathcal{L}^{-1}\{[I s - A_c]^{-1}\} \quad (4-60)$$

The matrix  $\Phi_c(t)$  is called the state transition matrix for (4-57). The inverse Laplace transform of (4-59) is then

$$v(t) = \Phi_c(t)v(0) + \int_0^t \Phi_c(t - \tau) B_c u(\tau) d\tau \quad (4-61)$$

The state transition matrix  $\Phi_c(t)$  can be calculated as given in (4-60). However, a different expression for  $\Phi_c(t)$  may be derived. This expression is found by assuming  $\Phi_c(t)$  in (4-61) to be an infinite series.

$$\Phi_c(t) = K_0 + K_1 t + K_2 t^2 + K_3 t^3 + \dots \quad (4-62)$$

where the  $K_i$  are constant matrices. Choosing  $u(t) = 0$ , the substitution of (4-61), with  $\Phi_c(t)$  given by (4-62), into (4-57) yields [1]

$$\Phi_c(t) = I + A_c t + A_c^2 \frac{t^2}{2!} + A_c^3 \frac{t^3}{3!} + \dots = \sum_{k=0}^{\infty} \frac{A_c^k t^k}{k!} \quad (4-63)$$

This expression for  $\Phi_c(t)$  will prove to be useful in deriving discrete-time state models for sampled-data systems.

A problem that often arises in determining state models of physical systems will now be discussed. This problem is illustrated by the system shown in Fig. 4-18(a).

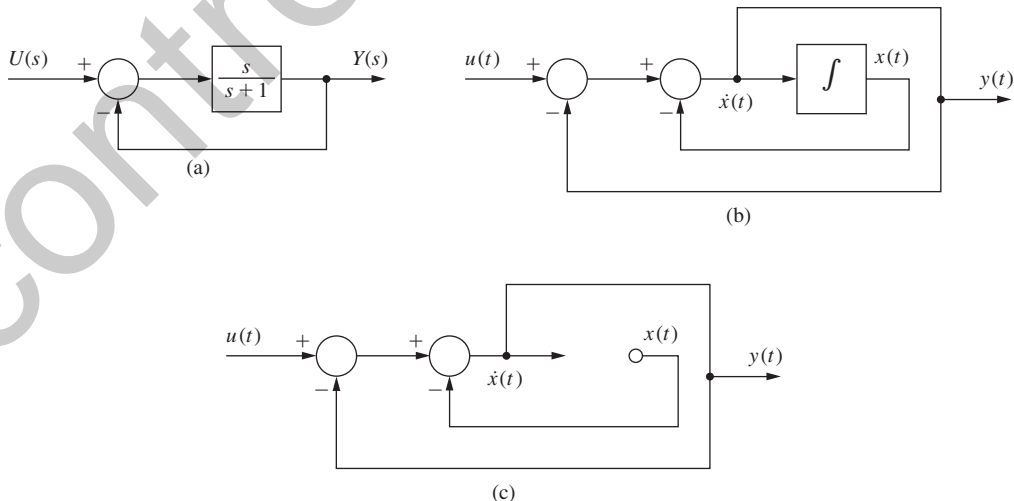


FIGURE 4-18 System with an algebraic loop.

State equations, written from the simulation diagram of Fig. 4-18(b), are

$$\dot{x}(t) = -x(t) + [u(t) - \dot{x}(t)]$$

$$y(t) = \dot{x}(t)$$

These equations are not in the standard format for state equations, since  $\dot{x}(t)$  appears on the right-hand side of both the  $\dot{x}(t)$  equation and the  $y(t)$  equation. The  $\dot{x}(t)$  equation must be solved for  $\dot{x}(t)$ , resulting in

$$\dot{x}(t) = -0.5x(t) + 0.5u(t)$$

and hence

$$y(t) = -0.5x(t) + 0.5u(t)$$

We now present a procedure for writing the state equations in the standard format for systems of the type shown in Fig. 4-18. If we redraw the simulation diagram of Fig. 4-18(b) with the integrator omitted, Fig. 4-18(c) results. Now we write the equations for  $\dot{x}(t)$  and  $y(t)$  in terms of the inputs in this diagram,  $x(t)$  and  $u(t)$ . This technique is standard for writing state equations from simulation diagrams, although we usually do not redraw the diagram with the integrators removed. Applying Mason's gain formula to the diagram in Fig. 4-18(c), we obtain the state equations

$$\dot{x}(t) = \frac{-1}{1+1}x(t) + \frac{1}{1+1}u(t) = -0.5x(t) + 0.5u(t)$$

$$y(t) = \frac{-1}{1+1}x(t) + \frac{1}{1+1}u(t) = -0.5x(t) + 0.5u(t)$$

which is the desired result.

The loop in Fig. 4-18(c) that does not contain an integrator is called an *algebraic loop*. We now present a second procedure for writing the state equations for systems with algebraic loops. The system equations are initially written as [2]

$$\dot{\mathbf{x}}(t) = \mathbf{A}_1\dot{\mathbf{x}}(t) + \mathbf{A}_2\mathbf{x}(t) + \mathbf{B}_1\mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}_1\dot{\mathbf{x}}(t) + \mathbf{C}_2\mathbf{x}(t) + \mathbf{D}_1\mathbf{u}(t)$$

Solving the first equation, we obtain

$$\dot{\mathbf{x}}(t) = [\mathbf{I} - \mathbf{A}_1]^{-1}\mathbf{A}_2\mathbf{x}(t) + [\mathbf{I} - \mathbf{A}_1]^{-1}\mathbf{B}_1\mathbf{u}(t)$$

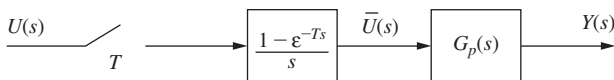
Then we substitute this equation into the one for  $y(t)$ :

$$\mathbf{y}(t) = [\mathbf{C}_1[\mathbf{I} - \mathbf{A}_1]^{-1}\mathbf{A}_2 + \mathbf{C}_2]\mathbf{x}(t) + [\mathbf{C}_1[\mathbf{I} - \mathbf{A}_1]^{-1}\mathbf{B}_1 + \mathbf{D}_1]\mathbf{u}(t)$$

Thus we have the state equations for the system in standard form.

## 4.10 DISCRETE-TIME STATE EQUATIONS

A technique is developed in this section for determining the discrete state equations of a sampled-data system directly from the continuous-time state equations. In fact, the states of the continuous-time model become the states of the discrete model. Thus the natural states of the system are preserved.



**FIGURE 4-19** Sampled-data system.

To develop this technique, consider the state equations for the continuous-time portion of the system shown in Fig. 4-19.

$$\begin{aligned}\dot{\mathbf{v}}(t) &= \mathbf{A}_c \mathbf{v}(t) + \mathbf{B}_c u(t) \\ y(t) &= \mathbf{C}_c \mathbf{v}(t) + D_c u(t)\end{aligned}\quad (4-64)$$

As shown in Section 4.9, the solution to these equations is

$$\mathbf{v}(t) = \Phi_c(t - t_0) \mathbf{v}(t_0) + \int_{t_0}^t \Phi_c(t - \tau) \mathbf{B}_c u(\tau) d\tau \quad (4-65)$$

where the initial time is  $t_0$ , and where, from (4-63),

$$\Phi_c(t - t_0) = \sum_{k=0}^{\infty} \frac{\mathbf{A}_c^k (t - t_0)^k}{k!} \quad (4-66)$$

To obtain the discrete model we evaluate (4-65) at  $t = kT + T$  with  $t_0 = kT$ , that is,

$$\mathbf{v}(kT + T) = \Phi_c(T) \mathbf{v}(kT) + u(kT) \int_{kT}^{kT+T} \Phi_c(kT + T - \tau) \mathbf{B}_c d\tau \quad (4-67)$$

Note that we have replaced  $u(t)$  with  $u(kT)$  since, during the time interval  $kT \leq t < kT + T$ ,  $u(t) = u(kT)$ . It is emphasized that this development is valid only if  $u(t)$  is the output of a zero-order hold.

Compare (4-67) with the discrete state equations [see (4-52)]

$$\begin{aligned}\mathbf{x}(k + 1) &= \mathbf{A} \mathbf{x}(k) + \mathbf{B} u(k) \\ y(k) &= \mathbf{C} \mathbf{x}(k) + D u(k)\end{aligned}\quad (4-68)$$

Thus, if we let

$$\begin{aligned}\mathbf{x}(kT) &= \mathbf{v}(kT) \\ \mathbf{A} &= \Phi_c(T) \\ \mathbf{B} &= \int_{kT}^{kT+T} \Phi_c(kT + T - \tau) \mathbf{B}_c d\tau\end{aligned}\quad (4-69)$$

we obtain the discrete state equations for the sampled-data system. Hence the discrete-system  $\mathbf{A}$  and  $\mathbf{B}$  matrices are given by (4-69).

The output equation in (4-64), when evaluated at  $t = kT$ , yields

$$\begin{aligned} y(kT) &= \mathbf{C}_c \mathbf{v}(kT) + D_c u(kT) \\ &= \mathbf{C}_c \mathbf{x}(kT) + D_c u(kT) \end{aligned} \quad (4-70)$$

Thus the discrete  $\mathbf{C}$  and  $D$  matrices are equal to the continuous-time  $\mathbf{C}_c$  and  $D_c$  matrices, respectively.

The relationship for  $\mathbf{B}$  can be simplified. In (4-69) in the equation for  $\mathbf{B}$ , let  $kT - \tau = -\sigma$ . Then this equation becomes

$$\mathbf{B} = \left[ \int_0^T \Phi_c(T - \sigma) d\sigma \right] \mathbf{B}_c \quad (4-71)$$

The discrete system matrices  $\mathbf{A}$  and  $\mathbf{B}$  may be evaluated by finding  $\Phi_c(T)$  using the Laplace transform approach of (4-60). However, in general this approach is cumbersome. A more tractable technique is to use a computer evaluation of  $\Phi_c(T)$  in (4-63); that is, with  $t = T$ , (4-66) and (4-63) becomes

$$\Phi_c(T) = \mathbf{I} + \mathbf{A}_c T + \mathbf{A}_c^2 \frac{T^2}{2!} + \mathbf{A}_c^3 \frac{T^3}{3!} + \cdots \quad (4-72)$$

Since this is a convergent series, the series can usually be truncated with adequate resulting accuracy.

The integral of (4-71), necessary for the computation of  $\mathbf{B}$ , can also be evaluated using a series expansion. In (4-71), let  $\tau = T - \sigma$ . Hence

$$\begin{aligned} \int_0^T \Phi_c(T - \sigma) d\sigma &= \int_T^0 \Phi_c(\tau)(-d\tau) = \int_0^T \Phi_c(\tau) d\tau \\ &= \int_0^T \left( \mathbf{I} + \mathbf{A}_c \tau + \mathbf{A}_c^2 \frac{\tau^2}{2!} + \mathbf{A}_c^3 \frac{\tau^3}{3!} + \cdots \right) d\tau \\ &= \mathbf{I}T + \mathbf{A}_c \frac{T^2}{2!} + \mathbf{A}_c^2 \frac{T^3}{3!} + \mathbf{A}_c^3 \frac{T^4}{4!} + \cdots \end{aligned} \quad (4-73)$$

and, from (4-71) and (4-73),

$$\mathbf{B} = \left[ \mathbf{I}T + \mathbf{A}_c \frac{T^2}{2!} + \mathbf{A}_c^2 \frac{T^3}{3!} + \cdots \right] \mathbf{B}_c \quad (4-74)$$

Comparing (4-72) with (4-73), we see that the computer program used to evaluate  $\Phi_c(T)$  may also be used to evaluate  $\int_0^T \Phi_c(\tau) d\tau$ , by expanding the program somewhat.

In the derivations above, only the single-input, single-output case was considered. For the multiple-input, multiple-output case, the results are the same, with  $\mathbf{D}_c$  and  $\mathbf{D}$  now matrices. For the multiple-input case, each input must be the output of a zero-order hold. In certain practical cases in which some of the inputs are not outputs of zero-order holds, the procedure above is



still used. The resulting discrete model is reasonably accurate, provided that these inputs change slowly over any sample period.

The derivations above will now be illustrated by an example.

### EXAMPLE 4.13

For the sampled-data system of Fig. 4-19, let  $T = 0.1$  s and

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

For example, this plant could be a servomotor of the type described in Chapter 1. A continuous-time state-variable model of this system is from Fig. 4-20(a),

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 10 \end{bmatrix} \mathbf{u}(t) \\ y(t) &= [1 \quad 0] \mathbf{x}(t)\end{aligned}$$

For this example, since the system is second order,  $\Phi_c(t)$  will be found.

$$\begin{aligned}\Phi_c(t) &= \mathcal{L}^{-1}[(s\mathbf{I} - \mathbf{A}_c)^{-1}] \\ &= \mathcal{L}^{-1} \begin{bmatrix} s & -1 \\ 0 & s+1 \end{bmatrix}^{-1} = \mathcal{L}^{-1} \begin{bmatrix} \frac{1}{s} & \frac{1}{s(s+1)} \\ 0 & \frac{1}{s+1} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 - e^{-t} \\ 0 & e^{-t} \end{bmatrix}\end{aligned}$$

Also,

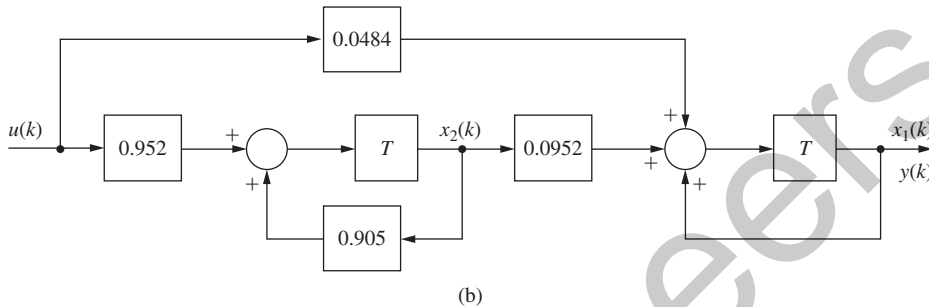
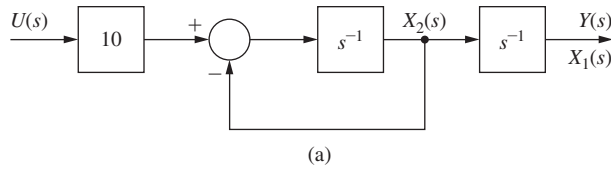
$$\int_0^T \Phi_c(\tau) d\tau = \begin{bmatrix} \tau & \tau + e^{-\tau} \\ 0 & -e^{-\tau} \end{bmatrix}_0^T = \begin{bmatrix} T & T - 1 + e^{-T} \\ 0 & 1 - e^{-T} \end{bmatrix}$$

Then

$$\mathbf{A} = \Phi_c(T)|_{T=0.1} = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix}$$

and

$$\begin{aligned}\mathbf{B} &= \left[ \int_0^T \Phi_c(\tau) d\tau \right] \mathbf{B}_c = \begin{bmatrix} 0.1 & 0.00484 \\ 0 & 0.0952 \end{bmatrix} \begin{bmatrix} 0 \\ 10 \end{bmatrix} \\ &= \begin{bmatrix} 0.0484 \\ 0.952 \end{bmatrix}\end{aligned}$$



**FIGURE 4-20** Simulation diagram for the system of Example 4.13.

Hence the discrete state equations are

$$\mathbf{x}(k + 1) = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.0484 \\ 0.952 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k)$$

A simulation diagram of this model is shown in Fig. 4-20(b). Fig. 4-20(a) is the simulation diagram of the analog plant. Even though the states, the input, and the output of the two diagrams in Fig. 4-20 are equal at the sampling instants, the two diagrams bear no resemblance to each other. In general, the two simulation diagrams for such a system are not similar.

The discrete matrices in the example above may also be calculated by computer. For three-significant-Fig. accuracy, three terms in the series expansion of  $\Phi_c(t)$  [see (4-72)] are required. Five terms in the series expansion yield six-significant-Fig. accuracy.

## 4.11 PRACTICAL CALCULATIONS

As stated in the preceding section, all calculations required to develop the discrete model of an analog plant may be performed by computer. Calculation by computer is required for high-order systems, and is preferred for low-order systems to reduce errors.

We will now list the steps necessary for the calculations.

1. Derive the state model for the analog part of the system, in the form

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c \mathbf{x}(t) + \mathbf{B}_c u(t)$$

$$y(t) = \mathbf{C}_c \mathbf{x}(t) + D_c u(t) \tag{4-75}$$

2. If the transfer function of the analog part of the system is required, calculate [1, 3]

$$G_p(s) = \mathbf{C}_c[s\mathbf{I} - \mathbf{A}_c]^{-1}\mathbf{B}_c + D_c \quad (4-76)$$

3. Calculate the discrete matrices of the analog part of the system [Eq. (4-74)]

$$\begin{aligned} \mathbf{A} &= \mathbf{I} + \mathbf{A}_cT + \mathbf{A}_c^2T^2/2 + \dots \\ \mathbf{B} &= (\mathbf{IT} + \mathbf{A}_cT^2/2! + \mathbf{A}_c^2T^3/3! + \dots)\mathbf{B}_c \\ \mathbf{C} &= \mathbf{C}_c, \quad D = D_c \end{aligned} \quad (4-77)$$

4. Calculate the pulse transfer function [see (2-84)] using [3]

$$G(z) = \mathbf{C}[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B} + D \quad (4-78)$$

The calculations required in steps 2, 3, and 4 may be implemented in MATLAB by the statements

2. `[numc,denc] = ss2tf(Ac,Bc,Cc,Dc)`
3. `[A,B] = c2d(Ac,Bc,T)`
4. `[numz,denz] = ss2tf(A,B,C,D)`

In these statements, numc are the numerator-polynomial coefficients of  $G_p(s)$ , and denc are those of the denominator. Defined in a like manner are numz and denz for  $G(z)$ .

Note that these procedures give the analog transfer function, the discrete state model, and the pulse transfer function. We do not directly use the  $z$ -transform tables in any of the steps; all steps are performed on the computer. However, we must, by some procedure, derive the analog state model. An example is now given to illustrate these procedures.

#### EXAMPLE 4.14

Consider again the system of Example 4.13. A MATLAB program that calculates the discrete state matrices  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $D$ , and the plant transfer function  $G(z) = \text{numz}/\text{denz}$  is given by

```
Ac = [0 1; 0 -1];
Bc = [0; 10];
C = [1 0];
D = 0;
T = 0.1;
[A,B] = c2d(Ac,Bc,T)
[numz,denz] = ss2tf(A,B,C,D)
Gz = tf(numz,denz,T)
```

For certain systems, the direct programming of (4-74) and (4-77) yields unacceptable numerical errors. For these cases the discrete  $\mathbf{A}$  and  $\mathbf{B}$  matrices must be evaluated using different algorithms [4–6].

## 4.12 SUMMARY

In this chapter we have examined various aspects of open-loop discrete-time systems. In particular we discussed the starred transform and showed that it possesses the properties of the  $z$ -transform defined in Chapter 2. Next, the starred transform was used to find the pulse transfer function of open-loop systems. The pulse transfer function was extended to the analysis of open-loop systems containing digital filters. In order to analyze systems containing ideal time delays, the modified  $z$ -transform and its properties were derived. Then techniques for finding discrete state-variable models of open-loop sampled-data systems were presented. These developments form the basis for the computer calculations of discrete state models and the pulse transfer function. The foundation built in this chapter for open-loop systems will serve as a basis for presenting the analysis of closed-loop discrete-time systems in Chapter 5.

## References and Further Readings

- [1] C. L. Phillips and J. Parr, *Feedback Control Systems*, 5th ed. Upper Saddle River, NJ: Prentice-Hall, 2011.
- [2] D. M. Look, "Direct State Space Formulation of Second-Order Coupled Systems," M.S. thesis, Auburn University, Auburn, AL, 1971.
- [3] J. L. Melsa and S. K. Jones, *Computer Programs for Computational Assistance*. New York: McGraw-Hill Book Company, 1973.
- [4] R. C. Ward, "Numerical Computation of the Matrix Exponential with Accuracy Estimate," *SIAM J. Numer. Anal.*, pp. 600–610, 1977.
- [5] C. Moler and C. Van Loam, "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Rev.*, pp. 801–836, 1978.
- [6] D. Westreich, "A Practical Method for Computing the Exponential of a Matrix and Its Integral," *Commun. Appl. Numer. Methods*, pp. 375–380, 1990.
- [7] J. A. Cadzow and H. R. Martens, *Discrete-Time and Computer Control Systems*. Upper Saddle River, NJ: Prentice-Hall, 1970.
- [8] P. M. De Russo, R. J. Roy, C. M. Close, and A. A. Desrochers, *State Variables for Engineers*, 2nd ed. New York: John Wiley & Sons, Inc., 1998.
- [9] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Half Moon Bay, CA: Ellis-Kagle Press, 2006.
- [10] E. I. Jury, *Theory and Application of the  $z$ -Transform Method*. Huntington, NY: R.E. Krieger Publishing Co., Inc., 1973.
- [11] B. C. Kuo, *Digital Control Systems*, 2nd ed. New York: Oxford University Press, 1995.
- [12] K. Ogata, *Discrete-Time Control Systems*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1995.

## Problems

- 4.2-1. (a) Show that a pole of  $E(s)$  in the left half-plane transforms into a pole of  $E(z)$  inside the unit circle.  
 (b) Show that a pole of  $E(s)$  on the imaginary axis transforms into a pole of  $E(z)$  on the unit circle.  
 (c) Show that a pole of  $E(s)$  in the right half-plane transforms into a pole of  $E(z)$  outside the unit circle.
- 4.2-2. Let  $T = 0.08$  s and

$$E(s) = \frac{s + 4}{(s + 2)(s - 2)}$$

- (a) Without calculating  $E(z)$ , find its poles.
- (b) Give the rule that you used in part (a).
- (c) Verify the results of part (a) by calculating  $E(z)$ .
- (d) Compare the zero of  $E(z)$  with that of  $E(s)$ .
- (e) The poles of  $E(z)$  are determined by those of  $E(s)$ . Does an equivalent rule exist for zeros?

**4.2-3.** Find the  $z$ -transform of the following functions, using  $z$ -transform tables. Compare the pole-zero locations of  $E(z)$  in the  $z$ -plane with those of  $E(s)$  and  $E^*(s)$  in the  $s$ -plane (see Problems 3.4-1). Let  $T = 0.5$  s.

(a)  $E(s) = \frac{10}{(s+1)(s+3)}$                       (b)  $E(s) = \frac{s+7}{s(s+3)}$

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

(e)  $E(s) = \frac{s^2+3s+2}{s(s+1)(s+2)}$                       (f)  $E(s) = \frac{2}{s^2+2s+5}$

(g) Verify any partial-fraction expansions required, using MATLAB.

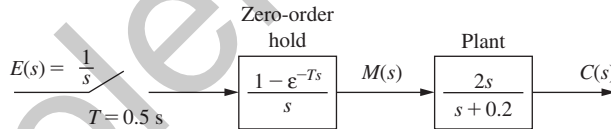
**4.2-4.** Repeat Problem 4.2-3 with  $T = 0.01$  s.

**4.3-1.** Find the  $z$ -transforms of the following functions:

(a)  $E(s) = \frac{\epsilon^3 - 1}{\epsilon^3 s(s+1)}$ ,  $T = 0.2$  s

(b)  $E(s) = \frac{(0.8s+1)(1-\epsilon^{0.5s})}{0.8s(s+0.5)}$ ,  $T = 0.5$  s

- 4.3-2.** (a) Find the system response at the sampling instants to a unit step input for the system of Fig. P4.3-2. Plot  $c(nT)$  versus time.  
 (b) Verify your results of (a) by determining the input to the plant,  $m(t)$ , and then calculating  $c(t)$  by continuous-time techniques.  
 (c) Find the steady-state gain for a constant input (dc gain), from both the pulse transfer function and from the plant transfer function.  
 (d) Is the gain in part (c) obvious from the results of parts (a) and (b)? Why?



**FIGURE P4.3-2** System for Problem 4.3-2.

**4.3-3.** Repeat Problem 4.3-2 for the case that  $T = 0.2$  s and the plant transfer function is given by:

(a)  $G_p(s) = \frac{8}{s^2+3s+2}$

(b)  $G_p(s) = \frac{5}{s^2+4s+4}$

Verify each  $G(z)$  by computer.

**4.3-4.** (a) Find the conditions on a transfer function  $G(z)$  such that its dc gain is zero. Prove your result.

(b) For

$$G(z) = \mathcal{Z} \left[ \frac{1 - \epsilon^{-Ts}}{s} G_p(s) \right]$$

find the conditions of  $G_p(s)$  such that the dc gain of  $G(z)$  is zero. Prove your result.

- (c) Normally, a pole at the origin in the  $s$ -plane transforms into a pole at  $z = 1$  in the  $z$ -plane. The function in the brackets in part (b) has a pole at the origin in the  $s$ -plane. Why does this pole *not* transform into a pole at  $z = 1$ ?

- (d) Find the conditions on  $G(z)$  such that its dc gain is unbounded. Prove your result.  
 (e) For  $G(z)$  as given in part (b), find the conditions of  $G_p(s)$  such that the dc gain of  $G(z)$  is unbounded. Prove your result.

4.3-5. Find the system response at the sampling instants to a unit-step input for the system of Fig. P4.3-5.

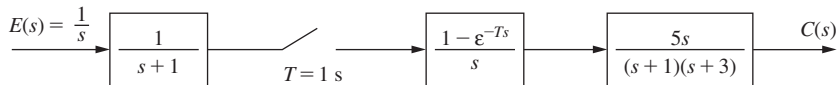


FIG. P4.3-5 System for Problem 4.3-5.

- 4.3-6. (a) Find the output  $c(kT)$  for the system of Fig. P4.3-6, for  $e(t)$  equal to a unit-step function.  
 (b) What is the effect on  $c(kT)$  of the sampler and data hold in the upper path? Why?  
 (c) Sketch the unit-step response  $c(t)$  of the system of Fig. P4.3-6. This sketch can be made without mathematically solving for  $C(s)$ .  
 (d) Repeat part (c) for the case that the sampler and data hold in the upper path is removed.

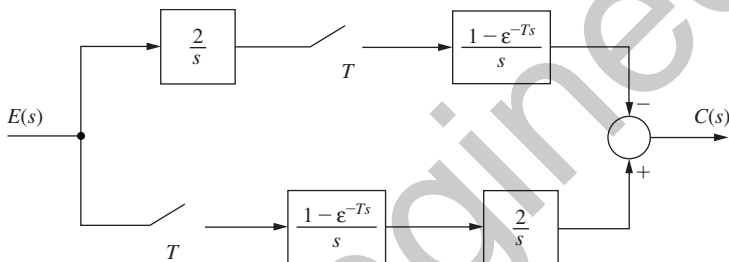


FIG. P4.3-6 System for Problem 4.3-6.

- 4.3-7. (a) Express each  $C(s)$  and  $C(z)$  as functions of the input for the systems of Fig. P4.3-7.  
 (b) List those transfer functions in Fig. P4.3-7 that contain the transfer function of a data hold.

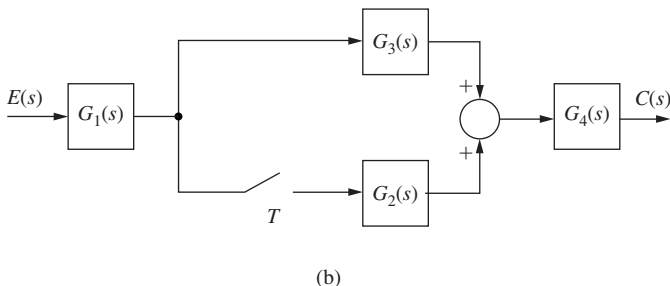
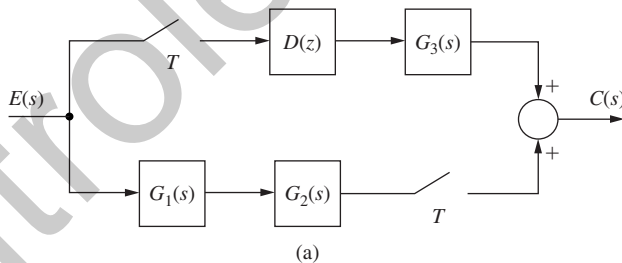
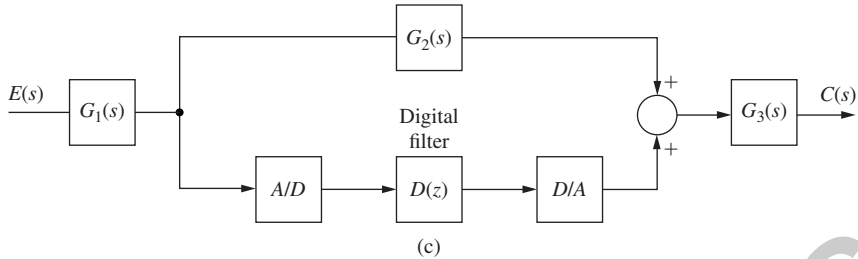


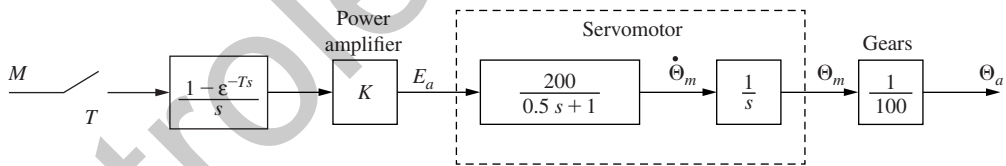
FIGURE P4.3-7 Systems for Problem 4.3-7.



**FIGURE P4.3-7** (continued)

**4.3-8.** Shown in Fig. P4.3-8 is the block diagram of one joint of a robot arm. This system is described in Problem 1.5-4. The signal  $M(s)$  is the sampler input,  $E_a(s)$  is the servomotor input voltage,  $\Theta_m(s)$  is the motor shaft angle, and the output  $\Theta_a(s)$  is the angle of the arm.

- Suppose that the sampling-and-data-reconstruction process is implemented with an analog-to-digital converter (A/D) and a digital-to-analog converter. Redraw Fig. P4.3-8 showing the A/D and the D/A.
- Suppose that the units of  $e_a(t)$  are volts, and of  $\Theta_m(t)$  are rpm. The servomotor is rated at 24 V (the voltage  $e_a(t)$  should be less than or equal to 24 V in magnitude). Commercially available D/As are usually rated with output voltage ranges of  $\pm 5$  V,  $\pm 10$  V, 0 to 5 V, 0 to 10 V, or 0 to 20 V. If the gain of the power amplifier is 2.4, what should be the rated voltage of the D/A? Why?
- Derive the analog transfer function  $\Theta_a(s)/E_a(s)$ .
- With  $K = 2.4$  and  $T = 0.1$  s, derive the pulse transfer function  $\Theta_a(z)/M(z)$ .
- Derive the steady-state output for  $m(t)$  constant. Justify this value from the motor characteristics.
- Verify the results of part (d) by computer.

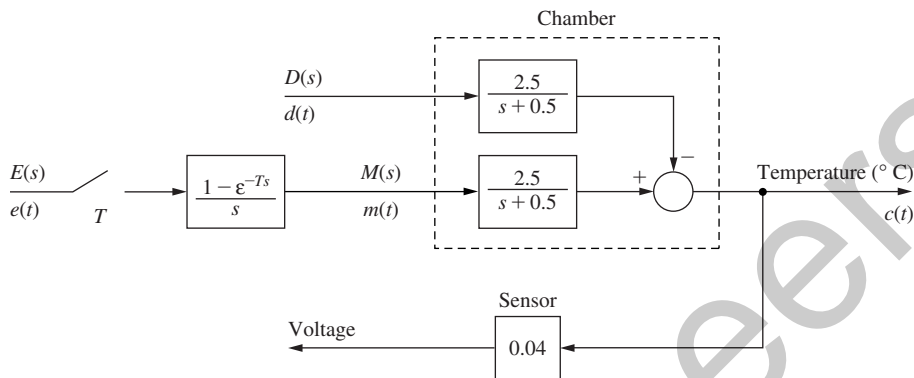


**FIG. P4.3-8** Model of robot arm joint.

**4.3-9** Fig. P4.3-9 illustrates a thermal stress chamber. This system is described in Problem 1.6-1. The system output  $c(t)$  is the chamber temperature in degrees Celsius, and the control-voltage input  $m(t)$  operates a valve on a steam line. The sensor is based on a thermistor, which is a temperature-sensitive resistor. The disturbance input  $d(t)$  models the opening of the door into the chamber. With the door closed,  $d(t) = 0$ ; if the door is opened at  $t = t_0$ ,  $d(t) = u(t - t_0)$ , a unit-step function.

- Suppose that the sampling-and-data-reconstruction process is implemented with an analog-to-digital converter (A/D) and a digital-to-analog converter (D/A). Redraw Fig. P4.3-9 showing the A/D and the D/A.
- Derive the transfer function  $C(z)/E(z)$ .
- A constant voltage of  $e(t) = 10$  V is applied for a long period. Find the steady-state temperature of the chamber, with the door closed. Note that this problem can be solved without knowing the sample period  $T$ .

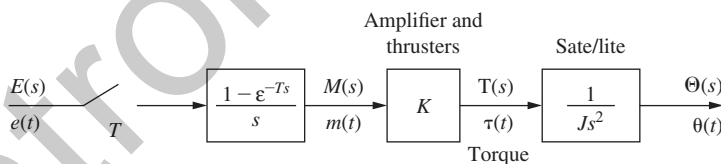
- (d) Find the steady-state effect on the chamber temperature of leaving the door open.
- (e) Find the expression for  $C(s)$  as a function of the Laplace-transform variable  $s$ , the control input, and the disturbance input. The  $z$ -transform variable  $z$  cannot appear in this expression.



**FIGURE P4.3-9** Block diagram for a thermal test chamber.

**4.3-10.** Given in Fig. P4.3-10 is the block diagram of a rigid-body satellite. The control signal is the voltage  $e(t)$ . The zero-order hold output  $m(t)$  is converted into a torque  $\tau(t)$  by an amplifier and the thrusters (see Section 1.4). The system output is the attitude angle  $\theta(t)$  of the satellite.

- (a) Find the transfer function  $\Theta(z)/E(z)$ .
- (b) Use the results of part (a) to find the system's unit-step response, that is, the response with  $e(t) = u(t)$ .
- (c) Sketch the zero-order-hold output  $m(t)$  in (b).
- (d) Use  $m(t)$  in part (c) to find  $c(t) = \mathcal{L}^{-1}[KM(s)/Js^2]$ .
- (e) In part (d), evaluate  $c(kT)$ . This response should equal that found in part (b).

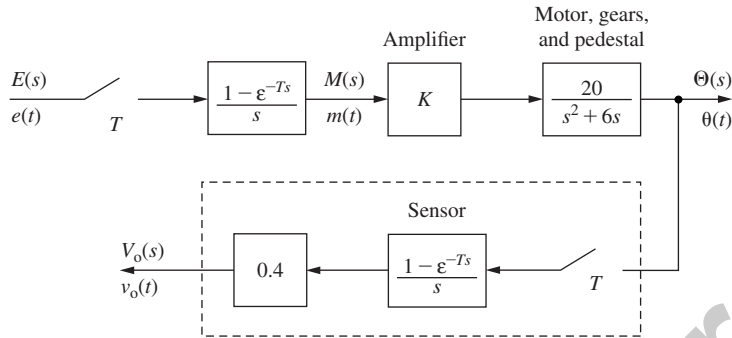


**FIG. P4.3-10** Block diagram for a satellite.

**4.3-11.** The antenna positioning system described in Section 1.5 and Problem 1.5-1 is depicted in Fig. P4.3-11. In this problem we consider the yaw angle control system, where  $\theta(t)$  is the yaw angle. The angle sensor (a digital shaft encoder and the data hold) yields  $v_o(kT) = [0.4 \theta(kT)]$ , where the units of  $v_o(t)$  are volts and  $\theta(t)$  are degrees. The sample period is  $T = 0.05$  s.

- (a) Find the transfer function  $\Theta(z)/E(z)$ .
- (b) The yaw angle is initially zero. The input voltage  $e(t)$  is set equal to 10 V at  $t = 0$ , and is zero at each sample period thereafter. Find the steady-state value of the yaw angle.
- (c) Note that in part (b), the coefficients in the partial-fraction expansion add to zero. Why does this occur?
- (d) The input voltage  $e(t)$  is set to a constant value. Without solving mathematically, give a description of the system response.



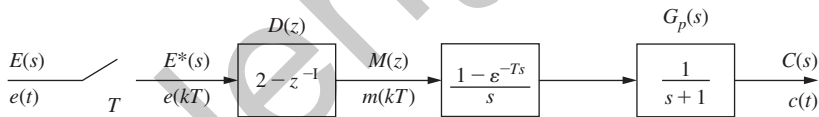


**FIGURE P4.3-11** Block diagram for an antenna control system.

(e) Suppose in part (d) that you are observing the antenna. Describe what you would see.

**4.4-1.** Example 4.3 calculates the step response of the system in Fig. 4-2. Example 4.4 calculates the step response of the same system preceded by a digital filter with the transfer function  $D(z) = (2 - z^{-1})$ . This system is shown in Fig. P4.4-1.

- Solve for the output of the digital filter  $m(kT)$ .
- Let the response in Example 4.3 be denoted as  $c_1(kT)$ . Use the results in part (a) to express the output  $c(kT)$  in Fig. P4.4-1 as a function of  $c_1(kT)$ .
- Use the response  $c_1(kT)$  calculated in Example 4.3 and the results in part (b) to find the output  $c(kT)$  in Fig. P4.4-1. This result should be the same as in Example 4.4.
- Use the response  $C_1(z)$  calculated in Example 4.3 and the result in part (b) to find the output  $C(z)$  in Fig. P4.4-1. This result should be the same as in Example 4.4.



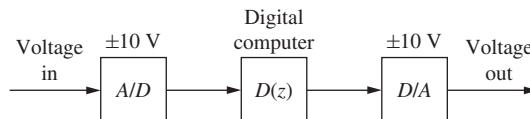
**FIG. P4.4-1** System for Problem 4.4-1.

**4.4-2.** Consider the *hardware* depicted in Fig. P4.4-2. The transfer function of the digital controller implemented in the computer is given by

$$D(z) = \frac{4.5(z - 0.90)}{z - 0.85}$$

The input voltage rating of the analog-to-digital converter is  $\pm 10$  V and the output voltage rating of the digital-to-analog converter is also  $\pm 10$  V.

- Calculate the dc gain of the controller.
- State exactly how you would verify, using the hardware, the value calculated in part (a). Give the equipment required, the required settings on the equipment, and the expected measurements.



**FIGURE P4.4-2** Hardware configuration for Problem 4.4-2.

4.4-3. For the system of Fig. P4.4-3, the filter solves the difference equation

$$m(k) = 0.8m(k - 1) + 0.1e(k)$$

The sampling rate is 1 Hz and the plant transfer function is given by

$$G_p(s) = \frac{1}{s + 0.2}$$

- Find the system transfer function  $C(z)/E(z)$ .
- Find the system dc gain from the results of part (a).
- Verify the results of part (b) by finding the dc gain of the filter using  $D(z)$  and that of the plant using  $G_p(s)$ .
- Use the results of part (b) to find the steady-state value of the unit-step response.
- Verify the results of part (d) by calculating  $c(kT)$  for a unit-step input.
- Note that in part (e), the coefficients in the partial-fraction expansion add to zero. Why does this occur?

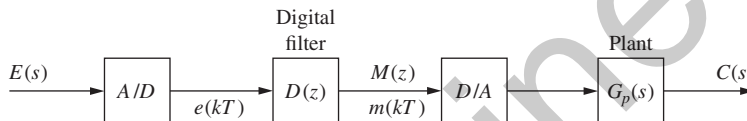


FIGURE P4.4-3 System for Problems 4.4-3 and 4.4-4.

4.4-4. Repeat Problem 4.4-3 for the case that the filter solves the difference equation

$$m(k + 1) = 0.5e(k + 1) - (0.5)(0.8)e(k) + 0.485m(k),$$

the sampling rate is 10 Hz, and the plant transfer function is given by

$$G_p(s) = \frac{5}{(s + 1)(s + 2)}$$

4.4-5. Consider the system of Fig. P4.4-5. The filter transfer function is  $D(z)$ .

- Express  $C(z)$  as a function of  $E$ .
- A discrete state model of this system does not exist. Why?
- What assumptions concerning  $e(t)$  must be made in order to derive an approximate discrete state model?



FIGURE P4.4-5 System for Problem 4.4-5.

4.4-6. Consider again the system of Fig. P4.4-5. Add a sampler for  $E(s)$  at the input. Given

$$G_1(s) = \frac{1}{s + 10} \quad D(z) = \frac{z - 0.5}{z - 1} \quad G_2(s) = \frac{s}{s^2 + 9s + 23}$$

find  $c(kT)$  for a unit-step input with  $T = 0.5$  s. Is this a good choice for the sampling interval? If not, what sampling rate would you recommend. Find  $c(kT)$  at your recommended sampling rate and plot both step responses using MATLAB.

**4.5-1.** Find the modified  $z$ -transform of the following functions:

$$(a) E(s) = \frac{20}{(s+2)(s+5)}$$

$$(b) E(s) = \frac{5}{s(s+1)}$$

$$(c) E(s) = \frac{s+2}{s(s+1)}$$

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

$$(e) E(s) = \frac{s^2+5s+6}{s(s+4)(s+5)}$$

$$(f) E(s) = \frac{2}{s^2+2s+5}$$

**4.5-2.** Find the  $z$ -transform of the following functions. The results of Problem 4.5-1 may be useful.

$$(a) E(s) = \frac{20\epsilon^{-0.3Ts}}{(s+2)(s+5)}$$

$$(b) E(s) = \frac{5\epsilon^{-0.6Ts}}{s(s+1)}$$

$$(c) E(s) = \frac{s+2\epsilon^{-1.1Ts}}{s(s+1)}$$

$$(d) E(s) = \frac{s+2\epsilon^{-0.2Ts}}{s^2(s+1)}$$

$$(e) E(s) = \frac{(s^2+5s+6)\epsilon^{-0.3Ts}}{s(s+4)(s+5)}$$

$$(f) E(s) = \frac{2\epsilon^{-0.75Ts}}{s^2+2s+5}$$

**4.5-3.** Find the modified  $z$ -transform of the following functions:

$$(a) E(s) = \frac{6}{(s+1)(s+2)(s+3)}$$

$$(b) E(s) = \frac{4}{s(s+2)^2}$$

$$(c) E(s) = \frac{s^2+2s+2}{s(s+2)^2}$$

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

$$(e) E(s) = \frac{s^2+2s+3}{s(s+2)^2(s+4)}$$

$$(f) E(s) = \frac{2s+7}{(s^2+2s+5)(s+3)}$$

**4.5-4.** Find the  $z$ -transform of the following functions. The results of Problem 4.5-3 may be useful.

$$(a) E(s) = \frac{6\epsilon^{-0.3Ts}}{(s+1)(s+2)(s+3)}$$

$$(b) E(s) = \frac{4\epsilon^{-0.6Ts}}{s(s+2)^2}$$

$$(c) E(s) = \frac{s^2+2s+2\epsilon^{-1.1Ts}}{s(s+2)^2}$$

$$(d) E(s) = \frac{(s+2)^2\epsilon^{-0.2Ts}}{s^2(s+1)^2}$$

$$(e) E(s) = \frac{s^2\epsilon^{-0.3Ts}+2s+3}{s(s+2)^2(s+4)}$$

$$(f) E(s) = \frac{(2s+7)\epsilon^{-0.75Ts}}{(s^2+2s+5)(s+3)}$$

**4.6-1.** Generally, a temperature control system is modeled more accurately if an ideal time delay is added to the plant. Suppose that in the thermal test chamber of Problem 4.3-9, the plant transfer function is given by

$$G_p(s) = \frac{C(s)}{E(s)} = \frac{2\epsilon^{-2s}}{s+0.5}$$

Hence the plant has a 2-s time delay before its response to an input. For this problem, let the sample period  $T = 0.6$  s.

- Find the unit step response for the system of Fig. P4.3-9; that is, find  $c(kT)$  with  $e(t) = u(t)$  and  $d(t) = 0$ , and with no delay.
- Repeat part (a), with the 2-s time delay included in  $G_p(s)$ , as given above.

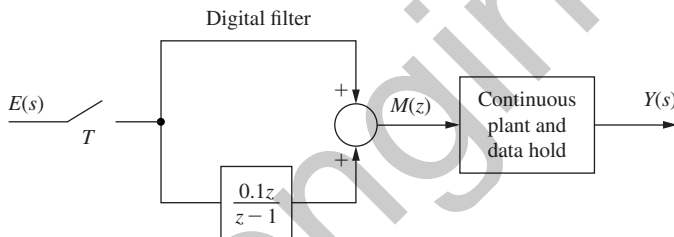
- (c) Solve for  $c(t)$  with no delay, using the Laplace transform and the plant input  $m(t)$ .
- (d) Find  $c(t)$  for the delay included, using the results of part (c).
- (e) Show that in part (c),  $c(t)$  evaluated at  $t = kT$  yields  $c(kT)$  in part (a).
- (f) Show that in part (d),  $c(t)$  evaluated at  $t = kT$  yields  $c(kT)$  in part (b).

**4.6-2.** For the thermal test chamber of Problems 4.3-9 and 4.6-1, let  $T = 0.6$  s. Suppose that a proportional-integral (PI) digital controller with the transfer function

$$D(z) = 1.2 + \frac{0.1z}{z - 1}$$

is inserted between the sampler and the zero-order hold in Fig. P4.3-9.

- (a) With  $d(t) = 0$  and  $e(t) = u(t)$ , solve for  $c(kT)$ , with the time delay of 2 s omitted.
  - (b) Explain what happens to the temperature in the chamber in part (a). Is this result physically possible?
  - (c) In Fig. P4.3-9, calculate  $m(kT)$ , the signal that controls the valve in the steam line, for the inputs of part (a).
  - (d) Considering the physical characteristics of a valve, what happens to the temperature in the physical test chamber?
- 4.6-3.** For Problem 4.4-6, assume that the total computational delay in  $D(z)$  plus the plant delay totals  $3.4T$  seconds. Repeat the problem under these new conditions.
- 4.8-1.** Find a discrete-state-variable representation for the system shown in Fig. P4.8-1.



**FIGURE P4.8-1** System for Problem 4.8-1.

A discrete-state-variable description of the continuous-time system is given by

$$\mathbf{x}(k + 1) = \begin{bmatrix} 0.9 & 1 & 2 \\ 0 & 0.9 & 0 \\ -1 & 0 & 0.5 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} m(k)$$

$$y(k) = [1 \quad 1.5 \quad 2.3]\mathbf{x}(k)$$

**4.8-2.** Repeat Problem 4.8-1 after replacing the digital filter with this transfer function:

$$D(z) = \frac{z - 0.5}{(z - 0.8)(z - 1)}$$

**4.9-1.** (a) The model of a continuous-time system with algebraic loops is given as

$$\dot{\mathbf{x}}(t) = -2\mathbf{x}(t) + 0.5\dot{\mathbf{x}}(t) + 3u(t)$$

$$y(t) = \dot{\mathbf{x}}(t) + 4u(t)$$

Derive the state equations for this system.

(b) Repeat part (a) for the system described by

$$\dot{x}_1(t) = -x_1(t) + 2\dot{x}_2(t) + u_1(t)$$

$$\dot{x}_2(t) = -\dot{x}_2(t) - x_2(t) - \dot{x}_1(t) + x_1(t) + u_2(t)$$

$$y(t) = \dot{x}_2(t)$$

Use the matrix technique of Section 4.9.

(c) Verify the results of part (b) by solving the given equations for  $\dot{x}_1(t)$  and  $\dot{x}_2(t)$  in the standard state-variable format.

**4.9-2.** The model of a continuous-time system with algebraic loops is given as

$$\begin{aligned}\dot{x}_1(t) &= -x_1(t) + 2\dot{x}_2(t) + u_1(t) \\ \dot{x}_2(t) &= -\dot{x}_2(t) - x_2(t) - \dot{x}_1(t) + x_1(t) + u_2(t) \\ \dot{x}_3(t) &= -\dot{x}_3(t) - x_3(t) - \dot{x}_2(t) + x_1(t) + u_2(t) \\ y(t) &= \dot{x}_3(t)\end{aligned}$$

Derive the state equations for this system. Use the matrix technique of Section 4.9. Verify the results by solving the given equations for  $\dot{x}_1(t)$ ,  $\dot{x}_2(t)$ , and  $\dot{x}_3(t)$  in the standard state-variable format.

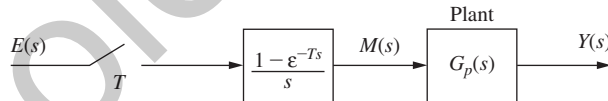
**4.10-1.** Consider the system of Fig. P4.10-1. The plant is described by the first-order differential equation

$$\frac{dy(t)}{dt} + 0.04y(t) = 0.2m(t)$$

Let  $T = 2$  s.

- Find the system transfer function  $Y(z)/E(z)$ .
- Draw a discrete simulation diagram, using the results of part (a), and give the state equations for this diagram.
- Draw a continuous-time simulation diagram for  $G_p(s)$ , and give the state equations for this diagram.
- Use the state-variable model of part (c) to find a discrete state model for the system. The state vectors of the discrete system and the continuous-time system are to be the same.
- Draw a simulation diagram for the discrete state model in part (d).
- Use Mason's gain formula to find the transfer function in part (e), which must be the same as that found in part (a).

Verify the results in parts (a) and (d) by computer.



**FIGURE P4.10-1** System for Problem 4.10-1.

**4.10-2.** Repeat Problem 4.10-1 for the plant described by the second-order differential equation for  $T = 2$

$$\frac{d^2y(t)}{dt^2} + 0.25\frac{dy(t)}{dt} + 0.005y(t) = 0.2m(t)$$

**4.10-3.** Consider the robot arm system of Fig. P4.3-8. Let  $T = 0.1$  s.

- Find the system transfer function  $\Theta_d(z)/M(z)$ .
- Draw a discrete simulation diagram, using the results of part (a), and give the state equations for this diagram.
- Draw a continuous-time simulation diagram for amplifier-servomotor-gears system and give the state equations for this diagram.
- Use the state-variable model of part (c) to find a discrete state model for the system. The states of the discrete systems are the same as those of the continuous-time system.
- Draw a simulation diagram for the discrete state model in part (d).
- Use Mason's gain formula to find the transfer function in part (e), which must be the same as found in part (a).

- 4.10-4.** Consider the thermal stress chamber depicted in Fig. P4.3-9. Let  $T = 0.6$  s. Ignore the disturbance input  $d(t)$  for this problem.
- Find the system transfer function  $C(z)/E(z)$ .
  - Draw a discrete simulation diagram, using the results of part (a), and give the state equations for this diagram.
  - Draw a continuous-time simulation diagram for the plant, and give the state equations for this diagram.
  - Use the state-variable model of part (c) to find a discrete state model for the system. The states of the discrete systems are the same as those of the continuous-time system.
  - Draw a simulation diagram for the discrete state model in part (d).
  - Use Mason's gain formula to find the transfer function in part (e), which must be the same as that found in part (a).
- 4.10-5.** Repeat Problem 4.10-4 for the satellite system of Fig. P4.3-10, with  $T = 1$  s. In part (a), the required transfer function is  $\Theta(z)/E(z)$ .
- 4.10-6.** Repeat Problem 4.10-4 for the antenna system of Fig. P4.3-11, with  $T = 0.05$  s. In part (a), the required transfer function is  $\Theta(z)/E(z)$ .
- 4.10-7.** Consider a proportional-integral (PI) digital controller with the transfer function

$$D(z) = 1.2 + \frac{0.1z}{z - 1}$$

This controller is placed in the designated system between the sampler and data hold. Find a discrete state model of:

- The system of Problem 4.10-1.
- The satellite system of Problem 4.10-5.
- In parts (a) and (b), use MATLAB to verify the results by calculating the system transfer functions from the state models.

# Closed-Loop Systems

## 5.1 INTRODUCTION

In Chapter 4 a technique was developed for determining the output functions of open-loop discrete linear time-invariant (LTI) systems. In this chapter we extend these techniques to determine the output functions of closed-loop discrete-time LTI systems. Also presented is a technique for developing state-variable models for closed-loop discrete-time systems. As a matter of convenience, we often more concisely refer to discrete-time systems as simply discrete systems.

## 5.2 PRELIMINARY CONCEPTS

Before considering simple closed-loop systems, open-loop systems with cascaded plants will be reviewed. As shown in Chapter 4, the output for the system of Fig. 5-1(a) is

$$C(z) = G_1(z)G_2(z)E(z) \quad (5-1)$$

and that of Fig. 5-1(b) is

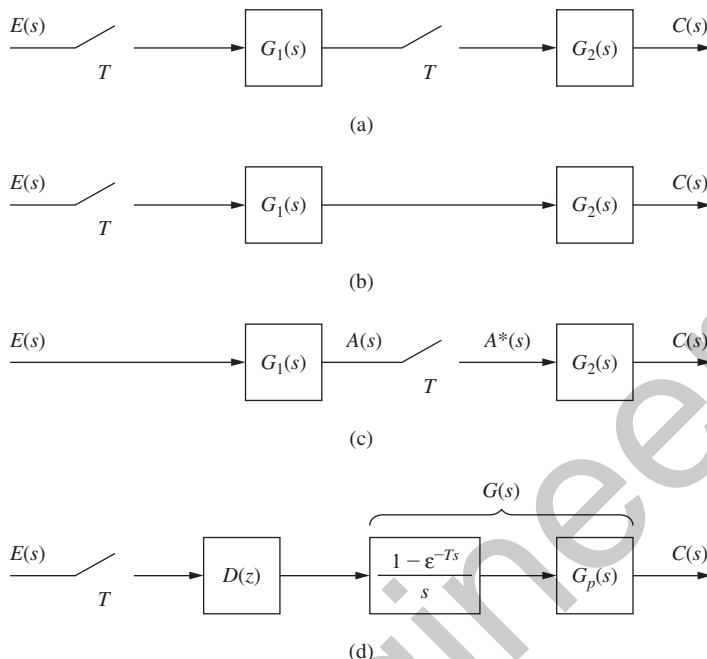
$$C(z) = \overline{G_1G_2}(z)E(z) \quad (5-2)$$

For the system of Fig. 5-1(c),

$$C(s) = G_2(s)A^*(s) = G_2(s)\overline{G_1E^*}(s) \quad (5-3)$$

and

$$C(z) = G_2(z)\overline{G_1E}(z) \quad (5-4)$$



**FIGURE 5-1** Open-loop sampled-data systems.

For this case, no transfer function can be found since  $E(z)$  cannot be factored from  $\overline{G_1 E}(z)$ . In general, no transfer function can be written for the system in which the input is applied to an analog element before being sampled. However, the output can always be expressed as a function of the input, and as will be shown later, this type of system presents no particular difficulties in either analysis or design. For the system of Fig. 5-1(d), the output is

$$C(z) = D(z)G(z)E(z)$$

where

$$G(z) = \mathcal{Z} \left[ \frac{1 - e^{-Ts}}{s} G_p(s) \right] = \frac{z - 1}{z} \mathcal{Z} \left[ \frac{G_p(s)}{s} \right]$$

We now derive the output function for the system of Fig. 5-2. First we write

$$C(s) = G(s)E^*(s) \tag{5-5}$$

and

$$E(s) = R(s) - H(s)C(s) \tag{5-6}$$

Substituting (5-5) into (5-6), we obtain

$$E(s) = R(s) - G(s)H(s)E^*(s) \tag{5-7}$$



and by taking the starred transform (see Section 4.3), we have

$$E^*(s) = R^*(s) - \overline{GH}^*(s)E^*(s) \quad (5-8)$$

Solving for  $E^*(s)$ , we obtain

$$E^*(s) = \frac{R^*(s)}{1 + \overline{GH}^*(s)} \quad (5-9)$$

and from (5-5),

$$C(s) = G(s) \frac{R^*(s)}{1 + \overline{GH}^*(s)} \quad (5-10)$$

which yields an expression for the continuous output. The sampled output is, then,

$$C^*(s) = G^*(s)E^*(s) = \frac{G^*(s)R^*(s)}{1 + \overline{GH}^*(s)}, \quad C(z) = \frac{G(z)R(z)}{1 + \overline{GH}(z)} \quad (5-11)$$

Here let us make an observation about practical implementations of systems as illustrated in Fig. 5-2. Since  $E^*(s)$  is a series of impulse functions, we must assume that  $G(s)$  receives the impulses through a *zero-order hold*. This will always be the case in what follows unless specified otherwise. Anytime a starred transform is applied to a Laplace transfer function, assume that the transfer function contains a *zero-order hold* to process its starred-transform input.

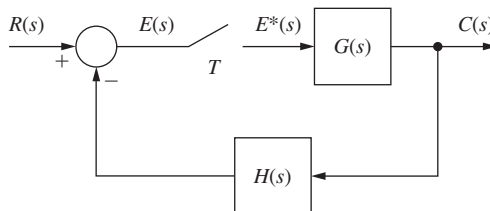
Problems can be encountered in deriving the output function of a closed-loop system. This can be illustrated for the case above. Suppose that (5-6) had been starred and substituted into (5-5). Then

$$C(s) = G(s)R^*(s) - G(s)\overline{HC}^*(s) \quad (5-12)$$

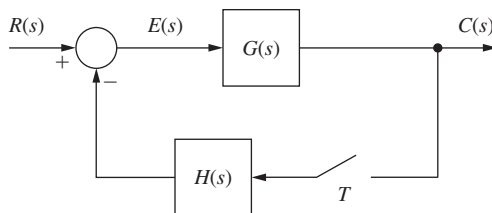
and  $C^*(s)$  is

$$C^*(s) = G^*(s)R^*(s) - G^*(s)\overline{HC}^*(s) \quad (5-13)$$

In general,  $C^*(s)$  cannot be factored from  $\overline{HC}^*(s)$ . Thus (5-13) cannot be solved for  $C^*(s)$ . In general, in analyzing a system, an equation should not be starred if a system signal is lost as a factor, as shown above. However, for systems more complex than the one above, solving the system equations can become quite complex. A simpler method of analysis will now be developed that avoids this problem.



**FIGURE 5-2** Closed-loop sampled-data system.



**FIGURE 5-3** Sampled-data system.

First, however, consider the system of Fig. 5-3. Since the input is not sampled before being applied to an analog element, no transfer function can be derived.

Nevertheless, the system can be analyzed. From Fig. 5-3 we note that

$$C(s) = G(s)E(s) \quad (5-14)$$

and

$$E(s) = R(s) - H(s)C^*(s) \quad (5-15)$$

Substituting (5-15) into (5-14), we obtain

$$C(s) = G(s)R(s) - G(s)H(s)C^*(s) \quad (5-16)$$

Starring (5-16) yields

$$C^*(s) = \overline{GR}^*(s) - \overline{GH}^*(s)C^*(s) \quad (5-17)$$

In this system, the forcing function  $R(s)$  is necessarily lost as a factor in (5-17), since  $r(t)$  is not sampled. Solving (5-17) for  $C^*(s)$ , we obtain

$$C^*(s) = \frac{\overline{GR}^*(s)}{1 + \overline{GH}^*(s)}, \quad C(z) = \frac{\overline{GR}(z)}{1 + \overline{GH}(z)} \quad (5-18)$$

The continuous output is obtained from (5-16) and (5-18) as

$$C(s) = G(s)R(s) - \frac{G(s)H(s)\overline{GR}^*(s)}{1 + \overline{GH}^*(s)} \quad (5-19)$$

For the system of Fig. 5-3, no transfer function can be derived; the problem is that the input is not sampled before being applied to an analog part of the system. In a practical system, we generally have more than one input. Consider, as an example, an aircraft flying in a closed-loop mode (i.e., flying on autopilot). Suppose also that the autopilot is implemented digitally. Thus this system is a closed-loop digital control system. The commands into this system (e.g., an altitude change) may be inputted into the control computer through an analog-to-digital converter, or may even be generated by the computer itself. Thus this input is sampled, and a transfer function may be developed. However, the vertical component of

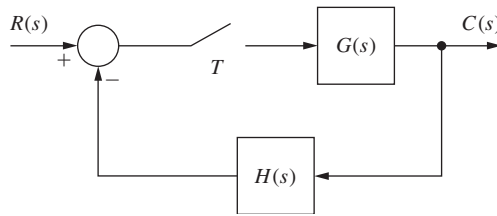


FIGURE 5-4 Sampled-data control system.



FIGURE 5-5 Original flow graph for the sampled-data system in Fig. 5-4.

the wind, which must also be considered as an input into the altitude control system, is not sampled. Hence a transfer function can be derived from the command input to the output (the altitude); however, a transfer function cannot be developed from wind input to aircraft altitude as the output.

### 5.3 DERIVATION PROCEDURE

The determination of transfer functions for sampled-data systems is difficult because a transfer function for the ideal sampler does not exist. A procedure for finding transfer functions will now be developed using the system of Fig. 5-4, which has the flow graph shown in Fig. 5-5. We omit the sampler from the system signal flow graph, since a transfer function cannot be written for the device. Note that the effect of the sampler is included in the flow graph, since the sampler output is shown in starred form,  $E_1^*$ , and  $E_1^*$  will be treated as an input. This flow graph is referred to as the original signal flow graph. The sampled output  $C^*(s)$  can be found for a discrete-time system of this type by employing the following step-by-step procedure.

1. Construct the *original signal flow graph*. This has been done for the system and is shown in Fig. 5-5.
2. Assign a variable to each sampler input. Then the sampler output is this variable starred. For this example, system  $E_1$  is the input to the sampler and  $E_1^*$  is the sampler output.
3. Considering each sampler output to be a source node (input), express the sampler inputs and the system output in terms of each sampler output (which is treated as an input in the flow graph), and the system input. For this example,

$$E_1 = R - GE_1^* \quad (5-20)$$

$$C = GE_1^* \quad (5-21)$$

where  $E_1 = E_1(s)$ , and so on. For convenience, the dependency on  $s$  will not be shown.

4. Take the starred transform of these equations and solve by any convenient method. For the example,

$$E_1^* = R^* - \overline{GH}^* E_1^* \quad (5-22)$$

$$C^* = G^* E_1^* \quad (5-23)$$

From (5-22),

$$E_1^* = \frac{R^*}{1 + \overline{GH}^*} \quad (5-24)$$

and, from (5-23) and (5-24),

$$C^*(s) = \frac{G^*(s)}{1 + \overline{GH}^*(s)} R^*(s) \quad (5-25)$$

If there is more than one sampler in the system, the equations for the starred variables may be solved using Cramer's rule, or any other technique that is applicable to the solution of linear simultaneous equations.

The systems equations can also be solved by constructing a signal flow graph from these equations and applying Mason's gain formula. This flow graph is called the *sampled signal flow graph*. This method is sometimes superior to Cramer's rule, provided that the sampled signal flow graph is simple enough so that all loops are easily identified. To illustrate this approach, consider again equations (5-21), (5-22), and (5-23). The signal flow graph for these equations is shown in Fig. 5-6. From the flow graph,

$$C^*(s) = \frac{G^*(s)}{1 + \overline{GH}^*(s)} R^*(s) \quad (5-26)$$

and this result agrees with (5-25). Note also from Fig. 5-6 that

$$C(s) = \frac{G(s)}{1 + \overline{GH}^*(s)} R^*(s) \quad (5-27)$$

This method for finding the output function for closed-loop systems will now be illustrated via the following examples.

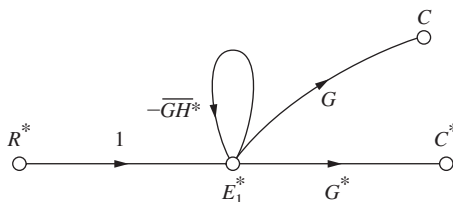


FIGURE 5-6 Sampled signal flow graph for the system in Fig. 5-4.

**EXAMPLE 5.1**

Consider the digital control system of Fig. 5-7(a) which has the model given in Fig. 5-7(b) (see Section 4.4). The original flow graph is shown in Fig. 5-8.

We can write

$$E = R - GHD^*E^*$$

$$C = GD^*E^*$$

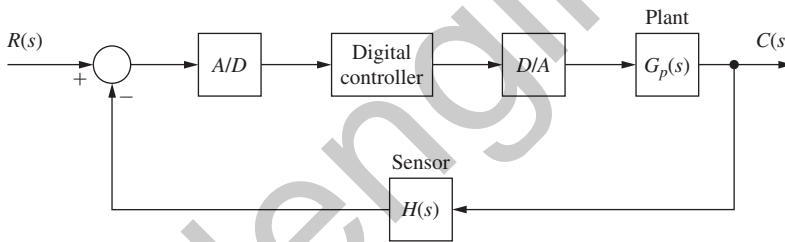
Hence

$$E^* = R^* - \overline{GH^*D^*}E^* \rightarrow E^* = \frac{R^*}{1 + D^*\overline{GH^*}}$$

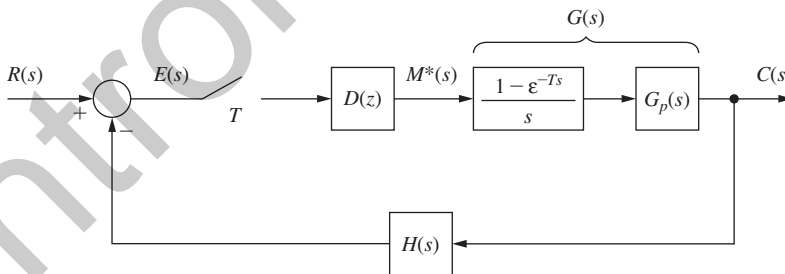
$$C^* = G^*D^*E^*$$

Thus

$$C(z) = \frac{D(z)G(z)}{1 + D(z)\overline{GH^*}(z)} R(z)$$

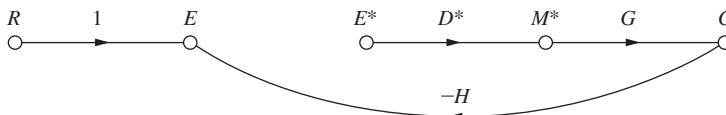


(a)



(b)

**FIGURE 5-7** Closed-loop digital control system.



**FIGURE 5-8** Original flow graph for Example 5.1.

**EXAMPLE 5.2**

Consider the system shown in Fig. 5-9. The original signal flow graph is shown in Fig. 5-10. The system equations are

$$\begin{aligned}
 E_1 &= R - G_2E_2^* \\
 E_2 &= G_1E_1^* - G_2HE_2^* \\
 C &= G_2E_2^*
 \end{aligned}$$

Starring these equations, we obtain

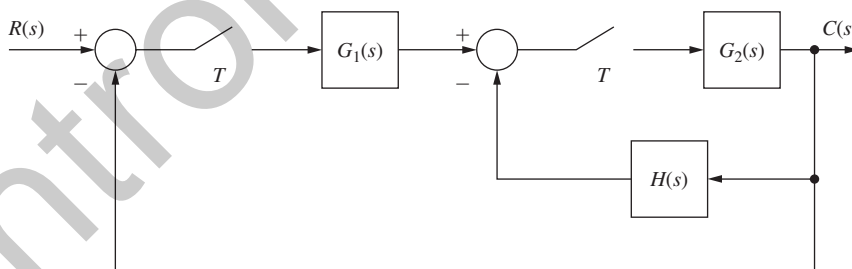
$$\begin{aligned}
 E_1^* &= R^* - G_2^*E_2^* \\
 E_2^* &= G_1^*E_1^* - \overline{G_2H^*}E_2^* \\
 C^* &= G_2^*E_2^*
 \end{aligned}$$

The sampled flow graph can then be drawn from these equations as shown in Fig. 5-11. Then applying Mason's gain formula, we obtain

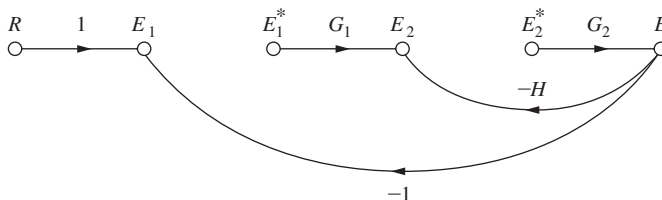
$$C^* = \frac{G_1^*G_2^*}{1 + G_1^*G_2^* + \overline{G_2H^*}}R^* \quad \text{or} \quad C(z) = \frac{G_1(z)G_2(z)R(z)}{1 + G_1(z)G_2(z) + \overline{G_2H(z)}}$$

and

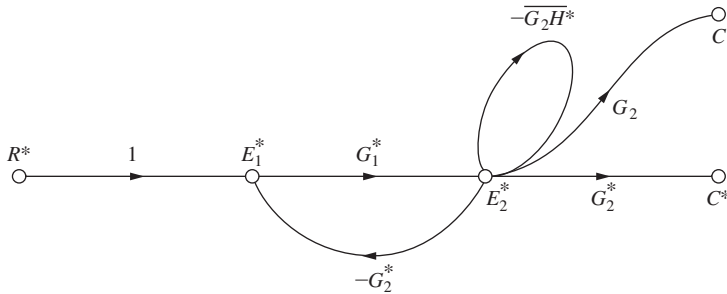
$$C(s) = \frac{G_2(s)G_1^*(s)}{1 + G_1^*(s)G_2^*(s) + \overline{G_2H^*(s)}}R^*(s)$$



**FIGURE 5-9** Sampled-data system for Example 5.2.



**FIGURE 5-10** Original flow graph for Example 5.2.



**FIGURE 5-11** Sampled flow graph for Example 5.2.

### EXAMPLE 5.3

As another example, consider the system shown in Fig. 5-12. Note that no transfer function may be derived for this system, since the input  $R(s)$  reaches  $G_2(s)$  without being sampled. The original signal flow graph is given in Fig. 5-13. From this Fig. note that

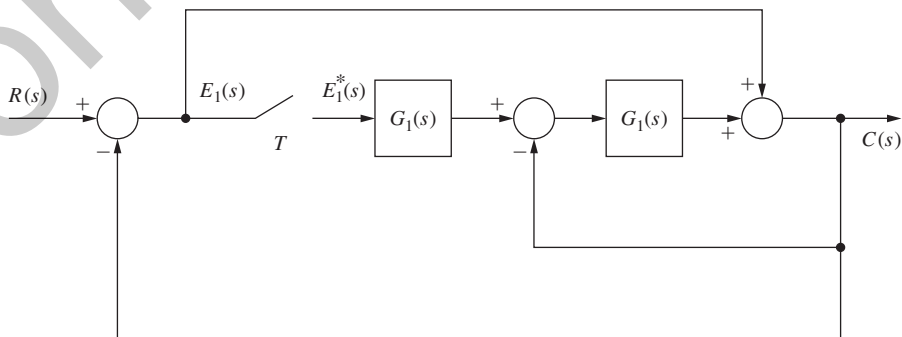
$$\begin{aligned}
 E_1 &= R - C \\
 C &= E_1 + G_2[G_1E_1^* - C] \\
 &= R - C + G_1G_2E_1^* - G_2C
 \end{aligned}$$

Therefore,

$$[1 + 1 + G_2]C = R + G_1G_2E_1^*$$

and

$$\begin{aligned}
 C &= \frac{R}{2 + G_2} + \frac{G_1G_2}{2 + G_2} E_1^* \\
 E_1 = R - C &= \frac{(1 + G_2)R}{2 + G_2} - \frac{G_1G_2}{2 + G_2} E_1^*
 \end{aligned}$$



**FIGURE 5-12** Sampled-data system for Example 5.3.

Or the equation for  $E_1$  and  $C$  can be written directly from Fig. 5-13 using Mason's formula. Hence

$$C^* = \left[ \frac{R}{2 + G_2} \right]^* + \left[ \frac{G_1 G_2}{2 + G_2} \right]^* E_1^*$$

$$E_1^* = \left[ \frac{(1 + G_2)R}{2 + G_2} \right]^* - \left[ \frac{G_1 G_2}{2 + G_2} \right]^* E_1^*$$

The sampled flow graph derived from these equations is given in Fig. 5-14. Then by employing Mason's gain formula, we obtain

$$C^*(s) = \left[ \frac{R}{2 + G_2} \right]^*(s) + \frac{\left[ \frac{G_1 G_2}{2 + G_2} \right]^*(s)}{1 + \left[ \frac{G_1 G_2}{2 + G_2} \right]^*(s)} \left[ \frac{(1 + G_2)R}{2 + G_2} \right]^*(s)$$

Note that, as stated above, no transfer function is possible, since the input is fed into a continuous element in the system,  $G_2(s)$ , without first being sampled.

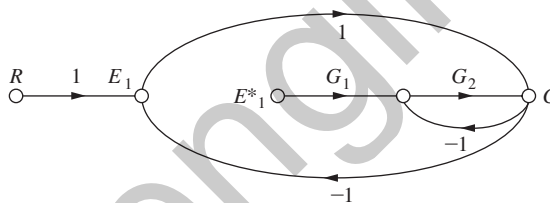


FIGURE 5-13 Original flow graph for Example 5.3.

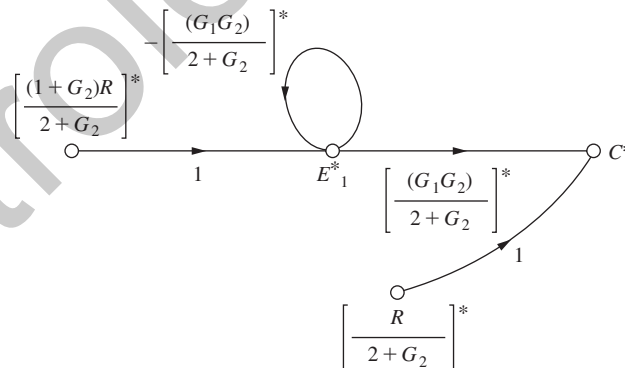


FIGURE 5-14 Sampled flow graph for Example 5.3.

As mentioned above, matrix methods, or Cramer's rule, may be simpler to use in solving the system equations, if there are many loops in the sampled signal flow graph. This is especially true when all the loops of the flow graph are not easily identified. However, once one has gained experience and proficiency in the use of signal flow graphs, these flow graphs can be easily used to obtain quick and accurate solutions for less complex systems.



### EXAMPLE 5.4

As a final example, consider the system of Fig. 5-15, which contains a digital controller with a computation time of  $t_0$  seconds,  $t_0 < T$ . The effect of the computation time is modeled as the ideal time delay (see Section 4.6).

The original signal flow graph is given in Fig. 5-16. The system equation is then, from Mason's gain formula,

$$C = \frac{GR}{1 + GH_1} - \frac{GH_2 \varepsilon^{-t_0 s}}{1 + GH_1} D^* C^*$$

and thus

$$C(z) = \left[ \frac{GR}{1 + GH_1} \right](z) - \left[ \frac{GH_2}{1 + GH_1} \right](z, m) D(z) C(z)$$

where  $mT = T - t_0$ . The second equation may be solved directly for  $C(z)$ .

$$C(z) = \frac{\left[ \frac{GR}{1 + GH_1} \right](z)}{1 + \left[ \frac{GH_2}{1 + GH_1} \right](z, m) D(z)}$$

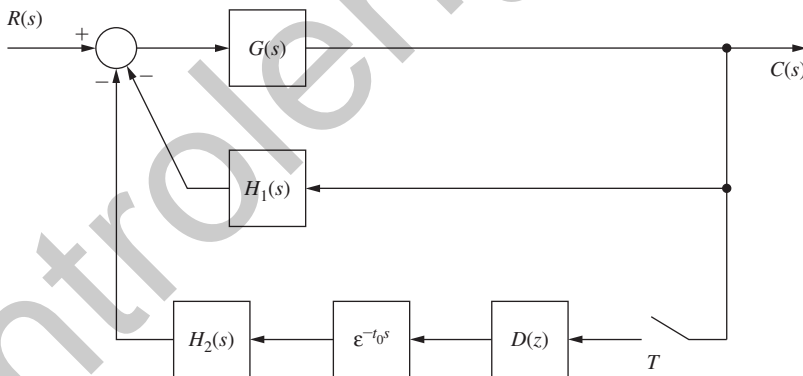


FIGURE 5-15 System for Example 5.4.

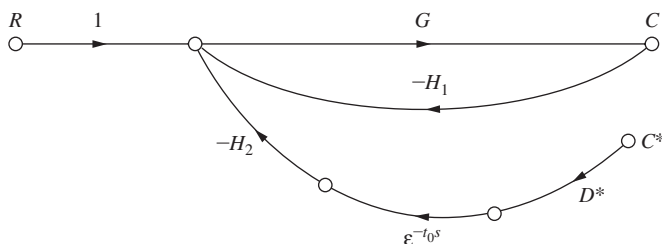


FIGURE 5-16 Original flow graph for Example 5.4.

If the computational delay is greater than one sampling interval (i.e., if  $t_0 = kT + \Delta T$ ,  $k$  is a positive integer), the denominator becomes

$$1 + z^{-k} \left[ \frac{GH_2}{1 + GH_1} \right] (z, m) D(z)$$

where  $mT = T - \Delta T$ .

## 5.4 STATE-VARIABLE MODELS

A technique for finding the transfer function of a closed-loop discrete-time system was presented above. As shown in Chapter 4, a discrete-time state-variable model may be generated directly from the transfer function. However, this technique has the disadvantage that the analog system physical variables generally do not appear as discrete-time state variables.

**CONVERTING CONTINUOUS STATE EQUATIONS.** The technique of converting continuous state equations to discrete-time state equations, presented in Section 4.10, may also be utilized in determining a discrete-time state-variable model for a closed-loop system. The application of this technique will be illustrated by an example. Consider the system of Example 5.2, which is repeated in Fig. 5-17(a). As a first step, the system is redrawn such that zero-order-hold outputs

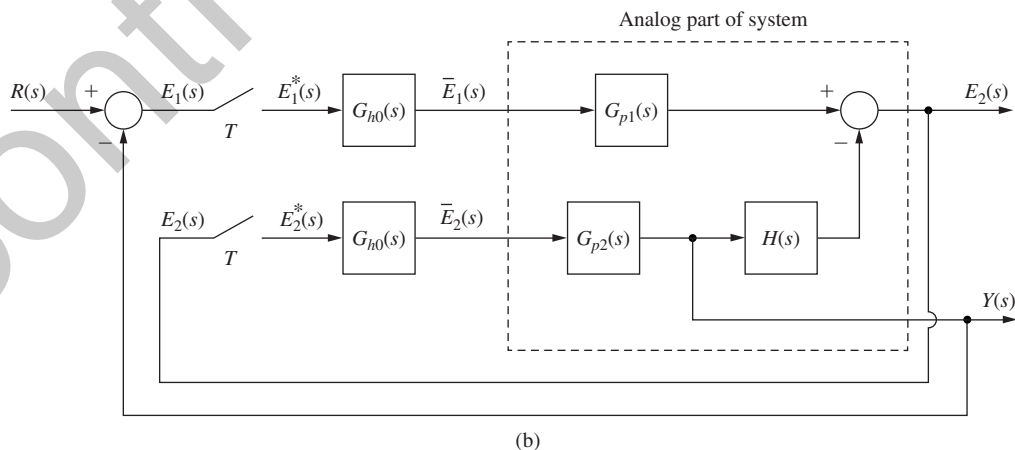
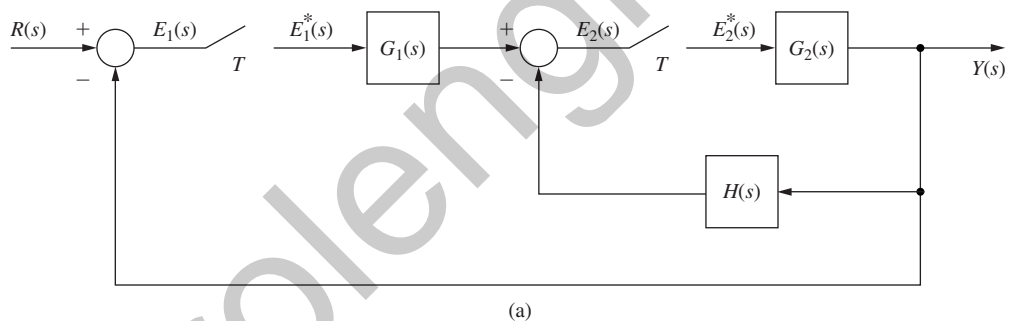
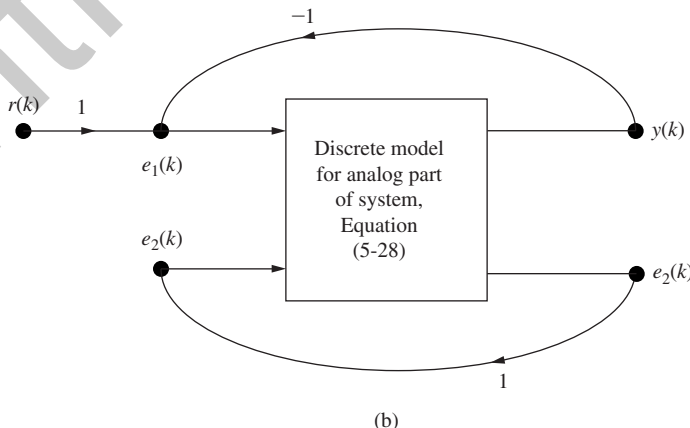
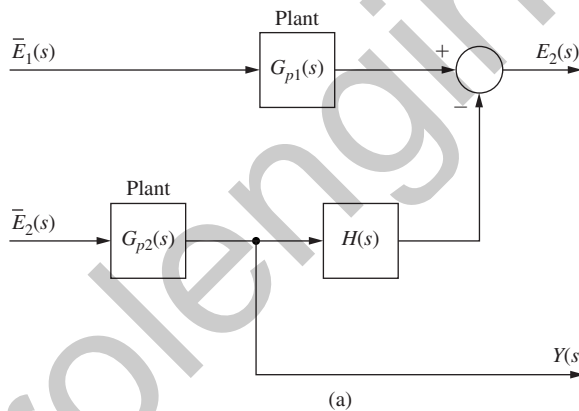


FIGURE 5-17 Closed-loop system.

are shown as inputs, and sampler inputs and the system output are shown as outputs. The results of this step are shown in Fig. 5-17(b). Hence we are considering the analog part of the system as in Fig. 5-18(a) [ $E_1(s)$  is not shown as an output, since it is determined directly from the system output  $Y(s)$ ]. Next the continuous state equations for this part of the system are written, and from these equations, the discrete-time state equations are generated. For this system, the discrete-time state equations will be of the form

$$\begin{aligned} \mathbf{v}(k+1) &= \mathbf{A}_1\mathbf{v}(k) + \mathbf{B}_1 \begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix} \\ \begin{bmatrix} y(k) \\ e_2(k) \end{bmatrix} &= \mathbf{C}_1\mathbf{v}(k) + \mathbf{D}_1 \begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix} \end{aligned} \quad (5-28)$$

Either a discrete simulation diagram or a flow graph is then constructed from these state equations, and should include all connecting paths of the closed-loop system external to the simulation diagram for (5-28). The result for the system considered is shown in Fig. 5-18(b). From this simulation diagram the system discrete state equations may be written. An example to illustrate this technique will now be given.



**FIGURE 5-18** Technique for determining the discrete-time state model.

**EXAMPLE 5.5**

The discrete state model for the system of Fig. 5-17 will be derived, with  $T = 0.1$  s, and with

$$\begin{aligned}
 G_1(s) &= \frac{1 - \epsilon^{-Ts}}{s^2(s + 1)} = \frac{1 - \epsilon^{-Ts}}{s} G_{p1}(s) \\
 G_2(s) &= \frac{2(1 - \epsilon^{-Ts})}{s(s + 2)} = \frac{1 - \epsilon^{-Ts}}{s} G_{p2}(s) \\
 G_{p1}(s) &= \frac{1}{s(s + 1)}, \quad G_{p2}(s) = \frac{2}{s + 2}, \quad H(s) = \frac{10}{s + 10}
 \end{aligned}$$

A flow graph of the system of Fig. 5.18(a) is shown in Fig. 5-19(a). From this flow graph we write the continuous state equations:

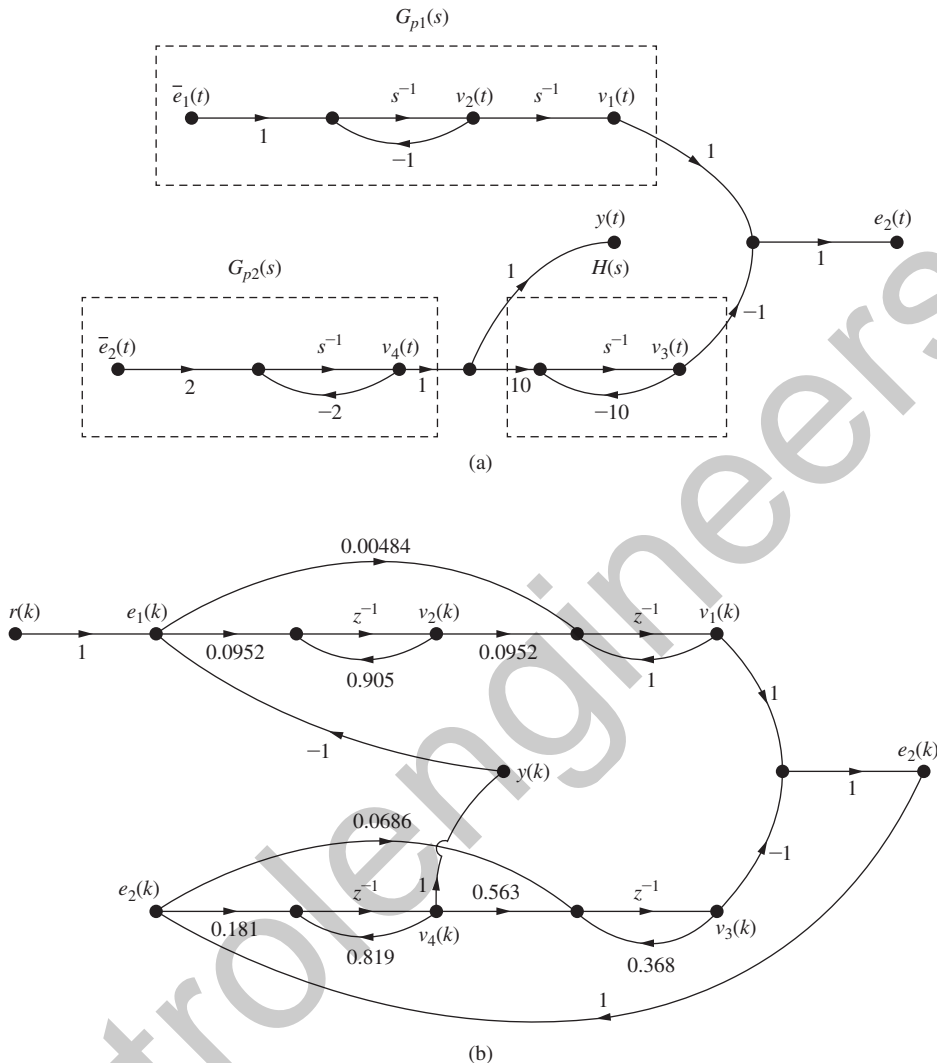
$$\begin{aligned}
 \dot{\mathbf{v}}(t) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -10 & 10 \\ 0 & 0 & 0 & -2 \end{bmatrix} \mathbf{v}(t) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \bar{e}_1(t) \\ \bar{e}_2(t) \end{bmatrix} \\
 \begin{bmatrix} y(k) \\ e_2(k) \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \mathbf{v}(k)
 \end{aligned}$$

In these equations,  $\bar{e}_1(t)$  is the zero-order-hold output with  $e_1^*(t)$  as its input, and  $\bar{e}_2(t)$  is defined in the same manner. To obtain the discrete state matrices, we use the technique of Section 4.10 and, for this example, calculate the following matrices by computer:

$$\begin{aligned}
 \mathbf{v}(k + 1) &= \begin{bmatrix} 1 & 0.0952 & 0 & 0 \\ 0 & 0.905 & 0 & 0 \\ 0 & 0 & 0.368 & 0.563 \\ 0 & 0 & 0 & 0.819 \end{bmatrix} \mathbf{v}(k) + \begin{bmatrix} 0.00484 & 0 \\ 0.0952 & 0 \\ 0 & 0.0686 \\ 0 & 0.181 \end{bmatrix} \begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix} \\
 \begin{bmatrix} y(k) \\ e_2(k) \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \mathbf{v}(k)
 \end{aligned}$$

A simulation diagram of this system, with all external connecting paths, is shown in Fig. 5-19(b). From this simulation diagram we write the discrete state equation for the closed-loop system.

$$\begin{aligned}
 \mathbf{v}(k + 1) &= \begin{bmatrix} 1 & 0.0952 & 0 & -0.00484 \\ 0 & 0.905 & 0 & -0.0952 \\ 0.0686 & 0 & 0.299 & 0.563 \\ 0.181 & 0 & -0.181 & 0.819 \end{bmatrix} \mathbf{v}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \\ 0 \\ 0 \end{bmatrix} r(k) \\
 \mathbf{y}(k) &= [0 \ 0 \ 0 \ 1] \mathbf{v}(k)
 \end{aligned}$$



**FIGURE 5-19** System for Example 5.5.

Note the similarity of this technique to that developed earlier for finding closed-loop transfer functions. The system is opened at each sampler, each zero order-hold output is assumed to be an input, and each sampler input is assumed to be an output. Discrete state equations are then written relating these specified inputs and outputs. These state equations are manipulated, through the use of a simulation diagram, to obtain the state equations of the closed-loop system.

The technique above can be applied to low-order systems; however, writing system equations from complex flow graphs is at best tenuous. Instead, a matrix procedure that can be implemented by a computer program is needed, and one will now be developed. Consider Example 5.5. The discrete state equations for the continuous system can be written as

$$\mathbf{v}(k + 1) = \mathbf{A}_1\mathbf{v}(k) + \mathbf{B}_1\mathbf{e}(k) \quad (5-29)$$

$$\mathbf{e}(k) = \mathbf{C}_1\mathbf{v}(k) + \mathbf{D}_1r(k) \quad (5-30)$$

where  $\mathbf{e}(k) = [e_1(k) \ e_2(k)]^T$ . Thus these equations can be combined to yield

$$\mathbf{v}(k + 1) = [\mathbf{A}_1 + \mathbf{B}_1\mathbf{C}_1]\mathbf{v}(k) + \mathbf{B}_1\mathbf{D}_1r(k) \quad (5-31)$$

which is the required equation. An example will now be given.

### EXAMPLE 5.6

Consider the system of Example 5.5. Since  $e_1(k) = r(k) - y(k) = r(k) - v_4(k)$  and  $e_2(k) = v_1(k) - v_3(k)$ , the equations for  $\mathbf{e}(k)$  can be written as

$$\begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \mathbf{v}(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r(k) = \mathbf{C}_1\mathbf{v}(k) + \mathbf{D}_1r(k)$$

Then, in (5-31),

$$\begin{aligned} \mathbf{B}_1\mathbf{C}_1 &= \begin{bmatrix} 0.00484 & 0 \\ 0.0952 & 0 \\ 0 & 0.0686 \\ 0 & 0.181 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & -0.00484 \\ 0 & 0 & 0 & -0.0952 \\ 0.0686 & 0 & -0.0686 & 0 \\ 0.181 & 0 & -0.181 & 0 \end{bmatrix} \end{aligned}$$

and

$$\begin{aligned} \mathbf{A}_1 + \mathbf{B}_1\mathbf{C}_1 &= \begin{bmatrix} 1 & 0.0952 & 0 & 0 \\ 0 & 0.905 & 0 & 0 \\ 0 & 0 & 0.368 & 0.563 \\ 0 & 0 & 0 & 0.819 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & -0.00484 \\ 0 & 0 & 0 & -0.0952 \\ 0.0686 & 0 & -0.0686 & 0 \\ 0.181 & 0 & -0.181 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0.0952 & 0 & -0.00484 \\ 0 & 0.905 & 0 & -0.0952 \\ 0.0686 & 0 & 0.2994 & 0.563 \\ 0.181 & 0 & -0.181 & 0.819 \end{bmatrix} \end{aligned}$$

Also, in (5-31),

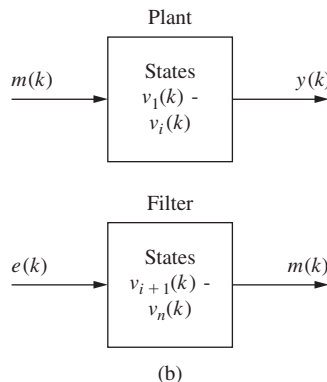
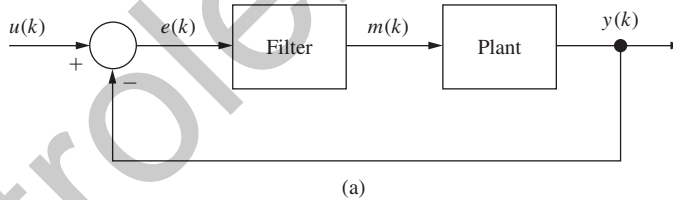
$$\mathbf{B}_1\mathbf{D}_1 = \begin{bmatrix} 0.00484 & 0 \\ 0.0952 & 0 \\ 0 & 0.0686 \\ 0 & 0.181 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.00484 \\ 0.0952 \\ 0 \\ 0 \end{bmatrix}$$

These results agree with those obtained in Example 5.5 and are less prone to error. Also, the matrix procedure of this example can be implemented on a digital computer.

As a final point, note that each system input must be sampled prior to being applied to an analog part of the system. If this is not the case, (5-30) cannot be written as a function of only  $r(k)$ , and no discrete model is possible. If an input that is not sampled varies slowly over a sample period (only low frequencies appear in the input signal), this input is often assumed to be sampled, even though the resultant discrete model is somewhat in error.

**INCLUDING A DIGITAL CONTROLLER.** System state equations are more difficult to derive if the system contains a digital controller. A single-loop digital control system is shown in Fig. 5-20(a). To obtain the state equations, we consider the digital filter and the plant separately and write the state equations for these two parts. We assign states  $v_1(k)$  through  $v_i(k)$  to the plant, where  $i$  is the order of the plant. Then we assign states  $v_{i+1}(k)$  through  $v_n(k)$  to the filter, where  $n - i$  is the order of the filter. Hence we can write the state equations

$$\mathbf{v}(k + 1) = \mathbf{A}_1\mathbf{v}(k) + \mathbf{B}_1m(k) + \mathbf{B}_2e(k) \quad (5-32)$$



**FIGURE 5-20** Discrete control system.

since both  $m(k)$  and  $e(k)$  are inputs, as shown in Fig. 5-20(b). Next we write for the filter

$$m(k) = \mathbf{C}_1 \mathbf{v}(k) + D_1 e(k) \quad (5-33)$$

and for the plant

$$y(k) = \mathbf{C} \mathbf{v}(k) \quad (5-34)$$

Thus, for the feedback path and from (5-34), we write

$$e(k) = u(k) - y(k) = u(k) - \mathbf{C} \mathbf{v}(k) \quad (5-35)$$

We obtain the system state equations by eliminating  $m(k)$  and  $e(k)$  from (5-32), (5-33), and (5-35). From (5-33) and (5-35),

$$m(k) = \mathbf{C}_1 \mathbf{v}(k) + D_1 [u(k) - \mathbf{C} \mathbf{v}(k)] = [\mathbf{C}_1 - D_1 \mathbf{C}] \mathbf{v}(k) + D_1 u(k) \quad (5-36)$$

Then, substituting (5-35) and (5-36) into (5-32), we obtain

$$\begin{aligned} \mathbf{v}(k+1) &= \mathbf{A}_1 \mathbf{v}(k) + \mathbf{B}_1 [(\mathbf{C}_1 - D_1 \mathbf{C}) \mathbf{v}(k) + D_1 u(k)] \\ &\quad + \mathbf{B}_2 [u(k) - \mathbf{C} \mathbf{v}(k)] \\ &= [\mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_1 + (-\mathbf{B}_2 - \mathbf{B}_1 D_1) \mathbf{C}] \mathbf{v}(k) + [\mathbf{B}_1 D_1 + \mathbf{B}_2] u(k) \end{aligned} \quad (5-37)$$

In our standard notation,

$$\begin{aligned} \mathbf{v}(k+1) &= \mathbf{A} \mathbf{v}(k) + \mathbf{B} u(k) \\ \mathbf{A} &= \mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_1 + (-\mathbf{B}_2 - \mathbf{B}_1 D_1) \mathbf{C} \\ \mathbf{B} &= \mathbf{B}_1 D_1 + \mathbf{B}_2 \end{aligned} \quad (5-38)$$

An example will now be given.

### EXAMPLE 5.7

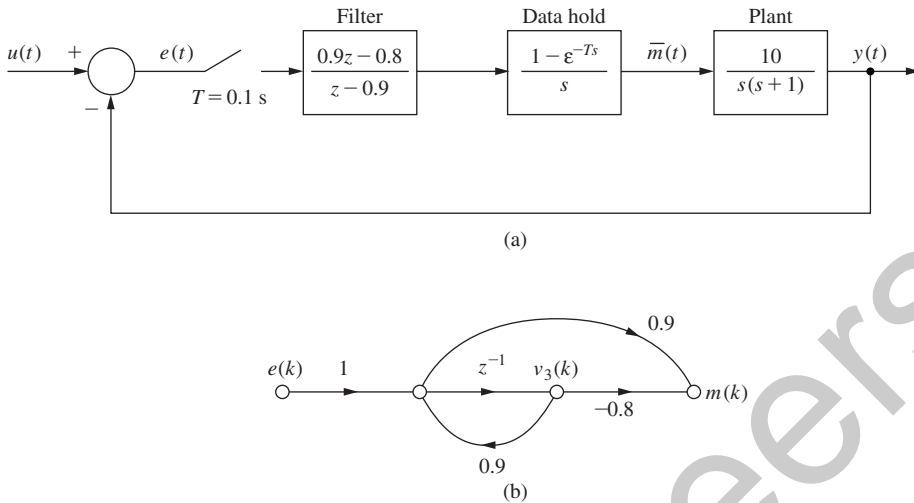
The state equations for the system of Example 4.13 (shown in Fig. 5-21(a)) will be developed. From Example 4.13, the state equations for the plant are

$$\begin{aligned} \begin{bmatrix} v_1(k+1) \\ v_2(k+1) \end{bmatrix} &= \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} + \begin{bmatrix} 0.0484 \\ 0.952 \end{bmatrix} m(k) \\ y(k) &= [1 \quad 0] \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} \end{aligned}$$

The filter is modeled as shown in Fig. 5-21(b). The state equations for the filter are

$$\begin{aligned} v_3(k+1) &= 0.9v_3(k) + e(k) \\ m(k) &= (0.81 - 0.8)v_3(k) + 0.9e(k) \\ &= 0.01v_3(k) + 0.9e(k) \end{aligned}$$





**FIGURE 5-21** System for Example 5.7.

Combining the state equations for  $\mathbf{v}(k)$ , we obtain

$$\begin{bmatrix} v_1(k+1) \\ v_2(k+1) \\ v_3(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.0952 & 0 \\ 0 & 0.905 & 0 \\ 0 & 0 & 0.9 \end{bmatrix} \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix} + \begin{bmatrix} 0.0484 \\ 0.952 \\ 0 \end{bmatrix} m(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} e(k)$$

Also, from Fig. 5-21, since  $e(t) = u(t) - y(t)$ ,

$$e(k) = u(k) - y(k) = u(k) - [1 \ 0 \ 0]\mathbf{v}(k)$$

and from above,

$$m(k) = [0 \ 0 \ 0.01]\mathbf{v}(k) + 0.9e(k)$$

Comparing the equation above for  $\mathbf{v}(k)$  with (5-32), we see that

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0.0952 & 0 \\ 0 & 0.905 & 0 \\ 0 & 0 & 0.9 \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} 0.0484 \\ 0.952 \\ 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

From the equation above for  $m(k)$  and (5-33),

$$\mathbf{C}_1 = [0 \ 0 \ 0.01], \quad D_1 = 0.9$$

and from the equation above for  $e(k)$  and (5-35),

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

Then, in (5-37),

$$\mathbf{B}_1\mathbf{C}_1 = \begin{bmatrix} 0.0484 \\ 0.952 \\ 0 \end{bmatrix} [0 \ 0 \ 0.01] = \begin{bmatrix} 0 & 0 & 0.000484 \\ 0 & 0 & 0.00952 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned}
 (-\mathbf{B}_2 - \mathbf{B}_1 D_1) \mathbf{C} &= \begin{bmatrix} 0 - 0.9(0.0484) \\ 0 - 0.9(0.952) \\ -1 - 0.9(0) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\
 &= \begin{bmatrix} -0.04356 & 0 & 0 \\ -0.8568 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

Thus

$$\begin{aligned}
 \mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_1 + (-\mathbf{B}_2 - \mathbf{B}_1 D_1) \mathbf{C} &= \begin{bmatrix} 0.9564 & 0.0952 & 0.000484 \\ -0.8568 & 0.905 & 0.00952 \\ -1 & 0 & 0.9 \end{bmatrix} \\
 \mathbf{B}_1 D_1 + \mathbf{B}_2 &= \begin{bmatrix} 0.04356 + 0 \\ 0.8568 + 0 \\ 0 + 1 \end{bmatrix}
 \end{aligned}$$

Thus the state equations for this system are, from (5-37) and (5-34),

$$\begin{aligned}
 \mathbf{v}(k+1) &= \begin{bmatrix} 0.9564 & 0.0952 & 0.000484 \\ -0.8568 & 0.905 & 0.00952 \\ -1 & 0 & 0.9 \end{bmatrix} \mathbf{v}(k) + \begin{bmatrix} 0.04356 \\ 0.8568 \\ 1 \end{bmatrix} u(k) \\
 y(k) &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{v}(k)
 \end{aligned}$$

The calculations above are implemented in the following MATLAB program:

```

>> % Use (5-38) to close the loop
A1=[1 0.0952 0;0 0.905 0;0 0 0.9];
B1=[0.0484;0.952;0]; B2=[0;0;1];
C1=[0 0 0.01]; C=[1 0 0]; D1=0.9; T=0.1;
A=A1+B1*C1+(-B2-D1*B1)*C % Closed-loop A matrix
B=D1*B1+B2 % Closed-loop B matrix
C=C % Closed-loop C matrix
D=0 % Closed-loop D matrix

A =
    0.9564    0.0952    0.0005
   -0.8568    0.9050    0.0095
   -1.0000         0    0.9000

B =
    0.0436
    0.8568
    1.0000

C =
    1     0     0

D =
    0
    
```

Next we will find the closed-loop transfer function  $T(z)$  and verify it by transfer function analysis.

```

>> % Continuing the program, find T(z) using the closed-loop
state matrices A, B, C, and D
T=0.1; [num,den]=ss2tf(A,B,C,D,1);
Tz=tf(num,den,T)
% Check the result by computing T(z) using G(z) and D(z)
Gp=tf([10],[1 1 0]);
Gz=c2d(Gp,T);
Dz=tf([0.9 -0.8],[1 -0.9],T);
Tz_check=feedback(Dz*Gz,1)

Tz =
      0.04356 z^2 + 0.003426 z - 0.03746
      -----
      z^3 - 2.761 z^2 + 2.623 z - 0.852

Tz_check =
      0.04354 z^2 + 0.00341 z - 0.03743
      -----
      z^3 - 2.761 z^2 + 2.623 z - 0.8518
  
```

Examples 5.6 and 5.7 illustrate the derivation of discrete state models for digital control systems. Of course, some systems are more complex than these in the examples above. However, the technique used to derive the state equations of (5-38) may be employed for more complex systems. For example, if  $y(k)$  in (5-34) is also a function of  $m(k)$ , the derivation is somewhat more complicated (see Problem 5.4-11).

## 5.5 SUMMARY

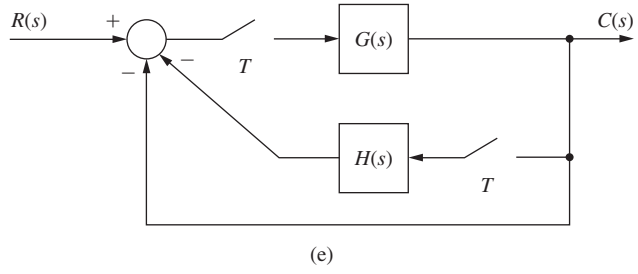
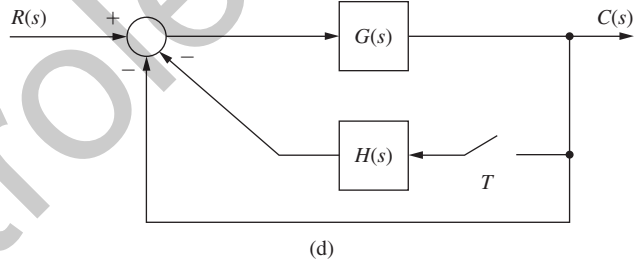
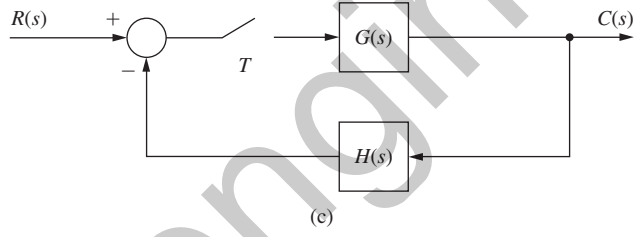
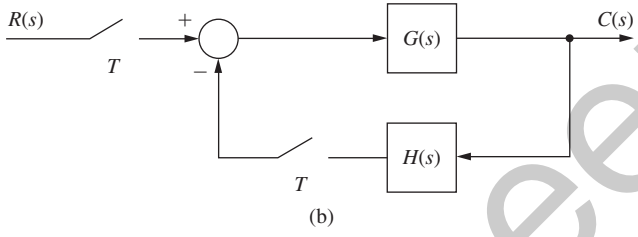
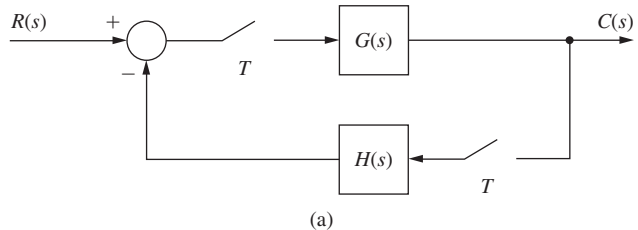
In this chapter, we have described a technique for finding the Laplace and  $z$ -transforms of the output of a closed-loop discrete-time system. The technique assumes the availability of a block diagram or signal flow graph of the discrete-time system. From this specification, a sampled signal flow graph is derived which can be used to determine the Laplace transform and the  $z$ -transform of the output of the closed-loop system. In addition, a technique is developed for determining the state-variable model of a closed-loop discrete-time system provided all inputs are sampled. In the following chapters these techniques will be utilized in analyzing and designing closed-loop discrete-time systems.

## References and Further Readings

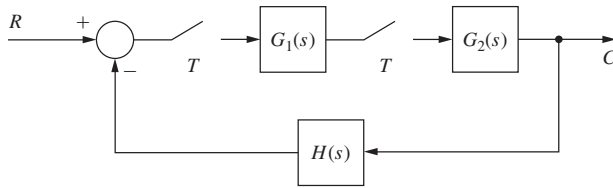
- [1] B. C. Kuo, *Digital Control Systems*, 2nd ed. New York: Oxford University Press, 1995.
- [2] C. L. Phillips and S. M. Seltzer, "Design of Advanced Sampled-Data Control Systems," Contract DAAHO1-72-C-0901, Auburn University, Auburn, AL, July 1973.
- [3] J. A. Cadzow and H. R. Martens, *Discrete-Time and Computer Control Systems*. Upper Saddle River, NJ: Prentice-Hall, 1970.
- [4] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Half Moon Bay, CA: Ellis-Kagle Press, 2006.
- [5] E. I. Jury, *Theory and Application of the z-Transform Method*. Huntington, NY: R.E. Krieger Publishing Co., Inc., 1973.
- [6] E. I. Jury, "Sampled-Data Systems Revisited: Reflections, Recollections, and Reassessments." *ASME J. Dynamic Systems, Measurement, and Control*, 102(4), pp. 208-217, 1980.

## Problems

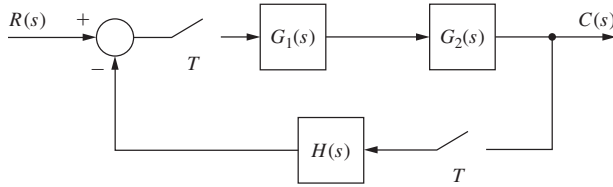
- 5.3-1.** For each of the systems of Fig. P5.3-1, express  $C(z)$  as a function of the input and the transfer functions shown.
- 5.3-2.** For each of the systems of Fig. P5.3-2, express  $C(z)$  as a function of the input and the transfer functions shown.
- 5.3-3.** (a) Derive the transfer function  $C(z)/R(z)$  for the system of Fig. P5.3-1(b).  
 (b) Derive the transfer function  $C(z)/R(z)$  for the system of Fig. P5.3-1(c).  
 (c) Even though the two systems are different, the transfer functions are equal. Explain why this is true.  
*Hint:* Consider the error signal.
- 5.3-4.** Consider the system of Fig. P5.3-4.  
 (a) Calculate the system output  $C(z)$  for the signal from  $G_2(s)$  disconnected from the middle summing junction.  
 (b) Calculate the system output  $C(z)$  for the signal from  $G_2(s)$  disconnected from the last summing junction.  
 (c) Calculate the system output  $C(z)$  for the system as shown.
- 5.3-5.** For the system of Fig. P5.3-1(a), suppose that the sampler in the forward path samples at  $t = 0, T, 2T, \dots$ , and the sampler in the feedback path samples at  $t = T/2, 3T/2, 5T/2, \dots$ .  
 (a) Find the system transfer function  $C(z)/R(z)$ .  
 (b) Suppose that both samplers operate at  $t = T/2, 3T/2, 5T/2, \dots$ . Find  $C(z)$  in its simplest form.
- 5.3-6.** The system of Fig. P5.3-6 contains a digital filter with the transfer function  $D(z)$ . Express  $\phi_m(z)$  as a function of the input. The roll-axis control system of the Pershing missile is of this configuration [2].
- 5.3-7.** Consider the two-loop system of Fig. P5.3-7. The gain  $K$  is used to give the inner loop certain specified characteristics. Then the controller  $D(z)$  is designed to compensate the entire system. The input  $R(z)$  is generated within the computer, and thus does not exist as a continuous signal. Solve for  $C(z)$ .
- 5.3-8.** The system of Fig. P5.3-8 is the same as that of Example 5.1, except that the sampler has been moved to the feedback path. This may occur for two reasons: (1) the sensor output is in sampled form and (2) the input function is more conveniently generated in the computer.  
 (a) Calculate both  $C(s)$  and  $C(z)$  for the system of Fig. P5.3-8. Note that  $C(s)$  cannot contain the variable  $z$ .  
 (b) Note that the output functions in part (a) are identical to those found in Example 5.1, even though the two configurations are different. Explain why.
- 5.3-9.** For Fig. P5.3-1, list, for each system, all transfer functions that contain the transfer function of a zero-order hold.
- 5.3-10.** For Fig. P5.3-2, list, for each system, all transfer functions that contain the transfer function of a zero-order hold.
- 5.3-11.** In the system of Fig. P5.3-11, the ideal time delay represents the time required to complete the computations in the computer.  
 (a) Derive the output function  $C(z)$  for this system.  
 (b) Suppose that the ideal delay is associated with the sensor rather than the computer, and the positions of  $H(s)$  and the ideal delay are reversed. Find  $C(z)$  for this case.



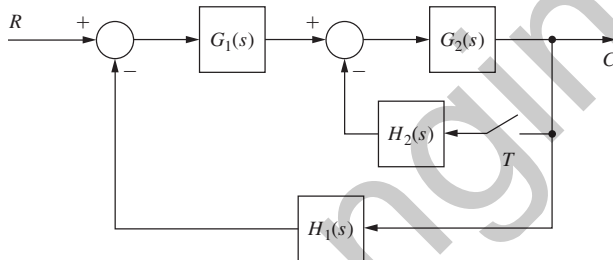
**FIGURE P5.3-1** Systems for Problem 5.3-1.



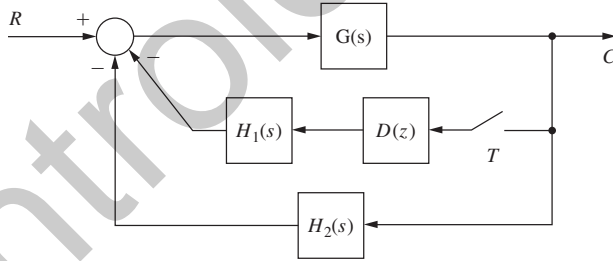
(a)



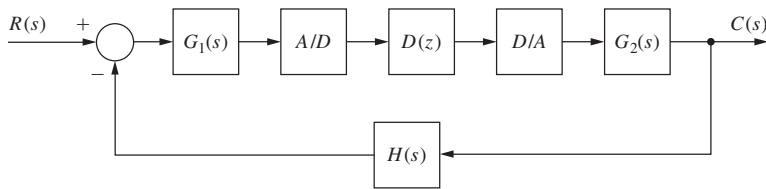
(b)



(c)

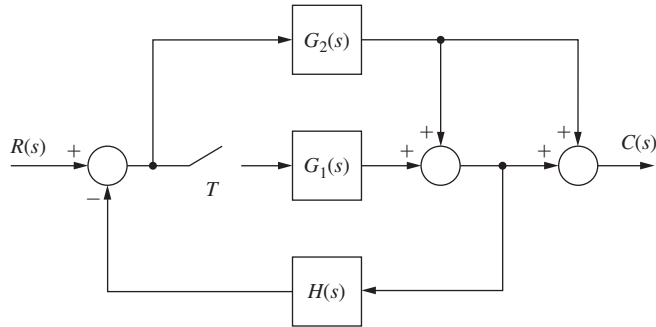


(d)

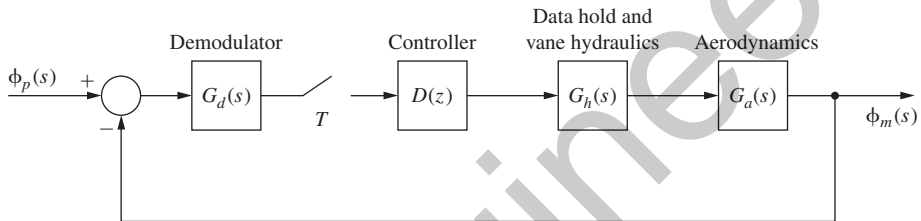


(e)

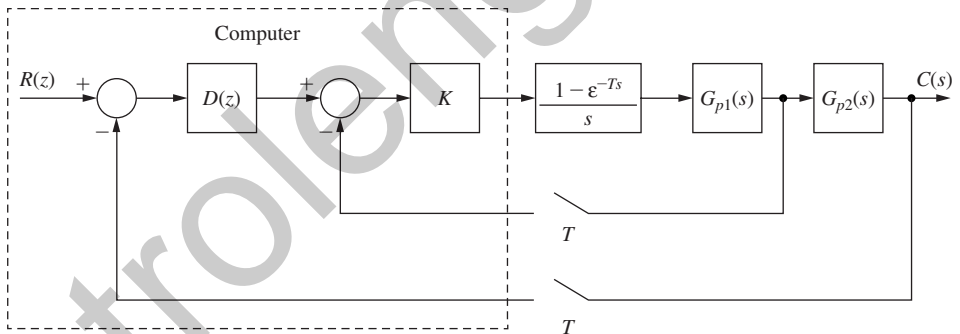
FIGURE P5.3-2 Systems for Problem 5.3-2.



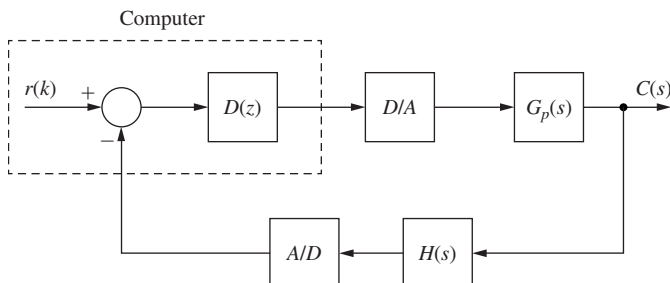
**FIGURE P5.3-4** Systems for Problem 5.3-4.



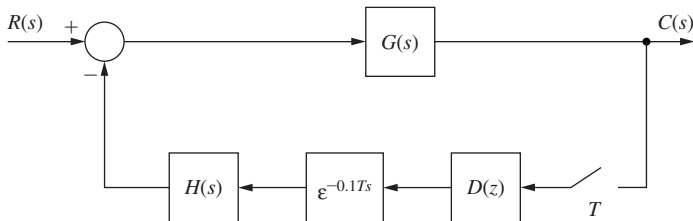
**FIGURE P5.3-6** Roll-axis control system for a Pershing missile.



**FIGURE P5.3-7** System for Problem 5.3-7.



**FIGURE P5.3-8** System for Problem 5.3-8.

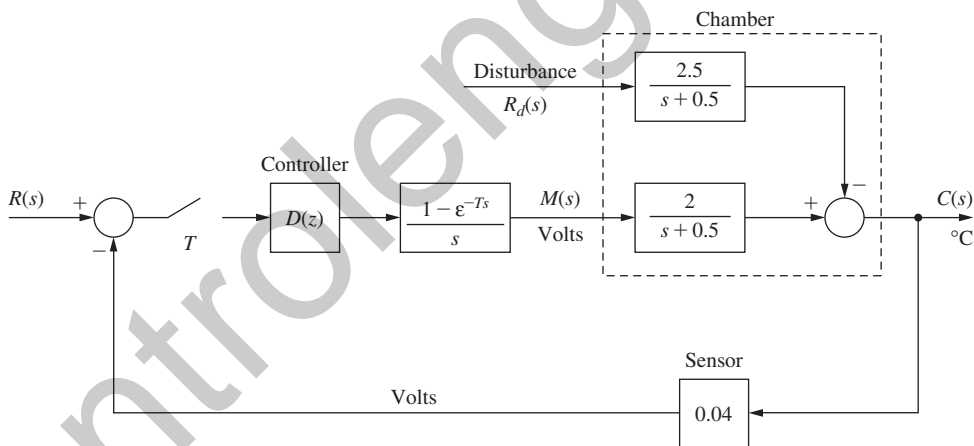


**FIGURE P5.3-11** System for Problem 5.3-11.

**5.3-12.** Shown in Fig. P5.3-12 is the block diagram for the temperature control system for a large test chamber. This system is described in Problem 1.6-1. The disturbance shown is the model of the effects of opening the chamber door. The following transfer functions are defined.

$$G(s) = \frac{2(1 - \epsilon^{-Ts})}{s(s + 0.5)}, \quad G_d(s) = \frac{2.5}{s + 0.5}, \quad H_k = 0.04$$

- Derive the transfer function  $C(z)/R(z)$ , in terms of the transfer functions just defined.
- With  $r(t) = 0$ , solve for the output function  $C(s)$  in terms of the disturbance input and the transfer functions just defined.
- Use superposition and the results of parts (a) and (b) to write the complete expression of  $C(z)$ .

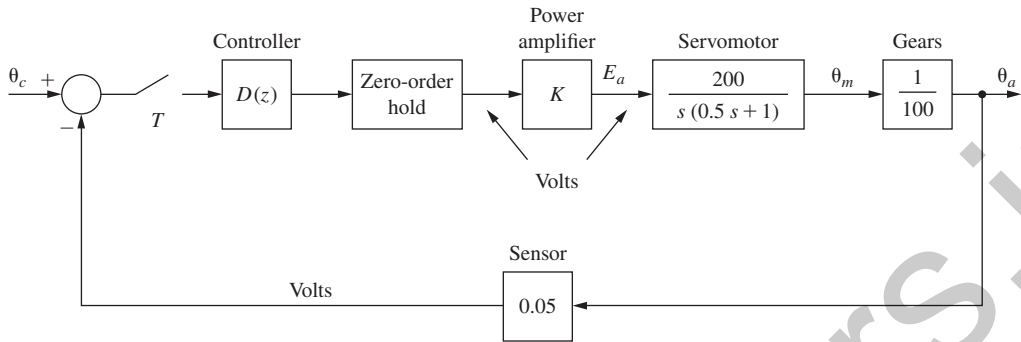


**FIGURE P5.3-12** Chamber temperature control system.

**5.3-13.** Consider the robot-joint control system of Fig. P5.3-13. This system is described in Problem 1.5-4.

- The sensor input is  $\theta_a$  in degrees and the output is in volts. If the robot joint movement is mechanically restricted to  $\pm 135^\circ$ , find the range of the sensor output voltage. What should be the input voltage range for the A/D?
- Let  $G_p(s)$  be the transfer function of the servomotor and gears, and  $H_k = 0.05$  be the sensor gain. Find the system transfer function as a function of  $K$ ,  $G_p(s)$ , and so on.
- Evaluate the system transfer function for  $K = 1.4$ ,  $T = 0.3$  s, and  $D(z) = 1.1$ .

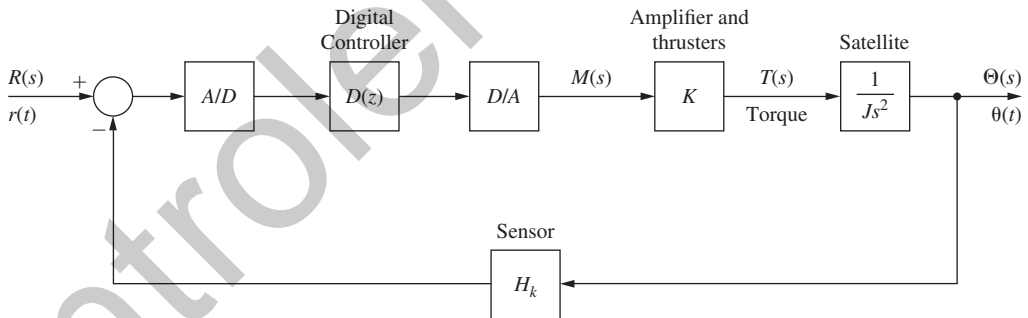




**FIGURE P5.3-13** Robot arm joint control system.

**5.3-14.** Consider the satellite control system of Fig. P5.3-14. The unit of the attitude angle  $\theta(t)$  is degrees, and the range is 0 to  $360^\circ$ . The sensor gain is  $H_k = 0.05$ .

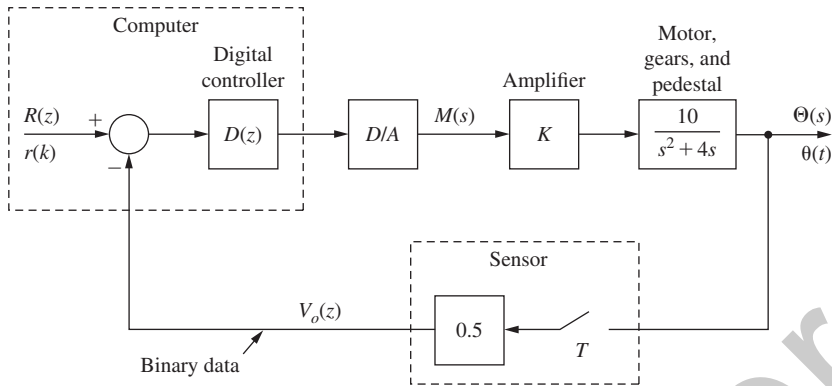
- Suppose that the input ranges for available A/Ds are 0 to 10 V, 0 to 20 V, 0 to 30 V,  $\pm 10$  V,  $\pm 20$  V, and  $\pm 30$  V. Which range should be chosen? Why?
- The input signal  $r(t)$  is a voltage. Find the required value of  $r(t)$  to command the satellite attitude angle  $\theta(t)$  to be  $80^\circ$ .
- Repeat parts (a) and (b) if the range of  $\theta(t)$  is  $\pm 180^\circ$ .
- Let  $G_p(s) = 1/(Js^2)$ , the satellite transfer function. Find the system transfer function as a function of the transfer functions  $D(z)$ ,  $K$ , and so on.
- Evaluate the system transfer function for  $D(z) = 1$ ,  $T = 0.5$  s,  $K = 4$ , and  $J = 0.2$ .



**FIGURE P5.3-14** Block diagram for a satellite control system.

**5.3-15.** Consider the antenna control system of Fig. P5.3-15. The unit of the antenna angle  $\theta(t)$  is degrees, and the range is  $\pm 135^\circ$ .

- The input signal  $r(k)$  is generated in the computer. Find the required values of  $r(k)$  to command the satellite attitude angle  $\theta(t)$  to be  $\pm 60^\circ$ .
- Let  $G_p(s) = 10/(s^2 + 4s)$ , the transfer function of the motor, gears, and pedestal. Find the system transfer function as a function of the transfer functions  $D(z)$ ,  $K$ , and so on.
- Evaluate the system transfer function for  $D(z) = 1$ ,  $T = 0.05$  s, and  $K = 20$ .
- Verify the results in part (c) using MATLAB.



**FIGURE P5.3-15** Block diagram for an antenna control system.

**5.4-1.** Given a closed-loop system described by the transfer function

$$\frac{C(z)}{R(z)} = \frac{z^2 + 0.6z + 0.4}{z^2 - 2z + 1.8}$$

- Express  $c(k)$  as a function of  $r(k)$ , as a single difference equation.
- Find a set of state equations for this system.
- Calculate the transfer function from the results of part (b), to verify these results.
- Verify the results in part (c) using MATLAB.

**5.4-2.** Repeat Problem 5.4-1 for each of the transfer functions

- $\frac{C(z)}{R(z)} = \frac{0.5z}{z - 1.8}$
- $\frac{C(z)}{R(z)} = \frac{0.4}{z - 0.16}$
- $\frac{C(z)}{R(z)} = \frac{0.1z - 0.05}{z - 0.9}$
- $\frac{C(z)}{R(z)} = \frac{0.27z + 0.5}{z^2 - 1.6z + 0.8}$

**5.4-3.** For the system of Fig. P5.3-1(a), let  $T = 0.5$  s and

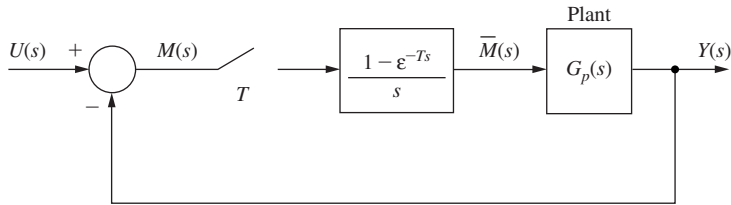
$$G(s) = \frac{1 - e^{-Ts}}{s^2}, \quad H(s) = \frac{1 - e^{-Ts}}{s^2 + s}$$

- Calculate  $G(z)$  and  $H(z)$ .
- Draw simulation diagrams for  $G(z)$  and  $H(z)$ , and interconnect these diagrams to form the control system of Fig. P5.3-1(a).
- Write the discrete state equations for part (b).
- Find the system characteristic equation, using the transfer functions of part (a).
- Show that the state model in part (c) has the same characteristic equation as in part (d).

**5.4-4.** Let  $T = 0.5$  s for the system of Fig. P5.4-4. Derive a set of discrete state equations for the closed-loop system, for the plant described by each of the differential equations.

- $\frac{dy(t)}{dt} + 5y(t) = 3\bar{m}(t)$

Choose the state variable to be  $y(kT)$ .



**FIGURE P5.4-4** System for Problem 5.4-4.

(b) 
$$\frac{d^2y(t)}{dt^2} + 9\frac{dy(t)}{dt} + 8y(t) = 4\bar{m}(t)$$

Choose the state variables to be  $y(kT)$  and  $dy(t)/dt|_{t=kT}$ .

**5.4-5.** Suppose that the plant in Fig. P5.4-4 has the discrete state model

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}m(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}m(k) \end{aligned}$$

Derive the state model for the closed-loop system, in terms of **A**, **B**, **C**, and **D**.

**5.4-6.** Find a discrete state variable model of the closed-loop system shown in Fig. P5.4-4 if the discrete state model of the plant is given by:

(a) 
$$\begin{aligned} x(k+1) &= 0.7x(k) + 0.3m(k) \\ y(k) &= 0.2x(k) + 0.5m(k) \end{aligned}$$

(b) 
$$\mathbf{x}(k+1) = \begin{bmatrix} 0 & 1 \\ -0.9 & 1.3 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.1 \\ 0.05 \end{bmatrix} m(k)$$

$$y(k) = [1.2 \quad -0.7] \mathbf{x}(k)$$

(c) 
$$\mathbf{x}(k+1) = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.9 & 1 \\ -1 & 0 & 0.9 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} m(k)$$

$$y(k) = [1 \quad 0 \quad 0] \mathbf{x}(k)$$

**5.4-7.** Consider the temperature control system of Problem 5.3-12 and Fig. P5.3-12. Suppose that the digital filter transfer function is given by

$$D(z) = 1.1 + \frac{0.02z}{z-1}$$

which is a PI (proportional-integral) controller. Suppose that  $T = 0.6$  s, and let  $R_d(s) = 0$  (ignore the disturbance input).

- Using the closed-loop transfer function, derive a discrete state model for the system.
- Derive a discrete state model for the plant from the plant transfer function. Then derive the state model of the closed-loop system by adding the filter and the feedback path to the flow graph of the plant.
- Calculate the transfer function from the state model of part (b), to verify these results.
- How are the states of parts (a) and (b) related? Do not solve for the exact relationship.

- 5.4-8.** Consider the satellite control system of Problem 5.3-14 and Fig. P5.3-14. Let  $D(z) = 2$ ,  $T = 2$  s,  $K = 4$ ,  $J = 0.2$ , and  $H_k = 0.04$ .
- Using the closed-loop transfer function, derive a discrete state model for the system.
  - Derive a discrete state model for the plant from the plant transfer function. Then derive a state model for the closed-loop system by adding the feedback path and system input to a flow graph of the plant.
  - Calculate the system transfer function from the state model found in part (b), to verify the state model.
  - Verify the results in part (c) using MATLAB.
- 5.4-9.** Consider the antenna control system of Problem 5.3-15 and Fig. P5.3-15. Let  $D(z) = 1$ ,  $T = 0.05$  s, and  $K = 20$ .
- Using the closed-loop transfer function, derive a discrete state model for the system in the observer canonical form of Fig. 2-10.
  - Derive a discrete state model for the plant from the plant transfer function. Then derive a state model for the closed-loop system by adding the feedback path and system input to a flow graph of the plant.
  - Calculate the system transfer function from the state model found in part (b), to verify the state model.
  - Verify the results in part (c) using MATLAB.
- 5.4-10.** Consider the robot joint control system of Problem 5.3-13 and Fig. P5.3-13. Let  $D(z) = 1$ ,  $T = 0.1$  s, and  $K = 2.4$ .
- Using the closed-loop transfer function, derive a discrete state model for the system.
  - Derive a discrete state model for the plant from the plant transfer function. Then derive a state model for the closed-loop system by adding the feedback path and system input to a flow graph of the plant.
  - Calculate the system transfer function from the state model found in part (b), to verify the state model.
- 5.4-11.** Suppose that, for the system of Fig. 5-20, equation (5-34) is
- $$y(k) = \mathbf{Cv}(k) + d_2m(k)$$
- Derive the state model of (5-37) for this case.
  - This system has an algebraic loop. Identify this loop.
  - The gain of the algebraic loop is  $-d_1d_2$ . What is the effect on the system equations if  $d_1d_2 = -1$ ?
  - We can argue that algebraic loops as in this case cannot occur in physical systems, since time delay is always present in signal transmission. Quite often we can ignore this delay. Under what conditions can we obviously *not* ignore delay in this system?
- 5.4-12.** Consider the satellite control system of Problem 5.3-14 and Fig. P5.3-14. Let  $D(z) = (z - 1)/(z - 0.5)$ ,  $T = 1$  s,  $K = 2$ ,  $J = 0.1$ , and  $H_k = 0.02$ . Use MATLAB to compute:
- A discrete state model for the system using the closed-loop transfer function.
  - A discrete state model for the plant from the plant transfer function. Then compute a state model for the closed-loop system by adding  $D(z)$  and including the feedback path and system input.
  - Calculate the system transfer function from the state model found in part (b) and compare it with part (a) to verify the state model.
- 5.4-13.** Consider the antenna control system of Problem 5.3-15 and Fig. P5.3-15. Let  $D(z) = (4z - 3)/(z - 0.5)$ ,  $T = 0.05$  s, and  $K = 20$ .
- A discrete state model for the system using the closed-loop transfer function.
  - A discrete state model for the plant from the plant transfer function. Then compute a state model for the closed-loop system by adding  $D(z)$  and including the feedback path and system input.
  - Calculate the system transfer function from the state model found in part (b) and compare it with part (a) to verify the state model.

- 5.4-14.** Consider the robot joint control system of Problem 5.3-13 and Fig. P5.3-13. Let  $D(z) = (24z - 21)/(z - 0.5)$ ,  $T = 0.1\text{s}$ , and  $K = 2.4$ .
- A discrete state model for the system using the closed-loop transfer function.
  - A discrete state model for the plant from the plant transfer function. Then compute a state model for the closed-loop system by adding  $D(z)$  and including the feedback path and system input.
  - Calculate the system transfer function from the state model found in part (b) and compare it with part (a) to verify the state model.

Controlengineers.ir

# 6

## System Time-Response Characteristics

### 6.1 INTRODUCTION

In this chapter we consider five important topics. First, the time response of a discrete-time system is investigated. Next, regions in the  $s$ -plane are mapped into regions in the  $z$ -plane. Then by using the correlation between regions in the two planes, the effect of the closed-loop  $z$ -plane poles on the system transient response is discussed. Next, the effects of the system transfer characteristics on the steady-state system error are considered. Finally, the simulations of analog and discrete-time systems are introduced.

### 6.2 SYSTEM TIME RESPONSE

In this section the time response of discrete-time systems is introduced via examples. In these examples some of the techniques of determining the system time response are illustrated.

#### EXAMPLE 6.1

The unit-step response will be found for the first-order system in Fig. 6-1(a). Since the plant of a temperature control system is often modeled as a first-order system, this system might then be the model of a temperature control system (see Section 1.6). Using the techniques developed in Chapter 5, we can express the system output as

$$C(z) = \frac{G(z)}{1 + G(z)}R(z)$$

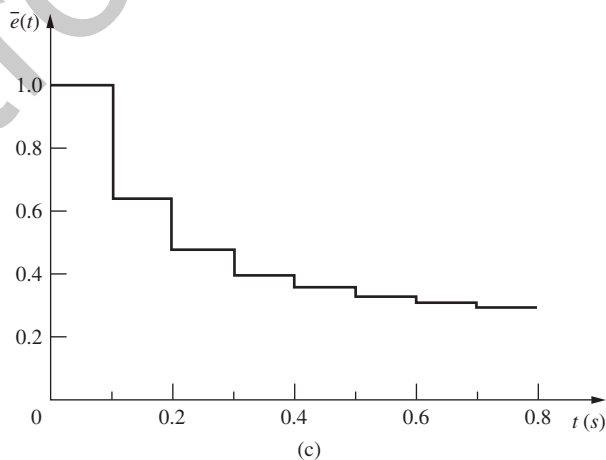
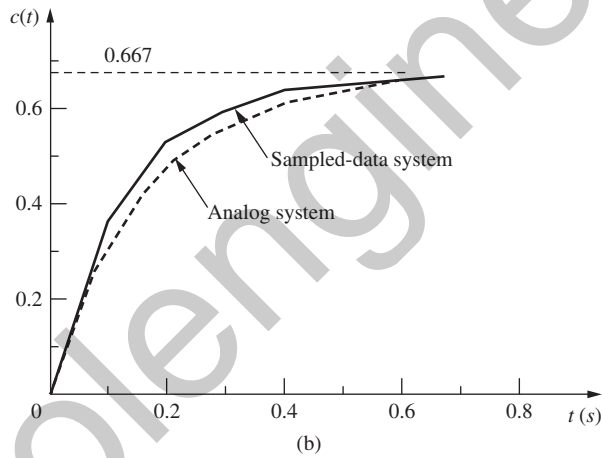
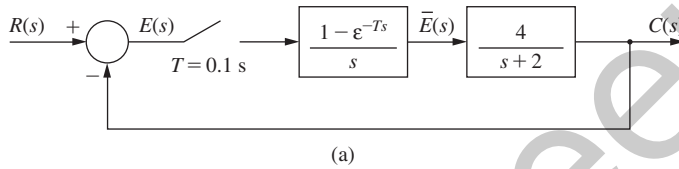
where  $G(z)$  is defined as

$$G(z) = \mathcal{Z} \left[ \frac{1 - e^{-Ts}}{s} \frac{4}{s + 2} \right] = \frac{z - 1}{z} \mathcal{Z} \left[ \frac{4}{s(s + 2)} \right]$$

$$= \frac{z-1}{z} \frac{2(1-\epsilon^{-2T})z}{(z-1)(z-\epsilon^{-2T})} = \frac{0.3625}{z-0.8187}, \quad T=0.1 \text{ s}$$

from the transform table in Appendix VI. Thus the closed-loop transfer function  $T(z)$  is given by

$$T(z) = \frac{G(z)}{1+G(z)} = \frac{0.3625}{z-0.4562}$$



**FIGURE 6-1** System and response for Example 6.1.

Since  $R(z) = \mathcal{Z}[1/s] = z/(z - 1)$ ,

$$C(z) = \frac{0.3625z}{(z - 1)(z - 0.4562)} = \frac{0.667z}{z - 1} + \frac{-0.667z}{z - 0.4562}$$

The inverse  $z$ -transform of this function yields the system time response at the sampling instants. Thus

$$c(kT) = 0.667[1 - (0.4562)^k]$$

This response is listed in Table 6-1. It is seen that the response reaches a steady-state value of 0.667. The computations above may be done in MATLAB:

```

>> T = 0.1; Gs = tf([4], [1 2]); Gz = c2d(Gs, T);
Tz = feedback(Gz, 1); Rz = tf([1 0], [1 -1], T);
Cz = zpkr(Rz*Tz)

Cz =
    0.3625 z
-----
(z-1) (z-0.4562)

>> % Partial fraction expansion of C(z)/z
[r,p,k]=residue([0.3625], [1 -1.4562 0.4562])

r =
    0.6666
   -0.6666

p =
    1.0000
    0.4562

k =
    []
    
```

### EXAMPLE 6.2

This example is a continuation of Example 6.1. To show some of the effects of sampling on the system response, we will remove the sampler and zero-order hold, and solve for the unit-step response of the resulting analog system. The closed loop transfer function  $T_a(s)$  is given by

$$T_a(s) = \frac{G_p(s)}{1 + G_p(s)} = \frac{4}{s + 6}$$

where  $G_p(s) = 4/(s + 2)$  is the plant transfer function. Hence the analog system unit-step response is given by

$$C_a(s) = \frac{4}{s(s + 6)} = \frac{0.667}{s} + \frac{-0.667}{s + 6}$$

and

$$c_a(t) = 0.667(1 - e^{-6t})$$

This response is also listed in Table 6-1. Both step responses are plotted in Fig. 6-1(b).



**TABLE 6-1 Responses for Example 6.1**

$kT$	$c(kT)$	$c_a(t)$
0	0	0
0.1	0.363	0.300
0.2	0.528	0.466
0.3	0.603	0.557
0.4	0.639	0.606
0.5	0.654	0.634
0.6	0.661	0.648
$\vdots$	$\vdots$	$\vdots$
1.0	0.666	0.665

We may also calculate the response at all instants of time for the sampled-data system. The continuous output of the system of Fig. 6-1(a) is given by

$$\begin{aligned}
 C(s) &= G(s)E^*(s) = G(s) \left[ \frac{R^*(s)}{1 + G^*(s)} \right] = G(s) \left[ \frac{R(z)}{1 + G(z)} \right]_{z=\epsilon^{Ts}} \\
 &= \frac{4(1 - \epsilon^{-Ts})}{s(s + 2)} \left[ \frac{\frac{z}{z - 1}}{1 + G(z)} \right]_{z=\epsilon^{Ts}} = \frac{4}{s(s + 2)} \left[ \frac{1}{1 + G(z)} \right]_{z=\epsilon^{Ts}}
 \end{aligned}$$

In this expression,

$$\begin{aligned}
 \frac{1}{1 + G(z)} &= \frac{1}{1 + \frac{0.3625}{z - 0.8187}} = \frac{z - 0.8187}{z - 0.4562} \\
 &= 1 - 0.363z^{-1} - 0.165z^{-2} - \dots
 \end{aligned}$$

In  $C(s)$  above, let the first factor be denoted as  $C_1(s)$ , that is,

$$C_1(s) = \frac{4}{s(s + 2)} = \frac{2}{s} - \frac{2}{s + 2}$$

Then

$$c_1(t) = 2(1 - \epsilon^{-2t})$$

Thus the output  $C(s)$  is

$$C(s) = C_1(s)[1 - 0.363\epsilon^{-Ts} - 0.165\epsilon^{-2Ts} - \dots]$$

and hence

$$\begin{aligned}
 c(t) &= 2(1 - \epsilon^{-2t}) - 0.363(2)(1 - \epsilon^{-2(t-T)})u(t - T) \\
 &\quad - 0.165(2)(1 - \epsilon^{-2(t-2T)})u(t - 2T) - \dots
 \end{aligned}$$

For example, since  $T = 0.1$  s,

$$\begin{aligned}
 c(3T) &= c(0.3) = 2(1 - \epsilon^{-0.6}) - 0.363(2)(1 - \epsilon^{-0.4}) \\
 &\quad - 0.165(2)(1 - \epsilon^{-0.2}) = 0.603
 \end{aligned}$$

This value checks that calculated by the  $z$ -transform approach and listed in Table 6-1. We see then the reason for the unusual shape of  $c(t)$  in Fig. 6-1(b). This response is the superposition of a number of delayed step responses of the open-loop system. The steps appear as a result of the sampler and zero-order hold. For example, for  $0 \leq t < 0.1$  s,

$$c(t) = 2(1 - \epsilon^{-2t})$$

Note that the time response of a sampled-data system of the configuration of that in Fig. 6-1(a) is always the superposition of a number of step responses, independent of the form of the system input function. The steps in the input to the plant are also shown in the plot of the zero-order hold output,  $\bar{e}(t)$ , in Fig. 6-1(c).

Note the difficulty in calculating the continuous output  $c(t)$  as compared with calculating the output  $c(kT)$ . For this reason, we seldom calculate the continuous output; if  $c(t)$  is needed, it is obtained by simulation. In fact, in practical situations, we generally calculate all time responses by simulation. Simulation is introduced in Section 6.6.

### EXAMPLE 6.3

We will consider the system of Examples 6.1 and 6.2 further. Recall that the response of a first-order analog system has the transient-response term  $ke^{-t/\tau}$ , where  $\tau$  is the *time constant*. We see then that the analog system of Fig. 6-1(a) has a time constant of 0.167s. A *rule of thumb* often used for selecting sample rates is that a rate of *at least five samples per time constant* is a good first choice. (Later results from experimentation with the system model and the actual physical system may indicate that a different rate is required.) For this system, we expect that reducing the sample period would decrease the effects of sampling, and that the sampled-data system characteristics would approach those of the analog system. In fact, if  $T$  is chosen to be 0.04 s, the unit-step response of the sampled-data system of Example 6.1 is essentially the same as that of the analog system. This point is illustrated further in the next example.

As an additional point, the final value of the unit-step response of the sampled-data system can be calculated using the final value theorem of the  $z$ -transform.

$$\begin{aligned}
 \lim_{n \rightarrow \infty} c(nT) &= [(z - 1)C(z)]|_{z=1} = \left[ (z - 1) \frac{G(z)}{1 + G(z)} R(z) \right] \Big|_{z=1} \\
 &= \left[ (z - 1) \frac{G(z)}{1 + G(z)} \frac{z}{z - 1} \right] \Big|_{z=1} = \frac{zG(z)}{1 + G(z)} \Big|_{z=1} \\
 &= \frac{G(1)}{1 + G(1)} = \frac{2}{1 + 2} = 0.667
 \end{aligned}$$

This value checks that in Table 6-1. Note that this derivation is general. Since the system input is a constant value of unity, we see from this derivation that the dc gain of a stable sampled-data system is simply the closed-loop transfer function evaluated at  $z = 1$  (see Section 4.3).

A final point will be made concerning this example. For a continuous-time system whose output is

$$C(s) = G_p(s)E(s)$$

the dc gain is given by (see Section 4.3)

$$\text{dc gain} = \lim_{s \rightarrow 0} G_p(s)$$

and this value is  $G_p(0)$  if  $c(t)$  has a final value for a constant input. In the system of Fig. 6-1(a), if the input to the sampler is constant, the output of the zero-order hold is also constant and equal to the sampler input. Thus the sampler and data hold have no effect and may be removed, resulting in a continuous-time system. Thus, for a stable sampled-data system, the system dc gain may be found by removing the sampler and data hold, and evaluating the resulting system transfer function at  $s = 0$ . For this example, the open-loop dc gain is

$$G_p(s)|_{s=0} = \left. \frac{4}{s+2} \right|_{s=0} = 2$$

Thus the closed-loop dc gain is  $2/(1+2) = 0.667$ , which agrees with the dc gain calculated above via the  $z$ -transform.

Two important points were made in this example concerning the calculation of the steady-state gain of a sampled-data system with a constant input applied (i.e., the calculation of the dc gain). First, for a stable system with a constant input, the system output approaches a constant value as time increases. The dc gain may be calculated by evaluating the transfer function with  $z = 1$ . Second, the same value of dc gain is obtained by evaluating the transfer function of the analog system (sampler and zero-order hold removed) with  $s = 0$ . This second calculation applies in any case that the input to the sampler is constant.

### EXAMPLE 6.4

The system for this example is shown in Fig. 6-2. As in the first example, we will calculate the unit-step response. This system will also appear in many of the following examples. As was demonstrated in Chapter 5, the system output can be expressed as

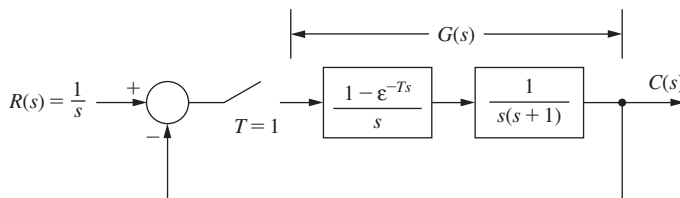


FIGURE 6-2 System used in Example 6.4.

$$C(z) = \frac{G(z)}{1 + G(z)}R(z)$$

where, from the tables in Appendix VI,

$$G(z) = \left(\frac{z-1}{z}\right) \mathcal{Z}\left[\frac{1}{s^2(s+1)}\right]_{T=1} = \frac{z-1}{z} \left[ \frac{z[(1-1+\epsilon^{-1})z + (1-\epsilon^{-1}-\epsilon^{-1})]}{(z-1)^2(z-\epsilon^{-1})} \right]$$

$$= \frac{0.368z + 0.264}{z^2 - 1.368z + 0.368}$$

Then

$$\frac{G(z)}{1 + G(z)} = \frac{0.368z + 0.264}{z^2 - z + 0.632}$$

Since

$$R(z) = \frac{z}{z-1}$$

then

$$C(z) = \frac{z(0.368z + 0.264)}{(z-1)(z^2 - z + 0.632)} = 0.368z^{-1} + 1.00z^{-2} + 1.40z^{-3}$$

$$+ 1.40z^{-4} + 1.15z^{-5} + 0.90z^{-6} + 0.80z^{-7} + 0.87z^{-8}$$

$$+ 0.99z^{-9} + 1.08z^{-10} + 1.08z^{-11} + 1.03z^{-12} + 0.98z^{-13}$$

$$+ \dots \tag{6-1}$$

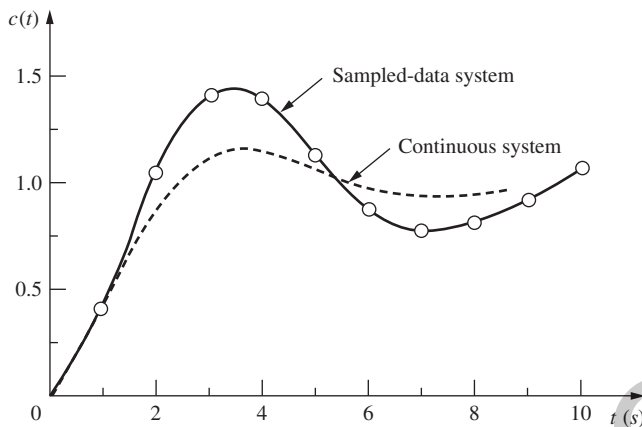
The final value of  $c(nT)$ , obtained using the final-value theorem, is

$$\lim_{n \rightarrow \infty} c(nT) = \lim_{z \rightarrow 1} (z-1)C(z) = \lim_{z \rightarrow 1} \left[ (z-1) \frac{z(0.368z + 0.264)}{(z-1)(z^2 - z + 0.632)} \right] = \frac{0.632}{0.632} = 1$$

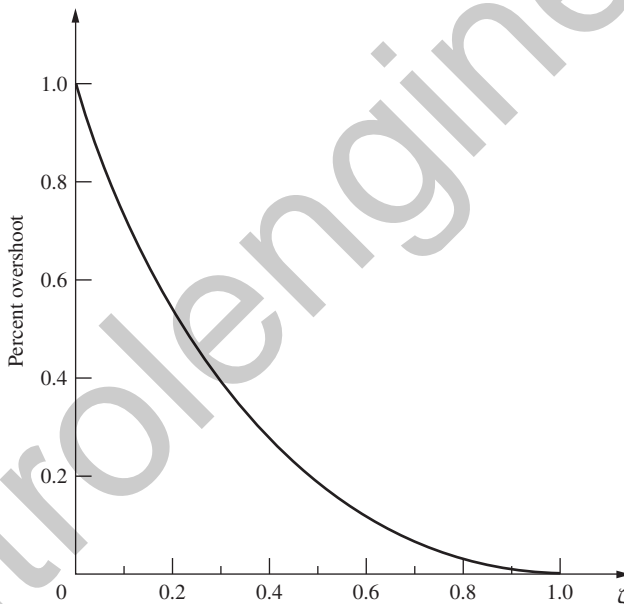
The step response for this system is plotted in Fig. 6-3. The response between sampling instants was obtained from a simulation of the system, and, of course, the response at the sampling instants is given in (6-1). Also plotted in Fig. 6-3 is the response of the system with the sampler and data hold removed. For this continuous-time system, the transfer function is, in standard notation [1],

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{1}{s^2 + s + 1}$$

Therefore,  $\omega_n = 1$ ,  $\zeta = 0.5$ , and hence the overshoot is approximately 18% (see Fig. 6-4 below). Thus, as shown in Fig. 6-3, the sampling has a destabilizing effect on the system. In general, it is desirable that the effects of sampling be negligible, that is, that the continuous-time system response and the discrete-time system response be approximately equal. For this system



**FIGURE 6-3** Step response of the systems analyzed in Example 6.4.



**FIGURE 6-4** Percent overshoot versus  $\zeta$ .

the sampling period and the plant time constant are equal. Hence the sampling frequency is too low, and should be increased, if allowed by hardware constraints. These effects will be discussed in detail later.

The system time response can also be calculated using a difference-equations approach. From the expression for the closed-loop transfer function

$$\frac{C(z)}{R(z)} = \frac{0.368z^{-1} + 0.264z^{-2}}{1 - z^{-1} + 0.632z^{-2}} \quad (6-2)$$

or

$$C(z)[1 - z^{-1} + 0.632z^{-2}] = R(z)[0.368z^{-1} + 0.264z^{-2}] \quad (6-3)$$

Taking the inverse  $z$ -transform of (6-3), we obtain the difference equation

$$c(kT) = 0.368 r(kT - T) + 0.264 r(kT - 2T) + c(kT - T) - 0.632 c(kT - 2T) \quad (6-4)$$

Both  $c(kT)$  and  $r(kT)$  are zero for  $k < 0$ . Thus, from (6-4),  $c(0) = 0$  and  $c(1) = 0.368$ . For  $k \geq 2$ , (6-4) becomes

$$c(kT) = 0.632 + c(kT - T) - 0.632 c(kT - 2T)$$

Solving (6-4) for  $c(kT)$  yields the same values as found in (6-1). A MATLAB program that solves (6-4) for the first 14 values of  $c(kT)$  is given below. Note the comparison with (6-1).

```
>> % Note that r=r(kT), rm1=r(kT-T), rm2=r(kT-2T), etc.
rm1=0; rm2=0; cm1=0; cm2=0;
for kk=1:14
    k=kk-1;
    r=1;
    c=0.368*rm1+0.264*rm2+cm1-0.632*cm2;
    [k,c]
    cm2=cm1; cm1=c; rm2=rm1; rm1=r; %Time delay
end

ans =
0 0
1.0000 0.3680
2 1
.
.
.
11.0000 1.0809
12.0000 1.0323
13.0000 0.9812
```

An additional point will be made concerning Example 6.4. The *damping ratio*  $\zeta$  is an important indicator for complex poles in a transfer function. If the transfer function is second order of the standard form

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where  $\omega_n$  is the *natural frequency*, the percent overshoot in the step response is given by [1]:

$$\text{Percent overshoot} = \epsilon^{-\zeta\pi/\sqrt{1-\zeta^2}} \times 100$$

This relationship is plotted in Fig. 6-4. For systems that are not modeled as the standard second-order transfer function, Fig. 6-4 gives an indication of the overshoot (or looseness) that will appear in a system response if the system transfer function has complex poles.

In this section the time responses of two different sampled-data systems were calculated using a  $z$ -transform approach. In addition, the continuous output was calculated for the first

example, using the Laplace transform. Because of the complexities involved, it should be evident to the reader that in general we do not calculate the continuous output. In fact, system response is normally determined from either a digital simulation or a hybrid simulation of the system, and not from a transform approach. For high-order systems, simulation is the only practical technique for calculating the time response.

In many of the examples presented, the sampling frequency has been purposely chosen low, for two reasons. First, a sample period of 1 s is often selected to make the numerical calculations simpler. Second, if the sample period is large, only a few terms in the series expansion of  $C(z)$  are required to give a good indication of the character of the response. In Example 6.4, if  $T = 0.1$  s, the response of the sampled-data system is approximately the same as that of the continuous-time system. However, for  $T = 0.1$  s, 21 terms in the series for  $C(z)$  are required to obtain the system response from  $t = 0$  to  $t = 2$  s. Hence the only practical technique of calculating the system response for this system is by simulation (which is usually the case).

### 6.3 SYSTEM CHARACTERISTIC EQUATION

Consider a single-loop sampled-data system of the type shown in Fig. 6-5. For this system,

$$C(z) = \frac{G(z)R(z)}{1 + \overline{GH}(z)} = \frac{K \prod_{i=1}^m (z - z_i)}{\prod_{i=1}^n (z - p_i)} R(z)$$

Using the partial-fraction expansion, we can express  $C(z)$  as

$$C(z) = \frac{k_1 z}{z - p_1} + \cdots + \frac{k_n z}{z - p_n} + C_R(z) \quad (6-5)$$

where  $C_R(z)$  contains the terms of  $C(z)$  which originate in the poles of  $R(z)$ . The first  $n$  terms of (6-5) are the natural response terms of  $C(z)$  [i.e., the terms that are always present in  $C(z)$ ]. If the system is stable, these terms are also called the *transient response*. The inverse  $z$ -transform of the  $i$ th term yields

$$\mathfrak{Z}^{-1} \left[ \frac{k_i z}{z - p_i} \right] = k_i (p_i)^k$$

It is seen that these terms determine the nature, or character, of the system natural response. Since the  $p_i$  originates in the roots of the equation

$$1 + \overline{GH}(z) = 0$$

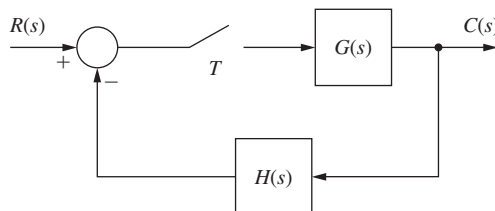


FIGURE 6-5 Single-loop sampled-data systems.

this equation is then the *system characteristic equation*. The roots of the characteristic equation are the poles of the closed-loop transfer function. If a transfer function cannot be written, the roots of the characteristic equation are the poles of  $C(z)$  that are independent of the input function.

### 6.4 MAPPING THE $s$ -PLANE INTO THE $z$ -PLANE

In studying the characteristics of analog systems, we are able to assign time-response characteristics to closed-loop pole locations (characteristic-equation zero locations) [1]. It is desirable to be able to do the same for sampled-data systems, and this topic is discussed in this section.

To introduce the topic, consider a function  $e(t)$ , which is sampled with the resulting starred transform  $E^*(s)$ . At the sampling instants, the sampled signal is of the same nature (and has the same values) as the continuous signal. For example, if  $e(t)$  is exponential, then the sampled signal is exponential at the sampling instants, with the same amplitude and time constant as the continuous function. If  $e(t) = \epsilon^{-at}$ ,

$$E(s) = \frac{1}{s + a}, \quad E^*(s) = \frac{\epsilon^{Ts}}{\epsilon^{Ts} - \epsilon^{-aT}}, \quad E(z) = \frac{z}{z - \epsilon^{-aT}}$$

Hence an  $s$ -plane pole at  $s = -a$  results in the  $z$ -plane pole at  $z = \epsilon^{-aT}$ . In general, from the  $z$ -transform tables of Appendix VI, we see that a pole of  $E(s)$  at  $s = s_1$  results in a  $z$ -plane pole of  $E(z)$  at  $z_1 = \epsilon^{s_1 T}$ . This characteristic is also evident from the second property of the starred transform given in Section 3.6 and from (4-4). We will use the inverse of this characteristic. A  $z$ -plane pole at  $z = z_1$  results in the transient-response characteristics at the sampling instants of the equivalent  $s$ -plane pole  $s_1$ , where  $s_1$  and  $z_1$  are related by  $z_1 = \epsilon^{s_1 T}$ .

Consider first the mapping of the left half-plane portion of the primary strip into the  $z$ -plane as shown in Fig. 6-6. Along the  $j\omega$  axis,

$$z = \epsilon^{sT} = \epsilon^{\sigma T} \epsilon^{j\omega T} = \epsilon^{j\omega T} = \cos(\omega T) + j \sin(\omega T) = 1 \angle \omega T$$

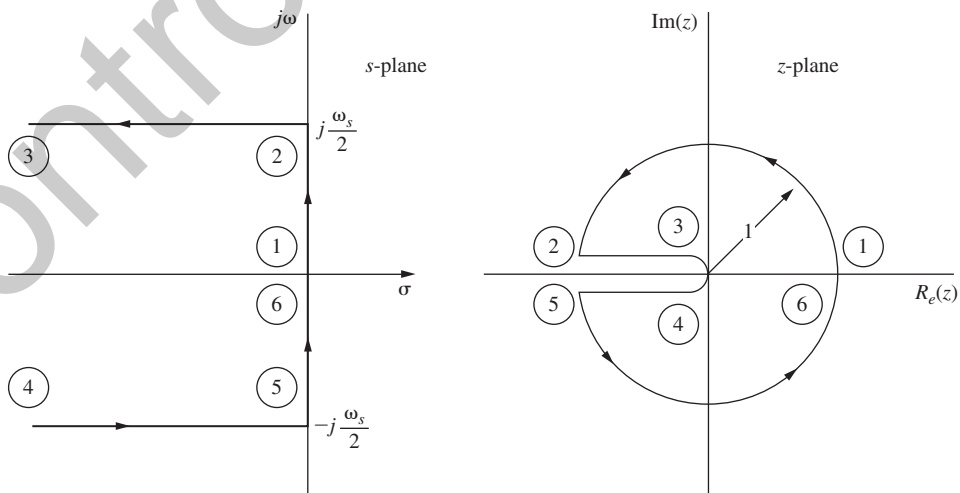


FIGURE 6-6 Mapping the primary strip into the  $z$ -plane.



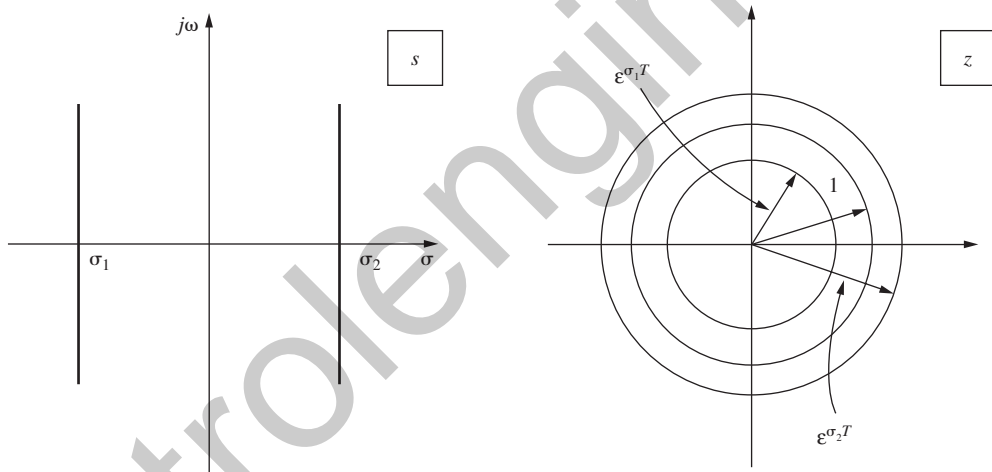
Hence poles located on the unit circle in the  $z$ -plane are equivalent to pole locations on the imaginary axis in the  $s$ -plane. Thus pole locations on the unit circle in the  $z$ -plane signify a system with a steady-state oscillation in its natural response. From the equation above, the frequency of oscillation is given by the angle of the pole (in radians) divided by  $T$ .

For  $\omega = \omega_s/2$ ,  $\omega T$  is equal to  $\pi$ , and hence the  $j\omega$  axis between  $-j\omega_s/2$  and  $j\omega_s/2$  maps into the unit circle in the  $z$ -plane. In fact, any portion of the  $j\omega$  axis of length  $\omega_s$  maps into the unit circle in the  $z$ -plane. The right-half-plane portion of the primary strip (see Section 3.6) maps into the exterior of the unit circle, and the left-half-plane portion of the primary strip maps into the interior of the unit circle. Thus, since the stable region of the  $s$ -plane is the left half-plane, the stable region of the  $z$ -plane is the interior of the unit circle. Stability will be discussed in detail in Chapter 7.

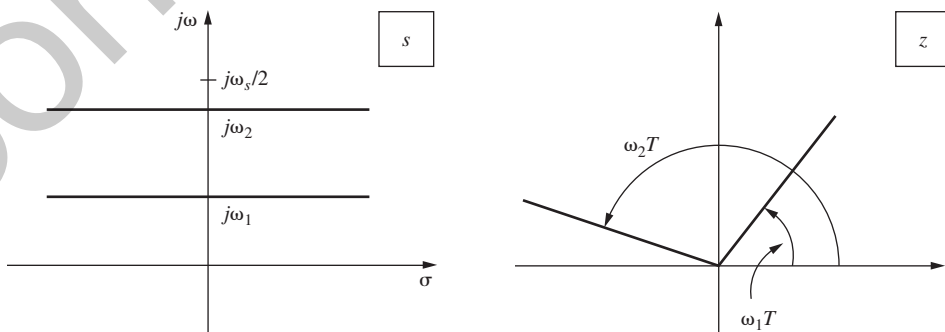
Constant damping loci in the  $s$ -plane (i.e., straight lines with  $\sigma$  constant) map into circles in the  $z$ -plane as shown in Fig. 6-7. This can be seen using the relationship

$$z = \epsilon^{\sigma_1 T} \epsilon^{j\omega T} = \epsilon^{\sigma_1 T} \angle \omega T$$

Constant frequency loci in the  $s$ -plane map into rays as shown in Fig. 6-8.



**FIGURE 6-7** Mapping constant damping loci into the  $z$ -plane.



**FIGURE 6-8** Mapping constant frequency loci into the  $z$ -plane.

For constant damping ratio loci,  $\sigma$  and  $\omega$  are related by

$$\frac{\omega}{\sigma} = \tan \beta$$

where  $\beta$  is constant. Then

$$z = \epsilon^{sT} = \epsilon^{\sigma T} \angle (\sigma T \tan \beta) \quad (6-6)$$

Since  $\sigma$  is negative in the second and third quadrants of the  $s$ -plane, (6-6) describes a logarithmic spiral whose amplitude decreases with  $\sigma$  increasing in magnitude. This is illustrated in Fig. 6-9.

As described above, the characteristics of a sampled time function at the sampling instants are the same as those of the time function before sampling. Thus, using the mappings illustrated in Figs. 6-6 through 6-9, we may assign time-response characteristics to characteristic-equation zero locations in the  $z$ -plane. The correspondence of several  $s$ -plane and  $z$ -plane pole locations is illustrated by several examples in Fig. 6-10. The time-response characteristics of the  $z$ -plane pole locations are illustrated in Fig. 6-11. Since

$$z = \epsilon^{sT} = \epsilon^{\sigma T} \epsilon^{j\omega T}$$

the time-response characteristics are a function not only of  $s$ , but also of  $T$ .

Consider the case in Fig. 6-10 that  $s$ -plane poles occur at  $s = \sigma \pm j\omega$ . These poles result in a system transient-response term of the form  $A e^{\sigma t} \cos(\omega t + \phi)$ . When sampling occurs, these  $s$ -plane poles result in  $z$ -plane poles at

$$z = \epsilon^{sT} |_{s=\sigma \pm j\omega} = \epsilon^{\sigma T} \epsilon^{\pm j\omega T} = \epsilon^{\sigma T} \angle (\pm \omega T) = r \angle (\pm \theta) \quad (6-7)$$

Thus roots of the characteristic equation that appear at  $z = r \angle (\pm \theta)$  result in a transient-response term of the form

$$A \epsilon^{\sigma k T} \cos(\omega k T + \phi) = A(r)^k \cos(\theta k + \phi)$$

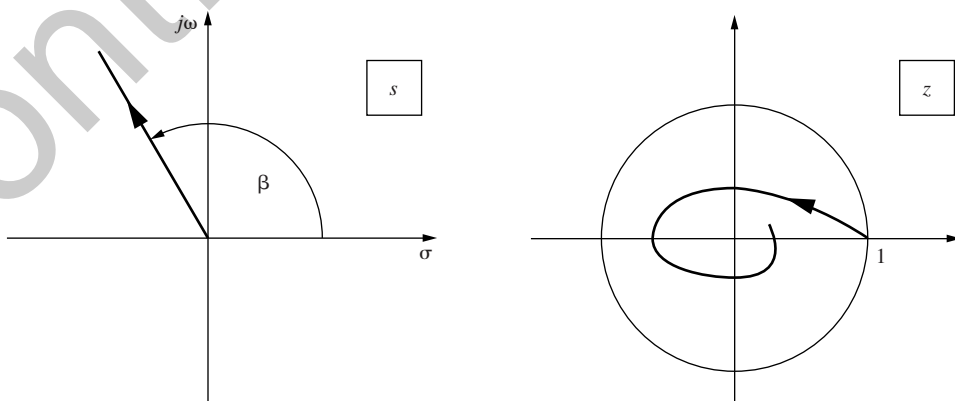
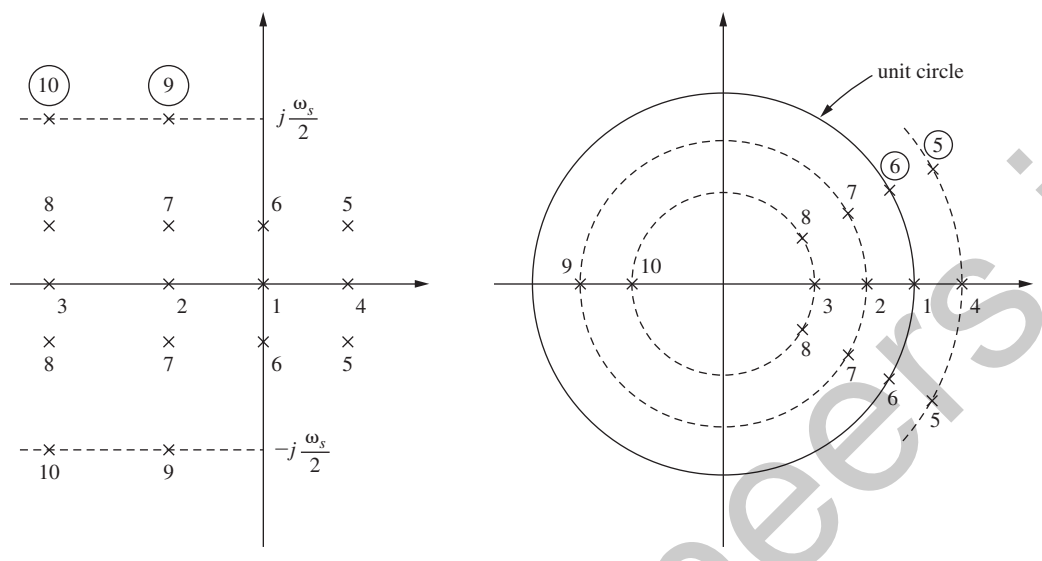
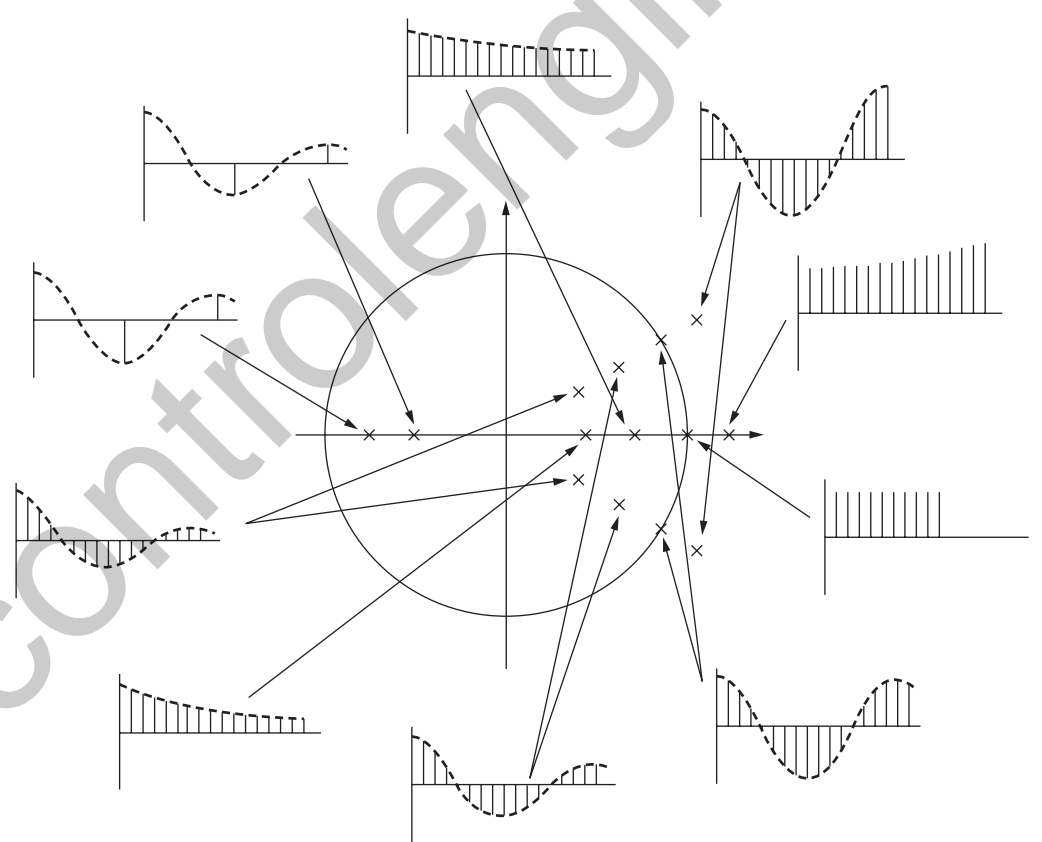


FIGURE 6-9 Mapping constant damping ratio loci into the  $z$ -plane.

6.4 Mapping the s-Plane into the z-Plane



**FIGURE 6-10** Corresponding pole locations between the s-plane and the z-plane.



**FIGURE 6-11** Transient response characteristics of the z-plane pole locations.

**EXAMPLE 6.5**

As an example, the time constant that appears in the sampled transient response of the first-order system of Example 6.1 will be calculated. The closed-loop transfer function for this system was found to be

$$\frac{G(z)}{1 + G(z)} = \frac{0.3625}{z - 0.4562}$$

Hence the closed-loop characteristic equation is

$$z - 0.4562 = 0$$

From Fig. 6-10, we see that a pole at  $z = 0.4562$  corresponds to an  $s$ -plane pole  $s_1$  on the negative real axis that satisfies

$$z_1 = 0.4562 = \epsilon^{s_1 T} = \epsilon^{0.1 s_1}$$

Hence  $s_1 = \ln(0.4562)/0.1$ , or  $s_1 = -7.848$ . Since the time constant is the reciprocal of the magnitude of a real pole in the  $s$ -plane, the closed-loop system has the time constant  $\tau = 0.127$  s. (From Example 6.3 the time constant of the system with the sampling removed is 0.167 s.) Also, since the transient response settles out in approximately four time constants, the transient response of this system will settle in approximately 0.5 s. This characteristic is seen in the step response of Fig. 6-1(b).

In the discussion above, we considered the relationship between  $s$ -plane poles and  $z$ -plane poles in a general way. We will now mathematically relate the  $s$ -plane pole locations and the  $z$ -plane pole locations. We express in standard form the  $s$ -plane second-order transfer function

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

which has the poles

$$s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$$

where  $\zeta$  is the damping ratio and  $\omega_n$  is the natural frequency. The equivalent  $z$ -plane poles occur at

$$z = \epsilon^{sT} \Big|_{s=s_{1,2}} = \epsilon^{-\zeta\omega_n T} \angle (\pm \omega_n T \sqrt{1 - \zeta^2}) = \angle r(\pm \theta)$$

Hence

$$\epsilon^{-\zeta\omega_n T} = r$$

or

$$\zeta\omega_n T = -\ln r$$

Also,

$$\omega_n T \sqrt{1 - \zeta^2} = \theta$$

Taking the ratio of the last two equations, we obtain

$$\frac{\zeta}{\sqrt{1 - \zeta^2}} = \frac{-\ln r}{\theta}$$

Solving this equation for  $\zeta$  yields

$$\zeta = \frac{-\ln r}{\sqrt{(\ln r)^2 + \theta^2}} \quad (6-8)$$

We then find  $\omega_n$  to be

$$\omega_n = \frac{1}{T} \sqrt{(\ln r)^2 + \theta^2} \quad (6-9)$$

The time constant,  $\tau$ , of the poles is then given by

$$\tau = \frac{1}{\zeta \omega_n} = \frac{-T}{\ln r} \quad (6-10)$$

This equation can also be expressed as

$$r = e^{-T/\tau}$$

Thus, given the complex pole location in the  $z$ -plane, we find the damping ratio, the natural frequency, and the time constant of the  $s$ -plane pole from (6-8), (6-9), and (6-10), respectively.

### EXAMPLE 6.6

For this example we will consider the system of Example 6.4. For this system the closed-loop transfer function was calculated to be

$$\frac{G(z)}{1 + G(z)} = \frac{0.368z + 0.264}{z^2 - z + 0.632}, \quad T = 1 \text{ s}$$

Thus the system characteristic equation is

$$z^2 - z + 0.632 = (z - 0.5 - j0.618)(z - 0.5 + j0.618) = 0$$

The poles are then complex and occur at

$$z = 0.5 \pm j0.618 = 0.795 \angle (\pm 51.0^\circ) = 0.795^\circ \angle (\pm 0.890 \text{ rad})$$

Since, in (6-8), (6-9), and (6-10),

$$z = e^{\sigma T} \angle (\pm \omega T) = r \angle (\pm \omega T) = 0.795 \angle (\pm 0.890)$$

then

$$\zeta = \frac{-\ln 0.795}{[(\ln 0.795)^2 + (0.890)^2]^{1/2}} = 0.250$$

$$\omega_n = \frac{1}{1} [(\ln 0.795)^2 + (0.890)^2]^{1/2} = 0.9191 \text{ rad/sec}$$

$$\tau = \frac{-1}{\ln 0.795} = 4.36 \text{ s}$$

The equivalent values for the closed-loop analog system (with sampler and data hold removed) were found in Example 6.4 to be  $\zeta = 0.50$  and  $\omega_n = 1$  rad/s. Also  $\tau = 1/\zeta\omega_n = 2$  s. Hence the effects of the sampling are seen to be destabilizing. However, if  $T = 0.1$  s, there is little effect from the sampling. From Problem 6.4-3, the parameters for this case are  $\zeta = 0.475$ ,  $\omega_n = 0.998$  rad/s, and  $\tau = 2.11$  s.

A word is in order concerning the probable location of characteristic-equation zeros. Transfer-function pole locations in the  $s$ -plane transform change into  $z$ -plane pole locations as

$$\begin{aligned} s + 1/\tau &\rightarrow z - \epsilon^{-T/\tau} \\ (s + 1/\tau)^2 + \omega^2 &\rightarrow z^2 - 2z\epsilon^{-T/\tau} \cos \omega T + \epsilon^{-2T/\tau} = (z - z_1)(z - \bar{z}_1) \end{aligned}$$

where

$$z_1 = \epsilon^{-T/\tau} \epsilon^{j\omega T} = \epsilon^{-T/\tau} \angle \omega T = r \angle \theta$$

The  $s$ -plane time constant for the real pole is  $\tau$ . In order for the sampling to have negligible effect,  $T$  must be much less than  $\tau$ . Thus for the real pole, the  $z$ -plane pole location will be in the vicinity of  $z = 1$ , since  $T/\tau$  is much less than 1. For the complex pole, an additional requirement is that several samples be taken per cycle of the sinusoid, or that  $\omega T < 1$ . Thus, once again,  $T \ll \tau$ , and since  $z\bar{z}_1 = \epsilon^{-2T/\tau}$  then  $|z_1| = \epsilon^{-T/\tau}$  and the  $z$ -plane pole locations again will be in the vicinity of  $z = 1$ . In general, for a discrete-time control system, the transfer-function pole locations (characteristic-equation zero locations) are placed in the vicinity of  $z = 1$ , if system constraints allow a sufficiently high sample rate to be chosen.

The ideas in the foregoing paragraph can be expressed mathematically. From (6-10),

$$\frac{\tau}{T} = -\frac{1}{\ln r} \quad (6-11)$$

We see that the ratio  $\tau/T$  is simply the number of samples per time constant. For example, if  $\tau = 1$  s and  $T = 0.25$  s, then we have four samples per time constant. Equation (6-11) is tabulated for certain values of  $r$  in Table 6-2.

**TABLE 6-2** Tabulated Values for (6-11) and (6-12)

Samples per time constant		Samples per period	
$r$	$\tau/T$	$\theta^\circ$	$T_d/T$
0.999	999.5	10°	36
0.99	99.5	20°	18
0.95	19.5	30°	12
0.9	9.5	45°	8
0.8	4.48	60°	6
0.7	2.80	90°	4
0.6	1.96	120°	3
0.4	1.09	180°	2
0.2	0.62		

In a like manner, we can manipulate the equation  $\omega T = \theta$  to obtain the number of samples per cycle of the sinusoid. Thus

$$\omega T = \frac{2\pi}{T_d} T = \theta$$

where  $T_d$  is the period of the sinusoid and  $\theta$  is in radians. Hence

$$\frac{T_d}{T} = \frac{2\pi}{\theta} = \frac{360^\circ}{\theta^\circ} \tag{6-12}$$

where  $\theta^\circ$  denotes  $\theta$  in degrees. Equation (6-12) is also tabulated for certain values of  $\theta$  in Table 6-2.

### 6.5 STEADY-STATE ACCURACY

An important characteristic of a control system is its ability to follow, or track, certain inputs with a minimum of error. The control system designer attempts to minimize the system error to certain anticipated inputs. In this section the effects of the system transfer characteristics on the steady-state system errors are considered.

Consider the system of Fig. 6-12. For this system,

$$\frac{C(z)}{R(z)} = \frac{G(z)}{1 + G(z)} \tag{6-13}$$

where  $G(z) = \mathcal{Z}[G(s)]$ . The plant transfer function can always be expressed as

$$G(z) = \frac{K \prod_{i=1}^m (z - z_i)}{(z - 1)^N \prod_{j=1}^p (z - z_j)}, \quad z_i \neq 1, \quad z_j \neq 1 \tag{6-14}$$

As we shall see, the value of  $N$  has special significance and is called the *system type*. For convenience in the following development, we define

$$K_{dc} = \left. \frac{K \prod_{i=1}^m (z - z_i)}{\prod_{j=1}^p (z - z_j)} \right|_{z=1} \tag{6-15}$$

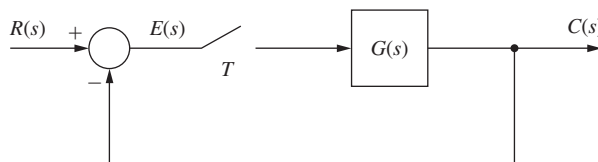


FIGURE 6-12 Discrete-time system.

Note that  $K_{dc}$  is the open-loop plant dc gain with all poles at  $z = 1$  removed.

For the system of Fig. 6-12, the system error,  $e(t)$ , is defined as the difference between the system input and the system output. Or

$$E(z) = \mathcal{Z}[e(t)] = R(z) - C(z) \quad (6-16)$$

Then, from (6-13) and (6-16),

$$E(z) = R(z) - \frac{G(z)}{1 + G(z)}R(z) = \frac{R(z)}{1 + G(z)}$$

The steady-state errors will now be derived for two common inputs—a position (step) input and a velocity (ramp) input. First, for the unit-step input,

$$R(z) = \frac{z}{z - 1}$$

Then, from the final-value theorem, the steady-state error is seen to be

$$e_{ss}(kT) = \lim_{z \rightarrow 1} (z - 1)E(z) = \lim_{z \rightarrow 1} \frac{(z - 1)R(z)}{1 + G(z)} \quad (6-17)$$

provided that  $e_{ss}(kT)$  has a final value. The steady-state error is then

$$e_{ss}(kT) = \lim_{z \rightarrow 1} \frac{z}{1 + G(z)} = \frac{1}{1 + \lim_{z \rightarrow 1} G(z)}$$

We now define the position error constant as

$$K_p = \lim_{z \rightarrow 1} G(z) \quad (6-18)$$

Then in (6-14), if  $N = 0$  [i.e., no poles in  $G(z)$  at  $z = 1$ ],  $K_p = K_{dc}$  and

$$e_{ss}(kT) = \frac{1}{1 + K_p} = \frac{1}{1 + K_{dc}} \quad (6-19)$$

For  $N \geq 1$  (system type greater than or equal to one),  $K_p = \infty$  and the steady-state error is zero.

Consider next the unit-ramp input. In this case  $r(t) = t$ , and from Appendix VI,

$$R(z) = \frac{Tz}{(z - 1)^2}$$

Then, from (6-17),

$$e_{ss}(kT) = \lim_{z \rightarrow 1} \frac{Tz}{(z - 1) + (z - 1)G(z)} = \frac{T}{\lim_{z \rightarrow 1} (z - 1)G(z)}$$

We now define the velocity error constant as

$$K_v = \lim_{z \rightarrow 1} \frac{1}{T}(z - 1)G(z) \quad (6-20)$$



Then if  $N = 0$ ,  $K_v = 0$  and  $e_{ss}(kT) = \infty$ . For  $N = 1$ ,  $K_v = K_{dc}/T$  and

$$e_{ss}(kT) = \frac{1}{K_v} = \frac{T}{K_{dc}} \quad (6-21)$$

For  $N \geq 2$  (system type greater than or equal to 2),  $K_v = \infty$  and  $e_{ss}(kT)$  is zero.

The development above illustrates that, in general, increased system gain and/or the addition of poles at  $z = 1$  to the open-loop forward-path transfer function tend to decrease steady-state errors. However, as will be demonstrated in Chapter 7, both large gains and poles of  $G(z)$  at  $z = 1$  have destabilizing effects on the system. Generally, trade-offs exist between small steady-state errors and adequate system stability (or acceptable system transient response).

### EXAMPLE 6.7

The steady-state errors will be calculated for the system of Fig. 6-12, in which the open-loop function is given as

$$G(s) = \frac{1 - \epsilon^{-Ts}}{s} \left[ \frac{K}{s(s+1)} \right]$$

Thus

$$\begin{aligned} G(z) &= K \mathfrak{Z} \left[ \frac{1 - \epsilon^{-Ts}}{s^2(s+1)} \right] = \frac{K(z-1)}{z} \mathfrak{Z} \left[ \frac{1}{s^2(s+1)} \right] \\ &= \frac{K(z-1)z[(\epsilon^{-T} + T - 1)z + (1 - \epsilon^{-T} - T\epsilon^{-T})]}{z(z-1)^2(z - \epsilon^{-T})} \\ &= \frac{K[(\epsilon^{-T} + T - 1)z + (1 - \epsilon^{-T} - T\epsilon^{-T})]}{(z-1)(z - \epsilon^{-T})} \end{aligned}$$

Then, from (6-14) and (6-20), the system is type 1 and

$$K_v = \frac{K_{dc}}{T} = \frac{K[(\epsilon^{-T} + T - 1) + (1 - \epsilon^{-T} - T\epsilon^{-T})]}{T(1 - \epsilon^{-T})} = K$$

Since  $G(z)$  has one pole at  $z = 1$ , the steady-state error to a step input is zero, and to a ramp input is, from (6-21),

$$e_{ss}(kT) = \frac{1}{K_v} = \frac{1}{K}$$

provided that the system is stable. The question of stability is considered in Chapter 7.

**EXAMPLE 6.8**

As a second example, consider again the system of Fig. 6-12, where, for this example,

$$G(z) = \mathfrak{Z} \left[ \frac{1 - \varepsilon^{-Ts}}{s(s+1)} \right] = \frac{1 - \varepsilon^{-T}}{z - \varepsilon^{-T}}$$

Suppose that the design specification for this system requires that the steady-state error to a unit ramp input be less than 0.01. Thus, from (6-20), it is necessary that the open-loop system be type 1 or greater and thus the open-loop function must have at least one pole at  $z = 1$ . Since the plant does not contain a pole at  $z = 1$ , a digital compensator of the form

$$D(z) = \frac{K_I z}{z - 1} + K_P$$

can be added, to produce the resultant system shown in Fig. 6-13.

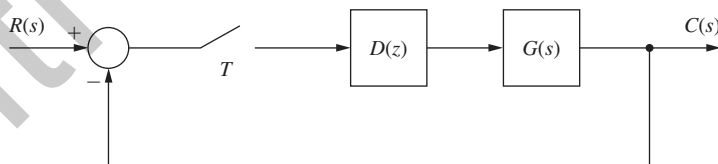
The compensator, called a PI or proportional-plus-integral compensator, is of a form commonly used to reduce steady-state errors. This compensator is discussed in Chapter 8. For this system (6-20) becomes

$$K_v = \lim_{z \rightarrow 1} \frac{1}{T} (z - 1) D(z) G(z)$$

Employing the expressions above for  $D(z)$  and  $G(z)$ , we see that

$$K_v = \lim_{z \rightarrow 1} (z - 1) \frac{(K_I + K_P)z - K_P}{T(z - 1)} \left[ \frac{1 - \varepsilon^{-T}}{z - \varepsilon^{-T}} \right] = \frac{K_I}{T}$$

Thus  $K_I$  equals  $100T$  for the required steady-state error, provided that the system is stable. The latter point is indeed an important consideration since the error analysis is meaningless unless stability of the system is guaranteed. Chapter 7 illustrates a number of techniques for analyzing system stability. This system is considered further in Example 7.6.



**FIGURE 6-13** System for Example 6.8.

**6.6 SIMULATION**

Thus far we have determined the time response of a system via a transform approach. In Example 6.1 we determined the system response at the sampling instants using the  $z$ -transform. A Laplace transform technique for calculating the system response for all time was also illustrated.

A different approach for determining a system's response is through simulation. The simulation of a continuous-time (analog) system may be via the integration of the system's differential equations using electronic circuits. An analog computer can be designed to perform this function. The interconnection of a digital computer with the analog computer is called a hybrid system. The analog parts of the control system can be simulated on the analog computer with the digital controller simulated on the digital computer. If a numerical algorithm is used to integrate an analog system's differential equations on a digital computer, we then can eliminate the analog computer and perform a digital simulation of the entire system. This is the type of simulation that we consider here.

The problem of numerical integration of a time function can be illustrated using Fig. 6-14. We wish to integrate  $y(t)$  numerically; that is, we wish to find  $x(t)$ , where

$$x(t) = \int_0^t y(\tau) d\tau + x(0) \quad (6-22)$$

Suppose that we know  $x[(k - 1)H]$  and we want to calculate  $x(kH)$ , where  $H$  is called the *numerical integration increment* and is the step size in the algorithm. Perhaps the simplest numerical integration algorithm is obtained by assuming  $y(t)$  constant at the value  $y[(k - 1)H]$  for  $(k - 1)H \leq t < kH$ . Then

$$x(kH) = x[(k - 1)H] + Hy[(k - 1)H] \quad (6-23)$$

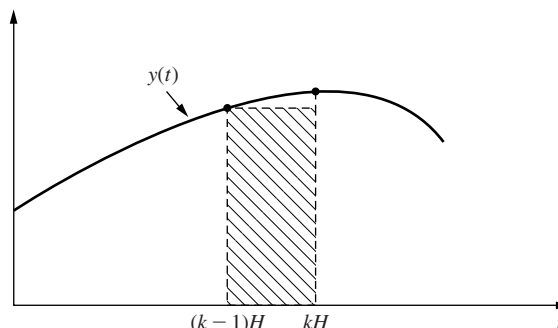
Of course,  $x(kH)$  in (6-23) is only an approximation to  $x(t)$  in (6-22) evaluated at  $t = kH$ . Note in Fig. 6-14 that we are approximating the area under the  $y(t)$  curve for  $(k - 1)H \leq t < kH$  with the area of the rectangle shown. For this reason, this numerical integration algorithm is called the *rectangular rule*, and is also known as the *Euler method*.

We are interested in the integration of differential equations [i.e., in (6-22),  $y(t) = \dot{x}(t)$ ]. We will illustrate this case with a simple example. Suppose that we wish to solve the differential equation

$$\dot{x}(t) + x(t) = 0, \quad x(0) = 1 \quad (6-24)$$

The solution is obviously

$$x(t) = e^{-t}$$



**FIGURE 6-14** Rectangular rule for numerical integration.

However, by numerical integration in (6-23),

$$x(kH) = x[(k - 1)H] + H[-x[(k - 1)H]] \quad (6-25)$$

since, from (6-24), for this differential equation

$$y(t) = \dot{x}(t) = -x(t)$$

Suppose that we chose  $H = 0.1$  s. Then, solving (6-25) iteratively starting with  $k = 1$  [we know  $x(0)$ ],

$$\begin{aligned} x(0.1) &= x(0) - Hx(0) = 1.0 - (0.1)(1.0) = 0.9 \\ x(0.2) &= x(0.1) - Hx(0.1) = 0.9 - 0.09 = 0.81 \\ &\vdots \\ x(1.0) &= x(0.9) - Hx(0.9) = 0.3487 \end{aligned}$$

Since, for this example, we know the solution, we calculate  $x(t)$  at  $t = 1.0$  s as

$$x(1.0) = e^{-1.0} = 0.3678$$

and we can see the error due to the numerical integration.

In the example above, if we choose  $H$  larger, the error is larger. If we decrease  $H$ , the error decreases to a minimum value. Then a further decrease in  $H$  will result in an increase in the error, due to round-off in the computer. With  $H$  equal to 0.1 s, 10 iterations are required to calculate  $x(1)$ . With  $H = 0.001$  s, 1000 iterations are required to calculate  $x(1)$ . The round-off errors in the computer are larger for the latter case, since more calculations are made. If  $H$  is made sufficiently smaller, this round-off error becomes appreciable.

Next we will consider the simulation of an analog system. Assume that the system's state equations are given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c \mathbf{x}(t) + \mathbf{B}_c \mathbf{u}(t) \quad (6-26)$$

Then (6-22) becomes

$$\mathbf{x}(t) = \int_0^t \dot{\mathbf{x}}(\tau) d\tau + \mathbf{x}(0) \quad (6-27)$$

and the rectangular rule for  $\mathbf{x}(t)$  a vector becomes

$$\mathbf{x}(kH) = \mathbf{x}[(k - 1)H] + H\dot{\mathbf{x}}[(k - 1)H] \quad (6-28)$$

where, from (6-26),

$$\dot{\mathbf{x}}[(k - 1)H] = \mathbf{A}_c \mathbf{x}[(k - 1)H] + \mathbf{B}_c \mathbf{u}[(k - 1)H] \quad (6-29)$$

The numerical integration algorithm becomes:

1. Let  $k = 1$ .
2. Evaluate  $\dot{\mathbf{x}}[(k - 1)H]$  in (6-29).

3. Evaluate  $\mathbf{x}(kH)$  in (6-28).
4. Let  $k = k + 1$ .
5. Go to step 2.

Refer again to Fig. 6-14. This figure illustrates the rectangular rule for numerical integration. Other numerical integration algorithms differ from this rule in the manner that  $\mathbf{x}[kH]$  is calculated, based on the knowledge of  $\dot{\mathbf{x}}[(k - 1)H]$ . As an example of a different algorithm, consider the trapezoidal rule illustrated in Fig. 6-15. The integral for  $[(k - 1)H] \leq t < kH$  is approximated by the area of the trapezoid. For this rule, with  $x(t)$  equal to the integral of  $y(t)$ ,

$$x(kH) = x[(k - 1)H] + H \left[ \frac{y[(k - 1)H] + y(kH)}{2} \right] \quad (6-30)$$

For  $y(t)$  equal to  $\dot{x}(t)$ , then

$$x(kH) = x[(k - 1)H] + \frac{H}{2} [\dot{x}[(k - 1)H] + \dot{x}(kH)] \quad (6-31)$$

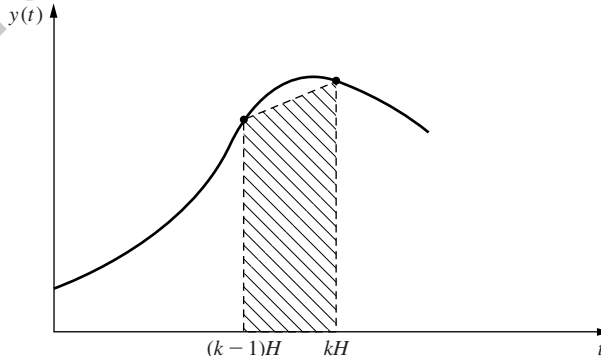
Since  $x[(k - 1)H]$  is known,  $\dot{x}[(k - 1)H]$  can be calculated from the differential equation that is to be solved. However, we cannot calculate  $\dot{x}(kH)$  without a knowledge of  $x(kH)$ . A method for solving this problem will now be presented.

We use a different rule to “predict” the value of  $x(kH)$ . Using this predicted value, we calculate the predicted value of  $\dot{x}(kH)$ , and substitute this value in the trapezoidal rule to “correct” the value of  $x(kH)$ . A method of this type is called a predictor-corrector algorithm. A commonly used rule for prediction is the rectangular rule.

The state model of an analog system given in (6-26) will be used to illustrate the trapezoidal rule. It is assumed that  $\mathbf{x}[kH]$  is known for  $k = 0$ , and thus  $\mathbf{x}[kH]$  is to be calculated for  $k = 1, 2, 3, \dots$ . The predictor algorithm, which is the rectangular rule, is given by

$$\dot{\mathbf{x}}[(k - 1)H] = \mathbf{A}_c \mathbf{x}[(k - 1)H] + \mathbf{B}_c \mathbf{u}[(k - 1)H] \quad (6-32)$$

$$\mathbf{x}(kH) = \mathbf{x}[(k - 1)H] + H \dot{\mathbf{x}}[(k - 1)H] \quad (6-33)$$



**FIGURE 6-15** Trapezoidal rule for numerical integration.

The corrector algorithm is given by

$$\dot{\mathbf{x}}(kH) = \mathbf{A}_c \mathbf{x}(kH) + \mathbf{B}_c \mathbf{u}(kH) \quad (6-34)$$

$$\mathbf{x}(kH) = \mathbf{x}[(k-1)H] + \frac{H}{2} [\dot{\mathbf{x}}[(k-1)H] + \dot{\mathbf{x}}(kH)] \quad (6-35)$$

The predicted value of  $\mathbf{x}(kH)$  is given by (6-33), and the corrected, and final, value of  $\mathbf{x}(kH)$  is given by (6-35). Thus, the numerical integration algorithm is given by:

1. Let  $k = 1$ .
2. Evaluate  $\dot{\mathbf{x}}[(k-1)H]$  in (6-32).
3. Evaluate  $\mathbf{x}(kH)$  in (6-33).
4. Evaluate  $\dot{\mathbf{x}}(kH)$  in (6-34), using the result of step 3.
5. Evaluate the final value of  $\mathbf{x}(kH)$  in (6-35).
6. Let  $k = k + 1$ .
7. Go to step 2.

This algorithm obviously requires more calculations per iteration than does the rectangular rule. However, for a specified accuracy of the solution, a much larger increment  $H$  may be used such that the total computer execution time for the trapezoidal rule is less than that for the rectangular rule. The trapezoidal rule will now be illustrated by an example.

### EXAMPLE 6.9

Consider once again the differential equation

$$\dot{x}(t) + x(t) = 0, \quad x(0) = 1$$

which has the solution

$$x(t) = e^{-t}$$

Since

$$\dot{x}(t) = -x(t)$$

implementing the trapezoidal rule for  $H = 0.1$  s yields

$K = 1$ :

$$\dot{x}(0) = -x(0) = -1$$

$$x(0.1) = x(0) + H\dot{x}(0) = 1 + 0.1(-1) = 0.9$$

$$\dot{x}(0.1) = -x(0.1) = -0.9$$

$$\begin{aligned} x(0.1) &= x(0) + H/2[\dot{x}(0) + \dot{x}(0.1)] \\ &= 1 + 0.05(-1 - 0.9) = 0.905 \end{aligned}$$

$K = 2$ :

$$\dot{x}(0.1) = -x(0.1) = -0.905$$

$$x(0.2) = x(0.1) + H\dot{x}(0.1) = 0.905 - 0.0905 = 0.8145$$

$$\begin{aligned}\dot{x}(0.2) &= -x(0.2) = -0.8145 \\ x(0.2) &= x(0.1) + H/2[\dot{x}(0.1) + \dot{x}(0.2)] \\ &= 0.905 + 0.05(-0.905 - 0.8145) = 0.8190\end{aligned}$$

for the first two iterations. If these calculations are continued, we find  $x(1.0)$  to be 0.3685. Recall that the exact value of  $x(1)$  is 0.3678, and the value of  $x(1)$  calculated above using the rectangular rule is 0.3487. Hence much greater accuracy results from use of the trapezoidal rule as compared with the rectangular rule, but at the expense of more calculations. If  $H$  is increased to 0.333 s,  $x(1)$  is calculated to be 0.3767 with the trapezoidal rule. Here the accuracy is still significantly greater than that of the rectangular rule with  $H = 0.1$  s, although the amount of calculations required is approximately the same.

In the discussion above, only two of many numerical integration rules were discussed. One of the more commonly used rules for digital simulation is the fourth-order Runge–Kutta rule [2–4]. The interested reader is referred to the many texts available in this area.

Two additional points concerning simulation should be made. First, in digital simulations we approximate differential equations by difference equations and solve the difference equations. Thus we are replacing a continuous-time system with a discrete-time system that has approximately the same response.

The second point is that, in the simulation of nonlinear systems, the nonlinearities appear only in the calculation of  $\dot{\mathbf{x}}[(k - 1)H]$ , given  $\mathbf{x}[(k - 1)H]$ . If the nonlinearities are of a form that can be easily expressed in a mathematical form, the simulation is not appreciably more difficult to write.

In the discussion above, we considered the simulation of analog systems only. The addition of a sampler and zero-order hold to an analog system requires that logic be added to the system simulation. The logic will hold the zero-order-hold output constant over any sample period, and equal to the value of sampler input at the beginning of that sample period. The addition of a digital controller requires that the controller difference equation be solved only at the beginning of each sample period, and that the controller output then remain constant over the sample period.

## 6.7 CONTROL SOFTWARE

Many commercial analysis and design control software packages are available; these packages have simulation capabilities of varying degrees. We employ MATLAB in this text. In general, the analog components in a digital control system are entered into the control software programs as transfer functions or state equations. Most packages will accept either model and then calculate the other model from the one given. If the digital controller is known, it is entered as a transfer function or, in some cases, as a state model. The packages can then simulate the system. In general, the packages have a number of numerical integration algorithms available; the user must choose one of these algorithms or accept the programs default selection.

## 6.8 SUMMARY

The time response of discrete-time closed-loop systems has been discussed. Both steady-state and transient responses have been considered. The correlation between the  $s$ -plane and the  $z$ -plane which has been presented provides a mechanism for the transfer of many of the continuous-system tools needed for both the analysis and design of closed-loop discrete-time systems. Finally, a brief introduction into the digital simulation of systems has been presented.

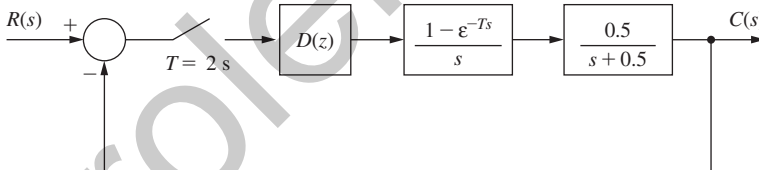
## References and Further Readings

- [1] C. L. Phillips and J. Parr, *Feedback Control Systems*, 5th ed. Upper Saddle River, NJ: Prentice-Hall, 2011.
- [2] M. L. Dertouzos, M. Athans, R. N. Spann, and S. J. Mason, *Systems, Networks, and Computation: Basic Concepts*. Huntington, NY: R.E. Krieger Publishing Co., Inc., 1979.
- [3] J. L. Melsa and S. K. Jones, *Computer Programs for Computational Assistance*. New York: McGraw-Hill Book Company, 1973.
- [4] C. F. Gerald, *Applied Numerical Analysis*, 7th ed. Reading, MA: Addison-Wesley Publishing Company, Inc., 2007.
- [5] J. A. Cadzow and H. R. Martens, *Discrete-Time and Computer Control Systems*. Upper Saddle River, NJ: Prentice-Hall, 1970.
- [6] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3d ed. Half Moon Bay, CA: Ellis-Kagle Press, 2006.
- [7] E. I. Jury, *Theory and Application of the z-Transform Method*. Huntington, NY: R.E. Krieger Publishing Co., Inc., 1973.
- [8] B. C. Kuo, *Digital Control Systems*, 2d ed. New York: Oxford University Press, 1995.

## Problems

**6.2-1.** Consider the closed-loop system of Fig. P6.2-1.

- (a) Calculate and plot the unit-step response at the sampling instants, for the case that  $D(z) = 1$ .
- (b) Calculate the system unit-step response of the analog system, that is, with the sampler, digital controller, and data hold removed. Plot the response on the same graph with the results of part (a).
- (c) For the system of Fig. P6.2-1, let  $D(z) = 1$  and  $T = 0.4$  s. Calculate the unit-step response and plot these results on the same graph used for parts (a) and (b).
- (d) Use the system dc gains to calculate the steady-state responses for each of the systems of parts (a), (b), and (c). Why are these gains equal?



**FIGURE P6.2-1** System for Problem 6.2-1.

**6.2-2.** Consider the system of Fig. P6.2-1, with  $D(z) = 1$ . Use the results of Problem 6.2-1 if available.

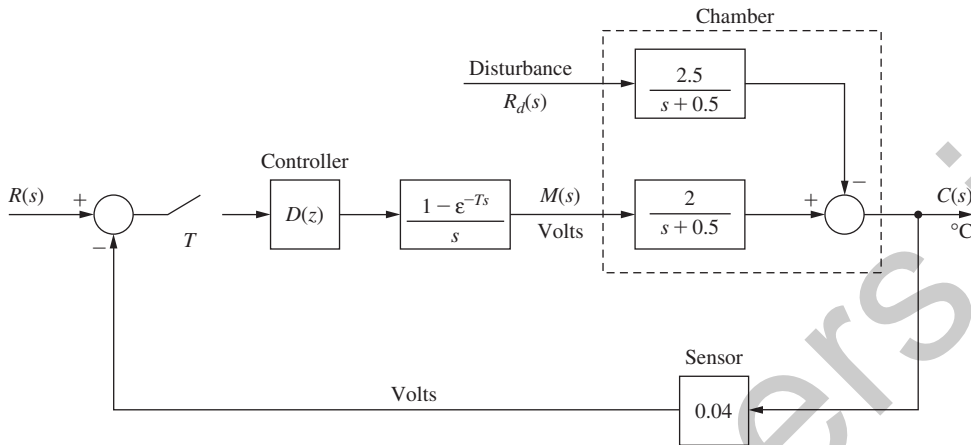
- (a) Find the system time constant  $\tau$  for  $T = 4$  s.
- (b) With the input a step function, find the time required for the system output  $c(kT)$  to reach 98 percent of its final value, for  $T = 4$  s. Recall that four time constants ( $4\tau$ ) are required.
- (c) Repeat parts (a) and (b) for  $T = 0.6$  s.
- (d) Repeat parts (a) and (b) for the analog system, that is, for the system with the sampler and data hold removed.

**6.2-3.** In Example 6.1 the response of a sampled-data system between sample instants was expressed as a sum of delayed step responses.

- (a) Use this procedure to find the system output  $y(t)$  of Fig. P6.2-1 at  $t = 1$  s.
- (b) Repeat part (a) at  $t = 3$  s.
- (c) The equation for  $c(t)$  in part (a) and that for  $c(t)$  in part (b) should give the same value at  $t = 2$  s. Why?   
*Hint:* Consider the step response of the analog plant.
- (d) Show that the statement in part (c) is true.

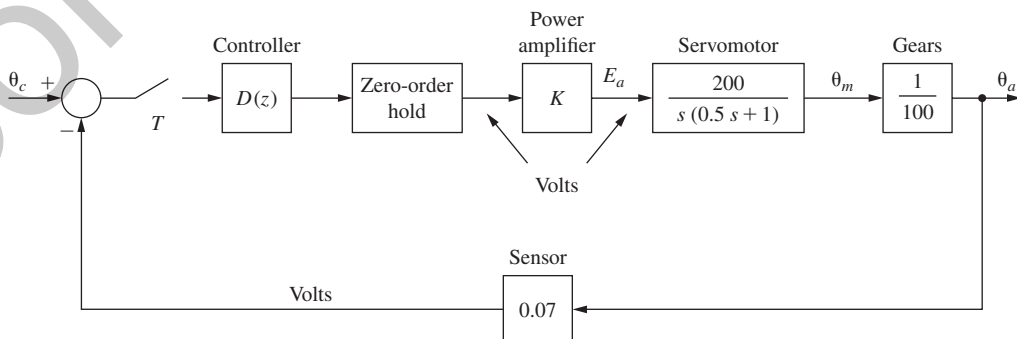
**6.2-4.** Shown in Fig. P6.2-4 is the block diagram of a temperature control system for a large test chamber. This system is described in Problem 1.6-1. Ignore the disturbance input for this problem.





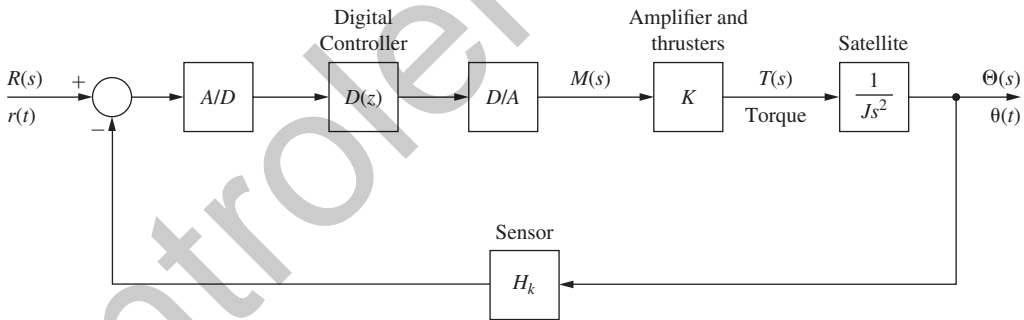
**FIGURE P6.2-4** Chamber temperature control system.

- With  $D(z) = 1$  and  $T = 0.6$  s, evaluate and plot the system response if the input is to command a  $10^\circ\text{C}$  step in the output. Note that the system input must be a step function with an amplitude of  $0.4$  V. Why?
  - Use the results of part (a) to plot the output of the zero-order hold.
  - Solve for the steady-state output for part (a).
  - Suppose that the gain of  $2$  in the plant is replaced with a variable gain  $K$ . What value does the output approach in the steady state as  $K$  becomes very large? Assume that the system remains stable as  $K$  is increased (an unrealistic assumption).
- 6.2-5.** Consider the temperature control system of Problem 6.2-4 and Fig. P6.2-4.
- Let  $T = 5$  s, and solve for the response to the input  $R(s) = 0.2 s^{-1}$ . Plot this response on the same graph with the response found in Problem 6.2-4. Note the effects of increasing  $T$  from  $0.6$  s to  $6$  s, when the plant has a time constant of  $2$  s.
  - Find  $c(t)$  for  $0 \leq t \leq 6$  s. This response can be calculated without the use of the  $z$ -transform.
- 6.2-6.** Consider the system of Fig. P6.2-4, with  $D(z) = 1$ . Use the results of Problems 6.2-4 and 6.2-5 if available.
- Find the system time constant  $\tau$  for  $T = 0.8$  s.
  - With the input a step function, find the time required for the system output  $c(kT)$  to reach 98 percent of its final value. Note that this time is approximately four time constants ( $4\tau$ ).
  - Repeat parts (a) and (b) for  $T = 8$  s.
  - Repeat parts (a) and (b) for the analog system, that is, for the system with the sampler and data hold removed.
- 6.2-7.** The block diagram of a control system of a joint in a robot arm is shown in Fig. P6.2-7. This system is discussed in Section 1.6. Let  $T = 0.1$  s and  $D(z) = 1$ .



**FIGURE P6.2-7** Robot arm joint control system.

- (a) Evaluate  $C(z)$  if the input is to command a  $20^\circ\text{C}$  step in the output and  $K = 10$ . Note that the system input must be a step function with an amplitude 1.4 V. Why?
- (b) Assuming the system to be stable, find the steady-state system output.
- (c) Find the (approximate) time required for the system response to reach steady state.
- (d) Simulate the system to verify the results in parts (b) and (c).
- 6.4-1.** The block diagram of a control system of a joint in a robot arm is shown in Fig. P6.2-7. Let  $T = 0.1$  s,  $K = 20$ , and  $D(z) = 1$ . The results of Problem 6.2-7 are useful in this problem if these results are available.
- (a) Find the damping ratio  $\zeta$ , the natural frequency  $\omega_n$ , and the time constant  $\tau$  of the open-loop system. If the system characteristic equation has two real zeros, find the two time constants. These values can be solved by inspection. Why?
- (b) Repeat part (a) for the closed-loop system.
- (c) Repeat parts (a) and (b) for the system with the sampler, digital controller, and data hold removed, that is, for the analog system.
- (d) Use the results in parts (b) and (c) to find the percent overshoot in the step responses for the sampled-data closed-loop system and for the analog closed-loop system.
- 6.4-2.** The block diagram of an attitude control system of a satellite is shown in Fig. P6.4-2. Let  $T = 1$  s,  $K = 200$ ,  $J = 0.2$ ,  $H_k = 0.04$ , and  $D(z) = 1$ .
- (a) Find the damping ratio  $\zeta$ , the natural frequency  $\omega_n$ , and the time constant  $\tau$  of the open-loop system. If the system characteristic equation has two real zeros, find the two time constants. These values can be solved by inspection. Why?
- (b) Repeat part (a) for the closed-loop system.
- (c) Repeat parts (a) and (b) for the system with the sampler, digital controller, and data hold removed, that is, for the analog system.
- (d) The closed-loop sampled-data system is seen to be unstable and the closed-loop analog system is seen to be marginally stable. If the satellite is operated with each of these control systems, describe the resulting movement for both the sampled-data system and the analog system.



**FIGURE P6.4-2** Block diagram for a satellite control system.

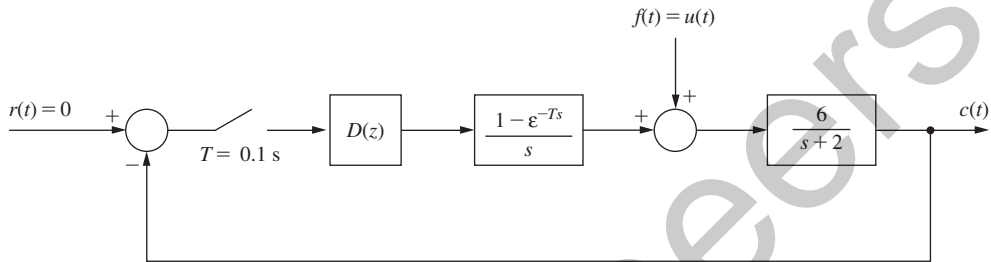
- 6.4-3.** It is shown in Example 6.6 that the system of Fig. 6-2 has the parameters  $\zeta = 0.250$ ,  $\omega_n = 0.9191$ , and  $\tau = 4.36$  s.
- (a) Find  $\zeta$ ,  $\omega_n$ , and  $\tau$  for the sample period  $T = 0.5$ .
- (b) Repeat part (a) for  $T = 0.1$ .
- (c) Repeat part (a) for the analog system, that is, the system with the sampler and data hold removed. Note that this case can be considered to be the limit as the sample period approaches zero.
- (d) Give a table listing the three parameters as a function of sampling frequency  $f_s = 1/T$ . State the result of decreasing the sampling frequency on the parameters.
- 6.4-4.** Consider the system of Fig. P6.4-4. This system is called a regulator control system, in which it is desired to maintain the output,  $c(t)$ , at a value of zero in the presence of a disturbance,  $f(t)$ . In this problem the disturbance is a unit step.

- (a) With  $D(z) = 1$  (i.e., no compensation) find the steady-state value of  $c(t)$ .  
 (b) For  $f(t)$  to have no effect on the steady-state value of  $c(kT)$ ,  $D(z)$  should have a pole at  $z = 1$ . Let

$$D(z) = 1 + \frac{0.1z}{z - 1}$$

Determine the steady-state value of  $c(kT)$ .

- (c) Repeat parts (a) and (b) with  $T = 1$  s.  
 (d) Why does the value of sample period  $T$  have no effect on the steady-state response for a constant input?



**FIGURE P6.4-4** System for Problem 6.4-4.

**6.4-5.** Consider the sampled-data systems with the following characteristic equations:

- |                      |                             |
|----------------------|-----------------------------|
| (i) $z - 0.199 = 0$  | (v) $z^2 - 1.9z + 0.9 = 0$  |
| (ii) $z - 0.99 = 0$  | (vi) $z^2 - 1 = 0$          |
| (iii) $z - 0.19 = 0$ | (vii) $z^2 - 2z + 0.99 = 0$ |
| (iv) $z + 0.9 = 0$   |                             |

- (a) What can you determine about the system natural-response characteristics for each system, for  $T = 0.2$  s?  
 (b) Repeat part (a) for  $T = 2$  s.  
 (c) For a given characteristic equation as a function of  $z$ , which parameters of the transient response vary with the sample period  $T$ , and which are independent of  $T$ ?
- 6.4-6.** Consider the system of Fig. P6.2-1. Suppose that an ideal time delay of 0.2 s is added to the plant, such that the plant transfer function is now given by

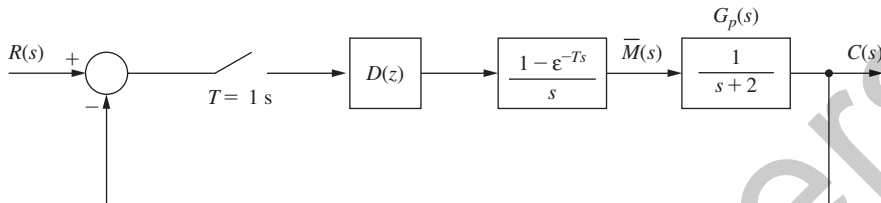
$$G_p(s) = \frac{0.5\epsilon^{-0.2T}}{s + 0.5}$$

- (a) Find the time constant  $\tau$  of the system if the time delay is omitted.  
 (b) Find the time constant  $\tau$  of the system if the time delay is included.  
 (c) Repeat part (b) for a time delay of 1 s.  
 (d) What is the effect on the speed of response of the closed-loop system of adding time delay to the plant?
- 6.5-1.** (a) Give the system type for the following systems, with  $D(z) = 1$ . It is not necessary to find the pulse transfer functions to find the system type. Why?
- |                    |
|--------------------|
| (i) Fig. P6.2-1.   |
| (ii) Fig. P6.2-4.  |
| (iii) Fig. P6.2-7. |
| (iv) Fig. P6.4-2.  |
| (v) Fig. P6.4-4.   |
- (b) It is desired that the systems of part (a) have zero steady-state error for a constant input. Give the required characteristics for each digital controller.  
 (c) It is desired that the systems of part (a) have zero steady-state error for a ramp input. Give the required characteristics for each digital controller.

**6.5-2.** Consider the system of Fig. P6.5-2. The digital filter is described by

$$m(kT) = e(kT) - 1.9e[(k-1)T] + m[(k-1)T]$$

- Find the system type.
- Find the steady-state response for a unit-step input, without finding  $C(z)$ .
- Find the approximate time for the system to reach steady state.
- Find the unit-step response for the system and verify parts (b) and (c).



**FIGURE P6.5-2** System for Problem 6.5-2.

**6.5-3.** Consider the system of Fig. P6.5-2, with the digital filter as described in Problem 6.5-2.

- Find the system type.
- Find the steady-state response for a unit ramp input, without finding  $C(z)$ .
- Find the approximate time for the system to reach steady state.
- Calculate the steady-state unit-ramp response for the system to verify parts (b) and (c).

**6.5-4.** Consider the system of Problem 6.5-2. Let the plant transfer function be given by

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

Hence the dc gain of the plant is zero. The digital filter is as given in Problem 6.5-2.

- Find the system type.
- Find the steady-state response for a unit-step input, without finding  $C(z)$ .
- Find the approximate time for the system to reach steady state.
- Find  $c(kT)$  and verify the results of parts (b) and (c).

**6.5-5.** For the system of Fig. 6-13,  $G_p(s) = 20(s+5)/(s(s+4)(s+6))$ , with  $T = 0.05$  and  $D(z)$  implementing the following difference equation

$$m(k) = 10e(k) - 16e(k-1) - 6.3e(k-2) + m(k-1)$$

- Find the system type.
- Plot the unit-step response.
- Find the rise time, overshoot, and settling time.

**6.5-6.** Repeat Problem 6.5-5 with  $T = 0.02$ .

- Find the system type.
- Plot the unit-step response.
- Find the rise time, overshoot, and settling time.
- Compare your results with those of Problem 6.5-5 and explain any differences.

**6.6-1.** Consider the system of Fig. P6.5-2, with  $D(z) = 1$ .

- With the sampler and zero-order hold removed, write the system differential equation.
- Using the rectangular rule for numerical integration with the numerical integration increment equal to 0.25 s, evaluate the unit-step response for  $0 \leq t \leq 1.5$  s.
- Repeat part (b) with the sampler and zero-order hold in the system, and with a sample period  $T = 0.5$  s.
- Solve for the exact unit-step responses for parts (b) and (c), and compare results.

**6.6-2.** Consider the numerical integration of the differential equation (6-24) in Section 6.6. Applying the rectangular rule results in the difference equation (6-25), which can be expressed as

$$x[(k+1)H] = x(kH) + H[-x(kH)]$$

- (a) Use the  $z$ -transform to solve this equation as a function of  $x(0)$  and  $H$ .
- (b) Use the results of part (a) to show that as given in Section 6.6,  $x(1.0) = 0.3487$  with  $x(0) = 1$  and  $H = 0.1$ .
- (c) Solve for  $x(1.0)$  with  $x(0) = 1$  and  $H = 0.01$ .
- (d) The exact value of  $x(1.0)$  is given in Section 6.6. Compare the errors in the results of parts (b) and (c).

**6.6-3.** Consider the numerical integration of the differential equation

$$\frac{dx(t)}{dt} + x(t) = r(t)$$

using the rectangular rule.

- (a) Develop the difference equation, as in (6-25), for the numerical integration of this differential equation.
- (b) Let  $r(t) = 1$ ,  $x(0) = 0$ , and  $H = 0.1$ . Use the  $z$ -transform to solve the difference equation of part (a) for  $x(1.0)$ .
- (c) Repeat part (b) for  $H = 0.01$ .
- (d) Solve the given differential equation, using the Laplace transform, for the exact value of  $x(1.0)$ .
- (e) Compare the errors in the results in parts (b) and (c).

**6.6-4.** Consider the numerical integration of the differential equation

$$\frac{dx(t)}{dt} + x(t) = 0$$

using the predictor-corrector of Section 6.6. The predictor method is the rectangular rule, and the corrector method is the trapezoidal rule.

- (a) Develop the difference equation, using (6-32) through (6-35), for the numerical integration of the given differential equation. The result should be one difference equation for  $x[(k + 1)H]$  as a function of  $x(kH)$ .
- (b) Let  $x(0) = 1$ , and  $H = 0.1$ . Use the  $z$ -transform to solve the difference equation of part (a) for  $x(1.0)$ . This value is given as 0.3685 in Example 6.9.
- (c) Repeat part (b) for  $H = 0.33333$ . This value is given as 0.3767 in Example 6.9.
- (d) Solve the given differential equation, using the Laplace transform, for the exact value of  $x(1.0)$ .
- (e) Give the errors in the results in parts (b) and (c).

**6.6-5.** Consider the numerical integration of the differential equation

$$\frac{dx(t)}{dt} + x(t) = r(t)$$

using the predictor-corrector of Section 6.6. The predictor method is the rectangular rule, and the corrector method is the trapezoidal rule.

- (a) Develop the difference equation, using (6-32) through (6-35), for the numerical integration of the given differential equation. The result should be one difference equation for  $x(kH)$  as a function of  $x[(k - 1)H]$  and the input function  $r[(k - 1)H]$ .
- (b) Let  $r(t) = 1$ ,  $x(0) = 0$ , and  $H = 0.1$ . Use the  $z$ -transform to solve the difference equation of part (a) for  $x(1.0)$ .
- (c) Repeat part (b) for  $H = 0.33333$ .
- (d) Solve the given differential equation, using the Laplace transform, for the exact value of  $x(1.0)$ .
- (e) Give the errors in the results in parts (b) and (c).

**6.6-6.** Repeat Problem 6.6-1, but with the process transfer function given by

$$G(s) = \frac{20}{s(3s + 1)}$$

Solve for the unit-step responses for  $0 \leq t \leq 1.0$  s.

# Stability Analysis Techniques

## 7.1 INTRODUCTION

In this chapter stability analysis techniques for linear time-invariant (LTI) discrete-time systems are emphasized. In general, the stability analysis techniques applicable to LTI continuous-time systems may also be applied to the analysis of LTI discrete-time systems, if certain modifications are made. These techniques include the Routh–Hurwitz criterion, root-locus procedures, and frequency-response methods; these techniques are developed in this chapter. The Jury stability test, a technique developed for LTI discrete systems, is also presented.

## 7.2 STABILITY

To introduce the concepts of stability, consider the LTI system shown in Fig. 7-1. For this system,

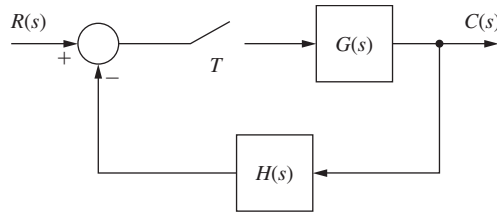
$$C(z) = \frac{G(z)R(z)}{1 + \overline{GH}(z)} = \frac{K \prod_{i=1}^m (z - z_i)}{\prod_{i=1}^n (z - p_i)} R(z)$$

where  $z_i$  are the zeros and  $p_i$  the poles of the system transfer function. Using the partial-fraction expansion, for the case that the transfer-function poles are distinct, we can express  $C(z)$  as

$$C(z) = \frac{k_1 z}{z - p_1} + \cdots + \frac{k_n z}{z - p_n} + C_R(z) \quad (7-1)$$

where  $C_R(z)$  contains the terms of  $C(z)$  which originate in the poles of  $R(z)$ . The first  $n$  terms of (7-1) are the natural-response terms of  $C(z)$ . If the inverse  $z$ -transform of these terms tends to zero as time increases, the system is stable, and these terms are called the *transient response*. The inverse  $z$ -transform of the  $i$ th term is

$$\mathfrak{Z}^{-1} \left[ \frac{k_i z}{z - p_i} \right] = k_i (p_i)^k \quad (7-2)$$



**FIGURE 7-1** Sampled-data system.

Thus, if the magnitude of  $p_i$  is less than 1, this term approaches zero as  $k$  approaches infinity. Note that the factors  $(z - p_i)$  originate in the characteristic equation of the system, that is, in

$$1 + \overline{GH}(z) = 0 \quad (7-3)$$

The system is stable provided that all the roots of (7-3) lie inside the unit circle in the  $z$ -plane. Of course, (7-3) can also be expressed as

$$1 + \overline{GH}^*(s) = 0 \quad (7-4)$$

and since the area within the unit circle of the  $z$ -plane corresponds to the left half of the  $s$ -plane, the roots in (7-4) must lie in the left half of the  $s$ -plane for stability. The system characteristic equation may be calculated by either (7-3) or (7-4).

For the case that a root of the characteristic equation is unity in magnitude (e.g.,  $p_i = 1 \angle \theta$ ), (7-2) is constant in magnitude. Hence the natural response has a term that neither dies out nor becomes unbounded as  $k$  approaches infinity. If the natural response approaches a bounded non-zero steady state, the system is said to be *marginally stable*. Hence, for a marginally stable system, the characteristic equation (7-3) has at least one zero on the unit circle, with no zeros outside the unit circle.

In the development above it was assumed that the poles of the system transfer function are distinct (no repeated poles). It can be shown that the same condition for stability applies if the transfer function has repeated poles (see Problem 7.2-1).

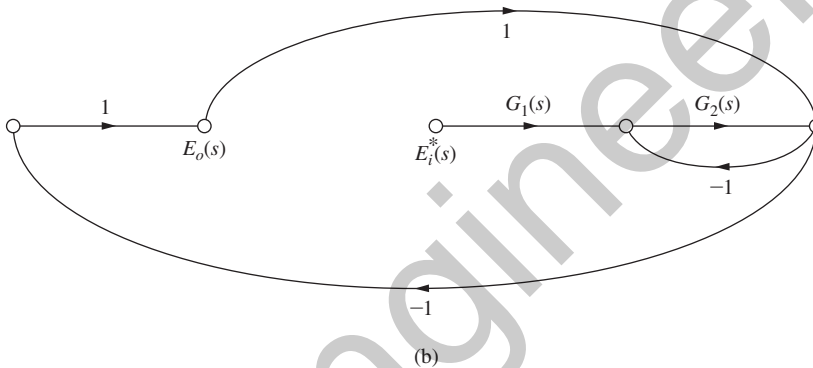
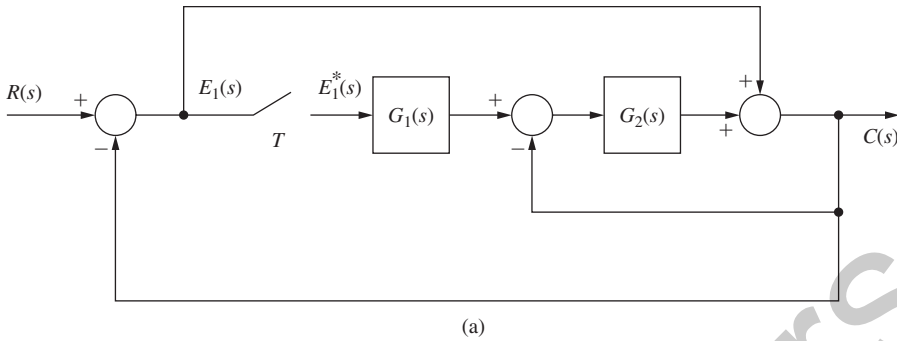
We have demonstrated previously that for certain discrete-time control systems, transfer functions cannot be derived. A method for finding the characteristic equation for control systems of this type will now be developed. To illustrate this method, consider the system of Fig. 7-2(a). This system was considered in Example 5.3, and the output expression developed there is

$$C(z) = \left[ \frac{R}{2 + G_2} \right] (z) + \frac{\left[ \frac{G_1 G_2}{2 + G_2} \right] (z)}{1 + \left[ \frac{G_1 G_2}{2 + G_2} \right] (z)} \left[ \frac{(1 + G_2)R}{2 + G_2} \right] (z)$$

Hence that part of the denominator of  $C(z)$  that is independent of the input  $R$  is

$$1 + \left[ \frac{G_1 G_2}{2 + G_2} \right] (z)$$

and this function set equal to zero is then the characteristic equation.



**FIGURE 7-2** Discrete-time system.

This characteristic function can be developed by a different procedure. Since the stability of a linear system is independent of the input, we set  $R(s) = 0$  in the system. In addition, we open the system in front of a sampler and derive a transfer function at this open. We open at a sampler, since we can always write a transfer function if an input signal is sampled prior to being applied to a continuous-time part of a system. If we opened the system at any other point, we would not be able to find a transfer function. We denote the input signal at this open as  $E_i(s)$ , and the output signal as  $E_o(s)$ . Thus, for the system of Fig. 7-2(a), the system flow graph with no system input and an open at the sampler appears as shown in Fig. 7-2(b). By Mason's gain formula (or any other applicable technique), we write

$$E_o(s) = \frac{-G_1 G_2}{2 + G_2} E_i^*(s)$$

Taking the  $z$ -transform of this equation, we obtain

$$E_o(z) = -\left[ \frac{G_1 G_2}{2 + G_2} \right] (z) E_i(z)$$

We will denote this open-loop transfer function as

$$G_{op}(z) = \frac{E_o(z)}{E_i(z)} = -\left[ \frac{G_1 G_2}{2 + G_2} \right] (z)$$



For the closed-loop system,  $E_i(z) = E_o(z)$ , and the foregoing equations yield

$$[1 - G_{op}(z)]E_o(z) = 0$$

Since we can set initial conditions on the system such that  $E_o(z) \neq 0$ , then

$$1 - G_{op}(z) = 0$$

and this relationship must be the system characteristic equation. Hence the characteristic equation for this system is

$$1 + \left[ \frac{G_1 G_2}{2 + G_2} \right](z) = 0$$

which checks the results of Example 5.3. Another example will now be given.

### EXAMPLE 7.1

Consider the system of Example 5.2, which is repeated in Fig. 7-3(a). The flow graph of the system is opened at the first sampler as shown in Fig. 7-3(b). The effect of the second sampler is included, by denoting its input as  $E_1(s)$  and its output as  $E_1^*(s)$ .

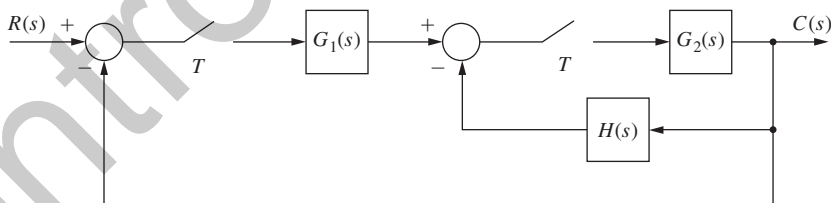
From this flow graph we write

$$E_1 = G_1 E_i^* - G_2 H E_1^*$$

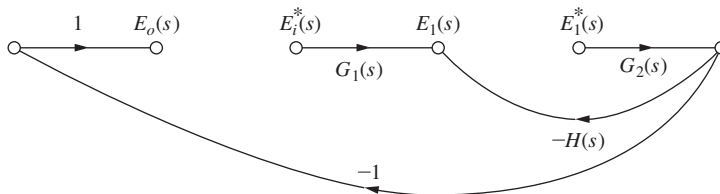
$$E_o = -G_2 E_1^*$$

Starring the first equation and solving for  $E_1^*$ , we obtain

$$E_1^* = \frac{G_1^* E_i^*}{1 + G_2 H^*}$$



(a)



(b)

FIGURE 7-3 System for Example 7.1.

Starting the second equation and substituting in the value for  $E_1^*$  we obtain

$$E_o^* = \frac{-G_1^* G_2^*}{1 + G_2^* H^*} E_i^*$$

Since  $E_i(z) = E_o(z)$  in the closed-loop system,

$$\left[ 1 + \frac{G_1(z)G_2(z)}{1 + G_2H(z)} \right] E_o(z) = 0$$

Thus we can write the characteristic equation

$$1 + G_1(z)G_2(z) + \overline{G_2H(z)} = 0$$

This result is verified in Example 5.2. We leave the derivation of the characteristic equation by opening the system at the second sampler as an exercise for the reader (see Problem 7.2-2).

From the discussion above, in general, the characteristic equation of a discrete system can be expressed as

$$1 + F(z) = 1 - G_{op}(z) = 0 \quad (7-5)$$

where  $G_{op}(z)$  is the *open-loop transfer function*. The function  $F(z)$  is important in analysis and design, and we will call it the *open-loop function*. For the system of Fig. 7-1, the open-loop function is  $\overline{GH(z)}$  and the open-loop transfer function is  $-\overline{GH(z)}$ .

The characteristic equation of an LTI discrete system can also be calculated from a state-variable approach. Suppose that the state-variable model of the system of Fig. 7-1 is

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}r(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) + D r(k) \end{aligned}$$

where the output is now denoted as  $y(k)$  rather than  $c(k)$ . By taking the  $z$ -transform of these state equations and eliminating  $\mathbf{x}(z)$ , it was shown in Chapter 2 [see (2-84)] that the system transfer function is given by

$$\frac{Y(z)}{R(z)} = \mathbf{C}[z\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B} + D$$

The denominator of the transfer function is seen to be  $|z\mathbf{I} - \mathbf{A}|$ , and thus the characteristic equation for the system is given by

$$|z\mathbf{I} - \mathbf{A}| = 0 \quad (7-6)$$

### 7.3 BILINEAR TRANSFORMATION

Many analysis and design techniques for continuous-time LTI systems, such as the Routh-Hurwitz criterion and Bode techniques, are based on the property that in the  $s$ -plane the stability boundary is the imaginary axis. Thus these techniques cannot be applied to LTI discrete-time

systems in the  $z$ -plane since the stability boundary is the unit circle. However, through the use of the transformation

$$z = \frac{1 + (T/2)w}{1 - (T/2)w} \quad (7-7)$$

or solving for  $w$ ,

$$w = \frac{2z - 1}{Tz + 1} \quad (7-8)$$

the unit circle of the  $z$ -plane transforms into the imaginary axis of the  $w$ -plane. This can be seen through the following development. On the unit circle in the  $z$ -plane,  $z = \epsilon^{j\omega T}$  and

$$w = \frac{2z - 1}{Tz + 1} \Big|_{z=\epsilon^{j\omega T}} = \frac{2\epsilon^{j\omega T} - 1}{T\epsilon^{j\omega T} + 1} = \frac{2\epsilon^{j\omega T/2} - \epsilon^{-j\omega T/2}}{T\epsilon^{j\omega T/2} + \epsilon^{-j\omega T/2}} = j\frac{2}{T} \tan \frac{\omega T}{2} \quad (7-9)$$

Thus it is seen that the unit circle of the  $z$ -plane transforms into the imaginary axis of the  $w$ -plane. The mappings of the primary strip of the  $s$ -plane into both the  $z$ -plane ( $z = \epsilon^{sT}$ ) and the  $w$ -plane are shown in Fig. 7-4. It is noted that the stable region of the  $w$ -plane is the left half-plane.

Let  $j\omega_w$  be the imaginary part of  $w$ . We will refer to  $\omega_w$  as the  $w$ -plane frequency. Then (7-9) can be expressed as

$$\omega_w = \frac{2}{T} \tan \frac{\omega T}{2} \quad (7-10)$$

and this expression gives the relationship between frequencies in the  $s$ -plane and frequencies in the  $w$ -plane.

For small values of real frequency ( $s$ -plane frequency) such that  $\omega T$  is small, (7-10) becomes

$$\omega_w = \frac{2}{T} \tan \frac{\omega T}{2} \approx \frac{2}{T} \left( \frac{\omega T}{2} \right) = \omega \quad (7-11)$$

Thus the  $w$ -plane frequency is approximately equal to the  $s$ -plane frequency for this case. The approximation is valid for those values of frequency for which  $\tan(\omega T/2) \cong \omega T/2$ . For

$$\frac{\omega T}{2} \leq \frac{\pi}{10}, \quad \omega \leq \frac{2\pi}{10T} = \frac{\omega_s}{10} \quad (7-12)$$

the error in this approximation is less than 4 percent. Because of the phase lag introduced by the zero-order hold, we usually choose the sample period  $T$  such that (7-12) is satisfied over most if not all of the band of frequencies that the system will pass (the system bandwidth). At  $\omega = \omega_s/10$ , the zero-order hold introduces a phase lag of  $18^\circ$  (see Fig. 3-13), which is an appreciable amount and, as we shall see, can greatly affect system stability.

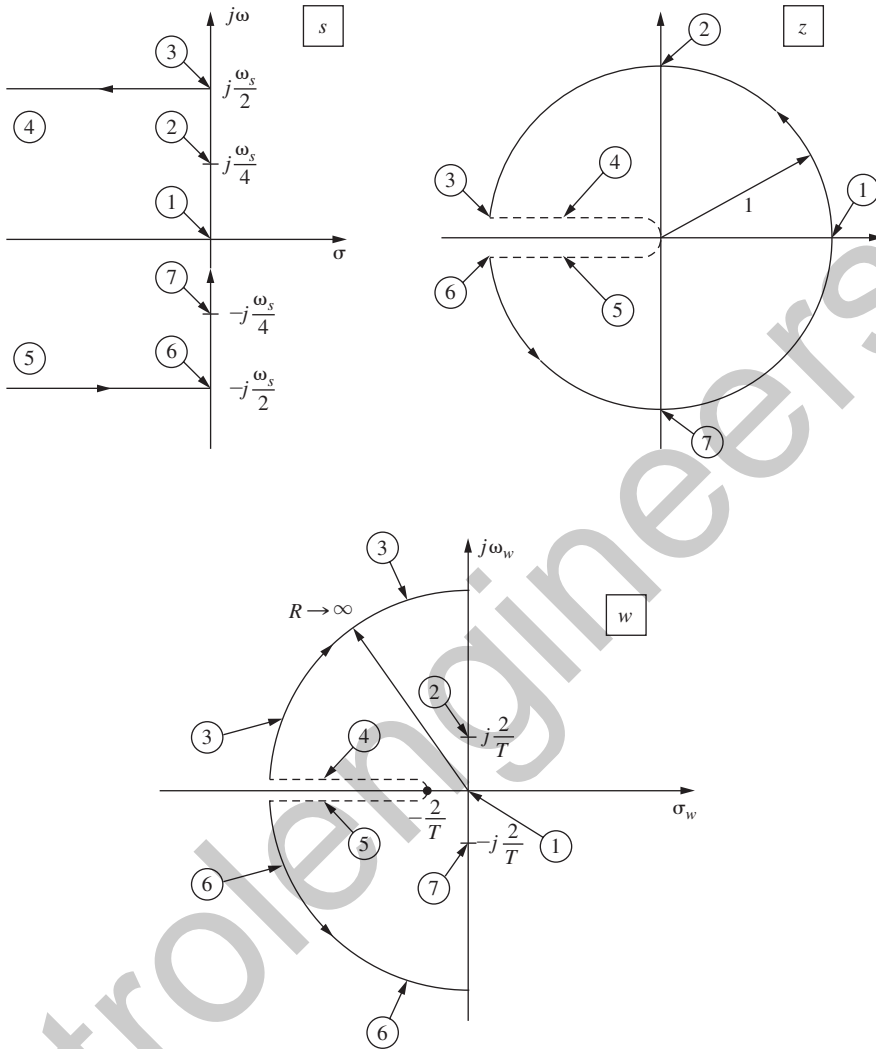


FIGURE 7-4 Mapping from  $s$ -plane to  $z$ -plane to  $w$ -plane.

### 7.4 THE ROUTH–HURWITZ CRITERION

The Routh–Hurwitz criterion [1] may be used in the analysis of LTI continuous-time systems to determine if any roots of a given equation are in the right half of the  $s$ -plane. If this criterion is applied to the characteristic equation of an LTI discrete-time system when expressed as a function of  $z$ , no useful information on stability is obtained. However, if the characteristic equation is expressed as a function of the bilinear transform variable  $w$ , then the stability of the system may be determined by directly applying the Routh–Hurwitz criterion.

We assume that the reader is familiar with the procedures for applying the Routh–Hurwitz criterion. The procedure is summarized in Table 7-1. The technique will now be illustrated via examples.

**TABLE 7-1 Basic Procedure for Applying the Routh–Hurwitz Criterion**

1. Given a characteristic equation of the form

$$F(w) = b_n w^n + b_{n-1} w^{n-1} + \cdots + b_1 w + b_0 = 0$$

form the Routh array as

$w^n$	$b_n$	$b_{n-2}$	$b_{n-4}$	$\cdots$
$w^{n-1}$	$b_{n-1}$	$b_{n-3}$	$b_{n-5}$	$\cdots$
$w^{n-2}$	$c_1$	$c_2$	$c_3$	$\cdots$
$\vdots$	$d_1$	$d_2$	$d_3$	$\cdots$
$w^1$	$j_1$			
$w^0$	$k_1$			

2. Only the first two rows of the array are obtained from the characteristic equation. The remaining rows are calculated as follows.

$$c_1 = \frac{b_{n-1}b_{n-2} - b_n b_{n-3}}{b_{n-1}} \qquad d_1 = \frac{c_1 b_{n-3} - b_{n-1} c_2}{c_1}$$

$$c_2 = \frac{b_{n-1}b_{n-4} - b_n b_{n-5}}{b_{n-1}} \qquad d_2 = \frac{c_1 b_{n-5} - b_{n-1} c_3}{c_1}$$

$$c_3 = \frac{b_{n-1}b_{n-6} - b_n b_{n-7}}{b_{n-1}} \qquad \vdots$$

3. Once the array has been formed, the Routh–Hurwitz criterion states that the number of roots of the characteristic equation with positive real parts is equal to the number of sign changes of the coefficients in the first column of the array.
4. Suppose that the  $w^{i-2}$  th row contains only zeros, and that the  $w^i$  th row above it has the coefficients  $\alpha_1, \alpha_2, \dots$ . The auxiliary equation is then

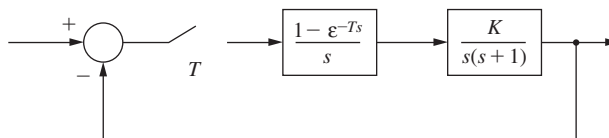
$$\alpha_1 w^i + \alpha_2 w^{i-2} + \alpha_3 w^{i-4} + \cdots = 0$$

This equation is a factor of the characteristic equation.

### EXAMPLE 7.2

Consider the system of Example 6.7 shown in Fig. 7-5, with  $T = 0.1$  s. With  $K = 1$ , the open-loop function is

$$G(s) = \frac{1 - \epsilon^{-Ts}}{s} \left[ \frac{1}{s(s+1)} \right]$$



**FIGURE 7-5** System for Examples 7.2 and 7.3.

Hence, from the  $z$ -transform tables we obtain

$$G(z) = \frac{z-1}{z} \left[ \frac{(\epsilon^{-T} + T - 1)z^2 + (1 - \epsilon^{-T} - T\epsilon^{-T})z}{(z-1)^2(z - \epsilon^{-T})} \right]$$

$$= \frac{0.00484z + 0.00468}{(z-1)(z-0.905)}$$

Then  $G(w)$  is given by

$$G(w) = G(z) \Big|_{z=[1+(T/2)w]/[1-(T/2)w]} = G(z) \Big|_{z=(1+0.05w)/(1-0.05w)}$$

or

$$G(w) = \frac{-0.0000420w^2 - 0.0491w + 1}{w^2 + 0.997w}$$

The bilinear transform can also be computed using MATLAB:

```
>> T=0.1; numz=[0.00484 0.00468]; denz=[1 -1.905 0.905];
Gz=tf(numz,denz,T);
Gw=d2c(Gz,'tustin') %Bilinear transform from discrete to continuous; d2c
returns a function of s, but is w in our notation here

Gw =
-4.199e-05 s^2 - 0.04913 s + 0.9995
-----
s^2 + 0.9974 s + 4.429e-15
```

Then the characteristic equation for any value of  $K$  is given by

$$1 + KG(w) = (1 - 0.000042K)w^2 + (0.997 - 0.0491K)w + K = 0$$

The Routh array derived from this equation is

$w^2$	$1 - 0.000042K$	$3.81K$	$\Rightarrow K < 23,800$
$w^1$	$0.997 - 0.491K$		$\Rightarrow K < 20.3$
$w^0$	$K$		$\Rightarrow K > 0$

Hence, for no sign changes to occur in the first column, it is necessary that  $K$  be in the range  $0 < K < 20.3$ , and this is the range of  $K$  for stability.

### EXAMPLE 7.3

Consider again the system of Example 6.4 (shown again in Fig. 7-5 with  $T = 1$  s) with a gain factor  $K$  added to the plant. This bilinear transform can be computed using MATLAB:

```
>> T=1; numz=[0.368 0.264]; denz=[1 -1.368 0.368];
Gz=tf(numz,denz,T)
Gw=d2c(Gz,'tustin') %Bilinear transform; change s to w.
```

$$Gw = \frac{-0.03801 s^2 - 0.386 s + 0.924}{s^2 + 0.924 s - 2.052e-16}$$

Then the characteristic equation is given by

$$1 + KG(w) = (1 - 0.03801K)w^2 + (0.924 - 0.386K)w + 0.924K = 0$$

The Routh array derived from this equation is

$w^2$	$1 - 0.03801 K$	$0.924 K$	$\Rightarrow K < 26.3$
$w^1$	$0.924 - 0.386 K$		$\Rightarrow K < 2.39$
$w^0$	$0.924 K$		$\Rightarrow K > 0$

Hence the system is stable for  $0 < K < 2.39$ .

From our knowledge of continuous-time systems, we know that the Routh–Hurwitz criterion can be used to determine the value of  $K$  at which the root locus crosses into the right half-plane (i.e., the value of  $K$  at which the system becomes unstable). That value of  $K$  is the gain at which the system is *marginally stable*, and thus can also be used to determine the resultant frequency of steady-state oscillation. Therefore,  $K = 2.39$  in Example 7.3 is the gain for which the system is marginally stable.

In a manner similar to that employed in continuous-time systems, the frequency of oscillation at  $K = 2.39$  can be found from the  $w^2$  row of the array. Recalling that  $\omega_w$  is the imaginary part of  $w$ , we obtain the auxiliary equation (see Table 7-1)

$$(1 - 0.03801K)w^2 + 0.924K|_{K=2.39} = 0.9092w^2 + 2.2084 = 0$$

or

$$w = \pm j \sqrt{\frac{2.2084}{0.9092}} = \pm j1.5585$$

Then  $\omega_w = 1.5585$  and from (7-10),

$$\omega = \frac{2}{T} \tan^{-1} \frac{\omega_w T}{2} = \frac{2}{1} \tan^{-1} \left[ \frac{(1.5585)(1)}{2} \right] = 1.324 \text{ rad/s}$$

and is the  $s$ -plane (real) frequency at which this system will oscillate with  $K = 2.39$ .

The same system was used in both examples in this section, but with different sample periods. For  $T = 0.1$  s, the system is stable for  $0 < K < 20.3$ . For  $T = 1$  s, the system is stable for  $0 < K < 2.39$ . Hence we can see the dependency of system stability on the sample period. The degradation of stability with increasing  $T$  (decreasing sampling frequency) is due to the delay (phase lag) introduced by the sampler and data hold, which is illustrated in the frequency response of the data hold, shown in Fig. 3-13.

## 7.5 JURY'S STABILITY TEST

For continuous-time systems, the Routh–Hurwitz criterion offers a simple and convenient technique for determining the stability of low-ordered systems. However, since the stability boundary in the  $z$ -plane is different from that in the  $s$ -plane, the Routh–Hurwitz criterion cannot be directly applied

to discrete-time systems if the system characteristic equation is expressed as a function of  $z$ . A stability criterion for discrete-time systems that is similar to the Routh–Hurwitz criterion and can be applied to the characteristic equation written as a function of  $z$  is the Jury stability test [2].

Jury’s test will now be presented. Let the characteristic equation of a discrete-time system be expressed as

$$Q(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0 = 0, \quad a_n > 0 \quad (7-13)$$

Then form the array as shown in Table 7-2. Note that the elements of each of the even-numbered rows are the elements of the preceding row in reverse order. The elements of the odd-numbered rows are defined as

$$\begin{aligned}
 b_k &= \begin{vmatrix} a_0 & a_{n-k} \\ a_n & a_k \end{vmatrix}, & c_k &= \begin{vmatrix} b_0 & b_{n-1-k} \\ b_{n-1} & b_k \end{vmatrix} \\
 d_k &= \begin{vmatrix} c_0 & c_{n-2-k} \\ c_{n-2} & c_k \end{vmatrix} \dots
 \end{aligned} \quad (7-14)$$

The necessary and sufficient conditions for the polynomial  $Q(z)$  to have no roots outside or on the unit circle, with  $a_n > 0$ , are as follows:

$$\begin{aligned}
 Q(1) &> 0 \\
 (-1)^n Q(-1) &> 0 \\
 |a_0| &< a_n \\
 |b_0| &> |b_{n-1}| \\
 |c_0| &> |c_{n-2}| \\
 |d_0| &> |d_{n-3}| \\
 &\vdots \\
 |m_0| &> |m_2|
 \end{aligned} \quad (7-15)$$

**TABLE 7-2** Array for Jury’s Stability Test

$z^0$	$z^1$	$z^2$	...	$z^{n-k}$	...	$z^{n-1}$	$z^n$
$a_0$	$a_1$	$a_2$	...	$a_{n-k}$	...	$a_{n-1}$	$a_n$
$a_n$	$a_{n-1}$	$a_{n-2}$	...	$a_k$	...	$a_1$	$a_0$
$b_0$	$b_1$	$b_2$	...	$b_{n-k}$	...	$b_{n-1}$	
$b_{n-1}$	$b_{n-2}$	$b_{n-3}$	...	$b_{k-1}$	...	$b_0$	
$c_0$	$c_1$	$c_2$	...	$c_{n-k}$	...		
$c_{n-2}$	$c_{n-3}$	$c_{n-4}$	...	$c_{k-2}$	...		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		
$l_0$	$l_1$	$l_2$	$l_3$				
$l_3$	$l_2$	$l_1$	$l_0$				
$m_0$	$m_1$	$m_2$					



Note that for a second-order system, the array contains only one row. For each additional order, two additional rows are added to the array. Note also that for an  $n$ th-order system, there are a total of  $n + 1$  constraints.

Jury's test may be applied in the following manner:

1. Check the three conditions  $Q(1) > 0$ ,  $(-1)^n Q(-1) > 0$ , and  $|a_0| < a_n$ , which requires no calculations. Stop if any of these conditions are not satisfied.
2. Construct the array, checking the conditions of (7-15) as each row is calculated. Stop if any condition is not satisfied.

### EXAMPLE 7.4

Consider again the system of Example 6.4 (and Example 7.3). Suppose that a gain factor  $K$  is added to the plant, and it is desired to determine the range of  $K$  for which the system is stable. Now, from Example 6.4, the system characteristic equation is

$$1 + KG(z) = 1 + \frac{(0.368z + 0.264)K}{z^2 - 1.368z + 0.368} = 0$$

or

$$z^2 + (0.368K - 1.368)z + (0.368 + 0.264K) = 0$$

The Jury array is

$z^0$	$z^1$	$z^2$
$0.368 + 0.264K$	$0.368K - 1.368$	$1$

The constraint  $Q(1) > 0$  yields

$$1 + (0.368K - 1.368) + (0.368 + 0.264K) = 0.632K > 0 \Rightarrow K > 0$$

The constraint  $(-1)^2 Q(-1) > 0$  yields

$$1 - 0.368K + 1.368 + 0.368 + 0.264K > 0 \Rightarrow K < \frac{2.736}{0.104} = 26.3$$

The constraint  $|a_0| < a_2$  yields

$$0.368 + 0.264K < 1 \Rightarrow K < \frac{0.632}{0.264} = 2.39$$

Thus the system is stable for

$$0 < K < 2.39$$

The system is marginally stable for  $K = 2.39$ . For this value of  $K$ , the characteristic equation is

$$z^2 + (0.368K - 1.368)z + (0.368 + 0.264K)|_{K=2.39} = z^2 - 0.488z + 1 = 0$$

The roots of this equation are

$$z = 0.244 \pm j0.970 = 1\angle(\pm 75.9^\circ) = 1\angle(\pm 1.32 \text{ rad}) = 1\angle(\pm \omega T)$$

Since  $T = 1$  s, the system will oscillate at a frequency of 1.32 rad/s. These results verify those of Example 7.3.

### EXAMPLE 7.5

Suppose that the characteristic equation for a closed-loop discrete-time system is given by the expression

$$Q(z) = z^3 - 1.8z^2 + 1.05z - 0.20 = 0$$

The first conditions of Jury's test are

$$\begin{aligned} Q(1) &= 1 - 1.8 + 1.05 - 0.2 = 0.05 > 0 \\ (-1)^3 Q(-1) &= -[-1 - 1.8 - 1.05 - 0.2] = 4.05 > 0 \\ |a_0| &= 0.2 < a_3 = 1 \end{aligned}$$

The Jury array is calculated to be

$z^0$	$z^1$	$z^2$	$z^3$
-0.2	1.05	-1.8	1
1	-1.8	1.05	-0.2
-0.96	1.59	-0.69	

where the last row has been calculated as follows:

$$\begin{aligned} b_0 &= \begin{vmatrix} -0.2 & 1 \\ 1 & -0.2 \end{vmatrix} = -0.96, & b_1 &= \begin{vmatrix} -0.2 & -1.8 \\ 1 & 1.05 \end{vmatrix} = 1.59 \\ b_2 &= \begin{vmatrix} -0.2 & 1.05 \\ 1 & -1.8 \end{vmatrix} = -0.69 \end{aligned}$$

Hence the last condition is

$$|b_0| = 0.96 > |b_2| = 0.69$$

Therefore, since all conditions are satisfied, the system is stable. The characteristic equation can be factored as

$$Q(z) = (z - 0.5)^2 (z - 0.8)$$

This form of the equation clearly indicates the system's stability.

### EXAMPLE 7.6

In this example we wish to determine the range of values of  $K_p$  in Example 6.8 so that the system is stable. Recall that in Example 6.8, a proportional-plus-integral (PI) compensator was designed to satisfy certain steady-state constraints. In this example we derive the requirements for this compensated system to be stable. The characteristic equation for the system is given by the expression

$$1 + D(z)G(z) = 0$$

Or, from Example 6.8,

$$1 + \left[ \frac{(K_I + K_p)z - K_p}{z - 1} \right] \left[ \frac{1 - \epsilon^{-T}}{z - \epsilon^{-T}} \right] = 0$$

Thus

$$z^2 - [(1 + \epsilon^{-T}) - (1 - \epsilon^{-T})(K_I + K_p)]z + \epsilon^{-T} - (1 - \epsilon^{-T})K_p = 0$$

Suppose that  $T = 0.1$  s and  $K_I = 100T = 10$ , as required in Example 6.8. Then the characteristic equation becomes

$$z^2 - (0.953 - 0.0952K_p)z + 0.905 - 0.0952K_p = 0$$

For this system, the Jury array is

$z^0$	$z^1$	$z^2$
$0.905 - 0.0952K_p$	$0.0952K_p - 0.953$	1

The constraint  $Q(1) > 0$  yields

$$1 + 0.0952K_p - 0.953 + 0.905 - 0.0952K_p > 0$$

Thus this constraint is satisfied independent of  $K_p$ . The constraint  $(-1)^2 Q(-1) > 0$  yields

$$1 - 0.0952K_p + 0.953 + 0.905 - 0.0952K_p > 0$$

or since  $K_p$  is normally positive,

$$0 < K_p < 15.01$$

The constraint  $|a_0| < a_2$  yields

$$|0.905 - 0.0952K_p| < 1$$

or

$$K_p < 20.0$$

Thus the stability constraint on positive  $K_p$  is  $K_p < 15.01$ , in order that the steady-state error constraint in Example 6.8 be satisfied.

## 7.6 ROOT LOCUS

For the LTI sampled-data system of Fig. 7-6,

$$\frac{C(z)}{R(z)} = \frac{KG(z)}{1 + KGH(z)}$$

The system characteristic equation is, then,

$$1 + \overline{KGH}(z) = 0 \quad (7-16)$$

The root locus for this system is a plot of the locus of roots in (7-16) in the  $z$ -plane as a function of  $K$ . Thus the rules of root-locus construction for discrete-time systems are identical to those for continuous-time systems, since the roots of any equation are dependent only on the coefficients of the equation and are independent of the designation of the variable. Since the rules for root-locus construction are numerous and appear in any standard text for continuous-time control systems [1, 4], only the most important rules will be repeated here in abbreviated form. These rules are given in Table 7-3.

Although the rules for the construction of both  $s$ -plane and  $z$ -plane root loci are the same, there are important differences in the interpretation of the root loci. For example, in the  $z$ -plane, the stable region is the interior of the unit circle. In addition, root locations in the  $z$ -plane have different meanings from those in the  $s$ -plane from the standpoint of the system time response, as was seen in Fig. 6-11.

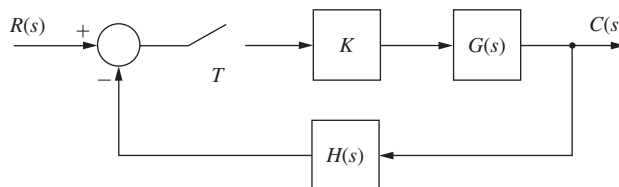


FIGURE 7-6 Sampled-data system.

**TABLE 7-3 Rules for Root-Locus Construction**

For the characteristic equation

$$1 + K\overline{GH}(z) = 0$$

1. Loci originate on poles of  $\overline{GH}(z)$  and terminate on the zeros of  $\overline{GH}(z)$ .
2. The root locus on the real axis always lies in a section of the real axis to the left of an odd number of poles and zeros on the real axis.
3. The root locus is symmetrical with respect to the real axis.
4. The number of asymptotes is equal to the number of poles of  $\overline{GH}(z)$ ,  $n_p$ , minus the number of zeros of  $\overline{GH}(z)$ ,  $n_z$ , with angles given by  $(2k + 1)\pi/(n_p - n_z)$ .
5. The asymptotes intersect the real axis at  $\sigma$ , where

$$\sigma = \frac{\sum \text{poles of } \overline{GH}(z) - \sum \text{zeros of } \overline{GH}(z)}{n_p - n_z}$$

6. The breakaway points are given by the roots of

$$\frac{d[\overline{GH}(z)]}{dz} = 0$$

or, equivalently,

$$\text{den}(z) \frac{d(\text{num}(z))}{dz} - \text{num}(z) \frac{d(\text{den}(z))}{dz} = 0, \quad \overline{GH}(z) = \frac{\text{num}(z)}{\text{den}(z)}$$

The two examples that follow review not only the root-locus technique, but also the part it plays in stability analyses.

### EXAMPLE 7.7

Consider again the system of Example 6.4. For this system

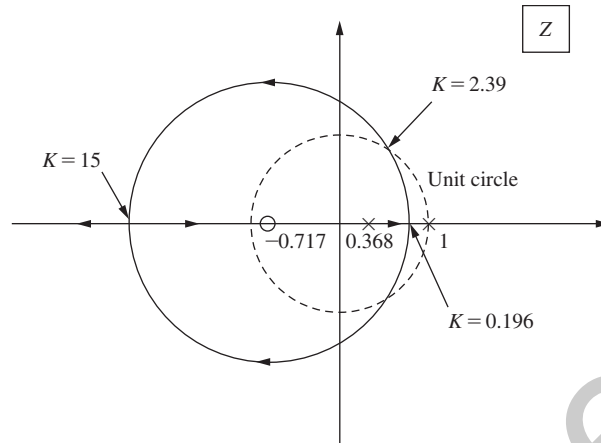
$$KG(z) = \frac{0.368K(z + 0.717)}{(z - 1)(z - 0.368)}$$

Thus the loci originate at  $z = 1$  and  $z = 0.368$ , and terminate at  $z = -0.717$  and  $z = \infty$ . There is one asymptote, at  $180^\circ$ . The breakaway points, obtained from

$$\frac{d}{dz}[G(z)] = 0$$

occur at  $z = 0.65$  for  $K = 0.196$ , and  $z = -2.08$  for  $K = 15.0$ . The root locus is then as shown in Fig. 7-7.

The points of intersection of the root loci with the unit circle may be found by graphical construction, the Jury stability test, or the Routh–Hurwitz criterion. To illustrate the use of the Jury stability test, consider the results of Example 7.4. The value of gain for marginal stability



**FIGURE 7-7** Root locus for Example 7.7.

(i.e., for the roots to appear on the unit circle) is  $K = 2.39$ . For this value of gain, the characteristic equation is, from Example 7.4,

$$z^2 - 0.488z + 1 = 0$$

The roots of this equation are

$$z = 0.244 \pm j0.970 = 1 \angle (\pm 75.8^\circ) = 1 \angle (\pm 1.32 \text{ rad}) = 1 \angle (\omega T)$$

and thus these are the points at which the root locus crosses the unit circle. Note that the frequency of oscillation for this case is  $\omega = 1.32 \text{ rad/s}$ , since  $T = 1 \text{ s}$ . This was also calculated in Example 7.3 using the Routh–Hurwitz criterion, and in Example 7.4 using the Jury test.

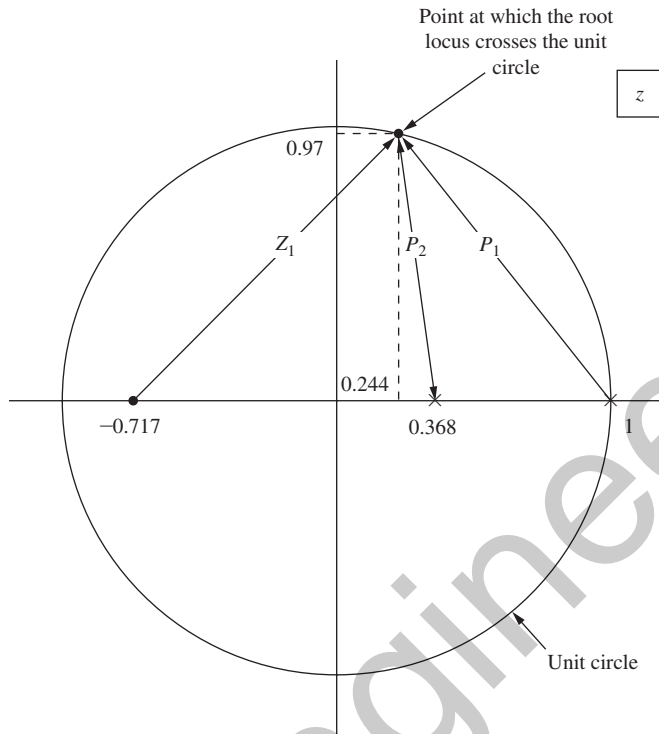
The value of the gain at points where the root locus crosses the unit circle can also be determined using the root-locus condition that at any point along the locus the magnitude of the open-loop function must be equal to 1 [i.e.,  $|KGH(z)| = 1$ ]. Using the condition and Fig. 7-8, we note that

$$\frac{0.368K(Z_1)}{(P_1)(P_2)} = 1$$

From Fig. 7-8 the following values can be calculated:  $Z_1 = 1.364$ ,  $P_1 = 1.229$ , and  $P_2 = 0.978$ . Using these values in the equation above yields  $K = 2.39$ . A MATLAB program that solves for and plots a root locus for this example is given by

```
>> T = 1;
num = [0 0.368 0.264];
den = [1 -1.368 0.368];
Gz = tf(num,den,T);
rlocus(Gz)
axis([-3 2 -2 2]) %Set to match Figure 7-7
```

Note that the breakaway points and their respective gains  $K$  can be found directly from the MATLAB *rlocus* plot, as can the gain and locations of the roots as they cross the unit circle.



**FIGURE 7-8** Determination of the system gain at the crossover point on the unit circle.

### EXAMPLE 7.8

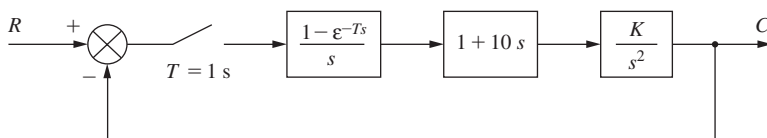
Consider the system shown in Fig. 7-9. The transfer function  $[1 + 10s]$  is that of an analog proportional-plus-derivative (PD) controller [1]. The digital PD controller is covered in Chapter 8. We wish to determine the form of the root locus and the range of  $K$  for stability.

We denote the open-loop function as

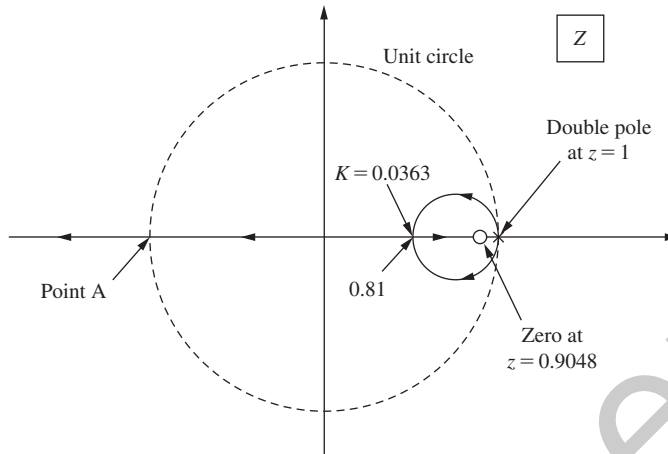
$$KG(s) = \frac{1 - \epsilon^{-sT}}{s} \left[ \frac{K(1 + 10s)}{s^2} \right]$$

Applying the  $z$ -transform, we obtain

$$KG(z) = \frac{10.5K(z - 0.9048)}{(z - 1)^2}$$



**FIGURE 7-9** System for Example 7.8.



**FIGURE 7-10** Root locus for the system of Example 7.8.

The loci originate at  $z = 1$  and terminate at  $z = 0.9048$  and  $z = -\infty$ . There is one asymptote at  $180^\circ$ . The root locus is shown in Fig. 7-10.

The system becomes unstable when the closed-loop pole leaves the interior of the unit circle at point A shown in Fig. 7-10. The value of  $K$  at this point can be determined from the condition  $KG(z) = -1$ . Therefore,

$$\left. \frac{10.5K(z - 0.9048)}{(z - 1)^2} \right|_{z=-1} = \frac{10.5K(-1.9048)}{4} = -1$$

Thus  $K = 0.2$ , and we see that the system is stable for  $0 < K < 0.2$ .

Consider the following MATLAB program that generates Fig. 7-10. The breakaway points and gain limit for stability can be verified directly from the MATLAB-generated root-locus plot.

```
>> T = 1;
Gs = tf([10 1],[1 0 0]);
Gz = c2d(Gs,T); % c2d includes the zero-order hold
rlocus(Gz)
axis([-1.5 1.5 -1.25 1.25])
```

### 7.7 THE NYQUIST CRITERION

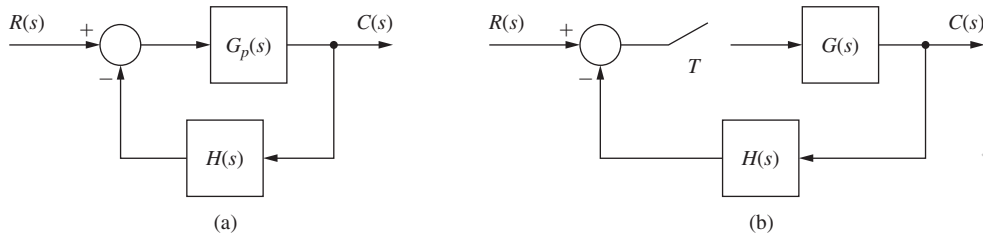
To develop the Nyquist criterion for discrete-time systems, we consider the two systems shown in Fig. 7-11. In Fig. 7-11(b),  $G(s)$  contains the transfer function of a data hold. The transfer function for the continuous-time system of Fig. 7-11(a) is

$$\frac{C(s)}{R(s)} = \frac{G_p(s)}{1 + G_p(s)H(s)} \tag{7-17}$$

and for the sampled-data system of Fig. 7-11(b) the transfer function is

$$\frac{C^*(s)}{R^*(s)} = \frac{G^*(s)}{1 + GH^*(s)} \tag{7-18}$$





**FIGURE 7-11** Continuous-time and sampled-data systems.

Thus the characteristic equation for the continuous-time system is

$$1 + G_p(s)H(s) = 0 \quad (7-19)$$

and for the sampled-data system, the characteristic equation is

$$1 + \overline{GH}^*(s) = 0 \quad (7-20)$$

The characteristic equation of the sampled-data system can also be written as

$$1 + \overline{GH}(z) = 0 \quad (7-21)$$

Recall that the continuous-time system is stable if the roots of (7-19) are all contained in the left half-plane. Similarly, the sampled-data system is stable if the roots of (7-20) all lie in the left half-plane, or if the roots of (7-21) all lie within the unit circle.

The Nyquist criterion is based on Cauchy's principle of argument [5].

**THEOREM.** Let  $f(z)$  be the ratio of two polynomials in  $z$ . Let the closed curve  $C$  in the  $z$ -plane be mapped into the complex plane through the mapping  $f(z)$ . If  $f(z)$  is analytic within and on  $C$ , except at a finite number of poles, and if  $f(z)$  has neither poles nor zeros on  $C$ , then

$$N = Z - P$$

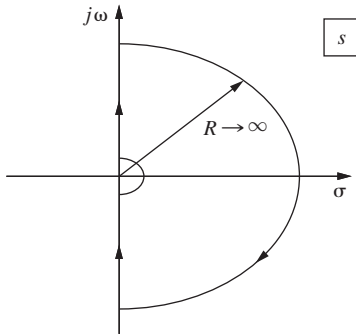
where  $Z$  is the number of zeros of  $f(z)$  in  $C$ ,  $P$  is the number of poles of  $f(z)$  in  $C$ , and  $N$  is the number of encirclements of the origin, taken in the same sense as  $C$ .

**S-PLANE NYQUIST DIAGRAM.** In order to determine the stability of an analog system via the Nyquist criterion, a complex-plane plot of the open-loop function,  $G_p(s)H(s)$ , is made. This plot is referred to as the Nyquist diagram. The closed curve of  $C$  in Cauchy's principle of argument is called the Nyquist path, and for continuous-time systems encloses the right half-plane as shown in Fig. 7-12.

The Nyquist criterion states that if the complex-plane plot of  $G_p(s)H(s)$ , for values of  $s$  on the Nyquist path, is made, then

$$N = Z - P \Rightarrow Z = N + P \quad (7-22)$$

where  $N$  is the number of clockwise encirclements of the  $-1$  point made by the plot of  $G_p(s)H(s)$ ,  $Z$  is the number of zeros of the characteristic equation enclosed by the Nyquist path, and  $P$  is the



**FIGURE 7-12** Nyquist path in the  $s$ -plane.

number of poles of the open-loop function enclosed by the Nyquist path. Since the Nyquist path encloses the right half-plane,  $Z$  in (7-22) must be zero for stability.

**EXAMPLE 7.9**

As an example of the Nyquist criterion applied to a continuous-time system, consider the system of Fig. 7-13. The Nyquist path and the resultant Nyquist diagram are shown in Fig. 7-14. The small detour taken by the Nyquist path around the origin, labeled I on Fig. 7-14(a), is necessary, since  $G_p(s)$  has a pole at the origin. Along this detour, let

$$s = \rho e^{j\theta}, \quad \rho \ll 1$$

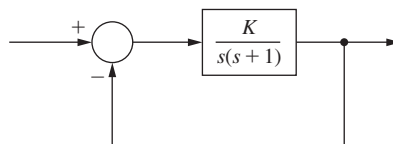
Then

$$G_p(s)|_{s = \rho e^{j\theta}} \approx \frac{K}{\rho e^{j\theta}} = \frac{K}{\rho} \angle(-\theta)$$

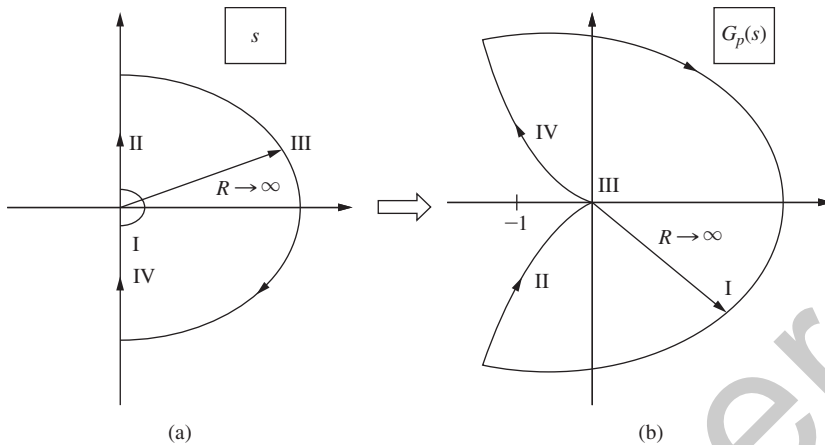
Thus the small detour generates the large arc on the Nyquist diagram. For the Nyquist path along the  $j\omega$ -axis (II),

$$G_p(s)|_{s = \rho e^{j\theta}} \approx \frac{K}{j\omega(1 + j\omega)} = \frac{K}{\omega\sqrt{1 + \omega^2}} \angle(-90^\circ - \tan^{-1}\omega)$$

This portion of the Nyquist diagram is simply the frequency response of the system. Along the large arc of the Nyquist path,  $G_p(s) \approx 0$ . The complete Nyquist diagram is given in Fig. 7-14b. Now let us apply the Nyquist criterion. We can see from the Nyquist diagram in Fig. 7-14



**FIGURE 7-13** System for Example 7.9.



**FIGURE 7-14** Nyquist diagram for Example 7.9.

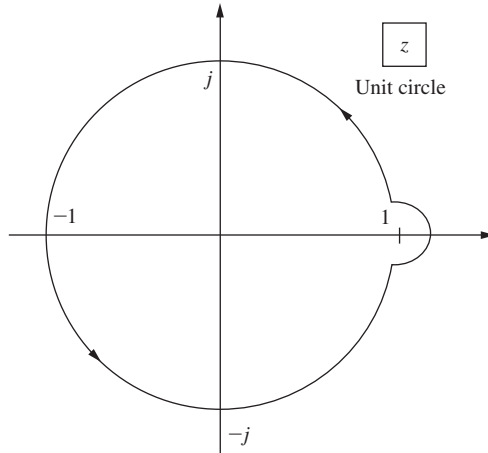
that  $N = 0$ . Since the open-loop transfer function has no poles in the right half-plane,  $P = 0$ , and therefore from the equation  $Z = N + P$  we find that  $Z = 0$ . Thus the system is stable. In addition, from the Nyquist diagram in Fig. 7-14 it can be seen that the system is stable for all  $K > 0$ .

Consider now the sampled-data system of Fig. 7-11(b). The characteristic equation is given in (7-20), and thus the Nyquist diagram for the system may be generated by using the  $s$ -plane Nyquist path of Fig. 7-12 (i.e., the same path as that used for continuous-time systems). However, recall from Chapter 3 that  $\overline{GH}^*(s)$  is periodic in  $s$  with period  $j\omega_s$ . Thus it is necessary only that  $\overline{GH}^*(j\omega)$  be plotted for  $-\omega_s/2 \leq \omega \leq \omega_s/2$ , in order to obtain the frequency response. Consider also the relationship, from (3-11),

$$\begin{aligned} \overline{GH}^*(j\omega) &= \frac{1}{T} \sum_{n=-\infty}^{\infty} GH(j\omega + jn\omega_s) \\ &= \frac{1}{T} [GH(j\omega) + GH(j\omega + j\omega_s) + GH(j\omega + j2\omega_s) + \cdots] \end{aligned} \quad (7-23)$$

Since physical systems are generally low pass,  $\overline{GH}^*(j\omega)$  may be approximated by only a few terms of (7-23). In general, however, the approximation will not apply for  $\omega > \omega_s/2$ . A digital computer program may be written for (7-23), and thus the Nyquist diagram of a sampled-data system may be obtained without calculating the  $z$ -transform of the transfer function.

**z-PLANE NYQUIST DIAGRAM.** The Nyquist diagram may also be generated directly from the  $z$ -plane. The Nyquist path for the  $z$ -plane is the unit circle, and the path direction is counterclockwise, as shown in Fig. 7-15. To apply Cauchy's principle of argument, let  $Z_i$  and  $P_i$  be the zeros of the characteristic equation and the poles of the open-loop function, respectively, inside



**FIGURE 7-15** Nyquist path in the  $z$ -plane.

the unit circle; and let  $Z_o$  and  $P_o$  be the zeros of the characteristic equation and the poles of the open-loop function, respectively, outside the unit circle. Then, from (7-22),

$$N = -(Z_i - P_i) \tag{7-24}$$

where  $N$  is the number of the clockwise encirclements of the  $-1$  point made by the Nyquist diagram of  $\overline{GH}(z)$ . The minus sign appears in (7-24) since the Nyquist path is counterclockwise. Now, in general, the order of the numerator and the order of the denominator of  $1 + \overline{GH}(z)$  are the same. Let this order be  $n$ . Then

$$Z_o + Z_i = n \tag{7-25}$$

$$P_o + P_i = n \tag{7-26}$$

Solving (7-25) and (7-26) for  $Z_i$  and  $P_i$ , respectively, and substituting the results in (7-24), we obtain

$$N = Z_o - P_o \tag{7-27}$$

Thus the Nyquist criterion is given by (7-27), with the Nyquist path shown in Fig. 7-15. The Nyquist diagram for the system of Fig. 7-11(b) is obtained by plotting  $\overline{GH}(z)$  for values of  $z$  on the unit circle. The subscripts in (7-27) are usually omitted; then

$$N = Z - P \Rightarrow Z = N + P \tag{7-28}$$

where  $N$  = number of clockwise encirclements of the  $-1$  point

$Z$  = number of zeros of the characteristic equation outside the unit circle

$P$  = number of poles of the open-loop function, or equivalently poles of the characteristic equation, that are outside the unit circle

### EXAMPLE 7.10

The Nyquist criterion will now be illustrated using the system of Fig. 7-16. From Example 6.4,

$$G(z) = \frac{0.368z + 0.264}{(z - 1)(z - 0.368)}$$

The Nyquist path and the Nyquist diagram are both shown in Fig. 7-17. The detour around the  $z = 1$  point on the Nyquist path (Fig. 7-17(a)) is necessary since  $G(z)$  has a pole at this point. On this detour,

$$z = 1 + \rho e^{j\theta}, \quad \rho \ll 1$$

and

$$G(z)|_{z=1+\rho e^{j\theta}} \approx \frac{0.632}{\rho e^{j\theta}(0.632)} = \frac{1}{\rho} \angle (-\theta)$$

Thus this detour generates the large arc on the Nyquist diagram (Fig. 7-17(b)). For  $z$  on the unit circle,

$$G(z)|_{z=e^{j\omega T}} = \frac{0.368e^{j\omega T} + 0.264}{(e^{j\omega T} - 1)(e^{j\omega T} - 0.368)}$$

In this equation  $\omega$  varies from  $-\omega_s/2$  to  $\omega_s/2$ . Since  $G(e^{j\omega T})$  for  $0 > \omega > -\omega_s/2$  is the complex conjugate of  $G(e^{j\omega T})$  for  $0 < \omega < \omega_s/2$ , it is necessary to calculate  $G(e^{j\omega T})$  only for  $0 < \omega < \omega_s/2$ . This calculation results in the frequency response for  $G(z)$ . Note that  $G(-1) = -0.0381$ .

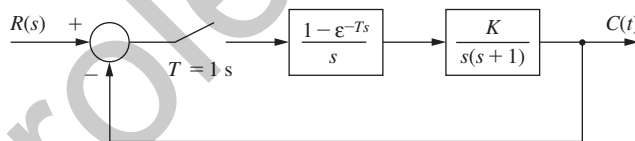


FIGURE 7-16 System for Example 7.10.

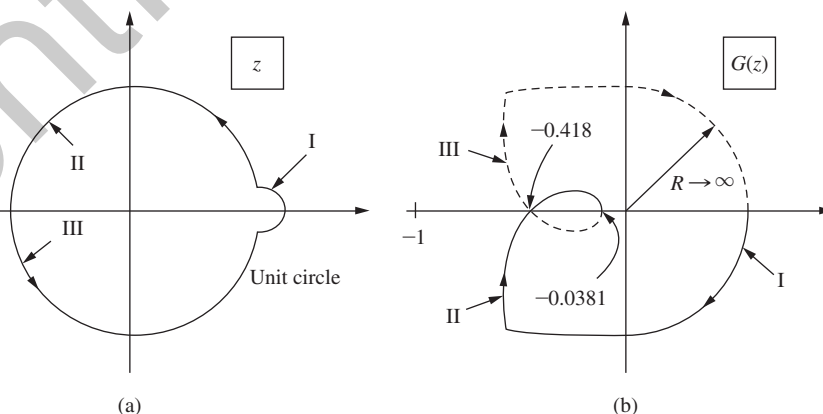


FIGURE 7-17 Nyquist path and Nyquist diagram for Example 7.10.

The calculation of the Nyquist diagram for the portion of the diagram labeled II in Fig. 7-17(b) may be accomplished in several ways. We may use

$$G^*(s) \Big|_{s=j\omega} \quad \text{or} \quad G(z) \Big|_{z=e^{j\omega T}}$$

as the basis for calculation. Generally, computer calculations are required to evaluate the frequency response. A MATLAB program to plot the Nyquist diagram of Fig. 7-17(b) follows.

```
>> T = 1;
num = [0 0.368 0.264];
den = [1 -1.368 0.368];
Gz = tf(num,den,T);
nyquist(Gz)
axis([-1.2 0.2 -0.7 0.7])
```

Note that the point at which the curve crosses the real axis ( $-0.0381$  and  $-0.418$ ) can be determined directly from the MATLAB Nyquist plot.

We see from Fig. 7-17 that  $N = 0$ , since the Nyquist diagram does not encircle the  $-1$  point. In addition,  $P = 0$  since  $G(z)$  has no poles outside the Nyquist path. Hence  $Z = (N + P) = 0$ , and the system is stable.

Suppose that a gain  $K$  is added to the plant. The system can be forced into instability by increasing this gain  $K$  by a factor of  $1/0.418$ , or  $2.39$ . However, the same system without sampling, as shown in Example 7.9, is stable for all positive values of gain  $K$ . As stated earlier, the destabilizing effect of the sampling can be seen from the phase lag introduced by the sampler and data hold.

As an additional point in Example 7.10, the analog system *model* is stable for all positive values of plant gain  $K$ . This does not carry over to the physical system that is being modeled. The second-order model will be accurate only over limited signal levels. A large increase in gain within the loop will result in large signals, and in general, the linear second-order model will no longer be accurate.

### EXAMPLE 7.11

Consider again the system of Example 7.10. For this system

$$G(z) = \frac{0.368z + 0.264}{z^2 - 1.368z + 0.368}$$

Then  $G(w)$  is given by, from Example 7.3,

$$G(w) = \frac{-0.0381w^2 - 0.386w + 0.924}{w^2 + 0.924w}$$

The Nyquist diagram can be obtained from this equation by allowing  $w$  to assume values from  $j0$  to  $j\infty$ . Since  $G(w)$  has a pole at the origin, the Nyquist path must detour around this point. The Nyquist diagram generated by MATLAB program below for this  $G(w)$  is, within numerical accuracy, the same as the one generated using  $G(z)$  as shown in Fig. 7-17.

```
>> num = [-0.0381 -0.368 0.924];
den = [1 0.924 0];
nyquist(num,den)
axis([-1.2 0.2 -0.7 0.7])
```

**TABLE 7-4** Functions for Calculating the Nyquist Diagram

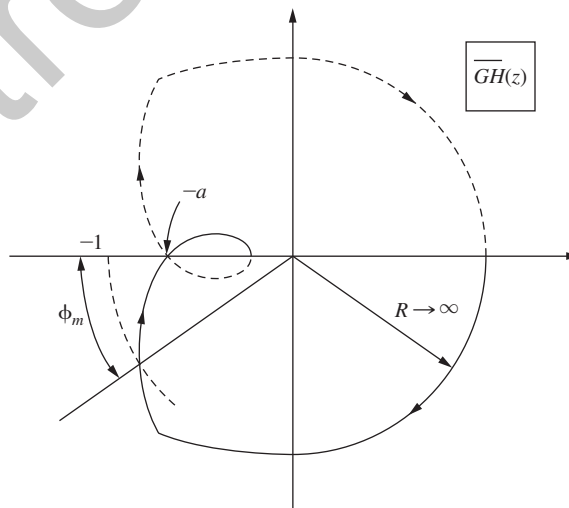
Open-loop function	Range of variable
$\overline{GH}^*(s)$	$s = j\omega, \quad 0 \leq \omega \leq \omega_s/2$
$\overline{GH}(z)$	$z = e^{j\omega T}, \quad 0 \leq \omega T \leq \pi$
$\overline{GH}^*(w)$	$w = j\omega_w, \quad 0 \leq \omega_w < \infty$

Three different transfer functions may be used to generate the Nyquist diagram:  $\overline{GH}^*(s)$ ,  $\overline{GH}(z)$ , and  $\overline{GH}(w)$ . Of course, the Nyquist diagrams generated will be identical in each case. Table 7-4 gives the functions and the range of variables required for the calculation of the Nyquist diagram. The range of variables gives only the upper half of the Nyquist path. If a pole of the open-loop function occurs in the range of the variable, a detour must be made around the point that the pole occurs.

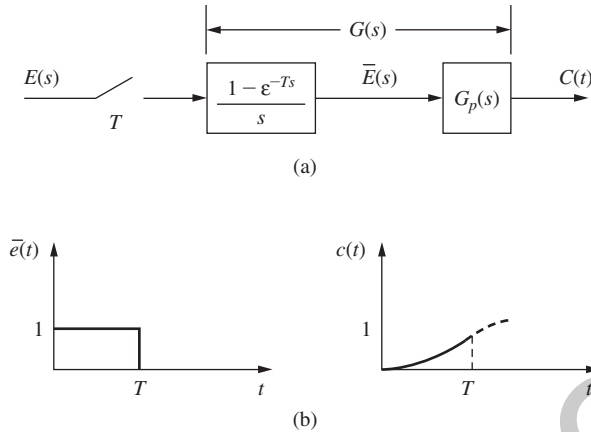
**GAIN AND PHASE MARGINS.** For a stable continuous-time system, the *gain margin* is defined as the factor by which the gain must change to force the system to marginal stability. The *phase margin* is defined as the angle through which the Nyquist diagram must be rotated such that the diagram intersects the  $-1$  point. The gain and phase margin definitions for stable discrete-time systems are exactly the same; that is, in the Nyquist diagram shown in Fig. 7-18, the gain margin is  $1/a$ , and the phase margin is  $\phi_m$ .

**PULSE TRANSFER FUNCTIONS.** At this point a discussion of certain properties of pulse transfer functions is in order. For the system of Fig. 7-19(a),

$$C(z) = G(z)E(z)$$



**FIGURE 7-18** Nyquist diagram illustrating stability margins.



**FIGURE 7-19** Discrete unit pulse response.

where we can express  $G(z)$  as

$$G(z) = \frac{b_m z^m + b_{m-1} z^{m-1} + \dots + b_0}{z^n + a_{n-1} z^{n-1} + \dots + a_0} \tag{7-29}$$

By dividing the denominator of  $G(z)$  into its numerator, we can express  $G(z)$  as

$$G(z) = g(n - m) z^{-(n-m)} + g(n - m + 1) z^{-(n-m+1)} + \dots \tag{7-30}$$

The *discrete unit impulse function* is defined by the transform

$$E(z) = 1 \tag{7-31}$$

Thus the discrete unit impulse function is a number sequence  $\{e(k)\}$  that has a value of unity for  $k = 0$ , and a value of zero for  $k = 1, 2, 3, \dots$ . Hence, if in Fig. 7-19,  $E(z)$  is the unit impulse function, then

$$C(z) = G(z)E(z) = G(z) \tag{7-32}$$

Thus (7-30) is the system impulse response.

Fig. 7-19(b) shows  $\bar{e}(t)$  for the case that  $E(z)$  is the unit impulse function. For most physical systems we would expect a response  $c(t)$  as shown in the figure. The plant does not respond instantaneously to the input, but it will have responded to some nonzero value at  $t = T$ . Hence we may express  $C(z)$  as

$$C(z) = c(1)z^{-1} + c(2)z^{-2} + \dots \tag{7-33}$$

Comparing (7-33) with (7-30), we see that, in general,

$$n - m = 1 \tag{7-34}$$

Then, for a physical system, we generally expect the order of the numerator of its pulse transfer function to be one less than the order of the denominator. If the plant can respond instantaneously, the order of the numerator will be equal to the order of the denominator. If the order of the numerator is greater than the order of the denominator, the plant will respond prior to the application of the input. This case, of course, is not physically realizable; the plant is said to be *noncausal*.



## 7.8 THE BODE DIAGRAM

The convenience of frequency-response plots for continuous-time systems in the form of the Bode diagram in both analysis and design stems from the straight-line approximations that are made, and these straight-line approximations are based on the independent variable,  $j\omega$ , being imaginary [1, 2]. Thus Bode diagrams of discrete-time systems may be plotted, using straight-line approximations, provided that the  $w$ -plane form of the transfer function is used. For convenience, a summary of the first-order terms used in the construction of a Bode diagram is given in Table 7-5 and Fig. 7-20. Since terms with complex zeros or poles are not amenable to straight-line approximations, these terms have been omitted.

Let us now employ the Bode diagram in the analysis of a familiar example.

### EXAMPLE 7.12

Consider again the system of Example 7.3. For this system,

$$G(w) = -\frac{0.0381(w - 2)(w + 12.14)}{w(w + 0.924)}$$

and

$$\begin{aligned} G(j\omega_w) &= -\frac{0.0381(j\omega_w - 2)(j\omega_w + 12.14)}{j\omega_w(j\omega_w + 0.924)} \\ &= \frac{-\left(j\frac{\omega_w}{2} - 1\right)\left(\frac{j\omega_w}{12.14} + 1\right)}{j\omega_w\left(\frac{j\omega_w}{0.924} + 1\right)} \end{aligned}$$

Note that the numerator break frequencies are  $\omega_w = 2$  and  $\omega_w = 12.14$  and the denominator break frequencies are  $\omega_w = 0$  and  $\omega_w = 0.924$ . The Bode diagram for this system, using straight-line approximations, is shown in Fig. 7-21. Both the gain and phase margins of the system are shown on the diagram.

As stated in Example 7-10, this system can be made unstable by increasing the gain. This can also be seen in Fig. 7-21. Increasing the gain is equivalent to shifting the entire magnitude curve vertically upward. Therefore, if the gain is increased by an amount equal to the gain margin, a condition of marginal stability will exist. A MATLAB program that plots the Bode diagram for this example is given by

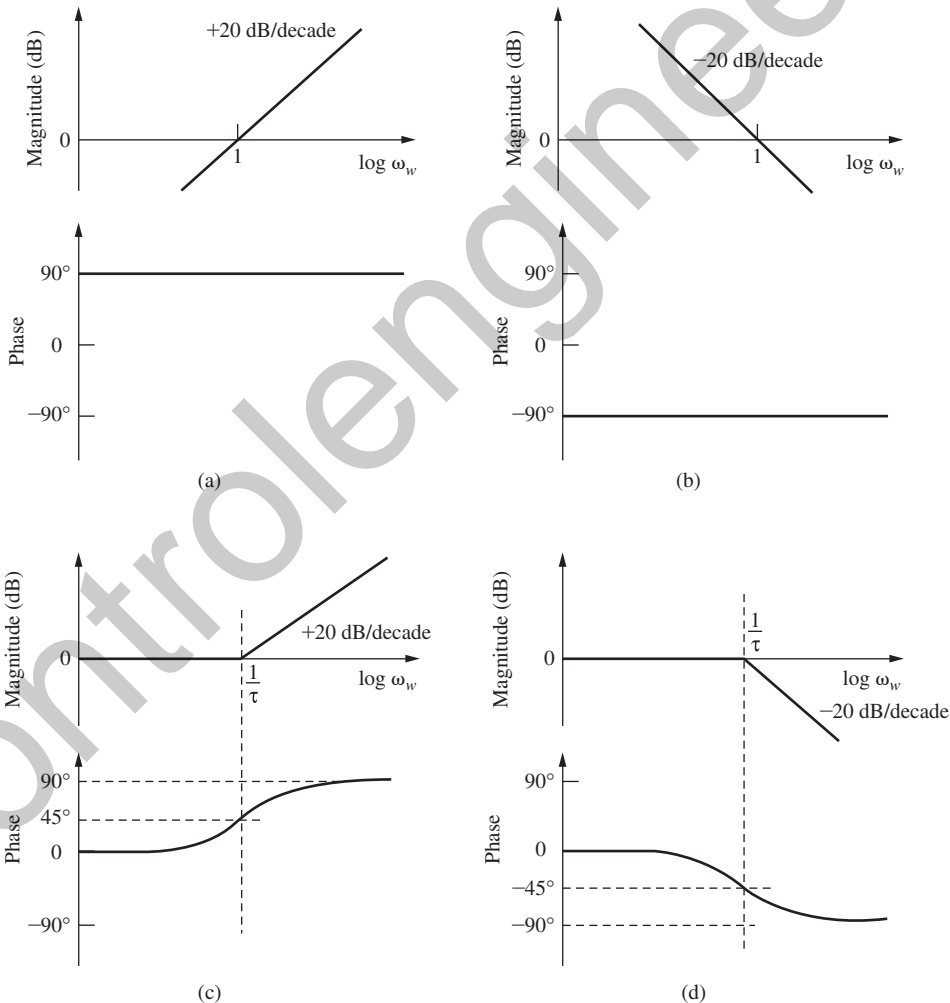
```
>>num = [-0.0381 -0.386 0.924];
den = [1 0.924 0];
Gw = tf(num,den);
bode(Gw), grid
```

The Bode diagram can also be plotted from  $G(z)$  using MATLAB.

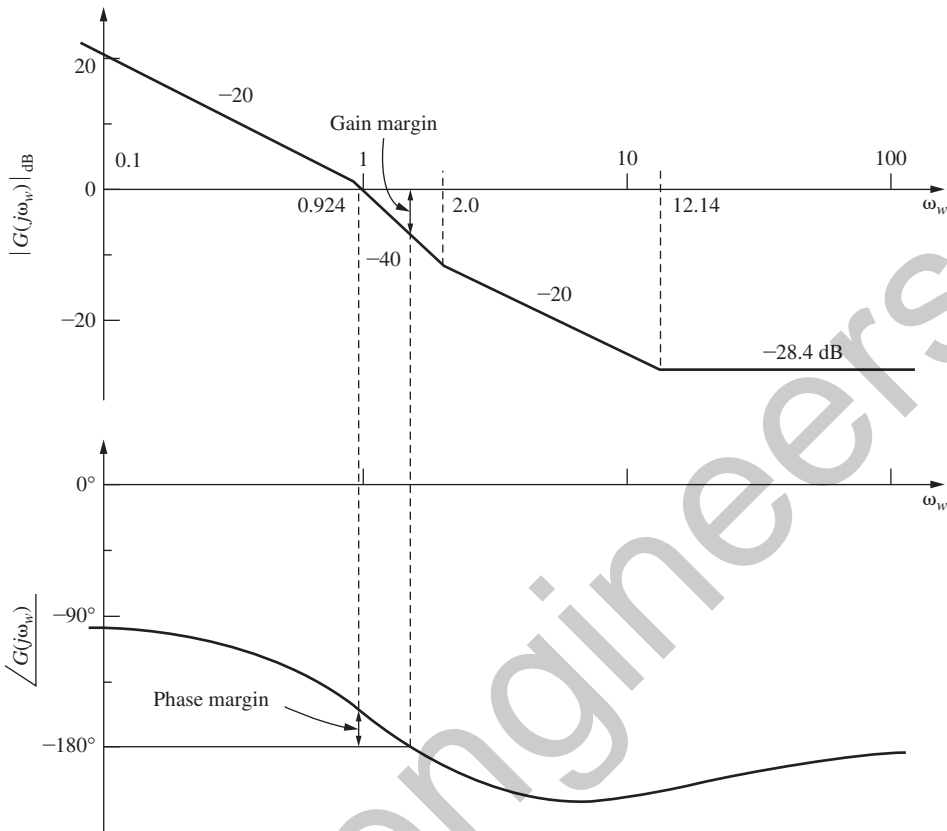
```
>> % G(s) for this example is given in Figure 6-2
num = [1]; den = [1 1 0]; Gs = tf(num,den); T=1;
Gz=c2d(Gs,T); %Assumes a zero-order hold
margin(Gz) %margin generates a Bode plot with the gain and phase margins
denoted.
```

**TABLE 7-5 Summary of Terms Employed in a Bode Diagram**

1. A constant term  $K$ . When this term is present, the log magnitude plot is shifted up or down by the amount  $20 \log_{10} K$ .
2. The term  $j\omega_w$  or  $1/j\omega_w$ . If the term  $j\omega_w$  is present, the log magnitude is  $20 \log_{10} \omega_w$ , which is a straight line with a slope of  $+20$  dB/decade, and the phase is constant at  $+90^\circ$ . If the term  $1/j\omega_w$  is present, the log magnitude is  $-20 \log_{10} \omega_w$ , which is a straight line with a slope of  $-20$  dB/decade, and the phase is constant at  $-90^\circ$ . The Bode plots for these terms are shown in Fig. 7-20.
3. The term  $(1 + j\omega_w\tau)$  or  $[1/(1 + j\omega_w\tau)]$ . The term  $(1 + j\omega_w\tau)$  has a log magnitude of  $20 \log_{10} \sqrt{1 + \omega_w^2\tau^2}$  which can be approximated as  $20 \log_{10} 1 = 0$  when  $\omega_w\tau \ll 1$  and as  $20 \log_{10} \omega_w\tau$ , when  $\omega_w\tau \gg 1$ . The corner or “break” frequency is  $\omega_w = 1/\tau$ . The phase is given by the expression  $\tan^{-1} \omega_w\tau$ . The term  $[1/(1 + j\omega_w\tau)]$  is handled in a similar manner. The Bode plots for these functions are shown in Fig. 7-20.



**FIGURE 7-20** Short summary of terms employed in Bode diagrams: (a) Bode plot for  $j\omega_w$ ; (b) Bode plot for  $1/j\omega_w$ ; (c) Bode plot for  $1 + j\omega_w\tau$ ; (d) Bode plot for  $1/(1 + j\omega_w\tau)$ .



**FIGURE 7-21** Bode diagram for Example 7.12.

The gain margin for this system is 7.58 dB at 1.32 rad/s; phase margin, 30.4° at 0.772 rad/s.

**GAIN-PHASE PLOTS** The gain-phase plot of the frequency response of a system is a plot of the same information shown by a Bode diagram, plotted on different axes. For the gain-phase plot, the frequency response is plotted as gain versus phase on rectangular axes, with frequency as a parameter. Thus the frequency response of a discrete-time system may be plotted on gain-phase axes, as well as on a Bode diagram. As an example, the gain-phase plot of the system of Example 7.12 is shown in Fig. 7-22. This plot is called a Nichols chart and is covered in more detail later in this chapter.

## 7.9 INTERPRETATION OF THE FREQUENCY RESPONSE

Throughout this chapter the term *frequency response* has been used in relation to discrete-time systems. The physical meaning of a system frequency response in relation to continuous-time systems is well known. In this section the physical meaning of the frequency response of a discrete-time system is developed.

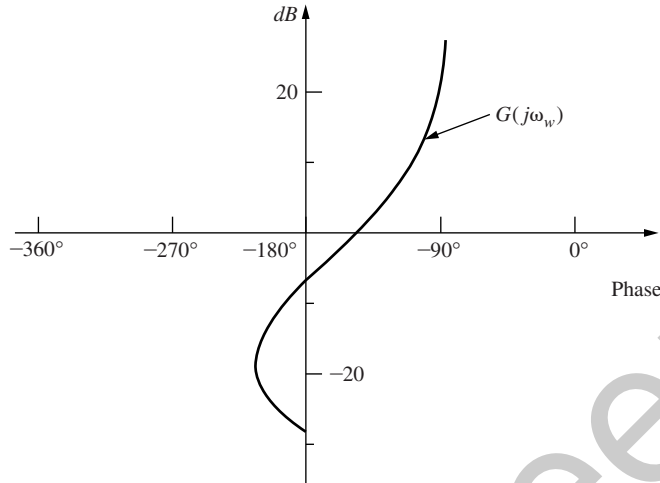


FIGURE 7-22 Gain–phase plot for Example 7.12.

Consider a discrete-time system described by

$$C(z) = G(z)E(z)$$

and suppose that the system input is a sampled sine wave; that is,

$$E(z) = \mathcal{Z}[\sin \omega t] = \frac{z \sin \omega T}{(z - \epsilon^{j\omega T})(z - \epsilon^{-j\omega T})}$$

Then

$$C(z) = \frac{G(z)z \sin \omega T}{(z - \epsilon^{j\omega T})(z - \epsilon^{-j\omega T})} = \frac{k_1 z}{z - \epsilon^{j\omega T}} + \frac{k_2 z}{z - \epsilon^{-j\omega T}} + C_g(z) \quad (7-35)$$

where  $C_g(z)$  are those components of  $C(z)$  that originate in the poles of  $G(z)$ . If the system is stable, these components of  $c(nT)$  will tend to zero with increasing time, and the system steady-state response is given by

$$C_{ss}(z) = \frac{k_1 z}{z - \epsilon^{j\omega T}} + \frac{k_2 z}{z - \epsilon^{-j\omega T}} \quad (7-36)$$

Now, from (7-35),

$$k_1 = \frac{G(\epsilon^{j\omega T}) \sin \omega T}{\epsilon^{j\omega T} - \epsilon^{-j\omega T}} = \frac{G(\epsilon^{j\omega T})}{2j}$$

Expressing  $G(\epsilon^{j\omega T})$  as

$$G(\epsilon^{j\omega T}) = |G(\epsilon^{j\omega T})| \epsilon^{j\theta}$$

we see that (7-36) becomes

$$k_1 = \frac{|G(\epsilon^{j\omega T})| \epsilon^{j\theta}}{2j} \quad (7-37)$$

Since  $k_2$  is the complex conjugate of  $k_1$ , then

$$k_2 = \frac{|G(\epsilon^{j\omega T})| \epsilon^{-j\theta}}{2(-j)} = \frac{-|G(\epsilon^{j\omega T})| \epsilon^{-j\theta}}{2j} \quad (7-38)$$

Thus, from (7-36), (7-37), and (7-38),

$$\begin{aligned} c_{ss}(kT) &= k_1(\epsilon^{j\omega T})^k + k_2(\epsilon^{-j\omega T})^k \\ &= |G(\epsilon^{j\omega T})| \frac{\epsilon^{j(\omega kT + \theta)} - \epsilon^{-j(\omega kT + \theta)}}{2j} \\ &= |G(\epsilon^{j\omega T})| \sin(\omega kT + \theta) \end{aligned} \quad (7-39)$$

From the development above it is seen that if the input to a stable discrete-time system is a sinusoid of frequency  $\omega$ , the steady-state system response is also sinusoidal at the same frequency. The amplitude of the response is equal to the amplitude of the input multiplied by  $|G(\epsilon^{j\omega T})|$ , and the phase of the response is equal to the phase of the input plus the angle of  $G(\epsilon^{j\omega T})$ . Thus it is seen that  $G(\epsilon^{j\omega T})$  is the true system frequency response at the sampling instants.

## 7.10 CLOSED-LOOP FREQUENCY RESPONSE

In this chapter many of the techniques presented for closed-loop stability analysis are based on the system open-loop frequency response. Of course, the system transfer characteristics are determined by the closed-loop frequency response. In this section the closed-loop frequency response of a control system is related graphically to its open-loop frequency response. The resultant graph is important conceptually, especially in the design of control systems; this graph is useful in illustrating the meaning of gain and phase margins as relative-stability parameters. However, if we want to calculate the closed-loop frequency response, we do not use graphical techniques; instead, we use computer software.

Consider the discrete-time system shown in Fig. 7-23(a). For this system,

$$\frac{C(z)}{R(z)} = \frac{G(z)}{1 + G(z)}$$

The closed-loop frequency response is given by

$$\frac{C(\epsilon^{j\omega T})}{R(\epsilon^{j\omega T})} = \frac{G(\epsilon^{j\omega T})}{1 + G(\epsilon^{j\omega T})} \quad (7-40)$$

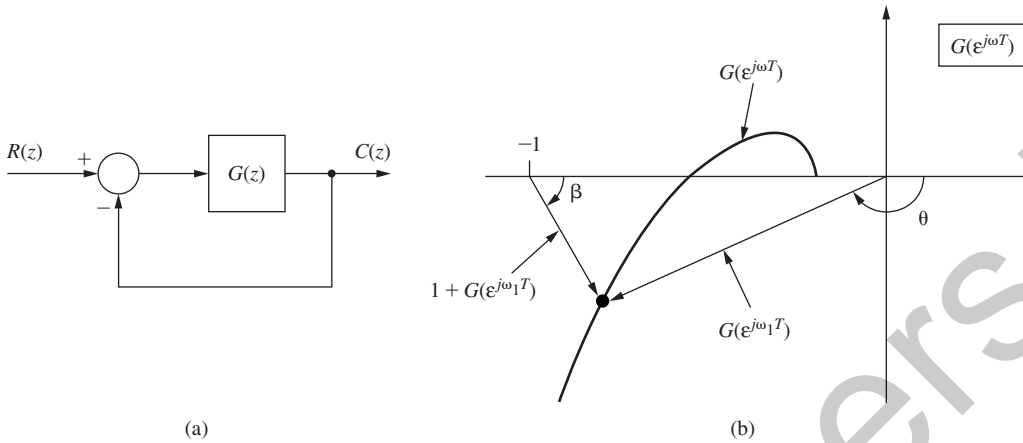


FIGURE 7-23 Determining closed-loop frequency response from open-loop frequency response.

Suppose that  $G(e^{j\omega T})$  is as shown in the polar plot in Fig. 7-23(b). Then the numerator and the denominator of (7-40) are the vectors shown for the frequency  $\omega = \omega_1$ , and the frequency response at  $\omega_1$  is the ratio of these vectors. Let the ratio of these vectors be denoted as

$$\frac{C(e^{j\omega_1 T})}{R(e^{j\omega_1 T})} = \frac{|G(e^{j\omega_1 T})| e^{j\theta}}{|1 + G(e^{j\omega_1 T})| e^{j\beta}} = M e^{j(\theta - \beta)} = M e^{j\phi} \quad (7-41)$$

**CONSTANT  $M$  CIRCLES.** The locus of points in the  $G(e^{j\omega T})$  plane for which the magnitude of the closed-loop frequency response,  $M$ , is a constant is called a constant magnitude locus, or a *constant  $M$  circle*. To see that these loci are in fact circles, consider the following development. Let

$$G(e^{j\omega T}) = X + jY \quad (7-42)$$

Then, from (7-40) and (7-41),

$$M^2 = \frac{X^2 + Y^2}{(1 + X)^2 + Y^2}$$

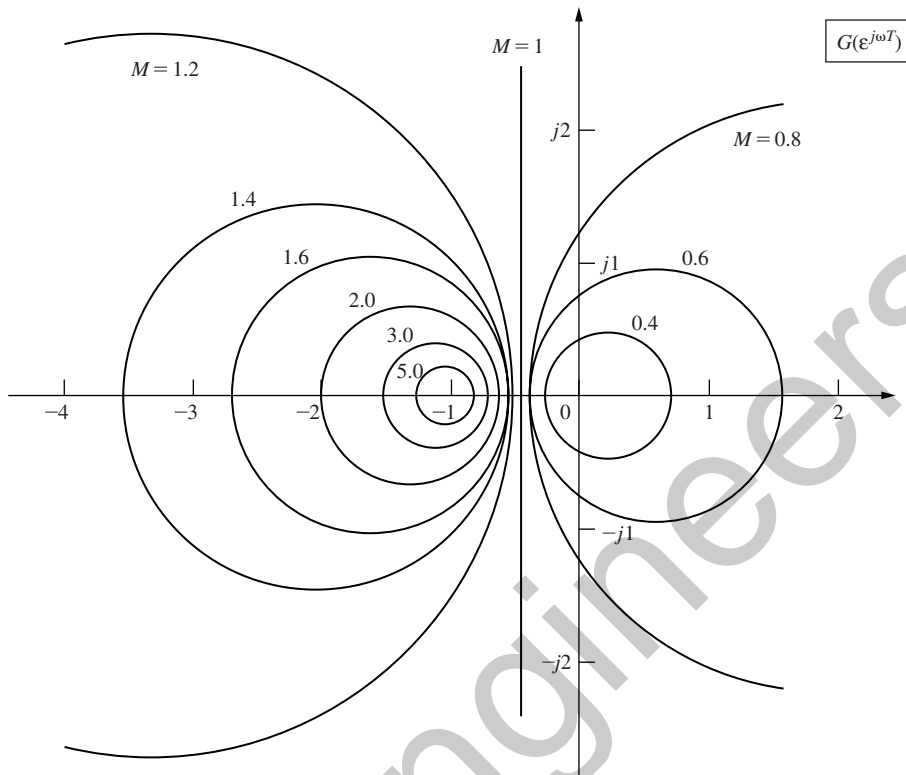
Hence

$$X^2(1 - M^2) - 2M^2X - M^2 + (1 - M^2)Y^2 = 0 \quad (7-43)$$

For  $M \neq 1$ , we can express this relationship as

$$\left[ X + \frac{M^2}{M^2 - 1} \right]^2 + Y^2 = \frac{M^2}{(M^2 - 1)^2} \quad (7-44)$$

This relationship is the equation of a circle of radius  $|M/(M^2 - 1)|$  with center at  $X = -M^2/(M^2 - 1)$  and  $Y = 0$ . For  $M = 1$ , (7-43) yields  $X = -\frac{1}{2}$ , which is a straight line. Fig. 7-24 illustrates the constant  $M$  circles.



**FIGURE 7-24** Constant magnitude circles.

**CONSTANT  $N$  CIRCLES.** The loci of points of constant phase are also circles. It is seen from (7-41) and (7-42) that

$$\phi = \theta - \beta = \tan^{-1}\left(\frac{Y}{X}\right) - \tan^{-1}\left(\frac{Y}{1+X}\right)$$

Then, letting  $N = \tan \phi$ ,

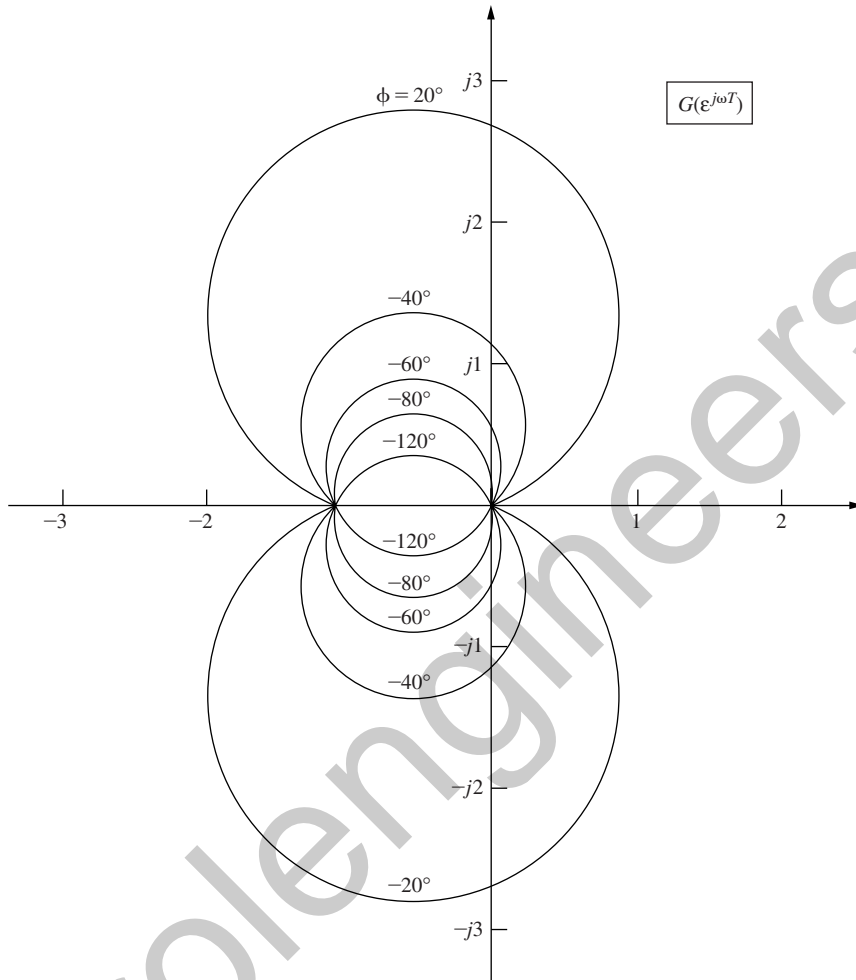
$$N = \tan(\theta - \beta) = \frac{\tan \theta - \tan \beta}{1 + \tan \theta \tan \beta} = \frac{\frac{Y}{X} - \frac{Y}{1+X}}{1 + \frac{Y}{X} \left[ \frac{Y}{1+X} \right]}$$

Hence

$$X^2 + X + Y^2 - \frac{1}{N}Y = 0$$

This equation can then be expressed as

$$\left(X + \frac{1}{2}\right)^2 + \left(Y - \frac{1}{2N}\right)^2 = \frac{1}{4} + \left(\frac{1}{2N}\right)^2 \quad (7-45)$$



**FIGURE 7-25** Constant phase circles.

This is the equation of a circle with radius of  $\sqrt{1/4 + (1/2N)^2}$  with center at  $X = -1/2$  and  $Y = 1/(2N)$ . Fig. 7-25 illustrates the *constant N (phase) circles*. Note that accuracy is very limited in the region of the  $-1$  point for Figs. 7-24 and 7-25. For this reason the Nichols chart is often used, instead of the constant  $M$  and  $N$  circles. The Nichols chart will now be discussed.

**NICHOLS CHART.** The gain–phase plane was introduced in Section 7.8 and was shown to be useful in graphically presenting a system frequency response. For example, the open-loop frequency response  $G(e^{j\omega T})$  presented as a polar plot in Fig. 7-23 could also be plotted in the gain–phase plane, as shown in Fig. 7-22. Also, the closed-loop constant  $M$  circles of Fig. 7-24 and the closed-loop constant  $N$  circles of Fig. 7-25 can be plotted in the gain–phase plane. Such a plot is called the *Nichols chart* [6], and is often used in relating the open-loop frequency response and the closed-loop frequency response. Fig. 7-26 illustrates the Nichols chart.

The frequency response  $G(e^{j\omega T})$  is usually computed via computer programs. Thus it is logical to add statements to these programs that will, at the same time, calculate the closed-loop frequency response, rather than using a graphical procedure as described above. The computer procedure is more accurate, saves time, and is not limited to the system of Fig. 7-23(a). However,



7.10 Closed-Loop Frequency Response

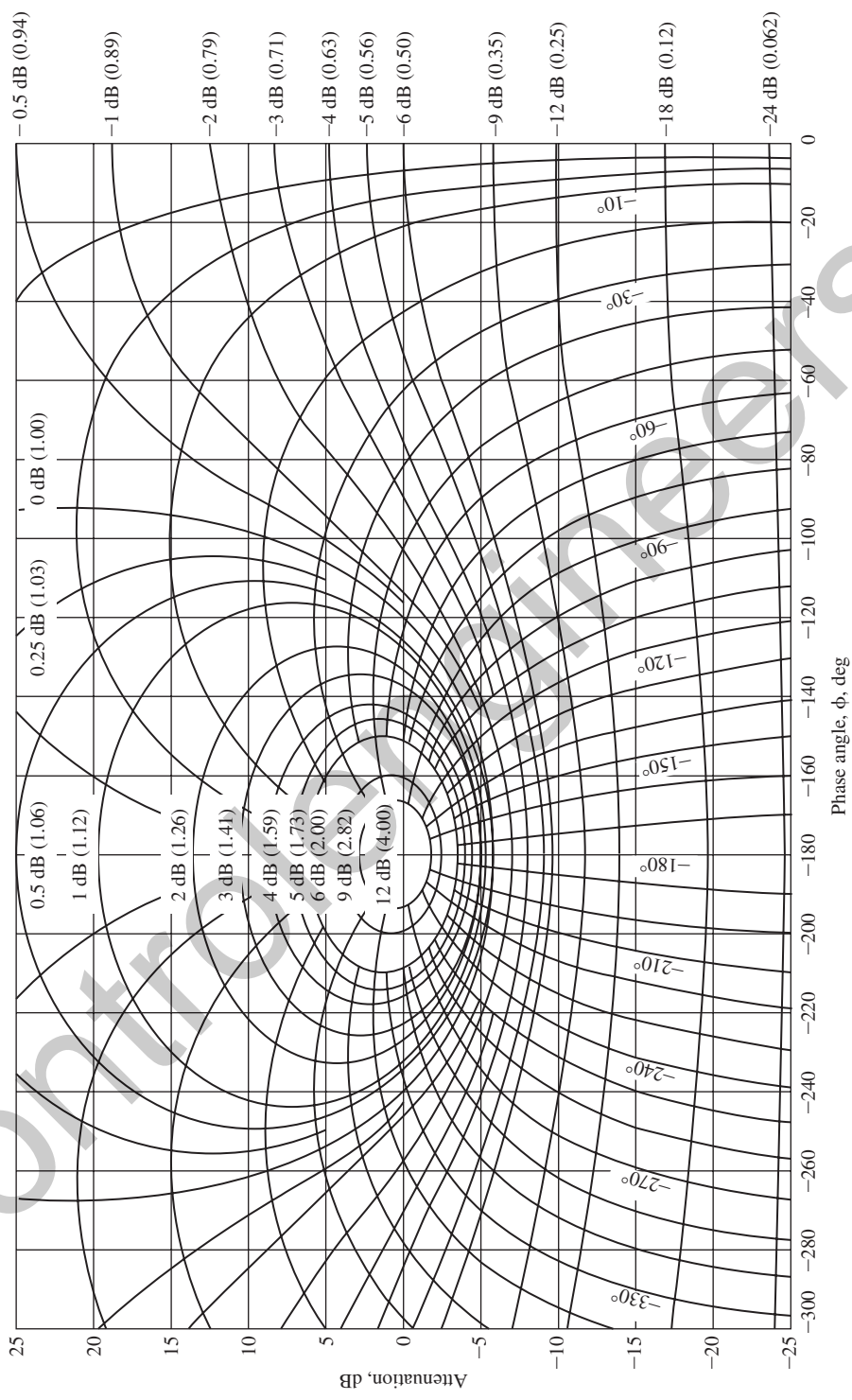


FIGURE 7-26 Nichols chart.

as stated above, the constant  $M$  and  $N$  circles and the Nichols chart are very useful in understanding the effects on the closed-loop frequency response of varying the open-loop frequency response. An example will now be given that illustrates closed-loop frequency response.

**EXAMPLE 7.13**

Again the system of Example 6.4 will be considered.

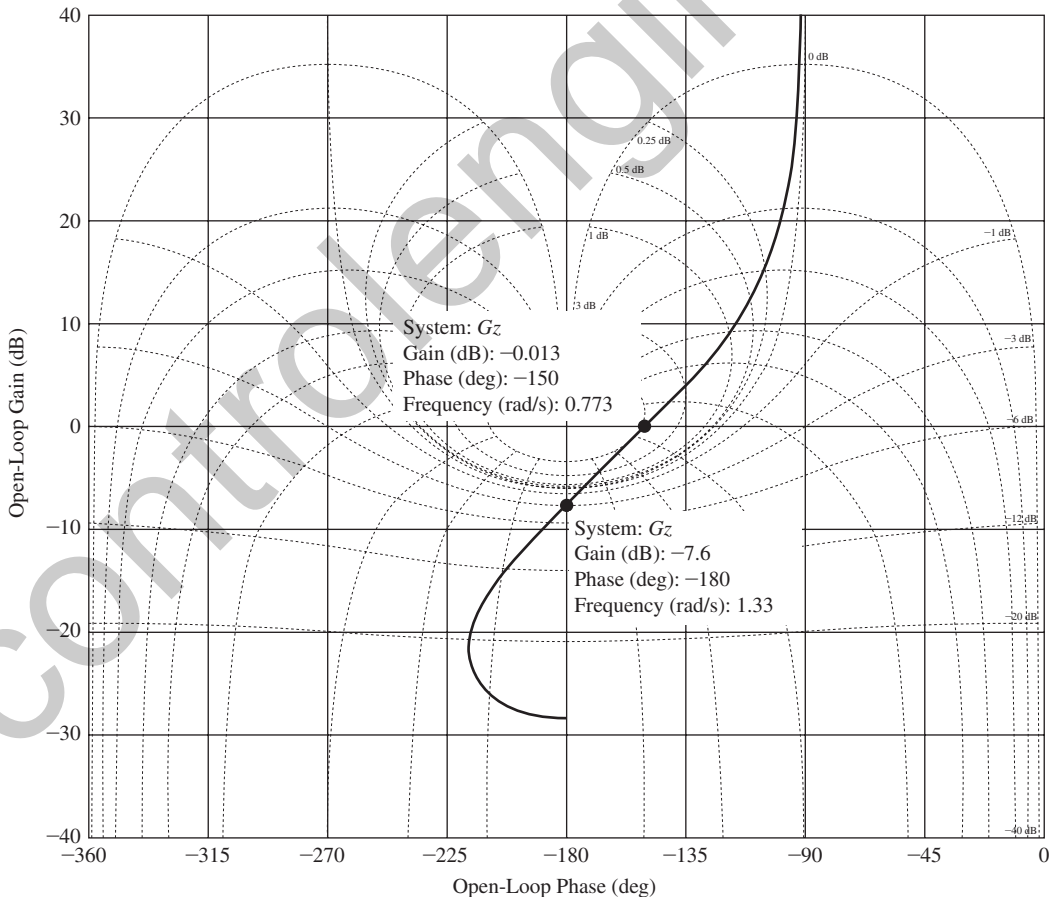
$$G(s) = \frac{1}{s(s + 1)}$$

First, let's compute the Nichols chart for this system using MATLAB.

```
>> num = [1]; den = [1 1 0]; Gp = tf(num,den); T = 1;
Gz=c2d(Gp,T); nichols(Gz), grid
```

The resulting Nichols chart is plotted in Fig. 7-27. Note that the gain margin (7.6 dB) and phase margin (30°) are labeled on the plot.

For this example, next we compare the closed-loop responses for  $T = 1$  s and  $T = 0.1$  s. The following MATLAB program computes both the closed-loop frequency and step responses.

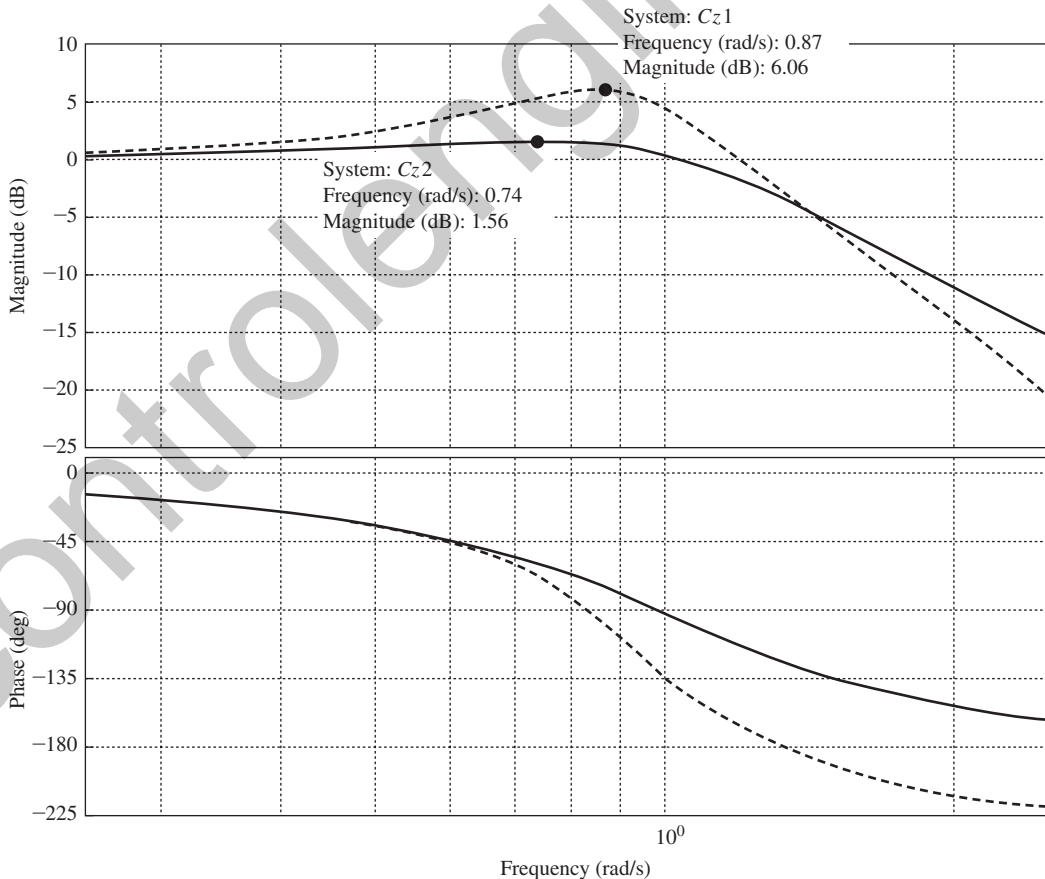


**FIGURE 7-27** Nichols chart for Example 7.13.

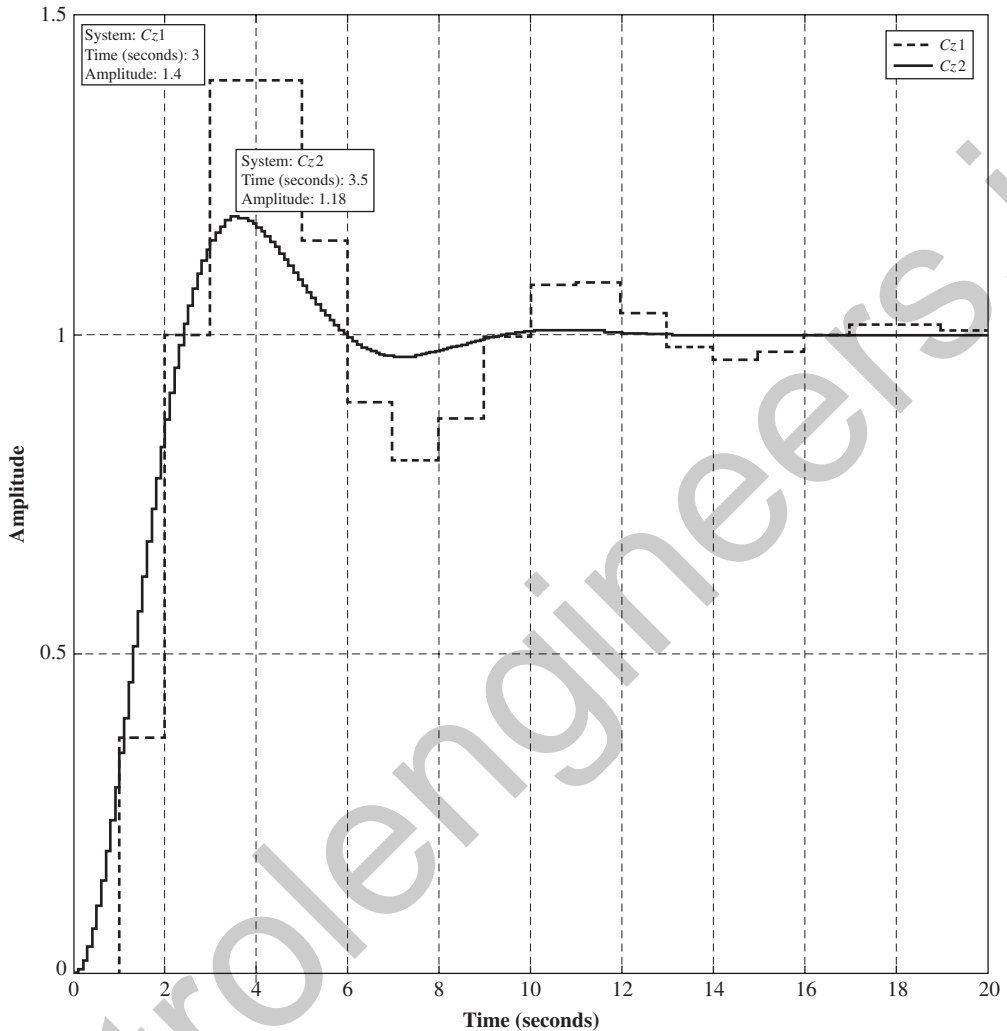
```
>> num = [1]; den = [1 1 0]; Gp = tf(num,den);
T1 = 1; Gz1=c2d(Gp,T1); [Gm1,Pm1] = margin(Gz1);
Cz1 = feedback(Gz1,1);
T2 = 0.1; Gz2 = c2d(Gp,T2); [Gm2,Pm2] = margin(Gz2);
Cz2 = feedback(Gz2,1);
%Plot the closed-loop frequency response
figure(1), bode(Cz1,'--k',Cz2,'-k'), axis([0.25 2.5 -225 10]), grid
%Plot the step response
figure(2), step(Cz1,'--k',Cz2,'-k'), axis([0 20 0 1.5]), grid
%List the step response and margin parameters
StepT1=stepinfo(Cz1), Pm1,Gm1=20*log10(Gm1)%Convert to dB
StepT2=stepinfo(Cz2), Pm2,Gm2=20*log10(Gm2)%Convert to dB
```

The closed-loop frequency response calculated above is plotted in Fig. 7-28 and the step response in Fig. 7-29. Note that the frequency response for the  $T = 1$  s case has a peak of 6.06 dB at 0.87 rad/s; for  $T = 0.1$  s, the peak is 1.56 dB at 0.74 rad/s. A more pronounced resonance in the closed-loop frequency response generally indicates more overshoot in the step response, as is readily apparent for these cases in Fig. 7-28. The peak overshoot for  $T = 1$  s is 40 percent, and that for  $T = 0.1$  s is 18 percent.

A resonance in a closed-loop frequency response can also be correlated with the system stability margins. For the system of Fig. 7-23(a), the denominator of the closed-loop transfer function



**FIGURE 7-28** Closed-loop frequency response magnitudes for Example 7.13.

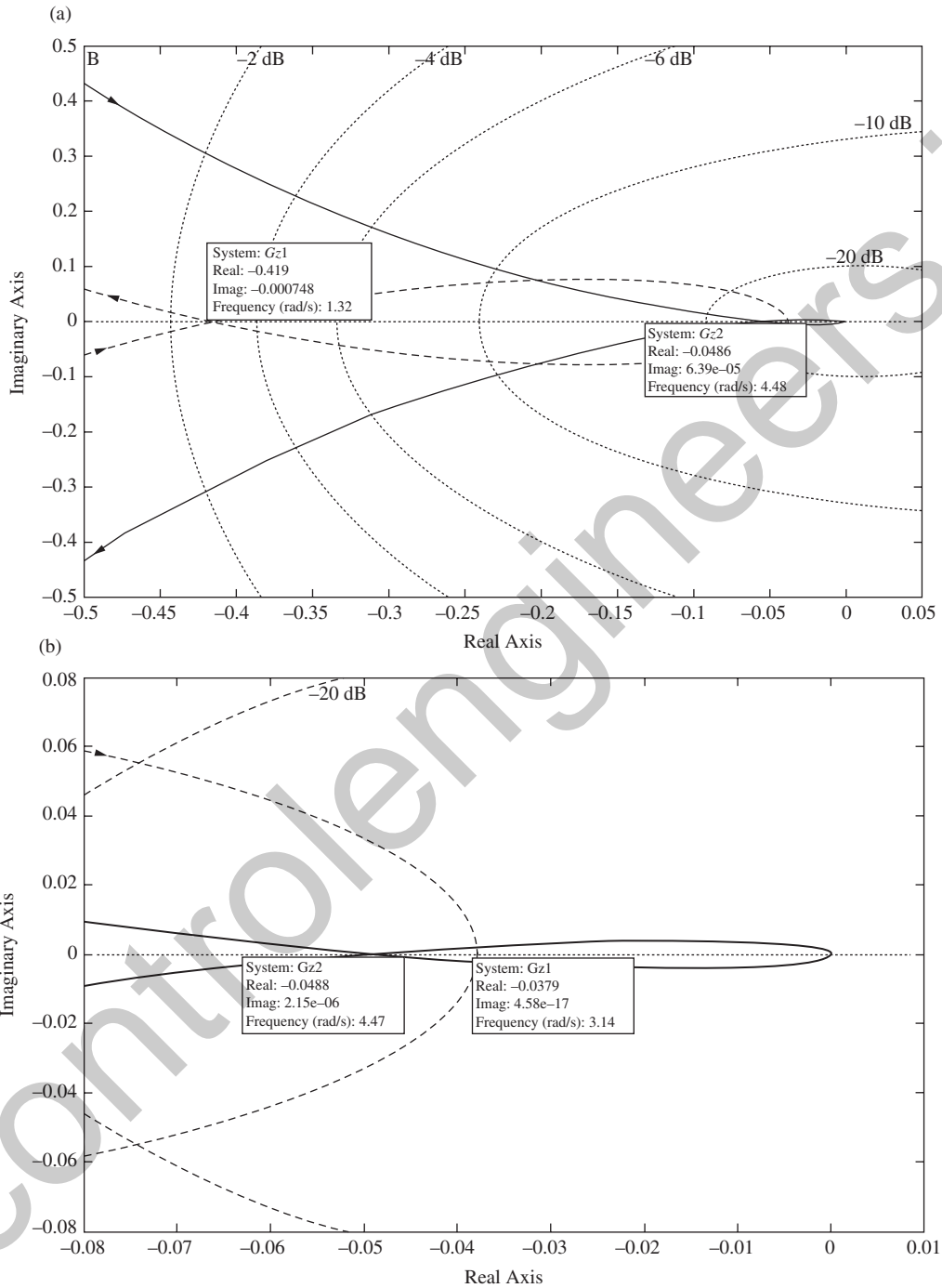


**FIGURE 7-29** Step responses for Example 7.13.

is the distance from the  $-1$  point to the Nyquist diagram, as shown in Fig. 7-23(b). If the stability margins are small, the Nyquist diagram passes close to the  $-1$  point, and the denominator of the closed-loop transfer function has a distinct minimum producing a closed-loop system resonance. For the systems of Example 7.13, the following MATLAB program generates the Nyquist diagrams:

```
>> num = [1]; den = [1 1 0]; Gs = tf(num,den);
T1 = 1; Gz1 = c2d(Gs,T1);
T2 = 0.1; Gz2 = c2d(Gs,T2);
figure(1),nyquist(Gz1,Gz2), axis([-0.5 0.05 -0.5 0.5]), grid
figure(2),nyquist(Gz1,Gz2), axis([-0.08 0.01 -0.08 0.08]), grid
```

From these Nyquist diagrams, for  $T = 0.1$  s, negative axis crossover point is  $-0.0488$  so the gain margin is  $20 \log_{10}(1/0.0488) = 26.2$  dB. For  $T = 1$  s, negative axis crossover point is  $-0.418$  making the gain  $20 \log_{10}(1/0.418) = 7.58$  dB. These results match those found earlier in this chapter. Please note that the correlation between the closed-loop frequency response and the time response of a system is investigated in greater detail in Chapter 8.



**FIGURE 7-30** Nyquist diagrams for Example 7.13. (a) Larger scale to examine the  $T = 1$  s case; (b) smaller scale to examine the  $T = 0.1$  s case.

In summary, we see from both the constant  $M$  circles of Fig. 7-24 and the Nichols chart of Fig. 7-26 that a peaking in the closed-loop frequency response can occur only if the open-loop frequency response passes near the  $-1$  point. Hence any control system that has a significant resonance also has small stability margins. Conversely, any system that has small stability margins will exhibit a significant resonance in its time response.

## 7.11 SUMMARY

In this chapter a number of techniques for analyzing the stability of discrete-time systems have been presented. It has been shown that many of the methods used in the analysis of continuous-time systems are applicable to sampled-data systems also. The chapter contains a number of examples, and the use of the same system in many of the examples throughout the chapter provides a common thread and basis for comparison among the various stability analysis techniques. Many of the analysis techniques presented in this chapter will be extended to design in Chapter 8.

---

## References and Further Readings

- |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>[1] C. L. Phillips and J. Parr, <i>Feedback Control Systems</i>, 5th ed. Upper Saddle River, NJ: Prentice-Hall, 2011.</p> <p>[2] E. I. Jury, <i>Theory and Application of the z-Transform Method</i>. Huntington, NY: R.E. Krieger Publishing Co., Inc., 1973.</p> <p>[3] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, <i>Digital Control of Dynamic Systems</i>, 6th ed. Upper Saddle River, NJ: Prentice-Hall, 2011.</p> <p>[4] R. C. Dorf and R. H. Bishop, <i>Modern Control Systems</i>, 12th ed. Upper Saddle River, NJ: Prentice-Hall, 2010.</p> | <p>[5] C. R. Wylie and L.C. Barrett, <i>Advanced Engineering Mathematics</i>, 6th ed. New York: McGraw-Hill Book Company, 1995.</p> <p>[6] H. M. James, N. B. Nichols, and R. S. Phillips, <i>Theory of Servomechanisms</i>. New York: McGraw-Hill Book Company, 1947.</p> <p>[7] J. A. Cadzow and H. R. Martens, <i>Discrete-Time and Computer Control Systems</i>. Upper Saddle River, NJ: Prentice-Hall, 1970.</p> <p>[8] W. R. Evans, <i>Control System Dynamics</i>. New York: McGraw-Hill Book Company, 1954.</p> <p>[9] G. F. Franklin, J. D. Powell, and M. Workman, <i>Digital Control of Dynamic Systems</i>, 3d ed. Half Moon Bay, CA: Ellis-Kagle Press, 2006.</p> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

## Problems

- 7.2-1.** Assume that for the system of Fig. 7-1, the system closed-loop transfer-function pole  $p_1$  is repeated such that the system characteristic equation is given by

$$(z - p_1)^r(z - p_{r+1})(z - p_{r+2}) \cdots (z - p_n) = 0$$

where  $r$  is an integer. Show that the requirement for system stability is that the magnitudes of all poles of the closed-loop transfer function are less than unity, that is,  $|p_i| < 1, i = 1, r + 1, r + 2, \dots, n$ .

- 7.2-2.** The system of Example 7.1 and Fig. 7-3 has two samplers. The system characteristic equation is derived in Example 7.1 as

$$1 + G_1(z)G_2(z) + \overline{G_2H}(z) = 0$$

Show that the same characteristic equation is obtained by opening the system at the second sampler.

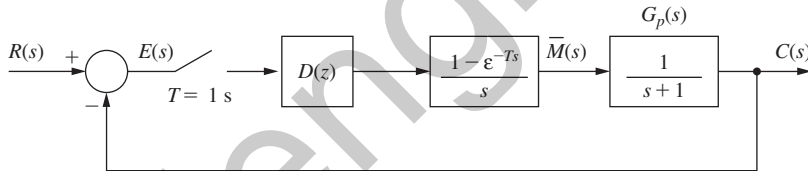
- 7.2-3.** (a) The unit-step response of a discrete system is the system response  $c(k)$  with the input  $r(k) = 1$  for  $k \geq 0$ . Show that if the discrete system is stable, the unit-step response,  $c(k)$ , approaches a constant as

- $k \rightarrow \infty$ . [Let  $T(z)$  be the closed-loop system transfer function. Assume that the poles of  $T(z)$  are distinct (no repeated poles).]
- Find the conditions on the closed-loop system transfer function  $T(z)$  such that the unit-step response approaches zero as  $k \rightarrow \infty$ .
  - The discrete unit-impulse response of a discrete system is the system response  $c(k)$  with the input  $r(k) = 1$  for  $k = 0$  and  $r(k) = 0$  for  $k \geq 1$ . Show that if the discrete system is stable, the unit impulse response,  $c(k)$ , approaches zero. [Let  $T(z)$  be the closed-loop system transfer function. Assume that the poles of  $T(z)$  are distinct (no repeated poles).]

**7.2-4.** Consider a sampled-data system with  $T = 0.5$  s and the characteristic equation given by

$$(z - 0.9)(z - 0.8)(z^2 - 1.9z + 1.0) = 0$$

- Find the terms in the system natural response.
  - A discrete LTI system is stable, unstable, or marginally stable. Identify the type of stability for this system.
  - The natural response of this system contains an undamped sinusoidal response term of the form  $A \cos(\omega kT + \theta)$ . Find the frequency  $\omega$  of this term.
- 7.2-5.** Consider the system of Fig. P7.2-5 with  $T = 1$  s. Let the digital controller be a variable gain  $K$  such that  $D(z) = K$ . Hence  $m(kT) = Ke(kT)$ .
- Write the closed-loop system characteristic equation.
  - Determine the range of  $K$  for which the system is stable.
  - Suppose that  $K$  is set to the lower limit of the range in part (b) such that the system is marginally stable. Find the natural-response term that illustrates the marginal stability.
  - Repeat part (c) for the upper limit of the range of  $K$ .



**FIGURE P7.2-5** System for Problem 7.2-5.

- 7.2-6.** Consider the system of Fig. P7.2-5, and let the digital controller be a variable gain  $K$  such that  $D(z) = K$ . Hence  $m(kT) = Ke(kT)$ .
- Write the closed-loop system characteristic equation as a function of the sample period  $T$ .
  - Determine the ranges of  $K > 0$  for stability for the sample periods  $T = 1$  s,  $T = 0.1$  s, and  $T = 0.01$  s.
  - Consider the system with all sampling removed and with  $G_p(s) = K/(s + 1)$ . Find the range of  $K > 0$  for which the analog system is stable.
  - Comparing the ranges of  $K$  from parts (b) and (c), give the effects on stability of reducing the sample period  $T$ .

**7.3-1.** Consider the general bilinear transformation

$$z = \frac{1 + aw}{1 - aw}$$

where  $a$  is real and nonzero.

- Show that this function transforms the stability boundary of the  $z$ -plane into the imaginary axis in the  $w$ -plane.
  - Find the relationship of frequency in the  $s$ -plane to frequency in the  $w$ -plane.
  - Find the stable region of the  $w$ -plane for  $a < 0$  and for  $a > 0$ .
- 7.5-1.** Given below are the characteristic equations of certain discrete systems.

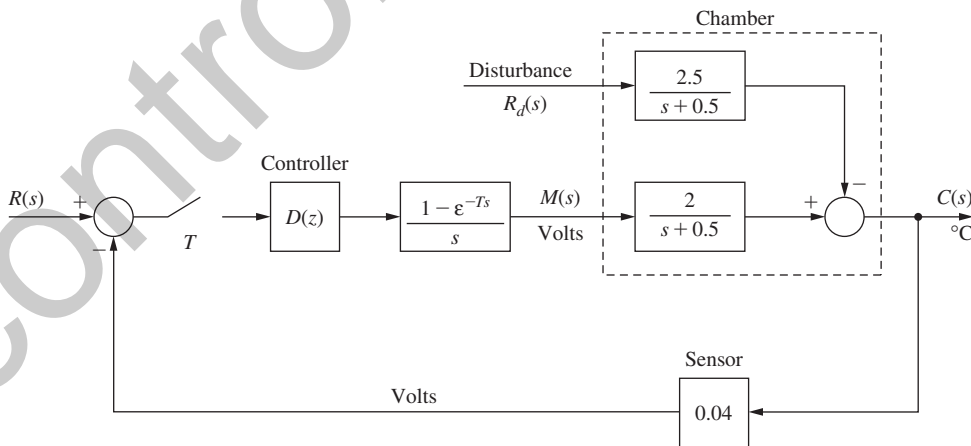
- |                              |                           |
|------------------------------|---------------------------|
| (i) $z^2 - 1.1z + 0.3 = 0$   | (ii) $z^2 - z + 0.25 = 0$ |
| (iii) $z^2 - 0.1z - 0.3 = 0$ | (iv) $z^2 - 0.25 = 0$     |

(v)  $z^2 - 1.6z + 1 = 0$                       (vi)  $z^2 - 2.0z + 0.99 = 0$   
 (vii)  $z^3 - 2.2z^2 + 1.55z - 0.35 = 0$       (viii)  $z^3 - 1.9z^2 + 1.4z - 0.45 = 0$

- (a) Use the Jury test to determine the stability of each of the systems.  
 (b) List the natural-response terms for each of the systems.  
 (c) For those systems in part (a) that are found to be either unstable or marginally stable, list the natural-response terms in part (b) that yield these results.
- 7.5-2.** Consider the system of Fig. P7.2-5 with  $T = 1$  s. Let the digital controller be a variable gain  $K$  such that  $D(z) = K$ . Hence  $m(kT) = Ke(kT)$ .
- (a) Write the closed-loop system characteristic equation.  
 (b) Use the Routh–Hurwitz criterion to determine the range of  $K$  for stability.  
 (c) Check the results of part (b) using the Jury test.  
 (d) Determine the location of all roots of the characteristic equation in both the  $w$ -plane and the  $z$ -plane for the value of  $K > 0$  for which the system is marginally stable.  
 (e) Determine the  $s$ -plane frequency at which the system will oscillate when marginally stable, for  $K > 0$ .  
 (f) Consider the system with all sampling removed and with  $G_p(s) = K/(s + 1)$ . Find the range of  $K$  for which the analog system is stable.  
 (g) Comparing the ranges of  $K$  from parts (b) and (f), give the effects on stability of adding sampling to the analog system.
- 7.5-3.** Consider the temperature control system of Fig. P7.5-3. This system is described in Problem 1.6-1. For this problem, ignore the disturbance input, let  $T = 0.6$  s, and let the digital controller be a variable gain  $K$  such that  $D(z) = K$ . Hence  $m(kT) = Ke(kT)$ , where  $e(t)$  is the input to the sampler. It was shown in Problem 6.2-4 that

$$\mathfrak{Z} \left[ \frac{1 - e^{-Ts}}{s} \frac{2}{s + 0.5} \right] = \frac{1.037}{z - 0.7408}$$

- (a) Write the closed-loop system characteristic equation.  
 (b) Use the Routh–Hurwitz criterion to determine the range of  $K$  for stability.  
 (c) Check the results of part (b) using the Jury test.  
 (d) Let  $T = 0.06$  s. Find the range of  $K$  for which the system is stable.



**FIGURE P7.5-3** Chamber temperature control system.

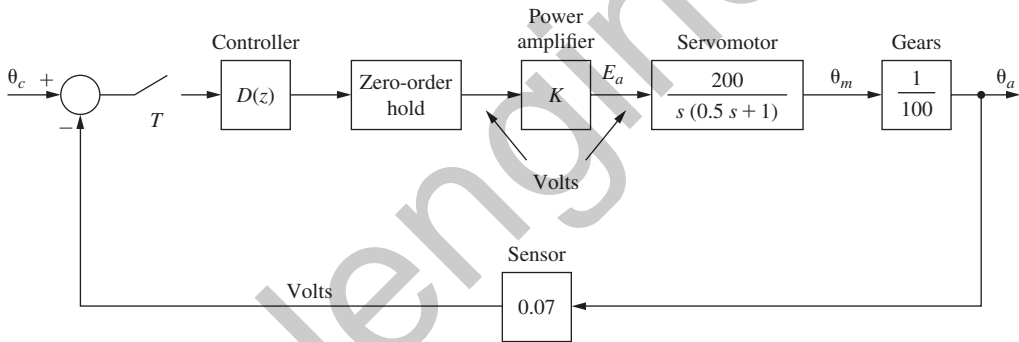


(e) Comparing the ranges of  $K$  from parts (b) and (d), give the effects on stability of adding sampling to the analog system.

**7.5-4.** Consider the robot arm joint control system of Fig. P7.5-4. This system is described in Problem 1.5-4. For this problem,  $T = 0.1$  s and  $D(z) = 1$ . It was shown in Problem 6-7 that

$$\mathfrak{Z} \left[ \frac{1 - \epsilon^{-Ts}}{s} \frac{4}{s(s+2)} \right] = \frac{0.01873z + 0.01752}{(z-1)(z-0.8187)}$$

- Write the closed-loop system characteristic equation.
- Use the Routh–Hurwitz criterion to determine the range of  $K$  for stability.
- Check the results of part (b) using the Jury test.
- Determine the location of all roots of the characteristic equation in both the  $w$ -plane and the  $z$ -plane for the value of  $K > 0$  for which the system is marginally stable.
- Determine both the  $s$ -plane frequency and the  $w$ -plane frequency at which the system will oscillate when marginally stable, using the results of part (d).
- Show that the frequencies in part (e) satisfy (7-10).

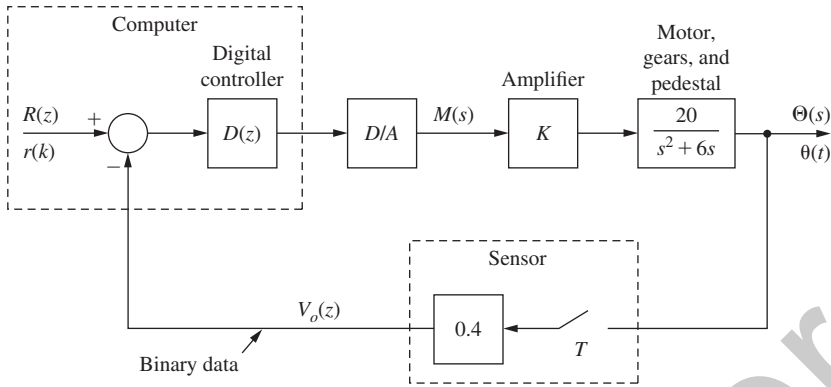


**FIGURE P7.5-4** Robot arm joint control system.

**7.5-5.** Consider the antenna control system of Fig. P7.5-5. This system is described in Problem 1.5-1. For this problem,  $T = 0.05$  s and  $D(z) = 1$ . It was shown in Problem 5.3-15 that

$$\mathfrak{Z} \left[ \frac{1 - \epsilon^{-Ts}}{s} \frac{20}{s(s+6)} \right] = \frac{0.02268z + 0.02052}{(z-1)(z-0.7408)}$$

- Write the closed-loop system characteristic equation.
- Use the Routh–Hurwitz criterion to determine the range of  $K$  for stability.
- Check the results of part (b) using the Jury test.
- Determine the location of all roots of the characteristic equation in both the  $w$ -plane and the  $z$ -plane for the value of  $K > 0$  for which the system is marginally stable.
- Determine both the  $s$ -plane frequency and the  $w$ -plane frequency at which the system will oscillate when marginally stable, using the results of part (d).
- Show that the frequencies in part (e) satisfy (7-10).

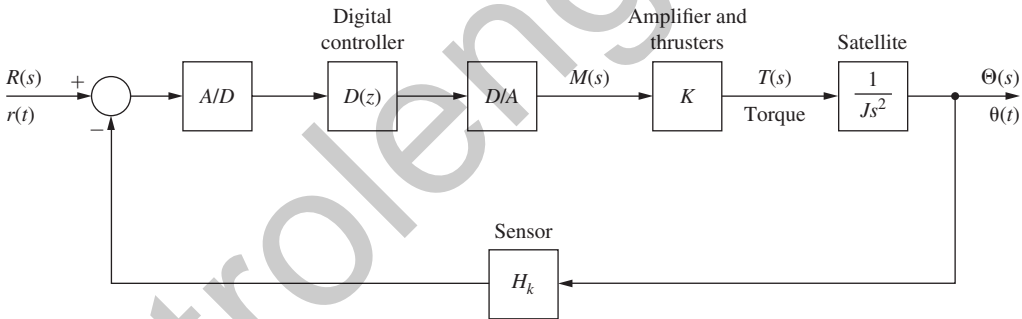


**FIGURE P7.5-5** Block diagram for an antenna control system.

**7.5-6.** Consider the satellite control system of Fig. P7.5-6. This system is described in Problem 1.4-1. For this problem,  $T = 0.1$  s,  $J = 0.1$ ,  $H_k = 0.02$ , and  $D(z) = 1$ . From the  $z$ -transform table,

$$\mathfrak{Z} \left[ \frac{1 - e^{-Ts}}{s} \frac{10}{s^2} \right] = \frac{0.05(z + 1)}{(z - 1)^2}$$

- (a) Write the closed-loop system characteristic equation.
- (b) Use the Routh–Hurwitz criterion to show that the system is unstable for all  $K$ .
- (c) Check the results of part (b) using the Jury test.

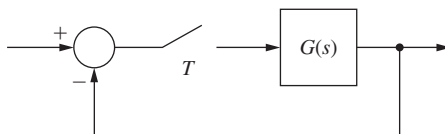


**FIGURE P7.5-6** Block diagram for a satellite control system.

**7.5-7.** For the system of Fig. P7.5-7,  $T = 2$  s and

$$G(z) = \frac{K(z + 0.8)}{(z - 1)(z - 0.6)}$$

- (a) Determine the range of  $K$  for stability using the Routh–Hurwitz criterion.
- (b) Determine the range of  $K$  for stability using the Jury test.
- (c) Show that the upper limit of  $K$  for stability in part (a) yields a marginally stable system.
- (d) Show that the upper limit of  $K$  for stability in part (b) yields a marginally stable system.



**FIGURE P7.5-7** System for Problem 7.5-7.

- 7.6-1.** For the system of Problem 7.2-5 and Fig. P7.2-5:
- Plot the  $z$ -plane root locus.
  - Plot the  $w$ -plane root locus.
  - Determine the range of  $K$  for stability using the results of part (a).
  - Determine the range of  $K$  for stability using the results of part (b).
- 7.6-2.** For the chamber temperature control system of Problem 7.5-3 and Fig. P7.5-3:
- Plot the  $z$ -plane root locus.
  - Plot the  $w$ -plane root locus.
  - Determine the range of  $K$  for stability using the results of part (a).
  - Determine the range of  $K$  for stability using the results of part (b).
- 7.6-3.** For the robot arm joint control system of Problem 7.5-4 and Fig. P7.5-4:
- Plot the  $z$ -plane root locus.
  - Plot the  $w$ -plane root locus.
  - Determine the range of  $K$  for stability using the results of part (a).
  - Determine the range of  $K$  for stability using the results of part (b).
- 7.6-4.** For the satellite control system of Problem 7.5-6 and Fig. P7.5-6:
- Plot the  $z$ -plane root locus.
  - Plot the  $w$ -plane root locus.
  - Determine the range of  $K$  for stability using the results of part (a).
  - Determine the range of  $K$  for stability using the results of part (b).
- 7.7-1.** For the robot arm joint control system of Problem 7.5-4 and Fig. P7.5-4, let  $K = 1$ .
- The frequency response for  $G(z)$  was calculated by computer and is given in Table P7.7-1. Sketch the Nyquist diagram for the open-loop function  $G(z)H_k$ , with  $H_k = 0.07 \Rightarrow -23.1$  dB.
  - Determine the stability of the system.
  - If the system is stable, determine the gain and phase margins. If the system is unstable, find the value of  $K$  that gives a phase margin of  $45^\circ$ .
  - From the Nyquist diagram, determine the value of  $K > 0$  for which the system is marginally stable.
  - Use the frequency response to find the frequency  $\omega$  at which the marginally stable system will oscillate.

**TABLE P7.7-1** Frequency Response for Problem 7.7-1

$\omega_w$	$\omega$	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.1	0.100	19.97513	26.00	-93.14
0.2	0.200	9.95054	19.95	-96.28
0.3	0.300	6.59317	16.38	-99.38
0.4	0.399	4.90325	13.80	-102.45
0.5	0.499	3.88102	11.77	-105.46
0.6	0.599	3.19331	10.08	-108.41
0.7	0.699	2.69741	8.61	-111.28
0.8	0.799	2.32198	7.31	-114.08
0.9	0.899	2.02741	6.13	-116.78
1.0	0.999	1.78990	5.05	-119.40
2.0	1.993	0.70945	-2.98	-140.61
3.0	2.977	0.37308	-8.56	-154.64
4.0	3.947	0.22743	-12.86	-164.43
5.0	4.899	0.15270	-16.32	-171.82

**TABLE P7.7-1** continued

$\omega_w$	$\omega$	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
6.0	5.829	0.10973	-19.19	-177.74
7.0	6.733	0.08291	-21.62	-182.72
8.0	7.610	0.06511	-23.72	-187.04
9.0	8.457	0.05270	-25.56	-190.88
10.0	9.273	0.04372	-27.18	-194.33
20.0	15.708	0.01403	-37.05	-217.40
30.0	19.656	0.00798	-41.96	-229.64
40.0	22.143	0.00558	-45.07	-236.77

- 7.7-2.** For the antenna control system of Problem 7.5-5 and Fig. P7.5-5, let  $K = 1$ .
- The frequency response for  $G(z)$  was calculated by computer and is given in Table P7.7-2. Sketch the Nyquist diagram for the open-loop function  $G(z)H_k$ , with  $H_k = 0.04 \Rightarrow -7.96$  dB.
  - Determine the stability of the system.
  - If the system is stable, determine the gain and phase margins. If the system is unstable, find the value of  $K$  that gives a phase margin of  $45^\circ$ .
  - From the Nyquist diagram, determine the value of  $K > 0$  for which the system is marginally stable.
  - Find the frequency  $\omega$  at which the marginally stable system will oscillate.

**TABLE P7.7-2** Frequency Response for Problem 7.7-2

$\omega_w$	$\omega$	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.1	0.100	33.32874	30.4564	-91.09
0.2	0.200	16.65748	24.4321	-92.19
0.3	0.300	11.09735	20.9042	-93.29
0.4	0.400	8.31502	18.3970	-94.38
0.5	0.500	6.64381	16.4479	-95.47
0.6	0.600	5.52819	14.8510	-96.56
0.7	0.699	4.73007	13.4965	-97.65
0.8	0.799	4.13040	12.3187	-98.73
0.9	0.899	3.66305	11.2754	-99.81
1.0	0.999	3.28834	10.3377	-100.89
2.0	1.998	1.58193	3.9759	-111.28
3.0	2.994	0.99511	-0.0619	-120.81
4.0	3.986	0.69524	-3.1949	-129.31
5.0	4.974	0.51456	-5.8345	-136.78
6.0	5.955	0.39576	-8.1482	-143.31
7.0	6.929	0.31327	-10.2199	-149.03

**TABLE P7.7-2 continued**

$\omega_w$	$\omega$	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
8.0	7.895	0.25375	-12.0995	-154.07
9.0	8.852	0.20951	-13.8203	-158.54
10.0	9.799	0.17582	-15.4069	-162.54
20.0	18.546	0.05320	-26.7969	-188.55
30.0	25.740	0.02706	-34.2817	-203.49
40.0	31.416	0.01738	-40.4880	-213.67

**7.7-3.** For the satellite control system of Problem 7.5-6, the frequency response for  $G(z)$  was calculated by computer and is given in Table P7.7-3.

- (a) Sketch the Nyquist diagram for the open-loop function  $G(z)H$ , with  $H = 0.02 \Rightarrow -34.0$  dB.  
 (b) Use the results in part (a) to determine the range of  $K$  for stability.

**TABLE P7.7-3 Frequency Response for Problem 7.7-3**

$\omega_w$	$\omega$	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.1	0.100	1000.01200	60.00	-180.28
0.2	0.200	250.01250	47.95	-180.57
0.3	0.300	111.12360	40.91	-180.85
0.4	0.399	62.51249	35.91	-181.14
0.5	0.499	40.01249	32.04	-181.43
0.6	0.599	27.79027	28.87	-181.71
0.7	0.699	20.42065	26.20	-182.00
0.8	0.799	15.63749	23.88	-182.29
0.9	0.899	12.35817	21.83	-182.57
1.0	0.999	10.01249	20.01	-182.86
2.0	1.993	2.51247	8.00	-185.71
3.0	2.977	1.12354	1.01	-188.53
4.0	3.947	0.63738	-3.91	-191.30
5.0	4.899	0.41231	-7.69	-194.03
6.0	5.829	0.29001	-10.75	-196.69
7.0	6.733	0.21622	-13.30	-199.29
8.0	7.610	0.16829	-15.47	-201.80
9.0	8.457	0.13538	-17.36	-204.22
10.0	9.273	0.11180	-19.03	-206.56
20.0	15.708	0.03536	-29.03	-225.00
30.0	19.656	0.02003	-33.96	-236.30
40.0	22.143	0.01398	-37.09	-243.43

- 7.8-1.** For the antenna control system of Problem 7.5-5 and Fig. P7.5-5:
- Plot the  $z$ -plane root locus.
  - Plot the  $w$ -plane root locus.
  - Determine the range of  $K$  for stability using the results of part (a).
  - Determine the range of  $K$  for stability using the results of part (b).
- 7.8-2.** For the system of Problem 7.2-5 and Fig. P7.2-5, let  $K = 1$ .
- Determine the stability of the system.
  - Sketch the Bode diagram, and use this diagram to sketch the Nyquist diagram.
  - If the system is stable, determine the gain and phase margins. If the system is unstable, find the value of  $K$  that gives a phase margin of  $45^\circ$ .
  - From the Nyquist diagram, determine the value of  $K > 0$  for which the system is marginally stable.
  - Find the frequency  $\omega$  at which the marginally stable system will oscillate.
- 7.8-3.** For the temperature control system of Problem 7.5-3 and Fig. P7.5-3, let  $K = 1$ .
- Determine the stability of the system.
  - Plot the Bode diagram, and the Nyquist diagram.
  - If the system is stable, determine the gain and phase margins. If the system is unstable, find the value of  $K$  that gives a phase margin of  $45^\circ$ .
  - From the Nyquist diagram, determine the value of  $K > 0$  for which the system is marginally stable.
  - Find the frequency  $\omega$  at which the marginally stable system will oscillate.
- 7.9-1.** Given the pulse transfer function  $G(z)$  of a plant. For  $\omega = 2$  rad/s,  $G(e^{j\omega T})$  is equal to the complex number  $1.3\angle(-25^\circ)$ . The signal  $5 \cos 2t$  is applied to the input (sampler and data hold) of the plant. Find the steady-state sampled output.
- 7.10-1.** For each case below, plot the Nichols chart and closed-loop frequency response. If the system is stable find the gain margin, phase margin, peak closed-loop frequency response, and system bandwidth.
- The system of Problem 7.2-5 with  $K = 1$ .
  - The temperature control system of Problem 7.5-3 with  $K = 1$ .
  - The robot arm joint control system of Problem 7.5-4 with  $K = 0.5$ .
  - The antenna control system of Problem 7.5-5 with  $K = 1$ .
  - The satellite control system of Problem 7.5-6 with  $K = 0.5$ .
  - The system of Problem 7.5-7 with  $K = 0.5$ .
- 7.10-2.** For the antenna control system of Problem 7.5-5, use MATLAB to:
- Find the range of stability for  $K$  (by *rlocus*).
  - Find the gain  $K$  for critical damping (by *rlocus*).
  - Find the gain and phase margins at critical damping (by *rlocus*).
  - Find the step response parameters rise time, settling time, and percent overshoot (by *stepinfo*).
  - Find the frequency response peak and bandwidth.
  - Graphically display the root locus, Bode plot, Nyquist diagram, Nichols chart, step response, and closed-loop frequency response.

# Digital Controller Design

## 8.1 INTRODUCTION

In the preceding chapters we have been concerned primarily with analysis. We have assumed that the control system was given, and we analyzed the system to determine stability, stability margins, time response, frequency response, and so on. Some simple design problems were considered: for example, the determination of gains required to meet steady-state error specifications.

In this chapter we consider the total design problem: How do we design a digital controller transfer function (or difference equation) that will satisfy design specifications for a given control system? We will investigate the classical design techniques of frequency response and root locus. First, phase-lag and phase-lead controllers are considered. Then a particular type of lag-lead controller, called a proportional-plus-integral-plus-derivative (PID) controller, is developed. Finally, design by root-locus procedures is introduced. Further design techniques, which are based on the state-variable model of the plant, are developed in Chapters 9 and 11.

The preceding paragraph requires additional comment. All numerical design procedures are based on an inexact model of the physical system. Hence numerical design simply gets us to the point that we can experiment with the physical system, or with an accurate simulation that includes the system nonlinearities, time-varying components, and so on. Design is generally too complex if the accurate simulation model, sometimes called a truth model in optimal filter design, is used. Thus, in one sense, any type of numerical design procedure is trial and error, with the final form and coefficients of the controller determined by several iterations of first numerical design, and then experimentation with either the physical system or an accurate simulation.

## 8.2 CONTROL SYSTEM SPECIFICATIONS

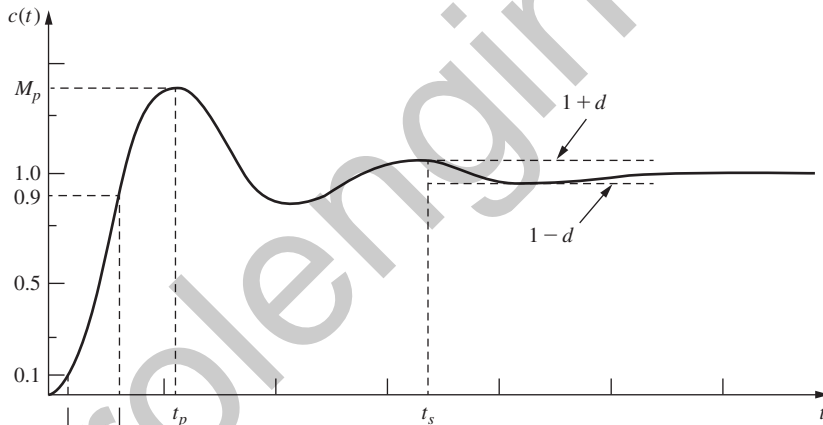
The design of a control system involves the changing of system parameters and/or the addition of subsystems (called compensators) to achieve certain desired system characteristics. The desired characteristics, or performance specifications, generally relate to steady-state accuracy, transient response, relative stability, sensitivity to change in system parameters, and disturbance rejection. These performance specifications will now be discussed [1].

### Steady-State Accuracy

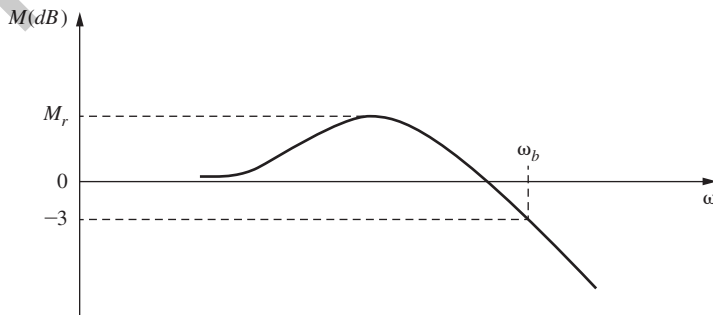
Since steady-state accuracy was discussed in detail in Section 6.5, only a brief review will be given here. In Section 6.5, it was shown that steady-state accuracy is increased if poles at  $z = 1$  are added to the open-loop function, and/or if the open-loop gain is increased. However, added poles at  $z = 1$  in the open-loop function introduce phase lag into the open-loop frequency response, resulting in reduced stability margins. Thus stability problems may ensue. In addition, an increase in the open-loop gain generally results in stability problems, as was seen in Chapter 7. Thus a control system design is usually a trade-off between steady-state accuracy and acceptable relative stability (acceptable stability margins).

### Transient Response

We define a physical system that has *two dominant poles* as one that can be modeled with reasonable accuracy by a second-order transfer function. Figure 8-1(a) illustrates a typical step response for a system that has two dominant complex poles. Typical performance criteria are rise time  $t_r$ , peak overshoot  $M_p$ , time-to-peak overshoot  $t_p$ , and settling time  $t_s$ . Rise time in this figure is the time required for the response to rise from 10 percent to 90 percent of the final



(a)



(b)

FIGURE 8-1 System step response and bandwidth.



value. However, other definitions are also used for rise time, but all are similar. Settling time  $t_s$  is defined as the time required for the response to settle to within a certain percent of the final value. Typical percentage values used are 2 percent and 5 percent.

For a given system, the time response is uniquely related to the closed-loop frequency response. However, except for first- and second-order systems, the exact relationship is complex and is generally not used. As indicated in Section 7.10, a typical closed-loop frequency response is as shown in Figure 8-1(b), where only the magnitude is shown. In this figure,  $M_r$  is the resonant peak value of the frequency response, and as was implied in Example 7.13, a larger resonant peak value indicates a larger peak overshoot  $M_p$  in the step response. For example, a control-system specification sometimes used is to limit  $M_r$  to 2 dB in order to limit  $M_p$  to a reasonable value.

Consider the standard second-order LTI analog system with the transfer function

$$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (8-1)$$

which has the unit-step response [1]

$$c(t) = 1 - \frac{1}{\beta} e^{-\zeta\omega_n t} \sin(\beta\omega_n t + \theta)$$

where  $\beta = \sqrt{1 - \zeta^2}$  and  $\theta = \tan^{-1}(\beta/\zeta)$ . Shown in Fig. 8-2 are the relationships of the parameters  $M_p$ ,  $M_r$ , and  $t_p$  of Fig. 8-1 to the damping ratio  $\zeta$  for this system. These plots are obtained from the equations [1]

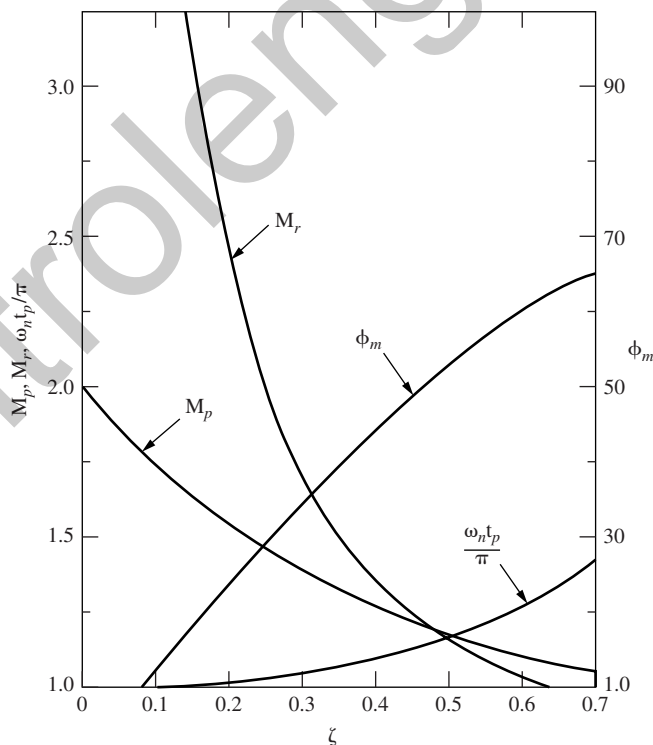


FIGURE 8-2 Characteristics of a second-order system.

$$\begin{aligned}
 M_p &= 1 + \varepsilon^{-\zeta\pi\sqrt{1-\zeta^2}} \\
 M_r &= \frac{1}{2\zeta\sqrt{1-\zeta^2}} \\
 \frac{\omega_n t_p}{\pi} &= \frac{1}{\sqrt{1-\zeta^2}}
 \end{aligned} \tag{8-2}$$

The relationships apply only approximately to second-order complex poles of higher-order discrete systems (see Section 6.4). However, we see the importance of the damping ratio  $\zeta$  of complex poles. The fourth curve in Fig. 8-2 will be discussed in the next section.

In the damped sinusoidal response term of the second-order system response  $c(t)$  just given, the time constant is seen to be  $\tau = 1/(\zeta\omega_n)$ . This time constant determines the settling time  $t_s$  of Fig. 8-1(a). For example, for the response  $c(t)$  to settle to within 2 percent of the final value,

$$\varepsilon^{-t_s/\tau} = 0.02$$

or  $t_s = 3.9\tau$ . Hence the system response settles out in approximately four time constants.

The transient response is also related to the system bandwidth, shown as  $\omega_b$  in Fig. 8-1(b). For a system, the product of rise time and bandwidth (i.e., the product  $t_r\omega_b$ ) is approximately constant [1, 2]. Thus, to decrease rise time and increase the speed of response, it is necessary that the system bandwidth be increased. However, if significant high-frequency noise sources are present in the system, a larger bandwidth will increase the system response to these noise sources. In this case, a trade-off must be made between a fast rise time and an acceptable noise response.

### Relative Stability

In Chapter 7 the relative stability measurements, gain margin and phase margin, were introduced. These margins are an approximate indication of the closeness of the Nyquist diagram (open-loop frequency response) to the  $-1$  point. As was shown in Section 7.10, the closeness of the open-loop frequency response to the  $-1$  point in the complex plane determines the resonant peak value  $M_r$  of the closed-loop frequency response (see Fig. 8-1(b)). And  $M_r$  is related, in an approximate sense, to the peak overshoot  $M_p$  (Fig. 8-1) in the step response. Thus, in an approximate sense, the stability margins are related to peak overshoot  $M_p$ . Consider the closed-loop function of (8-1) for which the open-loop function  $G_p(s)$  is given by

$$G_p(s) = \frac{\omega_n^2}{s(s + 2\zeta\omega_n)} \Rightarrow T(s) = \frac{G_p(s)}{1 + G_p(s)}$$

For this system, the phase margin  $\phi_m$  and the peak overshoot  $M_p$  are directly related, as shown in Fig. 8-2. The curve plotted in Fig. 8-2 is the relationship [1]

$$\phi_m = \tan^{-1} \left[ \frac{2\zeta}{(\sqrt{4\zeta^4 + 1} - 2\zeta^2)^{1/2}} \right]$$

This equation is often approximated by (see Problem 8.2-1)

$$\phi_m \approx 100\zeta$$

### Sensitivity

Generally, any control system will contain parameters that change with temperature, humidity, altitude, age, and so on. However, we prefer that the control system characteristics remain stable as these parameters vary. Of course, the system characteristics are a function of the system parameters, but in some cases the sensitivity of system characteristics to parameter variations can be reduced. A simple case will now be discussed.

Consider the discrete system of Fig. 8-3(a). For this system the closed-loop transfer function  $T(z)$  is given by

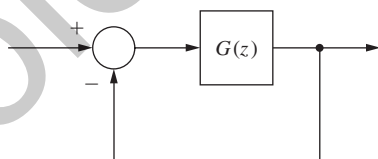
$$T(z) = \frac{G(z)}{1 + G(z)}$$

Sensitivity to a parameter  $a$  is normally defined as a measure of the percentage change in  $T(z)$  to a percentage change in the parameter  $a$ . One such definition is

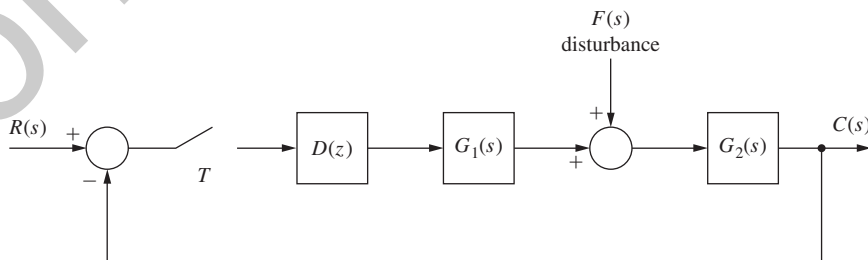
$$\text{sensitivity} \approx \frac{\Delta T/T}{\Delta a/a} = \frac{\Delta T}{\Delta a} \frac{a}{T} \quad (8-3)$$

where  $\Delta T$  is the variation in  $T$  caused by  $\Delta a$ , the variation in parameter  $a$ . If the limit of (8-3) is taken as  $\Delta a \rightarrow 0$ , we get the usual definition of *sensitivity*; that is,

$$S_a^T = \frac{\partial T}{\partial a} \frac{a}{T} \quad (8-4)$$



(a)



(b)

**FIGURE 8-3** Discrete control systems.

We will now find the sensitivity of  $T$  with respect to  $G$  for the system of Fig. 8-3(a).

$$\begin{aligned}
 S_G^T &= \frac{\partial T}{\partial G} \cdot \frac{G}{T} = \frac{1 + G(z) - G(z)}{[1 + G(z)]^2} \cdot \frac{G(z)}{G(z)/[1 + G(z)]} \\
 &= \frac{1}{1 + G(z)}
 \end{aligned}$$

At the frequency  $\omega$ , we let  $z = \epsilon^{j\omega T}$ , and

$$S_G^T = \frac{1}{1 + G(\epsilon^{j\omega T})} \quad (8-5)$$

For this sensitivity to be small within the system bandwidth, we require that  $G(\epsilon^{j\omega T}) \gg 1$ . Thus we can reduce the sensitivity of  $T$  to  $G$  by increasing the open-loop gain. But as noted before, increasing the open-loop gain can cause stability problems. Thus, once again, in design we are faced with trade-offs.

Consider now that  $G(z)$  is a function of the parameter  $a$ . Then we can express (8-4) as

$$S_a^T = \frac{\partial T}{\partial a} \frac{a}{T} = \frac{\partial T}{\partial G} \frac{\partial G}{\partial a} \frac{a}{T}$$

Thus

$$\begin{aligned}
 S_a^T &= \frac{1 + G(z) - G(z)}{[1 + G(z)]^2} \frac{a}{G(z)/[1 + G(z)]} \frac{\partial G(z)}{\partial a} \\
 &= \frac{\frac{\partial G(z)}{a} \frac{a}{\partial a}}{G(z)[1 + G(z)]}
 \end{aligned} \quad (8-6)$$

Then, as in (8-5), to reduce the sensitivity we must increase the open-loop gain.

### Disturbance Rejection

A control system will generally have inputs other than the one to be used to control the system output. An example is shown in Fig. 8-3(b). In this system  $F(s)$  is a disturbance. Since  $R(s)$  is the control input (or the reference input), we design the system such that  $c(t)$  is approximately equal to  $r(t)$ . If  $F(s)$  is zero, then

$$C(z) = \frac{D(z)\overline{G_1 G_2(z)}}{1 + D(z)\overline{G_1 G_2(z)}} R(z)$$

Hence, in terms of the frequency response, we require that

$$D(\epsilon^{j\omega T})\overline{G_1 G_2(\epsilon^{j\omega T})} \gg 1$$

over the desired system bandwidth. Then

$$C(\epsilon^{j\omega T}) \approx R(\epsilon^{j\omega T})$$

If we consider only the disturbance input in Fig. 8-3(b), then

$$C(z) = \frac{\overline{G_2 F}(z)}{1 + D(z)\overline{G_1 G_2}(z)}$$

Hence, over the desired system bandwidth,

$$C(e^{j\omega T}) \approx \frac{\overline{G_2 F}(e^{j\omega T})}{D(e^{j\omega T})\overline{G_1 G_2}(e^{j\omega T})}$$

Since the denominator of this expression is large, the disturbance response will be small, provided that the numerator is not large. Therefore, we generally have good disturbance rejection in a system provided that we have a high loop gain, and provided that the high loop gain does not occur in the direct path between the disturbance input and the system output [ $G_2(s)$  in Fig. 8-3(b)].

### Control Effort

Another criterion that must be considered in the design of a control system is the *control effort*. For example, generally in a radar tracking system, an electric motor is used to rotate the radar antenna. Any physical motor will have a maximum torque that can be developed. If we call this control effort (the torque)  $u(t)$ , then  $|u(t)|$  will be bounded. One procedure for including this constraint in the design of the system is to first design without considering the maximum torque available. Next the designed system is simulated under worst-case conditions to determine the maximum torque required. Then a physical motor is chosen that can produce this value of torque. This design procedure may be iterative, since the chosen motor may not have the model assumed in the initial design.

Another example of constraints on control effort is the maximum energy that may be available over a time period. This constraint is usually stated as

$$\int_0^{t_f} |u^2(t)| dt \leq M$$

For example, certain types of attitude controllers for satellites have limited energy available. The topic of constraints on the control effort is covered in the design of certain types of optimal control systems in Chapter 11.

## 8.3 COMPENSATION

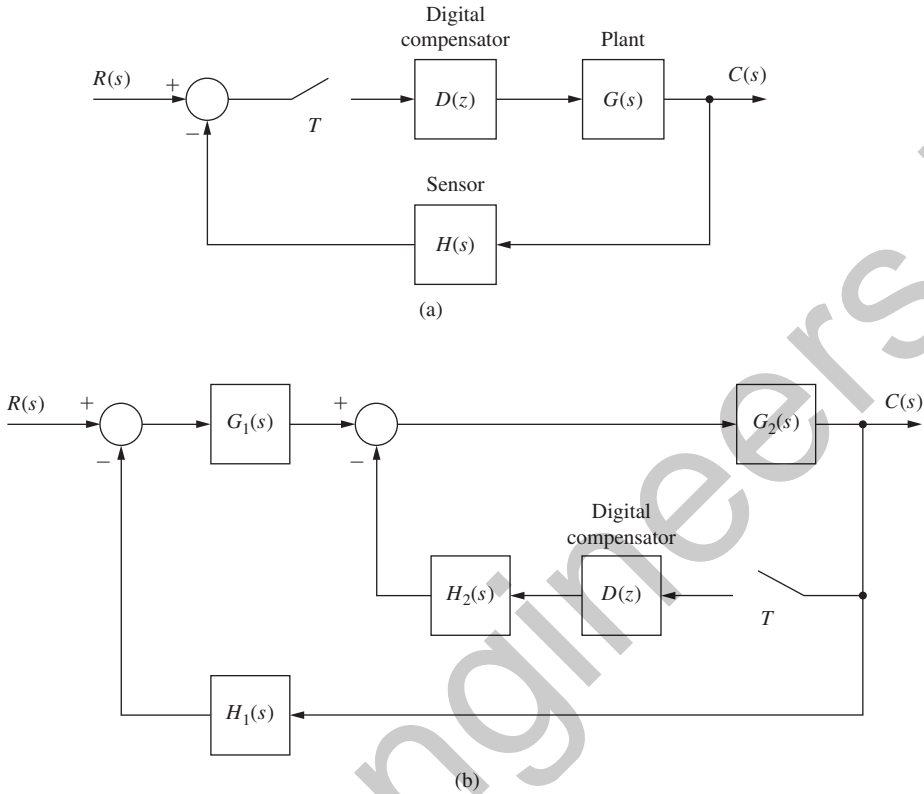
In this chapter we will, for the most part, limit the discussion to the design of compensators for single-input, single-output systems. A simple system of this type is shown in Fig. 8-4(a). For this system,

$$\frac{C(z)}{R(z)} = \frac{D(z)G(z)}{1 + D(z)\overline{GH}(z)} \tag{8-7}$$

and hence the characteristic equation is

$$1 + D(z)\overline{GH}(z) = 0 \tag{8-8}$$

We call compensation of the type shown in Fig. 8-4(a) *cascade*, or *series*, *compensation*. The effects of this compensation on system characteristics are given by the characteristic equation (8-8).



**FIGURE 8-4** Digital control systems.

It is sometimes more feasible to place the compensator within a loop internal to the system. Such a system is illustrated in Fig. 8-4(b). For this system,

$$C(z) = \frac{\left[ \frac{G_1 G_2 R}{1 + G_1 G_2 H_1} \right]}{1 + D(z) \left[ \frac{G_2 H_2}{1 + G_1 G_2 H_1} \right]} \quad (8-9)$$

The system characteristic equation is then

$$1 + D(z) \left[ \frac{G_2 H_2}{1 + G_1 G_2 H_1} \right] = 0 \quad (8-10)$$

This type of compensation is termed *feedback, parallel, or minor-loop compensation*. For this system, the effects of compensation on system characteristics are given by (8-10).

In the following three sections, basically we will consider compensation by a first-order device. Thus the compensator transfer function can be expressed as

$$D(z) = \frac{K_d(z - z_0)}{z - z_p} \quad (8-11)$$

The design of the compensator in these three sections will be performed in the frequency domain using Bode techniques; thus we will be working in the  $w$ -plane. The transformation of  $D(z)$  to the  $w$ -plane yields  $D(w)$ ; that is,

$$D(w) = D(z) \Big|_{z=[1+(T/2)w]/[1-(T/2)w]} \quad (8-12)$$

Thus  $D(w)$  is also first order, and we will assume it to be of the form

$$D(w) = a_0 \left[ \frac{1 + w/\omega_{w0}}{1 + w/\omega_{wp}} \right] \quad (8-13)$$

where  $\omega_{w0}$  is the zero location and  $\omega_{wp}$  is the pole location, in the  $w$ -plane. The dc gain of the compensator is found in (8-11) by letting  $z = 1$ , or in (8-13) by letting  $w = 0$ . Hence  $a_0$  is the compensator dc gain.

To realize the compensator, the transfer function must be expressed in  $z$ , as in (8-11). Then, from (7-8) and (8-13),

$$D(z) = a_0 \left. \frac{1 + \frac{w}{\omega_{w0}}}{1 + \frac{w}{\omega_{wp}}} \right|_{w=(2/T)[(z-1)/(z+1)]} = a_0 \frac{\omega_{wp}(\omega_{w0} + 2/T)}{\omega_{w0}(\omega_{wp} + 2/T)} \left[ \frac{z - \frac{2/T - \omega_{w0}}{2/T + \omega_{w0}}}{z - \frac{2/T - \omega_{wp}}{2/T + \omega_{wp}}} \right] \quad (8-14)$$

Hence, in (8-11),

$$K_d = a_0 \left[ \frac{\omega_{wp}(\omega_{w0} + 2/T)}{\omega_{w0}(\omega_{wp} + 2/T)} \right], \quad z_0 = \frac{2/T - \omega_{w0}}{2/T + \omega_{w0}}, \quad z_p = \frac{2/T - \omega_{wp}}{2/T + \omega_{wp}} \quad (8-15)$$

The compensator of (8-13) is classified by the location of the zero,  $\omega_{w0}$ , relative to that of the pole,  $\omega_{wp}$ . If  $\omega_{w0} < \omega_{wp}$ , the compensation is called *phase lead*. If  $\omega_{w0} > \omega_{wp}$ , the compensation is called *phase lag*. The phase-lag compensator will be discussed first.

## 8.4 PHASE-LAG COMPENSATION

In (8-13), for  $\omega_{w0} > \omega_{wp}$ , the frequency response of  $D(w)$  exhibits a negative phase angle, or phase lag. The frequency response of  $D(w)$ , as given by a Bode plot, is shown in Fig. 8-5. The dc gain is  $a_0$ , and the high-frequency gain is

$$(\text{high-frequency gain})_{\text{dB}} = 20 \log \frac{a_0 \omega_{wp}}{\omega_{w0}} \quad (8-16)$$

The phase characteristic is also shown in Fig. 8-5. The maximum phase shift is denoted as  $\phi_M$ , and has a value between 0 and  $-90^\circ$ , depending on the ratio  $\omega_{w0}/\omega_{wp}$ .

Design using phase-lag digital compensators will be discussed relative to the system of Fig. 8-6. For this system the characteristic equation is given by

$$1 + D(z)G(z) = 0 \quad (8-17)$$

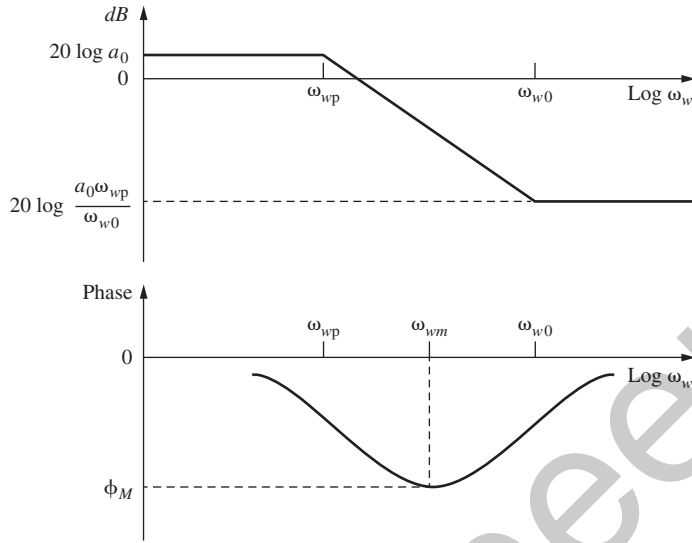


FIGURE 8-5 Phase-lag digital filter frequency-response characteristics.

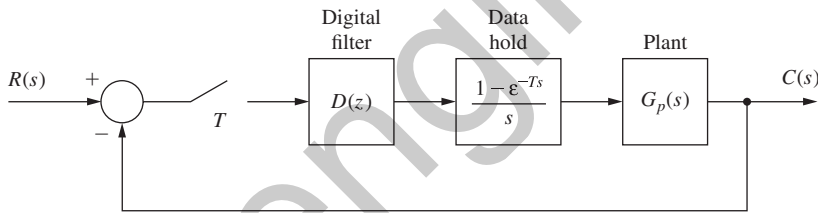


FIGURE 8-6 Digital control system.

where

$$G(z) = \mathfrak{Z} \left[ \frac{1 - e^{-Ts}}{s} G_p(s) \right] \tag{8-18}$$

For a system configuration differing from that of Fig. 8-6, the characteristic equation is formed as in (8-17), and the frequency response of the transfer function that multiplies  $D(z)$ , as in (8-17), is calculated. From that point, the design procedure follows that given below.

As is seen from Fig. 8-5, phase-lag filters reduce the high-frequency gain relative to the low-frequency gain and introduce phase lag. Since, in general, phase lag tends to destabilize a system (rotates the Nyquist diagram toward the  $-1$  point) the break frequencies,  $\omega_{wp}$  and  $\omega_{w0}$ , must be chosen such that the phase lag does not occur in the vicinity of the  $180^\circ$  crossover point of the plant frequency response  $G(j\omega_w)$ , where

$$G(w) = \mathfrak{Z} \left[ \frac{1 - e^{-Ts}}{s} G_p(s) \right]_{z=[1+(T/2)w]/[1-(T/2)w]} \tag{8-19}$$

for the system of Fig. 8-6. However, for stability purposes, it is necessary that the filter introduce the reduced gain in the vicinity of  $180^\circ$  crossover. Thus, both  $\omega_{wp}$  and  $\omega_{w0}$  must be much smaller



than the  $180^\circ$  crossover frequency. Fig. 8-7 illustrates design by phase-lag compensation, where the compensator dc gain is unity.

Note that in Fig. 8-7, both the system gain margin and the system phase margin  $\phi_m$  have been increased by the compensation, increasing relative stability. In addition, the low-frequency gain has not been reduced, and thus steady-state errors and low-frequency sensitivity have not been increased to attain the improved relative stability. The bandwidth has been decreased, which will generally result in a slower system time response.

Suppose that in Fig. 8-7, we keep both  $\omega_{w0}$  and the product  $a_0 \omega_{wp}$  constant, and increase  $a_0$  while decreasing  $\omega_{wp}$ . The high-frequency gain of the compensator remains constant, from (8-16). However, since we are increasing  $a_0$ , the system open-loop low-frequency gain increases. Hence the closed-loop low-frequency gain approaches unity and the steady-state response is improved. However, for a given system, the increase in phase lag added to system phase characteristics may push the phase characteristic below the  $-180^\circ$  line (see Fig. 8-7). Then we have a *conditionally stable system*, that is, one that can be forced to be unstable by reducing gain. If the system contains a saturation nonlinearity, large signals into this nonlinearity reduce its effective gain [1, 3]. Thus a phase-lag compensated system may exhibit instabilities for large signals (nonlinear operation).

A technique for determining  $\omega_{wp}$  and  $\omega_{w0}$  to yield a desired phase margin will now be given. It is assumed that the compensator dc gain,  $a_0$ , is determined from the system specifications, and that the phase margin of  $\phi_m$  is desired. Refer to Fig. 8-7. The design steps are as follows:

1. Determine the frequency,  $\omega_{w1}$ , at which the phase angle of  $G(j\omega_w)$  is approximately  $(-180^\circ + \phi_m + 5^\circ)$ . The phase margin of the compensated system will occur at approximately this frequency.

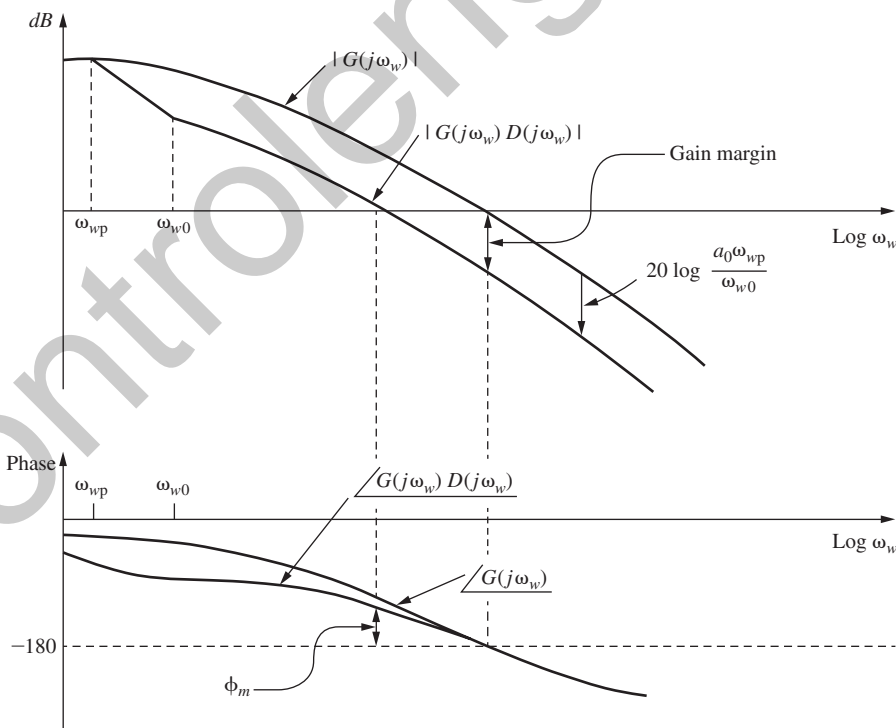


FIGURE 8-7 Design using phase-lag compensation.

2. Choose

$$\omega_{w0} = 0.1\omega_{w1} \quad (8-20)$$

to ensure that little phase lag is introduced at  $\omega_{w1}$ . Actually, the compensator will introduce approximately  $5^\circ$  phase lag, which has been accounted for in step 1.

3. At  $\omega_{w1}$ , we want  $|D(j\omega_{w1})G(j\omega_{w1})| = 1$ . Since the gain of the compensator at high frequencies is  $a_0\omega_{wp}/\omega_{w0}$  (where  $a_0$  is the compensator dc gain), then

$$\left| \frac{a_0\omega_{wp}}{\omega_{w0}} G(j\omega_{w1}) \right| = 1 \Rightarrow \frac{a_0\omega_{wp}}{\omega_{w0}} = \frac{1}{|G(j\omega_{w1})|}$$

(See Fig. 8-5.) Solving the last two equations for  $\omega_{wp}$  yields

$$\omega_{wp} = \frac{0.1\omega_{w1}}{a_0|G(j\omega_{w1})|} \quad (8-21)$$

The design is now complete, since the compensator dc gain, pole, and zero are known.

Once  $a_0$ ,  $\omega_{wp}$ , and  $\omega_{w0}$  are known,  $D(z)$  is obtained from (8-14). For the case that the sensor transfer function  $H(s)$  is not unity, replace  $G(j\omega_{w1})$  with  $\overline{GH}(j\omega_{w1})$  in step 1 and (8-21).

### EXAMPLE 8.1

We will consider the design of a servomotor system as described in Section 1.5. Suppose that the servo is to control the horizontal (azimuth) angle for pointing a radar antenna. Then, in the closed-loop system of Fig. 8-6,  $c(t)$  is the azimuth angle of the antenna, and  $r(t)$  is the commanded, or desired, azimuth angle. The plant transfer function derived in Section 1.5 is second order; however, we will assume that the armature inductance cannot be neglected, resulting in a third-order transfer function. Then suppose that the parameters of the plant are such that

$$G_p(s) = \frac{1}{s(s+1)(0.5s+1)}$$

Since the fastest time constant is 0.5 s, we will choose  $T$  to be one-tenth that value (a rule of thumb often used), or  $T = 0.05$  s. Then

$$\begin{aligned} G(z) &= \frac{z-1}{z} \mathfrak{Z} \left[ \frac{1}{s^2(s+1)(0.5s+1)} \right] \\ &= \frac{z-1}{z} \mathfrak{Z} \left[ \frac{1}{s^2} + \frac{-1.5}{s} + \frac{2}{s+1} + \frac{-0.5}{s+2} \right] \\ &= \frac{z-1}{z} \left[ \frac{0.05z}{(z-1)^2} - \frac{1.5z}{z-1} + \frac{2z}{z-0.9512} - \frac{0.5z}{z-0.9048} \right] \end{aligned}$$

The frequency response of this system was calculated by computer and is given in Table 8-1 and plotted in Fig. 8-8. Suppose that it is desired to design a unity dc gain phase-lag compensator

**TABLE 8-1** Frequency Response of the Plant in Example 8.1

$\omega$	$\omega_w$	$ G(\varepsilon^{j\omega T}) $	$ G(\varepsilon^{j\omega T}) _{dB}$	$\angle G(\varepsilon^{j\omega T})$
0.010	0.010	100.0	40.0	-90.9
0.050	0.050	19.97	26.0	-94.4
0.100	0.100	9.94	19.9	-98.7
0.200	0.200	4.88	13.8	-107.3
0.300	0.300	3.16	9.99	-115.6
0.360	0.360	2.57	8.21	-120.5
0.400	0.400	2.28	7.15	-123.7
0.500	0.500	1.74	4.79	-131.3
0.600	0.600	1.37	2.73	-138.5
0.700	0.700	1.105	0.87	-145.3
0.800	0.800	0.9064	-0.85	-151.6
0.900	0.900	0.7533	-2.46	-157.5
1.000	1.000	0.6330	-3.97	-163.0
1.200	1.200	0.4576	-6.79	-172.9
1.370	1.371	0.3550	-8.99	-180.3
1.500	1.501	0.2950	-10.6	-185.4
2.000	2.001	0.1584	-16.0	-201.4
3.000	3.006	0.0590	-24.6	-222.3
5.000	5.026	0.0151	-36.7	-244.3

( $a_0 = 1$ ) to achieve a phase margin of  $55^\circ$ . Then, using the foregoing procedure, we see that the frequency  $\omega_{w1}$  occurs where the phase of  $G(j\omega_w)$  is  $(-180^\circ + 55^\circ + 5^\circ) = -120^\circ$ , or  $\omega_{w1} \approx 0.36$ . At this frequency,  $|G(j\omega_{w1})| \approx 2.57$ . Then, from (8-20),

$$\omega_{w0} = 0.1\omega_{w1} = 0.036$$

and from (8-21),

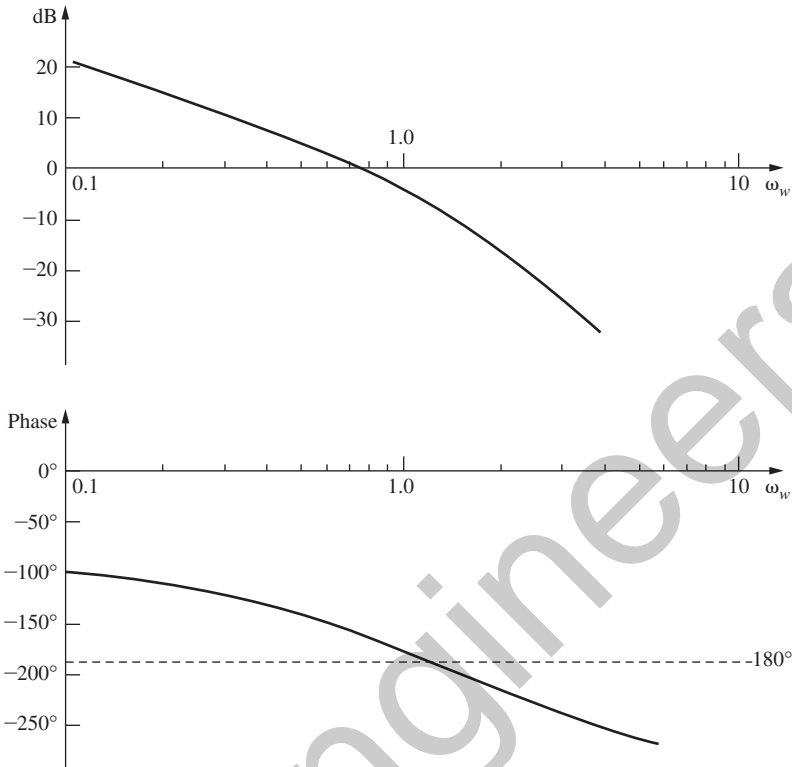
$$\omega_{wp} = \frac{0.1\omega_{w1}}{a_0|G(j\omega_{w1})|} = \frac{0.036}{(1)(2.57)} = 0.0140$$

Then  $D(w) = (1 + w/0.036)/(1 + w/0.0140)$ , and from (8-14),

$$D(z) = \frac{0.3890(z - 0.9982)}{(z - 0.993)} = \frac{0.3890z - 0.3883}{z - 0.9993}$$

A MATLAB program to perform the calculations in this example is now given.

```
>> Gp = tf([2],[1 3 2 0]); T = 0.05; a0 = 1;
Gz = c2d(Gp,T); %Assumes a ZOH
margin(Gz), grid
```



**FIGURE 8-8** Frequency response for the system of Example 8.1.

```

% Select ww1 from the Bode plot
% At ww1=0.360 and phase=-120deg
ww1 = 0.360;
[magGjww1,phaseGww1]=bode(Gz,ww1); % The design point
% Find D(z)
ww0 = 0.1*ww1; wwp = ww0/(a0*magGjww1);
Kd = a0*(wwp*(ww0+2/T))/(ww0*(wwp+2/T));
z0 = (2/T-ww0)/(2/T+ww0); zp = (2/T-wwp)/(2/T+wwp);
Dz = Kd*tf([1 -z0],[1 -zp],T) % Display D(z)
% Find the gain and phase margins of the compensated system
DzGz=Dz*Gz;
[Gm,Pm] = margin(DzGz); Gm dB = 20*log10(Gm), Pm
    
```

Calculation of the system open-loop frequency response shows that this compensator results in a gain margin of 16.8 dB and a phase margin of 55.9°.

With phase-lag compensation, numerical problems may occur in the realization of the filter coefficients. To illustrate this point, suppose that a microprocessor is used to implement the digital controller. Suppose, in addition, that filter coefficients are realized by a binary word that employs 8 bits to the right of the binary point. Then the fractional part of the coefficient can be represented as [4]

$$\text{fraction} = b_7 \cdot \frac{1}{2} + b_6 \cdot \frac{1}{4} + b_5 \cdot \frac{1}{8} + \dots + b_0 \cdot \frac{1}{2^8}$$

where  $b_i$  is the  $i$ th bit, and has a value of either 0 or 1. For example, the binary number

$$(0.11000001)_2 = \left( \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^8} \right)_{10} = (0.75390625)_{10}$$

The maximum value that the fraction can assume is  $[1 - 1/(2)^8]$ , or 0.99609375. Note, in Example 8.1, that a denominator coefficient of 0.999300 is required, but a value of 0.99609375 will be implemented ( $b_7$  to  $b_0$  are all equal to 1). The numerator coefficients, when converted by standard decimal-to-binary conversion algorithms [4], become

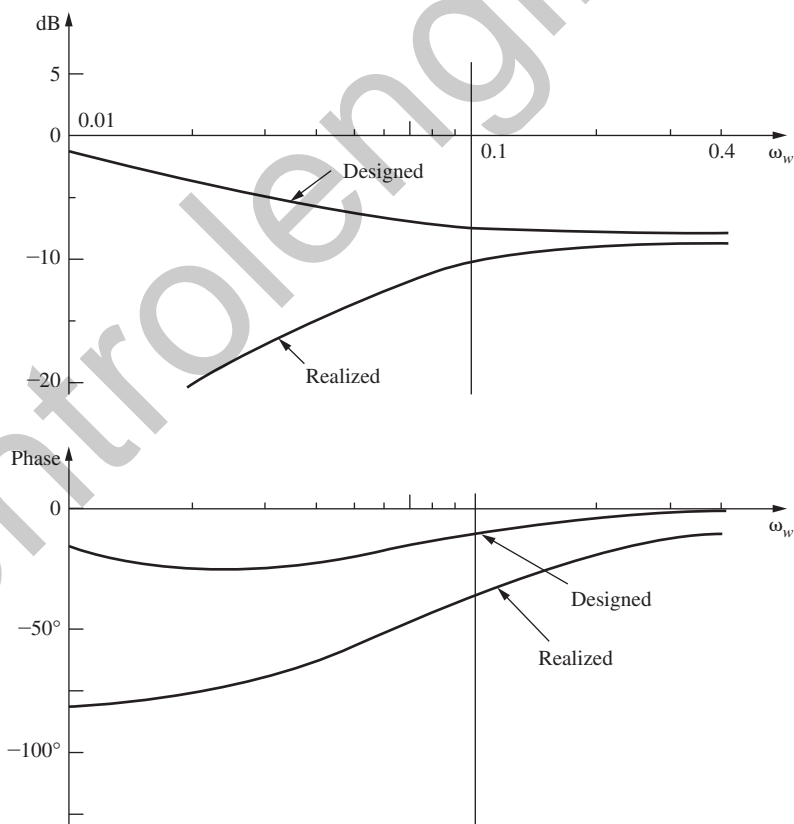
$$(0.3890)_{10} \Rightarrow (0.01100011)_2 = 0.38671875$$

$$(0.3883)_{10} \Rightarrow (0.01100011)_2 = 0.38671875$$

Thus the compensator zero has been shifted to  $z = 1$ , and the digital filter that is implemented has the transfer function

$$D(z) = \frac{0.38671875z - 0.38671875}{z - 0.99609375}$$

Shown in Fig. 8-9 are the frequency responses of the designed filter and the implemented filter, and the effects of coefficient quantization are evident. The resultant system stability margins,



**FIGURE 8-9** Frequency responses of designed and of realized digital controllers.

when the implemented filter is used, are phase margin  $72.7^\circ$  (designed value  $55.9^\circ$ ) and gain margin 18.1 dB (designed value 16.8 dB). However, the implemented filter has a dc gain of zero; thus the system will not respond correctly to a constant input. Hence more bits must be used to represent the filter coefficients.

We can view the coefficient quantization problem as one that results from the choice of the sample period  $T$ . We place digital filters in a physical system in order to change its real ( $s$ -plane) frequency response, and we want this change to occur over a certain real frequency ( $\omega$ ) range. The choice of  $T$  places this frequency range on a certain part of the unit circle in the  $z$ -plane, since  $z = e^{j\omega T} = 1 \angle (\omega T)$ . Thus since phase-lag filtering occurs for  $\omega$  small, the choice of  $T$  to be small requires that the filtering occur in the vicinity of the  $z = 1$  point. Thus the phase-lag pole and zero will occur close to  $z = 1$ , and thus close to each other. If  $T$  can be chosen to be a larger value, the phase-lag pole and zero will move away from the  $z = 1$  point (and each other), and the numerical accuracy required for the filter coefficients will not be as great.

### 8.5 PHASE-LEAD COMPENSATION

Phase-lead compensation will now be discussed. For a phase-lead compensator, in (8-13)  $\omega_{w0} < \omega_{wp}$  and the compensator frequency response is as shown in Fig. 8-10. The maximum phase shift,  $\theta_M$ , occurs at a frequency  $\omega_{wm}$ , where  $\omega_{wm}$  is the geometric mean of  $\omega_{w0}$  and  $\omega_{wp}$ , that is,

$$\omega_{wm} = \sqrt{\omega_{w0}\omega_{wp}} \quad (8-22)$$

A plot of  $\theta_M$  versus the ratio  $\omega_{wp}/\omega_{w0}$  is given in Fig. 8-11. This plot was obtained through the following development. We can express (8-13) as

$$D(j\omega_w) = |D(j\omega_w)|e^{j\theta} = a_0 \left[ \frac{1 + j(\omega_w/\omega_{w0})}{1 + j(\omega_w/\omega_{wp})} \right] \quad (8-23)$$

Then

$$\tan\theta = \tan \left[ \tan^{-1} \frac{\omega_w}{\omega_{w0}} - \tan^{-1} \frac{\omega_w}{\omega_{wp}} \right] = \tan(\alpha - \beta) \quad (8-24)$$

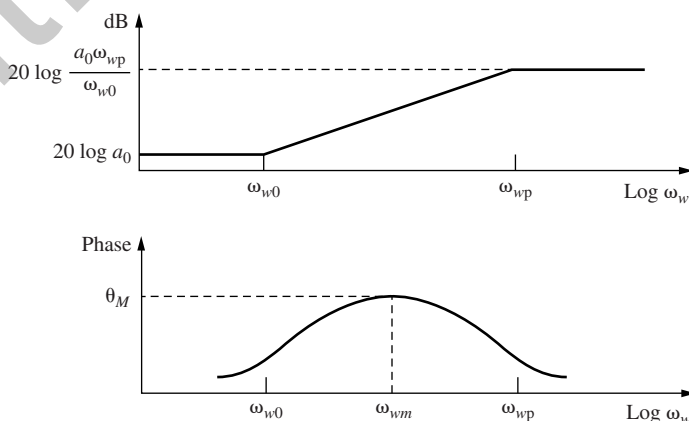
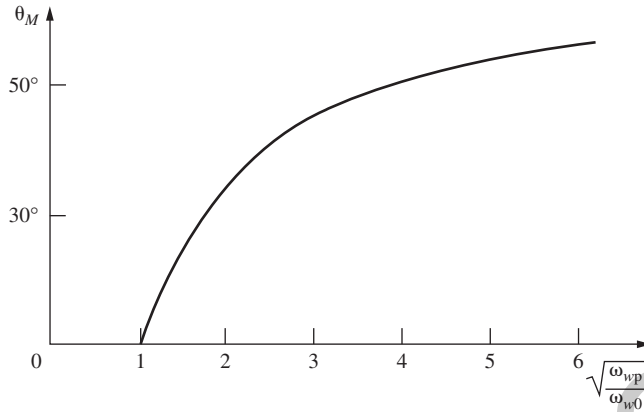


FIGURE 8-10 Phase-lead digital filter frequency-response characteristics.



**FIGURE 8-11** Maximum phase shift for a phase-lead filter.

Thus

$$\tan \theta = \frac{\tan \alpha - \tan \beta}{1 + \tan \alpha \tan \beta} = \frac{\omega_w/\omega_{w0} - \omega_w/\omega_{wp}}{1 + \omega_w^2/\omega_{w0}\omega_{wp}} \quad (8-25)$$

Then, from (8-22) and (8-25),

$$\tan \theta_M = \frac{1}{2} \left[ \sqrt{\frac{\omega_{wp}}{\omega_{w0}}} - \sqrt{\frac{\omega_{w0}}{\omega_{wp}}} \right] \quad (8-26)$$

From this equation,  $\theta_M$  is seen to be a function only of the ratio  $\omega_{wp}/\omega_{w0}$ . Figure 8-11 is a plot of (8-26). Note also that

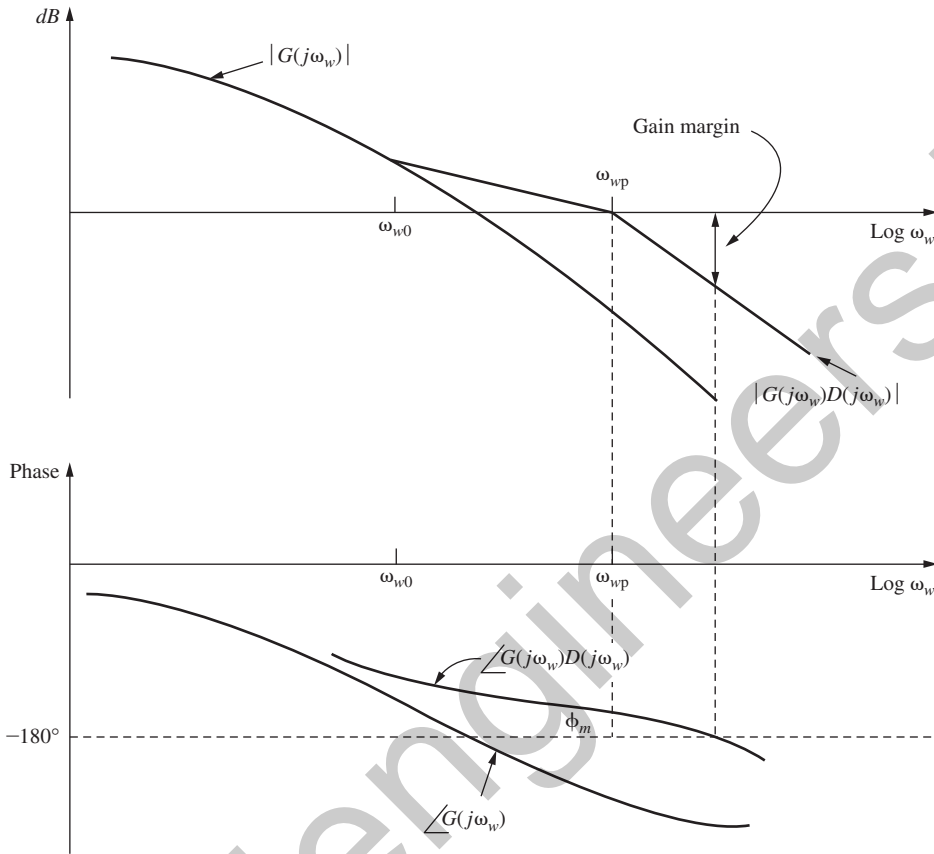
$$|D(j\omega_{wm})| = a_0 \frac{\sqrt{1 + (\omega_w/\omega_{w0})^2}}{\sqrt{1 + (\omega_w/\omega_{wp})^2}} \bigg|_{\omega_w = \omega_{wm}} = a_0 \sqrt{\frac{1 + (\omega_{wp}/\omega_{w0})}{1 + (\omega_{w0}/\omega_{wp})}} = a_0 \sqrt{\frac{\omega_{wp}}{\omega_{w0}}} \quad (8-27)$$

It is seen in Fig. 8-10 that phase-lead compensation introduces phase lead, which is a stabilizing effect, but also increases the high-frequency gain relative to the low-frequency gain, which is a destabilizing effect. Design using phase-lead compensation is illustrated in Fig. 8-12. The phase lead is introduced in the vicinity of the plant's  $180^\circ$  crossover frequency, in order to increase the system's stability margins. Note that system bandwidth is also increased, resulting in a faster time response. For Fig. 8-12, the compensator dc gain is unity.

The design of phase-lead compensation tends to be a trial-and-error procedure since, in the frequency range that the stabilizing phase lead is added, destabilizing gain is also added. Given in the next section is a procedure that will yield a specified phase margin, but has no control on the gain margin.

## 8.6 PHASE-LEAD DESIGN PROCEDURE

Presented in this section is a design procedure that will yield a specified phase margin in a discrete control system, provided that the designed system is stable. That is, the procedure will set the gain and phase of the open-loop function to specified values at a given frequency, and we



**FIGURE 8-12** Design using phase-lead compensation.

choose the specified gain to be 0 dB and the specified phase to be  $(180^\circ + \phi_m)$ , where  $\phi_m$  is the desired phase margin. Thus the procedure does not determine the gain margin, and may in fact result in an unstable system. Then, as a later step in the design procedure, it will be necessary to check the gain margin to ensure that it is adequate.

The characteristic equation for the system of Fig. 8-6 is

$$1 + D(w)G(w) = 0$$

where  $D(w)$  is given by (8-13). Our design problem may then be stated with respect to this system:

Determine  $D(w)$  in (8-28) such that, at some frequency  $\omega_{w1}$ ,

$$D(j\omega_{w1})G(j\omega_{w1}) = 1 \angle (180^\circ + \phi_m) \tag{8-28}$$

and, in addition, the system possesses an adequate gain margin.

The design equations will now be developed. Let  $D(w)$  be expressed as

$$D(w) = \frac{a_1w + a_0}{b_1w + 1} = a_0 \left[ \frac{1 + w/(a_0/a_1)}{1 + w/(b_1)^{-1}} \right] \tag{8-29}$$



where  $a_0$  is the compensator dc gain. Then, in (8-13), (8-14), and (8-15),

$$\omega_{w0} = \frac{a_0}{a_1}, \quad \omega_{wp} = \frac{1}{b_1} \quad (8-30)$$

This notation is more convenient than that used for the phase-lag compensator.

From (8-28) we see that  $D(j\omega_{w1})$  must satisfy the relationships

$$|D(j\omega_{w1})| = \frac{1}{|G(j\omega_{w1})|} \quad (8-31)$$

and

$$\angle D(j\omega_{w1}) = 1\angle(180^\circ + \phi_m) - \angle G(j\omega_{w1})$$

where the symbol  $\angle D$  denotes the angle associated with the complex number  $D$ . Let the angle associated with  $D(j\omega_{w1})$  be denoted as  $\theta$ ; that is,

$$\theta = \angle D(j\omega_{w1}) = 180^\circ + \phi_m - \angle G(j\omega_{w1}) \quad (8-32)$$

In the equation for  $D(w)$ , (8-29), we see that there are three unknowns, namely,  $a_0$ ,  $a_1$ , and  $b_1$ . Equations (8-31) and (8-32) give two constraints on  $D(j\omega_{w1})$ . Solving these two equations for  $a_1$  and  $b_1$  as functions of  $a_0$  and  $\omega_{w1}$  yields

$$a_1 = \frac{1 - a_0 |G(j\omega_{w1})| \cos\theta}{\omega_{w1} |G(j\omega_{w1})| \sin\theta} \quad (8-33a)$$

$$b_1 = \frac{\cos\theta - a_0 |G(j\omega_{w1})|}{\omega_{w1} \sin\theta} \quad (8-33b)$$

(See Appendix I.) In the case that the sensor transfer function  $H(s)$  is not unity, replace  $G(j\omega_{w1})$  with  $\overline{GH}(j\omega_{w1})$  in (8-31), (8-32), and (8-33).

If the compensator coefficients satisfy the preceding equations, the Nyquist diagram will pass through the point  $1\angle(180^\circ + \phi_m)$ . If the designed system is stable, this system has the required phase margin. However, nothing in the design procedure guarantees stability. Thus, once the coefficients are calculated, the Bode diagram (or Nyquist diagram) must be calculated to determine if the closed-loop system is stable.

This design procedure requires that the compensator dc gain  $a_0$  and the system phase-margin frequency  $\omega_{w1}$  be chosen. Then (8-33) determines the compensator coefficients  $a_1$  and  $b_1$ . The compensator dc gain is usually determined by steady-state specifications for the control system. The frequency  $\omega_{w1}$  can be approximately determined in the following manner. Since the compensator is phase lead, the angle  $\theta$  of (8-32) must be positive. Thus, from (8-32),  $\theta > 0$  yields

1.  $\angle G(j\omega_{w1}) < 180^\circ + \phi_m$

Also,  $|D(j\omega_{w1})| > a_0$  from Fig. 8-10 and from (8-31),

2.  $|G(j\omega_{w1})| < 1/a_0$

In addition, the coefficient  $b_1$  must be positive, to ensure a stable controller. Thus, from (8-33),

$$3. \cos\theta > a_0 |G(j\omega_{w1})|$$

Hence, the phase-margin frequency  $\omega_{w1}$  must be chosen to satisfy these three constraints.

It should be noted that (8-33) may also be utilized in phase-lag filter design, since nothing in the development of these equations is based on the filter begin phase lead. For phase-lag design, three equivalent constraints on  $\omega_{w1}$  can be developed (see Problem 8.6-3).

### EXAMPLE 8.2

Consider again the system of Example 8.1, whose frequency response is given in Figure 8-8 and Table 8-1. A phase margin of  $55^\circ$  is to be achieved, and a unity-dc-gain phase-lead compensator will be employed. Consider Table 8-1. We must choose a frequency  $\omega_{w1}$  such that  $\angle G(j\omega_{w1}) < -125^\circ$  (constraint 1) and  $|G(j\omega_{w1})| < 1$  (constraint 2). We rather arbitrarily choose  $\omega_{w1} = 1.200$  to satisfy these two constraints, and from Table 8-1,  $G(j\omega_{w1}) = 0.4574\angle(-172.9^\circ)$ . From (8-32),

$$\theta = 180^\circ + 55^\circ - (-172.9^\circ) = 407.9^\circ = 47.9^\circ$$

Constraint 3 yields  $\cos 47.9^\circ = 0.670 > 0.4576$ , and all constraints are satisfied. Hence, from (8-33),

$$a_1 = \frac{1 - (1)(0.4574) \cos(47.9^\circ)}{(1.2)(0.4574) \sin(47.9^\circ)} = 1.703$$

$$b_1 = \frac{\cos(47.9^\circ) - (1)(0.4576)}{(1.2) \sin(47.9^\circ)} = 0.2397$$

and thus, from (8-29),

$$D(w) = \frac{1 + 1.7028w}{1 + 0.2397w} = \frac{1 + w/0.5873}{1 + w/4.1719}$$

We then obtain the filter transfer function from (8-14):

$$D(z) = \frac{6.5278(z - 0.9711)}{z - 0.8111}$$

Calculation of the system open-loop frequency response shows that this compensator results in a phase margin of  $55^\circ$  and a gain margin of 12.4 dB. If we choose  $\omega_{w1}$  as a value different from 1.2 above, the phase margin will remain at  $55^\circ$ , but the gain margin will be different. If  $\omega_{w1}$  is chosen larger, then  $\theta$ , the angle of  $D(j\omega_{w1})$ , is larger (see Table 8-1), and the ratio  $\omega_{wp}/\omega_{w0}$  is larger. Thus, from Fig. 8-10, the high-frequency gain increases, which increases the system bandwidth. Choosing  $\omega_{w1}$  less than 1.2 will reduce the system bandwidth. The calculations in this example are implemented in the MATLAB program:

```
>> Gp = tf([2], [1 3 2 0]);
Gz = c2d(Gp, .05); %Assumes ZOH;
disp('Enter the required parameters: ')
```

```

a0 = input('Filter dc gain a0 = '); % enter 1
Pm = input('Desired phase margin Pm = '); % enter 55
ww1 = input('Phase margin frequency ww1 = '); % enter 1.2
[magGww1,phaseGww1] = bode(Gz,ww1)
thetad = 180+Pm-phaseGww1; thetar = thetad*pi/180;
% Find D(w) using equation (8-33)
disp(' ')
disp('D(w) = [a1*w + a0]/[b1*w + 1] ')
a1 = (1-a0*magGww1*cos(thetar))/(ww1*magGww1*sin(thetar))
a0
b1 = (cos(thetar)-a0*magGww1)/(ww1*sin(thetar))
% Find D(z) using equations 8-30 and 8-15
ww0 = a0/a1; wwp = 1/b1; T=0.05;
Kd = a0*(wwp*(ww0+2/T))/(ww0*(wwp+2/T));
z0 = (2/T-ww0)/(2/T+ww0); zp = (2/T-wwp)/(2/T+wwp);
Dz = Kd*tf([1 -z0],[1 -zp],T);
Dz = zpk(Dz)
disp('Gain and Phase Margins:')
Gp = tf([2],[1 3 2 0]); Gz = c2d(Gp,T);
DzGz = Dz*Gz; [Gm,Pm] = margin(DzGz); GmdB = 20*log10(Gm), Pm
    
```

The coefficient quantization problem observed for the phase-lag filter generally does not occur in phase-lead filters. For a phase-lag filter, the pole and zero are almost coincident, making their placement critical. For the phase-lead filter, the pole and zero are well separated, and any small shift in either one has little effect on the filter frequency response.

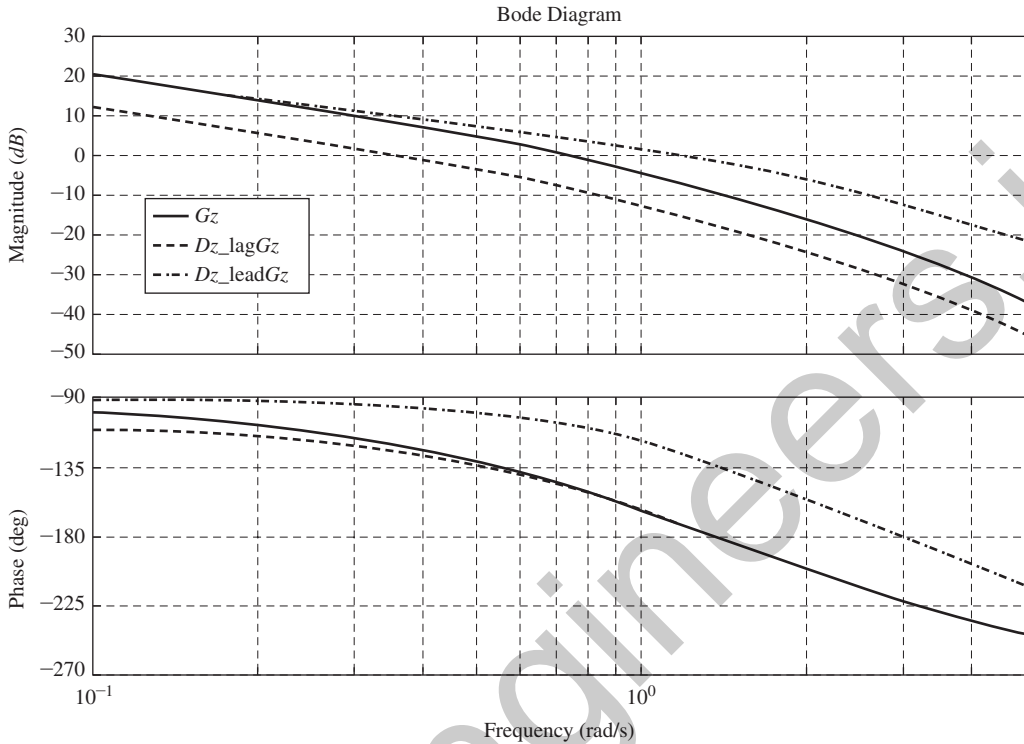
Examples 8.1 and 8.2 illustrate simple phase-lag and phase-lead compensation. The effect of phase-lag compensation is to reduce the open-loop gain at higher frequencies, which in turn reduces system bandwidth. The open-loop gain at lower frequencies is not reduced (can be increased), and thus steady-state errors are not increased (can be reduced). The effect of phase-lead compensation is to increase open-loop gain at higher frequencies, and thus increases system bandwidth.

The frequency and step response curves for Examples 8.1 and 8.2 are plotted in Figs. 8-13 and 8-14 by the MATLAB program:

```

>> T = 0.05; Gp = tf([2],[1 3 2 0]); Gz = c2d(Gp,T);
Dz_lag = tf([0.3890 -0.3883],[1 -0.9993],T);
Dz_lead = 6.5278*tf([1 -0.9711],[1 -0.8111],T);
Dz_lagGz = Dz_lag*Gz;
Dz_leadGz = Dz_lead*Gz;
% Plot Figure 8-13
figure(1),bode(Gz,'-k',Dz_lagGz,'--k',Dz_leadGz,'-.k'),grid
% Plot Figure 8-14(a)
Cz_uncomp = feedback(Gz,1);
Cz_lag = feedback(Dz_lagGz,1);
Cz_lead = feedback(Dz_leadGz,1);
figure(2),bode(Cz_uncomp,'-k',Cz_lag,'--k',Cz_lead,'-.k'),grid
% Plot Figure 8-14(b)
figure(3),step(Cz_uncomp,'-k',Cz_lag,'--k',Cz_lead,'-.k'),grid
    
```

The open-loop frequency responses of the system of the foregoing two examples without compensation, with phase-lag compensation, and with phase-lead compensation are shown in Fig. 8-13. Figure 8-14(a) gives the magnitudes of the closed-loop frequency responses for the two examples and for the uncompensated system. Note the reduced bandwidth for

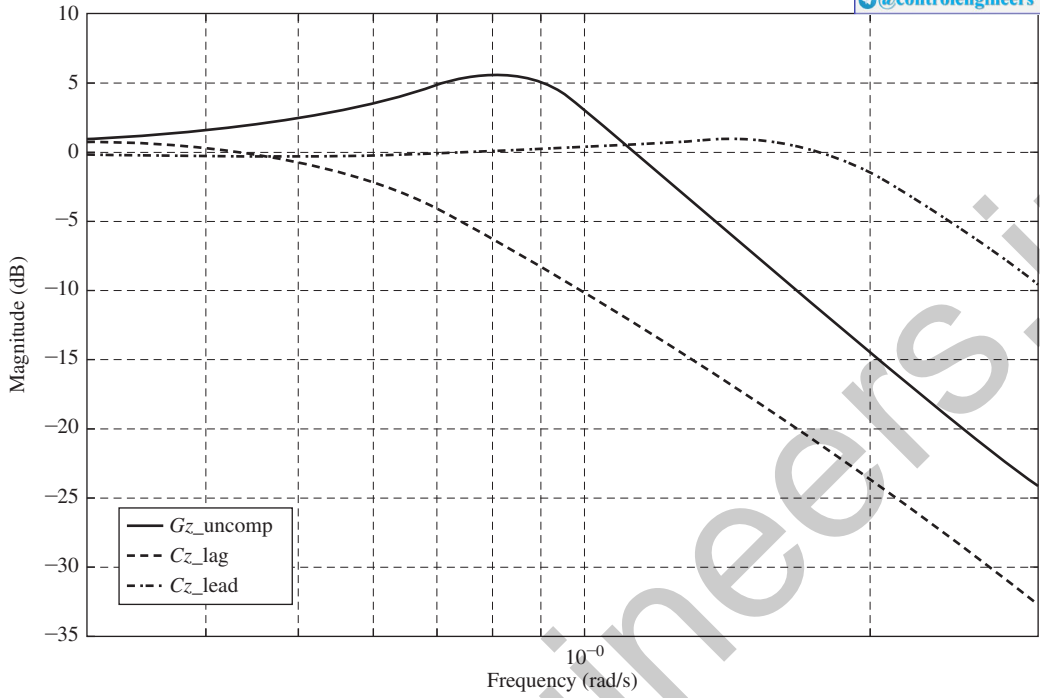


**FIGURE 8-13** Open-loop frequency responses for systems of Examples 8.1 and 8.2.

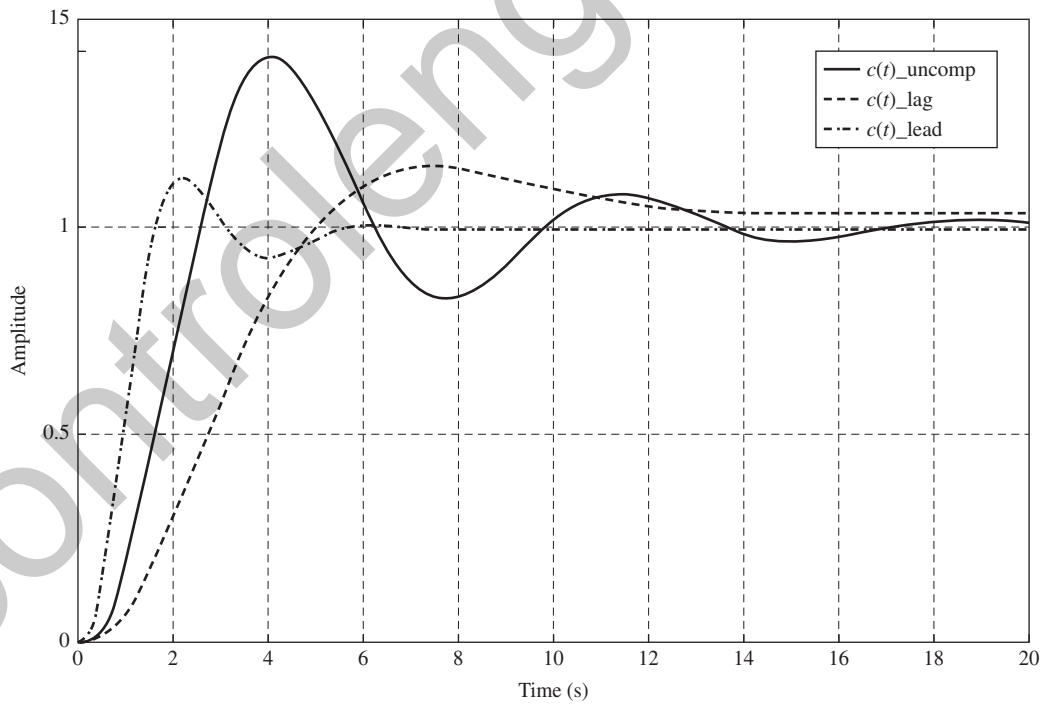
phase-lag compensation and increased bandwidth for phase-lead compensation. Shown in Fig. 8-14(b) are the step responses of the closed-loop system with phase-lag compensation and with phase-lead compensation. Note that the system step response is much faster for the phase-lead case, because of the increased system bandwidth. Table 8-2 gives several step-response characteristics (see Figure 8-1) for the two cases. Note that while the phase margins are equal, the peak overshoots are not. Thus phase margin alone does not determine peak overshoot.

**TABLE 8-2** Step-Response Characteristics

Characteristic	Un-compensated	Phase lag	Phase lead
Steady-state error	0	0	0
Percent overshoot	40.51	14.07	11.50
Rise time, $t_r$ (s)	1.55	3.15	1.05
Peak time $t_p$ (s)	4.10	7.45	2.20
Settling time, $t_s$ , for $d = 0.02$ (s)	16.25	32.35	5.20
Bandwidth, $\omega_b$ (rad/s)	1.26	0.65	2.19



(a)



(b)

**FIGURE 8-14** Closed-loop responses for Examples 8.1 and 8.2: (a) frequency, (b) unit-step.

Open-loop frequency response should not be confused with *closed-loop* frequency response. Figure 8-13 is a plot of the systems open-loop frequency responses, and Fig. 8-14(a) gives the closed-loop frequency responses for the same systems. Table 8-2 gives the closed-loop bandwidths, which were obtained from Fig. 8-14(a). These bandwidths are not available from Fig. 8-13 without extensive calculations. We design using the open-loop frequency response, for convenience, but the closed-loop frequency response more clearly indicates the input–output characteristics of the system.

An additional point should be made concerning phase-lead compensation. From Fig. 8-10 we see that the high-frequency gain of the digital filter can be quite large. Hence if the control system is burdened with high-frequency noise, the phase-lead compensation may lead to noise problems. If this is the case, some compromise in design may be required. One solution involves the use of a phase-lag compensator cascaded with a phase-lead compensator. The lag compensation is employed to realize a part of the required stability margins, thus reducing the amount of phase-lead compensation required. This compensation is discussed in the next section.

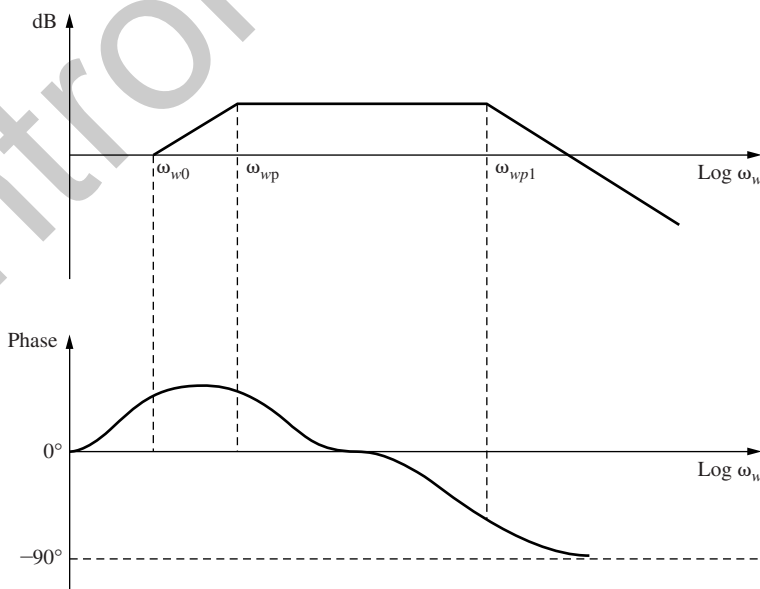
A second possible solution to this noise problem would be to add one or more poles to the filter transfer function. The pole (or poles) are placed at high frequencies such that the phase lag introduced by the poles does not significantly decrease the system stability margins. The required transfer function for a single pole is of the form

$$D_h(w) = \frac{1}{1 + w/\omega_{wp1}}$$

and the total filter transfer function is then

$$D(w) = \frac{a_1 w + a_0}{b_1 w + 1} \left[ \frac{1}{1 + w/\omega_{wp1}} \right]$$

The resultant filter frequency response is as shown in Fig. 8-15, for  $a_0 = 1$ .



**FIGURE 8-15** Phase-lead filter with an added pole.

Another problem originates in the large high-frequency gain of the phase-lead compensator. This problem is evident in the filter transfer function of Example 8.2. For this example,

$$D(z) = \frac{6.5278z - 6.3391}{z - 0.8111} = 6.5278 - 1.0444z^{-1} + \dots$$

Thus a step change of 1 unit in the filter input results in a step change of 6.5278 units in the filter output. This large signal out of the filter may force the plant into a nonlinear region of operation (e.g., an amplifier may saturate, or the digital-to-analog converter may saturate). Since the design is based on a linear plant model, the effects of forcing the system into nonlinear regions of operation will not, in general, be obvious.

**PHASE-LAG PROPERTIES.** In summary, some possible advantages of phase-lag compensation are:

1. The low-frequency characteristics are maintained or improved.
2. The stability margins are improved.
3. The bandwidth is reduced, which is an advantage if high-frequency noise is a problem. Also, for other reasons, reduced bandwidth may be an advantage.

**Some possible disadvantages are:**

1. The reduced bandwidth may be a problem in some systems as it slows down the system response.
2. The system transient response will have one very slow term. This will become evident when root-locus design is covered in Section 8.11.
3. Numerical problems with filter coefficients may result.

**PHASE-LEAD PROPERTIES.** For phase-lead compensation, some possible advantages are:

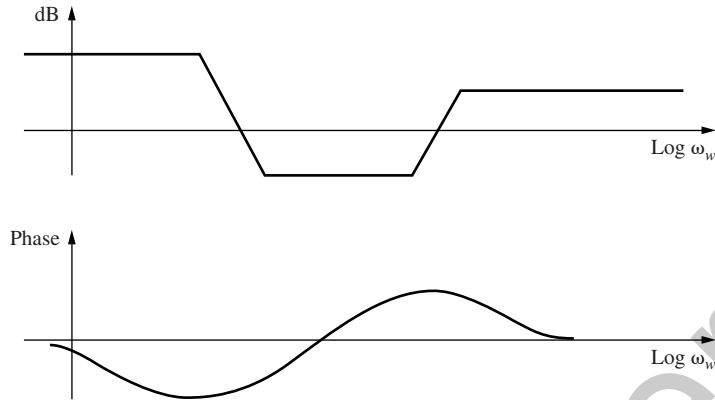
1. Stability margins are improved.
2. High-frequency performance, such as speed of response, is improved.
3. Phase-lead compensation is *required* to stabilize certain types of systems.

**Some possible disadvantages are:**

1. Any high-frequency noise problems are accentuated.
2. Large signals may be generated, which may damage the system or at least result in nonlinear operation of the system. Since the design assumed linearity, the results of the nonlinear operation will not be immediately evident.

## 8.7 LAG-LEAD COMPENSATION

In the preceding sections, only simple first-order compensators were considered. In many system design projects, however, the system specifications cannot be satisfied by a first-order compensator. In these cases higher-order filters must be used. To illustrate this point, suppose that smaller steady-state errors to ramp inputs are required for the system of Example 8.2. Then the low-frequency gain of the system must be increased. If phase-lead compensation is employed, this increase in gain must be reflected at all frequencies (see Fig. 8-10). It is then unlikely that one first-order section of phase-lead compensation can be designed to give adequate stability margins. One solution to this problem would be to cascade two first-order phase-lead filters. However, if noise in the control system is a problem, the increased gain at high frequencies may lead to noise problems. A different approach is to cascade a phase-lag filter with a phase-lead filter. This filter is usually referred to as a lag-lead compensator.



**FIGURE 8-16** Frequency response of a lag-lead filter.

A lag-lead filter has the characteristics shown in Fig. 8-16. The purpose of the lag section is to increase the low-frequency gain, and the lead section increases the bandwidth and the stability margins. The lag-lead filter will now be illustrated by an example.

**EXAMPLE 8.3**

In this example, the system of Examples 8.1 and 8.2 will again be considered. The steady-state error to a unit ramp input is given by

$$[eq. (6-21)] \quad e_{ss}(kT) = \frac{T}{\lim_{z \rightarrow 1} (z - 1)G(z)}$$

From Example 8.1,

$$\lim_{z \rightarrow 1} (z - 1)G(z) = \lim_{z \rightarrow 1} \frac{(z - 1)^2}{z} \left[ \frac{0.05z}{(z - 1)^2} - \frac{1.5z}{z - 1} + \frac{2z}{z - 0.9512} - \frac{0.5z}{z - 0.9048} \right] = 0.05$$

Hence, from (6-21),

$$e_{ss}(kT) = \frac{0.05}{0.05} = 1$$

Suppose that the design specifications require a steady-state error to a unit ramp input of 0.50 and a phase margin of 55°. We will use a phase-lag filter,  $D_1(z)$ , to increase the low-frequency gain by a factor of 2, to satisfy the steady-state error criterion. Then we design a phase-lead filter,  $D_2(z)$ , to yield the 55° phase margin. We will choose the pole-zero locations of  $D_1(z)$  to be the same as those of the phase-lag filter in Example 8.1. Then

$$a_0 = \lim_{z \rightarrow 1} D_1(z) = \lim_{z \rightarrow 1} \frac{K_d(z - 0.9982)}{z - 0.9993} = 2$$

From this expression we see that  $K_d = 0.7778$ , or

$$D_1(z) = \frac{0.7778(z - 0.9982)}{z - 0.9993}$$



To design the phase-lead filter, we must calculate the frequency response  $D_1(z)G(z)$ . The equations of Section 8.6 can then be utilized to find the phase-lead filter transfer function  $D_2(z)$ . We use  $\omega_{w1} = 1.20$ , as in Example 8.2. Calculation of the required frequency response yields

$$D_1(w)G(w)|_{w=j1.20} = 0.3557 \angle (-173.93)$$

We now substitute these values into (8-32) and (8-33), with  $G(j\omega_1)$  in these equations replaced with  $D(j\omega_1)G(j\omega_1)$  above. Hence, from (8-32),

$$\theta = 180^\circ + 55^\circ - (-173.93^\circ) = 408.93^\circ = 48.93^\circ$$

From (8-29) and (8-33), with  $a_0 = 1$ ,

$$a_1 = \frac{1}{\omega_{w0}} = \frac{1 - (1)(0.3557) \cos(48.93^\circ)}{(1.2)(0.3557) \sin(48.93^\circ)} = 2.3814 = \frac{1}{0.4199}$$

and from (8-33b),

$$b_1 = \frac{1}{\omega_{wp}} = \frac{\cos(48.93^\circ) - (1)(0.3557)}{(1.2) \sin(48.93^\circ)} = 0.3331 = \frac{1}{3.002}$$

Then  $D_2(w) = (1 + w/0.4199)/(1 + w/3.002)$ . Thus, from (8-14),

$$D_2(z) = \frac{6.7206(z - 0.9792)}{z - 0.8604}$$

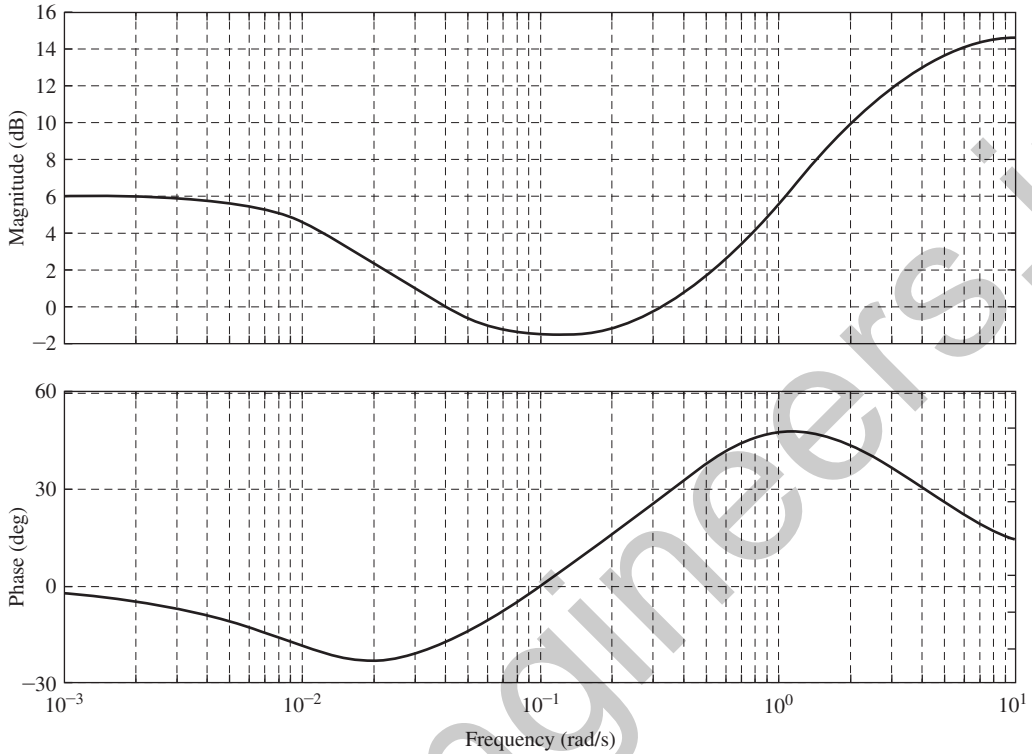
The total filter transfer function is

$$D(z) = D_1(z)D_2(z) = \frac{5.2273(z - 0.9982)(z - 0.9792)}{(z - 0.9993)(z - 0.8604)}$$

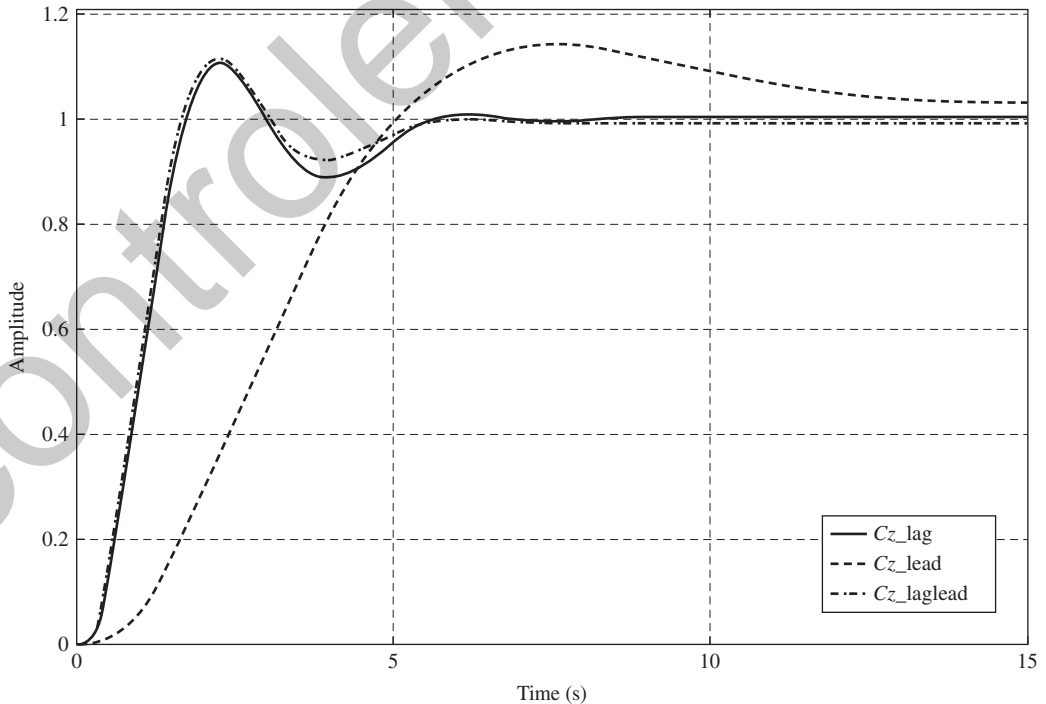
The filter's DC gain is 2.003. A sketch of the Bode diagram for this filter is shown in Fig. 8-17. Calculation of the compensated system open-loop frequency response shows that the compensated system has a phase margin of  $55^\circ$  and a gain margin of 11.2 dB. The step response for this system is plotted in Fig. 8-18, together with those from Examples 8.1 and 8.2. Note that the step responses of the phase-lead system and the lag-lead system are approximately the same; however, the steady-state error for a ramp input for the lag-lead system is only one-half that for the phase-lead system.

The calculations and plots in this example are implemented in this MATLAB program.

```
>> Gp = tf([2],[1 3 2 0]); T=0.05;
Gz = c2d(Gp,T); %Assumes ZOH;
% Set up lag filter
D1z = 0.7778*tf([1 -0.9982],[1 -0.9993],T);
D1z=zpk(D1z); D1zGz=D1z*Gz;
disp('Enter required design parameters: ')
a0_D2 = input('D2(z) filter dc gain a0_D2 = '); % enter 1
Pm = input('Desired system phase margin Pm = '); % enter 55
ww1_D2 = input('Phase margin frequency ww1_D2 = '); % enter 1.2
[magD1Gww1,phaseD1Gww1] = bode(D1zGz,ww1_D2)
thetad = 180+Pm-phaseD1Gww1; thetar = thetad*pi/180;
% Find lead filter D2(w) using equation (8-33)
disp('D2(w) = [a1_D2*w + a0_D2]/[b1_D2*w + 1] ')
a1_D2 = (1-a0_D2*magD1Gww1*cos(thetar))/(ww1_D2*magD1Gww1*sin(thetar))
```



**FIGURE 8-17** Frequency response for the lag-lead filter for Example 8.3.



**FIGURE 8-18** Step responses for Examples 8.1, 8.2, and 8.3.

```

b1_D2 = (cos(thetar)-a0_D2*magD1Gww1)/(ww1_D2*sin(thetar))
% Find lead filter D2(z) using equations 8-30 and 8-15
ww0_D2 = a0_D2/a1_D2; wwp_D2 = 1/b1_D2;
Kd_D2 = a0_D2*(wwp_D2*(ww0_D2+2/T))/(ww0_D2*(wwp_D2+2/T));
z0_D2= (2/T-ww0_D2)/(2/T+ww0_D2); zp_D2 = (2/T-wwp_D2)/(2/T+wwp_D2);
D2z = Kd_D2*tf([1 -z0_D2],[1 -zp_D2],T); D2z = zpk(D2z)
% Find the lead-lag filter D1(z)*D2(z)
Dz_laglead=D1z*D2z
Dz_lagleadGz = Dz_laglead*Gz;
% Find the stability margins
disp('Gain and Phase Margins:')
[Gm_laglead,Pm_laglead] = margin(Dz_lagleadGz);
GmdB_laglead = 20*log10(Gm_laglead), Pm_laglead
% Plot Figure 8-17: lag-lead filter frequency response
figure(1), bode(Dz_laglead,'-k'), grid
% plot Figure 8-18: Step responses for Examples 8.1-8.3
Dz_lag = tf([0.3890 -0.3883],[1 -0.9993],T);
Dz_lead = 6.5278*tf([1 -0.9711],[1 -0.8111],T);
Dz_lagGz = Dz_lag*Gz;
Dz_leadGz = Dz_lead*Gz;
Dz_lagleadGz = Dz_laglead*Gz;
Cz_uncomp = feedback(Gz,1);
Cz_lag = feedback(Dz_lagGz,1);
Cz_lead = feedback(Dz_leadGz,1);
Cz_laglead = feedback(Dz_lagleadGz,1);
figure(2), step(Cz_lag,'--k',Cz_lead,':k',Cz_laglead,'-k'),grid
stepinfo(Cz_laglead)
    
```

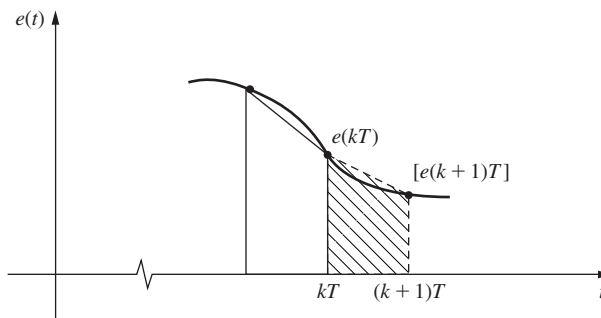
## 8.8 INTEGRATION AND DIFFERENTIATION FILTERS

A somewhat different controller is presented in this and the following section. This controller is used extensively in many different industries.

To introduce this controller, we first consider a technique for digital-filter integration. Suppose that we desire to integrate a signal  $e(t)$  digitally, and to accomplish this, we will use the trapezoidal technique [6]. The trapezoidal rule is illustrated in Fig. 8-19.

Let  $m(kT)$  be the numerical integral of  $e(t)$ . Then, from Fig. 8-19, the value of the integral at  $t = (k + 1)T$  is equal to the value at  $kT$  plus the area added from  $kT$  to  $(k + 1)T$ . From Fig. 8-19,

$$m[(k + 1)T] = m(kT) + \frac{T}{2}\{e[(k + 1)T] + e(kT)\} \quad (8-34)$$



**FIGURE 8-19** Trapezoidal rule for numerical integration.

Taking the  $z$ -transform, we obtain

$$zM(z) = M(z) + \frac{T}{2}[zE(z) + E(z)] \quad (8-35)$$

Thus

$$\frac{M(z)}{E(z)} = \frac{T}{2} \left[ \frac{z+1}{z-1} \right] \quad (8-36)$$

Hence (8-36) is the transfer function of a discrete integrator. Of course, there are many other discrete transfer functions that may be used to integrate a number sequence as shown in Section 6.6.

Now consider a technique for the digital-filter differentiation of a function  $e(t)$ . Figure 8-20 illustrates one method. The slope of  $e(t)$  at  $t = kT$  is approximated to be the slope of the straight line connecting  $e[(k-1)T]$  with  $e(kT)$ . Then, from Fig. 8-20, letting the numerical derivative of  $e(t)$  at  $t = kT$  be  $m(kT)$ , we can write

$$m(kT) = \frac{e(kT) - e[(k-1)T]}{T} \quad (8-37)$$

The  $z$ -transform of this equation yields

$$\frac{M(z)}{E(z)} = \frac{z-1}{Tz} \quad (8-38)$$

Note that this differentiator is the reciprocal of the transfer function for the rectangular-rule integrator (see Problem 2.2-1). In general, the reciprocal of the transfer function of a numerical integrator can be used as the transfer function of a numerical differentiator. For the development in this section, we use (8-36) and the reciprocal of this transfer function for integration and differentiation, respectively. Then, from (8-36), letting  $D_I(z)$  be the integrator transfer function, we obtain

$$D_I(w) = \frac{T}{2} \left( \frac{z+1}{z-1} \right) \Bigg|_{z=[1+(T/2)w]/[1-(T/2)w]} = \frac{1}{w} \quad (8-39)$$

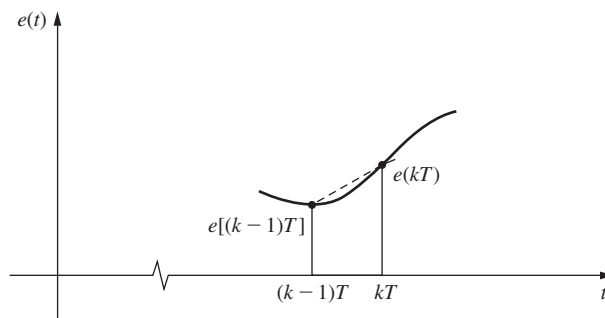


FIGURE 8-20 Illustration of numerical differentiation.

Letting  $D_D(z)$ , the differentiator transfer function, be the reciprocal of (8-39),

$$D_D(w) = D_D(z) \Big|_{z=[1+(T/2)w]/[1-(T/2)w]} = w \quad (8-40)$$

Recall that for continuous systems, a differentiator has a transfer function of  $s$ , and an integrator a transfer function  $1/s$ . The frequency responses are obtained by replacing  $s$  with  $j\omega$ .

The frequency responses of the discrete transfer functions (8-39) and (8-40) will now be compared with those of the continuous integrator and differentiator. From (7-10),

$$\omega_w = \frac{2}{T} \tan \frac{\omega T}{2} \quad (8-41)$$

For  $\omega T/2$  small, we see that

$$\omega_w \approx \omega \quad (8-42)$$

For the integrator transfer function (8-39),

$$D_I(j\omega_w) \approx \frac{1}{j\omega_w} \quad (8-43)$$

Substituting (8-42) into (8-43), we obtain

$$D_I(j\omega) \approx \frac{1}{j\omega} \quad (8-44)$$

In a similar manner,

$$D_D(j\omega_w) = j\omega_w \approx j\omega$$

These approximations are good provided that since  $\omega_s = 2\pi/T$ ,

$$\tan\left(\frac{T}{2}\omega\right) = \tan\left(\pi\frac{\omega}{\omega_s}\right) \approx \pi\frac{\omega}{\omega_s}$$

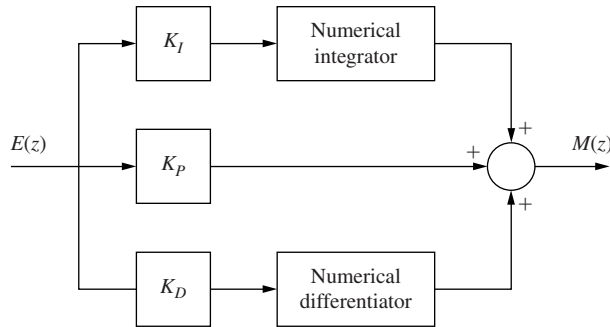
If this expression is satisfied, we would expect to obtain accurate differentiation and integration for (8-40) and (8-39), respectively.

## 8.9 PID CONTROLLERS

We will now discuss a frequency-response design technique that considers phase-lead phase-lag controllers from a somewhat different viewpoint. The resultant controller, called a proportional-plus-integral-plus-derivative (PID) controller, has the block diagram shown in Fig. 8-21. This controller is a special type of the lag-lead controller.

The transfer function of a digital PID controller, using the integrator and differentiator transfer functions developed in the preceding section, is given by

$$D(w) = K_P + \frac{K_I}{w} + K_D w \quad (8-45)$$



**FIGURE 8-21** Discrete PID controller.

In this expression,  $K_P$  is the gain in the proportional path,  $K_I$  the gain in the integral path, and  $K_D$  the gain in the derivative path.

Consider first a proportional-plus-integral (PI) controller. The filter transfer function is

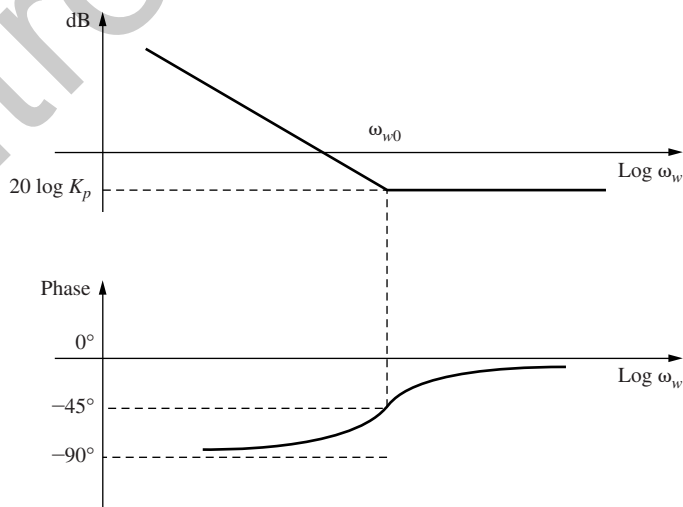
$$D(w) = K_P + \frac{K_I}{w} = \frac{K_P w + K_I}{w} = K_I \frac{1 + w/\omega_{w0}}{w} \quad (8-46)$$

where  $\omega_{w0} = K_I/K_P$ . Note that this is a phase-lag filter of the type given in (8-13), with the pole placed at  $\omega_{wp} = 0$ . The filter frequency response is illustrated in Fig. 8-22.

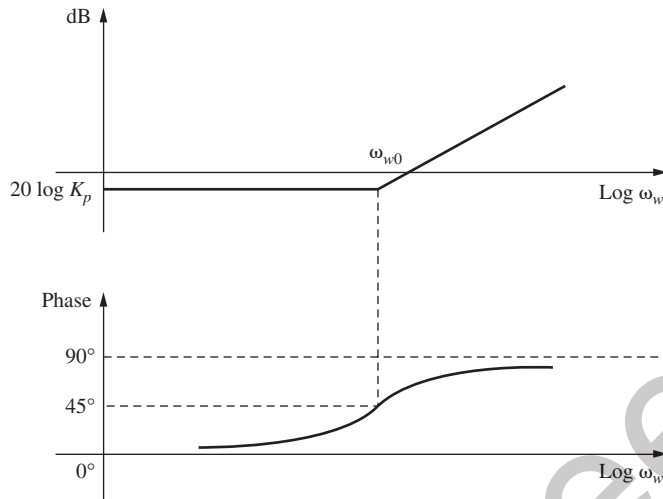
Note that the PI controller low-frequency gain is increased (infinite at  $\omega_w = 0$ ), and thus steady-state errors are reduced. The purpose is then the same as that of the phase-lag controller of (8-13): to increase stability margins and/or reduce steady-state errors.

Consider next the PD (proportional-plus-derivative) controller. The filter transfer function is

$$D(w) = K_P + K_D w = K_P \left( 1 + \frac{w}{\omega_{w0}} \right) \quad (8-47)$$



**FIGURE 8-22** Frequency response for a PI controller.



**FIGURE 8-23** Frequency response for a PD controller.

where  $\omega_{w0} = K_p/K_D$ . Note that this is a phase-lead controller of the type given in (8-13), with the pole placed at  $\omega_{wp} = \infty$ . The filter frequency response is illustrated in Fig. 8-23. The purposes of the PD controller are to add positive phase angles to the open-loop frequency response so as to improve system stability, and to increase closed-loop system bandwidth so as to increase the speed of response. The effects of the PD controller appear at high frequencies, as opposed to the low-frequency effects of the PI controller.

The PID filter is a composite of the two filters discussed above and has the transfer function

$$D(w) = K_p + \frac{K_I}{w} + K_D w \quad (8-48)$$

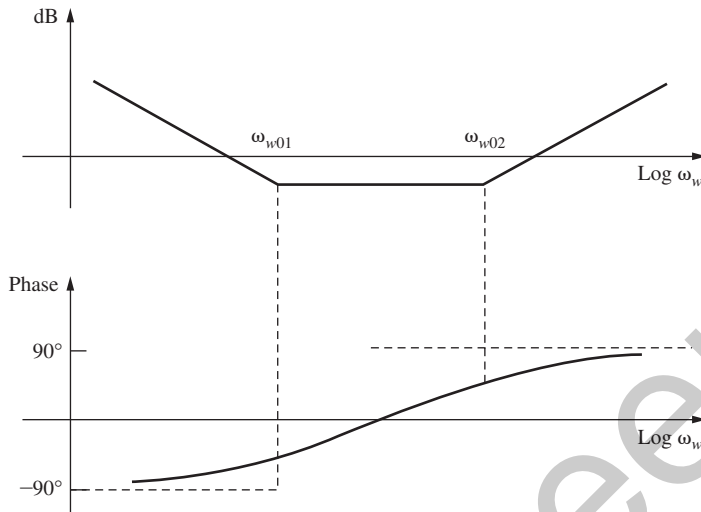
The frequency response of the PID controller is illustrated in Fig. 8-24. From (8-48),

$$D(w) = \frac{K_D w^2 + K_p w + K_I}{w} = \frac{K_I (1 + w/\omega_{w01})(1 + w/\omega_{w02})}{w}$$

Hence the filter has two zeros,  $\omega_{w01}$  and  $\omega_{w02}$ , and two poles, with one at  $\omega_w = 0$  and the other at  $\omega_w = \infty$ . The design techniques can follow those given for phase-lag (the PI part) and phase-lead (the PD part) filters. However, a different design procedure will be developed in the following section. Since the  $K_p$  term is common to both the PI and PD parts, the design of the PD part affects the PI part, and vice versa.

A problem is obvious in the use of the PID filter. As is seen in Fig. 8-24, the filter gain continues to increase without limit as frequency increases. As  $\omega_w$  approaches infinity,  $\omega$  approaches  $\omega_s/2$  and  $z$  approaches  $-1$ . In general,  $G(z)$  does not have a zero at  $z = -1$ . Thus  $|G(z)D(z)|$  will approach infinity as  $z$  approaches  $-1$ , which will probably lead to encirclements of the  $-1$  point on the system Nyquist diagram. Hence the system will be unstable, and the compensator of (8-48) is not practical. To avert this problem, a pole (or poles) must be added to the derivative term, resulting in the filter transfer function

$$D(w) = K_p + \frac{K_I}{w} + \frac{K_D w}{\prod (1 + w/\omega_{wp_i})} \quad (8-49)$$



**FIGURE 8-24** Frequency response for a PID controller.

The usual solution to this problem is to add a single pole using the differentiator of Fig. 8-20, which has the transfer function given in (8-38).

$$D_D(w) = \frac{z - 1}{Tz} \bigg|_{z=[1+(T/2)w]/[1-(T/2)w]} = \frac{w}{1 + (T/2)w} = \frac{w}{1 + (w/\omega_{wp})} \quad (8-50)$$

Hence this algorithm for differentiation adds a pole at  $2/T$ ; and since  $\omega_s = 2\pi/T$ , then

$$\omega_{wp} = \frac{2}{T} = \frac{\omega_s}{\pi}$$

This pole position is normally well outside the bandwidth of the system and will contribute very little phase lag at the frequency that the phase margin occurs. Hence in many cases this pole can be ignored (at least initially) in the design process. If high-frequency noise is a problem in the control system, additional poles may be required in the derivative term.

A practical PID filter transfer function as a function of  $z$  is, from (8-45), (8-36), and (8-38),

$$D(z) = K_P + K_I T \left[ \frac{z + 1}{2(z - 1)} \right] + K_D \left[ \frac{z - 1}{Tz} \right] \quad (8-51)$$

which may be reformulated as

$$D(z) = K_d \frac{z^2 + az + b}{z(z - 1)}$$

$$K_d = K_P + K_I T/2 + K_D/T$$

$$a = (K_I T/2 - K_P - 2K_D/T)/K_d$$

$$b = K_D/(TK_d) \quad (8-52)$$



This filter uses trapezoidal integration and the differentiation of (8-38) and Fig. 8-20. The filter difference equations can be written directly from this equation (see Problem 8.9-1). In some cases it is possible to determine  $K_p$ ,  $K_I$ , and  $K_D$  experimentally using the physical control system, when only a rudimentary knowledge of the plant characteristics is available. An educated guess of  $K_p$ ,  $K_I$ , and  $K_D$  is made, which must result in a stable closed-loop system. Then  $K_p$ ,  $K_I$  and  $K_D$  are varied in some systematic manner until an acceptable response is obtained. A somewhat different form of the PID controller transfer function is considered in Problem 8-9-2.

The PID controller offers an advantage if, for some reason, the sample period  $T$  must be changed after the system design has been completed. Suppose that for a given design,  $T$  is chosen such that accurate differentiation and integration occur. If for a different value of  $T$ , accurate differentiation and integration still occur, the gains  $K_p$ ,  $K_I$ , and  $K_D$  will not change. Hence a new design is not required. A design procedure for PID controllers will be developed in the following section.

## 8.10 PID CONTROLLER DESIGN

In this section a design process is developed for PID controllers [6]. The development will closely parallel that of Section 8.6 for phase-lead filters (controllers). We assume initially that the PID controller has a transfer function given by

$$\text{[eq. (8-45)]} \quad D(w) = K_p + \frac{K_I}{w} + K_D w$$

Hence we are ignoring the effects of the required pole of (8-50) in the derivative path; these effects will be considered later. The controller frequency response is given by

$$D(j\omega_w) = K_p + j(K_D\omega_w - K_I/\omega_w) = |D(j\omega_w)|\varepsilon^{j\theta} \quad (8-53)$$

Here, as in Section 8.6, the design problem is to choose  $D(w)$  (i.e., choose  $K_p$ ,  $K_I$ , and  $K_D$ ) such that

$$D(j\omega_{w1})G(j\omega_{w1}) = 1 \angle (-180^\circ + \phi_m) \quad (8-54)$$

at a chosen frequency  $\omega_{w1}$ . Now, from (8-53),

$$K_p + j\left(K_D\omega_{w1} - \frac{K_I}{\omega_{w1}}\right) = |D(j\omega_{w1})|(\cos\theta + j\sin\theta) \quad (8-55)$$

where from (8-53) and (8-54),

$$\theta = -180^\circ + \phi_m - \angle G(j\omega_{w1}) \quad (8-56)$$

Therefore, from (8-54) and (8-55),

$$K_p = |D(j\omega_{w1})| \cos\theta = \frac{\cos\theta}{|G(j\omega_{w1})|} \quad (8-57)$$

$$K_D\omega_{w1} - \frac{K_I}{\omega_{w1}} = \frac{\sin\theta}{|G(j\omega_{w1})|} \quad (8-58)$$

In these design equations, if the transfer function of the sensor  $H(s)$  is not unity,  $G(j\omega_{w1})$  is replaced with  $\overline{GH}(j\omega_{w1})$ .

The design equations are then (8-56), (8-57), and (8-58). For a given plant,  $G(w)$ , the choice of  $\omega_{w1}$  and  $\phi_m$  uniquely determines  $K_p$ , from (8-57). However,  $K_D$  and  $K_I$  are not uniquely determined, as is evident in (8-58). This results from (8-55) yielding two equations, but with the three unknowns  $K_p$ ,  $K_I$ , and  $K_D$ . In satisfying (8-58), in general increasing  $K_D$  will increase the bandwidth, while increasing  $K_I$  will decrease steady-state errors, *provided* that the system retains an acceptable gain margin. Note that if (8-57) and (8-58) are satisfied, varying  $K_D$  and  $K_I$  will change the gain margin, while the phase margin remains constant.

Equations (8-57) and (8-58) also apply to the design of PI and PD controllers, with the appropriate gain ( $K_D$  or  $K_I$ ) set to zero. For this case all gains are uniquely determined.

As shown in Section 8.9, in general a pole is required in the derivative term. A commonly used transfer function was given in (8-50), which results in a PID controller transfer function

$$D(w) = K_p + \frac{K_I}{w} + \frac{K_D w}{1 + (T/2)w} \quad (8-59)$$

Then

$$\begin{aligned} D(j\omega_w) &= K_p - j\frac{K_I}{\omega_w} + \frac{K_D j\omega_w}{1 + j\omega_w T/2} \\ &= \left( K_p + \frac{K_D \omega_w^2 (2/T)}{(2/T)^2 + \omega_w^2} \right) + j \left( \frac{K_D \omega_w (2/T)^2}{(2/T)^2 + \omega_w^2} - \frac{K_I}{\omega_w} \right) \\ &= \frac{\cos\theta + j\sin\theta}{|G(j\omega_{w1})|} \end{aligned} \quad (8-60)$$

By equating the real and imaginary terms, (8-57) and (8-58) become

$$K_p + \frac{K_D \omega_{w1}^2 (2/T)}{(2/T)^2 + \omega_{w1}^2} = \frac{\cos\theta}{|G(j\omega_{w1})|} \quad (8-61)$$

and

$$\frac{K_D \omega_{w1} (2/T)^2}{(2/T)^2 + \omega_{w1}^2} - \frac{K_I}{\omega_{w1}} = \frac{\sin\theta}{|G(j\omega_{w1})|} \quad (8-62)$$

Note that if  $\omega_{w1} < 2/T$ , these equations reduce to (8-57) and (8-58); otherwise, no simple procedure has been found for calculating  $K_p$ ,  $K_I$ , and  $K_D$ . For the PI controller, (8-61) and (8-62) contain only two unknowns and may be solved directly

$$K_p = \frac{\cos\theta}{|G(j\omega_{w1})|} \quad K_I = -\frac{\omega_{w1} \sin\theta}{|G(j\omega_{w1})|} \quad (8-63)$$

For the PD controller, (8-62) and (8-61) also contain only two unknowns and may be solved directly for

$$K_D = \left[ \frac{\sin \theta}{|G(j\omega_{w1})|} \right] \left[ \frac{(2/T)^2 + \omega_{w1}^2}{(2/T)^2 \omega_{w1}} \right] \quad K_P = \frac{\cos \theta}{|G(j\omega_{w1})|} - \frac{K_D \omega_{w1}^2 (2/T)}{(2/T)^2 + \omega_{w1}^2} \quad (8-64)$$

Two examples of PID design will now be given.

### EXAMPLE 8.4

First the design problem of Example 8.3 will be repeated, but a PI filter will be utilized. In Example 8.3 we required a  $55^\circ$  phase margin and a steady-state error to a unit ramp input of 0.5. Since  $D(z)$  adds a pole at  $z = 1$  to the one already present in  $G(z)$ , then  $D(z)G(z)$  has two poles at  $z = 1$ . Thus the steady-state error to a ramp input is zero (see Section 6.5), satisfying the steady-state error design criteria. From Table 8-1, the frequency response of  $G(z)$ , we choose  $\omega_{w1} = 0.400$ . Then

$$G(j\omega_{w1}) = G(j0.4) = 2.2761 \angle (-123.68^\circ)$$

From (8-56),

$$\theta = 180^\circ + 55^\circ - (-123.68^\circ) = 358.68^\circ = -1.32^\circ$$

We see then that the phase angle of the filter is very small at the phase-margin frequency, as desired. Then, from (8-63),

$$K_P = \frac{\cos(-1.32^\circ)}{2.2768} = 0.4392$$

$$K_I = \left[ \frac{-\sin(-1.3^\circ)}{2.278} \right] 0.4 = 0.0040$$

From (8-52), the PI controller is

$$D(z) = \frac{0.4393z - 0.4391}{z - 1}$$

From (8-46), the zero of the PI controller is placed at

$$\omega_{w0} = \frac{K_I}{K_P} = 0.0092$$

which is quite low. Hence we have reduced the system bandwidth considerably. Even though we get good steady-state error response, a relatively long time will be required to achieve it. The gain margin is 16.0 dB and the phase margin is  $55^\circ$ . The unit-step response displays the rise time (2.95 s), overshoot (13.8%), and settling time (19.4 s). The system responds to a unit ramp with an error of 1.99 s after 20 s, and has decreased to only 1.52 s at the end of 50 s. If we choose  $\omega_{w1} = 0.3$ , the PI gains are calculated as  $K_P = 0.3125$  and  $K_I = 0.0154$ . Then

$$D(z) = \frac{0.3129z - 0.3121}{z - 1}$$

and

$$\omega_{w0} = \frac{K_I}{K_P} = 0.0493$$

and the system bandwidth has been increased. For this PI controller, the gain margin is increased to 18.4 dB, the phase margin is  $55^\circ$ , the unit-step rise time is delayed to 3.65 s, the overshoot is increased to 18.6%, and the settling time is delayed to 43.0 s. The error to a unit-ramp input is reduced to 1.37 s after 20 s, and has decreased to 0.27 s at the end of 50 s.

A MATLAB program that performs the above calculations is given by

```
>> Gp = tf([2],[1 3 2 0]); T=0.05;
Gz = c2d(Gp,T); %Assumes ZOH;
Pm = input('Desired system phase margin Pm = '); % enter 55
ww1 = input('Phase margin frequency ww1 = '); % enter 0.4 & 0.3
[magGww1,phaseGww1] = bode(Gz,ww1)
thetad = 180+Pm-phaseGww1; thetar = thetad*pi/180;
KD = 0;
% Equation (8-63)
KP = cos(thetar)/magGww1, KI = -ww1*sin(thetar)/magGww1
ww0=KI/KP
% Equation (8-52)
Dz = KP+(KI*T/2)*tf([1 1],[1 -1],T)+(KD/T)*tf([1 -1],[1 0],T)
DzGz = Dz*Gz;
Cz=feedback(DzGz,1);
disp('Gain and Phase Margins:')
[Gm,Pm] = margin(DzGz); GmDB = 20*log10(Gm), Pm
disp('Unit-step performance metrics:')
stepinfo(Cz)
step2ramp = tf([T],[1 -1],T); % Converts unit-step to unit-ramp
ct = step(step2ramp*Cz,60); % Generates c(t)
disp('Unit-ramp errors at 20 s and 50 s:')
ct20err = 20-ct(20/T) % error = 20 - c(20)
ct50err = 50-ct(50/T) % error = 50 - c(50)
```

### EXAMPLE 8.5

The design problem of Example 8.4 will be repeated, except that in this example a PID controller will be utilized. The design equations (8-57) and (8-58) will be used (i.e., initially the pole in the derivative term will be ignored). From (8-57),

$$K_p = \frac{\cos \theta}{|G(j\omega_{w1})|}$$

Thus the larger we choose  $\omega_{w1}$ , the smaller is  $|G(j\omega_{w1})|$  (see Table 8-1), and hence the larger is  $K_p$ . This increases the open-loop gain, which is desirable for many reasons. We choose  $\omega_{w1} = 1.2$ . Then

$$G(j\omega_{w1}) = G(j1.2) = 0.4574 \angle (-172.88^\circ)$$

From (8-56),

$$\theta = 180^\circ + 55^\circ - (-172.88^\circ) = 407.88^\circ = 47.88^\circ$$

and from (8-57),

$$K_p = \frac{\cos(47.88^\circ)}{0.4574} = 1.4664$$

From (8-58),

$$K_D \omega_{w1} - \frac{K_I}{\omega_{w1}} = \frac{\sin\theta}{|G(j\omega_{w1})|}$$

Hence, with  $G(j\omega_{w1})$  from above,

$$\begin{aligned} K_D &= \left[ \frac{\sin(47.88^\circ)}{0.4574} \right] \frac{1}{1.2} + K_I \left( \frac{1}{1.2} \right)^2 \\ &= 1.3513 + 0.6944K_I \end{aligned}$$

If  $K_I$  is chosen as 0.004 as was the case in Example 8.4, then  $K_D$  is 1.3541. From (8-52), the resulting PID controller is

$$D(z) = \frac{28.55z^2 - 55.63z - 27.08}{z^2 - z}$$

This configuration produces a phase margin of  $53.7^\circ$  and a gain margin of 23.2 dB. The unit-step response displays the rise time (1.00 s), overshoot (14.3%), and settling time (3.60 s). The system responds to a unit ramp with an error of 0.699 s after 20 s.

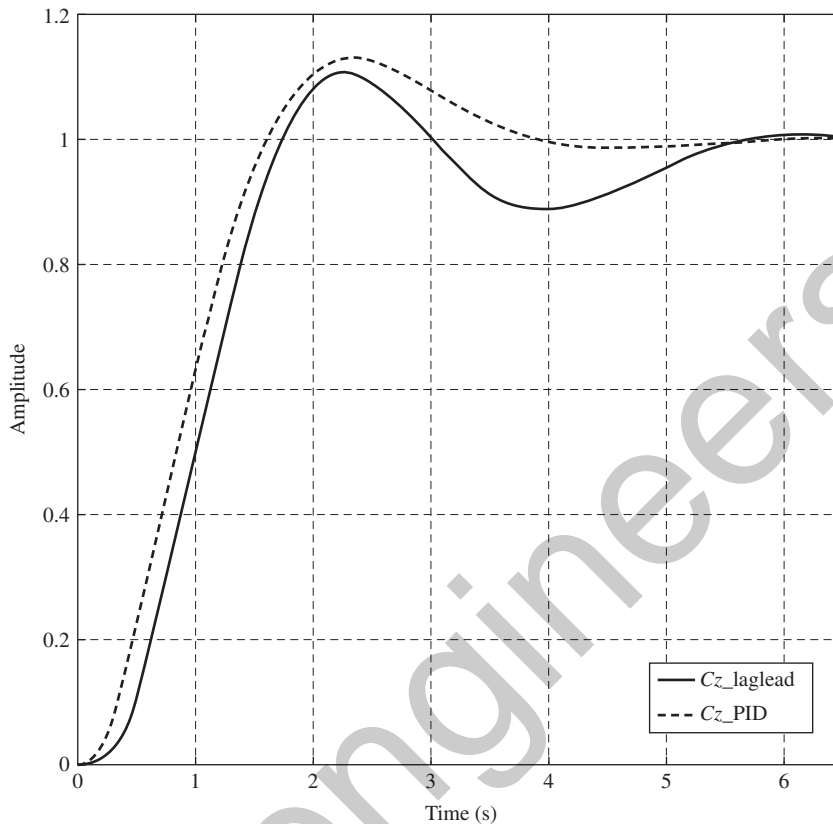
Note the error on the achieved phase margin when we neglect the high frequency pole of the practical derivative filter in (8-52) and employ (8-57) and (8-58) to compute the PID gains. The desired phase margin can be achieved by using (8-61) and (8-62) to compute the PID gains. With  $K_I = 0.004$  (as determined in Example 8.4) inserted into (8-61) and (8-62),  $K_D = 1.3553$ ,  $K_P = 1.4177$ , and the resulting PID controller is

$$D(z) = \frac{28.52z^2 - 55.63z - 27.11}{z^2 - z}$$

This controller with slightly modified numerator coefficients produces the desired phase margin of  $55^\circ$  and a gain margin of 23.3 dB. The unit-step response displays the rise time (1.05 s), overshoot (13.0%), and settling time (3.60 s). After 20 s, the system responds to a unit ramp with an error of 0.720 s.

A MATLAB program that performs the computations in this example is now given. The code is set for entering  $K_I$  and employing (8-61) and (8-62) to compute  $K_D$  and  $K_P$ .

```
>> Gp = tf([2],[1 3 2 0]); T=0.05;
Gz = c2d(Gp,T); %Assumes ZOH;
Pm = input('Desired system phase margin Pm = '); % enter 55
ww1 = input('Phase margin frequency ww1 = '); % enter 1.2
KI = input('Code is to set to enter KI= '); % enter 0.004
[magGww1,phaseGww1] = bode(Gz,ww1)
thetad = 180+Pm-phaseGww1; thetar = thetad*pi/180
% disp('Case (a): Use Equations (8-57) and (8-58)')
% KP = cos(thetar)/magGww1
% KD = (KI/ww1 + sin(thetar)/magGww1)/ww1
disp('Case (b): Use Equations (8-61) and (8-62)')
disp('Option (1): Enter KI, compute KD and KP')
KD = (sin(thetar)/magGww1/ww1 + KI/ww1^2)*(1 + ww1^2/(2/T)^2)
KP = cos(thetar)/magGww1 - KD*ww1^2/(2/T+ww1^2*T/2)
% disp('Option (2): Enter KD, compute KP and KI')
% KP = cos(thetar)/magGww1 - (KD*ww1^2*(2/T))/((2/T)^2+ww1^2)
```



**FIGURE 8-25** Responses of Examples 8.3 and 8.5.

```
% KI = ww1*((KD*ww1*(2/T)^2)/((2/T)^2+ww1^2) - sin(thetar)/magGww1)
% Find Dz using Equation (8-51)
Dz_PID = KP + (KI*T/2)*tf([1 1],[1 -1],T) + (KD/T)*tf([1 -1],[1 0],T)
Dz_PIDGz=Dz_PID*Gz; Cz=feedback(Dz_PIDGz,1);
disp('Gain and Phase Margins:')
[Gm,Pm] = margin(Dz_PIDGz); GmDB = 20*log10(Gm), Pm
disp('Unit-step performance metrics:')
stepinfo(Cz), step(Cz)
step2ramp = tf([T],[1 -1],T); % Converts unit-step to unit-ramp
ct = step(step2ramp*Cz,60); % Generates ramp response c(t)
disp('Unit-ramp error at 20 s:')
ct20err = 20-ct(20/T) % error = 20 - c(20)
```

The step response of the PID controller is shown in Fig. 8-25, together with the step response of the lag-lead controller of Example 8.3. For the lag-lead controller the rise time is 1.05 s; overshoot, 10.8%; and settling time, 5.30 s. For the PID controller, the rise time is 1.05 s; the overshoot, 13.0%; and the settling time, 3.60 s.

MATLAB includes a PID design tool (named *pidtool*) to automate the computation of the control gains. The format for one of MATLAB's PID controllers (named PIDF) is as follows.

$$D(z) = K_P + K_I \left[ \frac{T}{z-1} \right] + K_D \left[ \frac{1}{T_F + \frac{T}{z-1}} \right] \quad (8-65)$$

where  $T_F$  is a parameter that places a high frequency pole to improve performance. Remember that the derivative filter in (8-51) places a high frequency pole at a specific location ( $\omega_{wp} = 2/T$ ). Equation (8-65) can be reformulated into the same format as was used in (8-51).

$$D(z) = K_P + K_I \left[ \frac{T}{z-1} \right] + K_D \left[ \frac{1}{T_F} \left( \frac{z-1}{z-a} \right) \right], \quad a = \frac{T_F - T}{T_F} \quad (8-66)$$

The high frequency pole for this controller is given by

$$\omega_{wp} = \frac{2}{T} \left( \frac{1-a}{1+a} \right)$$

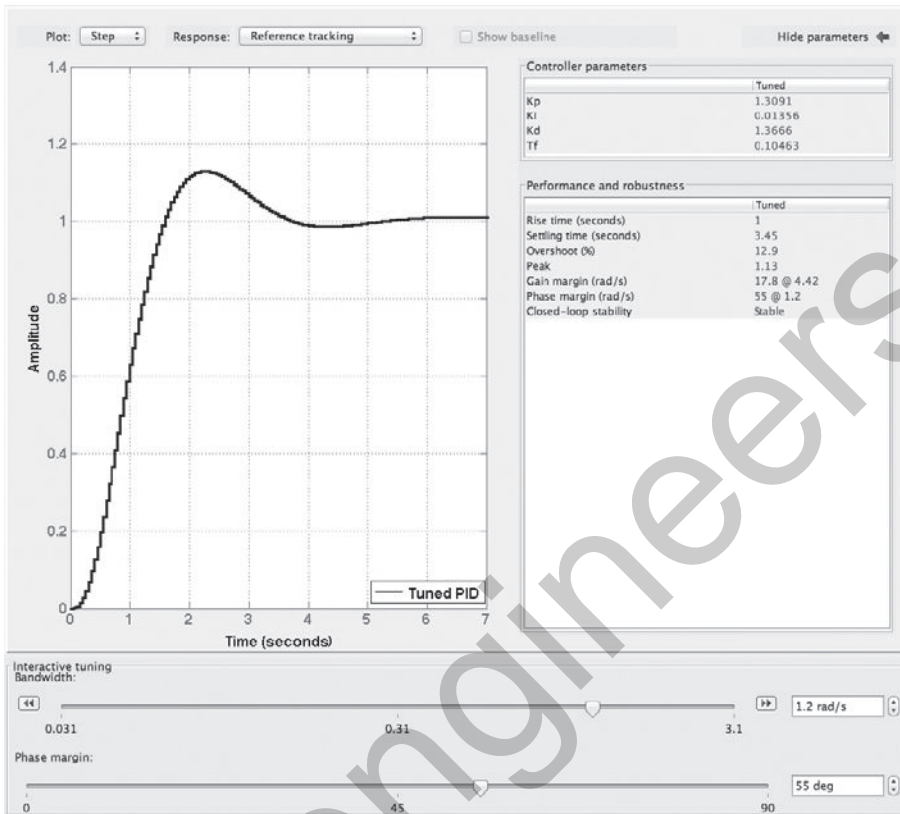
Note that setting  $T_F = T$  makes  $a = 0$  and (8-65) and (8-66) simplify to (8-51). If  $T_F = 2T$ ,  $\omega_{wp} = 2/(3T)$ , reducing the frequency breakpoint by 33.3%. The use of the MATLAB tool is illustrated in the following example.

### EXAMPLE 8.6

The design problem of Example 8.5 will be repeated, except that in this example a PID controller will be designed using the MATLAB *pidtool*. The first step is to generate in MATLAB the z-transform of the plant (labeled  $G_z$  in the previous example). Next, enter the command *pidtool(Gz)*. This opens a graphical user interface (GUI) with several options as shown in Fig. 8-26. One set of parameters employs (8-66) to calculate (or tune) the PID gains. Choose the following set:

- Design Mode: *Extended*
- Form: *Parallel*
- Type: *PIDF*
- Plot: *Step*
- Response: *Reference tracking*

Next click the arrow at the upper-right edge of the window labeled *Show parameters*. Now we are ready to input the controller specifications using the two interactive-tuning sliders at the bottom of the window labeled *Bandwidth* and *Phase margin*. In this case, we set the phase margin to 55. We can now set (using the bandwidth slider) the frequency  $\omega_{w1}$  at which we want to achieve the specified phase margin. MATLAB's *pidtool* displays the resulting constants ( $K_P$ ,  $K_I$ ,  $K_D$ , and  $T_F$ ) in the *Controller parameters* display area. We can also note the resulting step and frequency response parameters in the *Performance and robustness* display area. Entering our desired phase-margin frequency ( $\omega_{w1} = 1.2$  rad/s) using the bandwidth slider allows



**FIGURE 8-26** Screenshot of MATLAB *pidtool*. Source: Reprinted with permission from The MathWorks, Inc.

*pidtool* to display the following set of controller parameters (see Fig. 8-26):  $K_P = 1.3091$ ,  $K_I = 0.01356$ ,  $K_D = 1.3666$ , and  $T_F = 0.10463$ . Inserting these values into (8-65) produces

$$D(z) = \frac{14.37(z - 0.9995)(z - 0.9569)}{(z - 1)(z - 0.5221)}$$

The resulting closed-loop performance parameters are rise time, 1.0 s; settling time, 3.45 s; overshoot, 12.9%; gain margin, 17.8 dB; and phase margin,  $55^\circ$ . The following MATLAB program computes the designed PID controller transfer function and checks its performance parameters:

```
>> T = 0.05; Gp = tf([2],[1 3 2 0]); Gz = c2d(Gp,T);
disp(' ')
disp('Now opening MATLAB pidtool GUI')
disp('Set Pm = 55, wwl = 1.2')
pidtool(Gz,'pidf')
% Gains from MATLAB pidtool with Pm = 55, wwl = 1.2
KP = 1.3091, KI = 0.01356, KD = 1.3666, TF = 0.10463
% Find Dz using Equation (8-65)
disp('PIDF using Equation (8-65)')
Dz_PIDF = KP + (KI*T)*tf([1],[1 -1],T) + KD*tf([1 -1],[TF -TF+T],T)
Dz_PIDF = zpk(Dz_PIDF)
```



```
Dz_PIDFGz = Dz_PIDF*Gz;
Cz_PIDF = feedback(Dz_PIDFGz,1);
disp('PIDF Gain and Phase Margin:')
[Gm,Pm] = margin(Dz_PIDFGz); Gm dB = 20*log10(Gm), Pm
% disp('Unit-step performance metrics:')
PIDF_stepinfo = stepinfo(Cz_PIDF)
```

The MATLAB *pidtool* allows us to hold the phase margin at 55° and explore various settings for our phase-margin frequency design point. Increasing  $\omega_{w1}$  from 1.2 rad/s to 1.3 rad/s has very little effect on the rise time (decreases from 1.0 s to 0.95 s), overshoot (decreases from 12.9% to 12.6%), and gain margin (decreases from 17.8 dB to 17.5 dB). However, the settling time begins to increase more quickly (from 3.45 s to 4.15 s). An additional increase in  $\omega_{w1}$  to 1.35 rad/s decreases the rise time to 0.90 s, increases the overshoot to 15.2%, decreases the gain margin to 17.1 dB, and dramatically increases the settling time to 14.8 s.

## 8.11 DESIGN BY ROOT LOCUS

In the frequency-response design procedures described above, we attempted to reshape the system open-loop frequency response to achieve certain stability margins, transient-response characteristics, steady-state response characteristics, and so on. Even though design equations were developed, the techniques are still largely trial and error.

A different design technique is presented in this section: the root-locus procedure. Recall that the root locus for a system is a plot of the roots of the system's characteristic equation as gain is varied. Hence the character of the transient response of a system is evident from the root locus. The design procedure is to add poles and zeros via a digital controller so as to shift the roots of the characteristic equation to more appropriate locations in the  $z$ -plane.

Consider again the system of Fig. 8-6, which has the characteristic equation

$$1 + KD(z)G(z) = 0 \tag{8-67}$$

where  $K$  is the added gain that is to be varied to generate the root locus. Then a point  $z_a$  is on the root locus provided that (8-67) is satisfied for  $z = z_a$ , or that

$$K = \frac{1}{|D(z_a)G(z_a)|} \tag{8-68}$$

$$\angle(D(z_a)G(z_a)) = \pm 180^\circ \tag{8-69}$$

Since we allow  $K$  to vary from zero to infinity, a value of  $K$  will always exist such that (8-68) is satisfied. Then the condition for a point  $z_a$  to be on the root locus is simply (8-69). If the sensor transfer function  $H(s)$  is not unity,  $G(z)$  in the foregoing equations is replaced with  $\overline{GH}(z)$ .

Suppose, as an example, that  $D(z) = 1$  and  $KG(z)$  is given by

$$KG(z) = \frac{K(z - z_1)}{(z - z_2)(z - z_3)} \tag{8-70}$$

where  $z_1$ ,  $z_2$ , and  $z_3$  are all real. Figure 8-27 illustrates the testing of point  $z$  to determine if it is on the root locus.

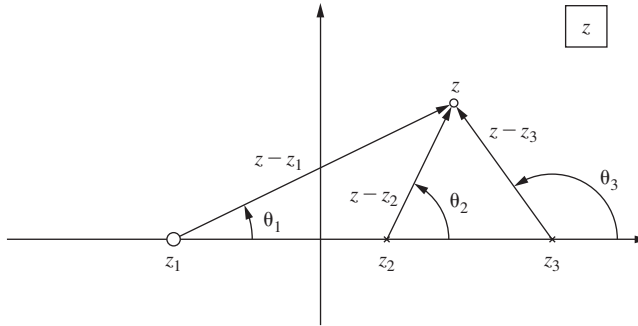


FIGURE 8-27 Point on the root locus.

If  $z$  is on the root locus, then, from (8-69),

$$\theta_1 - \theta_2 - \theta_3 = \pm 180^\circ$$

The value of  $K$  that places a root of the characteristic equation at this point is, from (8-68),

$$K = \frac{|z - z_2| |z - z_3|}{|z - z_1|}$$

With this brief discussion of the root locus, we will now consider design procedures. Consider the first-order controller

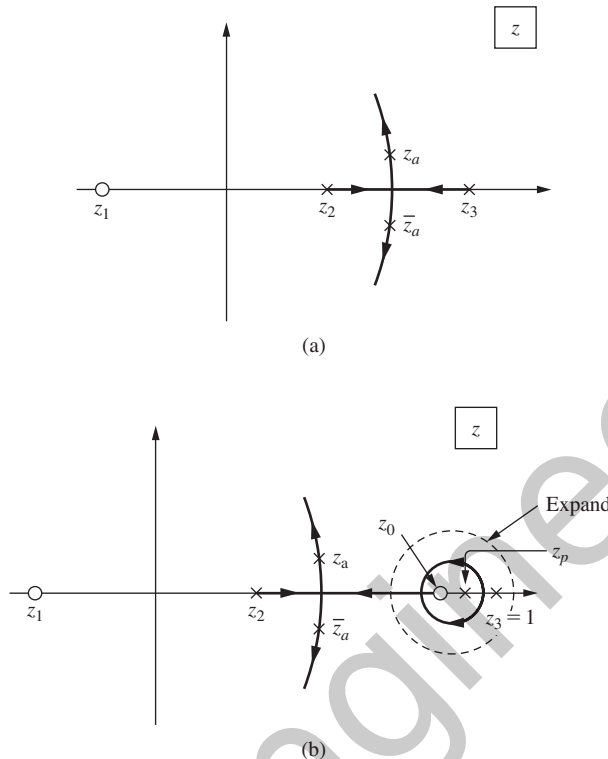
$$D(z) = \frac{K_d(z - z_0)}{z - z_p} \tag{8-71}$$

We will require that the filter dc gain equal unity ( $D(1) = 1$ ) so as not to affect the steady-state response; hence

$$K_d = \frac{1 - z_p}{1 - z_0} \tag{8-72}$$

In this design, the gain  $K$  is utilized to meet steady-state error requirements. The controller pole is restricted to real values inside the unit circle. For a phase-lead controller  $z_0 > z_p$ , and thus  $K_d > 1$ . For a phase-lag controller  $z_0 < z_p$  and  $K_d < 1$ .

**PHASE-LAG DESIGN.** We will first consider phase-lag design. We will illustrate the design for the plant,  $G(z)$ , of (8-70). The uncompensated root locus is sketched in Fig. 8-28(a). Suppose that the root locations  $z_a$  and  $\bar{z}_a$  give a satisfactory transient response, but that the loop gain  $K$  must be increased to produce smaller steady-state errors, improved disturbance rejection, and so on. We add the controller pole and zero as shown in Fig. 8-28(b), assuming that  $z_3 = 1$ . If  $z_3 \neq 1$ , the controller pole and zero are placed close to  $z = 1$ . Since the pole and zero are very close to  $z = 1$  (recall Example 8.1), the scale in the vicinity of this point is greatly expanded. Hence these two poles and one zero will essentially appear as a single pole as in Fig. 8-28(a), when



**FIGURE 8-28** Phase-lag design.

determining the root location at  $z_a$ . We see, then, that the compensator pole and zero cause the root at  $z_a$  to shift only slightly to  $z'_a$ , that is,  $z'_a \approx z_a$ . However,

$$K_c D(z)G(z) = \frac{K_c K_d (z - z_0)(z - z_1)}{(z - z_p)(z - z_2)(z - z_3)}$$

where  $K_c$  is the gain in the compensated system. Hence, for a root to appear at  $z'_a$ , from (8-68),

$$K_c = \frac{|z'_a - z_p| |z'_a - z_2| |z'_a - z_3|}{K_d |z'_a - z_0| |z'_a - z_1|} \approx \frac{|z_a - z_2| |z_a - z_3|}{K_d |z_a - z_1|} \quad (8-73)$$

Let  $K_u$  be the gain required to place the root at  $z_a$  in the uncompensated system, as shown in Fig. 8-28(a). Then

$$K_u = \frac{|z_a - z_2| |z_a - z_3|}{|z_a - z_1|} \quad (8-74)$$

Hence, from (8-73) and (8-74),

$$K_c \approx \frac{K_u}{K_d}$$

Now,  $K_u$  is the gain for the uncompensated system, and  $K_d < 1$  since the controller is phase lag. Hence the gain for the compensated system,  $K_c$ , is greater than that for the uncompensated system,  $K_u$ . So we see that phase-lag compensation allows us to increase the open-loop gain while maintaining the approximate same roots in the characteristic equation. Of course, as seen from Fig. 8-28(b), we have added a root close to  $z = 1$ , which has a long time constant. Thus the system will have an increased settling time.

The root-locus phase-lag design procedure is then as follows:

1. Choose the desired root location  $z_a$  and find  $K_u$ , the gain required to place the root at  $z_a$  in the uncompensated system.
2. Determine from the design specifications the required compensated-system gain  $K_c$ .
3. The compensator gain  $K_d$  is calculated from  $K_d = K_u/K_c$ .
4. Choose the compensator pole  $z_p$  sufficiently close to  $z = 1$  such that (8-73) is satisfied.
5. From (8-72), the compensator zero is given by

$$z_0 = 1 - \frac{1 - z_p}{K_d} \quad (8-75)$$

From (8-73), this procedure is only approximate.

### EXAMPLE 8.7

Consider the system of Fig. 8-29, with  $T = 1$  s and

$$G_p(s) = \frac{1}{s(s + 1)} \Rightarrow G(z) = \frac{0.368(z + 0.718)}{(z - 1)(z - 0.368)}$$

The root locus for the uncompensated system is illustrated in Fig. 8-28(a), with  $z_1 = -0.718$ ,  $z_2 = 0.368$ , and  $z_3 = 1$ . To find  $z_a$  and  $K_u$ , use MATLAB to generate the root locus of the uncompensated system. For our design, choose  $z_a$  to be the point of critical damping. The following MATLAB code generates the needed root-locus plot displayed in Fig. 30(a):

```
>> T = 1; Gp=tf([1],[1 1 0]), Gz=c2d(Gp,T)
rlocus(Gz), axis([0 1 -0.5 0.5]), grid
```

This system is shown to be critically damped with two poles at  $z = 0.648$  for  $K = 0.196$ . We will design a phase-lag compensator that results in critically damping with two poles at  $z \approx 0.65$  with  $K = 0.8$ . Hence the steady-state errors are reduced by a factor of approximately 4. From the design procedure:

1.  $K_u = 0.196$  and  $z_a = 0.648$ .
2.  $K_c = 0.8$ .

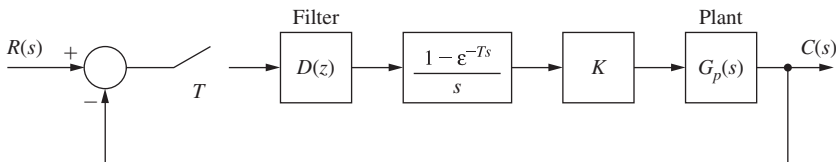
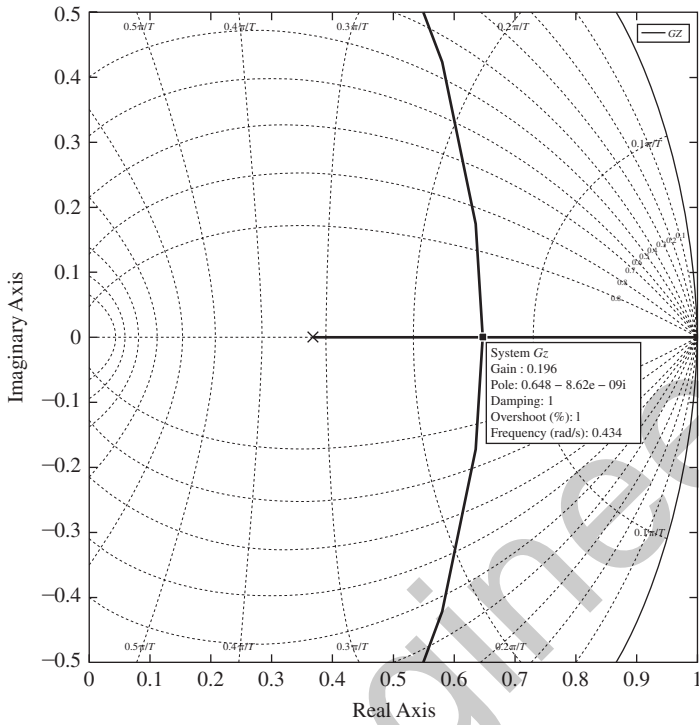
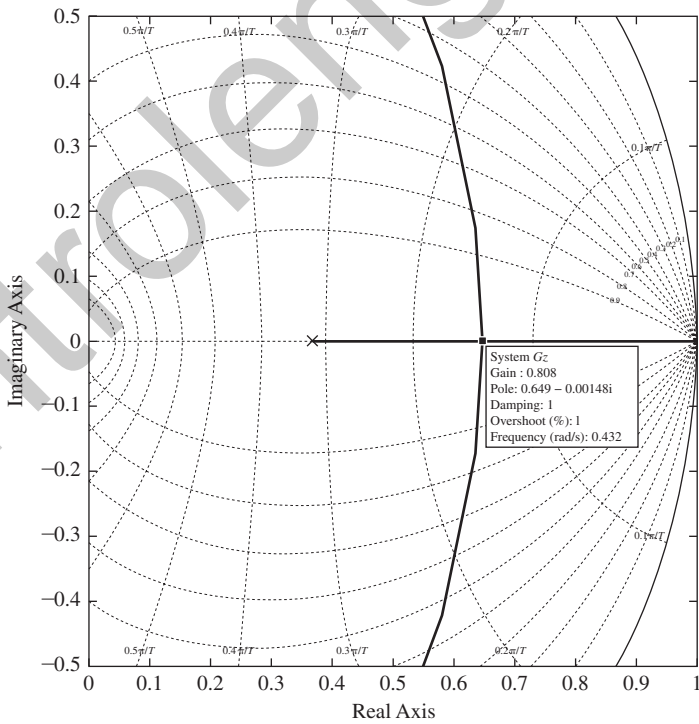


FIGURE 8-29 System for Examples 8.7 and 8.8.



(a)



(b)

**FIGURE 8-30** Root-locus plots for Example 8.7. Source: Reprinted with permission from The MathWorks, Inc.

3.  $K_d = \frac{K_u}{K_c} = \frac{0.196}{0.8} = 0.245$
4. Let  $z_p = 0.9990$ , a somewhat arbitrary choice.
5.  $z_0 = 1 - \frac{1 - 0.999}{0.245} = 0.9959$

Thus

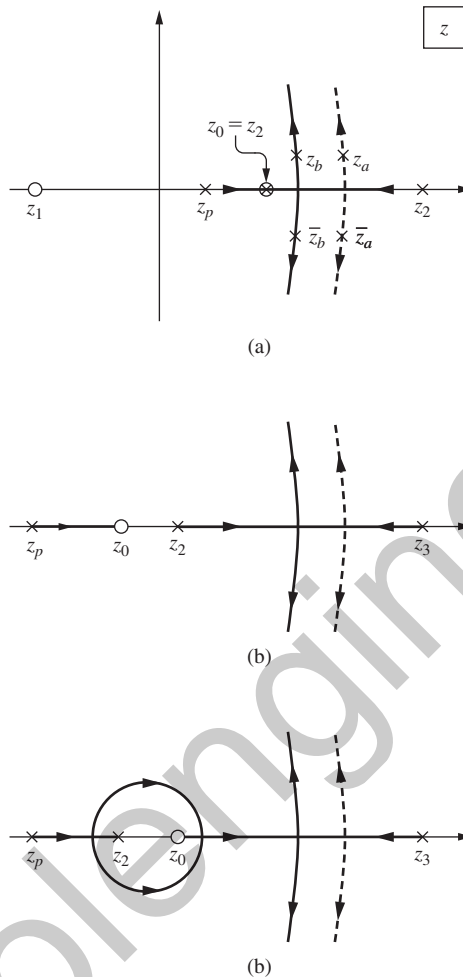
$$D(z) = \frac{0.245(z - 0.9959)}{z - 0.999} = \frac{0.245z - 0.2440}{z - 0.999}$$

Figure 8-30(b) shows the compensated root-locus plot. The plot gives two poles at  $z \approx 0.649$  for  $K = 0.808$ . A more accurate value is 0.80781. The pole location is very sensitive to small gain changes. The closed-loop third pole is at  $z \approx 0.9959$  (see Figs. 8-28(b)). Closed-loop performance metrics for this design include gain margin, 19.8 dB; phase margin, 69.1°; rise time, 7.0 s; settling time, 12.0 s; and overshoot, 1.46%. A MATLAB program to perform the computations is now given.

```

>> T = 1; Gp=tf([1],[1 1 0]); Gz=c2d(Gp,T);
Gz=zpk(Gz)
% Get Kc and za from system specifications
Kc = 0.8 % At critical damping
% Uncompensated Root Locus
rlocus(Gz), axis([0 1 -0.5 0.5]), grid
% Get Ku from rlocus and calculate Kd = Ku/Kc
Ku = 0.196
Kd = Ku/Kc
% Set zp=0.999 and calculate z0=1-(1-zp)/Kd
zp=0.9990
z0=1-(1-zp)/Kd
% Find D(z)
Dz=zpk(z0,zp,Kd,T)
DzGz=Dz*Gz
rlocus(DzGz,'-k'), axis([0 1 -0.5 0.5]), grid
% Root-locus near z = 0.995
% rlocus(DzGz,'-k'), axis([0.99 1.001 -0.00025 0.00025]), grid
% Set Kc to designed compensated system critical gain
% Closed-loop poles and zeroes
format long
Kc = 0.807813075
format short
Cz=feedback(Kc*DzGz,1); Cz=zpk(Cz)
disp('Gain and Phase Margin:')
[Gm,Pm] = margin(DzGz); GmdB = 20*log10(Gm), Pm
disp('Unit-step performance metrics:')
Cz_stepinfo=stepinfo(Cz)
    
```

**PHASE-LEAD DESIGN.** Phase-lead design is illustrated in Fig. 8-31. Here, to simplify the discussion, we place the controller zero coincident with the plant pole at  $z = z_2$ . The controller pole is placed to the left of the zero, which yields the phase-lead controller. Thus the root locus is shifted to the left. The resulting root at  $z_b$  has a smaller time constant than that at  $z_a$ ; thus the system responds faster (larger bandwidth).



**FIGURE 8-31** Phase-lead design.

A phase-lead design procedure is as follows:

1. Choose the desired root location  $z_b$ .
2. Choose the compensator zero  $z_0$  to cancel a pole of  $G(z)$ .
3. Choose either the gain  $K_c$  of the compensated system or the location of the compensator pole  $z_p < z_0$ , such that the compensator is phase lead.
4. Then, from (8-67),

$$K_c D(z_b) G(z_b) = -1 \quad (8-76)$$

Solve for the unknown, which is either  $K_c$  or  $z_p$ . The pole  $z_p$  appears twice in (8-76), as seen from (8-71) and (8-72).

Note that this design procedure sets only one root of the characteristic equation; the locations of the other roots may not be satisfactory. Hence iterations of steps 3 and 4 may be required. As in the case of frequency-response design, phase-lead design tends to be by trial and error.

As a practical consideration, exact pole-zero cancellation cannot occur. Hence, we actually have two different cases. The first case is illustrated in Fig. 8-31(b), where the compensator zero is to the left of the plant pole to be canceled. For this case, the closed-loop transfer function will have an additional pole slightly to the left of the compensator zero. For the second case, the compensator zero is to the right of the plant pole, as illustrated in Fig. 8-31(c). The closed-loop transfer function will then have an additional pole slightly to the right of the compensator zero. In either case, the amplitude of the transient-response term associated with this added closed-loop pole, as excited by the input, will be small, since the closed-loop transfer function has a zero (from the compensator) that is almost coincident with the pole.

In summary, the phase-lag controller shifts the root locus very little, but allows a higher open-loop gain to be used. Or if the same open-loop gain is used, the system is more stable. The phase-lead controller shifts the root locus to the left, resulting in a system that responds faster. A phase-lead example will now be given.

### EXAMPLE 8.8

Phase-lead design by trial and error will be illustrated in this example. Consider the system in Fig. 8-29. Suppose that the plant transfer function is given by

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

Since the time constant of the pole at  $s = -1$  is 1 s, we choose  $T = 0.1$  s. Then

$$\begin{aligned}
 G(z) &= \frac{z - 1}{z} \left[ \frac{1}{s^2(s + 1)} \right] \\
 &= \frac{0.004837(z + 0.9672)}{(z - 1)(z - 0.9048)}
 \end{aligned}$$

The root locus for  $G(z)$  is shown in Fig. 8-32(a). Now we follow the four design steps:

1. Choose the desired root location  $z_b$ . Note for the uncompensated system that  $K$  is equal to 0.244 for critical damping, with the two roots coincident at  $z = 0.952$ . We will design a critically damped compensated system, so  $z_b$  will lie on the real axis and be less than 0.952.
2. Choose the compensator zero  $z_0$  to cancel a pole of  $G(z)$ . We choose the pole at 0.9048.
3. Choose either gain  $K_c$  or the location of compensator pole  $z_p$ . In our case, we will try a pole location of 0.7 (which is less than our  $z_0$  choice as required). Since  $D(z)$  must be unity gain,

$$\begin{aligned}
 K_d &= \frac{1 - z_p}{1 - z_0} = \frac{1 - 0.7}{1 - 0.9048} = 3.15 \\
 D(z) &= \frac{K_d(z - z_0)}{z - z_p} = \frac{3.15(z - 0.9048)}{z - 0.7}
 \end{aligned}$$

4. We can find  $z_b = 0.844$  and  $K_c = 0.814$  from the compensated root locus which is plotted in Fig. 8-32(b). We can check the results using (8-76)

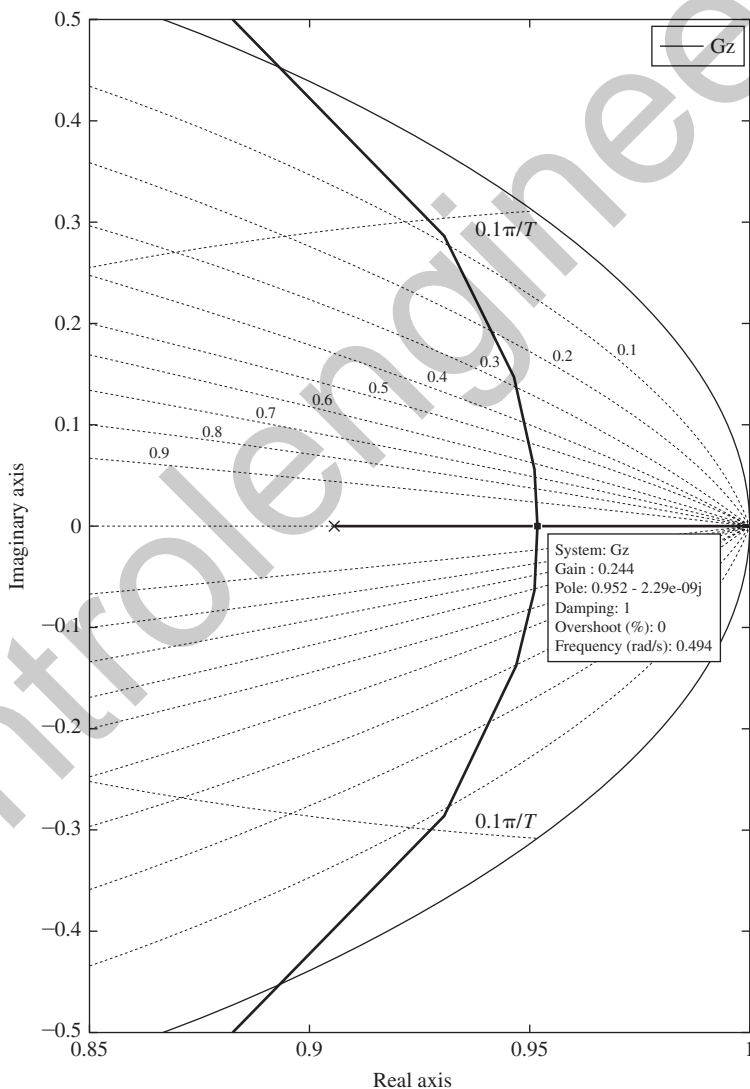


$$K_c = \frac{-1}{D(z_b)G(z_b)} = -\left(\frac{z_b - 0.7}{3.15(z_b - 0.9048)}\right)\left(\frac{(z_b - 1)(z_b - 0.9048)}{0.004837(z_b + 0.9672)}\right)$$

$$= -\left(\frac{z_b - 0.7}{3.15}\right)\left(\frac{(z_b - 1)}{0.004837(z_b + 0.9672)}\right)\Bigg|_{z_b=0.844} = 0.8136$$

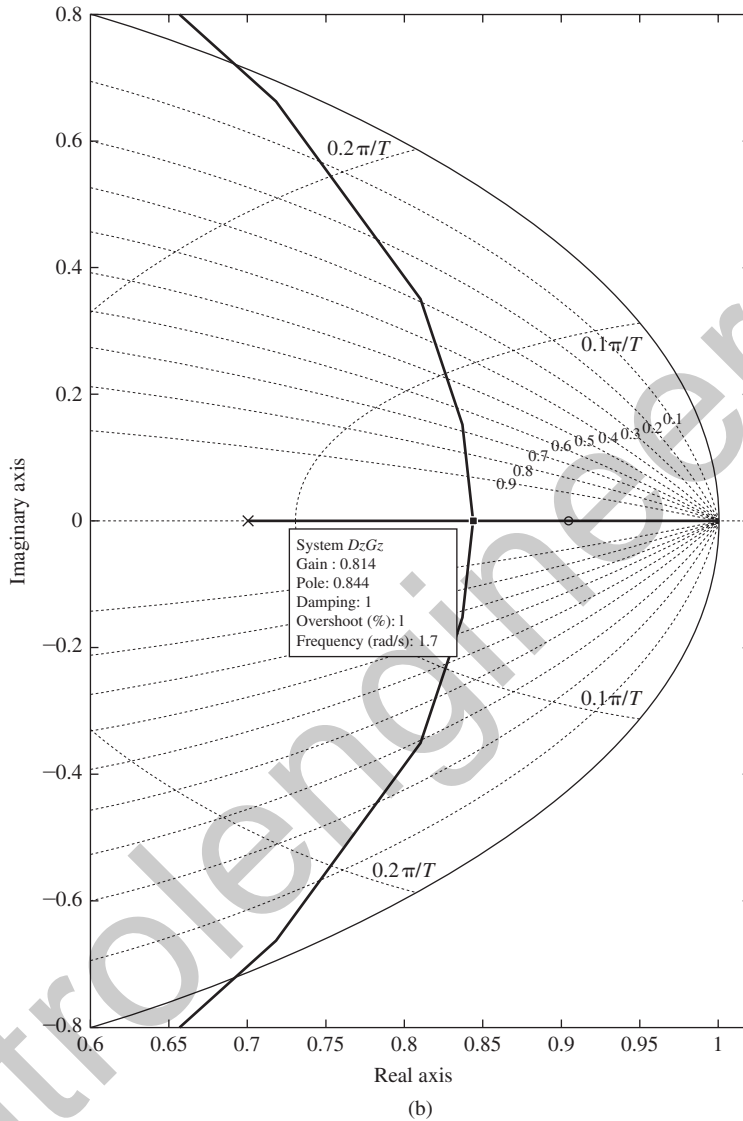
In this example, we choose the design of a critically damped system, so there are two closed-loop poles on the real axis. A pole in the  $s$ -plane at  $s = -a$  has a time constant of  $\tau = 1/a$  and an equivalent  $z$ -plane location  $\epsilon^{-aT} = \epsilon^{-T/\tau}$ . Thus, for the uncompensated critically damped case,

$$\epsilon^{-0.1/\tau} = 0.952$$



(a)

**FIGURE 8-32** Root locus for Example 8.8. Source: Reprinted with permission from The MathWorks, Inc.



**FIGURE 8-32** (Continued)

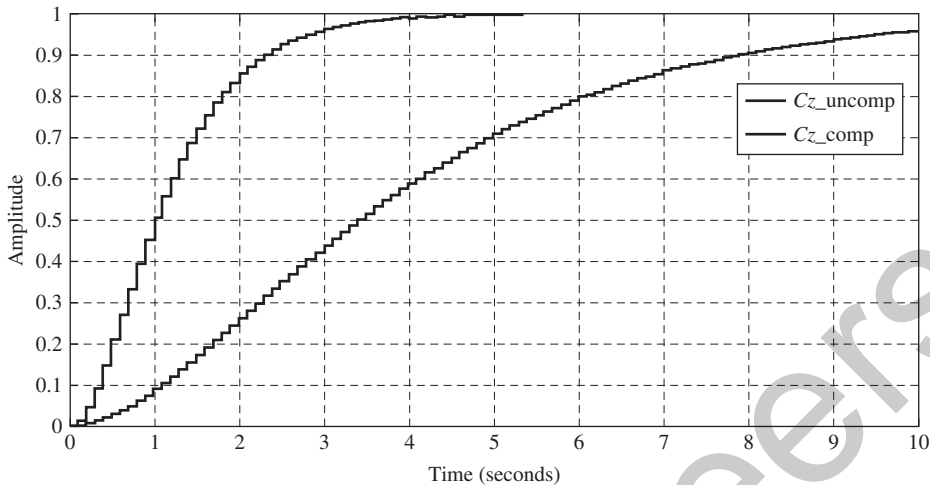
or  $\tau = 2.03$  s. For the compensated critically damped case,

$$e^{-0.1/\tau} = 0.844$$

or  $\tau = 0.59$  s. Thus the compensated system should respond about 3.4 times faster. Plots of the step responses for both the uncompensated system and the compensated system are shown in Fig. 8-33. The step performance metrics for the uncompensated system are rise time, 6.80 s; settling time, 11.90 s; and overshoot, 0%. The parameters for the compensated system are rise time, 1.90 s; settling time, 3.50 s; and overshoot, 0%. Note that the rise and settling times for the compensated system are indeed about 3.4 times faster.

A MATLAB program for this example follows.

```
>> Gp=tf([1],[1 1 0]); T = 0.1; Gz=c2d(Gp,T); Gz=zpk(Gz)
% Uncompensated rlocus
```



**FIGURE 8-33** Step responses for Example 8.8.

```
% Plot Figure 8-32a
figure(1), rlocus(Gz), axis([0.85 1 -0.5 0.5]), grid
% Critical damping at Ku=0.244 with za = 0.952
% Choose zb as critically-damped compensated poles
% Choose z0 to cancel the Gp pole at 0.9048
Ku = 0.244, z0 = 0.9048
% Try the compensator at 0.7
zp = 0.7
% Find D(z)
Kd = (1-zp)/(1-z0)
Dz = zpk(z0,zp,Kd,T)
% Find Kc and zb from the compensated rlocus
DzGz=Dz*Gz
% Plot Figure 8-32b
figure(2), rlocus(DzGz), axis([.6 1.02 -0.8 0.8]), grid
% At Kc = 0.814, zb = 0.844
% Check results using equation (8-76)
zb = 0.844
Kc = -(zb-0.7)*(zb-1)/0.015244/(zb+0.9672)
Cz_uncomp=feedback(Ku*Gz,1); Cz_comp=feedback(Kc*DzGz,1);
% Plot Figure 8-33
figure(3), step(Cz_uncomp, '-k',Cz_comp,'-k',10), grid
disp('Unit-step performance metrics:')
Cz_uncomp_info=stepinfo(Cz_uncomp), Cz_comp_info=stepinfo(Cz_comp)
```

A point must be made concerning the last example. It appears that we can increase the speed of response of this system by a very large factor, simply by moving the pole of the controller to the left. However, this movement of the pole increases the high-frequency gain of the controller, and this increase may not be acceptable. To illustrate this point, suppose that the plant in Fig. 8-29 is a servomotor, and the gain  $K$  is a power amplifier that drives the motor. For the uncompensated system with a unit-step input, the maximum error is 1, which appears at  $t = 0$  as shown in Fig. 8-33. Since the amplifier gain  $K = 0.244$ , the maximum amplifier output is 0.244. For the compensated system, the maximum error is also 1, at  $t = 0$ . At this instant, the controller output is 3.15 (the interested reader can prove this). Hence, since the amplifier gain is 0.814, the amplifier output is 2.56, which is an increase by more than a factor of 10 over that of the uncompensated system. This increase

in motor voltage accounts for the faster response. However, the larger voltages generated by the phase-lead compensation may force the system into nonlinear regions of operation. A discussion of the possible effects of this operation is beyond the scope of this book; however, in nonlinear regions of operation the results of the linear analysis given above will no longer be applicable.

As a last example in this chapter, let us return to the system of Example 8-1. For this example, we will first use root locus to design a critically-damped lead controller, then increase the open-loop gain appropriately to achieve a phase margin of approximately 55°.

### EXAMPLE 8.9

We can use the MATLAB program of Example 8.8 to compute the parameters for this controller. First we examine the discrete open-loop transfer function  $G(z)$

$$G(z) = \frac{4.014 \times 10^{-5}(z + 3.595)(z + 0.258)}{(z - 1)(z - 0.9512)(z - 0.9048)}$$

and determine that we can use the controller zero to cancel the discrete plant pole at 0.9512. If we again assume that the controller pole will arbitrarily be 0.7, the controller gain must be  $K_d = 6.148$  to make  $D(1) = 1$ . Next, we plot the root locus for  $D(z)G(z)$ . From that plot, a closed-loop gain  $K_c = 0.42$  would make the controlled system critically damped with closed-loop roots at 0.957 and no overshoot. However, in this case we want to increase  $K_c$  to make the system respond faster, allowing some overshoot to take place. By trial and error, we can increase the gain by a factor of 2.57 yielding the final controller as

$$D(z) = \frac{(2.57)(0.42)(6.148)(z - 0.9512)}{(z - 0.7)} = \frac{6.64(z - 0.9512)}{(z - 0.7)}$$

This controller achieves a phase margin of 55.1°; margin, 16.7 dB, rise time, 1.2 s; settling time, 4.2 s, and overshoot, 12.9%. Using (6-21), the steady-state unit ramp error is 0.916.

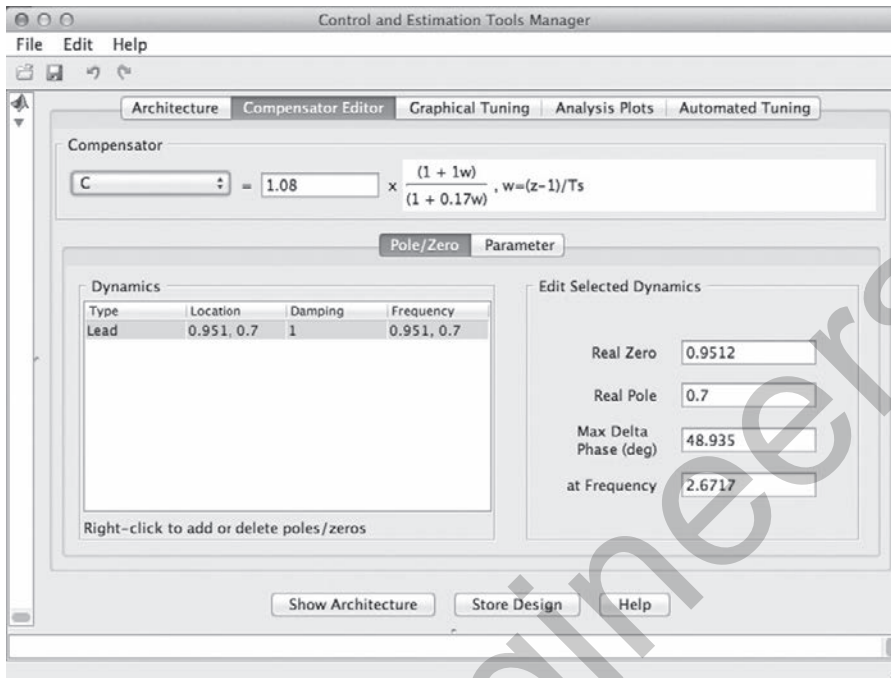
Trial-and-error design as performed above may also be accomplished using MATLAB's *sisotool*. To use *sisotool*, first create  $G(z)$  in the MATLAB Workspace, then launch the tool and import the plant model into System Data.

```
>> Gp = tf([2], [1 3 2 0]); Gz = c2d(Gp, .05); Gz=zpk(Gz)
sisotool
```

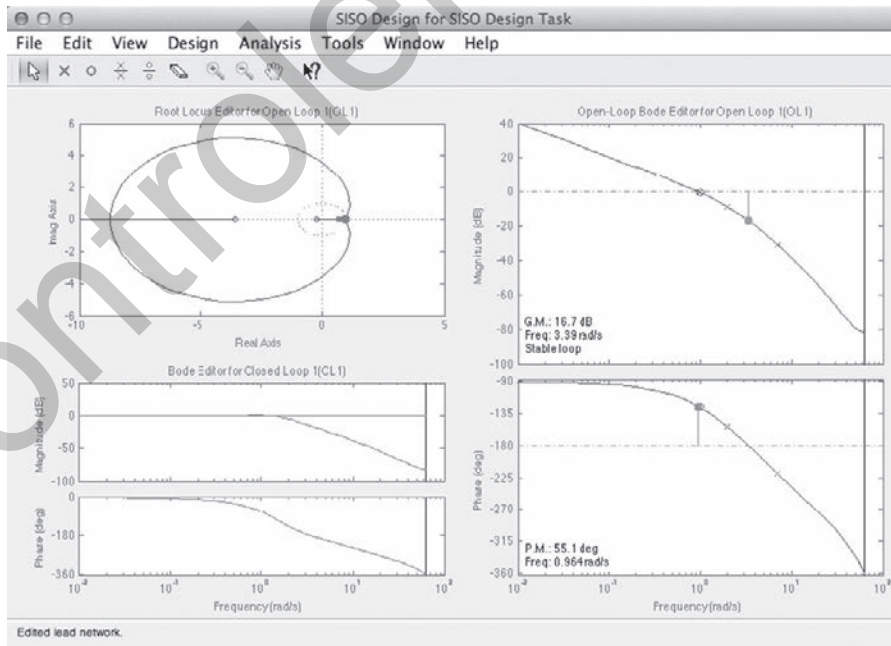
Next go to the Compensation Editor and *Add Pole/Zero*. The choices offered are Real Pole, Complex Pole, Integrator, Real Zero, Complex Zero, Differentiator, Lead, Lag, and Notch. For this example, select the Lead option. The *sisotool* then places a real zero at 0.95123 and a real pole at 0.60653 with a damping factor of 1.0 and dc gain of 1.0. Change the real zero to 0.9512 and the real pole to 0.70 (as was designed above) and adjust the compensator gain to achieve a 55° phase margin. This may be done by trying various values for the dc filter gain in the Control and Estimation Tools Manager window as shown in Fig. 8-34 (change 1.0 to 1.08), or by adjusting the dc filter gain in the open-loop Bode plot in the SISO Design Task window as shown in Figure 8-35 (click on the square marker and raise the curve to reduce the gain margin to 16.7 dB). The resulting filter gain is

$$K_d = a_0 \frac{1 - z_p}{1 - z_0} = 1.08 \left( \frac{1 - 0.70}{1 - 0.9512} \right) = 6.64$$

which is the value obtained above.



**FIGURE 8-34** MATLAB *sisotool* Control and Estimation Manager window. Source: Reprinted with permission from The MathWorks, Inc.



**FIGURE 8-35** MATLAB *sisotool* SISO Design Task window. Source: Reprinted with permission from The MathWorks, Inc.

**TABLE 8-3 Comparison of Performance Parameters for Examples 8.1 – 8.6 and 8.9**

Design type	Controller $D(z)$	Gain margin (dB)	Phase margin (deg)	Rise time (s)	Settling time (s)	Over-shoot (%)	Ramp error
8.1 lag	$\frac{0.389(z - 0.9982)}{z - 0.9993}$	16.8	55.9	3.15	32.4	14.1	1.00
8.2 lead	$\frac{6.528(z - 0.9711)}{z - 0.8111}$	12.4	55.0	1.05	5.2	11.4	1.00
8.3 lag-lead	$\frac{5.227(z - 0.9982)(z - 0.9792)}{(z - 0.9993)(z - 0.8604)}$	11.2	55.0	1.05	5.3	10.7	0.50
8.4 PI	$\frac{0.4393(z - 0.99954)}{z - 1}$	16.0	55.0	2.95	19.4	13.8	0
8.5 PID	$\frac{28.52(z - 0.99986)(z - 0.9504)}{z(z - 1)}$	23.3	55.0	1.05	3.6	13.0	0
8.6 pidtool	$\frac{14.37(z - 0.9995)(z - 0.9569)}{(z - 1)(z - 0.5221)}$	17.8	55.0	1.0	3.45	12.9	0
8.9 sisotool	$\frac{6.64(z - 0.9512)}{z - 0.7}$	16.7	55.1	1.25	4.25	12.6	0.98

In this chapter, seven different controllers were designed for this system. A summary of the performance parameters for each of those designs is provided in Table 8-3.

## 8.12 SUMMARY

Various criteria used in the specification of control systems are presented in this chapter. Next, digital controller design techniques using phase-lead and phase-lag compensation are developed. These techniques, based on frequency responses, tend to be largely trial and error, but are among the more commonly used techniques in compensator design. Then the three-term controller (PID) is developed, and is seen to be a special type of lag-lead design. Next, the MATLAB pidtool is illustrated in an example. Finally, design by root-locus techniques is developed and the MATLAB sisotool is introduced.

## References and Further Readings

- [1] C. L. Phillips and J. Parr, *Feedback Control Systems*, 5th ed. Upper Saddle River, NJ: Prentice-Hall, 2011.
- [2] M. E. Van Valkenberg, *Network Analysis*. Upper Saddle River, NJ: Prentice-Hall, 1974.
- [3] K. Ogata, *Modern Control Engineering*, 5th ed. Upper Saddle River, NJ: Prentice-Hall, 2009.
- [4] C. H. Roth, Jr. and L. L. Kinney, *Fundamentals of Logic Design*, 7th ed. Independence, KY: Cengage Learning, 2013.
- [5] M. L. Dertouzos, M. Athans, R. N. Spann, and S. J. Mason, *Systems, Networks, and Computation: Basic Concepts*. Huntington, NY: R.E. Krieger Publishing Co., Inc., 1979.
- [6] C. I. Huang, "Computer Aided Design of Digital Controllers," M.S. thesis, Auburn University, Auburn, AL, 1981.
- [7] W. R. Wakeland, "Bode Compensator Design," *IEEE Trans. Autom. Control* AC-21, p. 771, 1976.

[8] J. R. Mitchell, "Comments on Bode Compensator Design," *IEEE Trans. Autom. Control* AC-22, p. 869, 1977.

[9] B. C. Kuo, *Digital Control Systems*, 2d ed. New York: Oxford University Press, 1995.

[10] J. A. Cadzow and H. R. Martens, *Discrete-Time and Computer Control Systems*. Upper Saddle River, NJ: Prentice-Hall, 1970.

[11] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 12th ed. Upper Saddle River, NJ: Prentice-Hall, 2010.

## Problems

8.2-1. Consider the second-order analog system described by

$$T(s) = \frac{G_p(s)}{1 + G_p(s)}, \quad G_p(s) = \frac{\omega_n^2}{s(s + 2\zeta\omega_n)}$$

where  $T(s)$  is the closed-loop transfer function and  $G_p(s)$  is the plant transfer function. Show that the phase margin  $\phi_m$  and the damping ratio  $\zeta$  are related by  $\phi_m \approx 100\zeta$  by sketching this relationship onto Fig. 8-2.

8.2-2. Consider the system of Fig. P8.2-2 with  $T = 0.2$ .

(a) Show that the pulse transfer function of the plant is given by

$$G(z) = \frac{z - 1}{z} \mathfrak{Z} \left[ \frac{1}{s^2(s + 1)} \right] = \frac{0.01873z + 0.01752}{(z - 1)(z - 0.8187)}$$

- (b) The frequency response for  $G(z)$  is given in Table P8.2-2. From this frequency response, sketch the Bode and the Nyquist diagrams for the uncompensated system, indicating the gain and phase margins.
- (c) Calculate the value of  $G(j\omega_w)$  as  $\omega_w \rightarrow \infty$ . It is not necessary to find  $G(w)$  to calculate this value.
- (d) Use the results of part (b) to estimate the overshoot in the unit-step response.
- (e) Based on the results in part (d), are the zeros of the uncompensated system characteristic equation real or complex? Why?
- (f) Use MATLAB to find the rise time  $t_r$ . The step response will show a 21 percent overshoot.

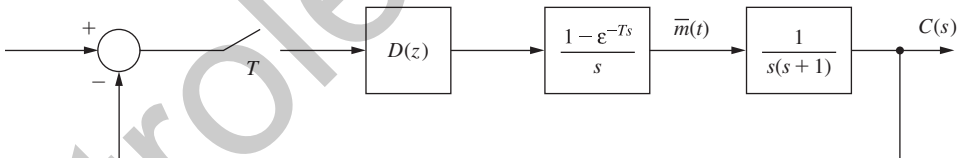


FIGURE P8.2-2 System for Problem 8.2-2.

TABLE P8.2-2 Frequency Response for PROBLEM 8.2-2

$\omega_w$	$\omega$	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.1	0.100	9.95054	19.95	-96.28
0.2	0.200	4.90325	13.80	-102.45
0.3	0.299	3.19331	10.08	-108.41
0.4	0.399	2.32198	7.31	-114.08
0.5	0.499	1.78990	5.05	-119.40
0.6	0.599	1.43046	3.10	-124.36
0.7	0.698	1.17191	1.37	-128.95
0.8	0.798	0.97794	-0.19	-133.17

**TABLE P8.2-2 Continued**

$\omega_w$	$\omega$	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.9	0.897	0.82799	-1.63	-137.05
1.0	0.996	0.70945	-2.98	-140.61
2.0	1.974	0.22743	-12.86	-164.43
3.0	2.914	0.10973	-19.19	-177.74
4.0	3.805	0.06511	-23.72	-187.04
5.0	4.636	0.04372	-27.18	-194.33
6.0	5.404	0.03186	-29.93	-200.38
7.0	6.107	0.02459	-32.18	-205.55
8.0	6.747	0.01980	-34.06	-210.03
9.0	7.328	0.01646	-35.67	-213.95
10.0	7.854	0.01403	-37.05	-217.40
20.0	11.071	0.00558	-45.07	-236.77
30.0	12.490	0.00352	-49.07	-243.95
40.0	13.258	0.00259	-51.73	-246.94

- 8.2-3.** Consider the system of Problem 8.2-2. It is desired that the steady-state error constant for a unit-ramp input,  $K_v$ , of (6-20), be 4, resulting in a steady-state error of 0.25.
- Repeat Problem 8.2-2(a), (b), (c), and (d) for this system.
  - Use the phase margin  $\phi_m$  and Fig. 8-2 to estimate the characteristics  $\zeta$ ,  $M_r$ , and  $M_p$ .
  - Verify the value of  $M_p$  in part (b) using MATLAB. The step response will show that  $M_p \approx 1.62$ .
- 8.2-4.** Consider the system of Problem 8.2-2. The plant frequency response  $G(j\omega_w)$  is given in Table P8.2-2.
- Let the digital controller realize a gain  $K$ , that is,  $D(z) = K$ . Find  $K$  such that the system phase margin is  $45^\circ$ .
  - Obtain the system step response of part (a) using MATLAB, and find the rise time  $t_r$  and the percent overshoot.
- 8.4-1.** Consider the system of Fig. P8.2-2 with  $T = 1$  s. The plant frequency response  $G(j\omega_w)$  is given in Table P8.4-1.
- Let  $D(z) = 1$ . If the system is stable, find its phase margin.
  - Let the digital controller realize a gain  $K$ , that is,  $D(z) = K$ . Find  $K$  such that the system phase margin is  $45^\circ$ .
  - Obtain the system step response of part (b) using MATLAB, and find the rise time  $t_r$  and the percent overshoot.
  - Design a unity-dc-gain phase-lag compensator that yields a phase margin of approximately  $45^\circ$ .
  - Obtain the system step response of part (d) using MATLAB, and find the rise time  $t_r$  and the percent overshoot.

**TABLE P8.4-1 Frequency Response for PROBLEM 8.4-1**

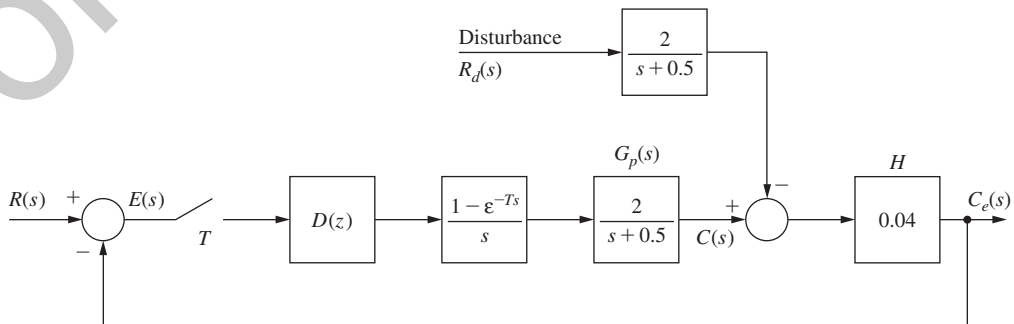
$\omega_w$	$\omega$	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.1	0.099	9.95473	19.96	-98.56
0.2	0.199	4.91192	13.82	-106.98
0.3	0.297	3.20693	10.12	-115.10
0.4	0.394	2.34103	7.38	-122.83



**TABLE P8.4-1 Continued**

$\omega_w$	$\omega$	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.5	0.490	1.81474	5.17	-130.10
0.6	0.582	1.46124	3.29	-136.87
0.7	0.673	1.20853	1.64	-143.14
0.8	0.761	1.02011	0.17	-148.92
0.9	0.845	0.87530	-1.15	-154.24
1.0	0.927	0.76140	-2.36	-159.13
2.0	1.570	0.30058	-10.44	-190.88
3.0	1.965	0.18220	-14.78	-205.37
4.0	2.214	0.13244	-17.55	-212.26
5.0	2.380	0.10579	-19.51	-215.43
6.0	2.498	0.08942	-20.97	-216.61
7.0	2.585	0.07848	-22.10	-216.68
8.0	2.651	0.07073	-23.00	-216.11
9.0	2.704	0.06502	-23.73	-215.18
10.0	2.746	0.06068	-24.33	-214.06
20.0	2.942	0.04455	-27.02	-203.02
30.0	3.008	0.04097	-27.75	-196.54
40.0	3.041	0.03964	-28.03	-192.77

- 8.4-2.** To satisfy the steady-state constraints for the system of Problem 8.2-3, the dc gain of the digital controller must be equal 4.
- Design a phase-lag controller with this dc gain that will result in a  $45^\circ$  phase margin.
  - Estimate the percentage of overshoot in the step response of the compensated system.
  - Verify the step-response results in part (b) using MATLAB. The step response will show an overshoot in the step response of approximately 24 percent.
- 8.4-3.** Shown in Fig. P8.4-3 is the block diagram of the temperature control system described in Problem 1.6-1. For this problem, ignore the disturbance input. In Fig. P8.4-3 the sensor gain of  $H = 0.04$  has been shifted into the forward path to yield a unity feedback system. The stability characteristics are unchanged, since the loop transfer function has not changed. Note that  $c(t)$  is the chamber temperature in degrees Celsius. It is shown in Problem 6.2-4 that with  $T = 0.6$ , the function  $G(z)H$  is given by



**FIGURE P8.4-3** Modified temperature control system.

**TABLE P8.4-3** Frequency Response for PROBLEM 8.4-3

$\omega_w$	$\omega$	$ G(j\omega_w) $	$ G(j\omega_w) _{dB}$	$\angle G(j\omega_w)$
0.1	0.100	3.92295	11.87	-13.11
0.2	0.199	3.71673	11.40	-25.38
0.3	0.299	3.43700	10.72	-36.29
0.4	0.398	3.13670	9.92	-45.71
0.5	0.496	2.84938	9.09	-53.74
0.6	0.593	2.59043	8.26	-60.60
0.7	0.690	2.36393	7.47	-66.52
0.8	0.785	2.16851	6.72	-71.68
0.9	0.879	2.00067	6.02	-76.23
1.0	0.971	1.85649	5.37	-80.30
2.0	1.801	1.12345	1.01	-107.02
3.0	2.442	0.87830	-1.12	-122.59
4.0	2.920	0.76932	-2.27	-133.12
5.0	3.276	0.71225	-2.94	-140.64
6.0	3.545	0.67895	-3.36	-146.21
7.0	3.754	0.65796	-3.63	-150.48
8.0	3.920	0.64393	-3.82	-153.83
9.0	4.053	0.63411	-3.95	-156.52
10.0	4.163	0.62698	-4.05	-158.72
20.0	4.685	0.60357	-4.38	-169.11
30.0	4.867	0.59912	-4.44	-172.71
40.0	4.958	0.59756	-4.47	-174.52

$$G(z) = (0.04) \frac{z-1}{z} \mathfrak{F} \left[ \frac{2}{s(s+0.5)} \right] = \frac{0.04147}{z-0.7408}$$

The frequency response for  $G(z)$  is given in Table P8.4-3. The sensor gain  $H = 0.04$  is *not* included in this table.

- Calculate the dc gains of both the forward path and the closed-loop system, with  $D(z) = 1$ .
- Find the steady-state error  $e_{ss}(kT)$  for a unit step input using (6-19). Express this error in percent.
- Let  $D(z) = K$ , a pure gain. Find  $K$  such that the steady-state error in part (b) is 0.05, or 5 percent. Assume that the resulting system is stable.
- It was shown in Problem 7.5-3 that the system is unstable for  $K > 41.96$ . Can the system in part (c) be implemented to yield a 5 percent steady-state error? Explain your answer.
- Design a phase-lag compensator with a dc gain of 118.8 to yield a phase margin of  $45^\circ$ .
- What is the percent steady-state error for a constant input for the system in part (e)?
- Simulate the system of part (e) using MATLAB to show that the steady-state error is 5 percent.

**8.4-4.** This problem is based on the solution in Problem 8.4-3. Suppose that in Fig. P8.4-3, the sensor gain  $H = 0.04$  is moved back to the feedback path. What effect does this have on the percent steady-state error of Problem 8.4-3, where this error is defined as

$$\text{steady-state error} = \frac{\text{commanded output} - \text{actual output}}{\text{commanded output}} \times 100$$

Note that this result allows us to convert a nonunity feedback gain system to a unity-gain feedback system, without affecting the percentage steady-state error.

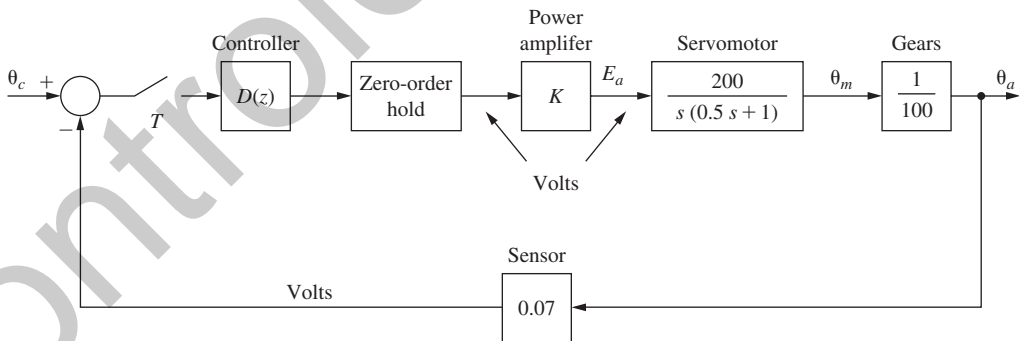
- 8.6-1.** Consider the system of Problem 8.4-1.
- Design a unity-dc-gain phase-lead compensator that yields a phase margin of approximately  $45^\circ$ .
  - Obtain the system step response of part (a) using MATLAB, and find the rise time  $t_r$  and the percent overshoot.
  - Compare the step-response parameters of part (b) with Problem 8-2(c) and (e).
- 8.6-2.** Repeat Problem 8.4-2 using a phase-lead controller. In part (c), the overshoot is approximately 26 percent.
- 8.6-3.** Three constraints are given on the choice of the phase-margin frequency,  $\omega_{w1}$ , in Section 8.6 for phase-lead design. Derive the three equivalent constraints on  $\omega_{w1}$  for the design of phase-lag controllers using (8-33).
- 8.6-4.** Consider the block diagram of a robot-arm control system shown in Fig. P8.6-4. This system is described in Problem 1.5-4. Let  $T = 0.1$ . It was shown in Problem 4.3-8 that

$$G(z) = \frac{z-1}{z} \mathfrak{Z} \left[ \frac{1}{s^2(s+2)} \right] = \frac{0.01873z + 0.0175}{(z-1)(z-0.818)}$$

The frequency response for  $G(z)$  is given in Table P7-23. Note that the sensor gain  $H = 0.05$  is not included in this table.

- Find the system phase margin with  $D(z) = 1$ .
- Design a phase-lag controller with the dc gain of 10 that yields a system phase margin of  $45^\circ$ .
- Design a phase-lead controller with the dc gain of 10 that yields a system phase margin of  $45^\circ$ .
- Using MATLAB, find the step response for the systems of parts (b) and (c), with  $\theta_c(t) = 0.05u(t)$ . Compare the rise times and the percent overshoot for the two systems. The percent overshoot is defined as

$$\text{percent overshoot} = \frac{\text{maximum value} - \text{final value}}{\text{final value}} \times 100$$



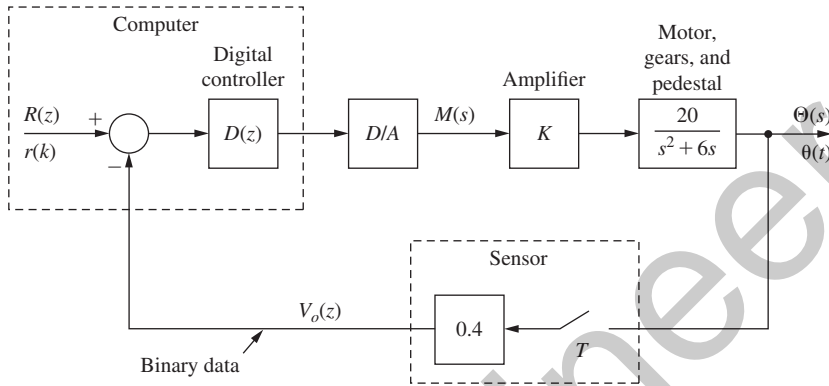
**FIGURE P8.6-4** Robot arm joint control system.

- 8.6-5.** Consider the block diagram of an antenna control system shown in Fig. P8.6-5. This system is described in Section 1.5. Let  $T = 0.05$  and the sensor gain be unity ( $H = 1$ ). It was shown in Problem 7.5-5 that

$$G(z) = \frac{z-1}{z} \mathfrak{Z} \left[ \frac{20}{s^2(s+6)} \right] = \frac{0.02268z + 0.02052}{(z-1)(z-0.7408)}$$

The frequency response for  $G(z)$  is given in Table P7-24.

- Find the system phase margin with  $K = 1$  and  $D(z) = 1$ .
- To reduce steady-state errors,  $K$  is increased to 5. Design a unity-dc-gain phase-lag controller that yields a system phase margin of  $45^\circ$ .
- Design a unity-dc-gain phase-lead controller, with  $K = 5$ , that yields a system phase margin of  $45^\circ$ .
- Using MATLAB, find the unit step response for the systems of parts (b) and (c). Compare the rise times and the percent overshoot for the two systems.



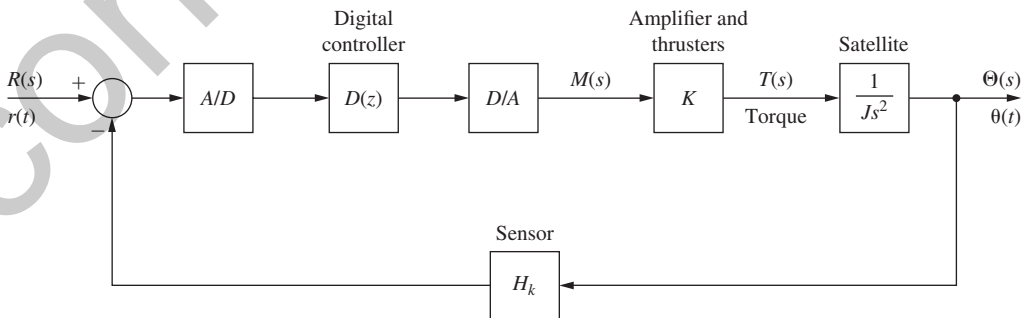
**FIGURE P8.6-5** Block diagram for an antenna control system.

**8.6-6.** Consider the block diagram of a satellite control system shown in Fig. P8.6-6. This system is described in Problem 1.4-1. Let  $T = 0.1$  s,  $K = 1$ ,  $J = 0.1$ , and  $H_k = 0.02$ . It was shown in Problem 7.5-6 that

$$G(z) = \frac{z-1}{z} \mathfrak{Z} \left[ \frac{10}{s^3} \right] = \frac{0.05(z+1)}{(z-1)^2}$$

The frequency response of  $G(z)$  is given in Table P7-25. Note that the gain  $H_k=0.02$  is not included in this table.

- Sketch the complete Nyquist diagram for  $D(z) = 1$ . Is this system stable?
- This system cannot be stabilized by any type of phase-lag controller. Why? *Hint:* Consider the Nyquist criterion and the Nyquist diagram in part (a).
- Design a unity-dc-gain phase-lead controller that yields a system phase margin of  $45^\circ$ . *Hint:* Use  $\omega_{w1} = 1$ .
- Using MATLAB, find the unit step response for the system of part (c). Find the rise time and the percent overshoot for this system.



**FIGURE P8.6-6** Block diagram for a satellite control system.

- 8.7-1.** Design a lag-lead compensator for the system of Problem 8.4-1. Note that  $G_p(s) = 1/(s^2 + s)$ ,  $T = 1$  s, and  $H = 1$ .
- Use the lag compensator of Problem 8.4-1 and decrease its gain by a factor of 3.
  - Design a cascaded lead compensator for a target phase margin of 55 degrees for the lag-lead combination. Why might this be hard to achieve?
  - Plot the step responses for this system for the lag compensator of Problem 8.4-1, the lead compensator of 8.6-1, and the lag-lead design of this problem.
- 8.7-2.** Design a lag-lead compensator for the system of Problem 8.2-2. Note that  $G_p(s) = 1/(s^2 + s)$ ,  $T = 0.2$  s, and  $H = 1$ .
- Use the lag compensator of Problem 8.4-2 and increase its gain by a factor of 2.
  - Cascade a lead compensator to achieve a phase margin of 60 degrees for the lag-lead combination.
  - Plot the step responses for this system for the lag compensator of Problem 8.4-2, the lead compensator of 8.6-2, and the lag-lead design of this problem.
- 8.9-1.** Write the difference equations required for the realization of the PID controller transfer function  $D(z)$  in (8-52). Let  $E(z)$  be the controller input and  $M(z)$  be the controller output.
- 8.9-2.** If the rectangular integrator [see below (8-38)] is used in the PID transfer function in (8-52), the resulting controller is described by

$$\frac{M(z)}{E(z)} = D(z) = K_p + K_I \left[ \frac{Tz}{z-1} \right] + K_D \left[ \frac{z-1}{Tz} \right]$$

This transfer function is implemented in many commercial digital controllers. Write a difference equation required to realize this controller.

- 8.10-1.** Repeat Problem 8.4-1, using a proportional-plus-integral digital filter.
- 8.10-2.** Repeat Problem 8.6-1, using a proportional-plus-derivative filter.
- 8.10-3.** Consider Problem 8.4-2 again. In this problem a proportional-plus-integral (PI) filter is to be designed.
- The gain added in Problem 8.4-6 to reduce steady-state errors is no longer necessary. Why?
  - Repeat Problem 8.4-6 using a proportional-plus-integral filter, with the gain of 4 omitted.
- 8.10-4.** Repeat Problem 8.6-2 using a proportional-plus-derivative (PD) filter.
- 8.10-5.** Repeat Problem 8.4-3 for a proportional-plus-integral (PI) compensator. Note that in this case, the steady-state error for a constant input is zero.
- 8.10-6.** In this problem we consider the effects of the disturbance input in Fig. P8.4-3.
- Suppose that the disturbance input is a unit step, which models the door to the chamber remaining open. Find the steady-state error in the temperature  $c(t)$  for the uncompensated system in Problem 8.4-3.
  - Repeat part (a) for the phase-lag compensated system in Problem 8.4-3(e).
  - Repeat part (a) for the PI-compensated system in Problem 8.10-5.
- 8.10-7.**
- Design a PI controller for Problem 8.6-4(b).
  - Design a PD controller for Problem 8.6-4(c).
  - Use the results of parts (a) and (b) to repeat Problem 8.6-4(d).
- 8.10-8.**
- Design a PI controller for Problem 8.6-5(b).
  - Design a PD controller for Problem 8.6-5(e).
  - Use the results of parts (a) and (b) to repeat Problem 8.6-5(d).
- 8.10-9.**
- Repeat Problem 8.6-6(a) and (b).
  - Design a PD controller for Problem 8.6-6(c). *Hint:* Use  $\omega_{w1} = 1$ .
  - Use the results of part (b) to repeat Problem 8.6-6(d).
- 8.10-10.** Consider the system of Example 8.1.
- Design a PI filter to achieve a phase margin of 60°.
  - Obtain the system step response using MATLAB. Compare this response to that of the system of Example 8.1, which is plotted in Fig. 8-14.
- 8.10-11.** Use the MATLAB *pidtool* to design a PID controller for the system of Problem 8.6-1. Start with these parameters, then try to improve the design ( $G_p = 1/(s^2 + s)$ ,  $T = 1$ ,  $H = 1$ ,  $P_m = 45$ ,  $\omega_{w1} = 1.0$ ).

- 8.10-12.** Use the MATLAB pidtool to design a PID controller for the system of Problem 8.6-2. Start with these parameters, then try to improve the design ( $G_p = 1/(s^2 + s)$ ,  $T = 0.2$ ,  $H = 1$ ,  $P_m = 45$ ,  $\omega_{w1} = 3.0$ ).
- 8.10-13.** Use the MATLAB pidtool to design a PID controller for the system of Problem 8.6-4. Start with these parameters, then try to improve the design ( $G_p = 4/(s^2 + 2s)$ ,  $H = 0.07$ ,  $T = 0.1$ ,  $P_m = 45$ ,  $\omega_{w1} = 6.0$ ).
- 8.10-14.** Use the MATLAB pidtool to design a PID controller for the system of Problem 8.6-5. Start with these parameters, then try to improve the design ( $G_p = 20/(s^2 + 6s)$ ,  $H = 1$ ,  $T = 0.05$ ,  $P_m = 45$ ,  $\omega_{w1} = 20.0$ ).
- 8.10-15.** Use the MATLAB pidtool to design a PID controller for the system of Problem 8.6-6. Start with these parameters, then try to improve the design ( $G_p = 10/s^2$ ,  $H = 0.02$ ,  $T = 0.1$ ,  $P_m = 45$ ,  $\omega_{w1} = 1.0$ ).
- 8.11-1.** Consider the chamber temperature control system of Problem 8.4-3. Suppose that a variable gain  $K$  is added to the plant. The pulse transfer function for this system is given in Problem 8.4-3.
- Plot the root locus for this system and find the value of  $K > 0$  for which the system is stable.
  - Find the time constant for the system with  $K = 1$ .
  - Design a phase-lag controller such that the characteristic-equation root is approximately the same as that in part (b) but for the gain  $K = 3$ . *Hint:* Choose  $z_p = 0.999$ .
  - Find the steady-state errors for both the uncompensated system of part (b) and the compensated system of part (c).
  - By computer, verify the design in part (c) by finding the characteristic-equation roots.
- 8.11-2.** Consider the chamber temperature control system of Problem 8.4-3. Suppose that a variable gain  $K$  is added to the plant. The pulse transfer function for this system is given in Problem 8.4-3.
- Plot the root locus for this system, and find the value of  $K > 0$  for which the system is stable.
  - Find the time constant for the system with  $K = 1$ .
  - Assuming the resulting system to be stable, find  $K$  such that the steady-state error in the uncompensated system is 0.05, or 5 percent. Is the resulting system stable? [See part (a).]
  - Design a phase-lag controller such that the characteristic-equation root is approximately the same for the gain  $K$  found in part (c). *Hint:* Choose  $z_p = 0.99999$ .
  - By computer, verify the design in part (d) by finding the characteristic-equation roots.
- 8.11-3.** Consider the system of Problem 8.10-10.
- Plot the root locus for this system, and find the value of  $K > 0$  for which the system is stable.
  - Find the time constant for the system with  $K = 1$ .
  - Design a phase-lead controller such that the characteristic-equation root in part (b) moves to yield a time constant of 0.5 s for  $K = 1$ .
  - By computer, verify the design in part (c) yields the correct characteristic-equation zero.
- 8.11-4.** Consider the system of Fig. P8.2-2 with a gain factor  $K$  added to the plant.
- Show that the pulse transfer function is given by
 
$$\frac{z - 1}{z} \mathcal{Z} \left[ \frac{1}{s^2(s + 1)} \right] = \frac{0.368(z + 0.717)}{(z - 1)(z - 0.368)}$$
  - Sketch the root locus for this system, and find the value of  $K$  that results in critical damping, that is, in two real and equal roots of the system characteristic equation.
  - Find the time constants of the two poles in part (b).
  - It is desired that the gain in part (b) be multiplied by 1.8, while the characteristic-equation roots remain approximately the same. Design a phase-lag controller that meets these specifications. *Hint:* Use  $z_p = 0.999$ .
  - By computer, verify the characteristic-equation roots in part (d).
- 8.11-5.** Consider the system of Example 8.7.
- Design a phase-lag compensator such that with  $K = 0.5$ , the system is critically damped with roots having a time constant of approximately 2.03 s. This time constant is the same as that of the uncompensated system with  $K = 0.244$ .
  - By computer, verify the characteristic-equation roots in part (a).

# 9

## Pole-Assignment Design and State Estimation

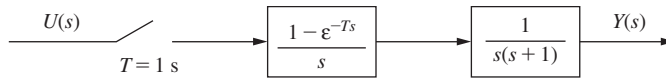
### 9.1 INTRODUCTION

In Chapter 8 we considered discrete control design both from a frequency-response point of view and from a root-locus point of view. In every case we considered only one signal in the feedback loop. It seems reasonable that if we can determine more about the present condition of a system and use this additional information to generate the control input, then we should be able to control the system in a manner that is, in some sense, better. Of course, we have a complete description of the condition of a system if we measure its state vector. Hence we might conclude that if we are able to specify mathematically what defines the very best control system, the design of this control system might require that we have all the states of the system available for feedback. Generally, the design of systems of this type does require that we assume that the full-state vector is available for feedback.

For most control systems the measurement of the full-state vector is impractical. To implement a design based on full-state feedback, we must estimate the states of a system using measurements that are practical. The state estimation is accomplished by designing a dynamic system (a set of equations) for the computer that estimates the states using all information available. Fortunately, we can separate the design into two phases. During the first phase, we design the system as though all states of the system will be measured. The second phase is then the design of the state estimator. In this chapter we consider both phases of the design process and the effects that the state estimator has on closed-loop system operation. Then, in Chapter 11, after the introduction of the appropriate mathematics, we consider the Kalman filter, which is an optimal state estimator.

### 9.2 POLE ASSIGNMENT

In this section a design procedure generally known as pole assignment, or pole placement, is developed. The design results in the assignment of the poles of the closed-loop transfer function (zeros of the characteristic equation) to any desired locations. There are, of course, practical implications that will be discussed as the technique is developed.



**FIGURE 9-1** Servomotor system.

To introduce the pole-assignment technique, we will consider the model of a servomotor, developed in Chapter 1. This model is second-order, and an example is shown in Fig. 9-1. The state model for this system was calculated following the procedure used in Example 4.13 and is given by

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.0048374 \\ 0.095163 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k) \end{aligned} \quad (9-1)$$

In this model  $x_1(k)$  is the position (angle) of the motor shaft, which can easily be measured. The state  $x_2(k)$  is shaft velocity, which can be measured using a tachometer or some other suitable sensor. Thus for this system the full-state vector can be measured.

We choose the control input  $u(k)$  to be a linear combination of the states; that is,

$$u(k) = -K_1 x_1(k) - K_2 x_2(k) = -\mathbf{K} \mathbf{x}(k) \quad (9-2)$$

where the gain matrix  $\mathbf{K}$  is

$$\mathbf{K} = [K_1 \quad K_2]$$

Then (9-1) can be written as

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix} \mathbf{x}(k) - \begin{bmatrix} 0.0048374 \\ 0.095163 \end{bmatrix} [K_1 x_1(k) + K_2 x_2(k)] \\ &= \begin{bmatrix} 1 - 0.0048374 K_1 & 0.095163 - 0.0048374 K_2 \\ -0.095163 K_1 & 0.904837 - 0.095163 K_2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \end{aligned} \quad (9-3)$$

The matrix of (9-3) is the closed-loop system matrix, which we will call  $\mathbf{A}_f$ . The closed-loop system equation is then

$$\mathbf{x}(k+1) = \mathbf{A}_f \mathbf{x}(k) \quad (9-4)$$

and the characteristic equation is

$$\alpha(z) = |z\mathbf{I} - \mathbf{A}_f| = 0 \quad (9-5)$$

Evaluation of (9-5) yields, after some calculations (see the MATLAB program below), the characteristic equation

$$\begin{aligned} z^2 + (0.0048374 K_1 + 0.095163 K_2 - 1.904837)z + \\ 0.0046788 K_1 - 0.095163 K_2 + 0.904837 = 0 \end{aligned} \quad (9-6)$$



Suppose that, by some process, we choose the desired characteristic-equation zero locations to be  $\lambda_1$  and  $\lambda_2$ . Then the desired characteristic polynomial, denoted  $\alpha_c(z)$ , is given by

$$\alpha_c(z) = (z - \lambda_1)(z - \lambda_2) = z^2 - (\lambda_1 + \lambda_2)z + \lambda_1\lambda_2 \quad (9-7)$$

Equating coefficients in (9-6) and (9-7) yields the equations

$$\begin{aligned} 0.0048374K_1 + 0.095163K_2 &= -(\lambda_1 + \lambda_2) + 1.904837 \\ 0.0046788K_1 - 0.095163K_2 &= \lambda_1\lambda_2 - 0.904837 \end{aligned} \quad (9-8)$$

These equations are linear in  $K_1$  and  $K_2$ , and upon solving yield

$$\begin{aligned} K_1 &= 105.0833[\lambda_1\lambda_2 - (\lambda_1 + \lambda_2) + 1.0] \\ K_2 &= 14.67494 - 5.34172\lambda_1\lambda_2 - 5.16661(\lambda_1 + \lambda_2) \end{aligned} \quad (9-9)$$

Thus we can find the gain matrix  $\mathbf{K}$  that will realize *any* desired characteristic equation.

We wish now to make the following points. By some process we choose the root locations so as to satisfy design criteria such as speed of response, overshoot in the transient response, and the like (see Section 6.4). However, at this time no input has been shown; hence we cannot speak of overshoot to a step input. Once we have chosen  $\lambda_1$  and  $\lambda_2$ , (9-9) will give the gain matrix needed to realize these characteristic-equation zeros. Then, to the extent that (9-1) is an accurate model of the physical servomotor, we will realize these zeros in the physical system. Obviously, we cannot increase the speed of response of a servo system without limit, even though (9-9) seems to indicate that we can. If we attempt to force the system to respond too rapidly, large signals will be generated and the plant will enter a nonlinear mode of operation; then (9-1) will no longer accurately model the plant. Hence, in choosing  $\lambda_1$  and  $\lambda_2$ , we must consider only those root locations that can be attained by the physical system.

We will now develop a general procedure of pole assignment. Our  $n$ th-order plant is modeled by

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \quad (9-10)$$

We generate the control input  $u(k)$  by the relationship

$$u(k) = -\mathbf{K}\mathbf{x}(k) \quad (9-11)$$

where

$$\mathbf{K} = [K_1 \ K_2 \ \cdots \ K_n] \quad (9-12)$$

Then (9-10) can be written as

$$\mathbf{x}(k + 1) = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}(k) \quad (9-13)$$

and, in (9-4),  $\mathbf{A}_f = (\mathbf{A} - \mathbf{B}\mathbf{K})$ . We choose the desired pole locations

$$z = \lambda_1, \lambda_2, \dots, \lambda_n \quad (9-14)$$

Then the closed-loop system characteristic polynomial is

$$\alpha_c(z) = |z\mathbf{I} - \mathbf{A} + \mathbf{BK}| = (z - \lambda_1)(z - \lambda_2) \cdots (z - \lambda_n) \quad (9-15)$$

In this equation there are  $n$  unknowns,  $K_1, K_2, \dots, K_n$ , and  $n$  known coefficients in the right-hand-side polynomial. We can solve for the unknown gains by equating coefficients in (9-15), as illustrated by the servomotor example above. An example will now be given to illustrate this procedure.

### EXAMPLE 9.1

In this example the servomotor system of Fig. 9-1, with the state equations given in (9-1), will be considered. With full-state feedback, the characteristic equation is given in (9-6). Consider first that the closed-loop system is implemented with the usual unity feedback; then  $K_1 = 1$  and  $K_2 = 0$ . From (9-6), the characteristic equation is given by

$$z^2 - 1.90000z + 0.90952 = 0$$

This equation has roots at

$$z_{1,2} = 0.95369 \angle (\pm 0.087944 \text{ rad}) = r \angle (\pm \theta)$$

From (6-8) we calculate the damping factor of these roots to be

$$\zeta = \frac{-\ln r}{\sqrt{(\ln r)^2 + \theta^2}} = \frac{-\ln(0.95369)}{\sqrt{(\ln(0.95369))^2 + (0.087944)^2}} = 0.47462$$

and from (6-10), the time constant is

$$\tau = \frac{-T}{\ln r} = \frac{-0.1}{\ln(0.95369)} = 2.109 \text{ s}$$

Suppose that we decide that this value of  $\zeta$  is satisfactory, but that a time constant of 1.0 s is required. Then

$$\ln r = -\frac{T}{\tau} = -0.1$$

or  $r = 0.904837$ . Solving (6-8) for  $\theta$ , we have

$$\theta^2 = (\ln r)^2 \left( \frac{1}{\zeta^2} - 1 \right) = (\ln(0.904837))^2 \left( \frac{1}{(0.47462)^2} - 1 \right)$$

or  $\theta$  is equal to 0.185453 rad, or 10.63°. Then the desired root locations are

$$z_{1,2} = 0.904837 \angle (\pm 0.185453) = 0.889322 \pm j0.166845$$

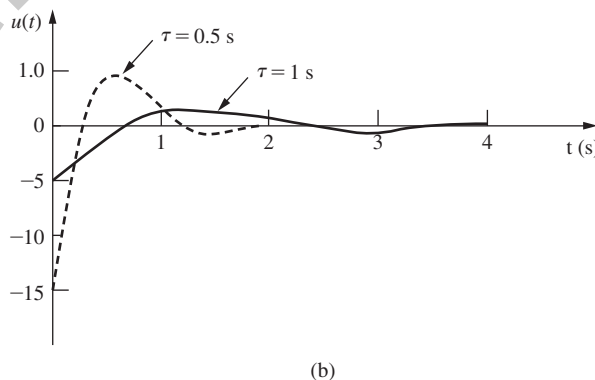
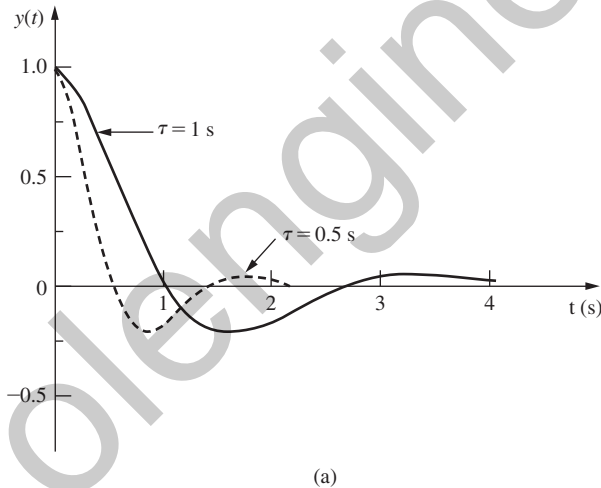
Hence the desired characteristic equation is given by

$$\begin{aligned}
 \alpha_c(z) &= z^2 - 2r\cos\theta z + r^2 \\
 &= z^2 - 2(0.904837)\cos(0.185453)z + (0.904837)^2 \\
 &= z^2 - 1.778644z + 0.818731
 \end{aligned}$$

From (9-9),

$$\begin{aligned}
 K_1 &= 105.0833[\lambda_1\lambda_2 - (\lambda_1 + \lambda_2) + 1.0] \\
 &= 105.0833[0.818731 - (1.77864) + 1.0] = 4.212 \\
 K_2 &= 14.67494 - 5.34172\lambda_1\lambda_2 - 5.16661(\lambda_1 + \lambda_2) \\
 &= 14.67494 - 5.34172(0.818731) - 5.16661(1.77864) = 1.112
 \end{aligned}$$

The initial-condition response of this system is given in Fig. 9-2, with  $x_1(0) = 1.0$  and  $x_2(0) = 0$ . Both  $y(t)$ , the output (Fig. 9-2(a)), and  $u(t)$ , the sampler input (Fig. 9-2(b)), are shown.



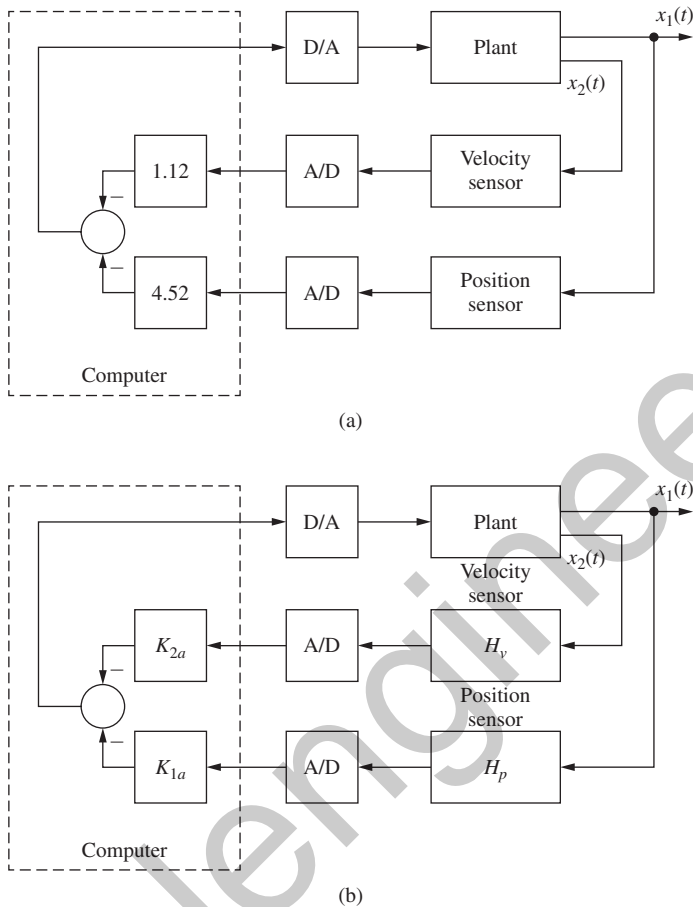
**FIGURE 9-2** Initial-condition responses for the systems of Example 9.1.

Suppose that we decide that the foregoing value of  $\zeta = 0.475$  is satisfactory, but that a time constant of  $\tau = 0.5$  s is required. This design is given as Problem 9.2-1. The gains obtained from this design are  $K_1 = 15.15$  and  $K_2 = 3.21$ , and the initial-condition response is also given in Fig. 9-2. Compare the control inputs  $u(t)$  for the two designs in Fig. 9-2. The maximum amplitude of  $u(t)$  for  $\tau = 1.0$  s is 4.21, while that for  $\tau = 0.5$  s is 15.15. Thus we see that an attempt to increase the speed of response of the system results in larger signals at the plant input. These larger signals may force the system into a nonlinear region of operation, which in some cases may be undesirable.

A MATLAB program to perform the computations needed for this example is given below. Note that the format *long* is needed to display the parameters in sufficient precision.

```
% First verify the closed loop roots with K1=1, K2=0
format long
disp(' ')
disp('Using MATLAB damp function')
Gp = tf([1],[1 1 0]); T = 0.1; Gz = c2d(Gp,T);
Cz = feedback(Gz,1); Cz_zpk = zpk(Cz); [Wn,zeta,P]= damp(Cz)
disp(' ')
disp('Given system roots and gains K1 & K2:')
% Using the format of Equation (1-18)
Ac =[0 1;0 -1]; Bc =[0;1]; Cc =[1 0]; Dc=0;
Sysc = ss(Ac,Bc,Cc,Dc); T = 0.1; Sysd = c2d(Sysc,T);
[A,B,C,D]=ssdata(Sysd);
a12=A(1,2); a22=A(2,2); b11=B(1,1); b21=B(2,1); K1=1; K2=0;
% Find alphac using Equation (9-6)
C1=b11, C2=b21, C3=-1-a22 % For Equation (9-6)
C4=-a22*b11+a12*b21, C5=-b21, C6=a22 % For Equation (9-6)
alphac =[1 C1*K1+C2*K2+C3 C4*K1+C5*K2+C6]
% alphac =[1 b11*K1+b21*K2-1-a22 (-a22*b11+a12*b21)*K1-b21*K2+a22]
root = roots(alphac); real=(root(1)+root(2))/2;
imag = abs(root(1)-root(2))/2; r = sqrt(real^2+imag^2)
r_squared = r^2, theta = atan(imag/real)
two_r_cos_theta = 2*r*cos(theta), theta_deg = theta*180/pi
zeta = -log(r)/sqrt(log(r)^2+theta^2), tau = -T/log(r)
alphac_check = [1 -2*r*cos(theta) r^2]
% Gains check using Equation (9-9): K1=1, K2=0
% Constants for Equation (9-9)
C1=1/(b11-a22*b11+a12*b21), K1 = C1*(r^2-2*r*cos(theta)+1)
X=b11/b21/(b11-a22*b11+a12*b21);
C2=(1+a22)/b21-X, C3=-X, C4=-1/b21+X
K2=C2+C3*r^2+C4*2*r*cos(theta)
% Desired zeta is 0.47462 with new tau of 1.0 s
disp(' ')
tau = input('Desired tau = '); % enter 1.0
disp('Desired new roots and gains K1 & K2:')
r = exp(-T/tau), theta = sqrt(log(r)^2*(1/zeta^2-1))
theta_deg=theta*180/pi
alphac = [1 -2*r*cos(theta) r^2], roots_alphac=roots(alphac)
zeta_check=-log(r)/sqrt(log(r)^2+theta^2), tau_check=-T/log(r)
% Find new K1 and K2 using Equation (9-9)
K1 = C1*(r^2-2*r*cos(theta)+1)
K2=C2+C3*r^2+C4*2*r*cos(theta)
```

The system of Example 9.1 can be realized with the hardware configuration of Fig. 9-3. In Fig. 9-3(a) it is assumed that the gains of the sensors are unity; in the practical case these



**FIGURE 9-3** Hardware implementation for the design of Example 9.1.

gains would probably not be unity. For example, suppose that the sensors for position and velocity can be modeled as the constant gains  $H_p$  and  $H_v$ , respectively, over the closed-loop system bandwidth. The hardware configuration of the system would then be as shown in Fig. 9-3(b), where

$$H_p K_{1a} = 4.212, \quad H_v K_{2a} = 1.112$$

In the procedure above the gain matrix  $\mathbf{K}$  was calculated by equating coefficients in the characteristic equation of (9-15). This calculation is simplified greatly if the state model of the plant is in the control canonical form (see Section 2.9), which is the form

$$\mathbf{x}(k+1) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ & & & \ddots & \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u(k) \quad (9-16)$$

This plant has the characteristic equation, from Section 2.10,

$$\alpha(z) = |z\mathbf{I} - \mathbf{A}| = z^n + a_{n-1}z^{n-1} + \cdots + a_1z + a_0 = 0 \quad (9-17)$$

For this state model, in (9-15)  $\mathbf{BK}$  is equal to

$$\mathbf{BK} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} [K_1 \ K_2 \ \cdots \ K_n] = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ K_1 & K_2 & \cdots & K_n \end{bmatrix} \quad (9-18)$$

Hence the closed-loop system matrix is

$$\mathbf{A} - \mathbf{BK} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -(a_0 + K_1) & -(a_1 + K_2) & \cdots & -(a_{n-1} + K_n) \end{bmatrix} \quad (9-19)$$

and the characteristic equation becomes

$$|z\mathbf{I} - \mathbf{A} + \mathbf{BK}| = z^n + (a_{n-1} + K_n)z^{n-1} + \cdots + (a_1 + K_2)z + (a_0 + K_1) = 0 \quad (9-20)$$

If we write the desired characteristic equation as

$$\alpha_c(z) = z^n + \alpha_{n-1}z^{n-1} + \cdots + \alpha_1z + \alpha_0 = 0 \quad (9-21)$$

we calculate the gains by equating coefficients in (9-20) and (9-21) to yield

$$K_{i+1} = \alpha_i - a_i, \quad i = 0, 1, \dots, n-1 \quad (9-22)$$

In general, the techniques for developing state models for plants do not result in the control canonical form (see Section 4.10). A more practical procedure for calculating the gain matrix  $\mathbf{K}$  is the use of Ackermann's formula [1]. The proof of Ackermann's formula will not be given here (see [1] or [2]), but this proof is based on transformations from a general system matrix  $\mathbf{A}$  to that of the control canonical form.

We begin with the plant model

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \quad (9-23)$$

The matrix polynomial  $\alpha_c(\mathbf{A})$  is formed using the coefficients of the desired characteristic equation (9-21).

$$\alpha_c(\mathbf{A}) = \mathbf{A}^n + \alpha_{n-1}\mathbf{A}^{n-1} + \cdots + \alpha_1\mathbf{A} + \alpha_0\mathbf{I} \quad (9-24)$$

Then Ackermann's formula for the gain matrix  $\mathbf{K}$  is given by

$$\mathbf{K} = [0 \ 0 \ \cdots \ 0 \ 1][\mathbf{B} \ \mathbf{A}\mathbf{B} \ \cdots \ \mathbf{A}^{n-2}\mathbf{B} \ \mathbf{A}^{n-1}\mathbf{B}]^{-1}\alpha_c(\mathbf{A}) \quad (9-25)$$

The problems with the existence of the inverse matrix in (9-25) are discussed in Section 9.6. Example 9.1 will now be solved using Ackermann's formula.

### EXAMPLE 9.2

We will solve the design in Example 9.1 via Ackermann's formula (9-25). The plant model is given by

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) = \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.0048374 \\ 0.095163 \end{bmatrix} u(k)$$

and the desired characteristic equation is

$$\alpha_c(z) = z^2 - 1.778644z + 0.818731$$

Hence

$$\alpha_c(\mathbf{A}) = \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix}^2 - 1.77867 \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix} + 0.818731 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

or

$$\alpha_c(\mathbf{A}) = \begin{bmatrix} 0.040087 & 0.012009 \\ 0 & 0.028078 \end{bmatrix}$$

Also,

$$\mathbf{A}\mathbf{B} = \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix} \begin{bmatrix} 0.0048374 \\ 0.095163 \end{bmatrix} = \begin{bmatrix} 0.01389 \\ 0.08611 \end{bmatrix}$$

Thus

$$[\mathbf{B} \quad \mathbf{A}\mathbf{B}]^{-1} = \begin{bmatrix} 0.0048374 & 0.0138934 \\ 0.095163 & 0.0861070 \end{bmatrix}^{-1} = \begin{bmatrix} -95.082 & 15.342 \\ 105.082 & -5.342 \end{bmatrix}$$

Then the gain matrix is, from (9-25),

$$\begin{aligned} \mathbf{K} &= [0 \quad 1] \begin{bmatrix} -95.082 & 15.342 \\ 105.082 & -5.342 \end{bmatrix} \begin{bmatrix} 0.040087 & 0.012009 \\ 0 & 0.028078 \end{bmatrix} \\ &= [4.212 \quad 1.112] \end{aligned}$$

These results are the same as those obtained in Example 9.1.

A MATLAB program to compute the gain matrix is given by

```

% Use the MATLAB acker function to find gains K
A = [1 0.095163; 0 0.904837]; B = [0.0048374; 0.095163]; T=0.1;
% Set tau and zeta for desired roots of characteristic equation
tau = 1.0; zeta = 0.47462;
r = exp(-T/tau), theta = sqrt(log(r)^2*(1/zeta^2-1))
alphac = [1 -2*r*cos(theta) r^2], roots_alphac=roots(alphac)
    
```

```

% P specifies the desired roots
P = roots(alpha_c)
K = acker(A,B,P) % Ackermann computation
    
```

### 9.3 STATE ESTIMATION

In Section 9.2 a design technique was developed which requires that all the plant states be measured. In general, the measurement of all the plant states is impractical, if not impossible, in all but the simplest of systems. In this section we develop a technique for estimating the states of a plant from the information that is available concerning the plant. The system that estimates the states of another system is generally called an *observer* [3], or a *state estimator*. Thus in this section we introduce the design of observers.

Suppose that the plant is described by the following equation:

$$\begin{aligned}
 \mathbf{x}(k + 1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\
 \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)
 \end{aligned}
 \tag{9-26}$$

where  $\mathbf{y}(k)$  are the plant signals that will be measured. Hence, in (9-26), we know the matrices  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$ , and the signals  $\mathbf{y}(k)$  and  $\mathbf{u}(k)$ . The control inputs  $\mathbf{u}(k)$  are known since we generate them. The problem of observer design can be depicted as shown in Fig. 9-4, where the observer is a set of difference equations to be solved by a digital computer.

The states of the system to be observed are  $\mathbf{x}(k)$ ; the states of the observer are  $\mathbf{q}(k)$  and we desire that  $\mathbf{q}(k)$  be approximately equal to  $\mathbf{x}(k)$ . Since the observer will be implemented on a computer, the signals  $\mathbf{q}(k)$  are then available for feedback calculations.

#### Observer Model

Equations describing observers can be developed in several different ways; we choose here to take a transfer-function approach. The observer design criterion to be used is that the transfer-function matrix from the input  $\mathbf{u}(k)$  to the observer state  $q_i(k)$  be equal to that from the input  $\mathbf{u}(k)$  to the system state  $x_i(k)$ , for  $i = 1, 2, \dots, n$ . Equations describing the observer of Fig. 9-4 will now be developed, based on this reasonable criterion.

First we take the  $z$ -transform of (9-26) and solve for  $\mathbf{X}(z)$ :

$$z\mathbf{X}(z) = \mathbf{A}\mathbf{X}(z) + \mathbf{B}\mathbf{U}(z) \tag{9-27}$$

or

$$\mathbf{X}(z) = (z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(z) \tag{9-28}$$

Now, since the observer has two inputs,  $\mathbf{y}(k)$  and  $\mathbf{u}(k)$ , we write the observer state equations as

$$\mathbf{q}(k + 1) = \mathbf{F}\mathbf{q}(k) + \mathbf{G}\mathbf{y}(k) + \mathbf{H}\mathbf{u}(k) \tag{9-29}$$

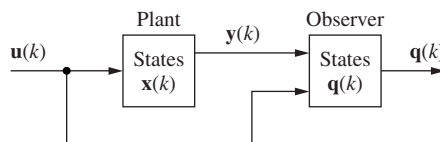


FIGURE 9-4 System illustrating state estimation.



where the matrices  $\mathbf{F}$ ,  $\mathbf{G}$ , and  $\mathbf{H}$  are unknown. Taking the  $z$ -transform of (9-29) and solving for  $\mathbf{Q}(z)$  yields

$$\mathbf{Q}(z) = (z\mathbf{I} - \mathbf{F})^{-1}[\mathbf{G}\mathbf{Y}(z) + \mathbf{H}\mathbf{U}(z)] \quad (9-30)$$

From (9-26),

$$\mathbf{Y}(z) = \mathbf{C}\mathbf{X}(z) \quad (9-31)$$

Next we substitute (9-31) and (9-28) in (9-30):

$$\begin{aligned} \mathbf{Q}(z) &= (z\mathbf{I} - \mathbf{F})^{-1}[\mathbf{G}\mathbf{C}\mathbf{X}(z) + \mathbf{H}\mathbf{U}(z)] \\ &= (z\mathbf{I} - \mathbf{F})^{-1}[\mathbf{G}\mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{H}]\mathbf{U}(z) \end{aligned} \quad (9-32)$$

Recall that the design criterion is that the transfer-function matrix from  $\mathbf{U}(z)$  to  $\mathbf{Q}(z)$  be the same as that from  $\mathbf{U}(z)$  to  $\mathbf{X}(z)$ . Thus, from (9-28),

$$\mathbf{Q}(z) = (z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(z) \quad (9-33)$$

must be satisfied. Then, from (9-32) and (9-33),

$$(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = (z\mathbf{I} - \mathbf{F})^{-1} \mathbf{G}\mathbf{C}(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + (z\mathbf{I} - \mathbf{F})^{-1}\mathbf{H} \quad (9-34)$$

or

$$[\mathbf{I} - (z\mathbf{I} - \mathbf{F})^{-1} \mathbf{G}\mathbf{C}](z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = (z\mathbf{I} - \mathbf{F})^{-1}\mathbf{H} \quad (9-35)$$

This equation can be expressed as

$$(z\mathbf{I} - \mathbf{F})^{-1}[z\mathbf{I} - (\mathbf{F} + \mathbf{G}\mathbf{C})](z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = (z\mathbf{I} - \mathbf{F})^{-1}\mathbf{H}$$

Equating the postmultiplying matrices of  $(z\mathbf{I} - \mathbf{F})^{-1}$  yields

$$(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = [z\mathbf{I} - (\mathbf{F} + \mathbf{G}\mathbf{C})]^{-1}\mathbf{H} \quad (9-36)$$

This equation is satisfied if we choose  $\mathbf{H}$  equal to  $\mathbf{B}$ , and

$$\mathbf{A} = \mathbf{F} + \mathbf{G}\mathbf{C} \quad (9-37)$$

Therefore, from (9-29) and (9-37), we write the observer state equations as

$$\mathbf{q}(k + 1) = (\mathbf{A} - \mathbf{G}\mathbf{C})\mathbf{q}(k) + \mathbf{G}\mathbf{y}(k) + \mathbf{B}\mathbf{u}(k) \quad (9-38)$$

and the design criterion (9-33) is satisfied. Note that  $\mathbf{G}$  is unspecified. Hence the observer is a dynamic system described by (9-38), with  $\mathbf{y}(k)$  and  $\mathbf{u}(k)$  as inputs and with the characteristic equation

$$|z\mathbf{I} - \mathbf{A} + \mathbf{G}\mathbf{C}| = 0 \quad (9-39)$$

We call the observer of (9-38) a *prediction observer*, since the estimate at time  $(k + 1)T$  is based on the measurements at time  $kT$ .

## Errors in Estimation

We now consider the errors in the state-estimation process. Define the error vector  $\mathbf{e}(k)$  as

$$\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{q}(k) \quad (9-40)$$

Then, from (9-40), (9-26), and (9-38),

$$\begin{aligned} \mathbf{e}(k+1) &= \mathbf{x}(k+1) - \mathbf{q}(k+1) \\ &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) - (\mathbf{A} - \mathbf{G}\mathbf{C})\mathbf{q}(k) - \mathbf{G}\mathbf{C}\mathbf{x}(k) - \mathbf{B}\mathbf{u}(k) \end{aligned} \quad (9-41)$$

or

$$\mathbf{e}(k+1) = (\mathbf{A} - \mathbf{G}\mathbf{C})[\mathbf{x}(k) - \mathbf{q}(k)] = (\mathbf{A} - \mathbf{G}\mathbf{C})\mathbf{e}(k) \quad (9-42)$$

Hence the error dynamics have a characteristic equation given by

$$|z\mathbf{I} - (\mathbf{A} - \mathbf{G}\mathbf{C})| = 0 \quad (9-43)$$

which is the same as that of the observer [see (9-38)].

We look next at the sources of the errors in state estimation. First, we have assumed that we have an exact model of the physical system involved. For example, in (9-41) we assumed that the  $\mathbf{A}$  matrix in the plant's state equations is identical to that in the observer's equations. In a practical case we will have system models that only approximate the physical system. However, more effort expended in obtaining an accurate plant model will lead to improved state estimation.

A second source of errors in our state estimation is in the choice of initial conditions for the observer. Since the initial states of the plant will not be known, the initial error  $\mathbf{e}(0)$  will not be zero. However, provided that the observer is stable, this error will tend to zero with increasing time (assuming exact models), with the dynamics given by (9-43).

The third source of errors is the plant disturbances and the sensor errors. The complete state equations of the plant, (9-26), become

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B}_1\mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \end{aligned} \quad (9-44)$$

when the plant disturbance vector  $\mathbf{w}(k)$  and the measurement inaccuracies  $\mathbf{v}(k)$  are considered. Hence, if (9-44) is used in deriving the error state equations (9-42), we obtain

$$\mathbf{e}(k+1) = (\mathbf{A} - \mathbf{G}\mathbf{C})\mathbf{e}(k) + \mathbf{B}_1\mathbf{w}(k) - \mathbf{G}\mathbf{v}(k) \quad (9-45)$$

For this case, the errors in the state estimation will tend to zero if the estimator is stable (we will certainly require this); however, the excitation terms in (9-45) will prevent the errors from reaching a value of zero.

## Error Dynamics

Consider now the observer state equations:

$$[\text{eq. (9-38)}] \quad \mathbf{q}(k+1) = (\mathbf{A} - \mathbf{G}\mathbf{C})\mathbf{q}(k) + \mathbf{G}\mathbf{y}(k) + \mathbf{B}\mathbf{u}(k)$$

All matrices except  $\mathbf{G}$  in this equation are determined by the plant equations (9-26); however, the observer design criterion (9-33) is satisfied independent of the choice of  $\mathbf{G}$ . Note that  $\mathbf{G}$

determines the error dynamics in (9-43). We generally use this fact in choosing  $\mathbf{G}$ . A characteristic polynomial  $\alpha_e(z)$  is chosen for the error dynamics. Note from (9-38) and (9-43) that this is also the characteristic polynomial of the observer. Then

$$\alpha_e(z) = |z\mathbf{I} - (\mathbf{A} - \mathbf{GC})| = z^n + \alpha_{n-1}z^{n-1} + \cdots + \alpha_1z + \alpha_0 \quad (9-46)$$

For the case of a single-output system [i.e.,  $\mathbf{y}(k)$  is a scalar],

$$\mathbf{G} = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_n \end{bmatrix} \quad (9-47)$$

and (9-46) yields  $n$  equations by equating coefficients. Hence the solution of these equations will yield the  $\mathbf{G}$  matrix of (9-47). For this case, Ackermann's equation may also be employed. First we compare (9-46) to (9-15).

[eq. (9-15)]  $\alpha_c(z) = |z\mathbf{I} - (\mathbf{A} - \mathbf{BK})|$

Ackermann's equation for the solution of this equation is given in (9-25).

[eq. (9-25)]  $\mathbf{K} = [0 \ 0 \ \cdots \ 0 \ 1][\mathbf{B} \ \mathbf{AB} \ \cdots \ \mathbf{A}^{n-2}\mathbf{B} \ \mathbf{A}^{n-1}\mathbf{B}]^{-1}\alpha_c(\mathbf{A})$

Thus Ackermann's equation for  $\mathbf{G}$  in (9-46) is seen to be

$$\mathbf{G} = \alpha_e(\mathbf{A}) \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (9-48)$$

(The derivation of this result is given as Problem 9.3-4.)

Thus we see that we can design the observer, once we decide on an appropriate characteristic equation in (9-46). An approach usually suggested for choosing  $\alpha_e(z)$  is to make the observer two to four times faster than the closed-loop control system. The fastest time constant of the closed-loop system, determined from the system characteristic equation,

$$|z\mathbf{I} - (\mathbf{A} - \mathbf{BK})| = 0 \quad (9-49)$$

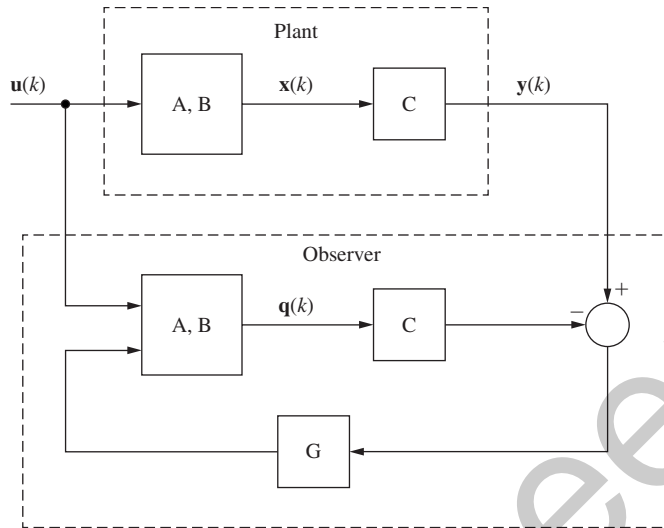
is calculated first. The time constants of the observer are then set at a value equal to from one-fourth to one-half this fastest time constant.

The choice of  $\mathbf{G}$  can be approached from a different viewpoint. The observer state equation, from (9-38), can be written as

$$\mathbf{q}(k + 1) = \mathbf{A}\mathbf{q}(k) + \mathbf{G}[\mathbf{y}(k) - \mathbf{C}\mathbf{q}(k)] + \mathbf{B}\mathbf{u}(k)$$

Thus the plant–observer system can be modeled as in Fig. 9-5.

The choice of  $\mathbf{G}$  can be viewed in the following manner. In Fig. 9-5, if  $\mathbf{x}(k)$  and  $\mathbf{q}(k)$  are approximately equal, there is little effect from the feedback through  $\mathbf{G}$ . Hence  $\mathbf{q}(k)$  is determined principally from  $\mathbf{x}(k)$ . However, if the effects of disturbances on the plant cause  $\mathbf{q}(k)$  to differ significantly from  $\mathbf{x}(k)$ , the effect of  $\mathbf{G}$  is much more important. Here the measurement of  $\mathbf{y}(k)$  is more important in determining  $\mathbf{q}(k)$  than in the first case. Thus we can view the choice of  $\mathbf{G}$  as relating to the relative importance that we attach to the effects of  $\mathbf{u}(k)$  and to the effects of



**FIGURE 9-5** Different view of state estimation.

the disturbances on  $\mathbf{x}(k)$ . And, of course, we will have the effects of measurement noise (inaccuracies) on  $\mathbf{y}(k)$ . If these effects are great, the observer design cannot allow  $\mathbf{y}(k)$  to have a large weight in determining  $\mathbf{q}(k)$ .

We will now summarize the paragraph above. We can view  $\mathbf{G}$  as furnishing a correction input to the plant model in the observer in Fig. 9-5, to account for unknowns in the plant. If the unknowns are significant,  $\mathbf{G}$  should be relatively large. However, if measurement noise in  $\mathbf{y}(k)$  is significant, we cannot rely heavily on the measurement  $\mathbf{y}(k)$ , and  $\mathbf{G}$  should be relatively small. Hence, for practical systems, perhaps the best method for choosing  $\mathbf{G}$  is through the use of an accurate simulation of the system of Fig. 9-5 that includes both the disturbances and the measurement noise. Simulations should be run for different choices of  $\mathbf{G}$  [obtained from different choices of  $\alpha_e(z)$ ], with the final choice of  $\mathbf{G}$  resulting from the best system response.

An example of the design of a prediction observer will now be given.

### EXAMPLE 9.3

We will design an observer for the system of Example 9.2, for which the plant state equations are

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.0048374 \\ 0.095163 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k)$$

With the gain matrix

$$\mathbf{K} = [4.212 \quad 1.112]$$

the closed-loop system characteristic equation is given by

$$\alpha_c(z) = z^2 - 1.778644z + 0.818731$$

The time constant of the roots of this equation, from Example 9.1, is 1.0 s. Thus we will choose the time constant of the observer to be 0.5 s. We also choose the observer to be critically damped, with the roots

$$z = \epsilon^{-T/\tau} = \epsilon^{-0.1/0.5} = 0.818731$$

The observer characteristic equation is then

$$\alpha_e(z) = (z - 0.818731)^2 = z^2 - 1.637462z + 0.670320 = 0$$

The matrix  $\mathbf{G}$  is given by (9-48):

$$\mathbf{G} = \alpha_e(\mathbf{A}) \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Now

$$\begin{aligned} \alpha_e &= \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.0904837 \end{bmatrix}^2 - 1.637462 \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix} + 0.670320 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0.032858 & 0.025444 \\ 0 & 0.007414 \end{bmatrix} \end{aligned}$$

$$\begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 1 & 0.095163 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ -10.508286 & 10.508286 \end{bmatrix}$$

Then

$$\begin{aligned} \mathbf{G} &= \begin{bmatrix} 0.032858 & 0.025444 \\ 0 & 0.007414 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -10.508286 & 10.508286 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 0.267376 \\ 0.077913 \end{bmatrix} \end{aligned}$$

The system matrix of the estimator is given by

$$\mathbf{F} = \mathbf{A} - \mathbf{GC} = \begin{bmatrix} 0.732624 & 0.095163 \\ -0.077913 & 0.904837 \end{bmatrix}$$

From (9-38), the estimator's state equations are

$$\mathbf{q}(k + 1) = \begin{bmatrix} 0.732624 & 0.095163 \\ -0.077913 & 0.904837 \end{bmatrix} \mathbf{q}(k) + \begin{bmatrix} 0.267376 \\ 0.077913 \end{bmatrix} y(k) + \begin{bmatrix} 0.0048374 \\ 0.095163 \end{bmatrix} u(k)$$

But

$$u(k) = -\mathbf{K}\mathbf{q}(k) = -4.212q_1(k) - 1.112q_2(k)$$

Then the estimator's state equations become

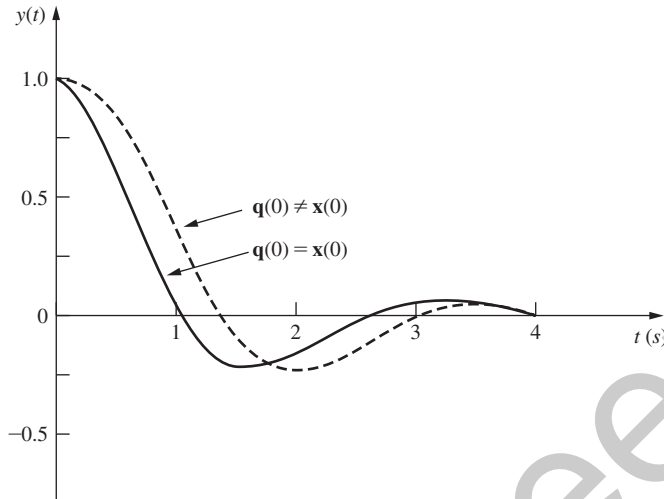


FIGURE 9-6 Initial conditions responses for Example 9.3.

$$\mathbf{q}(k + 1) = \begin{bmatrix} 0.712246 & 0.089784 \\ -0.478776 & 0.799022 \end{bmatrix} \mathbf{q}(k) + \begin{bmatrix} 0.267375 \\ 0.077913 \end{bmatrix} y(k)$$

Initial-condition responses for the observer-based control system are given in Fig. 9-6. Two responses are given: one with

$$\mathbf{q}(0) = \mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and the other with

$$\mathbf{q}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Thus we can see an effect of not knowing the initial state of the system. It should be noted that with  $\mathbf{q}(0)$  and  $\mathbf{x}(0)$  equal, the response of the observer-based control system is identical to that of the full-state feedback system of Example 9.1.

The effects of a certain disturbance, a unit-step function, will now be illustrated. The disturbance enters the system as given in (9-44), with, for example,

$$\mathbf{B}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Fig. 9-7 shows the disturbance response of both the observer-based system and the full-state feedback system with the unit-step disturbance.

Since the observer does not take disturbances into account, the response of the observer-based control system differs significantly from that of the full-state feedback system. A MATLAB program that calculates the observer gains is given below.

```
% Compute the plant's discrete state-space matrices
Ac=[0 1;0 -1]; Bc=[0;1]; Cc=[1 0]; Dc=0;
Sysc=ss (Ac,Bc,Cc,Dc); T=0.1; Sysd=c2d (Sysc,T);
```

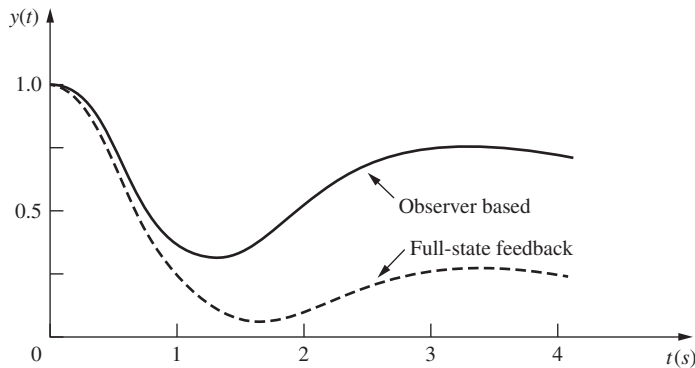


FIGURE 9-7 Disturbance responses for Example 9.3.

```
[A,B,C,D]=ssdata(Sysd);
% Set tau and zeta for desired alphac roots
tau = 1.0; zeta = 0.47462;
r = exp(-T/tau), theta = sqrt(log(r)^2*(1/zeta^2-1))
alphac = [1 -2*r*cos(theta) r^2], roots_alphac=roots(alphac)
P = roots(alphac) % P specifies the acker desired roots
K = acker(A,B,P) % Ackermann computation
% Set tau and zeta for desired alphae roots
tau = 0.5, zeta = 1
r = exp(-T/tau), theta = sqrt(log(r)^2*(1/zeta^2-1))
alphae = [1 -2*r*cos(theta) r^2],
roots_alphae=roots(alphae)
Perr = roots(alphae) % Perr specifies the acker desired roots
G =(acker(A',C',Perr))' % Ackermann computation
```

### Controller Transfer Function

We will now relate pole-assignment-estimator design to controller design by the procedures of Chapter 8. The observer equations are, from (9-38),

[eq.(9-38)]  $\mathbf{q}(k + 1) = (\mathbf{A} - \mathbf{GC})\mathbf{q}(k) + \mathbf{G}y(k) + \mathbf{B}u(k)$

and the control law is given by

$$u(k) = -\mathbf{K}\mathbf{q}(k) \tag{9-50}$$

Substitution of the equation for  $u(k)$  into the observer equations yields

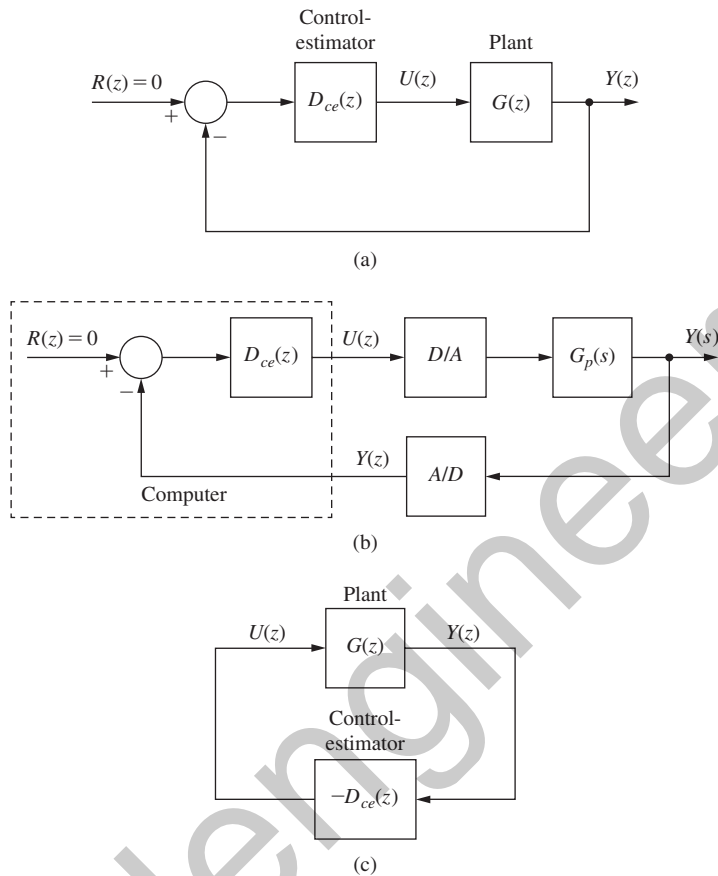
$$\mathbf{q}(k + 1) = (\mathbf{A} - \mathbf{GC} - \mathbf{BK})\mathbf{q}(k) + \mathbf{G}y(k)$$

Taking the  $z$ -transform of this equation and solving for  $\mathbf{Q}(z)$  yields

$$\mathbf{Q}(z) = (z\mathbf{I} - \mathbf{A} + \mathbf{BK} + \mathbf{GC})^{-1}\mathbf{G}Y(z)$$

Substituting this equation into that for  $U(z)$  results in the relationship

$$U(z) = -\mathbf{K}(z\mathbf{I} - \mathbf{A} + \mathbf{BK} + \mathbf{GC})^{-1}\mathbf{G}Y(z)$$



**FIGURE 9-8** Digital-controller realization of the system.

Hence we can consider the controller-observer combination to be a digital controller with the transfer function

$$D_{ce}(z) = -\frac{U(z)}{Y(z)} = \mathbf{K}(z\mathbf{I} - \mathbf{A} + \mathbf{BK} + \mathbf{GC})^{-1}\mathbf{G} \tag{9-51}$$

and model the closed-loop system as shown in Fig. 9-8(a).

Note that, for this controller,  $-Y(z)$  is the *input*, and  $U(z)$  is the *output*. In Fig. 9-8(a) we show the control system as an equivalent unity-gain feedback system, with the system input equal to zero. Fig. 9-8(b) gives the hardware configuration. We can also view the system with no input shown, as in Fig. 9-8(c). The system characteristic equation can be expressed as

$$1 + D_{ce}(z)G(z) = 0 \tag{9-52}$$

We will now determine the equivalent digital controller transfer function  $D_{ce}(z)$  for the design of Example 9.3.



### EXAMPLE 9.4

The system equations for the system of Example 9.3 are

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.0048374 \\ 0.095163 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k)$$

with control gains

$$\mathbf{K} = [4.212 \quad 1.112]$$

In Example 9.3 the observer equations were calculated to be

$$\mathbf{q}(k+1) = (\mathbf{A} - \mathbf{BK} - \mathbf{GC})\mathbf{q}(k) + \mathbf{G}y(k)$$

$$= \begin{bmatrix} 0.712247 & 0.089784 \\ -0.478776 & 0.799022 \end{bmatrix} \mathbf{q}(k) + \begin{bmatrix} 0.267376 \\ 0.077913 \end{bmatrix} y(k)$$

$$u(k) = -\mathbf{K}\mathbf{q}(k) = -[4.21240 \quad 1.11195]\mathbf{q}(k)$$

Now, in (9-51),

$$(z\mathbf{I} - \mathbf{A} + \mathbf{BK} + \mathbf{GC})^{-1} = \begin{bmatrix} z - 0.712247 & -0.089784 \\ 0.478785 & z - 0.799021 \end{bmatrix}^{-1}$$

and

$$\Delta = |z\mathbf{I} - \mathbf{A} + \mathbf{BK} + \mathbf{GC}| = z^2 - 1.51127z + 0.612087$$

The digital controller transfer function is, from (9-51),

$$D_{ce}(z) = \mathbf{K}(z\mathbf{I} - \mathbf{A} + \mathbf{BK} + \mathbf{GC})^{-1}\mathbf{G}$$

$$= [4.21246 \quad 1.11195] \frac{1}{\Delta} \begin{bmatrix} z - 0.79902 & 0.089784 \\ -0.47874 & z - 0.712245 \end{bmatrix} \begin{bmatrix} 0.26738 \\ 0.077913 \end{bmatrix}$$

$$= \frac{1}{\Delta} [4.21246z - 3.89817 \quad 1.11195z - 0.41377] \begin{bmatrix} 0.26738 \\ 0.077913 \end{bmatrix}$$

$$= \frac{1.2129(z - 0.8859)}{z^2 - 1.511z + 0.6121}$$

It is seen that a second-order digital controller has been designed for this system by using a modern control approach. MATLAB code for this example follows.

```

% Append this code to the program for Example 9.3
Aq = A-B*K-G*C, Bq = G, Cq = -K
[NUMce, DENce]=ss2tf(Aq,Bq,Cq,0);
Dce=tf(-NUMce, DENce,0.1);
Dce=zpk(Dce)
    
```

```

% Alternative method using (9-51)
% Store this NICE function
function [nr2,dr2]= nice(old,z)
% NICE takes the original expression and makes it look nice
% Input: (a*z^2 + b*z + c)/(d*z^2 + e*z + f)
% Output: (a1*z^2 + b1*z + c1)/(z^2 + e1*z + f1)
% old contains symbol of z
% rearrange the expression to collect coefficients
new = collect(old,z);
% obtain the coefficients of numerator and denominator
dr = fliplr(coeffs(feval(symengine,'denom',new)));
nr = fliplr(coeffs(feval(symengine,'numer',new)));
% normalize the coefficients with the coeff of dr z^2 term
dr2 = double(dr/dr(1)); nr2 = double(nr/dr(1));
end

syms z, Dce=K*inv((z*eye(2)-A+B*K+G*C))*G;
[num,den]=nice(Dce,z);
Dce=tf(num,den,T); Dce=zpk(Dce)
    
```

### Closed-Loop Characteristic Equation

As a final point, we must investigate the effects on the closed-loop system characteristic equation of the addition of the observer. For full-state feedback, the characteristic equation is given by

$$\alpha_c(z) = |z\mathbf{I} - \mathbf{A} + \mathbf{BK}| = 0$$

We will now derive the system characteristic equation for the observer-based control system.

To derive this equation, we will use the error variables of (9-40).

[eq. (9-40)]  $\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{q}(k)$

The plant state equations of (9-26) can be expressed as

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) - \mathbf{BK}\mathbf{q}(k) = (\mathbf{A} - \mathbf{BK})\mathbf{x}(k) + \mathbf{BK}\mathbf{e}(k) \quad (9-53)$$

The state equations for the error variables are

[eq. (9-42)]  $\mathbf{e}(k+1) = (\mathbf{A} - \mathbf{GC})\mathbf{e}(k)$

We can adjoin the variables of (9-53) and (9-42) into a single state vector, with the resulting equations

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{e}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{BK} & \mathbf{BK} \\ \mathbf{0} & \mathbf{A} - \mathbf{GC} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{e}(k) \end{bmatrix} \quad (9-54)$$

with the states of the observer a linear combination of  $\mathbf{x}(k)$  and  $\mathbf{e}(k)$ , given in (9-40).

[eq. (9-40)]  $\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{q}(k)$

Thus the characteristic equation of the state equations of (9-54) is also the closed-loop system characteristic equation. This equation is seen to be

$$|z\mathbf{I} - \mathbf{A} + \mathbf{BK}| |z\mathbf{I} - \mathbf{A} + \mathbf{GC}| = \alpha_c(z)\alpha_e(z) = 0 \quad (9-55)$$

We see then that the roots of the characteristic equation of the closed-loop system are the roots obtained by the pole-placement design plus those of the observer. Hence the pole-placement design is independent of the observer design.

### EXAMPLE 9.5

We will now calculate the characteristic equation for the system of Example 9.4. Now

$$G(z) = \frac{z-1}{z} \mathfrak{Z} \left[ \frac{1}{s^2(s+1)} \right] = \frac{0.004837z + 0.004679}{z^2 - 1.905z + 0.9048}$$

Hence, from Example 9.4 and Fig. 9-8, the closed-loop system characteristic equation can be expressed as

$$1 + D_{ce}(z)G(z) = 1 + \left( \frac{1.2129(z - 0.8859)}{z^2 - 1.511z + 0.6121} \right) \left( \frac{0.004837z + 0.004679}{z^2 - 1.905z + 0.9048} \right) = 0$$

This equation can be evaluated as

$$z^4 - 3.416z^3 + 4.401z^2 - 2.533z + 0.5488 = 0$$

The product  $\alpha_c(z)\alpha_e(z)$  is, from Examples 9.1 and 9.3,

$$\alpha_c(z)\alpha_e(z) = (z^2 - 1.7786z + 0.8187)(z^2 - 1.6375z + 0.6703)$$

and expansion of this equation yields the fourth-order polynomial above.

### Closed-Loop State Equations

The transfer-function model of the closed-loop system is given in Fig. 9-8. A state model of the closed-loop system will now be derived. From (9-26) and (9-50),

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) - \mathbf{BK}\mathbf{q}(k)$$

and from (9-26), (9-38), and (9-50),

$$\mathbf{q}(k+1) = \mathbf{GC}\mathbf{x}(k) + (\mathbf{A} - \mathbf{GC} - \mathbf{BK})\mathbf{q}(k)$$

We adjoin the foregoing two equations to form the closed-loop state model.

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{q}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{GC} & \mathbf{A} - \mathbf{GC} - \mathbf{BK} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{q}(k) \end{bmatrix} \quad (9-56)$$

This model is useful in writing simulations of the closed-loop system. If the disturbance inputs and the sensor-noise inputs are added as in (9-44), the closed-loop model is more complex (see Problem 9.3-5).

### EXAMPLE 9.6

The closed-loop state matrix of (9-56) will now be calculated for the system of Example 9.5. Now, from Example 9.3,

$$\mathbf{BK} = \begin{bmatrix} 0.0048374 \\ 0.095163 \end{bmatrix} [4.21246 \quad 1.11195] = \begin{bmatrix} 0.020377 & 0.0053790 \\ 0.40087 & 0.10582 \end{bmatrix}$$

$$\mathbf{GC} = \begin{bmatrix} 0.26738 \\ 0.077913 \end{bmatrix} [1 \quad 0] = \begin{bmatrix} 0.26738 & 0 \\ 0.077913 & 0 \end{bmatrix}$$

From Example 9.4,

$$\mathbf{A} - \mathbf{GC} - \mathbf{BK} = \begin{bmatrix} 0.71225 & 0.089784 \\ -0.47878 & 0.79902 \end{bmatrix}$$

The closed-loop system matrix, from (9-56), is then

$$\begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{GC} & \mathbf{A} - \mathbf{GC} - \mathbf{BK} \end{bmatrix} = \begin{bmatrix} 1 & 0.09516 & -0.02038 & -0.00538 \\ 0 & 0.90484 & -0.40087 & -0.10582 \\ 0.26738 & 0 & 0.71225 & 0.08978 \\ 0.07791 & 0 & -0.47878 & 0.79902 \end{bmatrix}$$

A MATLAB statement to generate this closed-loop matrix is `[A -B*K;G*C A-G*C-B*K]`. Of course, the characteristic equation of this matrix is the same as that of (9-52) and (9-55). The interested reader may verify that the closed-loop system matrix calculated in this example has the same characteristic equation as (9-52), which was calculated in Example 9.5.

This section has presented a brief introduction to observers. References [2], [4], [5], and [6] expand on the developments of this section.

## 9.4 REDUCED-ORDER OBSERVERS

In the example in Section 9.3, we estimated  $x_1(k)$  [position] and  $x_2(k)$  [velocity], given the measurement of position. However, if an accurate measurement of position is available, it is not necessary to estimate position; we already know it. In general, if accurate measurements of certain states are made, one needs to estimate only the remaining states, and the accurately measured signals are then used directly for feedback. The resulting observer is called a *reduced-order* observer. However, if the measurements are relatively inaccurate (noisy), the full-order observer may yield better results.

To develop the design equations for the reduced-order observer, we first partition the state vector as

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_a(k) \\ \mathbf{x}_b(k) \end{bmatrix}$$

where  $\mathbf{x}_a(k)$  are the states to be measured and  $\mathbf{x}_b(k)$  are the states to be estimated.

Then the plant state equation of (9-26) can be partitioned as

$$\begin{bmatrix} \mathbf{x}_a(k+1) \\ \mathbf{x}_b(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{aa} & \mathbf{A}_{ab} \\ \mathbf{A}_{ba} & \mathbf{A}_{bb} \end{bmatrix} \begin{bmatrix} \mathbf{x}_a(k) \\ \mathbf{x}_b(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_a \\ \mathbf{B}_b \end{bmatrix} \mathbf{u}(k)$$

$$\mathbf{y}(k) = [\mathbf{I} \quad 0] \begin{bmatrix} \mathbf{x}_a(k) \\ \mathbf{x}_b(k) \end{bmatrix} \quad (9-57)$$

Note that in this case we are considering both multiple inputs and multiple outputs.

The equations for the measured states can be written as

$$\mathbf{x}_a(k+1) = \mathbf{A}_{aa} \mathbf{x}_a(k) + \mathbf{A}_{ab} \mathbf{x}_b(k) + \mathbf{B}_a \mathbf{u}(k)$$

Collecting all the known terms on the left side of the equation, we write

$$\mathbf{x}_a(k+1) - \mathbf{A}_{aa} \mathbf{x}_a(k) - \mathbf{B}_a \mathbf{u}(k) = \mathbf{A}_{ab} \mathbf{x}_b(k) \quad (9-58)$$

For the reduced-order observer we consider the left side to be the “known measurements.” From (9-57), the equations for the estimated states are

$$\mathbf{x}_b(k+1) = \mathbf{A}_{ba} \mathbf{x}_a(k) + \mathbf{A}_{bb} \mathbf{x}_b(k) + \mathbf{B}_b \mathbf{u}(k) \quad (9-59)$$

The term  $[\mathbf{A}_{ba} \mathbf{x}_a(k) + \mathbf{B}_b \mathbf{u}(k)]$  is then considered to be the “known inputs.” We now compare the state equations for the full-order observer to those for the reduced-order observer.

[eq. (9-26)] 
$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$$

[eq. (9-59)] 
$$\mathbf{x}_b(k+1) = \mathbf{A}_{bb} \mathbf{x}_b(k) + [\mathbf{A}_{ba} \mathbf{x}_a(k) + \mathbf{B}_b \mathbf{u}(k)]$$

and

[eq. (9-26)] 
$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k)$$

[eq. (9-58)] 
$$\mathbf{x}_a(k+1) - \mathbf{A}_{aa} \mathbf{x}_a(k) - \mathbf{B}_a \mathbf{u}(k) = \mathbf{A}_{ab} \mathbf{x}_b(k)$$

We then obtain the reduced-order observer equations by making the following substitutions into the full-order observer equations (9-38):

$$\mathbf{x}(k) \leftarrow \mathbf{x}_b(k)$$

$$\mathbf{A} \leftarrow \mathbf{A}_{bb}$$

$$\mathbf{B}\mathbf{u}(k) \leftarrow \mathbf{A}_{ba} \mathbf{x}_a(k) + \mathbf{B}_b \mathbf{u}(k)$$

$$\mathbf{y}(k) \leftarrow \mathbf{x}_a(k+1) - \mathbf{A}_{aa} \mathbf{x}_a(k) - \mathbf{B}_a \mathbf{u}(k)$$

$$\mathbf{C} \leftarrow \mathbf{A}_{ab}$$

If we make these substitutions into (9-38),

[eq. (9-38)] 
$$\mathbf{q}(k+1) = (\mathbf{A} - \mathbf{GC})\mathbf{q}(k) + \mathbf{G}\mathbf{y}(k) + \mathbf{B}\mathbf{u}(k)$$

we obtain the equations

$$\begin{aligned} \mathbf{q}_b(k+1) &= (\mathbf{A}_{bb} - \mathbf{G}\mathbf{A}_{ab})\mathbf{q}_b(k) + \mathbf{G}[\mathbf{x}_a(k+1) - \mathbf{A}_{aa}\mathbf{x}_a(k) - \mathbf{B}_a\mathbf{u}(k)] \\ &\quad + \mathbf{A}_{ba}\mathbf{x}_a(k) + \mathbf{B}_b\mathbf{u}(k) \end{aligned} \quad (9-60)$$

From (9-57),

$$\mathbf{y}(k) = \mathbf{x}_a(k) \quad (9-61)$$

Then (9-60) can be written as

$$\begin{aligned} \mathbf{q}_b(k+1) &= (\mathbf{A}_{bb} - \mathbf{G}\mathbf{A}_{ab})\mathbf{q}_b(k) + \mathbf{G}\mathbf{y}(k+1) + (\mathbf{A}_{ba} - \mathbf{G}\mathbf{A}_{aa})\mathbf{y}(k) \\ &\quad + (\mathbf{B}_b - \mathbf{G}\mathbf{B}_a)\mathbf{u}(k) \end{aligned} \quad (9-62)$$

The following points should be made about the reduced-order observer. The observer characteristic equation is

$$\alpha_e(z) = |z\mathbf{I} - \mathbf{A}_{bb} + \mathbf{G}\mathbf{A}_{ab}| = 0 \quad (9-63)$$

and  $\mathbf{G}$  is determined from (9-63) in the same manner that the  $\mathbf{G}$  matrix is obtained from (9-46) for the full-order observer. For the case of a single measurement [i.e.,  $y(k)$  is  $x_1(k)$ ] Ackermann's formula is given by

$$\mathbf{G} = \alpha_e(\mathbf{A}_{bb}) \begin{bmatrix} \mathbf{A}_{ab} \\ \mathbf{A}_{ab}\mathbf{A}_{bb} \\ \vdots \\ \mathbf{A}_{ab}\mathbf{A}_{bb}^{n-2} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (9-64)$$

Note also that, in (9-62), the measurements  $\mathbf{y}(k+1)$  is required to estimate  $\mathbf{q}_b(k+1)$ . However, for the full-order observer, in (9-38),  $\mathbf{q}(k+1)$  is estimated using only the measurements  $\mathbf{y}(k)$ .

As in the case of the full-order prediction observer, we can derive the transfer function  $D_{ce}(z)$ , in Fig. 9-8, of the equivalent digital controller. Since the derivation of the transfer function follows the same procedure as in the full-order observer case, this derivation is given as Problem 9.4-4. It is assumed that  $y(k) = x_1(k)$ , and that the gain matrix  $\mathbf{K}$  is partitioned as

$$\begin{aligned} \mathbf{u}(k) &= -\mathbf{K}[\mathbf{y}(k) \mathbf{q}_b(k)]^T = -[\mathbf{K}_1 \quad \mathbf{K}_b][\mathbf{y}(k) \mathbf{q}_b(k)]^T \\ &= -\mathbf{K}_1\mathbf{y}(k) - \mathbf{K}_b\mathbf{q}_b(k) \end{aligned} \quad (9-65)$$

where  $[\cdot]^T$  indicates the transpose of the matrix  $[\cdot]$ . The derivation results in the transfer function

$$\begin{aligned} D_{ce}(z) &= \frac{-U(z)}{Y(z)} \\ &= \mathbf{K}_1 + \mathbf{K}_b[z\mathbf{I} - \mathbf{A}_{bb} + \mathbf{G}\mathbf{A}_{ab} + (\mathbf{B}_b - \mathbf{G}\mathbf{B}_a)\mathbf{K}_b]^{-1} \\ &\quad \cdot [\mathbf{G}z + \{\mathbf{A}_{ba} - \mathbf{G}\mathbf{A}_{aa} - \mathbf{K}_1(\mathbf{B}_b - \mathbf{G}\mathbf{B}_a)\}] \end{aligned} \quad (9-66)$$

An example of the reduced-order estimator will now be given.

### EXAMPLE 9.7

We will again consider the design of the system of Example 9.1. The plant model is given by

$$\mathbf{x}(k + 1) = \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.0048374 \\ 0.095163 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k)$$

with the gain matrix

$$\mathbf{K} = [4.212 \quad 1.112]$$

and the closed-loop system characteristic equation

$$\alpha_c(z) = z^2 - 1.778644z + 0.818731$$

We are measuring position,  $x_1(k)$ , and will estimate velocity,  $x_2(k)$ . In Example 9.3 we chose the estimator characteristic-equation roots to be at  $z = 0.819$ ; we will make the same choice here. However, the reduced-order observer is first order; hence

$$\alpha_e(z) = z - 0.818731 = 0$$

From the plant state equations above, and (9-57), the partitioned matrices are seen to be

$$\begin{aligned} \mathbf{A}_{aa} &= 1 & \mathbf{A}_{ab} &= 0.095163 & \mathbf{B}_a &= 0.0048374 \\ \mathbf{A}_{ba} &= 0 & \mathbf{A}_{bb} &= 0.904837 & \mathbf{B}_b &= 0.095163 \end{aligned}$$

Thus we have all the terms required for Ackermann's formula in (9-64).

$$\begin{aligned} \mathbf{G} &= \alpha_e(\mathbf{A}_{bb})[\mathbf{A}_{ab}]^{-1}[1] = [0.904837 - 0.818731][0.095163]^{-1}[1] \\ &= 0.9048 \end{aligned}$$

The observer equation is given by (9-62):

$$\begin{aligned} q(k + 1) &= [0.904837 - (0.9048)(0.095163)]q(k) + 0.9048 y(k + 1) \\ &\quad + [0 - (0.9048)(1)]y(k) + [0.095163 - (0.9048)(0.0048374)]u(k) \end{aligned}$$

or

$$q(k + 1) = 0.8187q(k) + 0.9048y(k + 1) - 0.9048y(k) + 0.0908u(k)$$

Here  $q(k)$  is the estimate of velocity,  $x_2(k)$ . Since we have considered the measurement  $y(k)$  to be the measurement at the present time, the implementation of the observer is more obvious if we replace  $k$  with  $(k - 1)$  in the observer equation:

$$q(k) = 0.8187q(k - 1) + 0.9048y(k) - 0.9048y(k - 1) + 0.0908u(k - 1)$$

and  $q(k)$  is the estimate at the present time. From Example 9.1 the control law is given by

$$u(k) = -4.212x_1(k) - 1.112x_2(k)$$

which is implemented as

$$u(k) = -4.212y(k) - 1.112q(k)$$

Hence we can write the observer equation as

$$q(k + 1) = 0.8187q(k) + 0.9048y(k + 1) - 0.9048y(k) + 0.0908[-4.212y(k) - 1.112q(k)]$$

or

$$q(k + 1) = 0.7177q(k) + 0.9048y(k + 1) - 1.2872y(k)$$

The control system is then implemented as follows. A measurement  $y(k)$  is made at  $t = kT$ . The observer state is calculated from

$$q(k) = 0.7177q(k - 1) + 0.9048y(k) - 1.2872y(k - 1)$$

Then the control input is calculated, using

$$u(k) = -4.212y(k) - 1.112q(k)$$

The initial-condition response, obtained by simulation, is approximately the same as that obtained using full-state feedback. In addition, the effects of the disturbance input are less than those for the full-order observer system.

The observer gain for this example is calculated by the MATLAB program

```
A = [1 0.095163; 0 0.904837]; B = [0.0048374; 0.095163];
alphae = [1 -0.818731];
Aaa = A(1,1); Aab = A(1,2); Aaa = A(2,1); Aab = A(2,2);
Ba=B(1,1); Bb=B(2,1);
Perr = -alphae(2)
G = (acker(Abb',Aab',Perr))' % Substitute Abb-->A and Aab-->C
```

### EXAMPLE 9.8

We will now calculate the transfer function of the controller-estimator for the system of Example 9.7. From Example 9.7,  $G = 0.9048$  and

$$\begin{array}{llll} A_{aa} = 1 & A_{ab} = 0.095163 & B_a = 0.0048374 & K_1 = 4.212 \\ A_{ba} = 0 & A_{bb} = 0.904837 & B_b = 0.095163 & K_b = 1.112 \end{array}$$

and from (9-66),

$$\begin{aligned} D_{ce}(z) &= \frac{-U(z)}{Y(z)} \\ &= K_1 + K_b[z + C_1]^{-1}[Gz + C_2] \\ &= K_1 + \frac{K_b(Gz + C_2)}{z + C_1} \\ &= \frac{K_1(z + C_1) + K_b(Gz + C_2)}{z + C_1} \end{aligned}$$

$$\begin{aligned} C_1 &= -A_{bb} + GA_{ab} + (B_b - GB_a)K_b, \\ C_2 &= A_{ba} - GA_{aa} - K_1(B_b - GB_a) \end{aligned}$$



$$\begin{aligned}
 &= \frac{(K_1 + K_b G)z + (K_1 C_1 + K_b C_2)}{z + C_1} \\
 &= (K_1 + K_b G) \frac{z + \frac{K_1 C_1 + K_b C_2}{K_1 + K_b G}}{z + C_1} \\
 &= \frac{C_3(z + C_4)}{z + C_1} \quad \begin{aligned} C_3 &= K_1 + K_b G, \\ C_4 &= (K_1 C_1 + K_b C_2)/C_3 \end{aligned} \\
 &= \frac{5.2181(z - 0.8537)}{z - 0.7178}
 \end{aligned}$$

Since the zero is closer to  $z = 1$  than is the pole, the control-estimator is phase lead. This must be true independent of the design procedure, since the system is designed to increase the speed of response, as specified in Example 9.1. MATLAB code for this example is given below.

```

% Using (9-66)
syms z
Aaa=1; Aab=0.095163; Aba=0; Abb=0.904837;
Ba=0.0048374; Bb=0.095163;
Kl=4.212; Kb=1.112; G=0.9048;
Dce=Kl+Kb*inv(z*eye(1)-Abb+G*Aab+(Bb-G*Ba)*Kb)*(G*z+(Aba-G*Aaa-Kl*(Bb-G*Ba)));
[num,den]=nice(Dce,z);
Dce=tf(num,den,-1); Dce=zpk(Dce)
    
```

## 9.5 CURRENT OBSERVERS

The estimator developed in Section 9.3 is a prediction observer, since the estimate of  $\mathbf{x}(k)$  is based on the measurement  $\mathbf{y}(k - 1)$ . However, the reduced-order observer of Section 9.4 estimates states at time  $kT$  using the measurement at time  $kT$ . We call this type of estimator a *current estimator*, or current observer. We will now consider a full-order current estimator.

As before, the system model is given by

$$\begin{aligned}
 \text{[eq. (9-26)]} \quad \mathbf{x}(k + 1) &= \mathbf{Ax}(k) + \mathbf{Bu}(k) \\
 \mathbf{y}(k) &= \mathbf{Cx}(k)
 \end{aligned}$$

We wish to estimate the state vector  $\mathbf{x}(k)$  with the vector  $\mathbf{q}(k)$ . One form of a full-order current observer is given by the two equations

$$\begin{aligned}
 \bar{\mathbf{q}}(k + 1) &= \mathbf{Aq}(k) + \mathbf{Bu}(k) \\
 \mathbf{q}(k + 1) &= \bar{\mathbf{q}}(k + 1) + \mathbf{G}[\mathbf{y}(k + 1) - \mathbf{C}\bar{\mathbf{q}}(k + 1)] \quad (9-67)
 \end{aligned}$$

In these equations,  $\bar{\mathbf{q}}(k + 1)$  is a first estimate of the state at time  $(k + 1)T$ , based on the dynamics of the system and on the signals at time  $kT$ . This estimate is then corrected in the second equation when the measurement at time  $(k + 1)T$  arrives. The estimator gain is  $\mathbf{G}$ , which determines

the weight placed on the difference between measurement at  $(k + 1)T$  and what we expect that measurement to be at that time. The final estimate is  $\mathbf{q}(k + 1)$ .

In (9-67), the first estimate  $\bar{\mathbf{q}}(k + 1)$  can be eliminated by substituting the first equation into the second one. The single equation for the estimate is then

$$\mathbf{q}(k + 1) = [\mathbf{A} - \mathbf{GCA}]\mathbf{q}(k) + [\mathbf{B} - \mathbf{GCB}]\mathbf{u}(k) + \mathbf{Gy}(k + 1) \quad (9-68)$$

We see then that the estimate at time  $(k + 1)T$  is based on the measurement at  $(k + 1)T$ . The dynamics of this estimator are described by the characteristic equation

$$|z\mathbf{I} - \mathbf{A} + \mathbf{GCA}| = 0 \quad (9-69)$$

It can be shown that this estimator has the same property as the prediction estimator; the transfer matrix  $\mathbf{Q}(z)/\mathbf{U}(z)$  is equal to the transfer matrix  $\mathbf{X}(z)/\mathbf{U}(z)$ . The proof of this property is straightforward, and is given as Problem 9.5-6.

As in the prediction-observer case, we define the estimation-error vector  $\mathbf{e}(k)$  by the relationship

$$\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{q}(k) \quad (9-70)$$

Then

$$\begin{aligned} \mathbf{e}(k + 1) &= \mathbf{x}(k + 1) - \mathbf{q}(k + 1) \\ &= \mathbf{Ax}(k) + \mathbf{Bu}(k) - [\mathbf{A} - \mathbf{GCA}]\mathbf{q}(k) \\ &\quad - [\mathbf{B} - \mathbf{GCB}]\mathbf{u}(k) - \mathbf{GC}[\mathbf{Ax}(k) + \mathbf{Bu}(k)] \\ &= [\mathbf{A} - \mathbf{GCA}][\mathbf{x}(k) - \mathbf{q}(k)] = [\mathbf{A} - \mathbf{GCA}]\mathbf{e}(k) \end{aligned} \quad (9-71)$$

from (9-26) and (9-68). Hence the error vector has dynamics with the same characteristic equation as the estimator, (9-69).

The characteristic equation for the prediction observer is

$$[\text{eq. (9-43)}] \quad |z\mathbf{I} - \mathbf{A} + \mathbf{GC}| = 0$$

and for the current observer is

$$[\text{eq. (9-69)}] \quad |z\mathbf{I} - \mathbf{A} + \mathbf{GCA}| = 0$$

Thus, for a single-input single-output system, Ackermann's formula for the current observer is obtained from that for the prediction observer, (9-48), by replacing  $\mathbf{C}$  with  $\mathbf{CA}$ ; hence, for the current observer,

$$\mathbf{G} = \alpha_e(\mathbf{A}) \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^n \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (9-72)$$

A pole-placement design implemented with a current observer can be viewed as a digital-controller design as shown in Fig. 9-8, with the controller-estimator transfer function given by

$$D_{ce}(z) = z\mathbf{K}[z\mathbf{I} - \mathbf{A} + \mathbf{GCA} + \mathbf{BK} - \mathbf{GCBK}]^{-1}\mathbf{G} \quad (9-73)$$

(see Problem 9.5-6). The current-observer design will now be illustrated with an example.

### EXAMPLE 9.9

We will again consider the system of the earlier examples:

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.0048374 \\ 0.095163 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k)$$

with the gain matrix

$$\mathbf{K} = [4.212 \quad 1.112]$$

and the closed-loop system characteristic equation is given by

$$\alpha_c(z) = z^2 - 1.778644z + 0.818731$$

We will use the same observer characteristic equation as used for the prediction observer of Example 9.3.

$$\alpha_e(z) = (z - 0.818731)^2 = z^2 - 1.637462z + 0.670320 = 0$$

From Example 9.3,

$$\alpha_e(\mathbf{A}) = \begin{bmatrix} 0.032858 & 0.025444 \\ 0 & 0.007414 \end{bmatrix}$$

Also,

$$\mathbf{CA} = [1 \quad 0.095163]$$

and thus

$$\mathbf{CA}^2 = (\mathbf{CA})\mathbf{A} = [1 \quad 0.095163] \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix} = [1 \quad 0.18127]$$

Then

$$\begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0.095163 \\ 1 & 0.18127 \end{bmatrix}^{-1} = \begin{bmatrix} 2.105171 & -1.105171 \\ -11.61346 & 11.61346 \end{bmatrix}$$

From (9-72),

$$\mathbf{G} = \alpha_e(\mathbf{A}) \begin{bmatrix} \mathbf{CA} \\ \mathbf{CA}^2 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.032858 & 0.025444 \\ 0 & 0.007414 \end{bmatrix} \begin{bmatrix} 2.105171 & -1.105171 \\ -11.61346 & 11.61346 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.032858 & 0.0254442 \\ 0 & 0.0074138 \end{bmatrix} \begin{bmatrix} -1.105171 \\ 11.613456 \end{bmatrix} = \begin{bmatrix} 0.259181 \\ 0.086106 \end{bmatrix}$$

The difference equation of the observer in (9-68) will now be evaluated.

$$\mathbf{GCA} = \begin{bmatrix} 0.259434 \\ 0.086102 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix}$$

$$= \begin{bmatrix} 0.259434 \\ 0.086102 \end{bmatrix} \begin{bmatrix} 1 & 0.095163 \end{bmatrix}$$

$$= \begin{bmatrix} 0.259434 & 0.024689 \\ 0.086102 & 0.008194 \end{bmatrix}$$

and

$$\mathbf{A} - \mathbf{GCA} = \begin{bmatrix} 0.740566 & 0.070474 \\ -0.086102 & 0.896643 \end{bmatrix}$$

Also, evaluation of  $(\mathbf{B} - \mathbf{GCB})$  yields

$$\mathbf{B} - \mathbf{GCB} = \begin{bmatrix} 0.003582 \\ 0.094746 \end{bmatrix}$$

We see then from (9-68) that the observer is implemented as

$$\mathbf{q}(k + 1) = \begin{bmatrix} 0.740566 & 0.070474 \\ -0.086102 & 0.896643 \end{bmatrix} \mathbf{q}(k) + \begin{bmatrix} 0.003582 \\ 0.094746 \end{bmatrix} u(k) + \begin{bmatrix} 0.259181 \\ 0.086106 \end{bmatrix} y(k + 1)$$

The gain vector  $\mathbf{G}$  is calculated by the MATLAB program

```
A=[1 0.095163; 0 0.904837];
alphae=[1 -1.637462 0.670320];B=[0.0048374; 0.0951630];C=[1 0];
Perr = roots(alphae); % Perr specifies the desired roots
CA = C*A;
G=(acker(A',CA',Perr))' % Substitute C*A-->A
```

### EXAMPLE 9.10

This example is a continuation of the last example. The transfer function of the controller estimator will be calculated from (9-73). From Example 9.9,

$$\mathbf{A} - \mathbf{GCA} = \begin{bmatrix} 0.740566 & 0.070474 \\ -0.086102 & 0.896643 \end{bmatrix} \quad \mathbf{B} - \mathbf{GCB} = \begin{bmatrix} 0.003582 \\ 0.094746 \end{bmatrix}$$

Hence

$$(\mathbf{B} - \mathbf{GCB})\mathbf{K} = \begin{bmatrix} 0.0035836 \\ 0.0947465 \end{bmatrix} [4.212 \quad 1.112] = \begin{bmatrix} 0.015096 & 0.0039848 \\ 0.399110 & 0.1053532 \end{bmatrix}$$

and

$$\mathbf{A} - \mathbf{GCA} - \mathbf{BK} + \mathbf{GCBK} = \begin{bmatrix} 0.725723 & 0.066514 \\ -0.485210 & 0.791290 \end{bmatrix}$$

$$\begin{aligned} [z\mathbf{I} - \mathbf{A} + \mathbf{GCA} + \mathbf{BK} - \mathbf{GCBK}]^{-1} &= \begin{bmatrix} z - 0.725723 & -0.066514 \\ 0.485210 & z - 0.791290 \end{bmatrix}^{-1} \\ &= \frac{1}{\Delta} \begin{bmatrix} z - 0.791290 & 0.066514 \\ -0.485210 & z - 0.725723 \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} \Delta &= |z\mathbf{I} - \mathbf{A} + \mathbf{GCA} + \mathbf{BK} - \mathbf{GCBK}| \\ &= z^2 - 1.517z + 0.6065 \end{aligned}$$

Then, from (9-73),

$$\begin{aligned} D_{ce}(z) &= z\mathbf{K}[z\mathbf{I} - \mathbf{A} + \mathbf{GCA} + \mathbf{BK} - \mathbf{GCBK}]^{-1}\mathbf{G} \\ &= \frac{z}{\Delta} [4.212 \quad 1.112] \begin{bmatrix} z - 0.791290 & 0.066514 \\ -0.485210 & z - 0.725723 \end{bmatrix} \begin{bmatrix} 0.259181 \\ 0.086100 \end{bmatrix} \\ &= \frac{z}{\Delta} [4.212 \quad 1.112] \begin{bmatrix} 0.259181z - 0.199360 \\ 0.086100z - 0.188242 \end{bmatrix} \\ &= \frac{1.188z - 1.049}{z^2 - 1.517z + 0.6065} = \frac{1.188z(z - 0.8834)}{z^2 - 1.517z + 0.6065} \end{aligned}$$

Here is a MATLAB program to compute the transfer function using (9-73). The code generates a transfer function in  $z$  using the *nice* function employed in Example 9.4.

```
%Compute the digital filter using Equation (9-73)
T=0.1; syms z
A=[1 0.095163;0 0.904837];B=[0.0048374;0.0951630];C=[1 0];
alphae=[1 -1.637462 0.670320]; Perr=roots(alphae); CA=C*A;
K=[4.212 1.112]; G=(acker(A',CA',Perr))';
Dce=z*K*inv(z*eye(2)-A+G*C*A+B*K-G*C*B*K)*G % Eqn (9-73)
[num,den]= nice(Dce,z);
Dce = tf(num,den,T)
Dce = zpk(Dce)
```

A problem can occur in the implementation of observer-based controllers. Thus far we have not considered the relative stability of these control systems. A system that has adequate stability margins is said to be *robust*; observer-based control systems may not be robust in terms of the gain and phase margins that appear at the input to the plant. To determine these stability margins, we see from Fig. 9-8 that the frequency response for the open-loop function  $D_{ce}(z)G(z)$  must be calculated.

This was done for the prediction-observer system of Example 9.4 and the current-observer system of Example 9.9. In both systems the phase margin is  $49^\circ$  and the gain margins are greater than 8 dB. In these systems no relative-stability problems appear. This problem is discussed further, when Kalman filters (which are optimal current observers) are discussed in Chapter 11.

As in the case of the prediction observer in (9-56), the state model of the closed-loop system employing the current observer can be derived as

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{q}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{GCA} & \mathbf{A} - \mathbf{GCA} - \mathbf{BK} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{q}(k) \end{bmatrix} \quad (9-74)$$

The derivation of this equation is straightforward and is given as an exercise in Problem 9.5-7.

## 9.6 CONTROLLABILITY AND OBSERVABILITY

In the preceding sections Ackermann's formula was useful in both pole-assignment design, (9-25), and observer design, (9-48). In pole-assignment design it was necessary that the inverse of the matrix

$$[\mathbf{B} \ \mathbf{AB} \ \mathbf{A}^2\mathbf{B} \ \dots \ \mathbf{A}^{n-1}\mathbf{B}] \quad (9-75)$$

exist, and in prediction-observer design it was necessary that the inverse of the matrix

$$\begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \quad (9-76)$$

exist. We will now relate the existence of the inverses of (9-75) and (9-76) to the important concepts of controllability and observability, respectively.

We introduce the concept of *controllability* with respect to the system of Fig. 9-9.

The system characteristic equation is given by

$$(z - 0.9)(z - 0.8) = 0$$

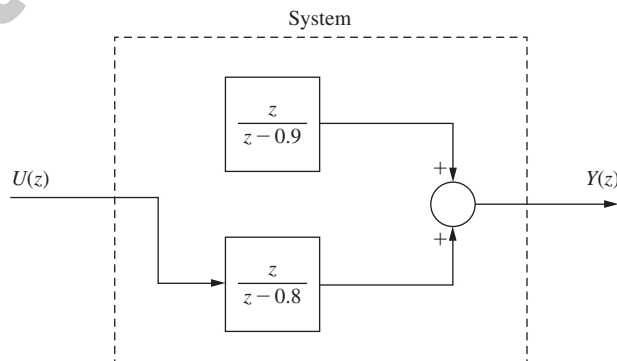


FIGURE 9-9 Uncontrollable system.

However, the mode of the transient response  $(0.9)^k$  is not excited by the input  $u(k)$  and hence cannot be controlled by  $u(k)$ . This system is said to be *uncontrollable*.

**DEFINITION 1.** A system

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \quad (9-77)$$

is controllable provided that there exists a sequence of input  $\mathbf{u}(0), \mathbf{u}(1), \mathbf{u}(2), \dots, \mathbf{u}(N)$  that will translate the system from any initial state  $\mathbf{x}(0)$  to any final state  $\mathbf{x}(N)$ , with  $N$  finite.

Note that in the system of Fig. 9-9, the input  $u(k)$  has no influence on the state of the upper block; thus the system is uncontrollable.

We will now derive the conditions for the system of (9-77) to be controllable. Now

$$\begin{aligned}
 \mathbf{x}(1) &= \mathbf{A}\mathbf{x}(0) + \mathbf{B}u(0) \\
 \mathbf{x}(2) &= \mathbf{A}\mathbf{x}(1) + \mathbf{B}u(1) \\
 &= \mathbf{A}^2\mathbf{x}(0) + \mathbf{A}\mathbf{B}u(0) + \mathbf{B}u(1) \\
 &\vdots \\
 &\vdots
 \end{aligned} \quad (9-78)$$

$$\mathbf{x}(N) = \mathbf{A}^N\mathbf{x}(0) + \mathbf{A}^{N-1}\mathbf{B}u(0) + \dots + \mathbf{A}\mathbf{B}u(N-2) + \mathbf{B}u(N-1)$$

$$\begin{aligned}
 &= \mathbf{A}^N\mathbf{x}(0) + [\mathbf{B} \ \mathbf{A}\mathbf{B} \ \dots \ \mathbf{A}^{N-1}\mathbf{B}] \begin{bmatrix} u(N-1) \\ u(N-2) \\ \vdots \\ u(0) \end{bmatrix} \\
 &\quad (9-79)
 \end{aligned}$$

Hence, with  $\mathbf{x}(N)$  and  $\mathbf{x}(0)$  known, this equation can be written as

$$[\mathbf{B} \ \mathbf{A}\mathbf{B} \ \dots \ \mathbf{A}^{N-1}\mathbf{B}] \begin{bmatrix} u(N-1) \\ u(N-2) \\ \vdots \\ u(0) \end{bmatrix} = \mathbf{x}(N) - \mathbf{A}^N\mathbf{x}(0) \quad (9-80)$$

Since the order of the state vector  $\mathbf{x}(k)$  is  $n$ , then (9-80) yields  $n$  linear simultaneous equations. Hence, for a solution to exist, the rank of the coefficient matrix [which is (9-75) for  $N$  equal to  $n$ ]

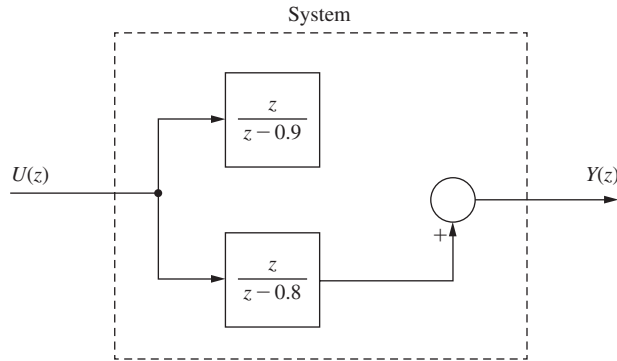
$$[\mathbf{B} \ \mathbf{A}\mathbf{B} \ \dots \ \mathbf{A}^{N-1}\mathbf{B}] \quad (9-81)$$

must be  $n$  [7]. For the case of a single input,  $\mathbf{B}$  is a column matrix and (9-81) is  $n \times n$  for  $N$  equal to  $n$ . Then the inverse of (9-81) must exist; this is also the condition for the existence of the solution to Ackermann's formula, in (9-25).

We consider next the concept of *observability*.

**DEFINITION 2.** A system

$$\begin{aligned}
 \mathbf{x}(k + 1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \\
 \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)
 \end{aligned} \quad (9-82)$$



**FIGURE 9-10** Unobservable system.

is observable provided that the initial state  $\mathbf{x}(0)$ , for any  $\mathbf{x}(0)$ , can be calculated from the  $N$  measurements  $\mathbf{y}(0), \mathbf{y}(1), \dots, \mathbf{y}(N - 1)$ , with  $N$  finite.

A system that is obviously unobservable is shown in Fig. 9-10, since the state of the upper block does not contribute to the output  $y(k)$ .

The derivation of the criteria for observability closely parallels that for controllability. To simplify the derivation, we will assume that  $u(k)$  is zero; the derivation for  $u(k)$  not equal to zero is similar, and is given as Problem 9.6-1. Now, from (9-82),

$$\begin{aligned} \mathbf{y}(0) &= \mathbf{C}\mathbf{x}(0) \\ \mathbf{y}(1) &= \mathbf{C}\mathbf{x}(1) = \mathbf{C}\mathbf{A}\mathbf{x}(0) \\ &\vdots \\ \mathbf{y}(N - 1) &= \mathbf{C}\mathbf{x}(N - 1) = \mathbf{C}\mathbf{A}^{N-1}\mathbf{x}(0) \end{aligned} \tag{9-83}$$

or

$$\begin{bmatrix} \mathbf{y}(0) \\ \mathbf{y}(1) \\ \vdots \\ \mathbf{y}(N - 1) \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{N-1} \end{bmatrix} \mathbf{x}(0) \tag{9-84}$$

Hence, by the same arguments as for controllability, the rank of the coefficient matrix (which is (9-76) for  $N$  equal to  $n$ )

$$\begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{N-1} \end{bmatrix} \tag{9-85}$$

must be  $n$ . For the case of a single output,  $\mathbf{C}$  is a row matrix and (9-85) is  $n \times n$  for  $N$  equal to  $n$ . Then the inverse of (9-85) must exist; this is the condition for the existence of the solution to Ackermann's formula, (9-48).



**EXAMPLE 9.11**

We will test the system of Fig. 9-11 for both controllability and observability. This system is composed of two cascaded first-order subsystems, each of which is controllable and observable, which is seen by inspection. The system state equations are given by

$$\mathbf{x}(k + 1) = \begin{bmatrix} -0.2 & 0 \\ -1 & 0.8 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k)$$

$$y(k) = [-1 \quad 1] \mathbf{x}(k) + \mathbf{u}(k)$$

First we will test for controllability, using (9-81).

$$\mathbf{AB} = \begin{bmatrix} -0.2 & 0 \\ -1 & 0.8 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.2 \\ -0.2 \end{bmatrix}$$

and thus

$$[\mathbf{B} \quad \mathbf{AB}] = \begin{bmatrix} 1 & -0.2 \\ 1 & -0.2 \end{bmatrix}$$

This matrix is of rank 1 and its inverse does not exist; hence the system is uncontrollable.

Next we will test for observability, using (9-85).

$$\mathbf{CA} = [-1 \quad 1] \begin{bmatrix} -0.2 & 0 \\ -1 & 0.8 \end{bmatrix} = [-0.8 \quad 0.8]$$

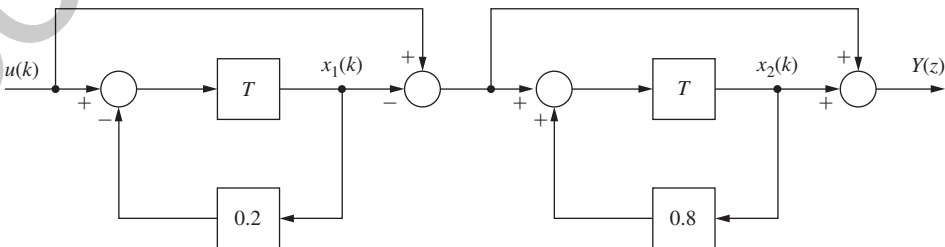
and thus

$$\begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ -0.8 & 0.8 \end{bmatrix}$$

This matrix is also of rank 1; hence the system is unobservable. It is instructive to calculate some transfer functions. By Mason's gain formula and from Fig. 9-11,

$$\frac{Y(z)}{U(z)} = \frac{1(1 - 0.6z^{-1} - 0.16z^{-2}) - z^{-1}(1 - 0.8z^{-1}) + z^{-1}(1 + 0.2z^{-1}) - z^{-2}}{1 + 0.2z^{-1} - 0.8z^{-1} - 0.16z^{-2}}$$

$$= 1$$



**FIGURE 9-11** System for Example 9.11.

$$\frac{X_2(z)}{U(z)} = \frac{z^{-1}(1 + 0.2z^{-1}) - z^{-2}}{1 - 0.6z^{-1} - 1.6z^{-2}} = \frac{z^{-1}}{1 + 0.2z^{-1}}$$

$$\frac{X_1(z)}{U(z)} = \frac{z^{-1}}{1 + 0.2z^{-1}}$$

$$\frac{Y(z)}{X_1(z)} = \frac{Y(z) U(z)}{U(z) X_1(z)} = \frac{1 + 0.2z^{-1}}{z^{-1}}$$

From the transfer function  $X_2(z)/U(z)$ , we see that the mode  $(0.8)^k$  is not excited by the input. From the transfer function  $Y(z)/X_1(z)$ , we see that the mode  $(-0.2)^k$  does not appear in the output.

Here is a MATLAB program to perform the computations of this example.

```
A=[-0.2 0; -1 0.8]; B=[1; 1]; C=[-1 1]; D=[1];
T=1; %Choose any sampling interval
Sysd=ss(A,B,C,D,T)
S_control=[B A*B]
Rank_S_control=rank(S_control)
S_observe=[C; C*A]
rank_S_observe=rank(S_observe)
Yz_over_Xz = tf(Sysd)
```

## 9.7 SYSTEMS WITH INPUTS

In the pole-assignment design procedures presented in this chapter, the design resulted in a system in which the initial conditions were driven to zero in some prescribed manner; however, the system had no input. This type of system is called a *regulator control system*. However, in many control systems, it is necessary that the system output  $y(t)$  follow, or track, a system input  $r(t)$ . An example of this type of system is the vertical (or altitude) control system of an aircraft automatic landing system. The aircraft follows a prescribed glide slope, which is a ramp function. Hence the altitude control system input is a ramp function. In this section some techniques will be presented for modifying the pole-assignment design method for systems that must track an input.

We first consider the case of low-order single-input single-output systems in which all states are measured; thus no observer is required. The system equations are given by

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \quad (9-86)$$

where, for a regulator control system,

$$u(k) = -\mathbf{K}\mathbf{x}(k) \quad (9-87)$$

The control input  $u(k)$  is the only variable that we can manipulate; thus this input must then be a function of the system input  $r(k)$ . Thus we modify  $u(k)$  to be a linear function of  $r(k)$ ,

$$u(k) = -\mathbf{K}\mathbf{x}(k) + Nr(k) \quad (9-88)$$

where  $N$  is a constant to be determined. Then, from (9-86) and (9-88),

$$\begin{aligned} \mathbf{x}(k + 1) &= (\mathbf{A} - \mathbf{BK})\mathbf{x}(k) + \mathbf{BN}r(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \quad (9-89)$$

Taking the  $z$ -transform of these equations and solving for the transfer function yields

$$\frac{Y(z)}{R(z)} = \mathbf{C}(z\mathbf{I} - \mathbf{A} + \mathbf{BK})^{-1}\mathbf{BN} \quad (9-90)$$

Note that the choice of  $N$  does not affect the system's relative frequency response, but only the amplitude of the frequency response; the transfer function is independent of  $N$  except as a multiplying factor.

A special case is worth noting. For the case that the output is a state, which we can call  $x_1(t)$  without loss of generality, we can write the output equation as

$$y(k) = \mathbf{C}\mathbf{x}(k) = [1 \ 0 \ \cdots \ 0]\mathbf{x}(k) = x_1(k)$$

The equation for  $u(k)$  can now be expressed as a function of the system error, which is  $e(k) = r(k) - y(k)$ . From (9-88), we let  $N = K_1$ , and

$$\begin{aligned} u(k) &= -\mathbf{K}\mathbf{x}(k) + K_1r(k) \\ &= K_1[r(k) - y(k)] - K_2x_2(k) - \cdots - K_nx_n(k) \end{aligned} \quad (9-91)$$

Thus the control is a function of the system error, as was the case for classical control design. The system block diagram for this case is given in Fig. 9-12. This system is investigated further in Example 9.12.

It is informative to determine the zeros of the system transfer function, (9-90). To locate the zeros, consider the  $z$ -transform of (9-89).

$$\begin{aligned} (z\mathbf{I} - \mathbf{A} + \mathbf{BK})\mathbf{X}(z) - \mathbf{BN}R(z) &= 0 \\ \mathbf{C}\mathbf{X}(z) &= Y(z) \end{aligned} \quad (9-92)$$

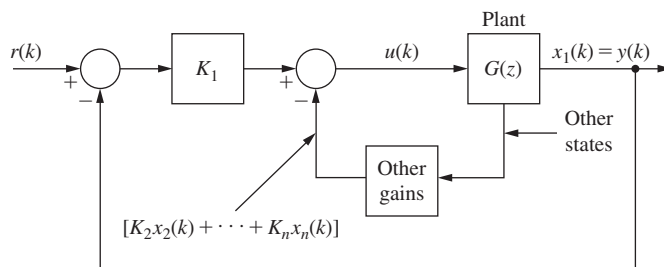


FIGURE 9-12 System with input.

If  $z = z_0$  is a zero of the transfer function, (9-90), then  $Y(z_0)$  is zero with  $R(z_0)$  and  $\mathbf{X}(z_0)$  non-zero. Hence we can express (9-92) as

$$\begin{bmatrix} z_0\mathbf{I} - \mathbf{A} + \mathbf{BK} & -\mathbf{BN} \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X}(z_0) \\ R(z_0) \end{bmatrix} = 0 \quad (9-93)$$

This equation has a nontrivial solution, since  $R(z_0)$  and  $\mathbf{X}(z_0)$  are not zero. Hence the determinant of the coefficient matrix, which is a polynomial in  $z_0$ , must be zero [7]. The roots of this polynomial are then the zeros of the system transfer function.

The locations of the zeros of the transfer function are not evident from (9-93). However, we can rewrite (9-93) as

$$\begin{bmatrix} z_0\mathbf{I} - \mathbf{A} & -\mathbf{B} \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X}(z_0) \\ NR(z_0) - \mathbf{KX}(z_0) \end{bmatrix} = 0 \quad (9-94)$$

Here the coefficient matrix is independent of  $\mathbf{K}$  and  $N$ , and is, in fact, that of the plant alone. Hence the zeros of the closed-loop transfer function are the same as those of the transfer function of the plant and cannot be changed by the design procedure. This design procedure is then of limited value. An example illustrating this design procedure will now be given.

### EXAMPLE 9.12

The system of Example 9.1 will again be considered, but with an input added as in (9-88). For this system the plant state equations are given by

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} 1 & 0.095163 \\ 0 & 0.904837 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.0048374 \\ 0.0951630 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k) \end{aligned}$$

The design in Example 9.1 resulted in the gain matrix

$$\mathbf{K} = [4.212 \quad 1.112]$$

Hence

$$\mathbf{BK} = \begin{bmatrix} 0.0048374 \\ 0.0951630 \end{bmatrix} [4.212 \quad 1.112] = \begin{bmatrix} 0.0204 & 0.0054 \\ 0.4008 & 0.1058 \end{bmatrix}$$

Then

$$z\mathbf{I} - \mathbf{A} + \mathbf{BK} = \begin{bmatrix} z - 0.9796 & -0.0898 \\ 0.4008 & z - 0.7990 \end{bmatrix}$$

and

$$(z\mathbf{I} - \mathbf{A} + \mathbf{BK})^{-1} = \frac{1}{z^2 - 1.7786z + 0.8187} \begin{bmatrix} z - 0.7990 & 0.0898 \\ -0.4008 & z - 0.9796 \end{bmatrix}$$

The system transfer function is given by (9-90):

$$\frac{Y(z)}{R(z)} = \mathbf{C}(z\mathbf{I} - \mathbf{A} + \mathbf{BK})^{-1}\mathbf{BN} = \frac{0.004837(z + 0.9672)N}{z^2 - 1.7786z + 0.8187}$$

Note that the system dc gain is given by

$$\left. \frac{Y(z)}{R(z)} \right|_{z=1} = 0.2373N$$

The dc gain can be increased by the choice of  $N$ . Since  $y(k) = x_1(k)$ , we can choose  $N = K_1$  as in (9-91), and the control will be a function of the error between the input and the output. Since  $K_1 = 4.212$ , the dc gain is now unity; the system shown in the inner loop of Fig. 9-12 is type 1. The step response for this case is shown in Fig. 9-13, and the steady-state error is zero. MATLAB code for this example follows.

```
% Append this code to that of Example 9.3
syms z
N=1; Tz=C*inv(z*eye(2)-A+B*K)*B*N; % Equation (9-90)
[num,den]=nice(Tz,z);
Tz=tf(num,den,-1)
```

We will now consider the case that a prediction observer is used to implement the control system. The plant is described by

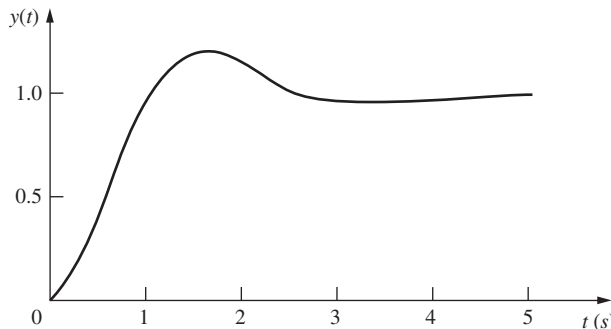
$$\begin{aligned} \mathbf{x}(k + 1) &= \mathbf{Ax}(k) + \mathbf{Bu}(k) \\ y(k) &= \mathbf{Cx}(k) \end{aligned} \tag{9-95}$$

and the observer, from (9-38), is described by

[eq. (9-38)]  $\mathbf{q}(k + 1) = (\mathbf{A} - \mathbf{GC})\mathbf{q}(k) + \mathbf{G}y(k) + \mathbf{Bu}(k)$

The control is implemented by

$$u(k) = -\mathbf{Kq}(k) \tag{9-96}$$



**FIGURE 9-13** Step response for Example 9.12.

To be general, we can allow the input to enter both (9-38) and (9-96). These equations become

$$\mathbf{q}(k + 1) = (\mathbf{A} - \mathbf{GC})\mathbf{q}(k) + \mathbf{G}y(k) + \mathbf{B}u(k) + \mathbf{M}r(k) \quad (9-97)$$

$$u(k) = -\mathbf{K}\mathbf{q}(k) + Nr(k) \quad (9-98)$$

The design problem then is to choose  $\mathbf{M}$  and  $N$  so as to achieve certain design objectives, such as zeros of the system transfer function, steady-state errors, and so on.

We will consider the case that only the error

$$e(t) = r(t) - y(t) \quad (9-99)$$

is measured, and thus only this signal is available for control. For example, in radar tracking systems, the radar return indicates only the error  $e(t)$  between the antenna pointing direction and the direction to the target. The error-control condition can be satisfied in (9-97) and (9-98) by choosing  $N$  to be zero and  $\mathbf{M}$  equal to  $-\mathbf{G}$ . Then (9-97) and (9-98) become, respectively,

$$\mathbf{q}(k + 1) = (\mathbf{A} - \mathbf{GC})\mathbf{q}(k) + \mathbf{G}[y(k) - r(k)] + \mathbf{B}u(k) \quad (9-100)$$

$$u(k) = -\mathbf{K}\mathbf{q}(k) \quad (9-101)$$

Note that this design procedure also gives no choice in the selection of the system transfer-function zeros. An example will now be given to illustrate this design technique.

### EXAMPLE 9.13

The system of Example 9.12 will be considered, with the observer designed in Example 9.3 to be employed. The plant state equations are given in Example 9.12. The gain matrices for the controller and for the observer are given in Example 9.3. The required matrices are then

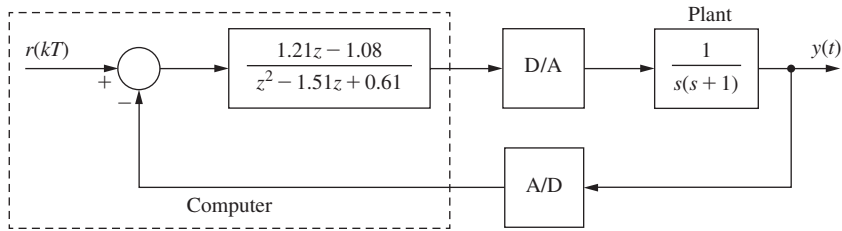
$$\mathbf{A} = \begin{bmatrix} 1 & 0.09516 \\ 0 & 0.90484 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.00484 \\ 0.09516 \end{bmatrix}, \quad \mathbf{C} = [1 \quad 0]$$

$$\mathbf{K} = [4.2124 \quad 1.1119], \quad \mathbf{G} = \begin{bmatrix} 0.2674 \\ 0.0779 \end{bmatrix}$$

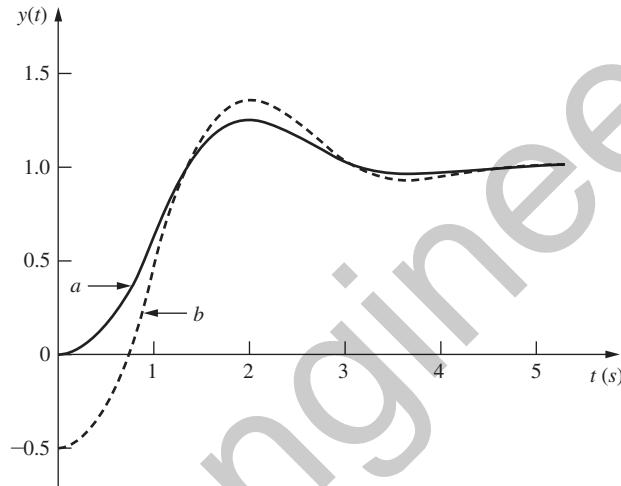
Substitution of these matrices into (9-100) and (9-101), and then substitution of (9-101) into (9-100), yields the observer equations

$$\mathbf{q}(k + 1) = \begin{bmatrix} 0.7122 & 0.0898 \\ -0.4788 & 0.7990 \end{bmatrix} \mathbf{q}(k) + \begin{bmatrix} 0.2674 \\ 0.0779 \end{bmatrix} [y(k) - r(k)]$$

Note that no design is required beyond that performed in Example 9.3. The system block diagram is given in Fig. 9-14. In this block diagram, the controller transfer function is as calculated in Example 9.4, and it is assumed that the input  $r(kT)$  is calculated in the computer (as is usually the case). Step responses for this design are given in Fig. 9-15. For the response shown by the



**FIGURE 9-14** System for Example 9.13.



**FIGURE 9-15** Responses for the system of Example 9.13.

solid line, the initial conditions of both the plant and the observer are all zero. For the response shown by the dashed line, the initial conditions are given by

$$\mathbf{x}(0) = \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}, \quad \mathbf{q}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

For this design procedure, the transient response appears satisfactory; in addition, the steady-state error is zero.

## 9.8 SUMMARY

In this chapter a design technique has been presented which is based on the state model of the system, compared to the design methods of Chapter 8, which are based on transfer functions. The design technique presented is the pole-assignment, or pole-placement, technique. This technique allows us to place all the zeros of the system characteristic equation at desirable locations; however, the technique requires that all the system states be known, a condition that at least is generally impractical. To circumvent this condition, an observer is constructed. An observer estimates the system states, based on the known system inputs and the outputs that can be measured. Then, for the pole-placement design, the estimated states, rather than the actual states, are used to calculate feedback signals.

The design above results in a regulator control system, that is, a control system with no reference input. A method was presented which modifies the pole-placement design for systems that have reference inputs. For the case that the system error is to be used in the control of the system, the modifications required are simple and straightforward.

A problem with observer-based design is that the relative-stability margins at the plant input may not be adequate. This problem can be acute, since the main weakness of analytical design is the inaccuracy of the linear plant model used in the design. In general, a physical system is not linear, and the system's characteristics vary with time, temperature, and so on. Hence an adequate margin of safety is needed, and the relative stability is a measure of that safety margin. In the designs presented in this chapter, the frequency response of the open-loop function of the closed-loop system must be calculated, to ensure that the system has good stability margins.

## References and Further Readings

- [1] J. E. Ackermann, "Der Entwurf linearer Regelungssysteme im Zustandsraum," *Regelungstechnik-Process-Datenverarb.*, 7, pp. 297–300, 1972.
- [2] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Half Moon Bay, CA: Ellis-Kagle Press, 2006.
- [3] D. G. Luenberger, "An Introduction to Observers," *IEEE Trans. Autom. Control*, AC-16, pp. 596–602, 1971.
- [4] Part II, *IEEE Trans. Autom. Control*, AC-26, 1981.
- [5] P. S. Maybeck, *Stochastic Models, Estimation and Control*, Vol. 3. New York: Academic Press, Inc., 1982.
- [6] J. O'Reilly, *Observers for Linear Systems*. London: Academic Press, Inc. (London) Ltd., 1983.
- [7] P. M. De Russo, R. J. Roy, C. M. Close, and A. A. Desrochers, *State Variables for Engineers*, 2nd ed. New York: John Wiley & Sons, Inc., 1998.
- [8] J. C. Williams and S. K. Mitter, "Controllability, Observability, Pole Allocation, and State Reconstruction," *IEEE Trans. Autom. Control*, AC-16, pp. 582–602, 1971.
- [9] W. M. Wonham, "On Pole Assignment in Multi-input Controllable Linear Systems," *IEEE Trans. Autom. Control*, AC-12, pp. 660–665, 1967.
- [10] C. T. Leondes and L. M. Novak, "Reduced-Order Observers for Linear Discrete-Time Systems," *IEEE Trans. Autom. Control*, AC-19, pp. 42–46, 1974.
- [11] R. F. Wilson, "An Observer Based Aircraft Automatic Landing System," M.S. thesis, Auburn University, Auburn, AL, 1981.

## Problems

- 9.2-1.** The plant of Example 9.1 has the state equations given (9-1). Find the gain matrix  $\mathbf{K}$  required to realize the closed-loop characteristic equation with zeros which have a damping ratio  $\zeta$  of 0.46 and a time constant  $\tau$  of 0.5 s. Use pole-assignment design.
- 9.2-2.** Consider the plant of Problem 9.2-1, which has the transfer function, from Example 9.5,

$$G(z) = \frac{z - 1}{z} \mathfrak{L} \left[ \frac{1}{s^2(s + 1)} \right] = \frac{0.00484z + 0.00468}{z^2 - 1.905z + 0.905}$$

- (a) Using the pole-assignment design, find the gain matrix  $\mathbf{K}$  required to realize the closed-loop characteristic equation with zeros that have the damping ratio  $\zeta = 0.707$  and the time constant  $\tau = 0.8$  s.
- (b) To verify the design, show that (9-15) yields the desired closed-loop characteristic equation.
- (c) Draw a flow graph of the form of Fig. 4-20 for the plant. Then add the feedback gains of part (a).



- (d) Write the state equations for the flow graph of part (c), to verify the system matrix ( $\mathbf{A} - \mathbf{BK}$ ) found in part (b).
- (e) Find all loop gains in the flow graph of part (c). Then use Mason's formula to verify that the flow graph of part (c) has the characteristic equation of parts (a) and (b).

**9.2-3.** A chamber temperature control system is modeled as shown in Fig. P9.2-3. This system is described in Problem 1.6-1. For this problem, ignore the disturbance input,  $T = 0.6$  s, and let  $D(z) = 1$ . It was shown in Problem 6.2-4 that

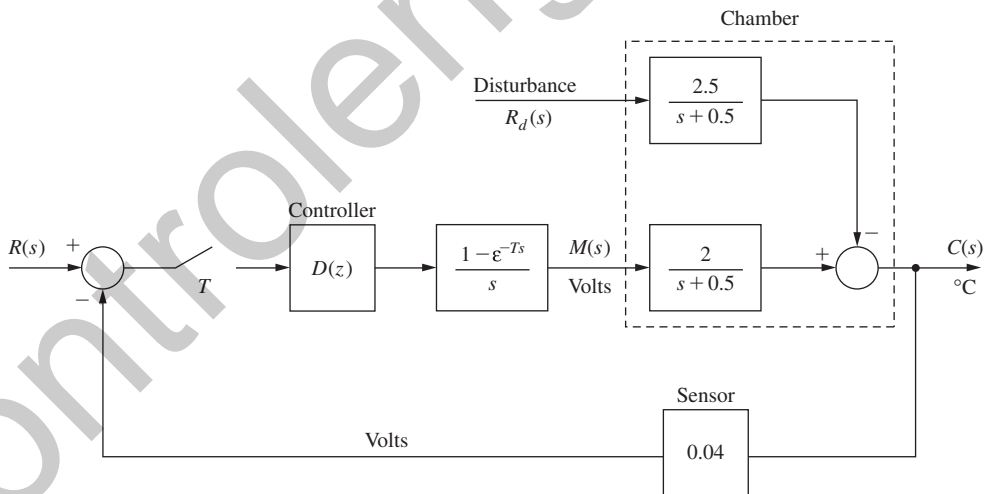
$$(0.04) \frac{z-1}{z} \mathcal{Z} \left[ \frac{2}{s(s+0.5)} \right] = \frac{0.04147}{z-0.7408}$$

Note that the sensor gain is included in this transfer function.

- (a) Draw a flow graph of the plant and sensor. Write the state equations with the state variable  $x(k)$  equal to the system output and the output  $y(k)$  equal to the sensor output.
- (b) Find the time constant  $\tau$  for this closed-loop system.
- (c) Using pole-placement design, find the gain  $K$  that yield the closed-loop time constant  $\tau = 1$  s. Note that the sensor gain does not enter these calculations.
- (d) Show that the gain  $K$  in part (b) yields the desired closed-loop characteristic equation, using (9-15).
- (e) Draw a block diagram for the system that includes the sensor. Let the digital computer realize a gain,  $K_1$ , such that the closed-loop time constant is as given in part (b). The sensor in this system must have the gain given.
- (f) Using the characteristic equation for the block diagram of part (e),

$$1 + K_1 G(z)H = 0$$

verify that this block diagram yields the desired characteristic equation.



**FIGURE P9.2-3** Chamber temperature control system.

**9.2-4.** A satellite control system is modeled as shown in Fig. P9.2-4. This system is described in Problem 1.4-1. For this problem, let  $D(z) = 1$ . In addition,  $K = 1$ ,  $T = 1$  s,  $J = 4$ , and  $H_k = 1$ . From the  $z$ -transform tables,

$$\frac{z-1}{z} \mathcal{Z} \left[ \frac{1}{4s^3} \right] = \frac{0.125(z+1)}{(z-1)^2}$$

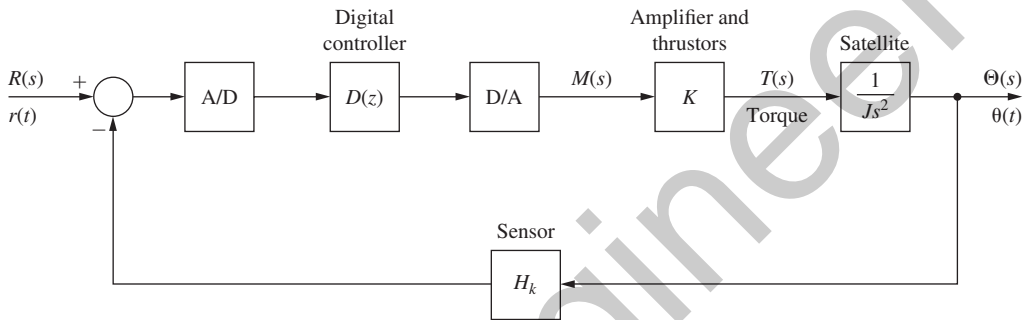
A state model for this system is given by

$$\mathbf{x}(k + 1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.125 \\ 0.25 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k)$$

where  $x_1(k)$  is angular position and  $x_2(k)$  is angular velocity.

- Show that the closed-loop system is unstable.
- Using pole-placement design, find the gain matrix  $\mathbf{K}$  that yields the closed-loop damping ratio  $\zeta = 0.707$  and the time constant  $\tau = 4$  s.
- Show that the gain matrix  $\mathbf{K}$  in part (b) yields the desired closed-loop characteristic equation, using (9-15).



**FIGURE P9.2-4** Block diagram for a satellite control system.

**9.3-1.** Consider the pole-placement design of Problem 9.2-2.

- Design a predictor observer for this system, with the time constant equal to one-half the value of Problem 9.2-2(a) and with the observer critically damped.
- To check the results of part (a), use (9-46) to show that these results yield the desired observer characteristic equation.
- Find the control-observer transfer function  $D_{ce}(z)$  in Fig. 9-8. Use control gain matrix of Problem 9.2-2(a).
- The characteristic equation of the closed-loop system of Fig. 9-8 is given by

$$1 + D_{ce}(z)G(z) = 0$$

Use  $G(z)$  as given and  $D_{ce}(z)$  in part (c) to show that this equation yields the same characteristic equation as  $\alpha_e(z)\alpha_e(z) = 0$

**9.3-2.** Consider the chamber temperature control system of Problem 9.2-3.

- Design a predictor observer for this system, with the time constant equal to one-half the value of Problem 9.2-3(b).
- To check the results of part (a), use (9-46) to show that these results yield the desired observer characteristic equation.
- Find the control-observer transfer function  $D_{ce}(z)$  in Fig. 9-8. Use the value of control gain of Problem 9.2-3(b). Do not include the sensor gain in  $D_{ce}(z)$ .
- Draw a block diagram of the system in part (b), with  $D(z)$  in Fig. P9.2-3 equal to  $D_{ce}(z)$ .
- Verify the system characteristic equation using the block diagram in part (d).
- What would be the effect of designing a reduced-order estimator for this system?

**9.3-3.** Consider the satellite control system of Problem 9.2-4.

- Design a predictor observer for this system, with the time constant equal to one-half the value of Problem 9.2-4(b) and with the observer critically damped.

- (b) To check the results of part (a), use (9-46) to show that these results yield the desired observer characteristic equation.
- (c) Find the control-observer transfer function  $D_{ce}(z)$  in Fig. 9-8. Use the control gain matrix of Problem 9.2-4(b),  $\mathbf{K} = [0.3893 \ 1.769]$ .
- (d) The characteristic equation of the closed-loop system of Fig. 9-8 is given by

$$1 + D_{ce}(z)G(z) = 0$$

Use  $G(z)$  as given and  $D_{ce}(z)$  in part (c) to show that this equation yields the same characteristic equation as  $\alpha_c(z)\alpha_e(z) = 0$

- 9.3-4.** Assume that equation (9-25), Ackermann's formula for pole-assignment design, is the solution of (9-15). Based on this result, show that (9-48), Ackermann's formula for observer design, is the solution of (9-46).
- 9.3-5.** Given in (9-56) is the closed-loop state model for the pole-placement prediction-estimator design. Extend this model to include plant disturbances and sensor noise, as described in (9-44).
- 9.4-1.** Consider the control system of Problem 9.2-2.

- (a) Design a reduced-order observer for this system with the time constant equal to one-half the value of Problem 9.2-2(a).
- (b) To check the results of part (a), use (9-63) to show that these results yield the desired observer characteristic equation.
- (c) Find the control-observer transfer function  $D_{ce}(z)$  in Fig. 9-8. Use the control gain matrix of Problem 9.2-2(a).
- (d) The characteristic equation of the closed-loop system of Fig. 9-8 is given by

$$1 + D_{ce}(z)G(z) = 0$$

Use  $G(z)$  as given and  $D_{ce}(z)$  in part (c) to show that this equation yields the same characteristic equation as  $\alpha_c(z)\alpha_e(z) = 0$

- 9.4-2.** Consider the satellite control system of Problem 9.2-4.
- (a) Design a reduced-order observer for this system, with the time constant equal to one-half the value of Problem 9.2-4(b).
- (b) To check the results of part (a), use (9-63) to show that these results yield the desired observer characteristic equation.
- (c) Find the control-observer transfer function  $D_{ce}(z)$  in Fig. 9-8. Use the control gain matrix of Problem 9.2-4(b),  $\mathbf{K} = [0.3893 \ 1.769]$ .
- (d) The characteristic equation of the closed-loop system of Fig. 9-8 is given by

$$1 + D_{ce}(z)G(z) = 0$$

Use  $G(z)$  as given and  $D_{ce}(z)$  in part (c) to show that this equation yields the same characteristic equation as  $\alpha_c(z)\alpha_e(z) = 0$ .

- 9.4-3.** Consider the reduced-order observer designed in Problem 9.4-2. In this problem, velocity  $[dy/dt]$  is estimated, using position  $[y]$  plus other information. We could simply calculate velocity, using one of the numerical differentiators described in Section 8.8. This calculated velocity would then replace the estimate of velocity in the control system. (This is the approach taken in the PID controller.)
- (a) Consider the plant observer as an open-loop system, as shown in Fig. 9-4. Calculate the transfer function  $Q(z)/Y(z)$  with  $U(z) = 0$ , where  $q(kT)$  is the observer estimate of velocity. *Hint:* Use (9-62) with  $u(k) = 0$ .
- (b) Plot the Bode diagram for the transfer function in part (a) versus  $\omega_w$ .
- (c) One numerical differentiator given in Section 8.8 is

$$D(z) = \frac{z - 1}{Tz}$$

Plot the Bode diagram for this transfer function on the same graph as in part (b).

- (d) Comment on the similarities and differences in the two frequency responses.

- 9.4-4.** Consider that in Fig. 9-8, the observer is reduced order, and that the system is single-input single-output. Show that the transfer function  $D_{ce}(z)$  of the equivalent controller is given by (9-66).
- 9.5-1.** Repeat all parts of Problem 9.3-1 using a current observer.
- 9.5-2.** Repeat all parts of Problem 9.3-2, using a current observer.
- 9.5-3** (a) Find the closed-loop state equations for the system of Problem 9.3-2(a), of the form of (9-56).  
 (b) Find the system characteristic equation using the results in part (a), and show that this is the desired equation.  
 (c) Repeat parts (a) and (b) for the closed-loop system of Problem 9.5-2.
- 9.5-4.** Consider the chamber temperature control system of Fig. P9.2-3. For this problem, replace the sensor gain  $H = 0.04$  with the gain  $H = 1$ . The system is now a unity-gain feedback system.  
 (a) Work Problem 9.2-3 with  $H = 1$ .  
 (b) Work Problem 9.3-2 with  $H = 1$ .  
 (c) Work Problem 9.5-2 with  $H = 1$ .  
 (d) Work Problem 9.5-3 with  $H = 1$ .
- 9.5-5.** Repeat all parts of Problem 9.3-3, using a current observer.
- 9.5-6.** (a) Show that for the current observer specified in Section 9.5, the transfer matrix from the input  $\mathbf{U}(z)$  to the estimated states  $\mathbf{Q}(z)$  is equal to that from the input to the states  $\mathbf{X}(z)$ ; that is, show that  $\mathbf{Q}(z)/\mathbf{U}(z) = \mathbf{X}(z)/\mathbf{U}(z)$ .  
 (b) Show that for the current observer specified in Section 9.5, the transfer function of the control-observer combination of Fig. 9-8 is given by (9-73), which is

$$D_{ce}(z) = z\mathbf{K}[z\mathbf{I} - \mathbf{A} + \mathbf{GCA} + \mathbf{BK} - \mathbf{GCBK}]^{-1}\mathbf{G}$$

- 9.5-7.** Derive the closed-loop state model for the pole-placement current-estimator design, given in (9-74).
- 9.6-1.** Consider a system described by (9-82).

$$\begin{aligned}\mathbf{x}(k + 1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k)\end{aligned}$$

For the case that  $\mathbf{u}(k)$  is not zero, derive the conditions for observability.

- 9.6-2.** Consider the plant of Example 9.2, which is

$$\mathbf{x}(k + 1) = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.9048 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k)$$

Suppose that the output is given by

$$y(k) = [0 \quad 1]\mathbf{x}(k)$$

- (a) Is this system observable?  
 (b) Explain the reason for your answer in part (a) in terms of the physical aspects of the system.
- 9.6-3.** Consider the satellite control system of Problem 9.2-4. Suppose that the output is the measurement of angular velocity, such that
- $$y(k) = [0 \quad 1]\mathbf{x}(k)$$
- (a) Is this system observable?  
 (b) Explain your answer in part (a) in terms of the physical aspects of the system. We are attempting to estimate position, given velocity.
- 9.6-4.** Consider the temperature control system of Problem 9.2-3.  
 (a) Determine if this system is controllable.  
 (b) An observer is added to this system in Problem 9.3-2, with the equation [see (9-38)]

$$\begin{aligned}q(k + 1) &= (\mathbf{A} - \mathbf{GC})q(k) + \mathbf{G}y(k) + \mathbf{B}u(k) \\ &= 0.3012q(k) + 10.99y(k) + 1.0368u(k)\end{aligned}$$

Construct a single set of state equations for the plant–observer system, with the state vector  $[x(k) \ q(k)]^T$  and the input  $u(k)$  (i.e., there is no feedback).

- (c) Show that this plant–observer system is uncontrollable.  
 (d) Based on the results of part (a), which state is uncontrollable?

**9.6-5.** Consider the first-order plant described by

$$x(k + 1) = Px(k) + Qu(k)$$

$$y(k) = Rx(k)$$

and a prediction observer described by (9-38).

- (a) Construct a single set of state equations for the plant–observer system, with the state vector  $[x(k) \ q(k)]^T$  and the input  $u(k)$  (i.e., there is no feedback).  
 (b) Show that this system is always uncontrollable, regardless of the value of observer gain  $G$ .
- 9.6-6.** Repeat Problem 9.6-5 for the current observer.
- 9.6-7.** For the system of Problem 9.6-5, use a transfer function approach to show that the transfer function  $Q(z)/U(z)$  is first order (even though the system is second order) and is equal to  $X(z)/U(z)$ . Hence the mode of  $Q(z)$ , which is  $(A - GC)^k$ , is not excited by the input signal  $u(k)$ .
- 9.6-8.** Problem 9.6-7 is to be repeated for the current observer, with the mode of  $Q(z)$  equal to  $(P - GRP)^k$ .
- 9.7-1.** In Problem 9.2-4, a pole-placement design for a satellite control system results in the gain matrix  $\mathbf{K} = [0.194 \ 0.88]$ . It is desired to have an input signal  $r(t)$  applied to the system, so as to realize the system of Fig. 9-12. Write the resulting state equations in the form

$$\mathbf{x}(k + 1) = \mathbf{A}_1\mathbf{x}(k) + \mathbf{B}_1r(k)$$

$$y(k) = \mathbf{C}_1\mathbf{x}(k)$$

Evaluate all matrices with  $\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ ,  $\mathbf{B} = \begin{bmatrix} 0.125 \\ 0.25 \end{bmatrix}$ , and  $\mathbf{A}\mathbf{B} = \begin{bmatrix} 0.375 \\ 0.25 \end{bmatrix}$

**9.7-2.** Repeat Problem 9.7-1 for the temperature control system of Problem 9.2-3, where the state equations are given by

$$x(k + 1) = 0.2402x(k) + 1.2343u(k)$$

$$y(k) = 0.07x(k)$$

and where the control gain is  $K = 0.3652$ .

# 10

## System Identification of Discrete-Time Systems

### 10.1 INTRODUCTION

System identification is defined as the process of constructing the mathematical model of a system from a judicious combination of the measurements of its outputs and possibly some inputs, if any. System identification becomes particularly important when one does not know the exact physical law describing the underlying operating principles of a system, which is often the case with many engineering systems as well as systems that exist in nature. For example, one may not know the exact mathematical relationship between the angle by which a robot may turn when an external disturbance torque is applied on it while it is performing a given task. This information may, in fact, be crucial for designing the robot itself so that it can sustain the worst-case turns under adversarial conditions during a mission. One way to solve this problem would be to apply various torques of increasing magnitude to the robot, measure the respective turn angles using sophisticated motion sensors, and then apply system identification techniques using the input (torque) and output (turn angle) measurements to derive a mathematical relationship between the two. The derived relationship can, thereafter, be used to predict the angle by which the robot may turn if any torque is applied. It can also be used to design appropriate controllers so that the robot survives the worst-case torque.

The complexity of system identification techniques often depends on how much prior knowledge one has about the system under study. For example, one may have very limited or almost no knowledge of how a given system works, or what may be the laws of physics governing its internal principles, or what may be a possible order of its dynamics, in which case the identification is referred to as unstructured or black-box identification. Depending on circumstances, however, sometimes the user may know several pieces of prior information about the system that may be helpful in identifying it. For example, given two different robots, we know *a priori* that both of their motions are dictated by Newton's laws, which helps us in visualizing a structured form of their mathematical models, but we may not know

the actual values of their respective inertias or time constants. Such cases are referred to as structured, or grey-box identification, indicating that the search for the perfect model is not totally unguided. Grey-box identification is often alternatively referred to as parameter estimation, meaning that the overall structure of the model may be known but the model parameters are not known, and need to be estimated from measurements of inputs and outputs. Once a reliable mathematical model of the system is identified, it can be used for both prediction and control.

In this chapter we will describe several system identification techniques for discrete-time static and dynamic systems. For simplicity, we will limit our discussion only to linear time-invariant (LTI) systems, although there exists a rich research literature of identification of nonlinear systems too. We will first describe methods for constructing algebraic functions for unknown static systems. Thereafter, we will present methods to identify state-variable models (identification in time domain) and transfer function models (identification in frequency domain) for dynamic systems using input and output measurements. An interesting point to note in this context is that identification, especially the identification of dynamic systems, is quite an open-ended problem as one can always add an uncountable number of fast poles or zeros to an identified model while preserving its input and output trajectories to nearly the same as originally measured, and thereby make the model non-unique. The identification methods described in this chapter are, therefore, meant to identify the simplest possible models that match the input and output measurements without any further pole-zero cancellations in their transfer functions. Another important factor dictating the accuracy of these techniques is the right choice of sampling interval. If the outputs, for example, are sampled too slowly then they may not capture all the important features of the transient responses of its state variables, and thereby produce an incorrect model. Depending on the time constant of the system, one, therefore, must ensure that the sampling rate is high enough to capture the contribution of all the poles and zeros in the measured output.

## 10.2 IDENTIFICATION OF STATIC SYSTEMS

Static systems are systems with no memory. They are modeled by algebraic relationships between their inputs  $u(k)$  and their outputs  $y(k)$ , where  $k$  denotes sample number. Identification of such system models is relatively easier and more straightforward than for dynamic systems. A typical approach is to use a parametric dependence between  $y(k)$  and nonlinear functions of  $u(k)$ , which are often referred to as *basis* functions. The model is of the form

$$y(k) = \beta_0 + \beta_1\varphi_1(u(k)) + \beta_2\varphi_2(u(k)) + \cdots + \beta_n\varphi_n(u(k)) \quad (10-1)$$

where the coefficients  $\beta_i$  are unknown real numbers,  $\varphi_i(u(k))$  are known real-valued functions of the input  $u(k)$ , and  $n > 0$  is an integer that defines the functional degree of  $\varphi_i(u(k))$ . The integer  $n$  may be known or unknown. The problem is to estimate the constants  $\beta_i$  using measured values of  $u(k)$  and  $y(k)$ . If the constant  $n$  is not known, then a usual approach is to start with a guess for  $n$ , estimate  $\beta_0, \beta_1, \dots, \beta_n$ , and check whether  $\hat{y}(k)$ , the output of the identified model, matches the measured output  $y(k)$  well enough for the same values of the input  $u(k)$ . If the two outputs don't match well, then one may use a higher value of  $n$  and repeat the estimation until a reasonable matching is obtained. For a given value of  $n$  one way to estimate the parameters  $\beta_i$  is

to collect all the measured outputs in a matrix  $\mathbf{y}$ , all  $\beta_i$  in a vector  $\boldsymbol{\beta}$ , all  $\varphi_i(u(k))$  in a matrix  $\boldsymbol{\Phi}$ , and represent the equation in (10-1) in the following matrix form:

$$\underbrace{\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & \varphi_1(u(0)) & \varphi_2(u(0)) & \cdots & \varphi_n(u(0)) \\ 1 & \varphi_1(u(1)) & \varphi_2(u(1)) & \cdots & \varphi_n(u(1)) \\ 1 & \varphi_1(u(2)) & \varphi_2(u(2)) & \cdots & \varphi_n(u(2)) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \varphi_1(u(N)) & \varphi_2(u(N)) & \cdots & \varphi_n(u(N)) \end{bmatrix}}_{\boldsymbol{\Phi}} \underbrace{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}}_{\boldsymbol{\beta}} \quad (10-2)$$

The number of samples  $N$  must be chosen to be much larger than  $n$ . This means that the matrix  $\boldsymbol{\Phi}$  is a skinny matrix, that is, it has more rows than columns. The unknown parameter vector  $\boldsymbol{\beta}$  can then be estimated using the formula

$$\hat{\boldsymbol{\beta}} = \underbrace{(\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T}_{\boldsymbol{\Phi}^+} \mathbf{y} \quad (10-3)$$

where the matrix  $\boldsymbol{\Phi}^+$  is referred to as the Moore–Penrose pseudo-inverse of  $\boldsymbol{\Phi}$ . Yet another way to derive the same answer is to pose the problem in the following way. For every sample  $k$ , define an error term  $e(k)$  as

$$e(k) = y(k) - \boldsymbol{\varphi}^T(k) \boldsymbol{\beta} \quad (10-4)$$

where  $y(k)$  is measured output, and  $\boldsymbol{\varphi}^T(k) = [1 \ \varphi_1(u(k)) \ \varphi_2(u(k)) \ \cdots \ \varphi_n(u(k))]$ . The unknown parameter vector  $\boldsymbol{\beta}$  can then be estimated as a solution of the optimization problem

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sqrt{\sum_{k=0}^N e^2(k)} \quad (10-5)$$

The solution of the above problem will be the same as in (10-3), which is often referred to as the least-squares solution. The resulting  $\hat{\boldsymbol{\beta}}$  is referred to as the least square estimate, which is optimal in the sense of minimizing the sum of the squared error between the theoretical output and the measured output as in (10-5).

A pertinent question to ask at this point is how to choose the basis functions  $\varphi_i$  in (10-1) for any given set of input and output measurements. While the answer depends on how much prior information about the system behavior is known, a common practice for identifying many real-world applications is to use  $\varphi_i$  as a polynomial function of the input  $u(k)$ . For example, choosing a degree  $n$ , a polynomial approximation of (10-1) is given by

$$y(k) = \beta_0 + \beta_1 u(k) + \beta_2 u^2(k) + \cdots + \beta_{n-1} u^{n-1}(k) + \beta_n u^n(k) \quad (10-6)$$

Equation (10-2) in that case takes the special form



$$\underbrace{\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & u(0) & u^2(0) & \cdots & u^n(0) \\ 1 & u(1) & u^2(1) & \cdots & u^n(1) \\ 1 & u(2) & u^2(2) & \cdots & u^n(2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u(N) & u^2(N) & \cdots & u^n(N) \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}}_{\beta} \quad (10-7)$$

the solution for which follows from (10-3). In MATLAB, the Equation (10-7) can be solved for the unknown parameter vector  $\beta$  using the function *polyfit*. We next cite an example to highlight the use of this function for a specific set of input–output measurements.

### EXAMPLE 10.1

Consider the following data set, where  $u(k)$  is the independent variable (input), and  $y(k)$  is the dependent variable (output) of a static system, for  $k = 1, 2, \dots, 10$ . Find a polynomial fit between  $y(k)$  and  $u(k)$ .

$k$	1	2	3	4	5	6	7	8	9	10
$u(k)$	1	2	3	4	5	6	7	8	9	10
$y(k)$	0	100	800	4200	15800	46900	108000	262600	532000	1000700

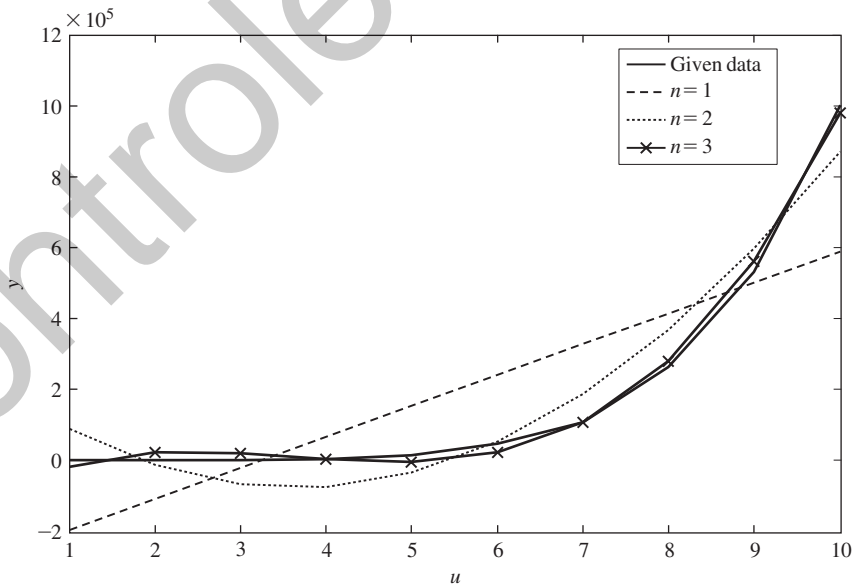


FIGURE 10-1 Polynomial fits for input–data data.

Using the *polyfit* function in MATLAB we get the following results for different values of the degree  $n$ :

$$n = 1: \quad y(k) = -282260 + 87160u(k)$$

$$n = 2: \quad y(k) = 235870 - 171910u(k) + 23550u^2(k)$$

$$n = 3: \quad y(k) = -127940 + 150770u(k) - 46410u^2(k) + 4240u^3(k)$$

The three different polynomial fits are shown in Fig. 10-1 along with the actual data. It can be seen that the fit for  $n = 3$  fits the actual data fairly closely.

### 10.3 IDENTIFICATION OF DYNAMIC SYSTEMS

While the Least Square method stated in (10-5) gives a reasonably accurate way of identifying static systems, the problem of identifying dynamic systems is much more complex. For a discrete-time LTI system the problem can be posed as follows. Consider the state-variable model

$$\begin{aligned} \mathbf{x}(k + 1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \end{aligned} \tag{10-8}$$

where  $\mathbf{x} \in R^n$ ,  $\mathbf{u} \in R^m$ , and  $\mathbf{y} \in R^p$ ,  $n$  is the order of the system. The integer  $n$  and the four matrices  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  are unknown. Given  $N$  measured samples of  $\mathbf{y}(k)$  and  $\mathbf{u}(k)$ , possibly corrupted with measurement noise, our objective is to estimate  $n$  and the four system matrices  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $\mathbf{D}$  up to a similarity transformation. There can be several ways of solving this problem, depending on noise levels in the measurements. We next describe one possible way to estimate the four matrices, assuming that the measurements do not have any noise. The case for noisy measurements can be found in details in [1] and [2]. We first describe the method when absolutely no prior information about  $n$  and any of the system matrices are known, referred to as *black-box identification*, which is usually approached in time-domain. Thereafter, we revisit the *least-squares method* and describe how it can be used for solving the above problem when the number of poles and zeros of the transfer function of (10-8) is known *a priori*. This problem is solved in frequency-domain.

### 10.4 BLACK-BOX IDENTIFICATION

For simplicity of explanation, and without any loss of generality, we will assume (10-8) to be a single-input single-output (SISO) system, that is,  $m = p = 1$ . We will also assume the input  $u(k)$  to be of the form

$$u(k) = \begin{cases} 1, & k = 0 \\ 0, & k = 1, 2, \dots, N \end{cases} \tag{10-9}$$

indicating that the input is a short-lived unit impulse. Since the system in (10-8) is linear, we also assume without any loss of generality that the initial value of the state  $x(k)$  is zero, that is,  $\mathbf{x}(0) = \mathbf{0}$ . Before proceeding further, we need to make three important assumptions:

**Assumption 1:** The LTI system (10-8) is both controllable and observable.

**Assumption 2:** The input  $u(k)$  is persistently exciting, that is, the signal  $u(k)$  is rich enough to stir the contribution of all the dominant dynamic modes in the output  $y(k)$ .

**Assumption 3:** The intersection of the row space  $\mathbf{U}_f$  (the future inputs) and the row space of  $\mathbf{X}_p$  (the past states) is empty. These two matrices for any  $f$  and  $p$  are defined as follows:

$$\mathbf{U}_f = \begin{bmatrix} u(i) & u(i+1) & u(i+2) & \cdots & u(i+j-2) & u(i+j-1) \\ u(i+1) & u(i+2) & u(i+3) & \cdots & u(i+j-1) & u(i+j) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ y(f) & y(f+1) & y(f+2) & \cdots & y(f+j-2) & y(f+j-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ y(2i-1) & y(2i) & y(2i+1) & \cdots & y(2i+j-1) & y(2i+j-2) \end{bmatrix}$$

$$\mathbf{X}_p = \begin{bmatrix} x(0) & x(1) & x(2) & \cdots & x(j-2) & x(j-1) \\ x(1) & x(2) & x(3) & \cdots & x(j-1) & x(j) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x(p) & x(p+1) & x(p+2) & \cdots & x(p+j-2) & x(p+j-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x(i-1) & x(i) & x(i+1) & \cdots & x(i+j-3) & x(i+j-2) \end{bmatrix}$$

With these three assumptions in place, we next describe a step-by-step method for estimating  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $D$  in (10-8) using  $u(k)$  and  $y(k)$ ,  $k = 0, 1, 2, \dots, p$ . The first step is to note that since  $\mathbf{x}(0) = \mathbf{0}$  and  $u(0) = 1$ , from (10-8) we get

$$y(0) = \mathbf{C}\mathbf{x}(0) + Du(0) = D \quad (10-10)$$

This means that the value of the scalar  $D$  is simply equal to that of the very first sample of the output. We further exploit this fact to find an estimate for  $\mathbf{A}$  next. From the definition of the input, the initial state, and the model (10-8) we note the following pattern:

$$\begin{aligned}
 y(1) &= \mathbf{C}\mathbf{x}(1) + Du(1) = \mathbf{C}(\mathbf{A}\mathbf{x}(0) + \mathbf{B}u(0)) = \mathbf{C}\mathbf{B} \\
 y(2) &= \mathbf{C}\mathbf{x}(2) + Du(2) = \mathbf{C}(\mathbf{A}\mathbf{x}(1) + \mathbf{B}u(1)) = \mathbf{C}\mathbf{A}\mathbf{B} \\
 y(3) &= \mathbf{C}\mathbf{x}(3) + Du(3) = \mathbf{C}(\mathbf{A}\mathbf{x}(2) + \mathbf{B}u(2)) = \mathbf{C}\mathbf{A}^2\mathbf{B}
 \end{aligned}$$

and so on. In general, we can write

$$y(i) = \mathbf{C}\mathbf{A}^{i-1}\mathbf{B} \quad (10-11)$$

where  $i = 1, 2, \dots, p$ . We next construct a matrix  $\mathbf{H}_1$ , referred to as the Hankel matrix, by gathering the output samples in the following way:

$$\mathbf{H}_1 = \begin{bmatrix} y(1) & y(2) & y(3) & \cdots & y(M-1) & y(M) \\ y(2) & y(3) & y(4) & \cdots & y(M) & y(M+1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ y(i) & y(i+1) & y(i+2) & \cdots & y(M+i-2) & y(M+i-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ y(N) & y(N+1) & y(N+2) & \cdots & y(M+N-2) & y(M+N-1) \end{bmatrix} \quad (10-12)$$

The constants  $M$  and  $N$  may be chosen arbitrarily such that  $M + N - 1 < p$ , but both must be sufficiently large so that they preserve the order  $n$  of the system. Using (10-11) and (10-12), we can also write

$$\mathbf{H}_1 = \begin{bmatrix} CB & CAB & CA^2B & \cdots & CA^{M-2}B & CA^{M-1}B \\ CAB & CA^2B & CA^3B & \cdots & CA^{M-1}B & CA^M B \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{i-1}B & CA^iB & CA^{i+1}B & \cdots & CA^{M+i-3}B & CA^{M+i-2}B \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{N-1}B & CA^N B & CA^{N+1}B & \cdots & CA^{M+N-3}B & CA^{M+N-2}B \end{bmatrix} \quad (10-13)$$

Inspecting the structure of the matrix on the right-hand side (RHS) of (10-13), it is easy to see that  $\mathbf{H}_1$  can also be written as

$$\mathbf{H}_1 = \mathbf{O}\mathbf{L} \quad (10-14)$$

where

$$\mathbf{O} = [\mathbf{C}^T \quad (\mathbf{C}\mathbf{A})^T \quad (\mathbf{C}\mathbf{A}^2)^T \quad \cdots \quad (\mathbf{C}\mathbf{A}^{N-1})^T]^T$$

is the observability matrix, and

$$\mathbf{L} = [\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B} \quad \cdots \quad \mathbf{A}^{M-1}\mathbf{B}]$$

is the controllability matrix of the system (10-8). Since both  $M$  and  $N$  are chosen to be significantly larger than  $n$ , we next consider the singular value decomposition (SVD) of  $\mathbf{H}_1$  to eliminate any unnecessary modes contained in  $\mathbf{H}_1$ . The SVD is of the form

$$\mathbf{H}_1 = [\mathbf{U}_1^T \quad \mathbf{U}_2^T] \begin{bmatrix} \boldsymbol{\Sigma}_1 & 0 \\ 0 & \boldsymbol{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} \quad (10-15)$$

where  $\mathbf{U}_1$ ,  $\mathbf{U}_2$ ,  $\mathbf{V}_1$ , and  $\mathbf{V}_2$  are unitary matrices, while  $\boldsymbol{\Sigma}_1$  and  $\boldsymbol{\Sigma}_2$  are square diagonal matrices that have the singular values of  $\mathbf{H}_1$ .  $\boldsymbol{\Sigma}_1$  is chosen such that it has the most dominant nonzero singular values along its diagonal, while the diagonal entries of  $\boldsymbol{\Sigma}_2$  are zero or almost zero. These singular values are then treated as negligible, and the system order is simply reduced to the dimension of  $\boldsymbol{\Sigma}_1$ . Therefore, the value of  $n$  is considered to be equal to the number of the diagonal entries of  $\boldsymbol{\Sigma}_1$ . A reduced-order Hankel matrix is thereafter defined as:

$$\bar{\mathbf{H}}_1 = \mathbf{U}_1^T \boldsymbol{\Sigma}_1 \mathbf{V}_1 \quad (10-16)$$

From (10-14) and (10-16), we can also easily derive that

$$\bar{\mathbf{H}}_1 = \bar{\mathbf{O}}\bar{\mathbf{L}} \quad (10-17)$$

where

$$\bar{\mathbf{O}} = [\mathbf{C}^T \quad (\mathbf{C}\mathbf{A})^T \quad (\mathbf{C}\mathbf{A}^2)^T \quad \cdots \quad (\mathbf{C}\mathbf{A}^{n-1})^T]^T$$

and

$$\bar{\mathbf{L}} = [\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B} \quad \cdots \quad \mathbf{A}^{n-1}\mathbf{B}]$$

are the full-rank  $n \times n$  observability and controllability matrices, respectively. Equation (10-16) can also be written as:

$$\bar{\mathbf{H}}_1 = \mathbf{U}_1^T \boldsymbol{\Sigma}_1^{\frac{1}{2}} \boldsymbol{\Sigma}_1^{\frac{1}{2}} \mathbf{V}_1 = (\boldsymbol{\Sigma}_1^{\frac{T}{2}} \mathbf{U}_1)^T (\boldsymbol{\Sigma}_1^{\frac{1}{2}} \mathbf{V}_1) \quad (10-18)$$

Then using Assumption 3 it can be shown from (10-17) and (10-18) that the following is true:

$$\bar{\mathbf{O}} = (\boldsymbol{\Sigma}_1^{\frac{T}{2}} \mathbf{U}_1)^T \quad (10-19)$$

$$\bar{\mathbf{L}} = \boldsymbol{\Sigma}_1^{\frac{T}{2}} \mathbf{V}_1 \quad (10-20)$$

Since the value of  $n$  is already known, we next construct another Hankel matrix of the following form:

$$\mathbf{H}_2 = \begin{bmatrix} y(2) & y(3) & y(4) & \cdots & y(n) & y(n+1) \\ y(3) & y(4) & y(5) & \cdots & y(n+1) & y(n+2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ y(i) & y(i+1) & y(i+2) & \cdots & y(n+i-2) & y(n+i-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ y(n+1) & y(n+2) & y(n+3) & \cdots & y(2n-1) & y(2n) \end{bmatrix} \quad (10-21)$$

From (10-10) it is easy to see that  $\mathbf{H}_2$  can be written as

$$\mathbf{H}_2 = \begin{bmatrix} \mathbf{CA}^1\mathbf{B} & \mathbf{CA}^2\mathbf{B} & \mathbf{CA}^3\mathbf{B} & \cdots & \mathbf{CA}^{n-1}\mathbf{B} & \mathbf{CA}^n\mathbf{B} \\ \mathbf{CA}^2\mathbf{B} & \mathbf{CA}^3\mathbf{B} & \mathbf{CA}^4\mathbf{B} & \cdots & \mathbf{CA}^n\mathbf{B} & \mathbf{CA}^{n+1}\mathbf{B} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{CA}^i\mathbf{B} & \mathbf{CA}^{i+1}\mathbf{B} & \mathbf{CA}^{i+2}\mathbf{B} & \cdots & \mathbf{CA}^{n+i-2}\mathbf{B} & \mathbf{CA}^{n+i-1}\mathbf{B} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{CA}^n\mathbf{B} & \mathbf{CA}^{n+1}\mathbf{B} & \mathbf{CA}^{n+2}\mathbf{B} & \cdots & \mathbf{CA}^{2n-2}\mathbf{B} & \mathbf{CA}^{2n-1}\mathbf{B} \end{bmatrix} \quad (10-22)$$

which, using the definitions of  $\bar{\mathbf{O}}$  and  $\bar{\mathbf{L}}$ , can also be written as

$$\mathbf{H}_2 = \bar{\mathbf{O}} \bar{\mathbf{A}} \bar{\mathbf{L}} \quad (10-23)$$

Using (10-19) and (10-20), the above expression can be rewritten as

$$\mathbf{H}_2 = (\boldsymbol{\Sigma}_1^{\frac{T}{2}} \mathbf{U}_1)^T \bar{\mathbf{A}} (\boldsymbol{\Sigma}_1^{\frac{1}{2}} \mathbf{V}_1) \quad (10-24)$$

from which an estimate for  $\bar{\mathbf{A}}$  can be constructed as

$$\hat{\bar{\mathbf{A}}} = (\boldsymbol{\Sigma}_1^{\frac{T}{2}} \mathbf{U}_1)^{-T} \mathbf{H}_2 (\boldsymbol{\Sigma}_1^{\frac{1}{2}} \mathbf{V}_1)^{-1} \quad (10-25)$$

All the matrices on the RHS of (10-25) are known.  $\mathbf{H}_2$  is known as the user creates  $\mathbf{H}_2$  from the measured samples of  $y(k)$  as in (10-21), while  $\mathbf{U}_1$ ,  $\mathbf{V}_1$ , and  $\boldsymbol{\Sigma}_1$  are known from the SVD of the known matrix  $\mathbf{H}_1$ . Therefore, (10-25) gives us a possible way of computing  $\hat{\bar{\mathbf{A}}}$  from the knowledge of  $u(k)$  and  $y(k)$ .

Similarly, the matrices  $\mathbf{C}$  and  $\mathbf{B}$  can be computed in the following way. We know that

$$\bar{\mathbf{O}} = \tilde{\mathbf{C}} \underbrace{\begin{bmatrix} \mathbf{I} \\ \mathbf{A} \\ \vdots \\ \mathbf{A}^{n-1} \end{bmatrix}}_{\Lambda_1} \quad (10-26)$$

where  $\tilde{\mathbf{C}}$  is a block diagonal matrix whose diagonal blocks are  $\mathbf{C}$ . Then using (10-19) and (10-26) we can write

$$\tilde{\mathbf{C}} = (\Sigma_1^{\frac{1}{2}} \mathbf{U}_1)^T \Lambda_1^{-1} \quad (10-27)$$

Finally,  $\mathbf{C}$  can be estimated by extracting the first  $n$  elements of the first row of  $\tilde{\mathbf{C}}$ . In a similar way, we have

$$\bar{\mathbf{L}} = \underbrace{[\mathbf{I} \quad \mathbf{A} \quad \dots \quad \mathbf{A}^{n-1}]}_{\Lambda_2} \tilde{\mathbf{B}} \quad (10-28)$$

where  $\tilde{\mathbf{B}}$  is a block diagonal matrix with the diagonal blocks being  $\mathbf{B}$ . Thus, from (10-20) and (10-28) we get

$$\tilde{\mathbf{B}} = \Lambda_2^{-1} (\Sigma_2^{\frac{1}{2}} \mathbf{V}_1) \quad (10-29)$$

Then  $\mathbf{B}$  can be estimated by extracting the first  $n$  elements of the first column of  $\tilde{\mathbf{B}}$ . Another way to estimate  $\mathbf{C}$  and  $\mathbf{B}$  is to extract them directly from the structure of the two matrices  $\bar{\mathbf{O}}$  and  $\bar{\mathbf{L}}$ . The matrix  $\bar{\mathbf{O}}$  is known from (10-19) and  $\bar{\mathbf{L}}$  is known from (10-20). Therefore, using (10-17a),  $\mathbf{C}$  can simply be extracted as the first row of  $\bar{\mathbf{O}}$ , while  $\mathbf{B}$  can be extracted from the first column of  $\bar{\mathbf{L}}$ . This is often the easiest way to estimate  $\mathbf{C}$  and  $\mathbf{B}$ . In the Example 10.2 we will use this approach.

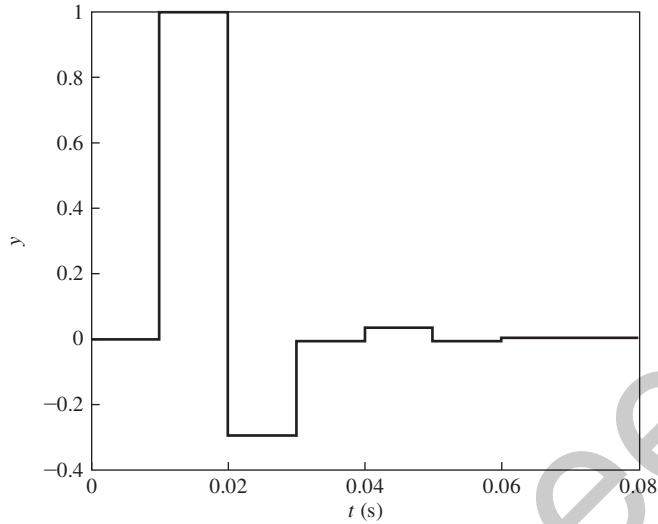
It must, however, be remembered that state-space identification is never unique, that is, the estimates of  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$ , and  $D$  are accurate only up to a similarity transformation of the actual state-space model.

### EXAMPLE 10.2

Consider the following impulse response data for a SISO discrete-time LTI system with a sampling time of  $T = 0.01$  s. The output is measured at the sampling time instants  $t = 0, 0.01, 0.02, \dots, 0.08$ , as shown in Fig. 10-2.

$k$	0	1	2	3	4	5	6	7	8
$y(k)$	0	0.996	-0.298	-0.0148	0.0338	-0.0081	-0.0011	0.0011	-0.0002

Our objective is to identify a similarity transformation of the system matrices  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$ , and the corresponding transfer function, using the given values of  $y(k)$  and with the information



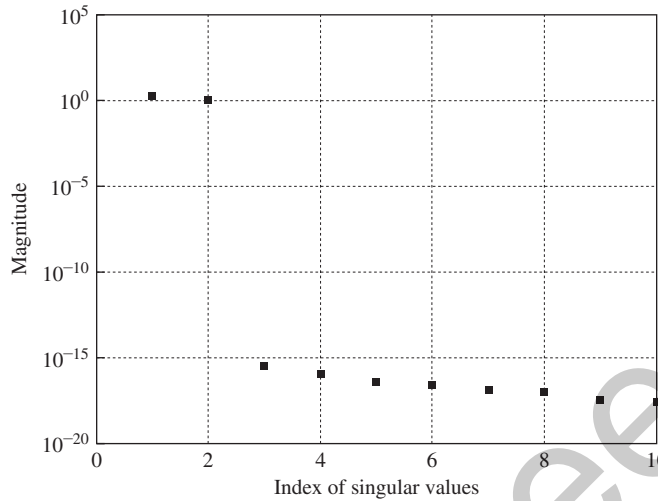
**FIGURE 10-2** Measured output  $y(k)$  plotted against time.

that  $u(k)$  is an impulse. We use the following MATLAB code to estimate these matrices based on the black-box identification algorithm described in Section 10.4.

```
T=0.01;
y=[0;0.996;-0.298;-0.0148;0.0338;-0.0081;-0.0011;0.0011;-0.0002];
p=length(y)-1;
M=p/2; N=M;
for i=1:N
    for j=1:M
        H1(i,j)=y(i+j);
    end
end;
[u,s,v]=svd(H1);
n=2;
ubar=(u(:,1:n))';
sbar=s(1:n,1:n);
vbar=v(1:n,:);
for i=1:M
    for j=1:N
        H2(i,j)=y(i+j+1);
    end
end;
int1=(sqrt(sbar)*ubar)';
int2=sqrt(sbar)*vbar;
Ahat=pinv(int1)*H2*pinv(int2)
Bhat=int2(:,1)
Chat=int1(1,:)
[num1,den1]=ss2tf(Ahat,Bhat,Chat,0);
Ghat=tf(num1,den1,T)
```

The singular values of the Hankel matrix  $\mathbf{H}_1$  are shown in Fig. 10-3, from which it is clear that the system order  $n$  can be well approximated as 2. The identified system matrices are

$$\hat{\mathbf{A}} = \begin{bmatrix} -0.2511 & 0.2599 \\ -0.3010 & -0.0284 \end{bmatrix}, \quad \hat{\mathbf{B}} = \begin{bmatrix} -1.0012 \\ 0.0853 \end{bmatrix}, \quad \hat{\mathbf{C}} = [-1.0012 \quad -0.0800], \quad \hat{\mathbf{D}} = 0$$



**FIGURE 10-3** Singular values of the Hankel matrix  $H_1$ .

and the corresponding discrete-time transfer function is

$$\hat{G}(z) = \frac{0.9956z - 0.01955}{z^2 + 0.2795z + 0.08534}$$

The actual system matrices from which the output  $y(k)$  is originally generated are

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -0.1 & -0.3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = [0 \quad 1]$$

with the actual transfer function being

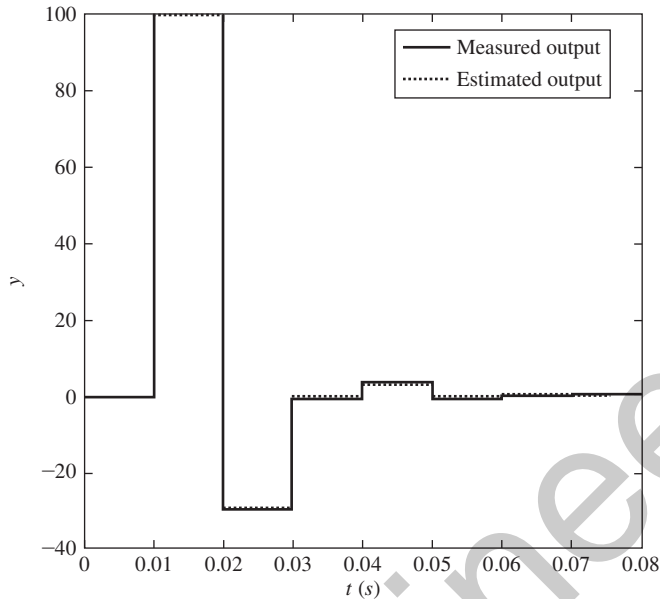
$$G(z) = \frac{z}{z^2 + 0.3z + 0.1}$$

The identified transfer function is, therefore, almost the same as the actual transfer function. It should be noted, however, that the identified state-variable model is not the same as the actual state-variable model. They are the similarity transformations of each other, neglecting minor differences due to numerical errors. We next present the following MATLAB code to compare the impulse response of the identified system with that of the actual system:

```
% Append this code to the program above
A=[0 1;-.1 -.3]; B=[0;1]; C=[0 1]; D=0;
sys1=ss(A,B,C,0,T);
sys2=ss(Ahat,Bhat,Chat,0,T);
t=[0:T:0.08];
stairs(t,impz(sys1,t),'-k','LineWidth',2);
hold on;stairs(t,impz(sys2,t),':k','LineWidth',2);
stepinfo(sys1), stepinfo(sys2)
```

The *stepinfo* of the two systems are compared below and the step responses are displayed in Fig. 10-4.





**FIGURE 10-4** Measured output versus estimated output.

Actual System	Identified System
Rise Time: 0	Rise Time: 0
Settling Time: 0.0400	Settling Time: 0.0400
Settling Min: 0.6900	Settling Min: 0.6960
Settling Max: 1	Settling Max: 0.9956
Overshoot: 40.0000	Overshoot: 39.2168
Undershoot: 0	Undershoot: 0
Peak: 1	Peak: 0.9956
Peak Time: 0.0100	Peak Time: 0.0100

## 10.5 LEAST-SQUARES SYSTEM IDENTIFICATION

In the previous section we presented an algorithm for black-box identification considering that no prior information about the model (10-8) is available to the user. In this section we will present a variant of the *least-squares* method introduced in Section 10.2 to identify this model when the number of poles and zeros of the transfer function of (10-2) from  $u(k)$  to  $y(k)$  is known. The method is referred to as *least-squares system identification*. We will see that it is a simple extension of least-squares curve fitting. Since the number of poles and zeros of (10-8) is known, we assume the system transfer-function model to be of the

form (2-51), but here we are changing the coefficient numbering order to make the notation easier to follow:

$$\frac{Y(z)}{U(z)} = G(z) = \frac{b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \cdots + b_{n-1}z + b_n}{z^n - a_1z^{n-1} - \cdots - a_{n-1}z - a_n} \quad (10-30)$$

where  $U(z)$  is the input and  $Y(z)$  is the output. Hence the system is described by the difference equation

$$y(k) = a_1y(k-1) + a_2y(k-2) + \cdots + a_ny(k-n) + b_1u(k-1) + b_2u(k-2) + \cdots + b_nu(k-n) \quad (10-31)$$

This model is often referred to as the *autoregressive moving-average* (ARMA) model. Our objective is to determine the coefficient vector

$$\boldsymbol{\theta} = (a_1 \ a_2 \ \cdots \ a_n \ b_1 \ b_2 \ \cdots \ b_n)^T \quad (10-32)$$

from measurements of the input–output sequences  $u(k)$  and  $y(k)$ .

To illustrate the procedure, we first consider the first-order case, with

$$\frac{Y(z)}{U(z)} = G(z) = \frac{b_1}{z - a_1}$$

Hence,

$$y(k) = a_1y(k-1) + b_1u(k-1)$$

and thus

$$y(1) = a_1y(0) + b_1u(0) + e(1)$$

$$y(2) = a_1y(1) + b_1u(1) + e(2)$$

$$y(3) = a_1y(2) + b_1u(2) + e(3)$$

where the error terms  $e(k)$  occur because of measurement inaccuracies, model inaccuracies, and so on. This equation can be expressed in vector-matrix form as

$$\begin{bmatrix} y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} y(0) & u(0) \\ y(1) & u(1) \\ y(2) & u(2) \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} e(1) \\ e(2) \\ e(3) \end{bmatrix}$$

which may be expressed as (this notation is explained later)

$$\mathbf{y}(3) = \mathbf{F}(3)\boldsymbol{\theta} + \mathbf{e}(3) \quad (10-33)$$

We wish then to calculate the coefficient vector  $\boldsymbol{\theta}$  that will minimize the sum of the squared errors between the predicted measurements and the actual measurements. The general procedure will now be developed.

Consider the  $n$ th-order ARMA model

$$y(k) = a_1y(k-1) + a_2y(k-2) + \dots + a_ny(k-n) + b_1u(k-1) + b_2u(k-2) + \dots + b_nu(k-n)$$

with the set of  $(N+1)$  measurement pairs.

$$\{u(0), y(0)\}, \{u(1), y(1)\}, \dots, \{u(N), y(N)\} \quad (10-34)$$

with  $N \geq n$ . Define the vector  $\mathbf{f}(k)$  by

$$\mathbf{f}^T(k) = [y(k-1) \ y(k-2) \ \dots \ y(k-n) \ u(k-1) \ \dots \ u(k-n)] \quad (10-35)$$

The first error that is a function of only known data measurements is  $e(n)$ . Then, for the sample periods  $n, n+1, \dots, N$ ,

$$\begin{aligned} y(n) &= \mathbf{f}^T(n)\boldsymbol{\theta} + e(n) \\ y(n+1) &= \mathbf{f}^T(n+1)\boldsymbol{\theta} + e(n+1) \\ &\vdots \\ y(N) &= \mathbf{f}^T(N)\boldsymbol{\theta} + e(N) \end{aligned} \quad (10-36)$$

where  $\boldsymbol{\theta}$  is given in (10-32). Using the notation

$$\mathbf{y}(N) = \begin{bmatrix} y(n) \\ y(n+1) \\ \vdots \\ y(N) \end{bmatrix}, \quad \mathbf{F}(N) = \begin{bmatrix} \mathbf{f}^T(n) \\ \mathbf{f}^T(n+1) \\ \vdots \\ \mathbf{f}^T(N) \end{bmatrix}, \quad \mathbf{e}(N) = \begin{bmatrix} e(n) \\ e(n+1) \\ \vdots \\ e(N) \end{bmatrix} \quad (10-37)$$

we can express (10-36) as

$$\mathbf{y}(N) = \mathbf{F}(N)\boldsymbol{\theta} + \mathbf{e}(N) \quad (10-38)$$

In (10-37) and (10-38),  $\mathbf{y}(N)$  is of order  $(N-n+1) \times 1$ ,  $\mathbf{F}(N)$  is  $(N-n+1) \times 2n$ ,  $\boldsymbol{\theta}$  is  $2n \times 1$ , and  $\mathbf{e}(N)$  is  $(N-n+1) \times 1$ .

Next the cost function  $J(\boldsymbol{\theta})$  is defined as the sum of the squared errors:

$$J(\boldsymbol{\theta}) = \sum_{k=n}^N e^2(k) = \mathbf{e}^T(N)\mathbf{e}(N) \quad (10-39)$$

Then, from (10-38) and (10-39),

$$\begin{aligned} J(\boldsymbol{\theta}) &= [\mathbf{y} - \mathbf{F}\boldsymbol{\theta}]^T[\mathbf{y} - \mathbf{F}\boldsymbol{\theta}] = \mathbf{y}^T\mathbf{y} - \boldsymbol{\theta}^T\mathbf{F}^T\mathbf{y} - \mathbf{y}^T\mathbf{F}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{F}^T\mathbf{F}\boldsymbol{\theta} \\ &= \mathbf{y}^T\mathbf{y} - 2\boldsymbol{\theta}^T\mathbf{F}^T\mathbf{y} + \boldsymbol{\theta}^T\mathbf{F}^T\mathbf{F}\boldsymbol{\theta} \end{aligned} \quad (10-40)$$

where the notational dependence of the terms on  $N$  has been omitted for convenience. Thus the value of  $\boldsymbol{\theta}$  that minimizes  $J(\boldsymbol{\theta})$  satisfies the equation

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -2\mathbf{F}^T \mathbf{y} + 2\mathbf{F}^T \mathbf{F} \boldsymbol{\theta} = 0 \quad (10-41)$$

Or we may write

$$\mathbf{F}^T \mathbf{F} \boldsymbol{\theta} = \mathbf{F}^T \mathbf{y} \quad (10-42)$$

The least-squares estimate of  $\boldsymbol{\theta}$  is then

$$\hat{\boldsymbol{\theta}}_{LS} = [\mathbf{F}^T(N)\mathbf{F}(N)]^{-1}\mathbf{F}^T(N)\mathbf{y}(N) \quad (10-43)$$

provided that the indicated inverse matrix exists. If the input sequence  $u(k)$  is *persistently exciting* (all system dynamics are sufficiently excited by the input) and if  $\boldsymbol{\theta}$  is *identifiable* (the parameters can be uniquely determined from the data) the inverse matrix will exist.

The structure of (10-43) is somewhat difficult to understand, because of the matrix transposes. As an example of this structure, consider a second-order system with four data points. Then  $N = 4$  and

$$\mathbf{F}^T(4)\mathbf{y}(4) = \begin{bmatrix} y(1) & y(0) & u(1) & u(0) \\ y(2) & y(1) & u(2) & u(1) \\ y(3) & y(2) & u(3) & u(2) \end{bmatrix}^T \begin{bmatrix} y(2) \\ y(3) \\ y(4) \end{bmatrix} \quad (10-44)$$

Note from (10-37) that the total number of data points is  $(N + 1)$ , and only the output in the last data point is used.

An example will now be given.

### EXAMPLE 10.3

Suppose that a first-order system yields the following data:

$k$	$u(k)$	$y(k)$
0	1.0	0
1	0.75	0.3
2	0.50	0.225

The assumed transfer function is

$$G(z) = \frac{b_1}{z - a_1}, \quad \boldsymbol{\theta} = \begin{bmatrix} a_1 \\ b_1 \end{bmatrix}$$

Thus

$$y(k) = a_1 y(k - 1) + b_1 u(k - 1) = [y(k - 1) \quad u(k - 1)] \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = \mathbf{f}^T(k)\boldsymbol{\theta}$$

From (10-37), since  $N = 2$ ,

$$\mathbf{F}(2) = \begin{bmatrix} \mathbf{f}^T(1) \\ \mathbf{f}^T(2) \end{bmatrix} = \begin{bmatrix} y(0) & u(0) \\ y(1) & u(1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0.3 & 0.75 \end{bmatrix}$$

Then

$$\mathbf{F}^T \mathbf{F} = \begin{bmatrix} 0 & 0.3 \\ 1 & 0.75 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0.3 & 0.75 \end{bmatrix} = \begin{bmatrix} 0.09 & 0.225 \\ 0.225 & 1.5625 \end{bmatrix}$$

and the inverse of this matrix is calculated to be

$$[\mathbf{F}^T \mathbf{F}]^{-1} = \begin{bmatrix} 17.361 & -2.5 \\ -2.5 & 1 \end{bmatrix}$$

The least-squares estimate of  $\boldsymbol{\theta}$  is then, from (10-43),

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{LS} &= [\mathbf{F}^T \mathbf{F}]^{-1} \mathbf{F}^T \mathbf{y} = \begin{bmatrix} 17.361 & -2.5 \\ -2.5 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0.3 \\ 1 & 0.75 \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.225 \end{bmatrix} \\ &= \begin{bmatrix} -2.5 & 3.333 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.225 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.3 \end{bmatrix} = \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \end{aligned}$$

Thus the transfer function is identified to be

$$G(z) = \frac{0.3}{z}$$

with the difference equation

$$y(k) = 0.3u(k - 1)$$

and the measurement data given above is seen to satisfy this difference equation exactly. For this example, the measurement data was initially generated from this difference equation. In a physical situation, the data would not fit exactly, and in fact the errors  $e(k)$  may be quite large.

In some cases we would like to weight the errors in the form

$$J(\boldsymbol{\theta}) = w(n)e^2(n) + w(n + 1)e^2(n + 1) + \cdots + w(N)e^2(N) \quad (10-45)$$

One common case is to weight the most recent data most heavily. In (10-45), the cost function can be expressed as

$$J(\boldsymbol{\theta}) = \sum_{k=n}^N w(k)e^2(k) = \mathbf{e}^T(N) \mathbf{W}(N) \mathbf{e}(N) \quad (10-46)$$

where  $\mathbf{W}(N)$  is a  $N \times N$  diagonal matrix with elements  $w_{ii} = w(i + n - 1)$ . If this cost function is minimized in the manner described above, the resulting weighted least-squares estimate is

$$\hat{\boldsymbol{\theta}}_{WLS} = [\mathbf{F}^T(N) \mathbf{W}(N) \mathbf{F}(N)]^{-1} \mathbf{F}^T(N) \mathbf{W}(N) \mathbf{y}(N) \quad (10-47)$$

A weighting term that is commonly used is

$$w(k) = a\gamma^{N-k}, \quad \gamma \leq 1 \quad (10-48)$$

If  $a$  is chosen such that  $a = (1 - \gamma)$ , the weighting is said to be *exponential*. For  $\gamma$ , smaller than unity, the most recent data dominates the estimation. As  $\gamma$  approaches unity, the more distant data has a larger influence. In (10-48), if  $a = \gamma = 1$ , the weighting matrix  $\mathbf{W}(N) = \mathbf{I}$ , and (10-47) becomes the equation for ordinary least-squares identification (10-43).

### EXAMPLE 10.4

Consider the input and output data of a LTI system with sampling time  $T = 0.1$  s:

$k$	$u(k)$	$y(k)$
0	1	0
1	1	1.0000
2	1	0.9000
3	1	0.7100
4	1	0.7490
5	1	0.7831
6	1	0.7719

We wish to generate the least-squares estimate of the transfer function from the given input–output data. For this one may write a simple MATLAB code using the function *arx* as follows.

```

y=[0; 1; 0.9; 0.71; 0.749; 0.7831; 0.7719];
u=ones(length(y),1);
T=0.1; %sampling time
% Create a data object using the measured values of the input
% vector u and the output vector y
data = iddata(y,u,T);
% Define constants for expected number of poles and zeros
na=2; % Expected number of poles
nb=1; % Expected number of zeros
nk=1; % Expected input time delay
% Estimate the transfer function coefficients using
m=arx(data, [na,nb,nk]);
m.a
m.b
    
```

Here, the integers  $na$ ,  $nb$ , and  $nk$  define the number of poles, zeros, and input delay of the transfer function, reliable values for which must be known *a priori*. Typically, one may choose  $na$  to be equal to the expected number of poles,  $nk$  to be the expected input delay, and  $nb \leq na - nk + 1$ . Variable  $m$  of *arx* is a MATLAB object, and returns two different vectors, namely  $m.a$  for the

denominator polynomial of the transfer function, and  $m.b$  for the numerator polynomial, both in negative powers of  $z$ . The estimated transfer function is given by:

$$\frac{Y(z)}{U(z)} = \frac{m.b(1) + m.b(2)z^{-1} + \dots + m.b(nb + nk)z^{-nk-nb+1}}{m.a(1) + m.a(2)z^{-1} + \dots + m.a(na + 1)z^{-na}}$$

with the first coefficient of  $m.a$  equal to 1, and the first  $nk$  coefficients of  $m.b$  equal to 0. For this example, the estimated transfer function using the  $arx$  function, with  $na = 2$ ,  $nb = 1$ , and  $nk = 1$ , is obtained as

m.a=	1.00	0.1	0.2
m.b=	0.0	1.0	

This means that the identified transfer function is

$$G(z) = \frac{z^{-1}}{1 + 0.1z^{-1} + 0.2z^{-2}} = \frac{z}{z^2 + 0.1z + 0.2}$$

which is exactly equal to the actual transfer function from which  $y(k)$  was generated.

## 10.6 ESTIMATING TRANSFER FUNCTIONS WITH PARTLY KNOWN POLES AND ZEROS

Depending on circumstances, one or more poles or zeros of the transfer function of a system may be known [5]. For example, one may know *a priori* that the process has an integrator, which corresponds to a continuous-time pole at  $s = 0$ , and thereby a discrete-time pole at  $z = 1$ . Similarly, one may know that the system under study has a differentiator, which corresponds to a continuous-time zero at  $s = 0$ , and thereby a discrete-time zero at  $z = 1$ . In that case, one need not waste time and computational efforts to estimate these poles and zeros that are already known, and instead apply a slightly modified form of (10-43) to estimate only the remaining part of the transfer function that is unknown. This can be done simply as follows. Suppose that it is already known that a system has a pole at  $z = \lambda_1$  and a zero at  $z = \lambda_2$ . That means the transfer function of the system is

$$G(z) = \frac{z - \lambda_2}{z - \lambda_1} \bar{G}(z)$$

where  $\bar{G}(z)$  is unknown. Thus one may write

$$\bar{G}(z) = \frac{\bar{Y}(z)}{U(z)}$$

where  $\bar{Y}(z) = (z - \lambda_1)Y(z)$ , and  $\bar{U}(z) = (z - \lambda_2)U(z)$ . In time-domain this means that

$$\bar{y}(k) = y(k + 1) - \lambda_1 y(k)$$

$$\bar{u}(k) = u(k + 1) - \lambda_2 u(k)$$

Therefore, given the values of  $u(k)$  and  $y(k)$  for different  $k$ , and the values of  $\lambda_1$  and  $\lambda_2$ , one may first generate the corresponding values of  $\bar{u}(k)$  and  $\bar{y}(k)$ , and then use these values to identify

$\bar{G}(z)$  via the least-squares method. Once  $\bar{G}(z)$  is identified, the actual transfer function can be constructed by multiplying  $\bar{G}(z)$  with  $(z - \lambda_2)/(z - \lambda_1)$ .

### EXAMPLE 10.5

Consider the input and output data of an LTI system:

$k$	$u(k)$	$y(k)$
0	1	0
1	1	1
2	1	1.6
3	1	1.91
4	1	2.066

It is given that the system has one zero and two poles. One of the poles is at  $z = 0.5$ . We wish to generate the least-squares estimate of the transfer function from the given input–output data. The transfer function can be written as

$$G(z) = \frac{1}{z - 0.5} \bar{G}(z) = \frac{z^{-1}}{1 - 0.5z^{-1}} \bar{G}(z)$$

which means that we first need to generate  $\bar{y}(k)$  using  $\lambda_1 = 0.5$ . This will naturally reduce the total number of output samples for  $\bar{y}(k)$  to 4 ( $k = 0, 1, 2, 3$ ) and introduces one input delay. To identify  $\bar{G}(z)$ , the corresponding input values need to be used. For this one may write the following MATLAB code, and compute  $\bar{G}(z)$  using the function *arx*.

```

y=[0; 1; 1.6; 1.91; 2.066];
u=[1; 1; 1; 1; 1];
p=length(y); T=1;
lambda=0.5;
for k=1:1:p-1
ybar(k)=y(k+1)-lambda*y(k);
end;
ubar=u(1:p-1);
data=iddata(ybar',ubar,T);
na=1; % The number of remaining poles
nb=1; % The number of remaining zeroes
nk=0; % The remaining input delay (zero in this case)
m=arx(data,[na,nb,nk]);
m.a
m.b
    
```

The estimates of the numerator and denominator of  $m$  are obtained as

```

m.a= 1.0 -0.1
m.b= 1.0
    
```

This means that the identified transfer function  $\bar{G}(z)$  is

$$\bar{G}(z) = \frac{1}{1 - 0.1z^{-1}} = \frac{z}{z - 0.1}$$



Therefore, the estimate for  $G(z)$  is

$$G(z) = \frac{z}{(z - 0.5)(z - 0.1)}$$

The actual transfer function from which the output data was generated is exactly the same as that estimated.

## 10.7 RECURSIVE LEAST-SQUARES SYSTEM IDENTIFICATION

If all measurements are accumulated and the coefficient vector  $\theta$  is calculated in (10-47), the process is said to be *batch least squares*. However, (10-47) can be manipulated into a difference equation such that the  $\theta$  vector can be recalculated with each data sample as it arrives; the method is then said to be *recursive least squares*. In this section we derive the defining equations for recursive least-squares identification.

From (10-35) and (10-36),

$$\begin{aligned} y(k) &= \mathbf{f}^T(k)\theta + \mathbf{e}(k) \\ &= [y(k-1) y(k-2) \dots u(k-1) \dots]\theta + e(k) \end{aligned}$$

Now, from (10-37),

$$\mathbf{F}^T(N+1) = [\mathbf{f}(n) \dots \mathbf{f}(N) \mathbf{f}(N+1)] \quad (10-49)$$

We will let the weighting factor be

$$w(k) = a\gamma^{N+1-k}, \quad \gamma \leq 1 \quad (10-50)$$

In (10-47) we manipulate the term

$$\begin{aligned} \mathbf{F}^T(N+1)\mathbf{W}(N+1)\mathbf{F}(N+1) &= \sum_{k=n}^{N+1} \mathbf{f}(k) a \gamma^{N+1-k} \mathbf{f}^T(k) \\ &= \sum_{k=n}^{N+1} \mathbf{f}(k) a \gamma \gamma^{N-k} \mathbf{f}^T(k) + \mathbf{f}(N+1) a \mathbf{f}^T(N+1) \\ &= \gamma \mathbf{F}^T(N)\mathbf{W}(N)\mathbf{F}(N) + \mathbf{f}(N+1) a \mathbf{f}^T(N+1) \end{aligned} \quad (10-51)$$

This relatively simple form results from  $\mathbf{W}(N)$  being diagonal. For convenience the  $2n \times 2n$  matrix  $\mathbf{P}(k)$  is defined as

$$\mathbf{P}(k) = [\mathbf{F}^T(k)\mathbf{W}(k)\mathbf{F}(k)]^{-1} \quad (10-52)$$

Then, from (10-51) and (10-52),

$$\mathbf{P}^{-1}(N+1) = \gamma \mathbf{P}^{-1}(N) + \mathbf{f}(N+1) a \mathbf{f}^T(N+1) \quad (10-53)$$

At this step we recall the matrix inversion lemma, which states the following identity:

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1} \quad (10-54)$$

We let the right side of (10-53) be  $(\mathbf{A} + \mathbf{BCD})$  with the following assignments:

$$\begin{aligned} \mathbf{A} &= \gamma\mathbf{P}^{-1}(N), & \mathbf{B} &= \mathbf{f}(N + 1), \\ \mathbf{C} &= a, & \mathbf{D} &= \mathbf{f}^T(N + 1) \end{aligned} \quad (10-55)$$

The matrix inverse of (10-52) can then be expressed as

$$\begin{aligned} \mathbf{P}(N + 1) &= \frac{1}{\gamma}\mathbf{P}(N) - \frac{1}{\gamma}\mathbf{P}(N)\mathbf{f}(N + 1) \\ &\quad \times \left[ \frac{1}{a} + \mathbf{f}^T(N + 1)\frac{1}{\gamma}\mathbf{P}(N)\mathbf{f}(N + 1) \right]^{-1} \mathbf{f}^T(N + 1)\frac{1}{\gamma}\mathbf{P}(N) \end{aligned} \quad (10-56)$$

Since this equation is valid for any  $N \geq n$ , we have developed a difference equation for  $\mathbf{P}(k)$ . Note that the factor in the inverse in (10-47) is scalar, and thus no matrix inversion is required.

The second factor in the batch equation for  $\theta$ , (10-47), can be manipulated as

$$\begin{aligned} \mathbf{F}^T\mathbf{W}\mathbf{y} &= [\mathbf{f}(n) \cdots \mathbf{f}(N + 1)] \begin{bmatrix} a\gamma^{N+1-n} & & & \\ & \ddots & & \\ & & a\gamma & \\ & & & a \end{bmatrix} \begin{bmatrix} y(n) \\ \vdots \\ y(N) \\ y(N + 1) \end{bmatrix} \\ &= \gamma\mathbf{F}^T(N)\mathbf{W}(N)\mathbf{y}(N) + \mathbf{f}(N + 1)a y(N + 1) \end{aligned} \quad (10-57)$$

where each matrix on the left side is defined at the sampling instant  $N + 1$ , and only the nonzero elements of  $\mathbf{W}(N + 1)$  are shown.

The substitution of (10-52), (10-56), and (10-57) into the weighted least-squares solution (10-47) yields the following set of difference equations after some manipulations:

$$\mathbf{P}(N) = [\mathbf{F}^T(N)\mathbf{W}(N)\mathbf{F}(N)]^{-1} \quad (10-58)$$

$$\mathbf{L}(N + 1) = \frac{1}{\gamma}\mathbf{P}(N)\mathbf{f}(N + 1) \left[ \frac{1}{a} + \mathbf{f}^T(N + 1)\frac{1}{\gamma}\mathbf{P}(N)\mathbf{f}(N + 1) \right]^{-1} \quad (10-59)$$

$$\hat{\boldsymbol{\theta}}_{WLS}(N + 1) = \hat{\boldsymbol{\theta}}_{WLS}(N) + \mathbf{L}(N + 1)[y(N + 1) - \mathbf{f}^T(N + 1)\hat{\boldsymbol{\theta}}_{WLS}(N)] \quad (10-60)$$

$$\mathbf{P}(N + 1) = \frac{1}{\gamma}[\mathbf{I} - \mathbf{L}(N + 1)\mathbf{f}^T(N + 1)]\mathbf{P}(N) \quad (10-61)$$

These equations apply for  $N \geq n$ ; hence, if  $N$  is replaced with  $k$ , the difference equations form the recursive solution for the weighted least-squares system identification. The solution is started by finding  $\mathbf{P}(k)$  and  $\hat{\boldsymbol{\theta}}(k)$  for some values of  $k$ . Next the vector  $\mathbf{f}(k + 1)$  is formed, where

$$\mathbf{f}(k + 1) = [y(k) y(k - 1) \cdots y(k - n) u(k) \cdots u(k - n)]^T$$

The next set of measurements are used to form  $y(k + 1)$ , and Equations (10-59), (10-60), and (10-61) are then solved recursively. The initial solution for  $\mathbf{P}(k)$  from (10-58) requires a matrix inversion. For that reason, the initial values of  $\mathbf{P}(k)$  and  $\hat{\boldsymbol{\theta}}(k)$  are estimated (not calculated) in some computer programs, and several iterations are required before the estimated coefficient vector settles to approximately constant values. Although recursive least squares is much faster compared with batch least squares for most problems, one disadvantage of the recursive approach is that it requires an initial guess for  $\boldsymbol{\theta}$ . If this initial guess is far from the true value, then it may take a large number of iterations to converge.

Recall that the least-squares identification only gives a “best” fit to the data. However, depending on the plant model, the noise level of the measurements, the sampling period, and several other practical factors, this “best” fit may turn out to be too inaccurate and thus useless. For example, suppose that for a given physical system, accurate modeling requires a second-order linear model. If we try to fit the data to a first-order model, the least-squares procedure will yield a “best” first-order transfer function whose output matches the measured output in the least-squares sense. However, the transfer function will not accurately model the actual physical system. Therefore, additional verification is often required once a transfer function is calculated using least squares. One such verification is the comparison of the time responses of the physical system and the derived model for different sets of inputs. Also, the frequency response of the physical system can be measured and compared with the frequency response of the derived model.

An additional problem with any least-squares procedure is that if one data pair is greatly in error, the square of this large error will tend to dominate the sum-of-squared errors. Thus the procedure will be heavily weighted to reduce this large error, resulting in a less accurate model. For this reason it is generally recommended that any data pairs that obviously do not match the remaining data be ignored in some consistent manner. An example of recursive least-squares identification will now be given.

### EXAMPLE 10.6

The data of Example 10.3 will be used to illustrate the ordinary least-squares recursive procedure. No weighting of data is employed, and  $a = \gamma = 1$  in (10-58)–(10-61). Suppose that an additional data point is taken, with  $y(3) = 0.15$  and  $u(3) = 0.40$ . We use the solution in Example 10.2 to start the recursive solution; hence  $n = 1$  and  $N = 2$ . Thus, from (10-58) and Example 10.2,

$$\mathbf{P}(N) = \mathbf{P}(2) = [\mathbf{F}^T(2)\mathbf{F}(2)]^{-1} = \begin{bmatrix} 17.361 & -2.5 \\ -2.5 & 1 \end{bmatrix}$$

and

$$\hat{\boldsymbol{\theta}}(N) = \hat{\boldsymbol{\theta}}(2) = \begin{bmatrix} 0 \\ 0.3 \end{bmatrix}$$

From the data in Example 10.2,

$$\begin{aligned} \mathbf{f}^T(3)\mathbf{P}(2)\mathbf{f}(3) &= [0.225 \quad 0.5] \begin{bmatrix} 17.361 & -2.5 \\ -2.5 & 1 \end{bmatrix} \begin{bmatrix} 0.225 \\ 0.5 \end{bmatrix} \\ &= [2.656 \quad -0.0625] \begin{bmatrix} 0.225 \\ 0.5 \end{bmatrix} = 0.5664 \end{aligned}$$

and

$$\begin{aligned} \mathbf{L}(3) &= \mathbf{P}(2)\mathbf{f}(3)[1 + \mathbf{f}^T(3)\mathbf{P}(2)\mathbf{f}(3)]^{-1} \\ &= \begin{bmatrix} 17.361 & -2.5 \\ -2.5 & 1 \end{bmatrix} \begin{bmatrix} 0.225 \\ 0.5 \end{bmatrix} [1/0.5664] = \begin{bmatrix} 4.690 \\ -0.110 \end{bmatrix} \end{aligned}$$

From (10-60), the estimated coefficient vector, after four data points, is then

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{WLS}(3) &= \hat{\boldsymbol{\theta}}_{WLS}(2) + \mathbf{L}(3)[y(3) - \mathbf{f}^T(3)\hat{\boldsymbol{\theta}}_{WLS}(2)] \\ &= \begin{bmatrix} 0 \\ 0.3 \end{bmatrix} + \begin{bmatrix} 4.690 \\ -0.110 \end{bmatrix} \left[ 0.15 - \begin{bmatrix} 0.225 & 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ 0.3 \end{bmatrix} \right] \\ &= \begin{bmatrix} 0 \\ 0.3 \end{bmatrix} \end{aligned}$$

Note that since the data is exact in this case, no correction in the coefficient vector results.

## 10.8 PRACTICAL FACTORS FOR IDENTIFICATION

Quite naturally, the accuracy of any system identification method, whether in time-domain using black-box algorithms or in frequency-domain using parametric least-squares estimation, depends on several practical factors. The three most important factors among them are:

1. Choice of excitation input  $u(k)$
2. Choice of sampling frequency
3. Proper scaling of input and output signals

We next briefly explain how these three factors decide the accuracy of identified models as compared to the actual real-world physical model of a system.

### Choice of Input

As explained in [5], the quality of both black-box identification and least-squares depends significantly on the input  $u(k)$ . In order to guarantee a reasonably good estimate of the actual model, the system must be excited by a sufficiently “rich” input, so that all the underlying physical properties of the system get reflected in the output. An easy way to think of this is as follows. Say, we take a glass of water and mix some sand in it. Then we take a fraction of the volume of that water, and try to estimate the amount of sand poured. If we don’t shake the glass well (equivalent to exciting the system with a persistent input), then all the sand will settle down at the bottom of the glass, and the sample may not even have any sand left to give us a proper indication of how much sand was poured into the glass. The choice of the input, therefore, plays a critical role in determining whether identification will lead to a reliable model or not. At the least, the input must have the following properties:

1. The input should be *sufficiently rich* or persistently exciting, as indicated in Assumption 2 in Section 10.4. In other words, the input must be dominant enough to guarantee that the matrix  $\mathbf{F}^T(N)\mathbf{F}(N)$  is well-conditioned and invertible to satisfy (10-43). For example, unimodal sinusoids should be avoided as inputs because a sinusoid of frequency  $\lambda$  will not allow us to distinguish between different transfer functions with exactly the same value at the point  $z = e^{j\lambda}$ . Good

inputs include square-waves, the sum of many sinusoids, or a chirp signal (i.e., a signal with time-varying frequency), impulsive inputs, and the like.

**2.** The input must be strong enough to produce an output that has high signal-to-noise ratio (SNR) [7]. In particular, large inputs are needed when one wants to probe the system at frequencies for which the noise is large. However, when the process is nonlinear but is being approximated by a linear model, large inputs may be problematic because the linearization may not be valid. As shown in [5], there is usually an “optimal” input level that needs to be determined by trial-and-error. As an alternative, nonlinear filtering and stochastic parameter estimation using Cramer–Rao bounds may also be used [7]. Those techniques, however, are beyond the scope of this chapter.

**3.** Since most engineering systems operate in a closed loop, that is, under the influence of some feedback controller, the excitation input should be *representative* of the class of inputs that are expected to appear in the feedback loop. In other words, the input should have strong components in the range of frequencies at which the system will operate.

**4.** An interesting point to note in this context is that in many engineered systems, especially in safety-critical systems such as power systems, aircrafts, robotic vehicles, etc., the user may not have complete freedom of choosing an excitation input for identifying the system model. For example, one may wish to identify the dynamic model of the power grid encompassing the entire North America, but for obvious reasons of safety and security, one will never have complete freedom to inject large input signals into the grid lest that brings down the regular normal operations of the system. In such cases, the user often has to depend on natural disturbance inputs, such as faults and failures, and implicitly assume their impacts on the measured output to be strong enough for successful identification. Another obstacle in such cases is that it may not always be possible to model the fault precisely in the form of a regular mathematical function, such as step, square wave, or impulse. Therefore, a popular technique that is often used is to identify such systems through “output-only” identification methods, where the estimated model parameters are completely decoupled from the input information [8].

### Choice of Sampling Frequency

Choosing the right sampling frequency is equally important, especially in terms of quantization errors. If the sampling is too fast, then the output values are too close to each other and their difference may, in fact, become smaller than one quantization level of the analog-to-digital converter (ADC). As shown in [5], the pattern of the output values that appear in solving the least-squares estimate (10-43) is mostly determined by the dynamics of the ADC, while the least-squares model may have very small amount of information about the system dynamics.

When the sampling interval is increased, the difference between the consecutive output values increases and eventually dominates the quantization noise. In practice, one should choose a sampling rate that is large enough so that the Tustin transformation is valid but sufficiently small so that the quantization noise does not dominate. A good rule of thumb is to sample, if practical, the system at a frequency 20–40 times larger than the desired closed-loop bandwidth for the system [5].

### Choice of Signal Scaling

Scaling the input  $u(k)$  and output  $y(k)$  properly is another deciding factor for numerical accuracy of an identification algorithm, especially for the least-squares method. For example, in some systems the magnitude of the measured output may be unusually large or unusually small depending

on the units in which they are being measured. Such large or small values may affect the well-posedness and the numerical stability of the matrix inverse operation in (10-43). If very small values are used, then any typical software program will approximate them by eliminating figures after decimal points, which may reduce the information reflected about the system behavior on the output. To maintain compatibility of units between the input and output data, it is, therefore, best to normalize the values of  $u(k)$  and output  $y(k)$  by dividing or multiplying their actual values by the same normalization constant. One can, for example, choose the normalization constant such that all input and output measurements lie in the same range, for example, in the interval  $[-1, 1]$ .

## 10.9 SUMMARY

In this chapter we presented some basic results on discrete-time system identification using input and output measurements. Black-box identification algorithms were presented in time-domain for state-variable models whose order and modeling matrices are completely unknown. Next, least-squares system identification was developed for models whose order and structure may be known to some extent but the model parameters are all unknown. Implementation of the least-squares algorithm in both batch and recursive ways were described. All of these identification methods are well suited to computer implementation, and can be very effective tools for creating data-based models for prediction, state estimation, and control.

## References and Further Readings

- [1] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ: Prentice Hall, 1999.
- [2] S. Soderstrom and P. G. Stoica, *Instrumental Variable Methods for System Identification*. Berlin: Springer, 1983.
- [3] P. Van Overschee and B. D. Moore, *Subspace Identification for Linear Systems*. Boston: Kluwer Academic Publishers, 1996.
- [4] T. E. Fortman, "A Matrix Inversion Identity," *IEEE Trans. Autom. Control*, AC-15, p. 599, 1970.
- [5] J. Hespanha, *Lectures Notes on Topics of Undergraduate Control Systems Design*, University of California Santa Barbara, 2012.
- [6] S. J. Qin, "An Overview of Subspace Identification," *Computers and Chemical Engineering*, 30, pp. 1502–1513, 2006.
- [7] H. Vincent Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer, 1994.
- [8] R. Brincker, L. Zhang, and P. Andersen, "Modal Identification of Output-only Systems Using Frequency Domain Decomposition," *Smart Materials and Structures*, 10, pp. 4441–4445, 2001.

## Problems

10.2-1. Suppose that a square-law circuit has the input–output relationship

$$y = kx^2$$

where  $x$  is the input voltage and  $y$  is the output voltage.

- (a) Derive a least-squares procedure for calculating  $k$ , similar to the procedure developed in Section 10.2.
- (b) Experimentation with the circuit yields the following data pairs:

$k$	$x$	$Y$
1	0	0.01
2	1.0	1.01
3	2.0	3.98

Find the least-squares value for  $k$  for these data.

- (c) To illustrate the effects of erroneous data, suppose that an additional data point,  $x = 1.5$  and  $y = 3.30$ , is taken. From part (b) we know that for  $x = 1.5$ ,  $y$  should be approximately 2.25. Hence the additional data point is erroneous, assuming that the data in part (b) are accurate. Find the percent error in  $\hat{k}$  using the four data pairs, as compared to the value calculated in part (b).
- 10.2-2.** Equation (10-5) gives a least-squares estimate for curve fitting. This equation was derived by finding the point at which the slope of the cost function is zero. The maximum of a function also occurs at the point at which the slope is zero. Show that (10-5) is a minimum, not a maximum, of the cost function.
- 10.2-3.** Consider the sinusoidal function  $y(k) = \sin(10k)$  for  $k = 0, 1, \dots, 9$ . Find the minimum degree of the polynomial which fits this function with a maximum least-squares error of  $10^{-3}$ .
- 10.2-4.** In Equation (10-7), show that if  $N < n$  then the solution of  $\mathbf{b}$  is non-unique. For the example in Problem 10.2-3, calculate the error bound when the polynomial order  $n$  is exactly equal to the number of samples  $N = 10$ .
- 10.4-1.** Using the data given in Problem 10.5-1, derive the state-variable model using the algorithm described in Section 10.4.
- 10.4-2.** Consider a variant of the matrix  $\mathbf{A}$  used in Example 10-2. In this example, we used  $\mathbf{A}(2,1) = -0.6$ .
- (a) Changing this to  $\mathbf{A}(2,1) = -0.9$ , and assuming exact same values for  $\mathbf{B}$  and  $\mathbf{C}$  as in the example, what changes in the impulse response do you observe and why?
- (b) In Example 10-2, we used 50 data samples to estimate the system matrices such that the estimated impulse response matches the measured output perfectly. Repeat the identification with  $\mathbf{A}(2,1) = -0.9$ , and check if we should use lesser or more number of samples to achieve the same accuracy in matching.
- 10.4-3.** Derive the SISO black-box identification algorithm described in Section 10.4 when the excitation input  $u(k)$  is a unit step signal.
- 10.4-4.** Consider the SISO continuous-time model:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \mathbf{C} = [0 \quad 1]$$

Assume  $u(k)$  to be an impulse input. Sample this system with  $T = 10$  s, 1 s, and 0.1 s, and compare how accurately the estimated impulse responses, computed using the black-box identification method of Section 10.4, for each case match the measured response  $y(k)$ , where  $k = 0, 1, 2, \dots, 50$ .

- 10.4-5.** Consider the following three data sets, which are the same data set given in Example 10.2 but corrupted with varying levels of white Gaussian noise. Three different SNRs are used to generate the noisy output, namely 1 dB, 10 dB, and 50 dB. It can be seen that as the SNR is increased the output values approach those given in the example. Assuming the input to be a unit impulse, repeat the identification for each of the three noisy cases, and compute how the poles of the identified transfer function differ from the actual

poles. Also, plot the percentage error in the estimated poles versus the actual poles against the increasing levels of SNR.

1. SNR = 1 dB

k	0	1	2	3	4	5	6	7	8
$y(k)$	0.979	0.7483	0.3272	-1.8435	-0.2816	-0.7421	-1.4067	0.4538	0.2511

2. SNR = 10 dB

K	0	1	2	3	4	5	6	7	8
$y(k)$	0.0106	0.5743	0.0585	0.0959	-0.0608	-0.0009	-0.084	-0.5524	-0.0905

3. SNR = 50 dB

k	0	1	2	3	4	5	6	7	8
$y(k)$	-0.0026	0.9929	-0.3017	-0.0165	0.0275	-0.0051	0.0005	0.001	-0.0003

10.5-1. (a) A first-order system yields the following input–output measurements:

k	Input	Output
0	10	0
1	10	12.2
2	10	20.1

Find the system transfer function by the least-squares batch procedure, using all the data.

(b) An additional data pair for  $k = 3$  is measured, with the input equal to zero and the output equal to 31.8. Start the recursive least-squares procedure using the results of part (a), and calculate  $\hat{\theta}$  using the additional data pair.

10.5-2. For a third-order discrete system, suppose that the data pairs  $[u(k), y(k)]$ ,  $k = 0, 1, \dots, 5$ , are available. In these data,  $u(k)$  is the input and  $y(k)$  is the output. Write the complete expression for  $\mathbf{F}^T(N)\mathbf{y}(N)$  as in (10-43), using all the data.

10.5-3. Derive (10-47), the weighted least-squares estimate.

10.5-4. Derive the transfer function for the LTI model in Examples 10-3 and 10-4 using weighted least squares. Manipulate the weights and test whether the transfer function can be identified with lesser number of measurements used in the examples. You may use a maximum tolerance of  $10^{-3}$  for the estimation error between the identified transfer function coefficients and their actual values.

10.6-1. Consider that  $b_1, b_2, \dots, b_m$  in (10-30) for some given positive integer  $m$  are all known constants. Using the method proposed in Section 10.6 re-derive (10.43) to compute the least-squares estimate of only the unknown coefficients of (10-30).

10.6-2. Consider the data set given in Example 10.2. Assume that the system has three poles and two zeros. One of the poles is at  $z = 0.1$  and one of the zeros is at  $z = 0.10002$ . Identify the system transfer function. Compare the estimated values of the two remaining poles with those given in Example 10.2.

10.6-3. Consider the following unit step response of a discrete-time LTI system:

k	0	1	2	3	4	5	6	7	8
$y(k)$	0	4	-8	16	-24	36	-48	64	-80



It is known that the transfer function of the system has three poles and two zeros. All three poles are at  $z = 1$ . Assuming the sampling time  $T = 0.1$  s, derive a least-squares estimate of the transfer function using this prior information.

- 10.7-1.** Using (10-47), (10-52), (10-56), and (10-57), derive the equations for the recursive least-squares estimation, (10-59)–(10-61).
- 10.7-2.** Use recursive least squares to estimate the transfer function for the LTI model given in Example 10-2 using the impulse response data  $y(k)$ . Verify that the poles of this transfer function match the eigenvalues of the system matrix identified in Example 10.2.
- 10.7-3.** Using the MATLAB function *awgn*, repeat the recursive least-squares estimation of Example 10.4 when the output is corrupted by a white Gaussian noise with 10.0 dB SNR. Compare the estimation results with those for 1.0 dB SNR.
- 10.8-1.** Consider the values of  $y(k)$  in Problem 10.6-3 at  $k = 0, 2, 4, 6,$  and  $8$ . Using these values and assuming  $T = 0.2$  s, repeat the least-squares estimation of the transfer function. Next, consider the values at  $k = 0, 4,$  and  $8,$  and assuming  $T = 0.4$  s repeat the estimation one more time. Compare both of these estimates with that obtained in Problem 10.6-3.
- 10.8-2.** Consider the transfer function

$$G(z) = \frac{z}{z^2 + 0.3z + 0.1}$$

Excite this transfer function with the following inputs, and collect the respective outputs in a vector  $y(k)$ ,  $k = 0, 1, \dots, 20$ . Assume the sampling time  $T = 0.1$  s.

- (a)  $u(k) = \sin(10k)$   
 (b)  $u(k) = 100 \sin(10k)$   
 (c)  $u(k) = \sin(10k) + \sin(20k)$   
 (d)  $u(k) = \sin(k) + \sin(10k) + \sin(100k)$   
 (e)  $u(k)$  is a unit square wave stretching from  $k = 0$  to  $k = 2$   
 (f)  $u(k)$  is a unit square wave stretching from  $k = 0$  to  $k = 10$   
 (g)  $u(k)$  is a unit step

Using  $y(k)$  and  $u(k)$  for  $k = 0, 1, \dots, 20$ , derive the transfer function of the system using least-squares method. Compare each solution to the actual transfer function  $G(z)$ .

# 11

## Linear Quadratic Optimal Control

### 11.1 INTRODUCTION

In Chapter 8 we presented design techniques, such as Bode plots and Nyquist plots, that are termed classical, or traditional. The two techniques developed are the frequency-response techniques and the root-locus technique. Both methods are very effective, but are largely trial and error, with experience being very useful. Even when an acceptable design is completed, the question remains as to whether a “better” design could be found. The pole-assignment design technique of Chapter 9 is termed a modern technique, and is based on the state-variable model of the plant, rather than on the transfer function as required by the classical methods. In this procedure we assumed that we know the exact locations required for the closed-loop transfer-function poles, and we can realize these locations, at least in the linear model. Of course, for the physical system, the regions in which the pole locations can be placed are limited.

In the pole-assignment technique, we assume that we know the pole locations that yield the “best” control system. In this chapter we develop a different technique that yields the “best” control system. This technique is an optimal design procedure, and assumes that we can write a mathematical function which is called the *cost function*. The optimal design procedure minimizes this cost function, hence the term *optimal*. However, in most cases the choice of the cost function involves some trial and error; that is, we are not sure of the exact form that the cost function should take.

For discrete systems, the cost function (also called the *performance index*) is generally the form

$$J_N = \sum_{k=0}^N L[\mathbf{y}(k), \mathbf{r}(k), \mathbf{u}(k)] \quad (11-1)$$

In this relationship  $k$  is the sample instant and  $N$  is the terminal sample instant. The control system outputs are  $\mathbf{y}(k)$ , the reference inputs are  $\mathbf{r}(k)$ , and  $\mathbf{u}(k)$  are the control inputs to the plant.

For a physical system, the control inputs are always constrained. For example, the amplitude of each component of the control vector may be limited, such that

$$|u_i(k)| \leq U_i$$

where each  $U_i$  is a given constant, and the subscript  $i$  denotes the  $i^{\text{th}}$  input. For the case of limited control energy, we have

$$u_i^2(k) \leq M_i$$

where each  $M_i$  is a given constant. The availability of finite control energy may be represented by a term in the cost function (11-1) as

$$\sum_{k=0}^N \mathbf{u}^T(k) \mathbf{R}(k) \mathbf{u}(k)$$

where  $\mathbf{R}(k)$  is a weighting matrix and is not related to  $\mathbf{r}(k)$ . This function is called a *quadratic form*, and will be considered in detail in the following sections. In any case, the control must satisfy certain constraints; any control that satisfies these constraints is called an *admissible control*.

In this chapter, the design technique for optimal linear regulator control systems with quadratic cost functions is developed. This design results in a control law of the form

$$\mathbf{u}(k) = -\mathbf{K}(k)\mathbf{x}(k) \quad (11-2)$$

Hence we have linear, time-varying, full-state feedback. The same limitations of full-state feedback apply here as in Chapter 9. It will generally be necessary to utilize an observer to implement the control law. The observer may be designed by the techniques presented in Chapter 9.

This chapter requires a background in the mathematics of matrices. Appendix IV presents a brief review of the required matrix mathematical definitions and manipulations.

## 11.2 THE QUADRATIC COST FUNCTION

We begin the development of optimal control design by considering the case of a quadratic cost function of the states and the control signals, that is,

$$J_N = \sum_{k=0}^N \mathbf{x}^T(k) \mathbf{Q}(k) \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{R}(k) \mathbf{u}(k) \quad (11-3)$$

where  $N$  is finite and  $\mathbf{Q}(k)$  and  $\mathbf{R}(k)$  are symmetric. The linear plant is described by

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}(k)\mathbf{x}(k) \end{aligned} \quad (11-4)$$

Note that both the plant and the cost-function matrices are allowed to be *time varying*. Recall that the design procedures developed earlier require linear *time-invariant* plants.

The quadratic cost function is considered because the development is simple and the cost function is logical. For example, consider a second-order single-output system with

$$y(k) = \mathbf{C}\mathbf{x}(k) = [c_1 \quad c_2]\mathbf{x}(k)$$

Suppose that we want to drive the output to zero, and hence we will choose the cost function to contain  $y^2(k)$ . Then

$$y^2(k) = y^T(k)y(k) = \mathbf{x}^T(k)\mathbf{C}^T\mathbf{C}\mathbf{x}(k) = \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k)$$

where  $\mathbf{Q} = \mathbf{C}^T\mathbf{C}$  is square. Hence we see for this case that the quadratic function appears naturally.

Consider the quadratic function

$$\begin{aligned} F &= \mathbf{x}^T\mathbf{Q}\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= q_{11}x_1^2 + (q_{12} + q_{21})x_1x_2 + q_{22}x_2^2 \end{aligned} \quad (11-5)$$

Note first that  $F$  is a scalar and that there is no loss of generality in assuming  $\mathbf{Q}$  to be symmetric. Now, if the quadratic form in (11-5) is positive semidefinite [1], which we will require, then

$$\begin{aligned} F &\geq 0, & \mathbf{x} &\neq 0 \\ F &= 0, & \mathbf{x} &= 0 \end{aligned}$$

Then, in general, minimizing  $F$  will minimize the magnitude of states that contribute to  $F$ , in some sense. For example, if

$$F = x_1^2 + x_2^2 = \mathbf{x}^T\mathbf{Q}\mathbf{x} = \mathbf{x}^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}$$

then minimizing  $F$  will tend to minimize the magnitudes of  $x_1$  and  $x_2$ . However, if

$$F = 100x_1^2 + x_2^2 = \mathbf{x}^T \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}$$

then minimizing  $F$  will also minimize  $|x_1|$  and  $|x_2|$ , but  $|x_1|$  should be much smaller than  $|x_2|$ . As a third case, suppose that

$$F = x_1^2 = \mathbf{x}^T \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{x}$$

(i.e.,  $F$  is not a function of  $x_2$ ). Then in minimizing  $F$  only  $x_1$  is minimized, and  $x_2$  is determined by its relationship to  $x_1$ .

Consider now the contribution of the control input  $\mathbf{u}(k)$ . Suppose that

$$\begin{aligned} G &= \mathbf{u}^T\mathbf{R}\mathbf{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ &= r_{11}u_1^2 + (r_{12} + r_{21})u_1u_2 + r_{22}u_2^2 \end{aligned} \quad (11-6)$$

Note again that there is no loss in generality in assuming the matrix in a quadratic form is symmetric. If  $G$  in (11-6) is positive definite [1], which we will require, then

$$\begin{aligned} G &> 0, & \mathbf{u} &\neq 0 \\ G &= 0, & \mathbf{u} &= 0 \end{aligned}$$

Thus if we minimize  $G$ , we are minimizing the control functions. If  $G$  were allowed to be only positive semidefinite, some components of the control vector could be quite large when  $G$  is minimized.

A mathematical test for positive definiteness is as follows. A square matrix is positive definite if and only if all of its eigenvalues are real and positive. If the eigenvalues are real and positive, except for some that have a value of zero, the matrix is positive semidefinite [1].

Consider the total cost function of (11-3):

$$[eq. (11-3)] \quad J_N = \sum_{k=0}^N \mathbf{x}^T(k)\mathbf{Q}(k)\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}(k)\mathbf{u}(k)$$

If  $\mathbf{Q}(k)$  were positive semidefinite and  $\mathbf{R}(k)$  were the null matrix, minimization of (11-3) would force the  $\mathbf{x}(k)$  vector toward zero very quickly, which in general would require a large  $\mathbf{u}(k)$ . For a physical system,  $\mathbf{u}(k)$  is always bounded, and in general the large  $\mathbf{u}(k)$  would not be realizable. Hence the positive definite matrix  $\mathbf{R}(k)$  is added to the cost function to limit  $\mathbf{u}(k)$  to realizable values. Then, in using the cost function (11-3), we include the term involving  $\mathbf{x}(k)$ , so that, in some sense, the magnitudes of the states are driven toward zero. The term involving  $\mathbf{u}(k)$  is included in order that the components of the control vector will be limited in magnitude such that the design is physically realizable.

### 11.3 THE PRINCIPLE OF OPTIMALITY

The optimal control design problem posed in Section 11.2 may be stated as follows:

For a linear discrete plant described by

$$[eq. (11-4)] \quad \begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}(k)\mathbf{x}(k) \end{aligned}$$

determine the control (11-7)

$$\mathbf{u}^o(k) = \mathbf{f}[\mathbf{x}(k)] \quad (11-7)$$

that minimizes the quadratic cost function

$$[eq. (11-3)] \quad J_N = \sum_{k=0}^N \mathbf{x}^T(k)\mathbf{Q}(k)\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}(k)\mathbf{u}(k)$$

where  $N$  is finite,  $\mathbf{Q}(k)$  is positive semidefinite, and  $\mathbf{R}(k)$  is positive definite. In (11-7), the superscript  $o$  denotes that the control is optimal.

The solution (11-7) can be obtained by several different approaches. The approach to be used here will be through the principle of optimality, developed by Richard Bellman [2, 3]. For our purposes, the principle may be stated as follows:

If a closed-loop control  $\mathbf{u}^o(k) = \mathbf{f}[\mathbf{x}(k)]$  is optimal over the interval  $0 \leq k \leq N$ , it is also optimal over any subinterval  $m \leq k \leq N$ , where  $0 \leq m \leq N$ .

The principle of optimality can be applied as follows: Define the scalar  $F_k$  as

$$F_k = \mathbf{x}^T(k)\mathbf{Q}(k)\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}(k)\mathbf{u}(k) \quad (11-8)$$

Then (11-3) can be expressed as

$$J_N = F_0 + F_1 + \cdots + F_{N-1} + F_N$$

Now, let  $S_m$  be the cost from  $k = m$  to  $k = N$ ; that is,

$$S_m = J_N - J_{N-m} = F_{N-m+1} + F_{N-m+2} + \cdots + F_{N-1} + F_N \quad (11-9)$$

These cost terms are illustrated in Fig. 11-1.

Note that  $k$  can vary from 0 to  $N$ , while  $m$  can vary from 1 to  $(N + 1)$ . The principle of optimality states that if  $J_N$  is the optimal cost, then so is  $S_m$ , for  $m = 1, 2, \dots, (N + 1)$ . We can apply the principle of optimality by first minimizing  $S_1 = F_N$ , then choosing  $F_{N-1}$  to minimize

$$S_2 = F_{N-1} + F_N = S_1^o + F_{N-1}$$

then choosing  $F_{N-2}$  to minimize

$$S_3 = F_{N-2} + F_{N-1} + F_N = S_2^o + F_{N-2}$$

and so on, until  $S_{N+1} = J_N$  is minimized. Design using this procedure is also known as dynamic programming.

A simple example will now be given to illustrate optimal control design using the principal of optimality. Then the general design procedure will be developed in Section 11.4. Consider the first-order plant described by

$$x(k + 1) = 2x(k) + u(k) \quad (11-10)$$

Note that this plant is unstable. We wish to determine the control law  $u(k)$  that minimizes

$$J_2 = \sum_{k=0}^2 (x^2(k) + u^2(k)) = \sum_{k=0}^2 F_k$$

Hence we wish to choose  $u(k)$  to minimize

$$J_2 = F_0 + F_1 + F_2 = x^2(0) + u^2(0) + x^2(1) + u^2(1) + x^2(2) + u^2(2) \quad (11-11)$$

subject to the relationship, or *constraint*, (11-10). Note first that  $u(2)$  does not affect any of the other terms in (11-11); thus we must choose  $u(2) = 0$ . This requirement will also appear in the following development.

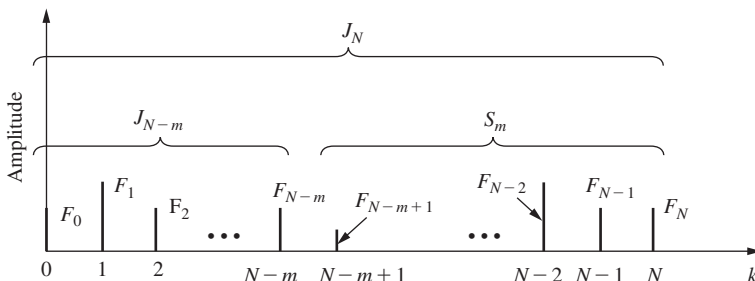


FIGURE 11-1 Terms in the cost function.

First, from (11-9),

$$S_1 = J_N - J_{N-1} = J_2 - J_1 = F_2 = x^2(2) + u^2(2)$$

By the principal of optimality,  $u(2)$  must minimize this function. Hence

$$\frac{\partial S_1}{\partial u(2)} = 2u(2) = 0$$

since  $x(2)$  is independent of  $u(2)$ . Thus  $u(2) = 0$ , and

$$S_1^o = x^2(2) \quad (11-12)$$

Next, we calculate  $S_2$ .

$$S_2 = S_1^o + F_1 = S_1^o + x^2(1) + u^2(1)$$

From (11-10) and (11-12),  $S_2$  may be expressed as

$$S_2 = [2x(1) + u(1)]^2 + x^2(1) + u^2(1)$$

By the principle of optimality,

$$\frac{\partial S_2}{\partial u(1)} = 2[2x(1) + u(1)] + 2u(1) = 0$$

since  $x(1)$  is independent of  $u(1)$ . Thus

$$u(1) = -x(1)$$

and

$$S_2^o = 3x^2(1)$$

By the same procedure,

$$S_3 = J_2 = S_2^o + F_2 = S_2^o + x^2(0) + u^2(0)$$

Hence

$$\begin{aligned} S_3 &= 3x^2(1) + x^2(0) + u^2(0) \\ &= 3[2x(0) + u(0)]^2 + x^2(0) + u^2(0) \end{aligned}$$

and

$$\frac{\partial S_3}{\partial u(0)} = 6[2x(0) + u(0)] + 2u(0) = 0$$

Thus

$$u(0) = -1.5x(0)$$

Therefore, the optimal control sequence is given by

$$u(0) = -1.5x(0)$$

$$u(1) = -x(1)$$

$$u(2) = 0$$

(11-13)

The minimum-cost function is then

$$\begin{aligned} S_3^o &= J_2^o = 3[0.5x(0)]^2 + x^2(0) + [-1.5x(0)]^2 \\ &= 4x^2(0) \end{aligned}$$

Thus *no* choice of  $\{u(0), u(1), u(2)\}$  will yield a smaller value of  $J_2$  than will that of (11-13).

The following points should be made about the example above. First, while no assumption was made concerning the form of the control law, the optimal control law is linear and of the form

$$u(k) = -K(k)x(k)$$

Next, even though the plant is time invariant, the feedback gains required,  $K(k)$ , are time varying. In the following section a general solution to this optimal control design problem will be derived, and the points above will be shown to also apply to the general case.

Note also that the optimal control design is solved in reverse time. The optimal gain  $K(i)$  cannot be calculated until all the remaining  $K(j)$ ,  $i < j \leq N$  are known.

## 11.4 LINEAR QUADRATIC OPTIMAL CONTROL

We will now solve the linear quadratic (LQ) optimal design problem posed in Section 11.3, which is as follows:

For a linear discrete plant described by

$$\begin{aligned} \text{[eq. (11-4)]} \quad \mathbf{x}(k+1) &= \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}(k)\mathbf{x}(k) \end{aligned}$$

determine the control

$$\text{[eq. (11-7)]} \quad \mathbf{u}^o(k) = \mathbf{f}[\mathbf{x}(k)]$$

that minimizes the quadratic cost function

$$\text{[eq. (11-3)]} \quad J_N = \sum_{k=0}^N \mathbf{x}^T(k)\mathbf{Q}(k)\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}(k)\mathbf{u}(k)$$

where  $N$  is finite,  $\mathbf{Q}(k)$  is positive semidefinite, and  $\mathbf{R}(k)$  is positive definite.

First a short review of the differentiation of quadratic functions is in order (see Appendix IV). Given the quadratic function  $\mathbf{x}^T\mathbf{Q}\mathbf{x}$ , then

$$\frac{\partial}{\partial \mathbf{x}} [\mathbf{x}^T\mathbf{Q}\mathbf{x}] = 2\mathbf{Q}\mathbf{x} \quad (11-14)$$

Given the bilinear form  $\mathbf{x}^T\mathbf{Q}\mathbf{y}$ , then

$$\frac{\partial}{\partial \mathbf{x}} [\mathbf{x}^T\mathbf{Q}\mathbf{y}] = \mathbf{Q}\mathbf{y} \quad (11-15)$$

and

$$\frac{\partial}{\partial \mathbf{y}} [\mathbf{x}^T\mathbf{Q}\mathbf{y}] = \mathbf{Q}^T\mathbf{x} \quad (11-16)$$



In (11-14) we can assume that  $\mathbf{Q}$  is symmetric. However, in (11-15) and (11-16),  $\mathbf{Q}$  is not necessarily symmetric. Note also that since  $\mathbf{x}^T \mathbf{Q} \mathbf{y}$  is a scalar,

$$\mathbf{x}^T \mathbf{Q} \mathbf{y} = (\mathbf{x}^T \mathbf{Q} \mathbf{y})^T = \mathbf{y}^T \mathbf{Q}^T \mathbf{x}$$

In (11-9) we defined  $S_m$  to be the cost from the  $(N - m)$  sample instant to the  $N$  (terminal) sample instant, that is,

[eq. (11-9)] 
$$S_m = J_N - J_{N-m}$$

Hence from Fig. 11-1 we see that  $S_{m+1}$  can be expressed as

$$S_{m+1} = S_m + F_{N-m}$$

From (11-8), we can then write

$$S_{m+1} = S_m + \mathbf{x}^T(N - m) \mathbf{Q}(N - m) \mathbf{x}(N - m) + \mathbf{u}^T(N - m) \mathbf{R}(N - m) \mathbf{u}(N - m) \quad (11-17)$$

Thus, from the principle of optimality,

$$\frac{\partial S_{m+1}}{\partial \mathbf{u}(N - m)} = \mathbf{0} \quad (11-18)$$

if  $S_m$  in (11-17) is replaced with  $S_m^o$ .

Consider first  $S_1$ :

$$S_1 = F_N = \mathbf{x}^T(N) \mathbf{Q}(N) \mathbf{x}(N) + \mathbf{u}^T(N) \mathbf{R}(N) \mathbf{u}(N) \quad (11-19)$$

Since  $\mathbf{x}(N)$  is independent of  $\mathbf{u}(N)$ ,  $\mathbf{u}^o(N) = 0$  and

$$S_1^o = \mathbf{x}(N)^T \mathbf{Q}(N) \mathbf{x}(N) \quad (11-20)$$

We see then that  $S_1^o$  is quadratic in  $\mathbf{x}(N)$ . For  $S_2$ , we get the expression, from (11-17), that

$$S_2 = S_1^o + \mathbf{x}^T(N - 1) \mathbf{Q}(N - 1) \mathbf{x}(N - 1) + \mathbf{u}^T(N - 1) \mathbf{R}(N - 1) \mathbf{u}(N - 1) \quad (11-21)$$

where, from (11-4) and (11-20),

$$S_1^o = [\mathbf{A}\mathbf{x}(N - 1) + \mathbf{B}\mathbf{u}(N - 1)]^T \mathbf{Q}(N) [\mathbf{A}\mathbf{x}(N - 1) + \mathbf{B}\mathbf{u}(N - 1)] \Big|_{\mathbf{u}^o(N-1)} \quad (11-22)$$

With (11-22) substituted into (11-21), we solve

$$\frac{\partial S_2}{\partial \mathbf{u}(N - 1)} = 0$$

for  $\mathbf{u}(N - 1)$ , with the result of the form

$$\mathbf{u}^o(N - 1) = -\mathbf{K}(N - 1) \mathbf{x}(N - 1) \quad (11-23)$$

The derivation of this result is straightforward and is given as Problem 11.4-1. We will not at this time solve for  $\mathbf{K}(N - 1)$ . However, when (11-23) is substituted into (11-21), the result is seen to be quadratic in  $\mathbf{x}(N - 1)$ . Hence we can write

$$S_2^o = \mathbf{x}^T(N - 1)\mathbf{P}(N - 1)\mathbf{x}(N - 1) \quad (11-24)$$

where  $\mathbf{P}(N - 1)$  is symmetric.

Following the development in the paragraph above in solving for  $S_3$ , we see that  $S_3^o$  is also quadratic. Hence we may write

$$S_3^o = \mathbf{x}^T(N - 2)\mathbf{P}(N - 2)\mathbf{x}(N - 2) \quad (11-25)$$

Thus the general relationship of  $S_m^o$  is seen to be

$$S_m^o = \mathbf{x}^T(N - m + 1)\mathbf{P}(N - m + 1)\mathbf{x}(N - m + 1) \quad (11-26)$$

From (11-4), we can then express  $S_m^o$  as

$$S_m^o = [\mathbf{Ax}(N - m) + \mathbf{Bu}(N - m)]^T\mathbf{P}(N - m + 1)[\mathbf{Ax}(N - m) + \mathbf{Bu}(N - m)] \quad (11-27)$$

Next we substitute (11-27) into (11-17) to get the expression for  $S_{m+1}$ . Then we differentiate  $S_{m+1}$  with respect to  $\mathbf{u}(N - m)$ , set this to zero, and solve for  $\mathbf{u}^o(N - m)$ . The algebra becomes unwieldy, and to simplify this, we will drop the notational dependence of  $(N - m)$ . Then, in the final solution, we will reinsert this dependence. From (11-27) and (11-17),

$$S_{m+1} = [\mathbf{Ax} + \mathbf{Bu}]^T\mathbf{P}(N - m + 1)[\mathbf{Ax} + \mathbf{Bu}] + \mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u} \quad (11-28)$$

Then, from (11-14), (11-15), and (11-16),

$$\frac{\partial S_{m+1}}{\partial \mathbf{u}} = \mathbf{B}^T\mathbf{P}(N - m + 1)[\mathbf{Ax} + \mathbf{Bu}] + \mathbf{B}^T\mathbf{P}(N - m + 1)[\mathbf{Ax} + \mathbf{Bu}] + 2\mathbf{R}\mathbf{u} = 0$$

or

$$2\mathbf{B}^T\mathbf{P}(N - m + 1)\mathbf{Ax} + 2[\mathbf{B}^T\mathbf{P}(N - m + 1)\mathbf{B} + \mathbf{R}]\mathbf{u} = 0$$

Thus the desired solution is

$$\mathbf{u}^o = -[\mathbf{B}^T\mathbf{P}(N - m + 1)\mathbf{B} + \mathbf{R}]^{-1}\mathbf{B}^T\mathbf{P}(N - m + 1)\mathbf{Ax} \quad (11-29)$$

Hence, from (11-23) and (11-29), the optimal gain matrix is

$$\begin{aligned} \mathbf{K}(N - m) &= [\mathbf{B}^T(N - m)\mathbf{P}(N - m + 1)\mathbf{B}(N - m) + \mathbf{R}(N - m)]^{-1} \\ &\quad \times \mathbf{B}^T(N - m)\mathbf{P}(N - m + 1)\mathbf{A}(N - m) \end{aligned} \quad (11-30)$$

and

$$\mathbf{u}^o(N - m) = -\mathbf{K}(N - m)\mathbf{x}(N - m) \quad (11-31)$$

Next we develop the expression for  $S_{m+1}^o$ . In (11-28), from (11-31),

$$\mathbf{Ax} + \mathbf{Bu} = [\mathbf{A} - \mathbf{BK}]\mathbf{x}$$

and

$$S_{m+1}^o = \{[\mathbf{A} - \mathbf{BK}]\mathbf{x}\}^T \mathbf{P}(N - m + 1)[\mathbf{A} - \mathbf{BK}]\mathbf{x} + \mathbf{x}^T \mathbf{Q}\mathbf{x} + [\mathbf{K}\mathbf{x}]^T \mathbf{R}\mathbf{K}\mathbf{x}$$

or

$$S_{m+1}^o = \mathbf{x}^T \{[\mathbf{A} - \mathbf{BK}]^T \mathbf{P}(N - m + 1)[\mathbf{A} - \mathbf{BK}] + \mathbf{Q} + \mathbf{K}^T \mathbf{R}\mathbf{K}\} \mathbf{x}$$

But, from (11-26),

$$S_{m+1}^o = \mathbf{x}^T \mathbf{P}(N - m)\mathbf{x}$$

Hence

$$\begin{aligned} \mathbf{P}(N - m) &= [\mathbf{A}(N - m) - \mathbf{B}(N - m)\mathbf{K}(N - m)]^T \mathbf{P}(N - m + 1) \\ &\quad \times [\mathbf{A}(N - m) - \mathbf{B}(N - m)\mathbf{K}(N - m)] \\ &\quad + \mathbf{Q}(N - m) + \mathbf{K}^T(N - m)\mathbf{R}(N - m)\mathbf{K}(N - m) \end{aligned} \quad (11-32)$$

The final design equations will now be summarized. The design progresses backward in time from  $k = N$ . We know the final value of the  $\mathbf{P}$  matrix from (11-20) and (11-26).

$$\mathbf{P}(N) = \mathbf{Q}(N) \quad (11-33)$$

For a time-invariant system and cost function, from (11-30) we obtain the optimal gain-matrix expression

$$\mathbf{K}(N - m) = [\mathbf{B}^T \mathbf{P}(N - m + 1)\mathbf{B} + \mathbf{R}]^{-1} \mathbf{B}^T \mathbf{P}(N - m + 1)\mathbf{A} \quad (11-34)$$

Hence we can solve for  $\mathbf{K}(N - 1)$ , and from (11-32),

$$\begin{aligned} \mathbf{P}(N - m) &= [\mathbf{A} - \mathbf{BK}(N - m)]^T \mathbf{P}(N - m + 1)[\mathbf{A} - \mathbf{BK}(N - m)] \\ &\quad + \mathbf{Q} + \mathbf{K}^T(N - m)\mathbf{R}\mathbf{K}(N - m) \end{aligned} \quad (11-35)$$

This equation can be expressed in other forms (see Problem 11.4-2). One of the simpler forms is

$$\mathbf{P}(N - m) = \mathbf{A}^T \mathbf{P}(N - m + 1)[\mathbf{A} - \mathbf{BK}(N - m)] + \mathbf{Q} \quad (11-36)$$

Thus we may solve either (11-35) or (11-36) for  $\mathbf{P}(N - 1)$ . We see then that (11-34) and either (11-35) or (11-36) form a set of nonlinear difference equations which may be solved recursively for  $\mathbf{K}(N - m)$  and  $\mathbf{P}(N - m)$ . Thus (11-33), (11-34), and (11-36) form the design equations.

We can also express the design equations as a function of  $k$ . In (11-31), (11-34), and (11-36), let  $k = N - m$ . The design equations are, for  $k = 0, 1, 2, \dots, N - 1$ ,

$$\mathbf{u}^o(k) = -\mathbf{K}(k)\mathbf{x}(k) \quad (11-37)$$

$$\mathbf{K}(k) = [\mathbf{B}^T \mathbf{P}(k + 1)\mathbf{B} + \mathbf{R}]^{-1} \mathbf{B}^T \mathbf{P}(k + 1)\mathbf{A} \quad (11-38)$$

$$\mathbf{P}(k) = \mathbf{A}^T \mathbf{P}(k + 1) [\mathbf{A} - \mathbf{B} \mathbf{K}(k)] + \mathbf{Q} \tag{11-39}$$

with  $\mathbf{P}(N) = \mathbf{Q}$  and  $\mathbf{K}(N) = 0$ .

Note from (11-9) that  $J_N$  can be expressed as

$$J_N = S_N + J_0 = S_N + \mathbf{x}^T(0) \mathbf{Q}(0) \mathbf{x}(0) + \mathbf{u}^T(0) \mathbf{R}(0) \mathbf{u}(0)$$

since  $F_0 = J_0$ . But from Fig. 11-1, this expression is simply  $S_{N+1}$ . Thus from (11-26), the minimum cost is given by

$$J_N^o = S_{N+1}^o = \mathbf{x}^T(0) \mathbf{P}(0) \mathbf{x}(0) \tag{11-40}$$

For the case that the system and/or the cost function are time varying, the following substitutions must be made in (11-38) and (11-39):

$$\begin{aligned} \mathbf{A} &\rightarrow \mathbf{A}(k) \\ \mathbf{B} &\rightarrow \mathbf{B}(k) \\ \mathbf{Q} &\rightarrow \mathbf{Q}(k) \\ \mathbf{R} &\rightarrow \mathbf{R}(k) \end{aligned} \tag{11-41}$$

Two examples illustrating this design procedure will now be given.

**EXAMPLE 11.1**

We wish to design an optimal control law for the system

$$x(k + 1) = 2x(k) + u(k)$$

with the cost function

$$J_2 = \sum_{k=0}^2 x^2(k) + u^2(k)$$

Note that this is the same design problem considered in Section 11.3. Now, the required parameters are

$$\begin{aligned} A &= 2 & Q &= 1 \\ B &= 1 & R &= 1 \end{aligned}$$

From (11-33),

$$P(2) = Q(2) = Q = 1$$

From (11-38), the gain required is

$$\begin{aligned} K(2 - 1) = K(1) &= [\mathbf{B}^T \mathbf{P}(2) \mathbf{B} + \mathbf{R}]^{-1} \mathbf{B}^T \mathbf{P}(2) \mathbf{A} \\ &= [1 + 1]^{-1} (1)(1)(2) = 1 \end{aligned}$$

From (11-39),

$$\begin{aligned} P(1) &= \mathbf{A}^T \mathbf{P}(2) \{ \mathbf{A} - \mathbf{B} \mathbf{K}(1) \} + \mathbf{Q} \\ &= 2(1) \{ 2 - (1)(1) \} + 1 = 3 \end{aligned}$$

Then, from (11-38),

$$\begin{aligned} K(0) &= [B^T P(1)B + R]^{-1} B^T P(1)A \\ &= [3 + 1]^{-1} (1)(3)(2) = 1.5 \end{aligned}$$

and from (11-39),

$$P(0) = 2(3)\{2 - 1.5\} + 1 = 4$$

Hence the optimal gain schedule is

$$\{K(0), K(1)\} = \{1.5, 1\}$$

and from (11-40), the minimum cost is

$$J_2^o = x^T(0)P(0)x(0) = 4x^2(0)$$

The results check those obtained in Section 11.3.

### EXAMPLE 11.2

As a second example, we consider the servo system utilized in several examples of pole-assignment design in Chapter 9.

The system is shown in Fig. 11-2, and has the state model (see Example 9.1)

$$\begin{aligned} \mathbf{x}(k + 1) &= \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k) \\ y(k) &= [1 \quad 0] \mathbf{x}(k) \end{aligned}$$

where  $x_1(t)$  is the shaft angle and  $x_2(t)$  is the shaft angular velocity. We choose the cost function to be

$$J_2 = \sum_{k=0}^2 \mathbf{x}^T(k) \mathbf{Q} \mathbf{x}(k) + R u^2(k)$$

with

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad R = 1$$

Thus velocity is ignored in the cost function. We are attempting to minimize the magnitude of the position  $x_1(k)$  without regard to the velocity  $x_2(k)$  required. The weight of the control,  $R$ , in the cost function is chosen arbitrarily.  $N$  is chosen equal to 2 so that the solution can be calculated by hand. Thus, from (11-33),

$$\mathbf{P}(2) = \mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

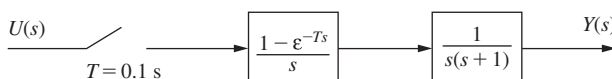


FIGURE 11-2 Servomotor system.

In (11-38), for  $k = 1$ ,

$$\begin{aligned} \mathbf{B}^T \mathbf{P}(2) \mathbf{B} + R &= [0.00484 \quad 0.0952] \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} + 1 \\ &= [0.00484 \quad 0] \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} + 1 = 1.0000234 \end{aligned}$$

Then, in (11-38),

$$\begin{aligned} \mathbf{K}(1) &= [\mathbf{B}^T \mathbf{P}(2) \mathbf{B} + R]^{-1} \mathbf{B}^T \mathbf{P}(2) \mathbf{A} \\ &= \frac{1}{1.0000234} [0.00484 \quad 0.0952] \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \\ &= [0.00484 \quad 0.000461] \end{aligned}$$

To calculate  $\mathbf{P}(1)$  from (11-39), we need

$$\begin{aligned} \mathbf{B} \mathbf{K}(1) &= \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} [0.00484 \quad 0.000461] \\ &= \begin{bmatrix} 2.34 \times 10^{-5} & 2.23 \times 10^{-6} \\ 4.61 \times 10^{-4} & 4.39 \times 10^{-5} \end{bmatrix} \end{aligned}$$

Then

$$[\mathbf{A} - \mathbf{B} \mathbf{K}(1)] = \begin{bmatrix} 1.00 & 0.0952 \\ -4.61 \times 10^{-4} & 0.905 \end{bmatrix}$$

and

$$\mathbf{A}^T \mathbf{P}(2) [\mathbf{A} - \mathbf{B} \mathbf{K}(1)] = \begin{bmatrix} 1 & 0.0952 \\ 0.0952 & 0.00906 \end{bmatrix}$$

Then, from (11-39),

$$\begin{aligned} \mathbf{P}(1) &= \mathbf{A}^T \mathbf{P}(2) [\mathbf{A} - \mathbf{B} \mathbf{K}(1)] + \mathbf{Q} \\ &= \begin{bmatrix} 2.0 & 0.0952 \\ 0.0952 & 0.00906 \end{bmatrix} \end{aligned}$$

The calculations above are carried through in detail to illustrate the calculations required for one step of the solution of the difference equations (11-38) and (11-39) for a second-order system with a single input. Obviously, a computer solution is required. If these difference equations are solved again, the results are

$$\mathbf{K}(0) = [0.0187 \quad 0.00298]$$

and

$$\mathbf{P}(0) = \begin{bmatrix} 3.0 & 0.276 \\ 0.276 & 0.0419 \end{bmatrix}$$

A MATLAB program that performs the calculations of this example is given by

```
format ShortEng
A = [1 0.0952;0 0.905]; B = [0.00484; 0.0952];
Q = [1 0;0 0]; R = 1; N = 2; P = Q;
disp('      k      Gains')
for n=1:N
KK = inv(B'*P*B+R)*B'*P*A;
P1 =A'*P*(A - B*KK)+Q;
P=P1;
k = N - n+1;
[k, KK]
end
disp('The final value of the P matrix is:'), P
```

### EXAMPLE 11.3

Next a more practical design will be performed, using a computer solution; that is, (11-38) and (11-39) are solved recursively via the computer. We will compare the results with those of Example 9.1. In Example 9.1, the plant equations are (see Fig. 11-2)

$$\mathbf{x}(k + 1) = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k)$$

and the closed-loop characteristic equation zeros had a time constant of 1.0 s. We generally consider the response to have settled out in five time constants, or 5 s in this case. Hence we will choose  $N$  of the cost function of (11-3) to be 51 and compare responses for the first 5 s. The  $\mathbf{Q}$  and  $\mathbf{R}$  matrices of (11-3) are chosen to be the same as in Example 11.2, that is,

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad R = 1$$

The optimal gains obtained for this design are plotted in Fig. 11-3. Also shown are the optimal gains for values of  $R$  of 0.1 and 0.03, with  $\mathbf{Q}$  unchanged. As the value of  $R$  is reduced, the contribution of the control effort to the cost function is reduced. Hence the control effort will be increased in order to reduce the magnitudes of the states. This effect is seen in Fig. 11-3.

Shown in Fig. 11-4 is the initial-condition response, for  $x_1(t)$ , of this system for  $R = 1.0$  and  $R = 0.03$ , with

$$\mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

The effects of the choice of  $R$  on the transient response of the closed-loop system can be seen from this figure. Also, by comparing Figs. 11-3 and 11-4, we see that the response has reached zero before the gains begin decreasing for the case that  $R = 0.03$ . This is not true for the case that  $R = 1.0$ .

The initial-condition response for Example 9.1 is given in Fig. 9-2. It is seen that the optimal system response settles faster for  $R = 0.03$ , but slower for  $R = 1.0$ .

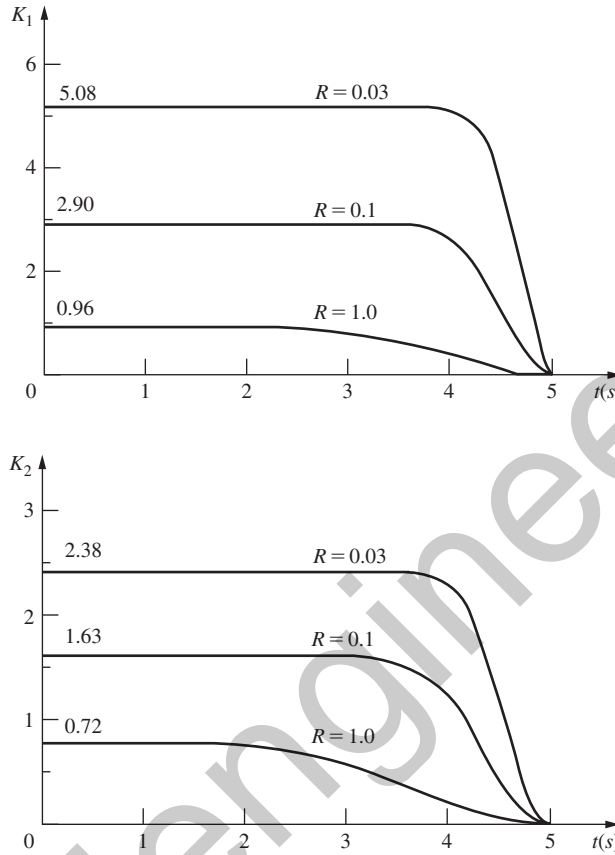


FIGURE 11-3 Optimal gains for Example 11.3.

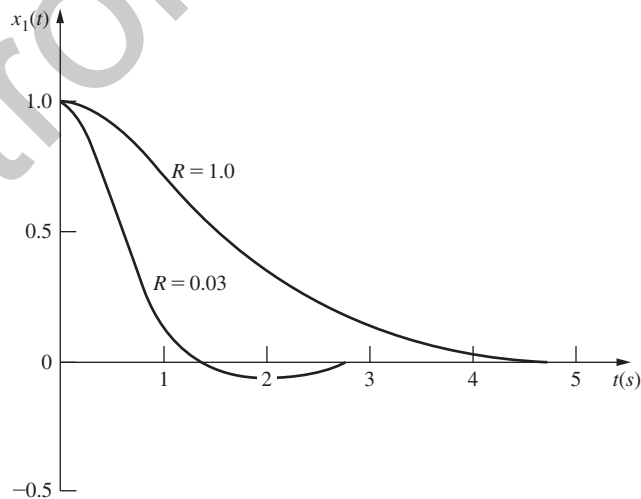


FIGURE 11-4 Time response for Example 11.3.



## 11.5 THE MINIMUM PRINCIPLE

In the developments above, Bellman's principle was used in deriving the optimal control design equations (11-38) and (11-39). The equations may also be derived utilizing the minimum principle [4]. The minimum principle is presented in this section. This principle will prove to be useful when steady-state optimal control is considered in the following section.

The optimal control problem posed in Section 11.2 may be stated as follows:

For a linear discrete plant described by

$$\begin{aligned} \text{[eq. (11-4)]} \quad \mathbf{x}(k+1) &= \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}(k)\mathbf{x}(k) \end{aligned}$$

determines the control

$$\text{[eq. (11-7)]} \quad \mathbf{u}^o(k) = \mathbf{f}[\mathbf{x}(k)]$$

that minimizes the quadratic cost function

$$\text{[eq. (11-3)]} \quad J_N = \frac{1}{2} \sum_{k=0}^N \mathbf{x}^T(k)\mathbf{Q}(k)\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}(k)\mathbf{u}(k)$$

where  $N$  is finite,  $\mathbf{Q}(k)$  is positive semidefinite, and  $\mathbf{R}(k)$  is positive definite. In (11-7), the superscript  $o$  denotes that the control is optimal. Note that the factor  $\frac{1}{2}$  added to (11-3) will not change the results, but will simplify the following derivations somewhat.

The minimum principle may be utilized to solve the optimal control problem.

**MINIMUM PRINCIPLE [4].** If the input  $\mathbf{u}^o(k)$  and the corresponding trajectory  $\mathbf{x}^o(k)$  are optimal, there exists a nontrivial vector sequence  $\{\mathbf{p}^o(k)\}$  such that  $\mathbf{u}^o(k)$  is the value of  $\mathbf{u}(k)$  that minimizes the Hamiltonian

$$H = \frac{1}{2}[\mathbf{x}^{oT}(k)\mathbf{Q}(k)\mathbf{x}^o(k) + \mathbf{u}^{oT}(k)\mathbf{R}(k)\mathbf{u}^o(k)] + [\mathbf{p}^o(k+1)]^T[\mathbf{A}(k)\mathbf{x}^o(k) + \mathbf{B}(k)\mathbf{u}^o(k)] \quad (11-42)$$

and the "costate vector"  $\mathbf{p}^o(k)$  satisfies

$$\mathbf{p}^o(k) = \frac{\partial H}{\partial \mathbf{x}^o(k)}, \quad \mathbf{p}^o(N) = \mathbf{Q}(N)\mathbf{x}^o(N) \quad (11-43)$$

for  $k \leq N$ .

From (11-43) we obtain

$$\mathbf{p}^o(k) = \mathbf{Q}(k)\mathbf{x}^o(k) + \mathbf{A}^T(k)\mathbf{p}^o(k+1) \quad (11-44)$$

and  $\partial H / \partial \mathbf{u}(k) = 0$  yields

$$\mathbf{u}^o(k) = -\mathbf{R}^{-1}(k)\mathbf{B}^T(k)\mathbf{p}^o(k+1) \quad (11-45)$$

Hence this approach leads to the state equations (11-4),

$$\text{[eq. (11-4)]} \quad \mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k)$$

and the costate equations (11-44), with the optimal control given by (11-45). The state equations will have an initial condition  $\mathbf{x}(k_0)$ , and are solved forward in time. The costate equations have the final state (11-43), and are solved backward in time. However, since the state and costate equations are coupled, these equations constitute a linear two-point boundary value problem. It is beyond the scope of this book to discuss the solution of two-point boundary value problems. However, it can be shown that the solution of (11-4), (11-44), and (11-45) yields the same optimal gain matrix as does the solution of (11-38) and (11-39).

## 11.6 STEADY-STATE OPTIMAL CONTROL

If we consider the design results of Example 11.3, given in Fig. 11-3, we see that the optimal gains approach constant values with decreasing time. In general, this is true for time-invariant systems and cost functions.

Note also that if we increase  $N$  in Example 11.3, the optimal gains for the final 51 sample instants are the same as those calculated in this example. The values of the gains at earlier sampling instants will simply be those constant values indicated in Fig. 11-3. Hence if we allow  $N$  to approach infinity, or if we allow the initial time to approach  $-\infty$ , we obtain a steady-state solution in which the optimal gains are constant values. We consider this steady-state solution in this section.

First we consider the difference equations (11-38) and (11-39) employed in the design of optimal control systems.

$$\text{[eq. (11-38)]} \quad \mathbf{K}(k) = [\mathbf{B}^T \mathbf{P}(k+1) \mathbf{B} + \mathbf{R}]^{-1} \mathbf{B}^T \mathbf{P}(k+1) \mathbf{A}$$

$$\text{[eq. (11-39)]} \quad \mathbf{P}(k) = \mathbf{A}^T \mathbf{P}(k+1) [\mathbf{A} - \mathbf{B} \mathbf{K}(k)] + \mathbf{Q}$$

In this section  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{Q}$ , and  $\mathbf{R}$  are assumed to be *constant*. If (11-38) is substituted into (11-39), the result is a difference equation for the matrix  $\mathbf{P}$ .

$$\mathbf{P}(k) = \mathbf{A}^T \mathbf{P}(k+1) \{ \mathbf{A} - \mathbf{B} [\mathbf{B}^T \mathbf{P}(k+1) \mathbf{B} + \mathbf{R}]^{-1} \times \mathbf{B}^T \mathbf{P}(k+1) \mathbf{A} \} + \mathbf{Q} \quad (11-46)$$

This equation is often written in a slightly different form,

$$\mathbf{P}(k) = \mathbf{A}^T \mathbf{P}(k+1) \mathbf{A} + \mathbf{Q} - \mathbf{A}^T \mathbf{P}(k+1) \times \mathbf{B} [\mathbf{B}^T \mathbf{P}(k+1) \mathbf{B} + \mathbf{R}]^{-1} \mathbf{B}^T \mathbf{P}(k+1) \mathbf{A} \quad (11-47)$$

and is referred to as the *discrete Riccati equation*. The inverse in (11-47) always exists since  $\mathbf{R}$  is positive definite and  $\mathbf{P}(k+1)$  is at least positive semidefinite [the cost function in (11-3) and (11-26) cannot be negative].

Before deriving the steady-state solution to (11-47), we will derive a non-recursive solution for  $\mathbf{P}(k)$  in the discrete Riccati equation for time-invariant systems. This solution will then yield the steady-state solution. We can express the state and costate equations of (11-4) and (11-44) as

$$\begin{bmatrix} \mathbf{x}^o(k) \\ \mathbf{p}^o(k) \end{bmatrix} = \mathbf{H} \begin{bmatrix} \mathbf{x}^o(k+1) \\ \mathbf{p}^o(k+1) \end{bmatrix} \quad (11-48)$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{A}^{-1} \mathbf{R}_c \\ \mathbf{Q} \mathbf{A}^{-1} & \mathbf{A}^T + \mathbf{Q} \mathbf{A}^{-1} \mathbf{R}_c \end{bmatrix} \quad (11-49)$$

and

$$\mathbf{R}_C = \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T \quad (11-50)$$

Note that  $\mathbf{H}$  is  $2n \times 2n$ . For a matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{D} & \mathbf{E} \\ \mathbf{F} & \mathbf{G} \end{bmatrix} \quad (11-51)$$

where the partitions  $\mathbf{D}$ ,  $\mathbf{E}$ ,  $\mathbf{F}$ , and  $\mathbf{G}$  are  $n \times n$ , then

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{D}^{-1} + \mathbf{D}^{-1}\mathbf{E}[\mathbf{G} - \mathbf{F}\mathbf{D}^{-1}\mathbf{E}]^{-1}\mathbf{F}\mathbf{D}^{-1} & -\mathbf{D}^{-1}\mathbf{E}[\mathbf{G} - \mathbf{F}\mathbf{D}^{-1}\mathbf{E}]^{-1} \\ -[\mathbf{G} - \mathbf{F}\mathbf{D}^{-1}\mathbf{E}]^{-1}\mathbf{F}\mathbf{D}^{-1} & [\mathbf{G} - \mathbf{F}\mathbf{D}^{-1}\mathbf{E}]^{-1} \end{bmatrix} \quad (11-52)$$

(see Appendix IV). Hence, from (11-49), (11-51), and (11-52),

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{A} + \mathbf{R}_C\mathbf{A}^{-T}\mathbf{Q} & -\mathbf{R}_C\mathbf{A}^{-T} \\ -\mathbf{A}^{-T}\mathbf{Q} & \mathbf{A}^{-T} \end{bmatrix} \quad (11-53)$$

where

$$\mathbf{A}^{-T} = (\mathbf{A}^{-1})^T = (\mathbf{A}^T)^{-1} \quad (11-54)$$

If we denote  $\mathbf{h}$  as an eigenvector of  $\mathbf{H}$ , and  $\lambda$  as an eigenvalue, then

$$\mathbf{H}\mathbf{h} = \lambda\mathbf{h} \quad (11-55)$$

Let

$$\mathbf{h} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad (11-56)$$

where  $\mathbf{f}$  and  $\mathbf{g}$  are  $n$ -vectors. Substituting (11-49) into (11-55) yields two equations. Also, substituting (11-53) into

$$\mathbf{H}^{-T} \begin{bmatrix} \mathbf{g} \\ -\mathbf{f} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{g} \\ -\mathbf{f} \end{bmatrix} \quad (11-57)$$

yields the same two equations. Thus  $\lambda$  is an eigenvalue of  $\mathbf{H}^{-1}$  and hence  $\mathbf{H}^{-T}$  and therefore  $1/\lambda$  is an eigenvalue of  $\mathbf{H}$ . Thus the eigenvalues of  $\mathbf{H}$  are such that the reciprocal of every eigenvalue is also an eigenvalue.

Next we define the vectors  $\mathbf{v}(k)$  and  $\mathbf{s}(k)$  and the similarity transformation  $\mathbf{W}$  (see Section 2.10) such that

$$\begin{bmatrix} \mathbf{x}^o(k) \\ \mathbf{p}^o(k) \end{bmatrix} = \mathbf{W} \begin{bmatrix} \mathbf{v}(k) \\ \mathbf{s}(k) \end{bmatrix} \quad (11-58)$$

Then, for the case of distinct eigenvalues of  $\mathbf{H}$ , from (11-48) we write

$$\begin{bmatrix} \mathbf{v}(k) \\ \mathbf{s}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{\Lambda} & 0 \\ 0 & \mathbf{\Lambda}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{v}(k+1) \\ \mathbf{s}(k+1) \end{bmatrix} = \mathbf{W}^{-1}\mathbf{H}\mathbf{W} \begin{bmatrix} \mathbf{v}(k+1) \\ \mathbf{s}(k+1) \end{bmatrix} \quad (11-59)$$

where  $\Lambda$  is a diagonal matrix of the eigenvalues of  $\mathbf{H}$  that occur outside the unit circle. Then, in (11-58), the columns of  $\mathbf{W}$  are the eigenvectors of  $\mathbf{H}$ , and  $\mathbf{W}$  is the modal matrix, as shown in Section 2.10.

Next, let the matrix  $\mathbf{W}$  be partitioned into four  $n \times n$  matrices.

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{W}_{21} & \mathbf{W}_{22} \end{bmatrix} \quad (11-60)$$

Since, from (11-43),

$$\mathbf{p}^o(N) = \mathbf{Q}\mathbf{x}^o(N)$$

from (11-58) and (11-60),

$$\mathbf{p}^o(N) = [\mathbf{W}_{21}\mathbf{v}(N) + \mathbf{W}_{22}\mathbf{s}(N)] = \mathbf{Q}\mathbf{x}^o(N) = \mathbf{Q}[\mathbf{W}_{11}\mathbf{v}(N) + \mathbf{W}_{12}\mathbf{s}(N)] \quad (11-61)$$

We solve this equation for  $\mathbf{s}(N)$ .

$$\mathbf{s}(N) = -[\mathbf{W}_{22} - \mathbf{Q}\mathbf{W}_{12}]^{-1}[\mathbf{W}_{21} - \mathbf{Q}\mathbf{W}_{11}]\mathbf{v}(N) = \mathbf{T}\mathbf{v}(N) \quad (11-62)$$

which defines the matrix  $\mathbf{T}$ . Also, for (11-59), we can express  $\mathbf{v}(N)$  and  $\mathbf{s}(N)$  as, with  $j$  as a positive integer,

$$\mathbf{v}(N) = \Lambda^{-j}\mathbf{v}(N-j) \quad (11-63)$$

$$\mathbf{s}(N) = \Lambda^j\mathbf{s}(N-j) \quad (11-64)$$

Thus, from (11-62) and (11-64),

$$\mathbf{s}(N-j) = \Lambda^{-j}\mathbf{s}(N) = \Lambda^{-j}\mathbf{T}\Lambda^{-j}\mathbf{v}(N-j) = \mathbf{G}(j)\mathbf{v}(N-j) \quad (11-65)$$

which defines the matrix  $\mathbf{G}(j)$ .

From (11-58) and (11-60),

$$\mathbf{x}^o(N-j) = \mathbf{W}_{11}\mathbf{v}(N-j) + \mathbf{W}_{12}\mathbf{s}(N-j) \quad (11-66)$$

$$\mathbf{p}^o(N-j) = \mathbf{W}_{21}\mathbf{v}(N-j) + \mathbf{W}_{22}\mathbf{s}(N-j) \quad (11-67)$$

Eliminating  $\mathbf{v}(N-j)$  and  $\mathbf{s}(N-j)$  for (11-65), (11-66), and (11-67) yields

$$\begin{aligned} \mathbf{p}^o(N-j) &= [\mathbf{W}_{21} + \mathbf{W}_{22}\mathbf{G}(j)][\mathbf{W}_{11} + \mathbf{W}_{12}\mathbf{G}(j)]^{-1}\mathbf{x}^o(N-j) \\ &= \mathbf{M}(j)\mathbf{x}^o(N-j) \end{aligned} \quad (11-68)$$

which defines the matrix  $\mathbf{M}(j)$ . This equation is a nonrecursive solution to the state and costate equations (11-4) and (11-44).

To find the optimal gain,  $\mathbf{K}(k)$ , of (11-36), we let  $N-j = k+1$  in (11-68).

$$\mathbf{p}^o(k+1) = \mathbf{M}(N-k-1)\mathbf{x}^o(k+1) = \mathbf{M}(N-k-1)[\mathbf{A}\mathbf{x}^o(k) + \mathbf{B}\mathbf{u}^o(k)] \quad (11-69)$$

Since the optimal control from (11-45) is

$$\mathbf{u}^o(k) = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{p}^o(k+1) \quad (11-70)$$

solving these two equations for  $\mathbf{u}^o(k)$  yields

$$\mathbf{u}^o(k) = -[\mathbf{R} + \mathbf{B}^T\mathbf{M}(N-k-1)\mathbf{B}]^{-1}\mathbf{B}^T\mathbf{M}(N-k-1)\mathbf{A}\mathbf{x}^o(k)$$

Comparing this equation with (11-37) and (11-38), we see that

$$\mathbf{M}(N-k-1) = \mathbf{P}(k+1) \quad (11-71)$$

Then, from (11-65) and (11-68),

$$\mathbf{M}(k+1) = [\mathbf{W}_{21} + \mathbf{W}_{22}\mathbf{G}(N-k-1)][\mathbf{W}_{11} + \mathbf{W}_{12}\mathbf{G}(N-k-1)]^{-1} \quad (11-72)$$

where

$$\mathbf{G}(N-k-1) = -\mathbf{A}^{-(N-k-1)}[\mathbf{W}_{22} - \mathbf{Q}\mathbf{W}_{12}]^{-1}[\mathbf{W}_{21} - \mathbf{Q}\mathbf{W}_{11}]\mathbf{A}^{-(N-k-1)} \quad (11-73)$$

The optimal gain is given by, from (11-70),

$$\mathbf{K}(k) = [\mathbf{R} + \mathbf{B}^T\mathbf{M}(N-k-1)\mathbf{B}]^{-1}\mathbf{B}^T\mathbf{M}(N-k-1)\mathbf{A} \quad (11-74)$$

and the nonrecursive solution of LQ design is (11-72) and (11-74).

For the steady-state solution such that the gains in (11-38) have become constant values, it must then be true in (11-47) that

$$\mathbf{P}(k) = \mathbf{P}(k+1) = \text{constant matrix} \quad (11-75)$$

We will denote this constant matrix as  $\hat{\mathbf{P}}$ . Then (11-47) becomes

$$\hat{\mathbf{P}} = \mathbf{A}^T\hat{\mathbf{P}}\mathbf{A} + \mathbf{Q} - \mathbf{A}^T\hat{\mathbf{P}}\mathbf{B}[\mathbf{B}^T\hat{\mathbf{P}}\mathbf{B} + \mathbf{R}]^{-1}\mathbf{B}^T\hat{\mathbf{P}}\mathbf{A} \quad (11-76)$$

This equation is referred to as the *algebraic Riccati equation*. Perhaps the simplest approach to finding the solution of this equation is that indicated in Example 11.3—set  $N$  to a large value and calculate the values of the  $\mathbf{P}$  matrix (by computer) until the matrix elements become constant values. Then we have the solution to (11-76).

We may also find the solution to the algebraic Riccati equation from the foregoing nonrecursive solution to the discrete Riccati equation. In (11-68),

$$\mathbf{p}^o(N-j) = \mathbf{M}(j)\mathbf{x}^o(N-j) \quad (11-77)$$

The steady-state solution is obtained by allowing  $j$  to approach  $\infty$ . From (11-65) and (11-68)

$$\lim_{j \rightarrow \infty} \mathbf{M}(j) = \mathbf{W}_{21}\mathbf{W}_{11}^{-1} \quad (11-78)$$

since  $\Lambda$  contains the eigenvalues outside the unit circle, and thus

$$\lim_{j \rightarrow \infty} \Lambda^{-j} = 0$$

Hence, from (11-71), the solution to the algebraic Riccati equation (11-76) is

$$\hat{\mathbf{P}} = \mathbf{W}_{21} \mathbf{W}_{11}^{-1} = \lim_{j \rightarrow \infty} \mathbf{M}(j) \quad (11-79)$$

and, for (11-74),

$$\hat{\mathbf{K}} = [\mathbf{R} + \mathbf{B} \hat{\mathbf{P}} \mathbf{B}^T]^{-1} \mathbf{B}^T \hat{\mathbf{P}} \mathbf{A} \quad (11-80)$$

### EXAMPLE 11.4

The nonrecursive method of solution of the optimal control problem will be illustrated using the first-order system of Example 11.1. We wish to design an optimal control law for the system

$$x(k+1) = 2x(k) + u(k)$$

with the cost function

$$J_N = \sum_{k=0}^N x^2(k) + u^2(k)$$

Thus the required parameters are

$$\begin{aligned} A &= 2 & Q &= 1 \\ B &= 1 & R &= 1 \end{aligned}$$

and

$$R_C = \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T = 1$$

Then in (11-49),

$$\mathbf{H} = \begin{bmatrix} A^{-1} & A^{-1} R_C \\ Q A^{-1} & A^T + Q A^{-1} R_C \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 2.5 \end{bmatrix}$$

The eigenvalues of  $\mathbf{H}$  satisfy the equation

$$|\lambda \mathbf{I} - \mathbf{H}| = (\lambda - 0.5)(\lambda - 2.5) - 0.25 = \lambda^2 - 3\lambda + 1 = 0$$

Thus the eigenvalues are 2.618 and 0.382. The eigenvectors satisfy the equation

$$\mathbf{H} \mathbf{h} = \lambda \mathbf{h}$$

Thus the eigenvectors are

$$\mathbf{h}_1 = \begin{bmatrix} 1 \\ 4.237 \end{bmatrix}, \quad \mathbf{h}_2 = \begin{bmatrix} 1 \\ -0.236 \end{bmatrix}$$

and the similarity transformation  $\mathbf{W}$  is

$$\mathbf{W} = [\mathbf{h}_1 \quad \mathbf{h}_2] = \begin{bmatrix} 1 & 1 \\ 4.237 & -0.236 \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$$

Then, from (11-59),

$$\begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda^{-1} \end{bmatrix} = \mathbf{W}^{-1}\mathbf{H}\mathbf{W} = \begin{bmatrix} 2.618 & 0 \\ 0 & 0.382 \end{bmatrix}$$

From (11-62),

$$T = -[W_{22} - QW_{12}]^{-1}[W_{21} - QW_{11}] = 2.619$$

and from (11-65),

$$G(j) = \Lambda^{-j}T\Lambda^{-j} = 2.619(0.382)^{2j}$$

From (11-71) and (11-72),

$$P(k+1) = M(N-k-1) = \frac{4.237 - 0.618(0.382)^{2(N-k-1)}}{1 + 2.619(0.382)^{2(N-k-1)}}$$

and the optimal gain, from (11-74), is

$$K(k) = \frac{2M(N-k-1)}{1 + M(N-k-1)}$$

For  $N = 2$  and  $k = 1$ ,

$$K(1) = \frac{2M(0)}{1 + M(0)} = \frac{2(1)}{1 + 1} = 1$$

and for  $k = 0$ ,

$$K(0) = \frac{2M(1)}{1 + M(1)} = \frac{2(3)}{1 + 3} = 1.5$$

These values check those of Example 11.1. The steady-state value,  $\hat{P}$ , is obtained from (11-79).

$$\hat{P} = W_{21}W_{11}^{-1} = 4.237$$

and the steady-state gain is

$$\hat{K} = \frac{2(4.237)}{1 + 4.237} = 1.618$$

Some general results regarding the steady-state solution of the discrete Riccati equation, (11-47), will now be given. Note that the steady-solution of this equation is the solution with  $N$  finite and  $m \rightarrow \infty$ .

First we define the term *stabilizable*.

**DEFINITION** The discrete-time system (11-4) is said to be stabilizable if there exists a matrix  $\mathbf{K}$  such that the eigenvalues of  $\mathbf{A} - \mathbf{BK}$  are all inside the unit circle.

Then we may state the following theorem.

**THEOREM 1** If the system (11-4) is either controllable or stabilizable, the discrete Riccati equation (11-47) has a limiting solution as  $m \rightarrow \infty$ ; that is,

$$\lim_{m \rightarrow \infty} \mathbf{P}(N - m) = \hat{\mathbf{P}}$$

as  $k \rightarrow -\infty$ . This theorem is given without proof, as is the following theorem [4].

**THEOREM 2** If the system (11-4) is observable, then  $\hat{\mathbf{P}}$  is positive definite and the optimal closed-loop system is asymptotically stable.

## 11.7 OPTIMAL STATE ESTIMATION—KALMAN FILTERS

In this section we extend the state-estimation concepts introduced in Chapter 9. We consider an optimal state estimation technique called Kalman filtering [5, 6]. The Kalman filter presented in this section has the equations of the current observer of Section 9.5; the only difference is that the gain matrix  $\mathbf{G}$  is calculated by a different procedure. There is also a Kalman filter in the form of the prediction observer [7]; we will not consider that form. In the following, the reader is assumed to have some understanding of random variables and stochastic processes [5, 6, 8].

The plant is assumed to be described by the equations

$$\begin{aligned} \mathbf{x}(k + 1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B}_1\mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \end{aligned} \quad (11-81)$$

where  $\mathbf{x}(k)[n \times 1]$  are the states,  $\mathbf{u}(k)[r \times 1]$  are the known inputs,  $\mathbf{w}(k)[s \times 1]$  are the random plant disturbances,  $\mathbf{y}(k)[p \times 1]$  are the measurements, and  $\mathbf{v}(k)[p \times 1]$  are the random inaccuracies in the measurements. The random inputs  $\mathbf{w}(k)$  and  $\mathbf{v}(k)$  in (11-81) are assumed to be uncorrelated and to have Gaussian distributions with the properties:

$$\begin{aligned} E[\mathbf{w}(k)] &= 0, & E[\mathbf{v}(k)] &= 0 \\ \text{cov}[\mathbf{w}(j), \mathbf{w}(k)] &= E[\mathbf{w}(j)\mathbf{w}^T(k)] = \mathbf{R}_w \delta_{jk} \\ \text{cov}[\mathbf{v}(j), \mathbf{v}(k)] &= E[\mathbf{v}(j)\mathbf{v}^T(k)] = \mathbf{R}_v \delta_{jk} \end{aligned} \quad (11-82)$$

$E[\cdot]$  is the mathematical expectation operation, and  $\text{cov}[\cdot]$  denotes the covariance. In (11-82),  $\delta_{jk}$  is the Kronecker delta function, defined by

$$\delta_{jk} = \begin{cases} 0, & j \neq k \\ 1, & j = k \end{cases}$$

Hence  $\mathbf{w}(k)$  and  $\mathbf{v}(k)$  are discrete white-noise sequences with Gaussian distributions.

We will not derive the equations for the Kalman filter; the derivation requires an extensive background in stochastic processes, which is beyond the scope of this book. However, the background acquired in study of observers in Chapter 9 will help to understand the purpose and the properties of the Kalman filter. The mathematical expectation in (11-82) can be considered to be



an averaging operation. Hence the expected value of a vector or a matrix can be viewed as the vector or matrix made up of the average values of the elements.

The estimates of the states  $\mathbf{x}(k)$  are denoted as  $\mathbf{q}(k)$ , and the errors of estimation are

$$\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{q}(k) \quad (11-83)$$

The covariance of the error vector is denoted as  $\mathbf{P}(k)[n \times n]$ , with

$$\mathbf{P}(k) = E[\mathbf{e}(k)\mathbf{e}^T(k)] \quad (11-84)$$

Hence the diagonal elements of  $\mathbf{P}(k)$  are the average squared errors of the estimation. The cost function for the minimization process is chosen as the trace of  $\mathbf{P}(k)$  (the sum of the diagonal elements or the sum of the average squared errors of estimation).

$$\begin{aligned} J(k) &= \text{tr}\mathbf{P}(k) = E[e_1^2(k)] + E[e_2^2(k)] + \cdots + E[e_n^2(k)] \\ &= \sigma_{e_1}^2(k) + \sigma_{e_2}^2(k) + \cdots + \sigma_{e_n}^2(k) \end{aligned} \quad (11-85)$$

where  $\sigma_{e_i}^2(k)$  is the variance of  $e_i(k)$ . This cost function can also be expressed as

$$J(k) = E[\mathbf{e}^T(k)\mathbf{e}(k)]$$

Note the *expected values* of the squared estimation errors are minimized, and not of the actual errors themselves. Since the system (11-81) has random inputs and the measurements have random errors, the actual estimation errors can never be determined. Hence we are forced to consider only the statistical characteristics of the estimation errors, and minimize a function of the expected values (average values) of these errors. Hence the Kalman filter is optimal only on the average.

An additional property of the Kalman filter is that the cost function

$$J(k) = \text{tr} E[\mathbf{e}(k)\mathbf{Q}\mathbf{e}^T(k)] \quad (11-86)$$

is also minimized, for  $\mathbf{Q}$  any positive semidefinite matrix [5]. Hence, for example, for a second-order system, the cost functions

$$\begin{aligned} J_1'(k) &= E[e_1^2(k)] \\ J_2'(k) &= E[e_1^2(k)] + E[e_2^2(k)] \\ J_3'(k) &= 10E[e_1^2(k)] + E[e_2^2(k)] \end{aligned}$$

are *all* minimized. In addition, the sum

$$J_N = \sum_{k=0}^N E[e_1^2(k)] + E[e_2^2(k)] = \sum_{k=0}^N E[\mathbf{e}^T(k)\mathbf{e}(k)]$$

is minimized, since the individual terms are minimized. Hence we see that the Kalman filter is optimal in many different ways.

As stated above, the mathematical development required to minimize (11-85) [or (11-86)] will not be given here; however, the resulting Kalman filter equations are [5, 6]

$$\begin{aligned}\mathbf{q}(k) &= \bar{\mathbf{q}}(k) + \mathbf{G}(k)[\mathbf{y}(k) - \mathbf{C}\bar{\mathbf{q}}(k)] \\ \bar{\mathbf{q}}(k+1) &= \mathbf{A}\mathbf{q}(k) + \mathbf{B}\mathbf{u}(k)\end{aligned}\quad (11-87)$$

In these equations,  $\bar{\mathbf{q}}(k)$  is the predicted state estimate at the sampling instant  $k$ , and  $\mathbf{q}(k)$  is the actual state estimate at  $k$ . The gain matrix  $\mathbf{G}(k)$  is the Kalman gain, and is calculated from the covariance equations

$$\begin{aligned}\mathbf{G}(k) &= \mathbf{M}(k)\mathbf{C}^T[\mathbf{C}\mathbf{M}(k)\mathbf{C}^T + \mathbf{R}_v]^{-1} \\ \mathbf{P}(k) &= \mathbf{M}(k) - \mathbf{G}(k)\mathbf{C}\mathbf{M}(k) \\ \mathbf{M}(k+1) &= \mathbf{A}\mathbf{P}(k)\mathbf{A}^T + \mathbf{B}_1\mathbf{R}_w\mathbf{B}_1^T\end{aligned}\quad (11-88)$$

In these equations,  $\mathbf{M}(k)$  is the covariance of the prediction errors:

$$\mathbf{M}(k) = E\{[\mathbf{x}(k) - \bar{\mathbf{q}}(k)][\mathbf{x}(k) - \bar{\mathbf{q}}(k)]^T\} \quad (11-89)$$

$\mathbf{P}(k)$  is defined in (11-84). The second equation in (11-88) can be substituted into the third equation in (11-88), and the result can be expressed as a single difference equation in  $\mathbf{M}(k)$ .

$$\mathbf{M}(k+1) = \mathbf{A}[\mathbf{I} - \mathbf{G}(k)\mathbf{C}]\mathbf{M}(k)\mathbf{A}^T + \mathbf{B}_1\mathbf{R}_w\mathbf{B}_1^T \quad (11-90)$$

The Kalman gains  $\mathbf{G}(k)$  can be precalculated, since they are independent of the measurements. Of course, the gains may also be calculated in real time. In either case, it is necessary to estimate both  $\bar{\mathbf{q}}(0)$ , the initial predicted state, and  $\mathbf{M}(0)$ , the covariance of the errors in  $\bar{\mathbf{q}}(0)$ . Once  $\mathbf{M}(0)$  is estimated, the Kalman gains  $\mathbf{G}(k)$  can be calculated from (11-88), using  $\mathbf{R}_w$  and  $\mathbf{R}_v$ , from (11-82). The Kalman filter equations (11-87) can then be solved in real time, using the gains  $\mathbf{G}(k)$ , the initial estimate  $\bar{\mathbf{q}}(0)$ , and the measurements  $\mathbf{y}(k)$ .

In the filter equations (11-87), the predicted estimate  $\bar{\mathbf{q}}(k)$  can be eliminated, resulting in the filter equation

$$\mathbf{q}(k) = [\mathbf{A} - \mathbf{G}(k)\mathbf{C}]\mathbf{q}(k-1) + [\mathbf{B} - \mathbf{G}(k)\mathbf{C}\mathbf{B}]\mathbf{u}(k-1) + \mathbf{G}(k)\mathbf{y}(k) \quad (11-91)$$

For the steady-state Kalman filter,  $\mathbf{G}(k)$  becomes constant. For this case, the filter equations of (11-87) [or (11-91)] have the characteristic equation

$$|z\mathbf{I} - (\mathbf{A} - \mathbf{G}\mathbf{C})| = 0 \quad (11-92)$$

If the gain  $\mathbf{G}$  is determined by some procedure other than (11-88), the filter is called the *current estimator*, as discussed in Section 9.6.

Suppose that the control system is designed by the linear-quadratic optimal procedure of the first part of this chapter, and the steady-state gains  $\mathbf{K}$  are used. Suppose that, in addition, the steady-state Kalman filter is employed to estimate the states. This design is called the *infinite-horizon linear-quadratic-Gaussian* (IH-LQG) design. The *implementation* is identical to that for the pole-placement current observer design of Chapter 9. Hence the control-estimator transfer function,  $D_{ce}(z)$ , is given by (see Fig. 9-8)

$$[\text{eq. (9-73)}] \quad D_{ce}(z) = z\mathbf{K}[z\mathbf{I} - \mathbf{A} + \mathbf{G}\mathbf{C} + \mathbf{B}\mathbf{K} - \mathbf{G}\mathbf{C}\mathbf{B}\mathbf{K}]^{-1}\mathbf{G}$$

Implementation of full-state feedback control systems using Kalman filters as state estimators can also result in systems that are not robust, as discussed in Section 9.5 [9–11]. Hence the stability margins at the plant input must be checked, to ensure adequate relative stability.

### EXAMPLE 11.5

The plant of Example 11.2, a servomotor, will be used to illustrate a Kalman filter design. This plant was also used in several examples for pole-placement and observer design in Chapter 9. The plant equations are

$$\mathbf{x}(k + 1) = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.905 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.00484 \\ 0.0952 \end{bmatrix} u(k) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} w(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k) + v(k)$$

Recall that the two plant states are motor shaft position and shaft velocity. We will assume that the units of the states are degrees of rotation and degrees/second, respectively. The plant disturbance,  $w(k)$ , affects both states equally. We will assume that the input covariances in (11-82) are

$$E[w(j)w(k)] = R_w \delta_{jk}, \quad R_w = 1 = \sigma_w^2$$

$$E[v(j)v(k)] = R_v \delta_{jk}, \quad R_v = 1 = \sigma_v^2$$

where  $\sigma$  denotes the standard deviation. For a Gaussian-distributed random variable  $a$  with zero mean and variance of  $\sigma^2$  [6],

$$\text{prob}[|a| \leq \sigma] = 0.683$$

$$\text{prob}[|a| \leq 2\sigma] = 0.955$$

$$\text{prob}[|a| \leq 3\sigma] = 0.997$$

For example, for the measurement inaccuracies, we are saying that 68 percent of the measurements are accurate to within  $1^\circ$  of shaft rotation, and in only 3 of 1000 measurements will the error be greater than  $3^\circ$ , on the average. Hence the probability of a Gaussian random variable with a value within the three-sigma ( $3\sigma$ ) value of its mean is almost a certainty. Also, the same numbers apply to the plant disturbance.

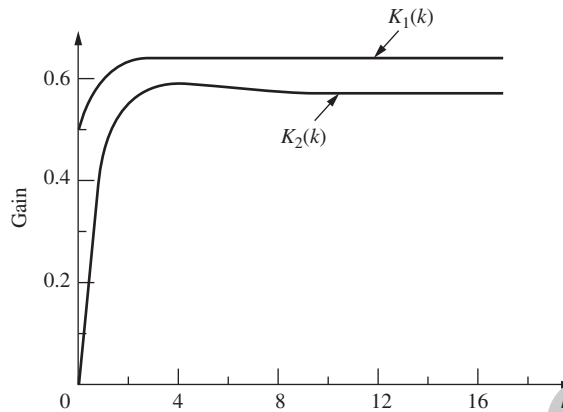
Because of the complexity of the equations, the Kalman gains for this example were calculated by computer, and the results are plotted in Fig. 11-5.

The initial value of the covariance matrix,  $\mathbf{M}(0)$ , was assumed to be the identity matrix. The gains attain the steady-state values in approximately 15 iterations, or in 1.5 s. The steady-state gains are

$$\mathbf{G} = \begin{bmatrix} 0.636 \\ 0.570 \end{bmatrix}$$

and the Kalman filter equations in (11-87) are completely specified. These equations can now be solved, once the initial estimate of the states,  $\bar{\mathbf{q}}(0)$ , is made. The steady-state error covariance matrix is calculated to be

$$\mathbf{P} = \begin{bmatrix} 0.636 & 0.570 \\ 0.570 & 0.599 \end{bmatrix}$$



**FIGURE 11-5** Kalman gain versus sample period.

Hence for the steady-state Kalman filter, the standard deviation in the estimation of  $x_1(k)$  is  $\sigma_{e1} = (0.636)^{1/2} = 0.797^\circ$  and  $\sigma_{e2} = 0.771^\circ \text{ s}^{-1}$ . For the estimate of the motor shaft position, the error is almost certain to be within the three-sigma value of approximately  $2.4^\circ$ . For example, if the measurement is  $7^\circ$ , the true position of the motor shaft is almost certain to be between  $4.6^\circ$  and  $9.4^\circ$ . If this error is too large, either more accurate instrumentation must be used, or the disturbances on the plant must be reduced. Recall that the filter itself is “best” for the error criteria used. A MATLAB program that calculates the Kalman gains is given by

```
format ShortEng
A = [1 0.0952; 0 0.905]; B1 = [1;1]; C = [1 0];
Rw = 1; Rv = 1; M = [1 0; 0 1]; N = 15;
disp('k      Gains')
for k=1:N
G = M*C'*inv(C*M*C'+Rv);
P = M - G*C*M;
M = A*P*A'+B1*Rw*B1';
[k,G']
end
disp('The final value of the P matrix is:')
P
```

### EXAMPLE 11.6

The design of Example 11.5 will be extended to a steady-state Kalman filter and an infinite-horizon linear-quadratic-Gaussian (IH-LQG) control system design. For the IH-LQG design, the optimal control design of Example 11.3 is used, where

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad R = 1$$

The steady-state gains from this example are

$$\mathbf{K} = [0.964 \quad 0.719]$$

Hence the IH-LQG controller can be represented by the transfer function (9-73), and evaluation of this transfer function yields

$$D_{ce}(z) = \frac{1.023z^2 - 0.9124z}{z^2 - 1.147z + 0.3061}$$

The plant transfer function is  $G(z)$ , and the system open-loop function, for the system opened at the plant input, is

$$D_{ce}(z)G(z) = \frac{1.023z^2 - 0.9124z}{z^2 - 1.147z + 0.3061} \left( \frac{0.00484z + 0.004683}{z^2 - 1.905z + 0.905} \right)$$

(See Fig. 9-8). Calculation of the frequency response of this function yields a phase margin of  $66.7^\circ$  and a gain margin of 18.4 dB. Hence the system has adequate stability margins. It is *always* necessary to check stability margins, since nothing in the design assures adequate margins. A MATLAB program for this example follows.

```
A = [1 0.0952; 0 0.905]; B = [0.00484; 0.0952]; C = [1 0];
K = [0.964 0.719]; G = [0.636; 0.570];
syms z, T = 0.1;
Dce = z*K*inv((z*eye(2) - A + G*C*A + B*K - G*C*B*K)) * G;
[n,d] = nice(Dce,z); Dce = tf(n,d,T)
Sys=ss(A,B,C,0,T); [num,den,T]=tfdata(Sys);
Gz = tf(num,den,T)
Tz=feedback(Dce*Gz,1);
[Gm,Pm] = margin(Dce*Gz); GmDb = 20*log10(Gm), Pm
Stepinfo = stepinfo(Tz)
```

It is informative to consider the IH-LQG design as depicted in Fig. 11-6. In this figure all signals shown are vectors; hence the order of multiplication is important. Any simplification on the block diagram must be performed with care. This block diagram can be derived from (11-87), if  $k$  is replaced with  $(k + 1)$ . First, the second equation of (11-87) is substituted in the first one.

$$\mathbf{q}(k + 1) = \mathbf{A}\mathbf{q}(k) + [\mathbf{B} - \mathbf{G}\mathbf{C}\mathbf{B}]\mathbf{u}(k) + \mathbf{G}\mathbf{y}(k + 1) - \mathbf{G}\mathbf{C}\mathbf{A}\mathbf{q}(k)$$

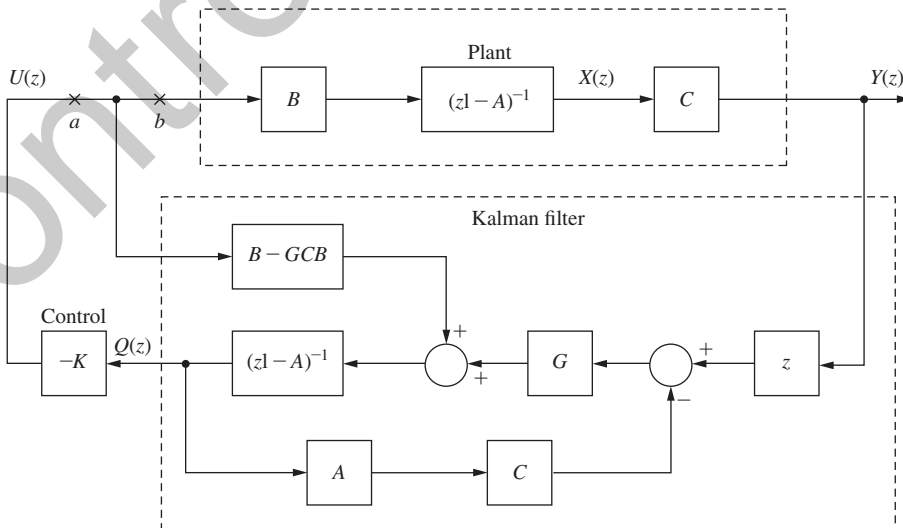


FIGURE 11-6 IH-LQG control system.

Taking the  $z$ -transform of this equation and rearranging yields

$$(z\mathbf{I} - \mathbf{A})\mathbf{Q}(z) = [\mathbf{B} - \mathbf{GCB}]\mathbf{U}(z) + z\mathbf{G}\mathbf{Y}(z) - \mathbf{GCAQ}(z)$$

Hence

$$\mathbf{Q}(z) = (z\mathbf{I} - \mathbf{A})^{-1}\{[\mathbf{B} - \mathbf{GCB}]\mathbf{U}(z) + z\mathbf{G}\mathbf{Y}(z) - \mathbf{GCAQ}(z)\}$$

This is seen to be the equation for  $\mathbf{Q}(z)$  at the output of the Kalman filter in Fig. 11-6.

The IH-LQG control system as represented in Fig. 11-6 is important in understanding the effects of the Kalman filter on the control system. It is seen that the Kalman filter has two inputs. One input is the plant input  $\mathbf{U}(z)$  and the other is the plant output  $\mathbf{Y}(z)$ . If the measurements are specified as being very noisy [the effects of the measurement noise on the estimation is much greater than the effects of the plant disturbances], the resulting value of  $\mathbf{G}$  will reduce the effects of the measurements on the state estimates, relative to the effects of the plant input and the plant dynamics. For example, if there are no plant disturbances ( $\mathbf{R}_w = \mathbf{0}$ ), the steady-state Kalman filter will completely ignore the measurements ( $\mathbf{G} = \mathbf{0}$ ) [5]. Of course, this is not desirable under any conditions.

Conversely, if the disturbances on the plant are large, the resulting value of  $\mathbf{G}$  will increase the effects of the measurements on the state estimates, and  $\mathbf{U}(z)$  and the plant dynamics will have less effect. In some cases  $\mathbf{R}_w$  is purposely increased to indicate model uncertainty.

It is shown in [9] that the stability margins for the system in Fig. 11-6 when opened at point  $a$  are always satisfactory. However, the stability margins for the system opened at point  $b$  may be quite small. Since the plant model is always suspect, we require large stability margins at point  $b$ .

A procedure is given in [9] for increasing the stability margins at point  $b$ , if these margins are small. This is accomplished by increasing the covariance matrix  $\mathbf{R}_w$  relative to its specified value; however, this increase must be performed in a prescribed manner. The stability margins can be increased, but the resulting Kalman filter is no longer optimal for the original specifications. Hence it is necessary to trade-off optimality for stability. Trade-offs of this type are not unusual in applying any theory to physical systems.

## 11.8 LEAST-SQUARES MINIMIZATION

The optimal procedures presented in Chapters 10 and 11 are all based on least-squares minimization, and hence are related. To show the similarities of the procedures, the equations are listed in Table 11-1 for LQ design, least-squares system identification, and Kalman-filter design. The similarities are evident.

## 11.9 SUMMARY

Presented in this chapter were some basic results in linear quadratic optimal control design. Once the cost function has been chosen, the design involves the straightforward solution of a difference equation. Even though the basic formulation is for a finite-time problem, the design procedure is easily extended to the infinite-time problem. The design implementation requires full-state feedback, with the feedback gains time varying for the finite-time problem and constant for the infinite-time problem, provided that the plant is time invariant.

**TABLE 11.1 Least-Squares Design Equations**

Cost Functions	
LQ	$J_N = \sum_{k=0}^N [\mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}\mathbf{u}(k)]$
Sys. Id.	$J = \sum_{k=n}^N e^2(k) = \mathbf{e}^T(k)\mathbf{e}(k)$
Kalman	$J(k) = \sum_{i=0}^N E[e_i^2(k)] = E[\mathbf{e}^T(k)\mathbf{e}(k)]$
Gain Equations	
LQ	$\mathbf{K}(k) = [\mathbf{R} + \mathbf{B}^T\mathbf{P}(k+1)\mathbf{B}]^{-1}\mathbf{B}^T\mathbf{P}(k+1)\mathbf{A}$
Sys. Id.	$\mathbf{L}(k+1) = \mathbf{P}(k)\mathbf{f}(k+1)[\mathbf{I} + \mathbf{f}^T(k+1)\mathbf{P}(k)\mathbf{f}(k+1)]^{-1}$
Kalman	$\mathbf{G}(k) = \mathbf{M}(k)\mathbf{C}^T[\mathbf{R}_v + \mathbf{C}\mathbf{M}(k)\mathbf{C}^T]^{-1}$
Update Equations	
LQ	$\mathbf{P}(k) = \mathbf{A}^T\mathbf{P}(k+1)[\mathbf{A} - \mathbf{B}\mathbf{K}(k)] + \mathbf{Q}$
Sys. Id.	$\mathbf{P}(k+1) = [\mathbf{I} - \mathbf{L}(k+1)\mathbf{f}^T(k+1)]\mathbf{P}(k)$
Kalman	$\mathbf{M}(k+1) = \mathbf{A}[\mathbf{I} - \mathbf{G}(k)\mathbf{C}]\mathbf{M}(k)\mathbf{A}^T + \mathbf{B}_1\mathbf{R}_v\mathbf{B}_1^T$

As the final topic, optimal state estimation by Kalman filtering was presented. This filter is a form of the current observer developed in Chapter 9; the only difference is the procedure for calculating the gain matrix  $\mathbf{G}$ . The infinite-horizon linear-quadratic-Gaussian (IH-LQG) design was then described.

## References and Further Readings

- [1] P. M. De Russo, R. J. Roy, C. M. Close, and A. A. Desrochers, *State Variables for Engineers*, 2d ed. New York: John Wiley & Sons, Inc., 1998.
- [2] R. Bellman, *Adaptive Control Process: A Guided Tour*. Princeton, NJ: Princeton University Press, 1961.  
*IEEE Trans. Autom. Control*, Bellman Special Issue, AC-26, 1981.
- [3] P. Dorato and A. H. Levis, "Optimal Linear Regulators: The Discrete Time Case," *IEEE Trans. Autom. Control* AC-16, pp. 613–20, 1971.
- [4] P. S. Maybeck, *Stochastic Models, Estimation and Control*, Vol. 3. New York: Academic Press, Inc., 1982.
- [5] R. G. Brown, *Introduction to Random Signal Analysis and Kalman Filtering*. New York: John Wiley & Sons, Inc., 1983.
- [6] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice Hall, 1979.
- [7] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 2nd ed. New York: McGraw-Hill Book Company, 1984.
- [8] J. C. Doyle and G. Stein, "Robustness with Observers," *IEEE Trans. Automat. Control* AC-24, pp. 607–611, 1979.
- [9] B. T. Oranc, "A Classical Approach to Robust Design of Linear-Quadratic-Gaussian Controllers," Ph.D. dissertation, Auburn University, Auburn, AL, 1987.
- [10] B. E. Sturgis, Jr., "A Study of Stability Margin Problems in Systems Implementing Kalman Filters," M.S. thesis, Auburn University, Auburn, AL, 1984.

- [11] T. E. Fortman, "A Matrix Inversion Identity," *IEEE Trans. Autom. Control* AC-15, p. 599, 1970.
- [12] R. Gran and F. Kozin, *Applied Digital Control Systems*. George Washington University Short Course Notes, Washington, D.C., 1979.
- [13] D. R. Vaughan, "A Nonrecursive Algebraic Solution for the Discrete Riccati Equation," *IEEE Trans. Autom. Control* AC-15, pp. 597–599, 1970.

## Problems

11.4-1. Given that (11-21) and (11-22) are valid, derive (11-23).

11.4-2. Show that (11-35) can also be expressed as

$$\mathbf{P}(N - m) = \mathbf{A}^T[\mathbf{P}(N - m + 1) - \mathbf{P}(N - m + 1)\mathbf{B}\mathbf{D}\mathbf{B}^T\mathbf{P}(N - m + 1)]\mathbf{A} + \mathbf{Q}$$

where  $\mathbf{D} = [\mathbf{B}^T\mathbf{P}(N - m + 1)\mathbf{B} + \mathbf{R}]^{-1}$ .

11.4-3. Given the discrete system

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)$$

with the cost function

$$J_N = \sum_{k=0}^N \mathbf{x}^T(k)\mathbf{Q}\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}\mathbf{u}(k)$$

show that the optimal gains which minimize  $J_N$  are unchanged if the elements both in  $\mathbf{Q}$  and in  $\mathbf{R}$  are multiplied by the positive scalar  $\beta$ .

11.4-4. Given the first-order plant described by

$$x(k + 1) = 0.6x(k) + 0.5u(k)$$

with the cost function

$$J_3 = \sum_{k=0}^3 [2x^2(k) + 7u^2(k)]$$

- (a) Calculate the feedback gains required to minimize the cost function, using the partial-differentiation procedure of Section 11.3.
- (b) Repeat part (a) using the difference-equation approach of Section 11.4.
- (c) Find the maximum magnitude of  $u(k)$  as a function of  $x(0)$ .

11.4-5. Given the plant of Problem 11.4-4, with the cost function

$$J_3 = \sum_{k=0}^3 2x^2(k)$$

- (a) Calculate the feedback gains required to minimize the cost function, using the difference-equation approach of Section 11.4.
- (b) Find the maximum magnitude of  $u(k)$  as a function of  $x(0)$ .
- (c) Compare the maximum magnitude of  $u(k)$  in part (b) with the value in Problem 11.4-4(c). Explain the difference.

11.4-6. A satellite control system is modeled as shown in Fig. P11.4-6. This system is described in Problem 1.4-1. For this problem, ignore the sensor gain and let  $D(z) = 1$ . In addition,  $K = 1$ ,  $T = 1$  s, and  $J = 4$ . As stated in Problem 9.2-4, a state model for this system is given by

$$\mathbf{x}(k + 1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.125 \\ 0.25 \end{bmatrix} u(k)$$

$$y(k) = [1 \quad 0]\mathbf{x}(k)$$

where  $x_1(k)$  is angular position and  $x_2(k)$  is angular velocity.

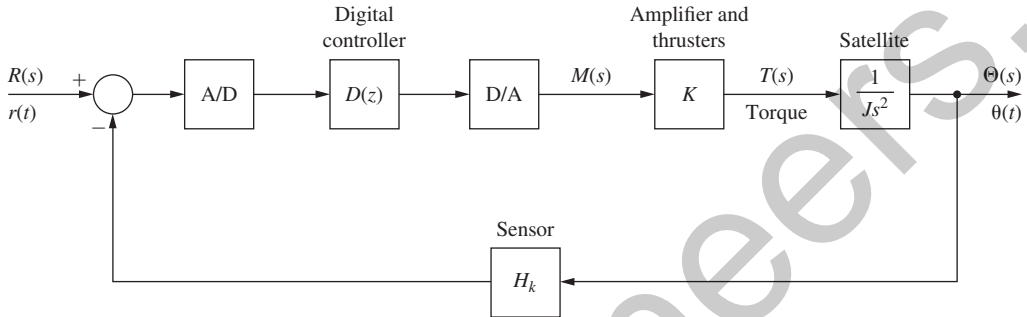


- (a) Determine by hand the gains required to minimize the cost function

$$J_N = \sum_{k=0}^N \mathbf{x}^T(k) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + 2u^2(k)$$

with  $N = 1$ . The value of  $N$  is chosen to be unity to limit the calculations.

- (b) Use MATLAB to solve part (a) for  $N = 20$ . Sketch the calculated gains versus  $k$ .



**FIGURE P11.4-6** Block diagram for a satellite control system.

- 11.4-7.** (a) Repeat Problem 11.4-6 for minimizing the cost function

$$J_N = \sum_{k=0}^N \mathbf{x}^T(k) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + 20u^2(k)$$

Plot the gains on the graph from Problem 11.4-6(c)

- (b) Explain the differences in the gains in part (a) and those in Problem 11.4-6.

- 11.4-8.** Consider the third-order continuous-time LTI system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

$$y = \mathbf{C}\mathbf{x}$$

with  $\mathbf{A} = \begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 3 \\ 0 & -8 & -6 \end{bmatrix}$ ,  $\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ , and  $\mathbf{C} = [1 \ 0 \ 0]$ . Using

$$\mathbf{Q} = \begin{bmatrix} 8 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 4 \end{bmatrix}, \quad R = 1.5$$

- (a) First design a LQ controller for this continuous time-system using the MATLAB function *lqr*. Let the optimal controller gain vector be  $\mathbf{K}$ . Simulate the closed-loop system

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BK})\mathbf{x}$$

with  $\mathbf{X}(0) = [2 \ 0 \ -2]$ . Sample the closed-loop output response  $y(t)$  with sampling time  $T = 0.2$  s.

- (b) Next, discretize the continuous-time system directly with sampling time  $T = 0.2$  s and design a LQ controller for this discrete-time LTI system with the same  $\mathbf{Q}$  and  $R$  as in part (a). Compare the closed-loop output response with the sampled continuous-time output obtained in part (a).

- (c) Increase  $\mathbf{Q}$  and  $R$  by a factor of 10 and repeat both the continuous-time and discrete-time designs. Describe how the respective output responses change.

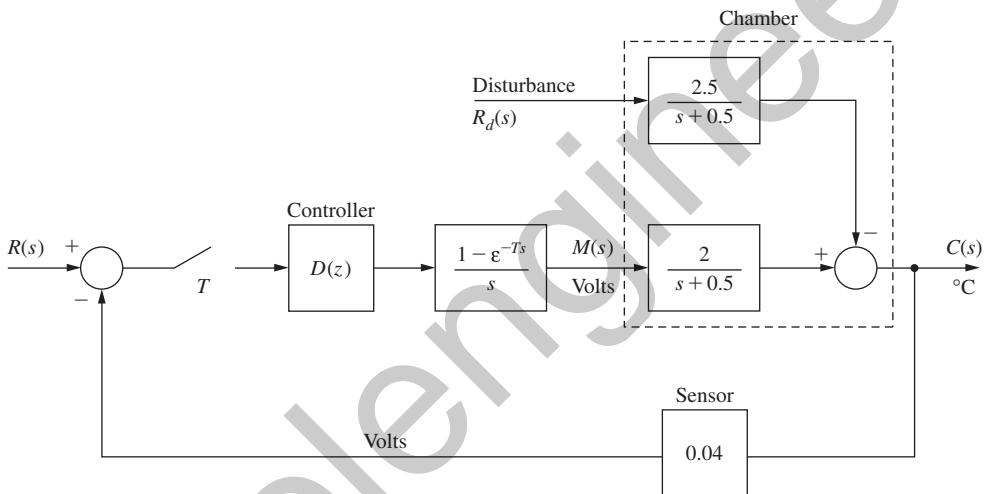
**11.4-9.** A chamber temperature control system is modeled as shown in Fig. P11.4-8. This system is described in Problem 1.6-1. For this problem, ignore the disturbance input,  $T = 0.6$  s and let  $D(z) = 1$ . It was shown in Problem 9.2-3 that the plant state model is given by

$$x(k + 1) = 0.7408x(k) + 1.0368u(k)$$

Let the cost function be given by

$$J_3 = \sum_{k=0}^3 [2x^2(k) + u^2(k)]$$

- Calculate the feedback gains required to minimize the cost function, using the partial-differentiation procedure of Section 11.3.
- Repeat part (a), using the difference-equation approach of Section 11.4.
- Find the maximum magnitude of  $u(k)$  as a function of  $x(0)$ .



**FIGURE P11.4-8** Chamber temperature control system.

**11.4-10.** Given the plant of Problem 11.4-9, with the cost function

$$J_3 = \sum_{k=0}^3 10x^2(k) + u^2(k)$$

- Calculate the feedback gains required to minimize the cost function, using the difference-equation approach of Section 11.4.
- Find the maximum magnitude of  $u(k)$  as a function of  $x(0)$ .
- Compare the maximum magnitude of  $u(k)$  in part (b) with the value in Problem 11.4-9(c). Explain the difference.

**11.4-11.** Given a general first-order plant described by

$$x(k + 1) = Ax(k) + Bu(k)$$

with the cost function

$$J_N = \sum_{k=0}^N [Qx^2(k) + Ru^2(k)]$$

- (a) Show that for  $R = 0$ , the optimal gain  $K(k)$  is constant for all  $k \geq 0$ .
- (b) Give the input sequence  $u(k)$ ,  $k \geq 0$ , for part (a), where  $u(k)$  is a function of  $x(0)$ .
- (c) Draw a flow graph of the closed-loop system for part (a).
- (d) Give the closed-loop characteristic equation for part (a).
- (e) How many sample periods are required for  $x(k)$ , with initial condition  $x(0)$ , to be driven to zero?

**11.6-1.** It is shown in [6] that given the partitioned matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{D} & \mathbf{E} \\ \mathbf{F} & \mathbf{G} \end{bmatrix}$$

where each partition is  $n \times n$ , the determinant of  $\mathbf{H}$  is given by

$$|\mathbf{H}| = |\mathbf{G}| |\mathbf{D} - \mathbf{E}\mathbf{G}^{-1}\mathbf{F}| = |\mathbf{D}| |\mathbf{G} - \mathbf{F}\mathbf{D}^{-1}\mathbf{E}|$$

Show that the determinant of  $\mathbf{H}$  in (11-49) is equal to unity. Considering the development in Section 11.6, is this result expected?

**11.6-2.** Give a first-order time-invariant discrete system with a cost function

$$J_N = \sum_{k=0}^N Qx^2(k) + Ru^2(k)$$

Show that the optimal gains are a function of only the ratio

$$\alpha = \frac{Q}{R}$$

and not of  $Q$  and  $R$  singly.

**11.6-3.** Consider the system of Problem 11.4-4.

- (a) Find the feedback gain required to minimize the given cost function for the infinite-time problem.
- (b) Find the closed-loop system characteristic equation.
- (c) Find the closed-loop time constant  $\tau$  as a function of the sample period  $T$ .

**11.6-4.** Consider the system of Problem 11.4-5.

- (a) Find the feedback gain required to minimize the given cost function for the infinite-time problem.
- (b) Find the closed-loop system characteristic equation.
- (c) Find the closed-loop time constant  $\tau$  as a function of the sample period  $T$ .
- (d) Why is the settling time in part (c) less than that found in Problem 11.4-6(c), where  $\tau = 0.410T$ .

**11.6-5.** Consider the satellite control system of Problem 11.4-6.

- (a) By computer, find the feedback gains required to minimize the given cost function for the infinite-time problem.
- (b) Find the closed-loop system characteristic equation.
- (c) Find the closed-loop time constants as a function of the sample period  $T$ .

**11.6-6.** Consider the satellite control system of Problem 11.4-7.

- (a) By computer, find the feedback gains required to minimize the given cost function for the infinite-time problem.
- (b) Find the closed-loop system characteristic equation.
- (c) Find the closed-loop time constants as a function of the sample period  $T$ .
- (d) Compare the time constants in part (c) with the value in Problem 11.6-5(c), which was  $\tau = 3.24T$ . Explain the difference.

**11.6-7.** Consider the system of Problem 11.4-9.

- (a) Find the feedback gain required to minimize the given cost function for the infinite-time problem.
- (b) Find the closed-loop system characteristic equation.
- (c) Find the closed-loop time constant.

**11.6-8.** Consider the system of Problem 11.4-10.

- Find the feedback gain required to minimize the given cost function for the infinite-time problem.
- Find the closed-loop system characteristic equation.
- Find the closed-loop time constant.
- Compare the gain in part (b) with the value in Problem 11.6-3(a). Explain the difference.

**11.7-1.** Suppose that a plant is described by

$$\begin{aligned}
 x(k+1) &= 0.8x(k) + 0.2u(k) + w(k), & T &= 0.2 \text{ s} \\
 y(k) &= x(k) + v(k)
 \end{aligned}$$

where  $w(k)$  and  $v(k)$  are random and uncorrelated, with Gaussian distributions, and

$$\begin{aligned}
 E[w(k)] &= 0, & E[w(j)w(k)] &= 2\delta_{ij} \\
 E[v(k)] &= 0, & E[v(j)v(k)] &= \delta_{ij}
 \end{aligned}$$

- Design a Kalman filter for this system. Continue the gain calculations until the gain is approximately constant. Use  $M(0) = 2$ .
- In part (a), we specified  $M(0) = 2$ . What are we stating about our estimate of the state  $x(0)$ ?
- Write the difference equations for the steady-state Kalman filter, as in part (a).
- Suppose that an LQ design is performed for this plant, with the resulting gain  $K = 0.2197$ . Find the control-estimator transfer function (see Fig. 9-8) for this IH-LQG design.
- Find the closed-loop system characteristic equation for part (d).
- Find the closed-loop system time constants.
- Suppose that the state  $x(k)$  is estimated to be 90.1 by the Kalman filter at a certain time  $kT$ . Give the three-sigma range about the value 90.1 that will almost certainly contain the true value of  $x(k)$  at that time  $kT$ .
- Find the system phase and gain margins.

**11.7-2.** The Kalman filter design in Problem 11.7-1 resulted in the steady-state filter equations

$$\begin{aligned}
 q(k) &= \bar{q}(k) + 0.7105[y(k) - \bar{q}(k)] \\
 \bar{q}(k+1) &= 0.8q(k) + 0.2u(k)
 \end{aligned}$$

- Consider the plant and filter to be open-loop; that is, the state estimate is not fed back for control purposes. Suppose that the input  $u(k)$  is constant at a value of 10. Find the steady-state values of the plant state  $x(k)$  and the plant output  $y(k)$ , if the random inputs  $w(k)$  and  $v(k)$  are zero.
- For the conditions specified in part (a), find the steady-state value of  $q(k)$ , the plant state estimate.
- The steady-state Kalman filter has the property that  $\mathbf{Q}(z)/\mathbf{U}(z) = \mathbf{X}(z)/\mathbf{U}(z)$ . Does this property verify your results in parts (a) and (b)?
- The Kalman filter requires that the average value of  $w(k)$  be zero. Suppose that  $w(k)$  is constant with a value of 5. Repeat parts (a) and (b), and calculate the percent error in the state estimate.

**11.7-3.** Consider the system of Problem 11.7-1.

- Suppose that all specifications are the same, except  $E[w(j)w(k)] = 0$ ; that is, there are no random plant disturbances. Design a Kalman filter and estimate the steady-state Kalman gain from the trends of the calculations. Use  $M(0) = 2$ .
- Write the difference for the steady-state Kalman filter.
- In part (a),  $P(k) \rightarrow 0$  in the steady-state. Hence the errors in the estimation of the state go to zero in the steady-state (perfect estimation). Since the measurements are noisy, how can the estimation error be zero?
- Repeat parts (a), (b), and (c) for the case that  $R_w = 2$  and  $R_v = 0$ . In repeating part (c), the measurements are perfect, but the plant has a random disturbance.

**11.7-4.** Consider the satellite described in Problem 11.4-8. The plant model is given as

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.111 \\ 0.11 \end{bmatrix} u(k) + \begin{bmatrix} 0.02 \\ 0.2 \end{bmatrix} w(k)$$

$$y(k) = [1 \quad 0] \mathbf{x}(k) + (k)$$

The plant disturbances are caused by random variations in the Earth's gravity field. Suppose that  $R_w = 1$  and  $R_v = 0.02$ .

- The measurement  $y(k)$  is in the units of angular degrees and is obtained from a stable platform. Describe the accuracy of this measurement; that is, what does the value of  $R_v$  tell us about the sensor accuracy?
- Calculate and plot the Kalman gains, as in Fig. 11-5, using  $\mathbf{M}(0) = \mathbf{I}$ .
- The diagonal elements of the steady-state error covariance matrix are

$$\mathbf{P}_{ss} = \begin{bmatrix} 0.00543 & - \\ - & 0.0286 \end{bmatrix}$$

Comment on the steady-state accuracy of this Kalman filter.

- The LQ design of Problem 11.4-8 yielded the steady-state gains of  $\mathbf{K} = [0.51922 \ 1.03]$ . Find the transfer function  $D_{ce}(z)$  of the control estimator.
  - Calculate and plot the Nyquist diagram for the closed-loop system opened at the plant input. What are the phase and gain margins?
- 11.7-5.** The system considered here is the classical system of Doyle and Stein [9] to illustrate robustness problems. The system of Doyle and Stein is analog; the discrete model of the system is used here [11]. The sample period,  $T = 0.006$  s, was chosen small so that the results approximate those of Doyle and Stein. The plant model is given by

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.999946 & 0.592847E-2 \\ -0.177854E-1 & 0.976233 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.178567E-4 \\ 0.592847E-2 \end{bmatrix} u(k) + \begin{bmatrix} 0.208907 \\ -0.383511 \end{bmatrix} w(k)$$

$$y(k) = [2 \quad 1] \mathbf{x}(k) + v(k)$$

- A pole-placement design is to yield closed-loop poles at  $s = -7 \pm j2$ . Show that these  $s$ -plane poles translate into the desired characteristic equation

$$\alpha_c(z) = z^2 - 1.917602z + 0.919431 = 0$$

- Find the gain matrix required to place the poles at the locations given in part (a).
  - Find the steady-state Kalman filter gains for the case that  $R_w = R_v = 166.67$ .
  - Find the plant transfer function  $G(z)$  and the control-estimator transfer function  $D_{ce}(z)$ .
  - Using the transfer functions of part (d), plot the Nyquist diagram for the system opened at the plant.
  - Find the system phase and gain margins. Doyle and Stein found the phase margin of the analog system to be approximately  $15^\circ$ .
- 11.7-6.** For the IH-LQG control system of Fig. 11-6, suppose that the plant is single-input single-output, such that  $\mathbf{B}$  and  $\mathbf{C}$  are vectors. To determine the system robustness, the open-loop transfer functions for the system opened at  $a$  and opened at  $b$  must be determined.
- Find the open-loop transfer function for the system opened at the point  $a$ .
  - Find the open-loop transfer function for the system opened at the point  $b$ .
  - Give the transfer functions for the frequency responses that must be calculated to determine robustness.

# 12

## Case Studies

### 12.1 INTRODUCTION

This chapter presents five case studies of digital control systems that have been implemented. Two of the control systems were designed using the frequency-response techniques of Chapter 8, and the design of the third, even though empirical, was based on the techniques of Chapter 8.

The first case study is a second-order position control system (servomotor) for which both a phase-lead and a phase-lag controller were designed. The responses are compared. Sampling rate selection is discussed, and certain significant effects from plant nonlinearities are noted.

The second system studied is an environmental chamber control system, which is composed of a temperature control system, a carbon dioxide control system, a chamber water-loss monitor, an outside rainfall monitor, and a data acquisition system. The temperature system controller is a PID compensator; the carbon dioxide system controller is a quasi-proportional compensator.

The third case study is the lateral control system of an automatic aircraft landing system for U.S. Marine fighter aircraft. The plant for this control system is the aircraft lateral dynamics, including the bank autopilot, and is ninth order. The digital controller generates bank commands into the autopilot, and the aircraft position is determined by a phased array radar. The system contains significant noise and disturbance inputs, which must be considered in the controller design. The digital controller is a PID compensator plus added filtering to reduce noise effects.

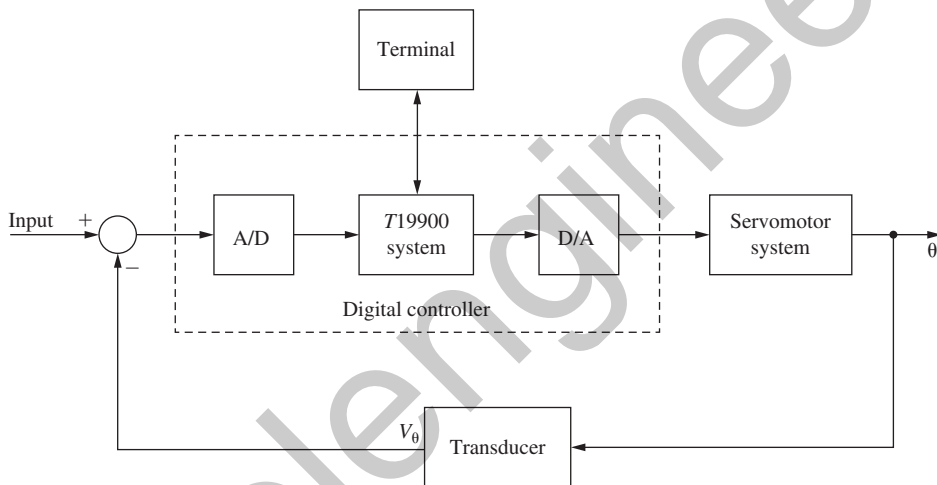
The fourth case study uses PID design methods to control oxygen delivery to a newborn infant being supported in an incubator. Various models of the system components are described, and several sampling rates are explored to assess their impact on system performance. MATLAB's *pidtool* is used for compensator design.

The fifth case study uses system identification to model a portion of a larger power system network. MATLAB's Power System Analysis Toolbox (PSAT) is used for system simulation, and CVX software package is used for convex optimization.

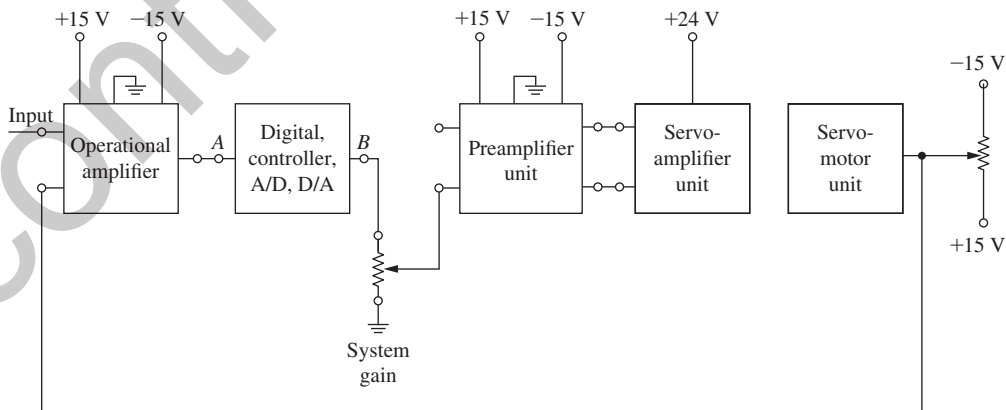
## 12.2 SERVMOTOR SYSTEM

The design of a digitally controlled servomotor system [1] is presented in this section. This system is low order and presents no particular design nor implementation difficulties. However, the system does contain nonlinearities that have an observable influence on system response. These nonlinear effects will be discussed as they are encountered.

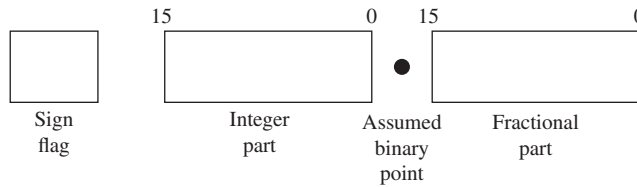
The control-system block diagram is shown in Fig. 12-1 and the system hardware configuration is given in Fig. 12-2. The Texas Instruments TI9900 microprocessor system [2] was chosen for the implementation of the digital controller. At the time of the system design, this microprocessor system was one of the few 16-bit processors available. In addition, the processor has hardware multiplication, and software support is available. The terminal indicated in Fig. 12-1 was chosen to be a Texas Instruments Microterminal [2] and is used to initialize the system, change filter parameters if desired, test system operation, and so on.



**FIGURE 12-1** System block diagram.



**FIGURE 12-2** System hardware.



**FIGURE 12-3** Computer data format.

The data (number) format for data internal to the computer is shown in Fig. 12-3 and is fixed-point format. The magnitudes of both the fractional part and the integer part of numbers are presented by 16 bits. Thus the dynamic range is from a (nonzero) minimum of [3]

$$2^{-16} = 0.000015$$

to a maximum of

$$(2^{16} - 1)_{\text{integer}} + (1 - 2^{-16})_{\text{fraction}} = 65,535.999985$$

The 32-bit precision is not needed for this application. It was chosen to facilitate data manipulations, since processing time is abundant. In addition, the sign flag was placed external to the integer part of the number to facilitate multiplication. The filter coefficients are stored in sign-magnitude format, which is the form required for the multiplication hardware.

### System Model

The control system can be mathematically modeled as shown in Fig. 12-4.

A first step in the design process was to determine the plant transfer function  $G_p(s)$ . This transfer function was obtained experimentally by removing the digital controller, the A/D, and the D/A shown in Fig. 12-2. Thus, in this figure, points *A* and *B* were connected. Since the servomotor is dc and armature-controlled, the plant transfer function was assumed to be [4]

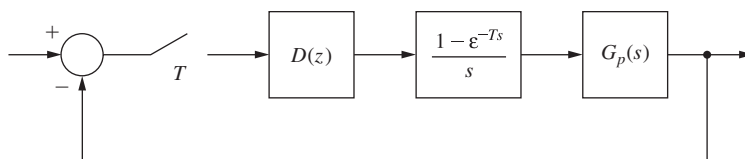
$$G_p(s) = \frac{\omega_n^2}{s(s + 2\delta\omega_n)} \tag{12-1}$$

giving a closed-loop transfer function of

$$T(s) = \frac{G_p(s)}{1 + G_p(s)} = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2} \tag{12-2}$$

Thus the system frequency response is given by

$$T(j\omega) = \frac{\omega_n^2}{(\omega_n^2 - \omega^2) + j2\delta\omega_n\omega} \tag{12-3}$$



**FIGURE 12-4** Control system model.



To obtain the system model experimentally, a sinusoid was applied to the system and the frequency varied until the system output lagged the input by  $90^\circ$  in phase. As seen from (12-3), this frequency is  $\omega_n$ . The amplitude of the response at this frequency was then used to calculate  $\delta$ . The resulting transfer function was found to be

$$T(s) = \frac{36}{s^2 + 3.6s + 36} \quad (12-4)$$

Next a value for the sample period,  $T$ , was determined experimentally. One criterion that has been used successfully is to choose  $T$  as approximately one-tenth of the system rise time [5, 6]. The system step response is recorded in Fig. 12-5, indicating a rise time of approximately 0.3 s. Thus a sample period of 30 ms is a value for consideration.

To test this sample period, the system was connected with the microcomputer in the loop, as shown in Fig. 12-1. The computer was programmed to be a simple gain of unity (i.e., the A/D, computer, and D/A performed the function of a sampler and zero-order hold). The step response was then run for several values of  $T$  in the vicinity of  $T$  equal to 30 ms. The results are given in Fig. 12-6.

First, note that the response for a sample period of 5 ms is quite close to that of the analog system (see Fig. 12-5). Next note that the system is approaching instability for  $T = 40$  ms. However, the small amplitude of the oscillation indicates a nonlinear effect, since generally a linear system oscillation triggered by a step input will have approximately the amplitude of the step. An investigation of this system determined that a jump resonance nonlinearity [7], usually caused by saturation, is present in the system. Jump resonance has been shown to produce limit cycles in sampled-data systems of this type [8]. Thus, for this system, the sampling rate is determined not by linear sampling theory, but by a system nonlinearity. It should be pointed out that a servo system is often designed such that the servoamplifier is saturated for a large percentage of the time. Thus the motor input voltage is maximum during this time, ensuring maximum speed of response.

Consideration of the step responses in Fig. 12-6 resulted in a choice for  $T$  of 5 ms. With this value of  $T$ , we can calculate  $G(z)$ . From (12-2) and (12-4),

$$G_p(s) = \frac{36}{s(s + 3.6)} \quad (12-5)$$

Then

$$G(z) = \mathcal{Z} \left[ \frac{36(1 - e^{-Ts})}{s^2(s + 3.6)} \right] = \frac{0.00044731z + 0.00044739}{z^2 - 1.982161z + 0.982161} \quad (12-6)$$

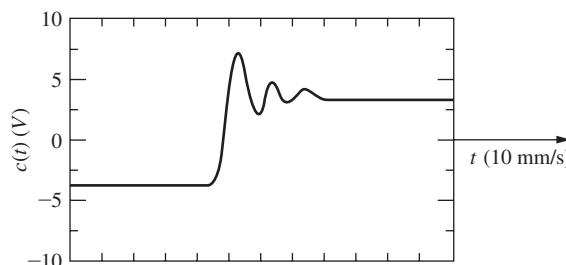
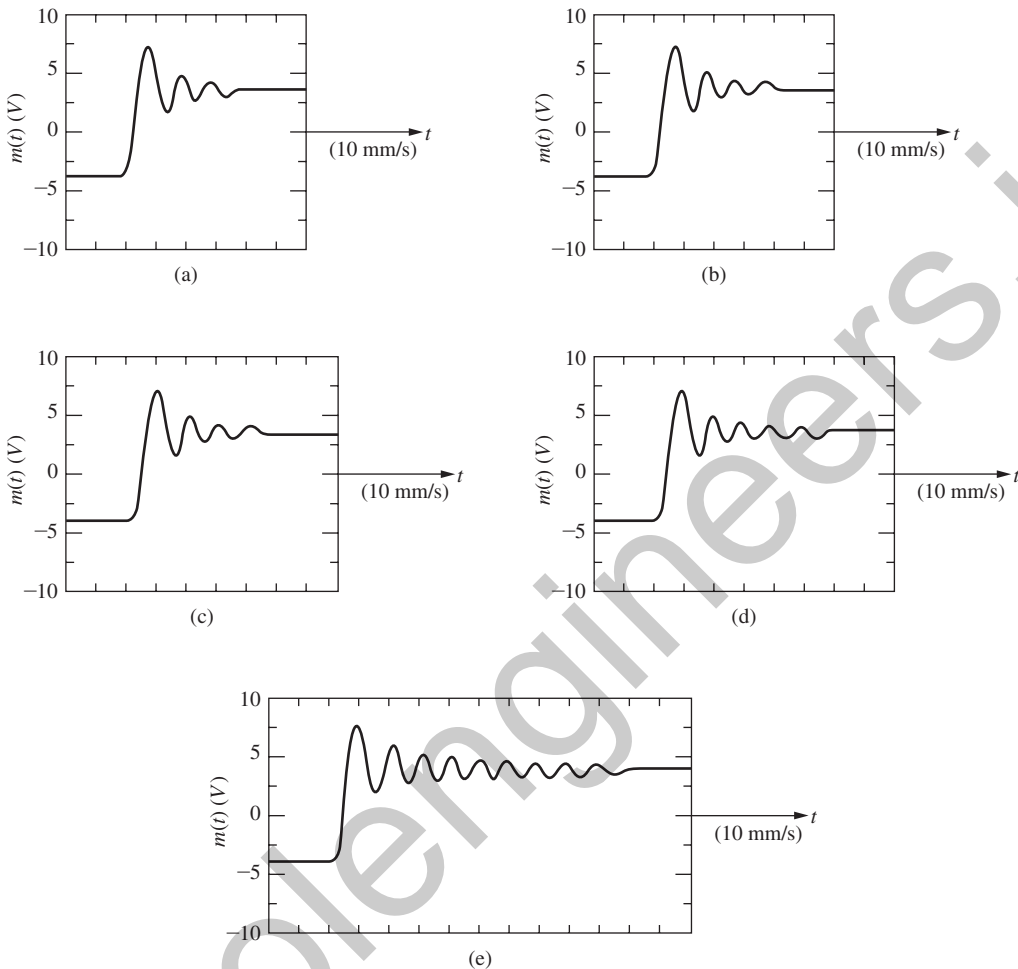


FIGURE 12-5 Step response of the servomotor system.



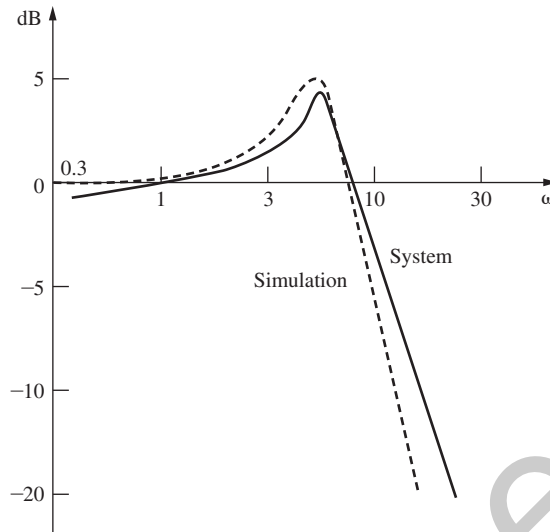
**FIGURE 12-6** Step response of the servomotor for different sampling periods: (a)  $T = 5$  ms; (b)  $T = 10$  ms; (c)  $T = 20$  ms; (d)  $T = 30$  ms; (e)  $T = 40$  ms.

The closed-loop transfer function is then

$$\frac{G(z)}{1 + G(z)} = \frac{0.00044731z + 0.00044739}{z^2 - 1.981714z + 0.982608} \quad (12-7)$$

Note the numerical problems. At low frequencies (in the vicinity of  $z = 1$ ), the frequency response is determined principally by the last digit of the numerator coefficients, and thus numerical inaccuracies will be present. These numerical problems are caused by using a sampling frequency that is very large compared with the system natural frequencies. To calculate the frequency response from (12-6) and (12-7), numerical accuracy greater than that shown must be employed. However, use of (7-23) to calculate the frequency response circumvents these problems.

Shown in Fig. 12-7 is the measured system frequency response and also the frequency response of an analog (linear) computer simulation of the system.



**FIGURE 12-7** Closed-loop frequency response.  $D(z) = 1$ ,  $T = 0.005$  s.

The system frequency response in the vicinity of the resonance peak is actually double valued, because of the jump resonance effects [7], but is shown as being single valued. Since both of these curves were obtained experimentally, they are plotted versus real, or  $s$ -plane, frequency. The  $w$ -plane frequency is obtained from (7-10):

$$\omega_w = \frac{2}{T} \tan \frac{\omega T}{2} \quad (12-8)$$

### Design

The frequency response for  $G(z)$ , obtained using (7-23), is plotted in Fig. 12-8.

As described in Chapter 8, phase lead design is principally trial and error. Use of the procedure of Chapter 8 resulted in a phase-lead filter transfer function of

$$D(w) = \frac{0.693(1 + w/2.21)}{1 + w/44.8} \quad (12-9)$$

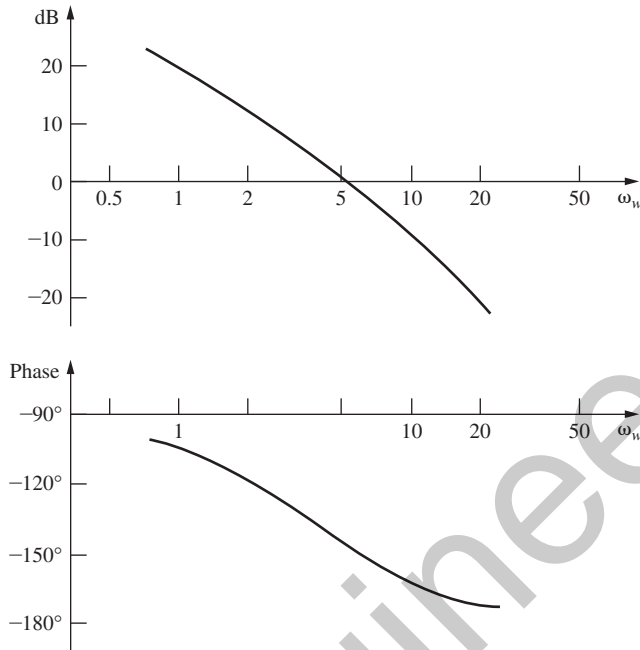
and thus, from (8-14),

$$D(z) = \frac{12.74z - 12.6}{z - 0.798} \quad (12-10)$$

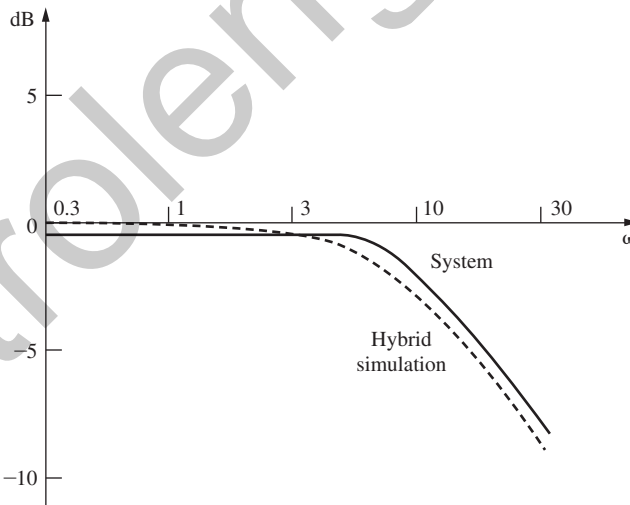
The computer calculates

$$m(nT) = 12.74e(nT) - 12.6e(nT - T) + 0.798m(nT - T)$$

The resulting phase margin was  $80^\circ$ . The system closed-loop frequency response is given in Fig. 12-9. Both responses were obtained experimentally. The analog computer simulation, with filter, was linear; thus the closeness of the two responses indicates little nonlinear effects in the system. The system step response is given in Fig. 12-10.



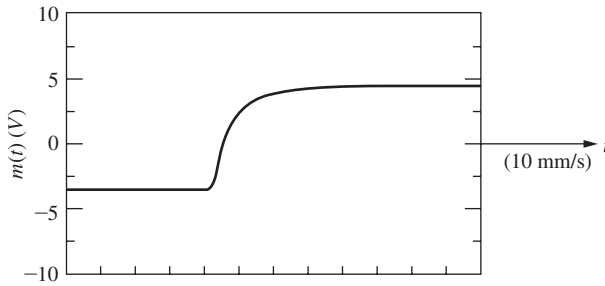
**FIGURE 12-8** Open-loop frequency response.  $T = 0.005$  s.



**FIGURE 12-9** Closed-loop frequency response with phase-lead compensation ( $T = 5$  ms).

A phase-lag filter was also designed, using the procedure given in Chapter 8. The resultant filter transfer function is

$$D(w) = \frac{1.36(1 + w/0.20)}{1 + w/0.044} \tag{12-11}$$



**FIGURE 12-10** Step response of the servomotor system with a phase-lead compensator ( $T = 5$  ms).

and thus

$$D(z) = \frac{0.3z - 0.2997}{z - 0.99978} \quad (12-12)$$

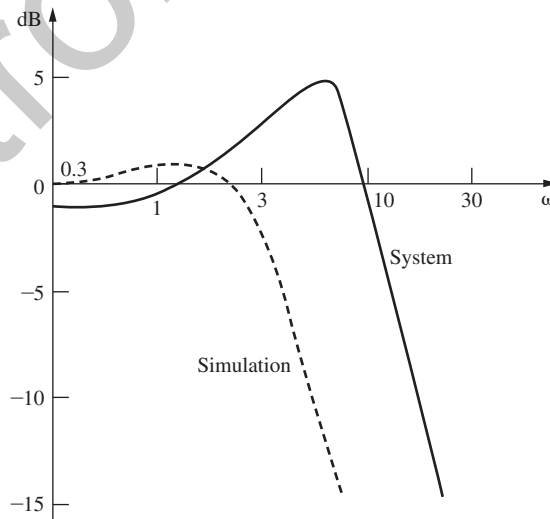
Here the computer calculates

$$y(nT) = 0.3x(nT) - 0.2997x(nT - T) + 0.99978y(nT - T)$$

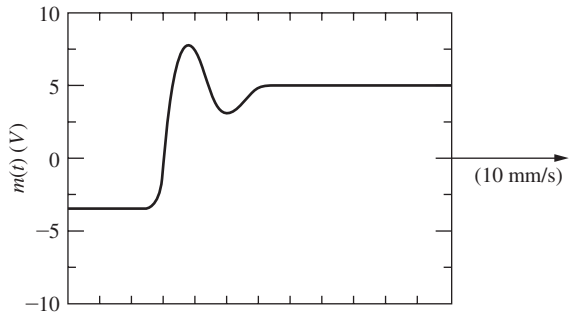
The phase margin for this system is  $60^\circ$ . However, when the frequency responses were obtained experimentally, the curves in Fig. 12-11 resulted. Here the differences in the frequency responses indicate gross nonlinear effects. The system step response is given in Fig. 12-12.

### 12.3 ENVIRONMENTAL CHAMBER CONTROL SYSTEM

This section presents the case study of a digital control system for an environmental chamber designed for the study of plant growth [9]. Two control systems were implemented: one to control dry-bulb temperature and the other to control the carbon dioxide ( $\text{CO}_2$ ) content in the chamber atmosphere. The chamber is constructed of plexiglass with measurements of approximately



**FIGURE 12-11** Closed-loop frequency response with phase-lag compensation ( $T = 5$  ms).

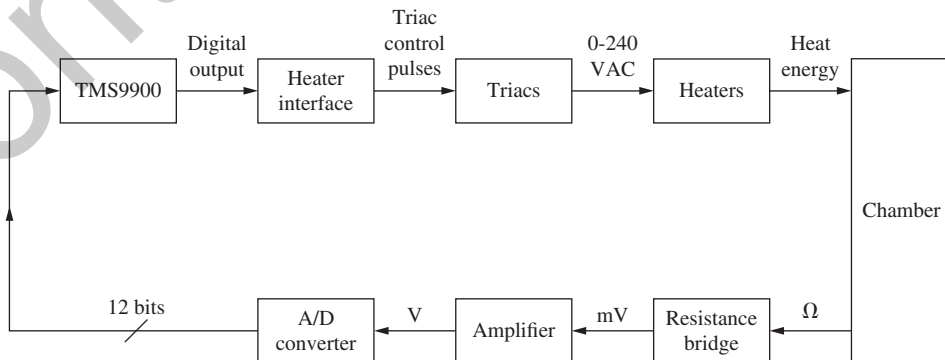


**FIGURE 12-12** Step response of the servomotor system with a phase-lag compensator ( $T = 5$  ms).

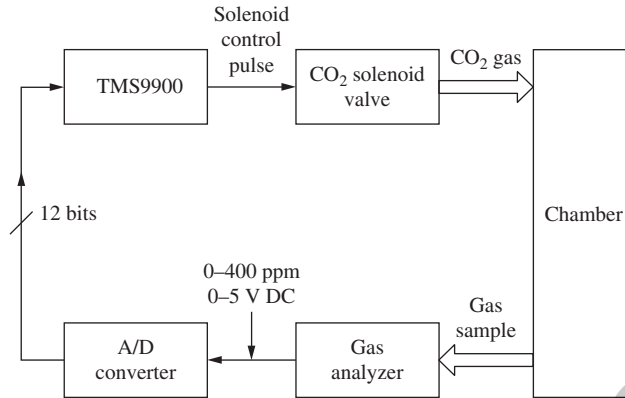
0.7 by 1.3 by 2 m. The chamber system is hermetically sealed such that both moisture content and carbon dioxide content can be accurately monitored.

Fig. 12-13 gives a hardware description of the temperature control system. The chamber is cooled (air conditioned) to the extent that heaters are required to maintain the desired temperature. Thus the heaters are the controlling elements in the closed-loop system. The resistance bridge is dictated by the temperature sensor used, and has an output in the millivolt range. An operational amplifier increases the amplitude of this signal to the  $\pm 5$ -V range required by the A/D converter. The measured temperature is then subtracted from the desired temperature, which is stored in the TMS9900 microcomputer system. Next this error signal is processed by the TMS9900 (the system compensation), resulting in an output signal to the heater interface. The heater interface is a complex logic circuit [9] that converts the computer output signal into the triacs' [10] control pulses. The triacs control the electrical energy into the heaters by in effect controlling the rms voltage applied to the heaters.

The hardware description of the CO<sub>2</sub> control system is given in Fig. 12-14. The CO<sub>2</sub> content of the chamber atmosphere is measured by the gas analyzer in parts per million (ppm). This signal is compared with the desired set point, and if the error is negative (a CO<sub>2</sub> deficit), the computer opens a solenoid valve from a CO<sub>2</sub> supply for a length of time dependent on the error magnitude. The control system has no capability to remove excess CO<sub>2</sub>. Because of the time lag required for the gas analysis, the sampling rate for this control system cannot be greater than 0.0222 ( $T = 45$  s). In addition, no compensation is employed. However, as will be shown later, satisfactory control occurs.



**FIGURE 12-13** Chamber temperature control hardware diagram.



**FIGURE 12-14** CO<sub>2</sub> control hardware.

### Temperature Control System

The temperature control system was designed empirically. Because of the complexity of the system, no attempt was made to derive a plant model. Instead, a step response was run between the heater input and the temperature sensor output. The temperature rise was found to be approximately exponential with a time constant of 60 s. Thus as a first try, the sample period was set to 6 s.

To aid in the empirical design, a proportional-plus-integral-plus-derivative (PID) controller was chosen for implementation (see Chapter 8). The analog version of the PID filter is

$$m(t) = K_p e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt} \quad (12-13)$$

where  $e(t)$  is the controller input and  $m(t)$  is the controller output. The discrete controller implementation of the integrator was chosen to be

$$m_i(k) = \frac{T}{2}[e(k) + e(k-1)] + m_i(k-1) \quad (12-14)$$

and for the differentiator,

$$m_d(k) = \frac{1}{T}[e(k) - e(k-1)] \quad (12-15)$$

[see (8-34) and (8-37)]. Hence the discrete controller implementation of (12-13) is

$$m(k) = K_p e(k) + K_I m_i(k) + K_D m_d(k) \quad (12-16)$$

It is evident from these equations that four parameters are to be determined:  $T$ ,  $K_p$ ,  $K_I$ , and  $K_D$ . However, if an appropriate choice is made for  $T$  such that accurate differentiation and integration occur, the parameters  $K_p$ ,  $K_I$ , and  $K_D$  will be independent of  $T$ .

The hardware for the temperature control system was chosen prior to design of the controller. In many applications the control system designer will find predefined hardware constraints, and must learn to adapt his design techniques, from ideal to practical. Two heaters were specified: one with a slow response (large mass) and one with a faster response (smaller mass). Thus the slow-response heater adds phase lag to the system, which is undesirable. Subsequently, experimentation on the physical system resulted in a choice for the form of the controller as shown in Fig. 12-15. Different PID gains ( $P1$ ,  $I1$ ,  $D1$  equivalent to  $K_P$ ,  $K_I$ ,  $K_D$ , respectively) are implemented for the fast-response heater [controlled by the signal  $c_1(k)$ ] from those employed with the slow-response heater, which is controlled by  $c_2(k)$ . This form for the controller results in a better system response than that obtained by controlling both heaters with a single signal.

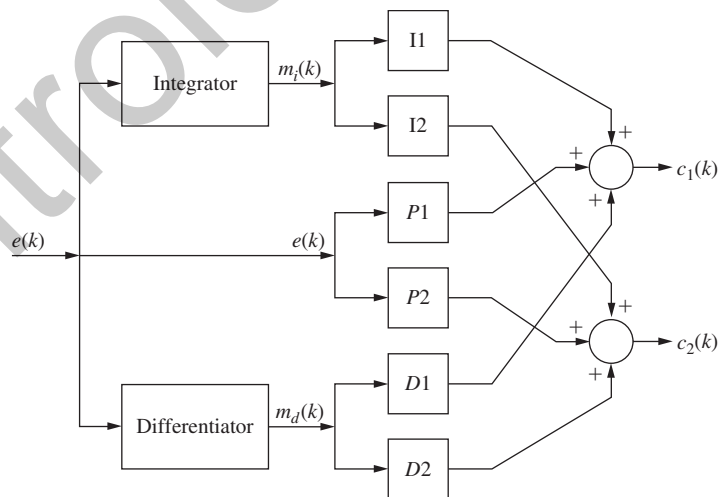
The controller gains were determined by first implementing a proportional-only controller, and varying  $P1$  and  $P2$  in Fig. 12-15. A typical step response is given in Fig. 12-16, with  $P1$  and  $P2$  both equal to 25.

The small oscillation visible in this response is caused by the air conditioner cycling on and off. The control system was commanded to a  $10^\circ$  Celsius step input, from  $25^\circ\text{C}$  to  $35^\circ\text{C}$ . Note the steady-state error in Fig. 12-16. During these tests, the sample period  $T$  ( $T_s$  in Fig. 12-16) was also varied, and a value of  $T = 1$  s was chosen; that is, choosing  $T$  less than 1 s caused no observable improvement in the system response.

Next the integral term was added to the controller, and a typical step response is given in Fig. 12-17.

Note the elimination of the steady-state error, as expected (see Section 8.9). Finally, the derivative term was added to the controller, resulting in a typical response as shown in Fig. 12-18.

Note that no improvement occurs in the rise time for the PID controller, because the heaters are full on for large error signals for both the P controller and the PI controller (a nonlinear effect). Thus the system cannot respond any faster for large errors and this effect is independent of the form of the controller. However, for small errors (e.g., small overshoot), the derivative term in the controller has a definite and predictable effect (normal linear operation).



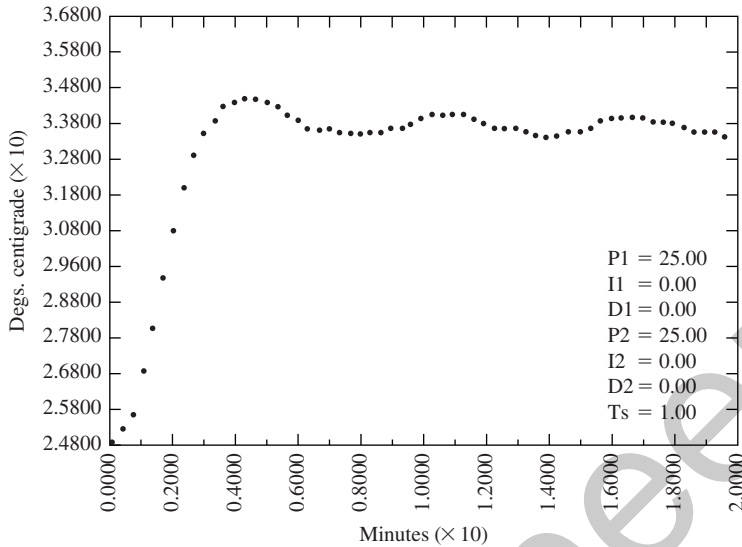
where  $e(k)$  = error at  $t = kT$

$$m_i(k) = T/2 e(k) + T/2 e(k-1) + m_i(k-1)$$

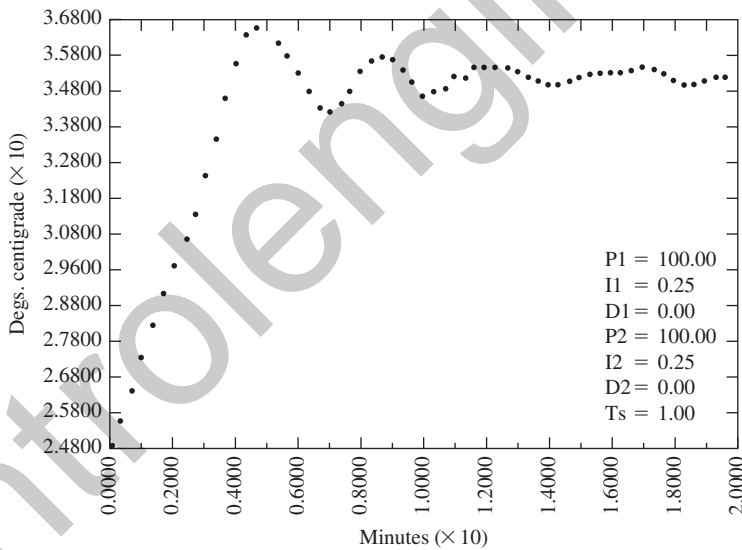
$$m_d(k) = 1/T e(k) - 1/T e(k-1)$$

**FIGURE 12-15** Block diagram of PID filter implementation.





**FIGURE 12-16** Proportional controller step response behavior.



**FIGURE 12-17** PI controller step response behavior.

In all the tests above, both heaters were controlled by the same signal. Attempts to increase the derivative signal beyond certain values while using this strategy resulted in a tendency toward instability. This effect is predictable, since the maximum phase lead attainable from a PID controller is  $90^\circ$ , but the gain may be increased without limit (theoretically). At this point, the PID gains for the fast heater were varied from those for the slow heater in order to reduce overshoot and settling time. The final values for the filter gains were chosen to be:  $P1 = 100$ ,  $I1 = 0.3$ ,  $D1 = 100$ ,  $P2 = 25$ ,  $I2 = 0.25$ ,  $D2 = 25$ , and  $T = 1$  s.

A typical 24-hour system response is given in Fig. 12-19. Time equal to zero is midnight. The carbon dioxide can be controlled only during daylight hours, since plants emit  $CO_2$  at night.

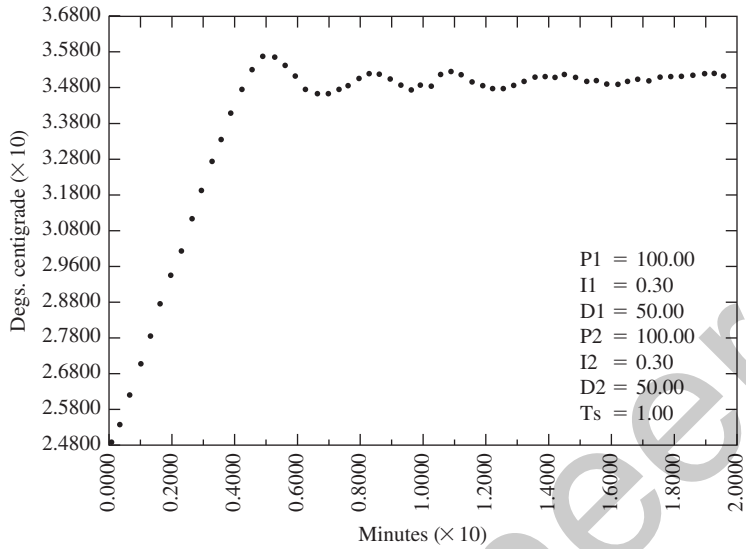


FIGURE 12-18 PID controller step response behavior.

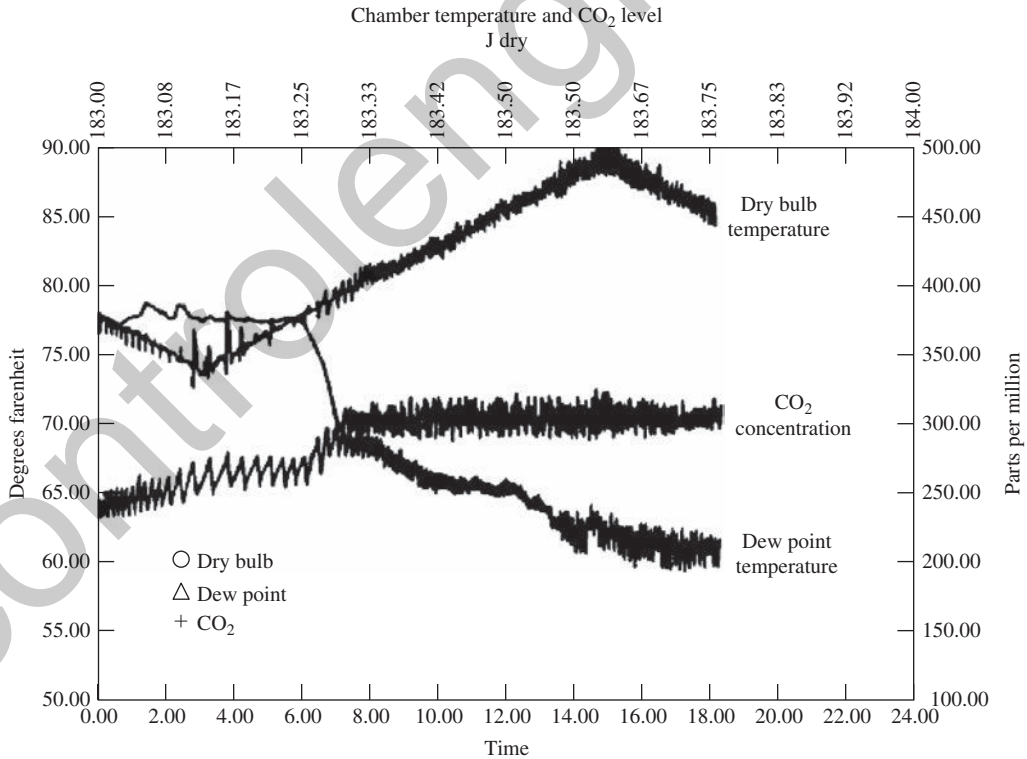


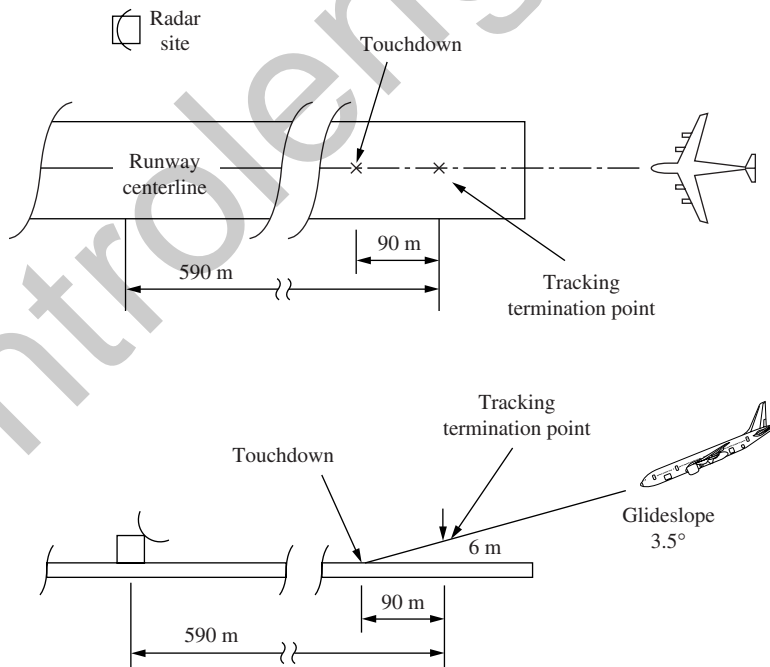
FIGURE 12-19 Chamber behavior for normal operating conditions.

For this figure, the temperature was commanded to follow a triangular wave which is an approximation to actual temperature variations during a typical 24-hour period.

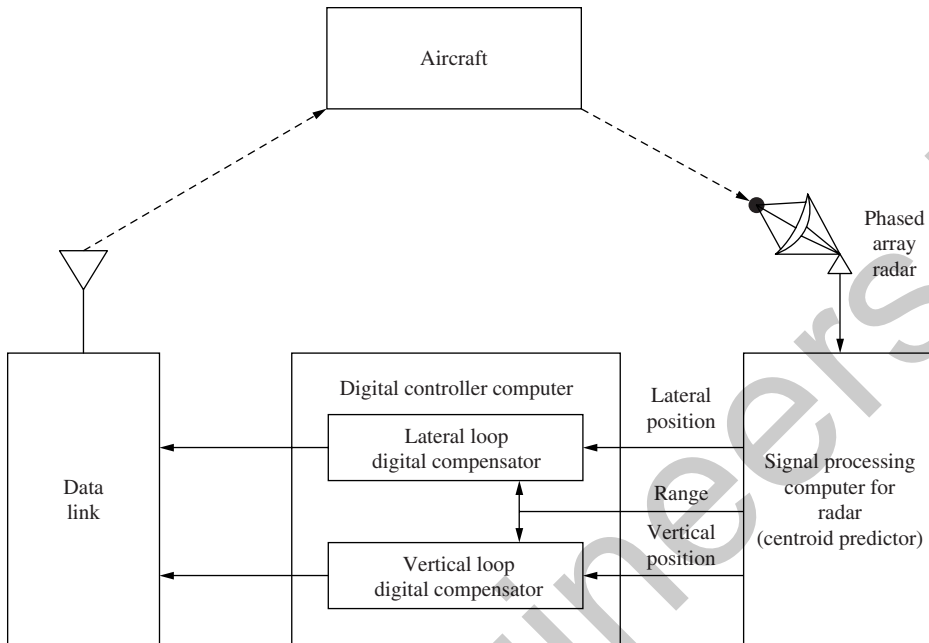
## 12.4 AIRCRAFT LANDING SYSTEM

This section presents the design of an automatic aircraft landing system. The particular control system described is that of the Marine Air Traffic Control and Landing System (MATCAL) [12, 13]. The system is illustrated in Fig. 12-20. The radar (the control system sensor) employs a phased array antenna, which was chosen because the radar beam can be rapidly repositioned to control a number of planes at sampling rates of up to 40 Hz. However, the positional information provided by the radar system is corrupted with significant noise, which presents problems to the control system designer.

The control operation is composed of two independent (decoupled) control systems: the vertical control system, which keeps the plane on a  $3.5^\circ$  glide slope, and the lateral control system, which maintains the plane on the extended centerline of the runway. These two control systems are illustrated in Fig. 12-21. Two separate computer-processing algorithms are utilized in each control system. The first algorithm processes the antenna return signals to determine the aircraft position (centroid position) with respect to the runway touchdown point. The second algorithm comprises the difference equations that describe the digital controllers. The outputs of the controllers are transmitted via a data link to the aircraft and applied to the appropriate



**FIGURE 12-20** Aircraft landing system.



**FIGURE 12-21** MATCALs automatic landing control loops.

autopilot inputs: the pitch autopilot for the vertical control system and the bank autopilot for the lateral control system.

In the following development, only the lateral control system will be discussed. The design of the vertical control system follows similar paths of development.

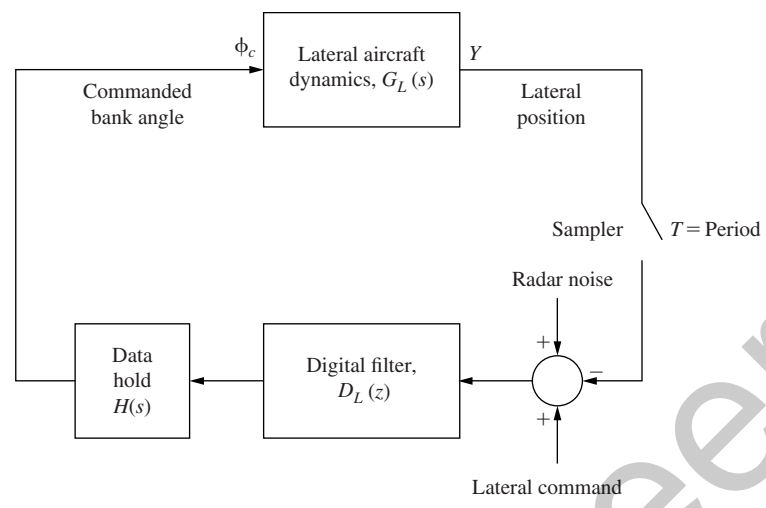
### Plant Model

A block diagram of the lateral control system is given in Fig. 12-22. The lateral aircraft dynamics, described by the transfer function  $G_L(s)$ , include the bank autopilot and typically have a frequency response as shown in Fig. 12-23. These dynamics are of course aircraft dependent; the frequency response of the McDonnell Aircraft Company F4J aircraft is given in Fig. 12-24, and was obtained from measurements on the aircraft in flight [14]. The lateral aircraft equations of motion will not be developed here (see [15]), but are ninth order for the F4J aircraft with autopilot included. Note from the  $-40$  dB slope on the magnitude curve of  $G_L(j\omega)$  in Fig. 12-24 that  $G_L(s)$  has a second-order pole at the origin.

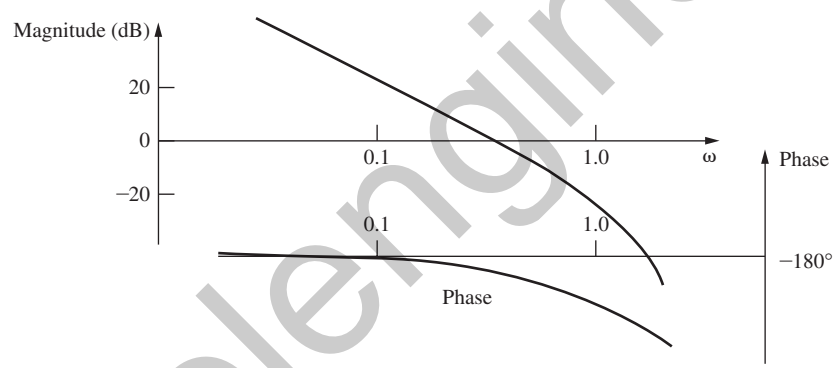
A sketch of the Nyquist diagram for the F4J aircraft lateral control system is given in Fig. 12-25, as can be seen from Fig. 12-24. Since the aircraft with autopilot must be (open-loop) stable, it is evident that phase-lead compensation must be employed to eliminate the encirclement of the  $-1$  point, in order to stabilize the system.

### Design

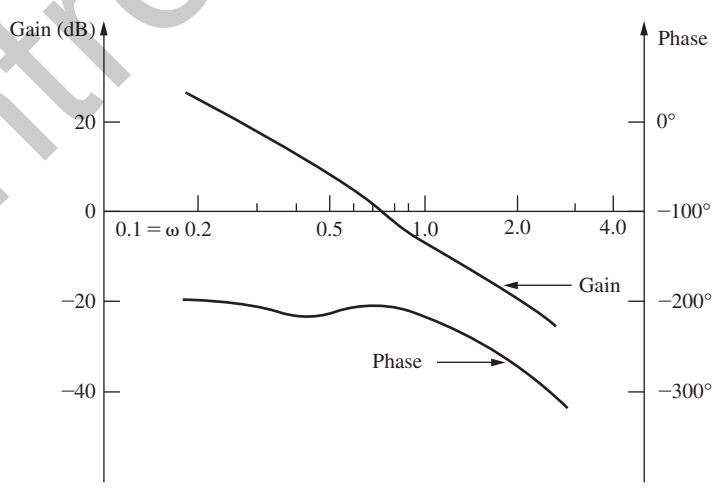
There are three significant disturbance sources in the lateral control system which must be considered in the system design. The first of these is the radar noise, and is indicated in Fig. 12-22. The other two disturbances, which will be considered first, are direct inputs to the aircraft lateral



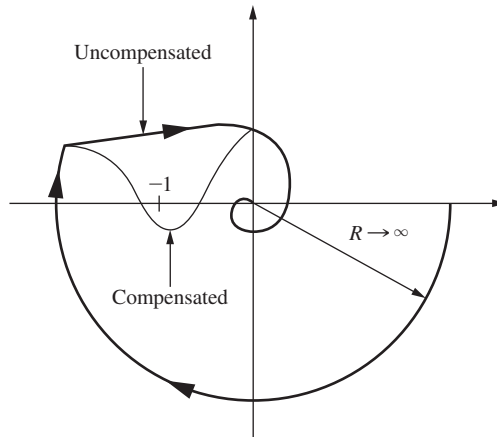
**FIGURE 12-22** Lateral control loop.



**FIGURE 12-23** Typical frequency response of  $G_L(s)$ .



**FIGURE 12-24** F4J lateral frequency response ( $Y/\phi_c$ ).



**FIGURE 12-25** Nyquist diagram of  $G_L(s)$ , and  $G_L(s)$  with possible compensation.

dynamics. Thus each of these disturbances may be modeled as shown in Fig. 12-26. Hence the influence of the disturbance on the lateral position  $y(k)$  is given by

$$Y(z) = \frac{\overline{G_D U(z)}}{1 + D(z)G(z)} \tag{12-17}$$

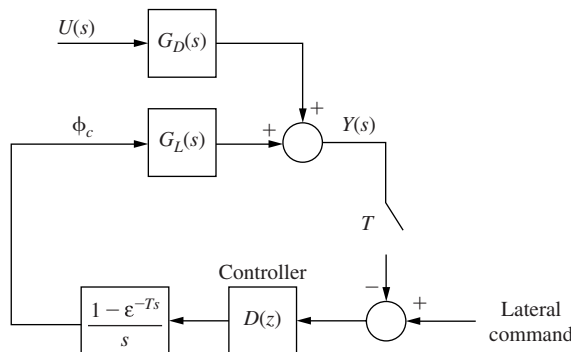
where

$$G(z) = \mathfrak{Z} \left[ \frac{1 - \epsilon^{-Ts}}{s} G_L(s) \right] \tag{12-18}$$

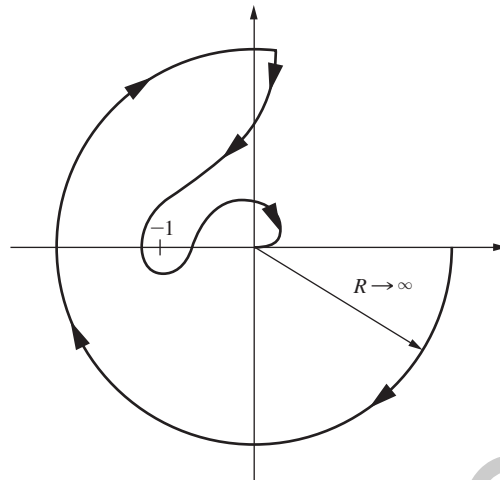
and, as stated previously,  $G_L(s)$  has two poles at  $s = 0$ , resulting in two poles of  $G(z)$  at  $z = 1$ .

One of the disturbances to be modeled as  $U(s)$  in (15-17) is a relatively constant error (called a dc bias) in the output of a rate gyroscope in the autopilot. This error can be treated as a system input. Note that, in general, a constant input for the disturbance in Fig. 12-26 will result in a constant output  $y(k)$  in the steady state. The aircraft would then land to one side of the runway centerline, which is unacceptable. Thus a system requirement is that the final-value theorem when applied to (12-17) yield a value of zero; that is,

$$\lim_{k \rightarrow \infty} y(k) = \left. \frac{(z - 1) \overline{G_D U(z)}}{1 + D(z)G(z)} \right|_{z=1} = 0 \tag{12-19}$$



**FIGURE 12-26** Lateral control system with a disturbance.



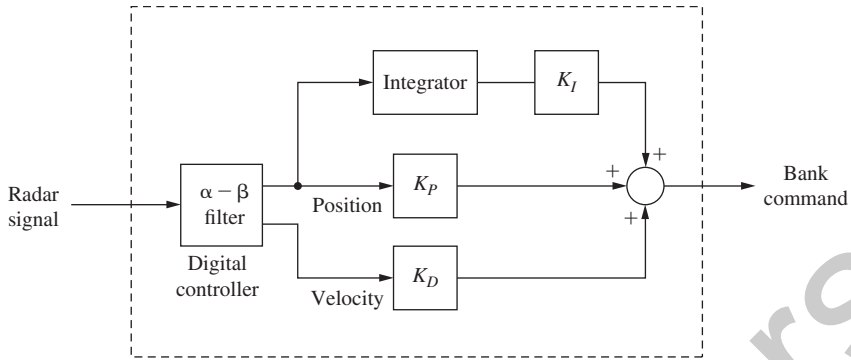
**FIGURE 12-27** Lateral control system Nyquist diagram.

with  $U(s) = 1/s$ . Now both  $G_D(s)$  and  $G_L(s)$  of Fig. 12-26 have two poles at  $s = 0$ . Thus  $\overline{G_D U}(z)$  has three poles at  $z = 1$ , and  $G(z)$  has two poles at  $z = 1$ . If  $D(z)$  is given a pole at  $z = 1$ , (12-19) is satisfied. Hence a proportional-plus-integral (PI) controller is a system requirement. As was seen from Fig. 12-25, phase lead is required for stability. Since the derivative term in the PID controller contributes phase lead [see (8-47)], the PID controller was chosen to compensate this system. The compensated-system Nyquist diagram then appears as shown in Fig. 12-27.

The second disturbance source of the type illustrated in Fig. 12-26 is the wind. It is desirable that the aircraft not respond to wind inputs, but of course this is not possible. So the controller is designed to reduce the effects of wind. Consider again the effect of the wind disturbance as modeled in Fig. 12-26 and equation (12-17). Of course,  $G_D(s)$  is different for wind disturbances from that for the dc bias disturbance. In (12-17), the response is reduced if  $D(z)G(z)$  is made large in the frequency range of the wind (turbulence) input signal. Since the frequency response of  $G(z)$  is fixed,  $D(z)$  should be designed to increase the system bandwidth. As we have seen in Chapter 8, the derivative path of the PID controller not only introduces phase lead, but also increases system bandwidth. Hence we require a high gain in the derivative path.

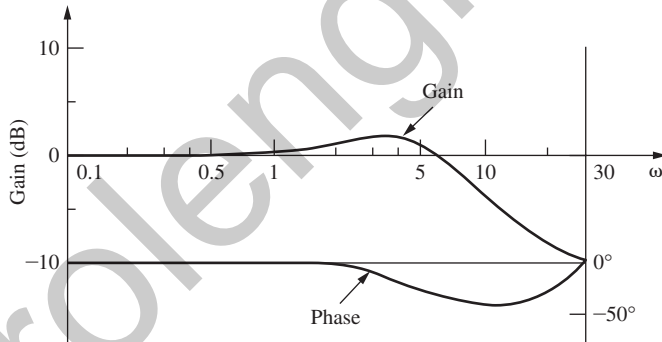
The third disturbance source, which is the noise present in the radar system output signal, is modeled as shown in Fig. 12-22. Note that the radar noise enters the system at the same point as does the lateral position command signal. Thus a system design that yields a good response to the command input also yields a good response to the radar noise. This noise response is a major problem in the design of this system. For aircraft-carrier-based systems [16], a parabolic radar antenna system is used. The signal from this type of antenna system is relatively noise-free, and the noise problem is not as critical as in the land-based system.

**$\alpha$ - $\beta$  FILTER.** To reduce the effects of the radar noise, an  $\alpha$ - $\beta$  tracking filter [17] was chosen to filter the radar signal at the controller input. (The  $\alpha$ - $\beta$  filter equations are given in Problem 2-21.) This filter is designed to estimate  $y(k)$ , the aircraft lateral position, and  $\dot{y}(k)$ , the aircraft lateral velocity, given a noisy radar system output signal. The estimate  $y(k)$  is then transmitted to the position path and the integrator path, and the estimate  $\dot{y}(k)$  to the derivative path, of the PID compensator. Thus the PID compensator has the basic configuration shown in Fig. 12-28.

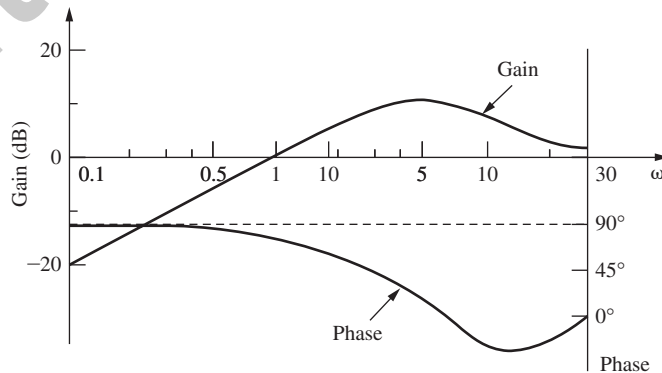


**FIGURE 12-28** Basic form of the controller.

The frequency responses plotted versus real frequency  $\omega$  of an  $\alpha$ - $\beta$  filter are given in Fig. 12-29. The sample period is  $T = 0.1$  s; thus  $\omega_s/2$  is 31.4 rad/s. Note that at low frequencies the position filter (Fig. 12-29(a)) has unity gain and no phase shift, but it attenuates high-frequency noise. At low frequencies the velocity filter response (Fig. 12-29(b)) has a slope of  $-20$  dB/decade, which is equal to that of an exact differentiator. However, the phase characteristic varies from that of a differentiator, which has a constant phase of  $90^\circ$ . This filter also attenuates high-frequency noise. For the filter plotted,  $\alpha = 0.51$  and  $\beta = 0.1746$ , which are the values used in the F4J controller.



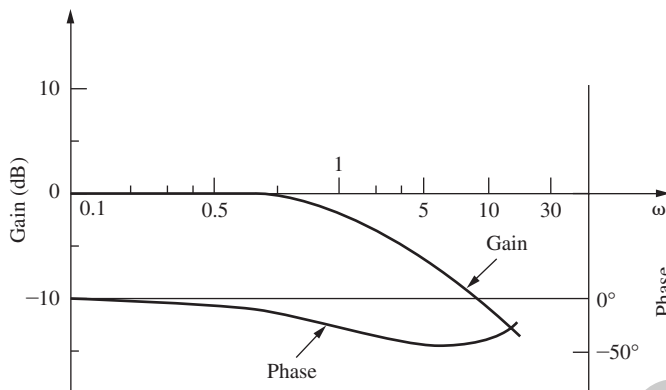
(a)



(b)

**FIGURE 12-29** (a)  $\alpha$ - $\beta$  position filter response; (b)  $\alpha$ - $\beta$  velocity filter response.





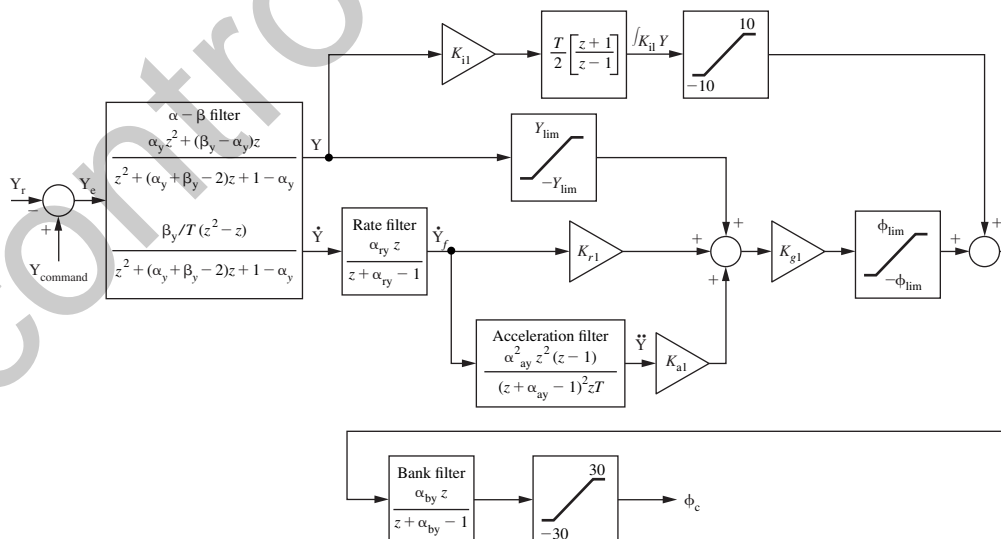
**FIGURE 12-30**  $\alpha$ -filter frequency response.

**$\alpha$ -FILTERS.** The noise-reduction characteristics of the  $\alpha$ - $\beta$  filter do not adequately attenuate the radar noise in this system. To reduce the high-frequency noise further, low-pass filters (at times called  $\alpha$ -filters [16]) are added at various points in the PID controller. The  $\alpha$ -filter has a transfer function given by

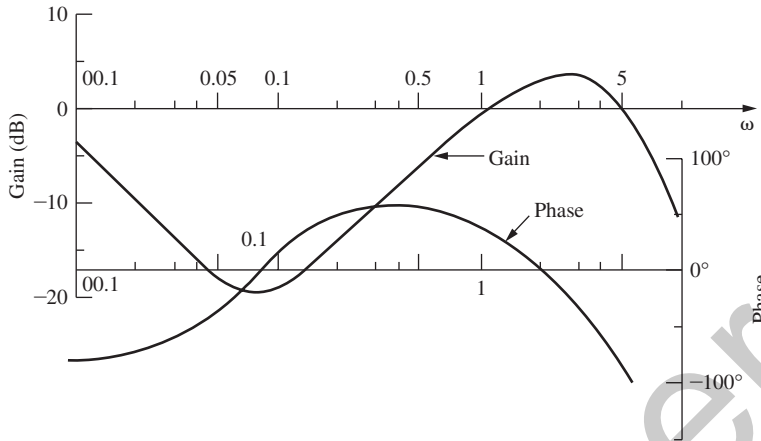
$$D_{\alpha}(z) = \frac{\alpha z}{z - (1 - \alpha)} \quad (12-20)$$

The frequency response of an  $\alpha$ -filter for  $\alpha = 0.234$  and  $T = 0.1$  s is given in Fig. 12-30. This filter is also employed in the F4J controller. As is normal in low-pass filtering, phase lag is added to the system. It is seen from the system Nyquist diagram in Fig. 12-27 that phase lag is already a major design problem. Thus the final design must include a trade-off between desired stability margins and radar noise rejection.

The foregoing system requirements resulted in a final filter design illustrated in Fig. 12-31. An additional differentiator is added to the  $\dot{y}(k)$  path to produce additional phase



**FIGURE 12-31** Lateral control system controller.



**FIGURE 12-32** F4J lateral controller frequency response,  $T = 0.1$  s.

lead, since that produced by the derivative term was insufficient. The differentiator has the transfer function

$$D_{\alpha}(z) = \frac{z - 1}{Tz} \quad (12-21)$$

[see (8-38)]. Because of the high-frequency noise amplification from this differentiation of a noisy signal, two sections of  $\alpha$ -filter are required for this path. The nonlinearities shown in the filter are not exercised during normal system operation. The limits  $\phi_{lim}$  and  $Y_{lim}$ , and the various gains shown, are all functions of range, with the gains increasing as the aircraft nears touchdown. The gains are constant for the final 1500 m of flight. The frequency response (versus real frequency  $\omega$ ) of the lateral controller for the F4J aircraft is given in Fig. 12-32; the standard PID filter response is seen, except for the high-frequency attenuation added to reduce noise effects.

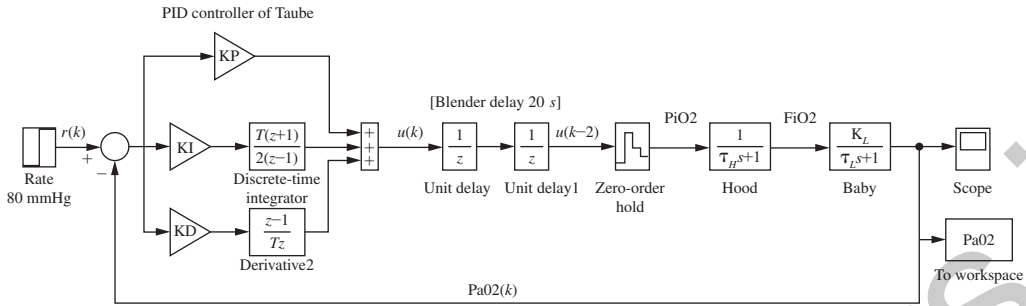
The PID path gains used, for range less than 1500 m, are  $K_P = 0.1$ ,  $K_I = 0.0033$ ,  $K_d = 0.75$ , and (acceleration path)  $K_A = 0.75$ . The resultant phase margin is  $60^\circ$ , and the gain margin is 8 dB. Two significant nonlinearities in the lateral control system are the mechanical limits on the rotations of the ailerons and of the rudder. For a landing in high wind turbulence, these nonlinearities are exercised and the stability characteristics are degraded.

## 12.5 NEONATAL FRACTIONAL INSPIRED OXYGEN

In this case study, we examine the performance of PID feedback controllers for neonatal fractional inspired oxygen [18, 19]. First, we rebuild the model and closed-loop system of Taube et al. [20–22] as shown in Fig. 12-33. Then, we use MATLAB's pidtool to design PIDF controllers at three different sampling intervals. The PIDF controller results are compared with those of Taube's original design.

### Plant Transfer Function

The main basis for this model was the Sano and Kikuchi model [23]. Some assumptions made by the Sano and Kikuchi model are as follows:



KP, Proportional gain of PID controller  
KI, Integral gain of PID controller  
KD, Derivative gain of PID controller  
PiO2, Partial fraction of delivered oxygen

FiO2, Fractional inspired oxygen  
PaO2, Blood partial pressure of oxygen  
Ref = 80 mmHg: Reference signal  
T = 10 s: Sampling interval

**FIGURE 12-33** Closed-loop system model in Simulink.

1. Rate of respiration and oxygen consumption are constant values.
2. Mixing of gases present inside the body and the inspired oxygen is instantaneous.
3. Respiratory exchange ratio (ratio of the number of molecules of oxygen taken in to the number of molecules of carbon dioxide breathed out in a single breath) is constant.

For the mathematical formulation of the analog model of the plant, we consider the blender, hood, and the baby separately as in [20]. The transfer functions of the blender, hood, and baby are given separately as follows:

$$\text{Blender} = K_b e^{-T_b s} \quad (12-22)$$

The exponential term represents the blender time delay factor in the transportation of oxygen.  $K_b$  is the oxygen blender gain. From experimental results [20],  $K_b = 1.0$  and  $T_b = 20$  s.

$$\text{Hood} = \frac{1}{1 + \tau_H s} \quad (12-23)$$

The term  $\tau_H$  is the hood time constant. This value represents the amount of time taken for oxygen from the blender to reach the hood. From [20], its value is 55 s.

$$\text{Baby} = \frac{K_L}{1 + \tau_L s} \quad (12-24)$$

$K_L$  is the lung transfer ratio. The parameter  $\tau_L$  is the lung time constant. The lung time constant represents a constant value of time that is required for an amount of air to be passively exhaled, irrespective of the starting volume of air in the lungs. From [20],  $K_L = 90$  and  $\tau_L = 12$  s.

So the analog transfer function of the entire plant is now given as

$$G_p(s) = \frac{K_L K_b e^{-T_b s}}{(1 + \tau_H s)(1 + \tau_L s)} \quad (12-25)$$

Substituting all the constants' values, the plant transfer function is obtained as

$$G_p(s) = \frac{90e^{-20s}}{(1 + 55s)(1 + 12s)} = \frac{(3/22)e^{-20s}}{s^2 + (67/660)s + (1/660)} \quad (12-26)$$

The above model is based on the simplified model proposed in [23] with the following steady-state relation between FiO2 and PaO2:

$$PaO_2 = (K_L)FiO_2 + K_2 \quad (12-27)$$

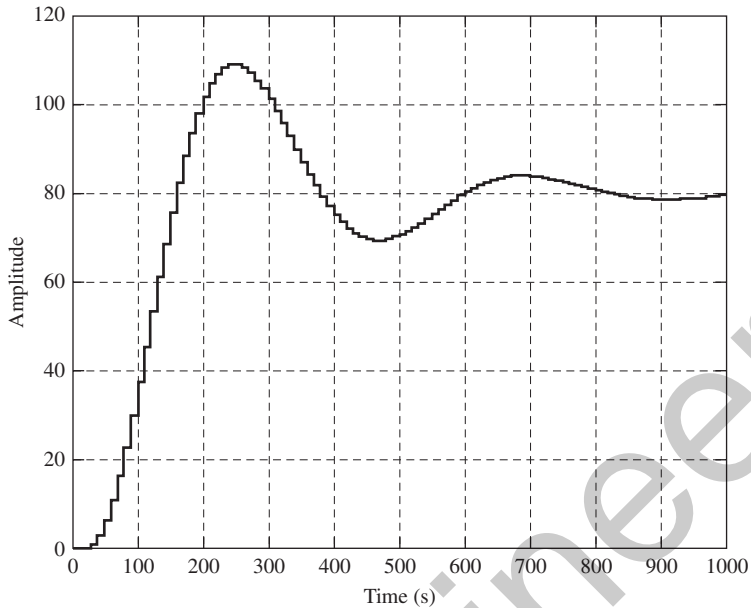
For an infant suffering severe respiratory distress,  $K_L$  and  $K_2$  can be 90 and 0, respectively, requiring an oxygen ratio of over 90% to maintain normal blood oxygen levels. The control objective is to maintain  $PaO_2 = 80$  mmHg:

### Taube's PID Controller

The controller for this system developed by Taube is a parameterized PID controller, which has some adaptive characteristics [20–22]. The gains and step performance of Taube's PID controller can be calculated using the following MATLAB program:

```
Kb=1; TH=55; TL=12; KL=90; Tb=20; T= 10; K=KL*Kb; d=1.5*T+Tb;
KI = 0.5/(K*d)
a = exp(T/TL); b = exp(T/TH); m = a*b; s = a+b;
KD = KI/(1+m-s)
KP = (s-2)*KD
Dz = tf([KP+KI*T/2+KD/T KI*T/2-KP-2*KD/T KD/T],[1 -1 0],10);
Dz_PID = zpk(Dz)
baby = tf(KL, [TL 1]);
hood = tf([1],[TH 1],'inputdelay',20); % hood with 20s delay
Gp = hood*baby;
Gz = c2d(Gp,10);
Cz = feedback(Dz*Gz,1);
opt = stepDataOptions;
opt.StepAmplitude = 80;
step(Cz,1000,opt),grid % Plot Taube Step Response, Ref=80mmHg
[Gm,Pm] = margin(Dz*Gz); Gm dB = 20*log10(Gm), Pm
disp('Unit-step performance metrics:')
stepinfo(Cz)
Gz =
          4.937 z + 3.523
z^(-2) * -----
          z^2 - 1.268 z + 0.3623
KI = 1.5873e-04
KD = 6.1189e-04
KP = 9.1806e-04
Dz_PID =
    0.0017729 (z^2 - 0.1392z + 0.03451)
-----
                z (z-1)
```

The resulting step response is shown in Fig. 12-34.



**FIGURE 12-34** Step response for proposed system [20–22].

The system performance parameters are

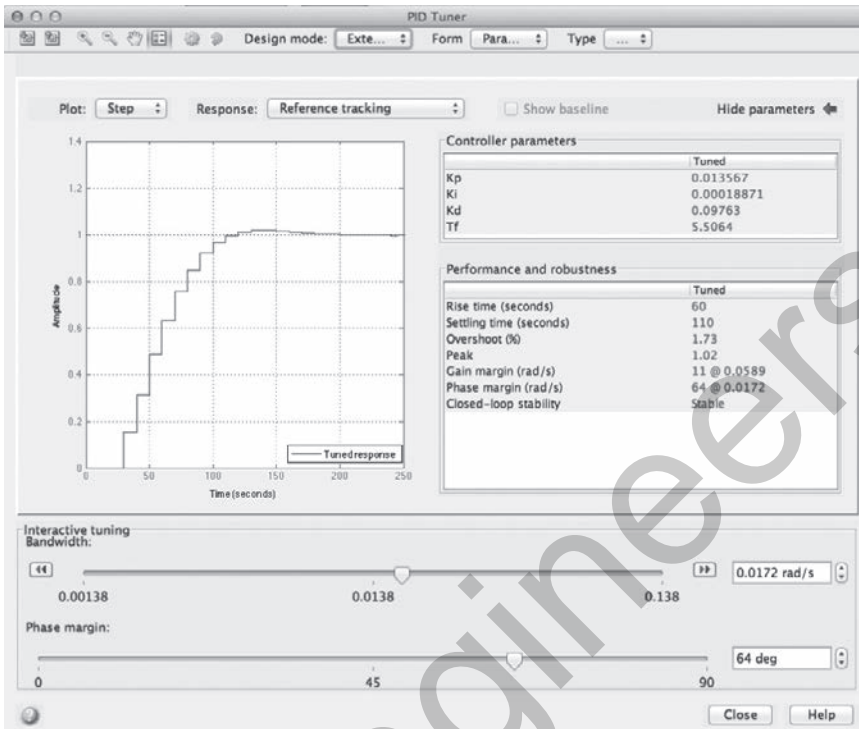
```
GmdB:      8.0819
Pm:        35.7669
RiseTime:   90
SettlingTime: 780
SettlingMin: 0.8678
SettlingMax: 1.3612
Overshoot:  36.1163
Undershoot: 0
Peak:       1.3612
PeakTime:   250
```

The settling time is 13 minutes. Our goal is to improve that with some different PID controller designs using the MATLAB pidtool as demonstrated in Example 8.6.

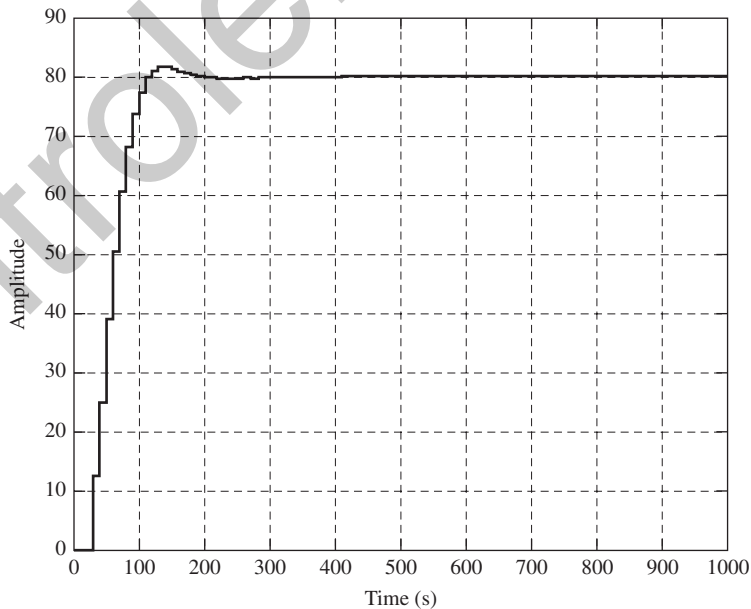
### MATLAB *pidtool* PIDF Controllers

First, we continue the approach used by Taube [20–22] and choose the sampling interval  $T = 10$  s so that the blender delay  $T_b = 2T$ . Hence, we can model the blender delay as  $z^{-2}$  as shown in (4-39). Details are given in the MATLAB program below. Fig. 12-35 shows the *pidtool* results and Fig. 12-36 shows the step response.

```
T = 10;
Gp = tf([3/22],[1 67/660 1/660]); % without blender delay
delay = tf([1],[1 0 0],T); % blender delay = z^(-2)
Gpz = c2d(Gp,T);
Gz=delay*Gpz
disp(' ')
disp('Now opening MATLAB pidtool GUI')
disp('Set Pm = 64, wwl = 0.0172') % Found by trial and error
```



**FIGURE 12-35** Screen shot of MATLAB pidtool ( $T = 10$  s). Source: Reprinted with permission from The MathWorks, Inc.



**FIGURE 12-36** Step response for PIDF controller ( $T = 10$  s).

% To get the same results shown below, first increase the  
 %phase margin to 64 using the up arrow on the value box, then  
 %increase the bandwidth wwl to 0.0172 using the same procedure  
 pidtool(Gz, 'pidf')

```
Gz =
      4.937 z + 3.523
-----
      z^4 - 1.268 z^3 + 0.3623 z^2
```

Now opening MATLAB pidtool GUI  
 Set Pm = 64, wwl = 0.0172

```
% Gains from MATLAB pidtool with Pm = 64, wwl = 0.0172
T, KP = 0.013567, KI = 0.00018871, KD = 0.09763, TF = 5.5064
disp('PIDF using Equation (8-65)') %Find Dz using Eqn (8-65)
Dz_PIDF=KP+(KI*T)*tf([1],[1 -1],T)+KD*tf([1 -1],[TF -TF+T],T)
Dz_PIDF = zpk(Dz_PIDF), Dz_PIDFGz = Dz_PIDF*Gz;
Cz_PIDF = feedback(Dz_PIDFGz,1); opt = stepDataOptions;
opt.StepAmplitude = 80;
step(Cz_PIDF,1000,opt),grid %Plot PIDF Step Response, Ref=80mmHg
disp('PIDF Gain and Phase Margin:')
[Gm,Pm] = margin(Dz_PIDFGz); GmdB = 20*log10(Gm), Pm
% disp('Unit-step performance metrics:')
PIDF_stepinfo=stepinfo(Cz_PIDF)
```

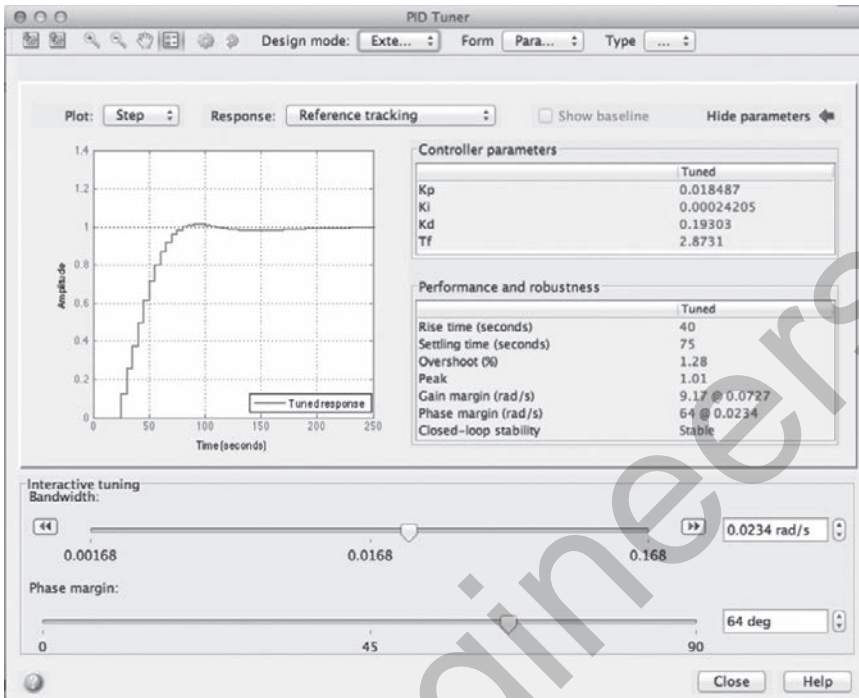
```
T = 10
KP = 0.0136
KI = 1.8871e-04
KD = 0.0976
TF = 5.5064
```

```
PIDF using Equation (8-65)
Dz_PIDF =
      0.031297 (z-0.841) (z-0.3115)
-----
      (z-1) (z+0.8161)
```

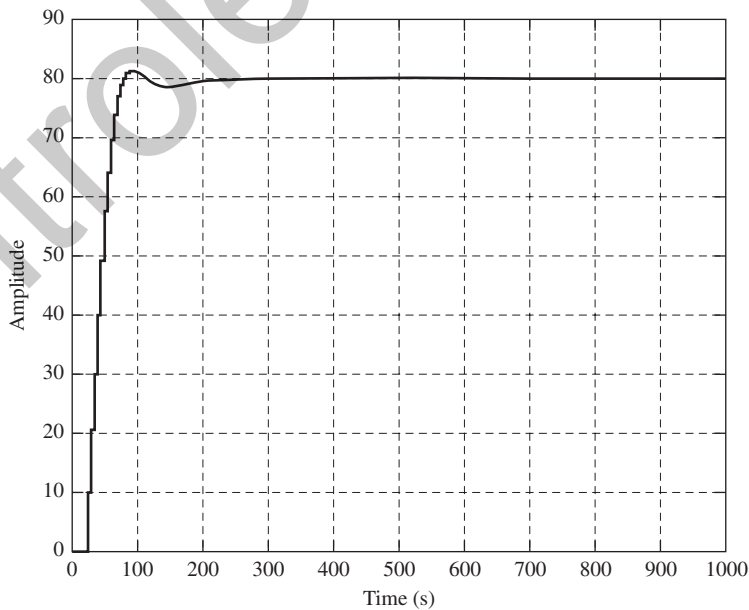
```
GmdB:      11.0470
Pm:        63.9920
RiseTime:   60
SettlingTime: 110
SettlingMin: 0.9207
SettlingMax: 1.0173
Overshoot:  1.7342
Undershoot: 0
Peak:       1.0173
PeakTime:  130
```

The performance of this PIDF controller is improved with the settling time being reduced from 13 minutes to 110 seconds. Noting that the lung time constant of the baby is 12 seconds and our sampling interval is 10 seconds, our sampling interval is perhaps too long. So we can choose  $T = 5$  s and model our blender delay as  $z^{-4}$ . Repeating the PIDF design example with these new parameters follows. Fig. 12-37 shows the *pidtool* tuning window and Fig. 12-38 shows the step response.

```
T = 5;
Gp = tf([3/22],[1 67/660 1/660]); % without blender delay
delay = tf([1],[1 0 0 0 0],T); % blender delay = z^(-4)
```



**FIGURE 12-37** Screen shot of MATLAB *pidtool* ( $T = 5$  s). Source: Reprinted with permission from The MathWorks, Inc.



**FIGURE 12-38** Step response for PIDF controller ( $T = 5$  s).



```
Gpz = c2d(Gp,T);
Gz=delay*Gpz
disp(' ')
disp('Now opening MATLAB pidtool GUI')
disp('Set Pm = 64, ww1 = 0.0234') % Found by trial and error
pidtool(Gz,'pidf')

Gz =
      1.445 z + 1.22
-----
z^6 - 1.572 z^5 + 0.602 z^4

% Gains from MATLAB pidtool with Pm = 64, ww1 = 0.0234
T, KP = 0.018487, KI = 0.00024205, KD = 0.19303, TF = 2.8731
% Find Dz using Equation (8-65)
disp('PIDF using Equation (8-65)')
Dz_PIDF=KP+(KI*T)*tf([1],[1 -1],T)+KD*tf([1 -1],[TF -TF+T],T);
Dz_PIDF = zpke(Dz_PIDF)
Dz_PIDFGz = Dz_PIDF*Gz;
Cz_PIDF = feedback(Dz_PIDFGz,1);
opt = stepDataOptions; opt.StepAmplitude = 80;
step(Cz_PIDF,1000,opt),grid%Plot PIDF Step Resp., Ref=80 mmHg
disp('PIDF Gain and Phase Margin:')
[Gm,Pm] = margin(Dz_PIDFGz); Gm dB = 20*log10(Gm), Pm
% disp('Unit-step performance metrics:')
PIDF_stepinfo=stepinfo(Cz_PIDF)

T = 5
KP = 0.0185
KI = 2.4205e-04
KD = 0.1930
TF = 2.8731

PIDF using Equation (8-65)
Dz_PIDF =
      0.085672 (z-0.9208) (z-0.6895)
-----
      (z-1) (z+0.7403)

Gm dB:      9.1688
Pm:         63.9993
RiseTime:   40
SettlingTime: 75
SettlingMin: 0.9201
SettlingMax: 1.0128
Overshoot:  1.2805
Undershoot: 0
Peak:       1.0128
PeakTime:   90
```

Note the further improvement in performance at this new sampling interval. The settling time is reduced from 110 to 75 seconds.

The time response of the system can also be improved by replacing the hood with a nasal cannula. Assume that this reduces the blender delay  $T_b$  from 20 s to 10 s, while also reducing the hood time constant  $\tau_H$  from 55 seconds to 2 seconds. Under these new conditions,

$$G_p(s) = \frac{90\epsilon^{-10s}}{(1+2s)(1+12s)} = \frac{(15/4)\epsilon^{-10s}}{s^2 + (7/12)s + (1/24)} \quad (12-28)$$

If we select  $T = 2$  s and model our blender delay as  $z^{-5}$ , the following MATLAB program generates a new design. Fig. 12-39 shows the pidtool results; Fig. 12-40, the step response.

```

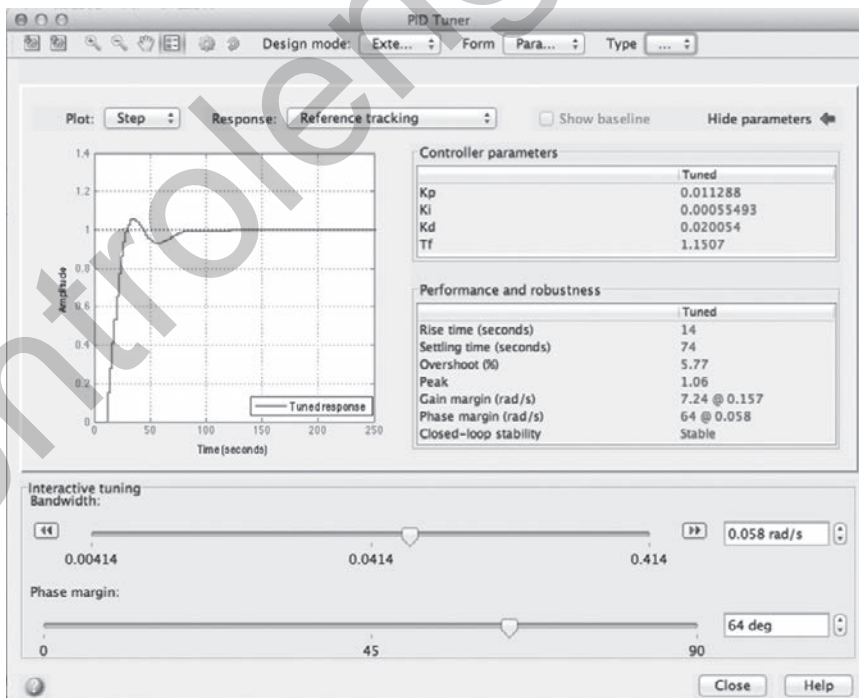
T = 2; tf([15/4],[1 7/12 1/24]); % without blender delay
delay = tf([1],[1 0 0 0 0 0],T); % blender delay = z^(-5)
Gpz = c2d(Gp,T); Gz=delay*Gpz
disp(' ')
disp('Now opening MATLAB pidtool GUI')
disp('Set Pm = 64, ww1 = 0.058') % Found by trial and error
pidtool(Gz,'pidf')

Gz =

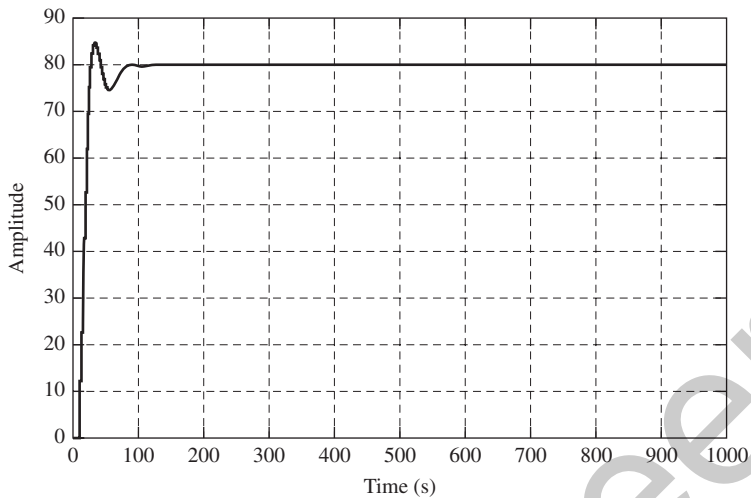
      5.202 z + 3.532
-----
      z^7 - 1.214 z^6 + 0.3114 z^5

Now opening MATLAB pidtool GUI
Set Pm = 64, ww1 = 0.058

% Gains from MATLAB pidtool with Pm = 64, ww1 = 0.058
T, KP = 0.011288, KI = 0.00055493, KD = 0.020054, TF = 1.1507
disp('PIDF using Equation (8-65)') % Find Dz using Eqn (8-65)
Dz_PIDF=KP+ KI*T)*tf([1],[1 -1],T)+KD*tf([1 -1],[TF -TF+T],T);
Dz_PIDF = zpk(Dz_PIDF), Dz_PIDFGz = Dz_PIDF*Gz;
Cz_PIDF = feedback(Dz_PIDFGz,1);
opt = stepDataOptions; opt.StepAmplitude = 80;
    
```



**FIGURE 12-39** Screen shot of MATLAB *pidtool* ( $T = 2$  s). Source: Reprinted with permission from The MathWorks, Inc.



**FIGURE 12-40** Step response for PIDF controller ( $T = 2$  s).

```
step(Cz_PIDF,1000, opt),grid %Plot PIDF Step Resp., Ref=80mmHg
disp('PIDF Gain and Phase Margin:')
[Gm,Pm] = margin(Dz_PIDFGz); Gm dB = 20*log10(Gm), Pm
% disp('Unit-step performance metrics:')
PIDF_stepinfo=stepinfo(Cz_PIDF)
```

```
T = 2
KP = 0.0113
KI = 5.5493e-04
KD = 0.0201
TF = 1.1507
```

```
PIDF using Equation (8-65)
Dz_PIDF =
0.028716 (z-0.8903) (z-0.3879)
-----
(z-1) (z+0.7381)
```

```
Gm dB: 7.2441
Pm: 63.9994
RiseTime: 14
SettlingTime: 74
SettlingMin: 0.9313
SettlingMax: 1.0577
Overshoot: 5.7672
Undershoot: 0
Peak: 1.0577
PeakTime: 34
```

Note the further improvement in performance in rise time at this new sampling interval. The rise time is reduced from 40 seconds to 14 seconds, but at the expense of an increase in overshoot from 1.3% to 5.8% and a decrease in gain margin from 9.2 dB to 7.2 dB. The settling time is essentially the same.

This case study demonstrates that finding appropriate PID gains for a specific application can be quite challenging. However, the MATLAB *pidtool* makes gain parameter tuning less time consuming.

## 12.6 TOPOLOGY IDENTIFICATION IN ELECTRIC POWER SYSTEM MODELS

Let us consider a power system network consisting of  $n$  synchronous generators connected each other by a given topology. Let the internal voltage phasor of the  $i^{\text{th}}$  machine be denoted as  $E_i \angle \delta_i$ . The electromechanical dynamics of the  $i^{\text{th}}$  generator, neglecting damping and the line resistance, can be written as:

$$\begin{aligned} \dot{\delta}_i &= \omega_i - \omega_s \\ M_i \dot{\omega}_i &= P_{mi} - \sum_{k \in N_i} \frac{E_i E_k}{x_{ik}} \sin(\delta_i - \delta_k) \end{aligned} \quad (12-29)$$

where,  $\omega_i$  (rad/s) is the angular velocity,  $P_{mi}$  is the mechanical power input per unit,  $M_i$  is the mechanical inertia,  $N_i$  is the set of all nodes that are connected to the  $i^{\text{th}}$  machine,  $x_{ik}$  is the reactance of the transmission line connecting machines  $i$  and  $k$ ,  $\delta_{ik} = \delta_i - \delta_k$ , and  $\omega_s = 120\pi$  (rad/s) is the synchronous speed for the 60 Hz system. Linearizing (12-29) about an initial equilibrium  $(\delta_{i0}, 120\pi)$  for all  $i = 1, 2, \dots, n$  results in the small signal model:

$$\begin{bmatrix} \Delta \dot{\delta} \\ \Delta \dot{\omega} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \mathbf{I} \\ \mathbf{M}^{-1} \mathbf{L} & 0 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \Delta \delta \\ \Delta \omega \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \boldsymbol{\varepsilon}_j \end{bmatrix}}_{\mathbf{B}} u \quad (12-30)$$

Here,

$$\begin{aligned} \Delta \dot{\delta} &= [\Delta \dot{\delta}_1 \quad \dots \quad \Delta \dot{\delta}_n]^T \\ \Delta \dot{\omega} &= [\Delta \dot{\omega}_1 \quad \dots \quad \Delta \dot{\omega}_n]^T \end{aligned}$$

while  $\mathbf{I}$  is the  $n$ -dimensional identity matrix, and  $\mathbf{M}$  is a diagonal matrix of the machine inertias.  $\boldsymbol{\varepsilon}_j$  is the  $j^{\text{th}}$  unit vector with all elements zero but the  $j^{\text{th}}$  element that is 1, considering that the input is modeled as a change in the mechanical power for the  $j^{\text{th}}$  machine. The input  $u$  is assumed to be an impulse at  $t = 0$ . The matrix  $\mathbf{L}$  is the  $n \times n$  Laplacian matrix whose elements are of the following form:

- If nodes  $i$  and  $j$  are connected, then  $L_{ij} = \frac{E_i E_j}{x_{ij}} \cos(\delta_{ij,0})$
- If nodes  $i$  and  $j$  are not connected, then  $L_{ij} = 0$
- For every row  $i$ ,  $L_{ii} = - \sum_{j \in N_i} \frac{E_i E_j}{x_{ij}} \cos(\delta_{ij,0})$

where,  $N_i$  is the set of indices for the generators that are directly connected to the  $i^{\text{th}}$  generator.

Next, assuming that the entire network consists of  $r$  coherent areas that are sparsely connected, and the  $i^{\text{th}}$  area consists of  $m_i$  nodes, one can then rewrite (12-30) in a block form, where the  $(i, j)^{\text{th}}$  off-diagonal block shows the connectivity between areas  $i$  and  $j$ . Its elements will be zero if two areas are not connected. Because of the clustered nature of the network, the off diagonal blocks will be much sparser compared with the diagonal blocks. Moreover,  $L_{ij}$  is a large positive number if nodes  $i$  and  $j$  are connected by a short tie-line with small reactance  $x_{ij}$ . Similarly,  $L_{ij}$  will be small positive number if nodes  $i$  and  $j$  are connected by a long transmission line with large reactance  $x_{ij}$ . As a result of this difference between the local and inter-cluster reactances, the eigenvalues of the matrix  $\mathbf{M}^{-1}\mathbf{L}$  will be divided into set of  $(r - 1)$  slow modes, while the remaining  $(n - r)$  modes will be characterized by fast eigenvalues representing the local oscillations inside areas. A standard assumption is that the  $(r - 1)$  slow modes can be attributed to  $r$  “equivalent” machines, which are connected to each other by an equivalent topology. Our objective is to find the equivalent topology that connects these  $r$  equivalent machines using measurements of  $\delta(t)$  and  $\omega(t)$ . Given these state measurements, our first task is to apply the subspace identification algorithms described in Section 11.4, identify the system matrix, compute its slow eigenvalues, and then extract the impulse response of these slow eigenvalues from the measured output. This “slow component” of the output can then be used for estimating the equivalent topology using a least squares algorithm, as described in Section 10.5.

It is straightforward to see from the model (12-30) that the topology of the reduced-order system can be identified by estimating the parameters of the reduced-order model, i.e., equivalent machine inertias and the equivalent reactances of the transmission lines. To start, one can assume the equivalent topology to be a complete graph, meaning that every area is connected to every other area with equivalent tie-line admittance  $w_{ij}$ , the equivalent inertia of the  $i^{\text{th}}$  area is  $M_i^e$ , and then estimate the values of every  $w_{ij}$  and  $M_i^e$ , using the slow components of the available measurements. The identified values for  $w_{ij}$  will determine the topology of the reduced-order network. One can also set a threshold  $w_L$  so that if  $w_{ij} < w_L$ , then  $w_{ij}$  will be set to zero, i.e., areas  $i$  and  $j$  will not be connected. We next present a linear least squares approach to estimate the parameters  $w_{ij}$  and  $M_i^e$ , for all  $i$  and  $j$ . For simplicity of notations we denote the set of all the weights  $w_{ij}$  as  $\mathbf{w}$ , and that for all the inertias  $M_i^e$  as  $\mathbf{M}^e$ .

The discrete-time representation of the reduced-order model is written as

$$\mathbf{x}(k + 1) = \mathbf{A}_d(\mathbf{w}, \mathbf{M}^e)\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}(k) \quad (12-31)$$

where  $\mathbf{A}_d$  is a function of the unknown system parameters  $(\mathbf{w}, \mathbf{M}^e)$ , and determined from the continuous-time dynamics in (12-30) via a sampling period  $T$ . The  $k^{\text{th}}$  time samples of the slow-mode dynamics are written as  $\mathbf{x}(k) = [x_1(k), \dots, x_r(k)]^T$  for  $k = 0, \dots, N$ . Thus, (12-31) can be rewritten as follows:

$$\underbrace{[\mathbf{x}(k + 1)]}_{\mathbf{G}_{k+1}} = \underbrace{\begin{bmatrix} (\mathbf{x}(k))^T & 0 & 0 & \mathbf{u}(k) & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & (\mathbf{x}(k))^T & 0 & 0 & \mathbf{u}(k) \end{bmatrix}}_{\mathbf{F}_k} \underbrace{\begin{bmatrix} A_{d11} \\ \vdots \\ A_{dnn} \\ B_{d1} \\ \vdots \\ B_{dn} \end{bmatrix}}_{\mathbf{y}} \quad (12-32)$$

The least squares formulation will be complete by making the concatenation of  $\mathbf{G}_k$ 's and  $\mathbf{F}_k$ 's by including all measurements as:

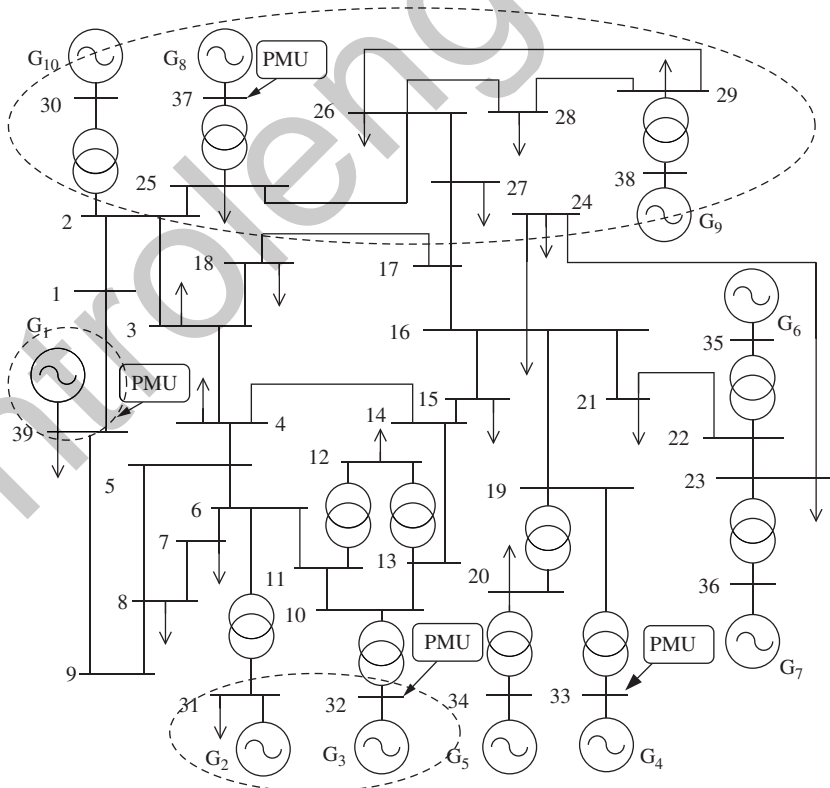
$$\min_y \left\| \mathbf{F}y - \mathbf{G} \right\|_2 \quad (12-33)$$

where  $\mathbf{F}$  and  $\mathbf{G}$  are formed as

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1 \\ \vdots \\ \mathbf{G}_m \end{bmatrix} \text{ and } \mathbf{F} = \begin{bmatrix} \mathbf{F}_0 \\ \vdots \\ \mathbf{F}_{m-1} \end{bmatrix}.$$

This linear LS problem is convex and can be solved using CVX software [26]. CVX is a MATLAB-based modeling system for convex optimization. The drawback of this formulation, however, is that  $\mathbf{F}$  is also dependent on the measurements that are usually perturbed by noise. For example, let us assume the perturbation in  $\mathbf{F}$  and  $\mathbf{G}$  are given by  $d\mathbf{F}$  and  $d\mathbf{G}$ , respectively. Then the upper-bound of deviation in estimation of  $y$ , denoted as  $dy$  is proportional to  $\kappa^2(\mathbf{F})$  where  $\kappa$  is the condition number of  $\mathbf{F}$  [27].

Next, we illustrate the LS-based identification using a case study of the IEEE 39-bus power system model. The model is shown in Fig. 12-41.

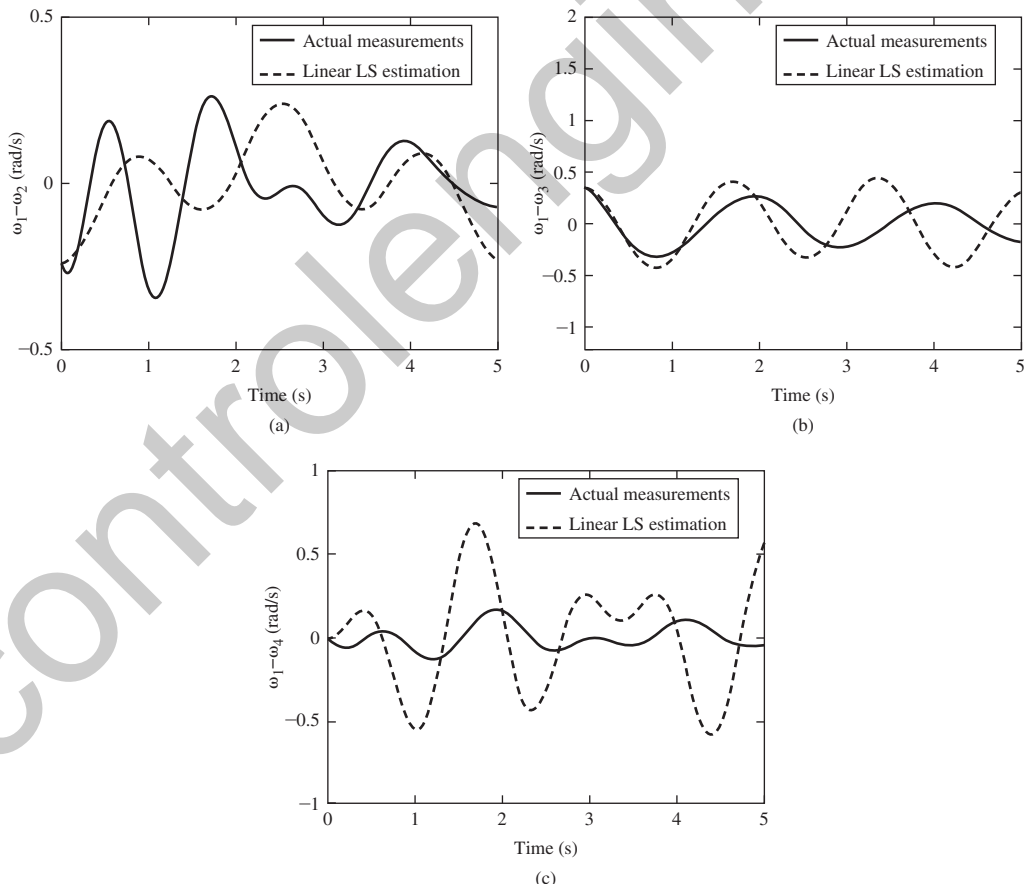


**FIGURE 12-41** Power system model with 10 generators divided into 4 areas

It has 10 generators and 39 buses, which are divided into 5 areas as shown in the figure. We consider 2<sup>nd</sup> order dynamics for all generators. We consider only five phase angle and frequency measurements at the buses marked with phasor measurement unit (PMU), meaning that a power system sensor known as PMU is installed at that bus. The disturbance of this system is modeled as a short-circuit that happens at  $t = 1$  s through  $t = 1.005$  s at bus 29. All simulations are performed in MATLAB-based Power System Analysis Toolbox (PSAT). The results of identification are shown as follows:

- Exact equivalent system matrix:

$$M^{-1}L = \begin{bmatrix} -7.9237 & 5.7529 & 0.3241 & 0.8105 & 1.0362 \\ 9.2245 & -21.3526 & 3.97 & 5.8124 & 2.3457 \\ 2.0571 & 15.7177 & -38.7316 & 15.4481 & 5.5086 \\ 2.7404 & 12.2565 & 8.2278 & -25.8022 & 2.5776 \\ 8.5838 & 12.1187 & 7.1880 & 6.3150 & -34.2054 \end{bmatrix}$$



**FIGURE 12-42** Comparison of measured output with LS-identified output.

- Identified equivalent system matrix:

$$M^{-1}L = \begin{bmatrix} -7.8459 & 4.9063 & 0.0 & 1.1716 & 1.768 \\ 9.1447 & -17.0256 & 0.0 & 7.8808 & 0.0 \\ 0.0 & 21.2502 & -44.5462 & 19.0870 & 4.2085 \\ 0.6348 & 16.7347 & 0.0 & -20.3650 & 2.9911 \\ 5.5511 & 23.5078 & 0.0005 & 9.50 & -38.5591 \end{bmatrix}$$

Fig. 12-42 compares the dynamics of the identified system using the LS algorithm with the actual slow component of the measured output.

## References

- [1] M. S. Paranjape, "Microprocessor Controller for a Servomotor System," M.S. thesis, Auburn University, Auburn, AL, 1980.
- [2] \_\_\_\_\_, *9900 Family Systems Design*. Dallas, TX: Texas Instruments Learning Center, 1978.
- [3] C. H. Roth, *Fundamentals of Logic Design*, 3d ed. St. Paul, MN: West Publishing Co., 1985.
- [4] C. L. Phillips and J. Parr, *Feedback Control Systems*, 5th ed. Upper Saddle River, NJ: Prentice-Hall, 2011.
- [5] J. D. Powell and P. Katz, "Sample Rate Selection for Aircraft Digital Control," *AIAA J.*, 13, pp. 975–979, 1975.
- [6] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3d ed. Half Moon Bay, CA: Ellis-Kagle Press, 2006.
- [7] J. C. Hsu and A. U. Meyer, *Modern Control Principles and Applications*. New York: McGraw-Hill Book Company, 1968.
- [8] A. Gelb and W. E. Vander Velde, *Multiple-Input Describing Functions and Nonlinear Design*. New York: McGraw-Hill Book Company, 1968.
- [9] R. E. Wheeler, "A Digital Control System for Plant Growth Chambers," M.S. thesis, Auburn University, Auburn, AL, 1980.
- [10] R. Boylestad and L. Nashelsky, *Electronic Devices and Circuit Theory*. Englewood Cliffs, NJ: Prentice Hall, 1978.
- [11] L. A. Leventhal, *Introduction to Microprocessors: Software, Hardware, Programming*. Englewood Cliffs, NJ: Prentice Hall, 1978.
- [12] "AN/TPN-22 Mode 1 Final Report," Contract N00039-75-C-0021, ITT Gilfillan, Van Nuys, CA, 1979.
- [13] "Software Implementation ALS Computer Program," Contract N00421-75-C-0058, Bell Aerospace Corporation, Buffalo, NY, Mar. 1975.
- [14] A. P. Schust, Jr., "Determination of Aircraft Response Characteristics in Approach/ Landing Configuration for Microwave Landing System Program," Report FT-61R-73, Naval Air Test Center, Patuxent River, MD, 1973.
- [15] C. L. Phillips, E. R. Graf, and H. T. Nagle, Jr., "MATCALC Error and Stability Analysis," Report AU-EE-75-2080-1, Auburn University, Auburn, AL, 1975.
- [16] R. F. Wigginton, "Evaluation of OPS-II Operational Program for the Automatic Carrier Landing System," Naval Electronic Systems Test and Evaluation Facility, Saint Inigoes, MD, 1971.
- [17] T. R. Benedict and G. W. Bordner, "Synthesis of an Optimal Set of Radar Track-While-Scan Smoothing Equations," *IRE Trans. Autom. Control*, Vol. AC-7, pp. 27–31, 1962.
- [18] Qifu Li, "Feedback Control of Neonatal Fractional Inspired Oxygen," ECE 592U Digital Control Systems Class Project, NC State University, May, 2003.
- [19] A. Gault, R. Sukumaran, and S. Dutta, "Digital Control of Oxygen Saturation in Infants," ECE 592 Digital Control Systems Class Project, NC State University, May, 2012.
- [20] J. C. Taube, "Adaptive control of neonatal fractional inspired oxygen," US patent 4,889,116, Dec. 26, 1989.
- [21] J. C. Taube, "Automatic Control of Neonatal Fractional Inspired Oxygen," Masters Thesis, Drexel University, 1989.
- [22] J. C. Taube, R. Pillutla, and J. Mills, "Criteria for an adaptive fractional inspired oxygen controller," Proc. of IEEE Symposium on Engineering of Computer-Based Medical Systems, pp. 129–132, 8–10 Jun 1988.



- [23] A. Sano and M. Kikucki, "Adaptive control of arterial oxygen pressure of newborn infants under incubator oxygen treatments," *Control Theory and Applications, IEE Proceedings D* 132(5), pp. 205–211, 1985.
- [24] R. Podmore, "Identification of coherent generators for dynamic equivalents." *IEEE Transactions on Power Systems* 97:1344–1354, 1978.
- [25] A. Chakraborty, J. H. Chow, and A. Salazar, "A Measurement-Based Framework for Dynamic Equivalencing of Large Power Systems Using Wide-Area Phasor Measurements," *IEEE Transactions on Smart Grid* 2(1), pp. 68–81, 2011.
- [26] CVX Research, Inc. CVX: MATLAB software for disciplined convex programming, version 2.0. <http://cvxr.com/cvx>, April 2011.
- [27] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, 1997.

Controlengineers.ir

## Design Equations

In this appendix, (8-33) is shown to be solutions to (8-31) and (8-32), by direct substitution. For convenience,  $D(j\omega_{w1})$  will be written as  $D$ , and similarly,  $G(j\omega_{w1})$  will be written as  $G$ . Then, from (8-29) and (8-33),

$$\begin{aligned}
 D &= \frac{a_0 + j\omega_{w1}a_1}{1 + j\omega_{w1}b_1} = \frac{a_0 + j \frac{1 - a_0|G| \cos \theta}{|G| \sin \theta}}{1 + j \frac{\cos \theta - a_0|G|}{\sin \theta}} \\
 &= \frac{a_0|G| \sin \theta + j(1 - a_0|G| \cos \theta)}{|G|[\sin \theta + j(\cos \theta - a_0|G|)]}
 \end{aligned} \tag{A1-1}$$

Then

$$\begin{aligned}
 |D|^2 &= \frac{a_0^2|G|^2 \sin^2 \theta + 1 - 2a_0|G| \cos \theta + a_0^2|G|^2 \cos^2 \theta}{|G|^2[\sin^2 \theta + \cos^2 \theta - 2a_0|G| \cos \theta + a_0^2|G|^2]} \\
 &= \frac{a_0^2|G|^2 + 1 - 2a_0|G| \cos \theta}{|G|^2[a_0^2|G|^2 + 1 - 2a_0|G| \cos \theta]} = \frac{1}{|G|^2}
 \end{aligned} \tag{A1-2}$$

and (8-31) is satisfied.

To show that the angle of  $D(j\omega_{w1})$  is  $\theta$  let the numerator angle and the denominator angle in (A1-1) be denoted as  $\theta_1$  and  $\theta_2$ , respectively. Then, from (A1-1),

$$\begin{aligned}
 \tan(\theta_1 - \theta_2) &= \frac{\tan \theta_1 - \tan \theta_2}{1 + \tan \theta_1 \tan \theta_2} \\
 &= \frac{\frac{1 - a_0|G| \cos \theta}{a_0|G| \sin \theta} - \frac{\cos \theta - a_0|G|}{\sin \theta}}{1 + \frac{\cos \theta - a_0|G| \cos^2 \theta - a_0|G| + a_0^2|G|^2 \cos \theta}{a_0|G| \sin^2 \theta}} \\
 &= \frac{\sin \theta[1 - 2a_0|G| \cos \theta + a_0^2|G|^2]}{\cos \theta[1 - 2a_0|G| \cos \theta + a_0^2|G|^2]} = \tan \theta
 \end{aligned} \tag{A1-3}$$

using

$$\sin^2 \theta - \cos^2 \theta = 1 - 2 \cos^2 \theta$$

Thus the angle of  $D(j\omega_{w1})$  is  $\theta$ , and (8-32) is satisfied.

## Mason's Gain Formula

In this appendix we present a technique for finding the transfer function of a system, given either the signal flow graph or the block diagram of that system.

By definition the transfer function of a continuous-time linear time-invariant (LTI) system is the ratio of the Laplace transform of the output variable to the Laplace transform of the input variable. Let  $E(s)$  be the Laplace transform of the input variable,  $C(s)$  be that of the output variable, and  $G(s)$  be the transfer function. One method of graphically denoting the relationship

$$C(s) = G(s)E(s) \quad (\text{A2-1})$$

is through a *block diagram*, as shown in Fig. A2-1(a). For the block shown, the output is equal *by definition* to the transfer function given in the block multiplied by the input. The input and the output are defined by the directions of the arrowheads, as shown.

A signal flow graph is also used to denote graphically the transfer function relationship. The signal flow graph that represents (A2-1) is given in Fig. A2-1(b). Each signal is represented by a *node* in the signal flow graph, as shown by  $E(s)$  and  $C(s)$  in the figure. Each transfer function is represented by a *branch*, shown in the figure by the line and arrowhead, with the notation for the respective transfer functions written near the arrowheads. *By definition*, the signal out of a branch is equal to the transfer function of the branch multiplied by the signal into the branch.

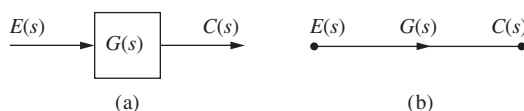
Two very important points are to be made. First, a block diagram and a signal flow graph contain exactly the same information. There is no advantage to one over the other; there is only personal preference. Second, since a block diagram (or signal flow graph) is constructed from an equation or a set of equations, it is simply a graphical representation of those equations, that is, it does not contain any extra information beyond what is available from the system equations.

One additional element is required to represent equations by a block diagram. This element is the *summing junction*, which is illustrated in Fig. A2-2 for the equation

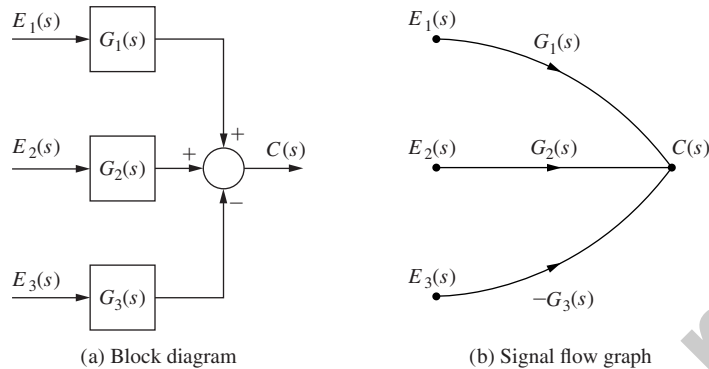
$$C(s) = G_1(s)E_1(s) + G_2(s)E_2(s) - G_3(s)E_3(s) \quad (\text{A2-2})$$

For the block diagram, the summing junction is represented by a circle, as in Fig. A2-2(a). *By definition*, the signal out of the summing junction is equal to the sum of the signals into the junction, with the sign of each component determined by the sign placed next to the arrowhead of the component. Note that whereas a summing junction can have any number of inputs, we show only one output here.

For the signal flow graph, the function of the summing junction is inherently implemented by a node. A summing junction is represented by branches into a node, as illustrated



**FIGURE A2-1** Block diagram and signal flow graph elements.



**FIGURE A2-2** Equivalent examples.

in Fig. A2-2(b). By definition, the signal at a node is equal to the sum of the signals from the branches connected to that node.

We now present a procedure that allows us to find the transfer function of either a block diagram or a signal flow graph simply by inspection. This procedure is called the *Mason's gain formula* [1]. However, even though this procedure is relatively simple, it must be used with extreme care since terms in either the numerator or the denominator of the transfer function can *easily* be overlooked. Furthermore, there is no method available that will give an indication if any of these terms have been overlooked. However, Mason's gain formula can be applied to simple systems with some confidence after one gains experience.

In this section we give the rules for Mason's gain formula as applied to signal flow graphs. Exactly the same rules apply to block diagrams. First, definitions are given as follows:

**Source node** A *source node* is a node for which signals flow *only away* from the node; that is, for the branches connected to a source node, the arrowheads are all directed away from the node.

**Sink node** A *sink node* is a node for which signals flow *only towards* the node.

**Path** A *path* is a continuous connection of branches from one node to another with all arrowheads in the same direction; that is, all signals flow in the same direction from the first node to the second one, from the second to the third, and so on until it reaches the final node of that path.

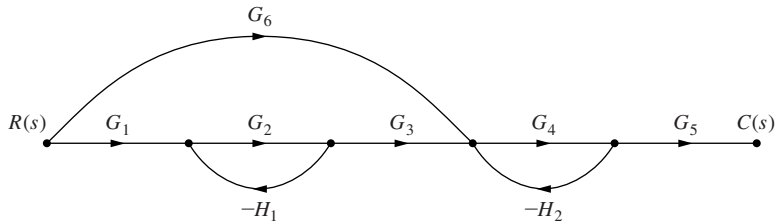
**Loop** A *loop* is a closed path (with all arrowheads in the same direction) in which no node is encountered more than once. Note that a source node cannot be a part of a loop, since each node in the loop must have at least one branch into the node and at least one branch out.

**Forward path** A *forward path* is a path that connects a source node to a sink node, in which no node is encountered more than once.

**Path gain** The *path gain* is the product of the transfer functions of all branches that form the path.

**Loop gain** The *loop gain* is the product of the transfer functions of all branches that form the loop.

**Nontouching loops** Two loops are *nontouching* if these loops have no nodes in common. A loop and a path are nontouching if they have no nodes in common.



**FIGURE A2-3** Signal flow graph.

All these definitions are illustrated using the signal flow graph of Fig. A2-3. There are two loops in the flow graph, one with a gain of  $-G_2H_1$  and the other with a gain of  $-G_4H_2$ . Note that these two loops do not touch. In addition, there are two forward paths connecting the input  $R(s)$  and the output  $C(s)$ . One of these forward paths has a gain of  $G_1G_2G_3G_4G_5$ , and the other has a gain of  $G_6G_4G_5$ . Note that the forward path  $G_6G_4G_5$  does not touch the loop  $-G_2H_1$  but does touch the other loop. The path  $G_1G_2G_3G_4G_5$  touches both loops.

Based on the preceding definitions, we may now state Mason's gain formula. The formula gives the transfer function from a source (input) node to a sink (output) node *only*, and may be stated as

$$T = \frac{1}{\Delta} \sum_{k=1}^p M_k \Delta_k = \frac{1}{\Delta} (M_1 \Delta_1 + M_2 \Delta_2 + \dots + M_p \Delta_p) \quad (\text{A2-3})$$

where  $T$  is the gain (transfer function) from the input node to the output node,  $p$  is the number of forward paths, and

$$\begin{aligned} \Delta = & 1 - (\text{sum of all individual loop gains}) \\ & + (\text{sum of the products of the loop gains of all possible combinations of nontouching loops taken two at a time}) \\ & - (\text{sum of the products of the loop gains of all possible combinations of nontouching loops taken three at a time}) \\ & + (\text{sum of the products of the loop gains of all possible combinations of nontouching loops taken four at a time}) \\ & - (\dots) \end{aligned}$$

$M_k$  = path sign of the  $k$ th forward path

$\Delta_k$  = value of  $\Delta$  for that part of the flow graph not touching the  $k$ th forward path.

An example is now given to illustrate the use of Mason's gain formula.

### EXAMPLE A2-1

Consider again the system of Fig. A2-3. Let  $L_i$  be the gain of the  $i$ th loop. Then the gains of the only two loops can be written as

$$L_1 = -G_2H_1; \quad L_2 = -G_4H_2$$

Two forward paths are present in Fig. A2-3; the forward path gains can be expressed as

$$M_1 = G_1G_2G_3G_4G_5; \quad M_2 = G_6G_4G_5$$

The value of  $\Delta$  can be written directly from Fig. A2-3:

$$\Delta = 1 - (L_1 + L_2) + L_1L_2 = 1 + G_2H_1 + G_4H_2 + G_2G_4H_1H_2$$

The last term is present in this equation since the two loops do not touch; that is, the loops have no nodes in common.

The determination of the  $\Delta_k$  of (A2-3) is more difficult. As just stated, the value of  $\Delta_1$  is the value of  $\Delta$  for that part of the flow graph not touching the first forward path. One method of evaluating  $\Delta_1$  is to redraw the flow graph with the first forward path removed. Of course, all nodes of the first forward path must also be removed. Fig. A2-4(a) gives the results of removing the first forward path. The second part of the figure illustrates removing the second forward path. Hence  $\Delta_1$  is simply the  $\Delta$  of the flow graph of Fig. A2-4(a), and  $\Delta_2$  is that of Fig. A2-4(b). Thus we can write

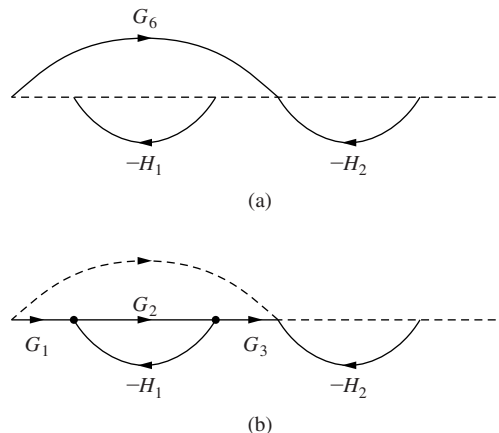
$$\Delta_1 = 1; \quad \Delta_2 = 1 - (-G_2H_1)$$

since Fig. A2-4(a) has no loops and Fig. A2-4(b) has one loop. Thus from (A2-3) we can write the transfer function of the system of Fig. A2-3 as

$$T = \frac{M_1\Delta_1 + M_2\Delta_2}{\Delta} = \frac{G_1G_2G_3G_4G_5 + G_6G_4G_5(1 + G_2H_1)}{1 + G_2H_1 + G_4H_2 + G_2G_4H_1H_2}$$

Note that even for this relatively simple flow graph, many terms appear in the transfer function. Hence terms can easily be overlooked. The only methods available to verify the results of Mason's gain formula are algebraic methods such as Cramer's rule [2].

As stated earlier, Mason's gain formula can be applied directly to block diagrams. In the definitions which appear previously in this appendix, the following replacements are made, with the term *signal* defined as any input or any output of a block or a summing junction.



**FIGURE A2-4** Flow graphs with forward path removed.

Signal flow graph		Block diagram
input node	→	input signal
output node	→	output signal
branch	→	block
node	→	signal

Note that we can identify any internal signal as the output signal.

---

## References

- [1] S. J. Mason, "Feedback Theory: Some Properties of Flow Graphs," *Proc. IRE* 41, pp. 1144–1156, 1953.
- [2] C. L. Phillips and R. D. Harbor, *Feedback Control Systems*, 2d ed. Englewood Cliffs, NJ: Prentice Hall, 1991.

Controlengineers.ir

## Evaluation of $E^*(s)$

$E^*(s)$ , as defined by (3-7), has limited usefulness in an analysis because it is expressed as an infinite series. However, for many useful time functions  $E^*(s)$  can be expressed in closed form. This closed form of  $E^*(s)$  will now be derived.

From (3-6),

$$e^*(t) = e(t)\delta_T(t) \quad (\text{A3-1})$$

However, this function is the inverse Laplace transform of  $E^*(s)$  only for cases in which  $e(t)$  is continuous at all sampling instants. Problems arise if  $e(t)$  is discontinuous at any sampling instant, since the inverse Laplace transform evaluated at a discontinuity will give the average value of the discontinuity. As defined in Section 3.3, however, if  $e(t)$  is discontinuous at a sampling instant, then at that instant  $E^*(s)$  assumes the value of  $e(t)$  from the right. For example, if  $e(t)$  is discontinuous at the origin, the calculation of  $E^*(s)$  from (A3-1) would yield a value for the function at  $t = 0$  of  $e(0)/2$ . Thus, if  $e(0) \neq 0$ , (A3-1) must be expressed as

$$e^*(t) = e(t)\delta_T(t) + e(0)\delta(t) \quad (\text{A3-2})$$

If  $e(t)$  is discontinuous at other sampling instants, then impulse functions with values  $\Delta e(kT)\delta(t - kT)$  must be added to (A3-2), where  $\Delta e(kT)$  is the amplitude of the discontinuity of  $e(t)$  at  $t = kT$ ; that is

$$\Delta e(kT) = e(kT^+) - e(kT^-)$$

where  $e(kT^-) = e(t)$  evaluated at  $t = kT - \varepsilon$ , and where  $\varepsilon$  is arbitrarily small.

The following derivation applies for the case in which  $e(t)$  is continuous at all sampling instants. From (A3-1),

$$E^*(s) = E(s) * \Delta_T(s) \quad (\text{A3-3})$$

where  $*$  denotes complex convolution [1] and  $\Delta_T(s)$  is the Laplace transform of  $\delta_T(t)$ . From (3-5),

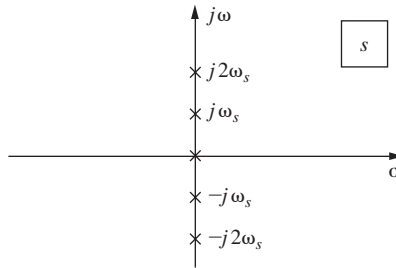
$$\Delta_T(s) = 1 + \varepsilon^{-Ts} + \varepsilon^{-2Ts} + \dots = \frac{1}{1 - \varepsilon^{-Ts}} \quad (\text{A3-4})$$

Therefore, the poles of  $\Delta_T(s)$  occur at values of  $s$  for which

$$\varepsilon^{-Ts} = 1 \quad (\text{A3-5})$$

Equation (A3-5) is satisfied for  $s = j(2\pi n/T) = jn\omega_s$ ,  $n = 0, \pm 1, \pm 2, \dots$ , where  $\omega_s$  is the sampling frequency expressed in radians per second. The poles of  $\Delta_T(s)$  are shown on the  $s$ -plane plot of Fig. A3-1.





**FIGURE A3-1** Poles of  $\Delta_T(s)$ .

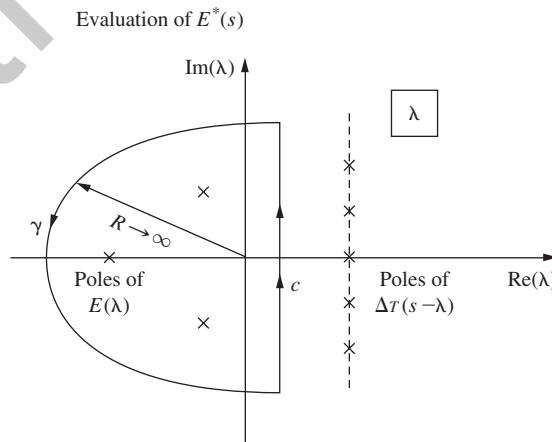
By definition, equation (A3-3) can be expressed as

$$\begin{aligned}
 E^*(s) &= \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} E(\lambda) \Delta_T(s - \lambda) d\lambda \\
 &= \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} E(\lambda) \frac{1}{1 - e^{-T(s-\lambda)}} d\lambda
 \end{aligned} \tag{A3-6}$$

where the poles of the integrand of (A3-6) occur as shown in Fig. A3-2. The value of  $c$  must be chosen such that the poles of  $E(\lambda)$  are to the left of the path of integration, and the value of  $s$  must be selected so that the poles of  $\Delta_T(s - \lambda)$  are to the right of the path of integration [1]. An examination of (A3-6) and Fig. A3-2 indicates that  $E^*(s)$  can be expressed as

$$E^*(s) = \frac{1}{2\pi j} \oint_{\gamma} E(\lambda) \Delta_T(s - \lambda) d\lambda - \frac{1}{2\pi j} \int E(\lambda) \Delta_T(s - \lambda) d\lambda \tag{A3-7}$$

Consider the second integral of (A3-7). This term is zero if  $\lim_{\lambda \rightarrow \infty} \lambda E(\lambda) = 0$  [i.e., if  $e(0) = 0$ ]. If  $e(0)$  is not zero, the second integral can be shown to be equal to  $e(0)/2$  following [2]. However,



**FIGURE A3-2** Pole locations for the integrand of (A3-6).

from (A3-2), if  $e(0)$  is not zero, an additional term equal to  $e(0)/2$  must be added to (A3-7). Therefore, in either case,

$$E^*(s) = \frac{1}{2\pi j} \oint_{\gamma} E(\lambda) \Delta_T(s - \lambda) d\lambda \quad (\text{A3-8})$$

The reader interested in a more detailed mathematical development of (A3-8) is referred to [1], [2], and [3].

The theorem of residues [3] can be used to evaluate (A3-8). We next state this theorem for a more pedagogical construction of (A3-8).

**Theorem of Residues.** If  $C$  is a closed curve and if  $f(z)$  is analytic within and on  $C$  except at finite number of singular points in the interior of  $C$ , then

$$\oint_C f(z) dz = 2\pi j [r_1 + r_2 + \dots + r_n]$$

where  $r_1, r_2, \dots, r_n$  are the residues of  $f(z)$  at the singular points within  $C$ .

Using this theorem, we can express (A3-8) as

$$E^*(s) = \sum_{\substack{\text{at poles} \\ \text{of } E(\lambda)}} \left[ \text{residues of } E(\lambda) \frac{1}{1 - \epsilon^{-T(s-\lambda)}} \right] \quad (\text{A3-9})$$

Recall that the residues are evaluated at the poles of the function  $E(\lambda)$ , since as shown in Fig. A3-2, the singular points that lie within the closed contour are derived from this function. For the case in which  $E(s)$  has only simple poles, we may use (2-33). Or, by letting

$$E(s) = \frac{N(s)}{D(s)} \quad (\text{A3-10})$$

where  $N(s)$  and  $D(s)$  are polynomials in  $s$ , (A3-9) can be expressed as [4]

$$E^*(s) = \sum_n \left[ \frac{N(\lambda_n)}{D'(\lambda_n)} \frac{1}{1 - \epsilon^{-T(s-\lambda_n)}} \right] \quad (\text{A3-11})$$

where  $\lambda_n$  are the locations of simple poles of  $E(\lambda)$  and

$$D'(\lambda) = \frac{dD(\lambda)}{d\lambda}$$

For multiple-ordered poles, the residue may be found using the expression illustrated in (2-34).

It is of interest to consider the case in which the function  $e(t)$  contains a time delay. For example, consider a delayed signal of the type

$$e(t) = e_1(t - t_0) u(t - t_0) \quad (\text{A3-12})$$

Then

$$E(s) = \epsilon^{-t_0 s} \mathcal{L}[e_1(t)] = \epsilon^{-t_0 s} E_1(s) \quad (\text{A3-13})$$

For this case, in general  $\lim_{\lambda \rightarrow \infty} \lambda E(\lambda)$  is not finite in the second integral in (A3-7) (see Fig. A3-2), and thus (A3-9) does not apply. Special techniques are required to find the starred transform of a

delayed signal in closed form, and these techniques are presented in Chapter 4, where the modified  $z$ -transform is developed. However, for the special case in which the time signal is delayed a whole number of sampling periods, (A3-9) can be applied in a slightly different form,

$$[\varepsilon^{-kTs}E_1(s)]^* = \varepsilon^{-kTs} \sum_{\text{at poles of } E_1(\lambda)} \left[ \text{residues of } E_1(\lambda) \frac{1}{1 - \varepsilon^{-T(s-\lambda)}} \right] \quad (\text{A3-14})$$

where  $k$  is a positive integer.

Equation (A3-6) can also be evaluated using the path  $\alpha$  shown in Fig. A3-3. In this case, the poles of  $\Delta_T(s - \lambda)$  are enclosed by  $\alpha$ . For the case in which  $\lim_{\lambda \rightarrow \infty} \lambda E(\lambda)$  is zero, the integral around the infinite semicircular portion of  $\alpha$  is zero and

$$E^*(s) = - \sum_{\text{poles of } \Delta_T(s-\lambda)} [\text{residues of } E(\lambda)\Delta_T(s - \lambda)] \quad (\text{A3-15})$$

This relationship is derived using the same steps that were used in the derivation of (A3-9). Recall that

$$\Delta_T(s - \lambda) = \frac{1}{1 - \varepsilon^{-T(s-\lambda)}} = \frac{N(\lambda)}{D(\lambda)} \quad (\text{A3-16})$$

has simple poles located at periodic intervals along a fixed line in the  $\lambda$ -plane as shown in Fig. A3-3. Therefore,  $E^*(s)$  can be expressed as

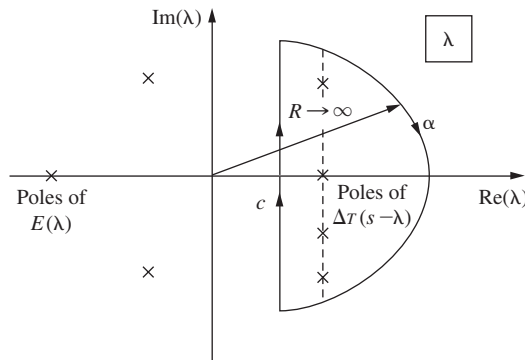
$$E^*(s) = - \sum_{n=-\infty}^{\infty} \frac{N(\lambda_n)}{D'(\lambda_n)} E(\lambda_n) \quad (\text{A3-17})$$

The poles of  $\Delta_T(s - \lambda)$  occur at

$$s - \lambda_n = j \frac{2\pi n}{T} = j\omega_s, \quad n = 0, \pm 1, \pm 2, \dots \quad (\text{A3-18})$$

Solving for  $\lambda_n$ ,

$$\lambda_n = s - j\omega_s, \quad n = 0, \pm 1, \pm 2, \dots \quad (\text{A3-19})$$



**FIGURE A3-3** Integration path in the  $\lambda$ -plane.

Now in this case

$$D'(\lambda) = -T\varepsilon^{-T(s-\lambda)} \quad (\text{A3-20})$$

Thus

$$D'(\lambda_n) = -T\varepsilon^{-j2\pi n} = -T \quad (\text{A3-21})$$

Then using (A3-16) and (A3-21), (A3-17) can be expressed as

$$E^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) \quad (\text{A3-22})$$

For the case in which  $\lim_{\lambda \rightarrow \infty} E(\lambda) = 0$ , but  $e(0) \neq 0$ , the integral around the infinite semi-circular portion of  $\alpha$  is also zero. However, because of the additional term in (A3-2), (A3-22) becomes

$$E^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) + \frac{e(0^+)}{2} \quad (\text{A3-23})$$

Therefore, the general expression for (A3-23) is

$$E^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) + \frac{1}{2} \sum_{n=0}^{\infty} \Delta e(nT) \varepsilon^{-nTs} \quad (\text{A3-24})$$

where  $\Delta e(nT)$  is the amplitude of the discontinuity of  $e(t)$  at  $t = nT$ .

In summary, there are three expressions of  $E^*(s)$ . These are:

$$E^*(s) = \sum_{n=0}^{\infty} e(nT) \varepsilon^{-nTs} \quad (\text{A3-25})$$

$$E^*(s) = \sum_{\substack{\text{at poles} \\ \text{of } E(\lambda)}} \left[ \text{residues of } E(\lambda) \frac{1}{1 - \varepsilon^{-T(s-\lambda)}} \right] \quad (\text{A3-26})$$

$$E^*(s) = \frac{1}{T} \sum_{n=-\infty}^{\infty} E(s + jn\omega_s) + \frac{1}{2} \sum_{n=0}^{\infty} \Delta e(nT) \varepsilon^{-nTs} \quad (\text{A3-27})$$

Equation (A3-25) is the defining equation for  $E^*(s)$ , and (A3-26) and (A3-27) are derived from (A3-25).

## References

- [1] M. F. Gardner and J. L. Barnes, *Transients in Linear Systems*, Vol. I. New York: John Wiley & Sons, Inc., 1942.
- [2] C. L. Phillips, D. L. Chenoweth, and R. K. Cavin III, "z-Transform Analysis of Sampled-Data Control Systems without Reference to Impulse Functions," *IEEE Trans. Educ.* E-11, pp. 141–144, 1968.
- [3] C. R. Wylie, Jr., *Advanced Engineering Mathematics*, 4th ed. New York: McGraw-Hill Book Company, 1975.
- [4] E. A. Guillemin, *The Mathematics of Circuit Analysis*. New York: John Wiley & Sons, Inc., 1949.

## Review of Matrices

Presented in this appendix is a brief review of matrices. Those readers interested in more depth are referred to [1] through [5].

The study of matrices originated from solving linear equations. As an example, consider the equations

$$\begin{aligned}
 x_1 + x_2 + x_3 &= 3 \\
 x_1 + x_2 - x_3 &= 1 \\
 2x_1 + x_2 + 3x_3 &= 6
 \end{aligned} \tag{A4-1}$$

In a *vector-matrix* format we write these equations as

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 2 & 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 6 \end{bmatrix} \tag{A4-2}$$

We *define* the following:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 2 & 1 & 3 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} 3 \\ 1 \\ 6 \end{bmatrix} \tag{A4-3}$$

Then (A4-2) may be expressed as

$$\mathbf{Ax} = \mathbf{u} \tag{A4-4}$$

In this equation  $\mathbf{A}$  is a  $3 \times 3$  (3 rows, 3 columns) *matrix*,  $\mathbf{x}$  is a  $3 \times 1$  *matrix*, and  $\mathbf{u}$  is a  $3 \times 1$  *matrix*. Usually matrices that contain only one row or only one column are called *vectors*. A matrix of only one row and one column is a *scalar*. In (A4-1),  $x_1$  is a scalar.

The general matrix  $\mathbf{A}$  is written as

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = [a_{ij}] \tag{A4-5}$$

This matrix has  $m$  rows and  $n$  columns, and thus is an  $m \times n$  matrix. The element  $a_{ij}$  is the element common to the  $i$ th row and  $j$ th column. If  $m = n$ , the matrix is said to be square.

Some special definitions are now given.

**Identity matrix.** The identity matrix is an  $n \times n$  (square) matrix with all main diagonal elements equal to 1 and all off-diagonal elements equal to zero. For example, the  $3 \times 3$  identity matrix is

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A4-6})$$

If  $\mathbf{A}$  is also  $n \times n$ ,

$$\mathbf{AI} = \mathbf{IA} = \mathbf{A} \quad (\text{A4-7})$$

where the multiplication of matrices is defined below.

**Diagonal matrix.** A diagonal matrix is an  $n \times n$  matrix with all off-diagonal elements equal to zero.

$$\mathbf{D} = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \quad (\text{A4-8})$$

**Symmetric matrix.** The square matrix  $\mathbf{A}$  is symmetric if  $a_{ij} = a_{ji}$ , for all  $i$  and  $j$ .

**Transpose of a matrix.** To take the transpose of a matrix, interchange rows and columns. For example,

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 2 & 1 & 3 \end{bmatrix}, \quad \mathbf{A}^T = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 1 & 1 \\ 1 & -1 & 3 \end{bmatrix} \quad (\text{A4-9})$$

where  $\mathbf{A}^T$  denotes the transpose of  $\mathbf{A}$ . A property of the transpose is

$$(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T \quad (\text{A4-10})$$

Consider the partitioned matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{D} & \mathbf{E} \\ \mathbf{F} & \mathbf{G} \end{bmatrix} \quad (\text{A4-11})$$

where  $\mathbf{D}$ ,  $\mathbf{E}$ ,  $\mathbf{F}$ , and  $\mathbf{G}$  are each  $n \times n$ . Then  $\mathbf{H}$  is  $2n \times 2n$ . The transpose of  $\mathbf{H}$  is given as

$$\mathbf{H}^T = \begin{bmatrix} \mathbf{D}^T & \mathbf{F}^T \\ \mathbf{E}^T & \mathbf{G}^T \end{bmatrix} \quad (\text{A4-12})$$

**Trace.** The trace of a matrix is equal to the sum of its diagonal elements. Given an  $n \times n$  matrix  $\mathbf{A}$ ,

$$\text{trace of } \mathbf{A} = \text{tr } \mathbf{A} = a_{11} + a_{22} + \cdots + a_{nn} \quad (\text{A4-13})$$

**Eigenvalues.** The *eigenvalues* (characteristic values) of a square matrix  $\mathbf{A}$  are the roots of the polynomial equation

$$|\lambda \mathbf{I} - \mathbf{A}| = 0 \quad (\text{A4-14})$$

where  $|\cdot|$  denotes the determinant, and  $\lambda$  is a scalar.

**Eigenvectors.** The *eigenvectors* (characteristic vectors) of a square matrix  $\mathbf{A}$  are the vectors  $\mathbf{x}_i$  that satisfy the equation

$$\lambda_i \mathbf{x}_i = \mathbf{A} \mathbf{x}_i \quad (\text{A4-15})$$

where  $\lambda_i$  are the eigenvalues of  $\mathbf{A}$ .

**Properties.** Two properties of an  $n \times n$  matrix  $\mathbf{A}$  are

$$|\mathbf{A}| = \prod_{i=1}^n \lambda_i \quad (\text{A4-16})$$

$$\text{tr } \mathbf{A} = \sum_{i=1}^n \lambda_i \quad (\text{A4-17})$$

Given the partitioned  $2n \times 2n$  matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{D} & \mathbf{E} \\ \mathbf{O} & \mathbf{G} \end{bmatrix} \quad (\text{A4-18})$$

where  $\mathbf{D}$ ,  $\mathbf{E}$ , and  $\mathbf{G}$  are  $n \times n$ , and  $\mathbf{O}$  is the  $n \times n$  null matrix. Then

$$|\mathbf{H}| = |\mathbf{D}| |\mathbf{G}| \quad (\text{A4-19})$$

With  $\mathbf{A}$  and  $\mathbf{B}$  square,

$$|\mathbf{AB}| = |\mathbf{A}| |\mathbf{B}| \quad (\text{A4-20})$$

**Minor.** The minor  $m_{ij}$  of element  $a_{ij}$  of a square matrix  $\mathbf{A}$  is the determinant of the array remaining when the  $i$ th row and  $j$ th column are deleted from  $\mathbf{A}$ . For example,  $m_{21}$  for  $\mathbf{A}$  of (A4-3) is

$$m_{21} = \begin{vmatrix} 1 & 1 \\ 1 & 3 \end{vmatrix} = 3 - 1 = 2 \quad (\text{A4-21})$$

**Cofactor.** The cofactor  $c_{ij}$  of element  $a_{ij}$  of the matrix  $\mathbf{A}$  is given by

$$c_{ij} = (-1)^{i+j} m_{ij} \quad (\text{A4-22})$$

For (A4-21),

$$c_{21} = (-1)^{2+1} (2) = -2 \quad (\text{A4-23})$$

**Adjoint.** The matrix of cofactors, when transposed, is called the adjoint of  $\mathbf{A}$  ( $\text{adj } \mathbf{A}$ ). For  $\mathbf{A}$  of (A4-3),

$$\text{adj } \mathbf{A} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}^T = \begin{bmatrix} 4 & -5 & -1 \\ -2 & 1 & 1 \\ -2 & 2 & 0 \end{bmatrix}^T \quad (\text{A4-24})$$

**Inverse.** The inverse of  $\mathbf{A}$  is given by

$$\mathbf{A}^{-1} = \frac{\text{adj } \mathbf{A}}{|\mathbf{A}|} \quad (\text{A4-25})$$

where  $\mathbf{A}^{-1}$  denotes the inverse of  $\mathbf{A}$  and  $|\mathbf{A}|$  denotes the determinant of  $\mathbf{A}$ . For  $\mathbf{A}$  of (A4-3) and (A4-24)

$$|\mathbf{A}| = -2$$

and

$$\mathbf{A}^{-1} = \begin{bmatrix} -2 & 1 & 1 \\ \frac{5}{2} & -\frac{1}{2} & -1 \\ \frac{1}{2} & -\frac{1}{2} & 0 \end{bmatrix} \quad (\text{A4-26})$$

Two properties of matrix inverses are

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I} \quad (\text{A4-27})$$

$$(\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1} \quad (\text{A4-28})$$

Note that the matrix inverse is defined only for a square matrix and exists only if the determinant of the matrix is nonzero. If  $\mathbf{A}$  has an inverse, so does  $\mathbf{A}^{-1}$ , with  $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$ . For  $\mathbf{A}$  square and  $|\mathbf{A}| \neq 0$ ,

$$(\mathbf{A}^{-1})^T = (\mathbf{A}^T)^{-1} = \mathbf{A}^{-T} \quad (\text{A4-29})$$

$$|\mathbf{A}^{-1}| = \frac{1}{|\mathbf{A}|} \quad (\text{A4-30})$$

**A useful determinant.** Given the partitioned matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{D} & \mathbf{E} \\ \mathbf{F} & \mathbf{G} \end{bmatrix} \quad (\text{A4-31})$$

where  $\mathbf{D}$ ,  $\mathbf{E}$ ,  $\mathbf{F}$ , and  $\mathbf{G}$  are each  $n \times n$ . Then  $\mathbf{H}$  is  $2n \times 2n$ . The determinant of  $\mathbf{H}$  is given by [6]

$$|\mathbf{H}| = |\mathbf{G}||\mathbf{D} - \mathbf{E}\mathbf{G}^{-1}\mathbf{F}| = |\mathbf{D}||\mathbf{G} - \mathbf{F}\mathbf{D}^{-1}\mathbf{E}| \quad (\text{A4-32})$$

provided that the indicated inverse matrices exist.

**The matrix inversion lemma.** The matrix inversion lemma is [6]

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1} \quad (\text{A4-33})$$

provided that the indicated inverses exist.



## ALGEBRA OF MATRICES

The algebra of matrices must be defined such that the operations indicated in (A4-2), and any additional operation we may wish to perform, lead us back to (A4-1).

**Addition.** To form the sum of matrices **A** and **B**, we add corresponding elements  $a_{ij}$  and  $b_{ij}$ , for each  $i$  and  $j$ . For example,

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix} \quad (\text{A4-34})$$

**Multiplication by a scalar.** To multiply a matrix **A** by a scalar  $k$ , multiply each element of **A** by  $k$ .

**Multiplication of vectors.** The multiplication of the  $1 \times n$  (row) vector with an  $n \times 1$  (column) vector is defined as

$$[x_1 \ x_2 \ \cdots \ x_n] \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = x_1y_1 + x_2y_2 + \cdots + x_ny_n \quad (\text{A4-35})$$

**Multiplication of matrices.** An  $n \times p$  matrix **A** may be multiplied by only a  $p \times m$  matrix **B**; that is, the number of columns of **A** must equal the number of rows of **B**. Let

$$\mathbf{AB} = \mathbf{C}$$

Then the  $ij$ th element of **C** is equal to the multiplication (as vectors) of the  $i$ th row of **A** with the  $j$ th column of **B**. As an example, consider the product  $\mathbf{AA}^{-1}$  from (A4-3) and (A4-26).

$$\begin{aligned} \mathbf{AA}^{-1} &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 2 & 1 & 3 \end{bmatrix} \begin{bmatrix} -2 & 1 & 1 \\ \frac{5}{2} & -\frac{1}{2} & -1 \\ \frac{1}{2} & -\frac{1}{2} & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{I} \end{aligned} \quad (\text{A4-36})$$

## OTHER RELATIONSHIPS

Other important matrix relationships will now be given.

**Differentiation.** The derivative of a matrix is obtained by differentiating the matrix element by element. For example, let

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (\text{A4-37})$$

Then

$$\frac{d\mathbf{x}}{dt} = \begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{bmatrix} \quad (\text{A4-38})$$

**Integration.** The integral of a matrix is obtained by integrating the matrix element by element. In (A4-37),

$$\int \mathbf{x} dt = \begin{bmatrix} \int x_1 dt \\ \int x_2 dt \end{bmatrix} \quad (\text{A4-39})$$

**A property.** Consider the scalar  $G$ :

$$G = \mathbf{x}^T \mathbf{y} = \mathbf{y}^T \mathbf{x} = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n \quad (\text{A4-40})$$

Then, by definition,

$$\frac{\partial G}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial G}{\partial x_1} \\ \vdots \\ \frac{\partial G}{\partial x_n} \end{bmatrix} = \mathbf{y}, \quad \frac{\partial G}{\partial \mathbf{y}} = \mathbf{x} \quad (\text{A4-41})$$

**Quadratic forms.** The scalar

$$F = \mathbf{x}^T \mathbf{Q} \mathbf{x} \quad (\text{A4-42})$$

is called a *quadratic form*. For example, if  $\mathbf{x}$  is second order,

$$\begin{aligned} F = \mathbf{x}^T \mathbf{Q} \mathbf{x} &= [x_1 \quad x_2] \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= q_{11} x_1^2 + (q_{12} + q_{21}) x_1 x_2 + q_{22} x_2^2 \end{aligned} \quad (\text{A4-43})$$

Hence

$$\mathbf{x}^T \mathbf{Q} \mathbf{x} = \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j \quad (\text{A4-44})$$

Note that  $\mathbf{Q}$  can be assumed symmetric with no loss of generality. Now

$$\frac{\partial F}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial F}{\partial x_1} \\ \frac{\partial F}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2q_{11}x_1 + (q_{12} + q_{21})x_2 \\ (q_{12} + q_{21})x_1 + 2q_{22}x_2 \end{bmatrix} = 2\mathbf{Q}\mathbf{x} \quad (\text{A4-45})$$

**Bilinear forms.** For two vectors  $\mathbf{x}^T$  and  $\mathbf{y}$ , the scalar

$$G = \mathbf{x}^T \mathbf{Q} \mathbf{y} \quad (\text{A4-46})$$

is called a *bilinear form*. For example, for  $\mathbf{x}$  and  $\mathbf{y}$  second order,

$$\begin{aligned} G = \mathbf{x}^T \mathbf{Q} \mathbf{y} &= [x_1 \quad x_2] \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\ &= q_{11}x_1y_1 + q_{12}x_1y_2 + q_{21}x_2y_1 + q_{22}x_2y_2 \end{aligned} \quad (\text{A4-47})$$

Hence

$$\mathbf{x}^T \mathbf{Q} \mathbf{y} = \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i y_j \quad (\text{A4-48})$$

Note that  $\mathbf{Q}$  cannot be assumed symmetric. Then

$$\frac{\partial G}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial G}{\partial x_1} \\ \frac{\partial G}{\partial x_2} \end{bmatrix} = \begin{bmatrix} q_{11}y_1 + q_{12}y_2 \\ q_{21}y_1 + q_{22}y_2 \end{bmatrix} = \mathbf{Q} \mathbf{y} \quad (\text{A4-49})$$

and

$$\frac{\partial G}{\partial \mathbf{y}} = \begin{bmatrix} \frac{\partial G}{\partial y_1} \\ \frac{\partial G}{\partial y_2} \end{bmatrix} = \begin{bmatrix} q_{11}x_1 + q_{12}x_2 \\ q_{21}x_1 + q_{22}x_2 \end{bmatrix} = \mathbf{Q}^T \mathbf{x} \quad (\text{A4-50})$$

Note that these relationships are directly evident from (A4-40) and (A4-41). For example, in (A4-40), replace  $\mathbf{y}$  with  $\mathbf{Q} \mathbf{y}$ , and (A4-49) is obtained from (A4-41).

**Sign definiteness.** If the scalar  $F = \mathbf{x}^T \mathbf{Q} \mathbf{x} > 0$  for all  $\mathbf{x} \neq 0$ , the quadratic form is *positive definite*. If  $F \geq 0$  for all  $\mathbf{x}$ , the quadratic form is *positive semidefinite*. One test for sign definiteness is given in Section 10.2. A second test is that the principal minors  $\Delta_i$ ,  $i = 1, 2, \dots, n$ , are all positive (nonnegative) for  $F$  positive definite (positive semidefinite), where

$$\Delta_1 = q_{11}, \quad \Delta_2 = \begin{vmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{vmatrix}, \dots, \quad \Delta_n = |\mathbf{Q}| \quad (\text{A4-51})$$

## References

- [1] F. R. Gantmacher, *Theory of Matrices*, Vols. I and II. New York: Chelsea Publishing Company, Inc., 1959.
- [2] P. M. DeRusso, R. J. Roy, and C. M. Close, *State Variables for Engineers*. New York: John Wiley & Sons, Inc., 1965.
- [3] K. Ogata, *Modern Control Engineering*, 2d ed. New York: McGraw-Hill Book Company, 1990.
- [4] G. Strang, *Linear Algebra and Its Applications*. New York: Academic Press, Inc., 1976.
- [5] G. H. Golub and C. F. Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins University Press, 1983.
- [6] T. E. Fortman, "A Matrix Inversion Identity," *IEEE Trans. Autom. Control* AC-15, p. 599, 1970.

## The Laplace Transform

A brief review of the Laplace transform is presented in this appendix. We will see that the Laplace transform is useful in the modeling of a linear time-invariant analog system as a transfer function. The Laplace transform may also be used to solve for the free and forced responses of this type of system; however, these days one generally uses *simulations*, i.e., solutions to the system equations using computer programs for solving such responses, especially if the system size is large. For those readers who are interested in more details of the Laplace transform, [1] [2], and [3] are suggested for supplemental reading.

### INTRODUCTION

By definition, the Laplace transform of a function of time  $f(t)$  is [1]

$$F(s) = \mathcal{L}[f(t)] = \int_0^{\infty} f(t)\epsilon^{-st} dt \quad (\text{A5-1})$$

where  $\mathcal{L}$  indicates the Laplace transform. Note that the variable time has been integrated out of the equation and that the Laplace transform is a function of the complex variable  $s$ . The inverse Laplace transform is given by

$$f(t) = \mathcal{L}^{-1}[F(s)] = \frac{1}{2\pi} \int_{\sigma-j\infty}^{\sigma+j\infty} F(s)\epsilon^{st} ds \quad (\text{A5-2})$$

where  $\mathcal{L}^{-1}$  indicates the inverse transform and  $j = \sqrt{-1}$

Equations (A5-1) and (A5-2) form the Laplace transform pair. Given a function  $f(t)$ , we integrate (A5-1) to find its Laplace transform  $F(s)$ . Then if this function  $F(s)$  is used to evaluate (A5-2), the result will be the original value of  $f(t)$ . The value of  $\sigma$  in (A5-2) is determined by the regions of convergence of (A5-1). We seldom use (A5-2) to evaluate an inverse Laplace transform; instead we use (A5-1) to construct a table of transforms for useful time functions. As an example, we will find the Laplace transform of the exponential function  $\epsilon^{-at}$ . From (A5-1),

$$\begin{aligned}
 F(s) &= \int_0^{\infty} \epsilon^{-at}\epsilon^{-st} dt = \int_0^{\infty} \epsilon^{-(s+a)t} dt = \left. \frac{-\epsilon^{-(s+a)t}}{s+a} \right|_0^{\infty} \\
 &= \frac{1}{s+a}, \quad \text{Re}(s+a) > 0
 \end{aligned} \quad (\text{A5-3})$$

where  $\text{Re}(\cdot)$  indicates the real part of the expression. Appendix VI contains a rather extensive table of Laplace transforms and  $z$ -transforms, with the first two columns of this table giving Laplace transforms. The remaining column in this table is useful when we consider digital control systems.

From the definition of the Laplace transform, (A5-1),

$$\mathcal{L}[kf(t)] = k\mathcal{L}[f(t)] = kF(s) \quad (\text{A5-4})$$

for  $k$  constant, and

$$\mathcal{L}[f_1(t) + f_2(t)] = \mathcal{L}[f_1(t)] + \mathcal{L}[f_2(t)] = F_1(s) + F_2(s) \quad (\text{A5-5})$$

The use of these two relationships greatly extends the application of the Laplace-transform table of Appendix VI.

We now present some examples of the Laplace transform and of the inverse Laplace transform. First, however, we need to note that using the complex inversion integral (A5-2) to evaluate the inverse Laplace transform results in  $f(t) = 0$  for  $t < 0$  [1]. Hence, to be consistent, we will always assign a value of zero to  $f(t)$  for all negative time. Also, to simplify notation, we define the unit step function  $u(x)$  to be

$$u(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (\text{A5-6})$$

In Equation (A5-3), the Laplace transform of  $\varepsilon^{-at}$  was derived. Note that the Laplace transform of  $e^{-at}u(t)$  is the same function. Thus for any function  $f(t)$ ,

$$\mathcal{L}[f(t)] = \mathcal{L}[f(t)u(t)] = F(s) \quad (\text{A5-7})$$

### EXAMPLE A5.1

The Laplace transform of the time function

$$f(t) = 5u(t) + 3\varepsilon^{-2t}$$

will now be found. From Appendix VI and (A5-4),

$$\mathcal{L}[5u(t)] = 5\mathcal{L}[u(t)] = \frac{5}{s}$$

$$\mathcal{L}[3\varepsilon^{-2t}] = 3\mathcal{L}[e^{-2t}] = \frac{3}{s + 2}$$

Then, from (A5-5),

$$F(s) = \mathcal{L}[5u(t) + 3\varepsilon^{-2t}] = \frac{5}{s} + \frac{3}{s + 2}$$

This Laplace transform can also be expressed as

$$F(s) = \frac{5}{s} + \frac{3}{s + 2} = \frac{8s + 10}{s(s + 2)}$$

The transforms are usually easier to manipulate in the combined form than in the sum-of-terms form.

This example illustrates an important point. As stated, we usually work with the Laplace transform expressed as a ratio of polynomials in the variable  $s$  (we call this ratio of polynomials a *rational function*). However, the tables used to find inverse transforms contain only low-order functions. Hence a method is required for converting from a general rational function to the forms that appear in the tables. This method is called the *partial-fraction expansion* method. A simple example is illustrated in the relationship

$$\frac{c}{(s+a)(s+b)} = \frac{k_1}{s+a} + \frac{k_2}{s+b}$$

Given the constants  $a$ ,  $b$ , and  $c$ , the problem is to find the coefficients of the partial-fraction expansion  $k_1$  and  $k_2$ . We now derive the general relationships required.

Consider the general rational function

$$F(s) = \frac{b_m s^m + \cdots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} = \frac{N(s)}{D(s)}, \quad m < n \quad (\text{A5-8})$$

where  $N(s)$  is the numerator polynomial and  $D(s)$  is the denominator polynomial. To perform a partial-fraction expansion, first the roots of the denominator must be found. Then  $F(s)$  can be expressed as

$$F(s) = \frac{N(s)}{D(s)} = \frac{N(s)}{\prod_{i=1}^n (s - p_i)} = \frac{k_1}{s - p_1} + \frac{k_2}{s - p_2} + \cdots + \frac{k_n}{s - p_n} \quad (\text{A5-9})$$

where  $\prod$  indicates the product of terms. Suppose that we wish to calculate the coefficient  $k_j$ . We first multiply (A5-9) by the term  $(s - p_j)$ .

$$(s - p_j)F(s) = \frac{k_1(s - p_j)}{s - p_1} + \cdots + k_j + \cdots + \frac{k_n(s - p_j)}{s - p_n} \quad (\text{A5-10})$$

If this equation is evaluated for  $s = p_j$ , we see then that all terms on the right side are zero except the  $j$ th term, and thus

$$k_j = (s - p_j)F(s)|_{s=p_j}, \quad j = 1, 2, \dots, n \quad (\text{A5-11})$$

In mathematics,  $k_j$  is called the *residue* of  $F(s)$  for the pole at  $s = p_j$ .

If the denominator polynomial of  $F(s)$  has repeated roots,  $F(s)$  can be expanded as in the example

$$\begin{aligned}
 F(s) &= \frac{N(s)}{(s - p_1)(s - p_2)^r} \\
 &= \frac{k_1}{s - p_1} + \frac{k_{21}}{s - p_2} + \frac{k_{22}}{(s - p_2)^2} + \cdots + \frac{k_{2r}}{(s - p_2)^r}
 \end{aligned} \quad (\text{A5-12})$$

where it is seen that a denominator root of multiplicity  $r$  yields  $r$  terms in the partial-fraction expansion. The coefficients of the repeated-root terms are calculated from the equation

$$k_{2j} = \frac{1}{(r - j)!} \frac{d^{r-j}}{ds^{r-j}} [(s - p_2)^r F(s)]|_{s=p_2} \quad (\text{A5-13})$$

For the proof of the derivation of this equation please see [4].

The preceding development applies to complex poles as well as real poles. Consider the case that  $F(s)$  has a pair of complex poles. If we let  $p_1 = a - jb$  and  $p_2 = a + jb$ , (A5-9) can be written as

$$F(s) = \frac{k_1}{s - a + jb} + \frac{k_2}{s - a - jb} + \frac{k_3}{s - p_3} + \cdots + \frac{k_n}{s - p_n} \quad (\text{A5-14})$$

The coefficients  $k_1$  and  $k_2$  can be evaluated using (A5-11) as before. It will be found, however, that these coefficients are complex valued, and that  $k_2$  is the conjugate of  $k_1$ . In order to achieve a convenient form for the inverse transform, we will use the following approach. From (A5-11),

$$\begin{aligned}
 k_1 &= (s - a + jb)F(s) \Big|_{s=a-jb} = R\epsilon^{j\theta} \\
 k_2 &= (s - a - jb)F(s) \Big|_{s=a+jb} = R\epsilon^{-j\theta} = k_1^*
 \end{aligned} \quad (\text{A5-15})$$

where the asterisk indicates the conjugate of the complex number. Define  $f_1(t)$  as the inverse transform of the first two terms of (A5-14). Hence

$$\begin{aligned}
 f_1(t) &= R\epsilon^{j\theta}\epsilon^{(a-jb)t} + R\epsilon^{-j\theta}\epsilon^{(a+jb)t} \\
 &= 2R\epsilon^{at} \left[ \frac{\epsilon^{j(bt-\theta)} + \epsilon^{-j(bt-\theta)}}{2} \right] \\
 &= 2R\epsilon^{at} \cos(bt - \theta)
 \end{aligned} \quad (\text{A5-16})$$

by Euler's identity [2]. This approach expresses the inverse transform in a convenient form and the calculations are relatively simple. The damped sinusoid has an amplitude of  $2R$  and a phase angle of  $\theta$ , where  $R$  and  $\theta$  are defined in (A5-15). Three examples of finding the inverse Laplace transform are given next.

### EXAMPLE A5.2

In this example the inverse Laplace transform of a rational function is found.

$$F(s) = \frac{5}{s^2 + 3s + 2} = \frac{5}{(s + 1)(s + 2)}$$

First the partial-fraction expansion is derived:

$$F(s) = \frac{5}{(s + 1)(s + 2)} = \frac{k_1}{s + 1} + \frac{k_2}{s + 2}$$

The coefficients in the partial-fraction expansion are calculated from (A5-11):

$$\begin{aligned}
 k_1 &= (s + 1)F(s) \Big|_{s=-1} = \frac{5}{s + 2} \Big|_{s=-1} = 5 \\
 k_2 &= (s + 2)F(s) \Big|_{s=-2} = \frac{5}{s + 1} \Big|_{s=-2} = -5
 \end{aligned}$$

Thus the partial-fraction expansion is

$$\frac{5}{(s+1)(s+2)} = \frac{5}{s+1} + \frac{-5}{s+2}$$

This expansion can be verified by recombining the terms on the right side to yield the left side of the equation. The inverse transform of  $F(s)$  is then

$$\mathcal{L}^{-1}[F(s)] = (5\epsilon^{-t} - 5\epsilon^{-2t})u(t)$$

The function  $u(t)$  is often omitted, but we must then understand that the inverse transform can be nonzero only for positive time and must be zero for negative time.

### EXAMPLE A5.3

As a second example of finding the inverse Laplace transform, consider the function

$$F(s) = \frac{2s+3}{s^3+2s^2+s} = \frac{2s+3}{s(s+1)^2} = \frac{k_1}{s} + \frac{k_{21}}{s+1} + \frac{k_{22}}{(s+1)^2}$$

The coefficients  $k_1$  and  $k_{22}$  can easily be evaluated:

$$k_1 = sF(s)|_{s=0} = \frac{2s+3}{(s+1)^2}|_{s=0} = 3$$

$$k_{22} = (s+1)^2 F(s)|_{s=-1} = \frac{2s+3}{s}|_{s=-1} = -1$$

We use (A5-13) to find  $k_{21}$ :

$$\begin{aligned} k_{21} &= \frac{1}{(2-1)!} \frac{d}{ds} [(s+1)^2 F(s)]|_{s=-1} = \frac{d}{ds} \left[ \frac{2s+3}{s} \right] \Big|_{s=-1} \\ &= \frac{s(2) - (2s+3)(1)}{s^2} \Big|_{s=-1} = \frac{-2-1}{1} = -3 \end{aligned}$$

Thus the partial-fraction expansion yields

$$F(s) = \frac{2s+3}{s(s+1)^2} = \frac{3}{s} + \frac{-3}{s+1} + \frac{-1}{(s+1)^2}$$

Then, from Appendix VI,  $f(t) = 3 - 3\epsilon^{-t} - t\epsilon^{-t}$ .

### EXAMPLE A5.4

To illustrate the inverse transform of a function having complex poles, consider

$$F(s) = \frac{10}{s^3+4s^2+9s+10} = \frac{10}{(s+2)(s^2+2s+5)} = \frac{10}{(s+2)[(s+1)^2+2^2]}$$



$$\begin{aligned}
 &= \frac{k_1}{s+2} + \frac{k_2}{s+1+j_2} + \frac{k_2^*}{s+1-j_2} \\
 &= \frac{k_1}{s+p_1} + \frac{k_2}{s+p_2} + \frac{k_2^*}{s+p_2^*}
 \end{aligned}$$

Evaluating the coefficient  $k_1$  as before,

$$k_1 = (s+2)F(s)|_{s=-2} = \frac{10}{(s+1)^2+4}|_{s=-2} = \frac{10}{5} = 2$$

Coefficient  $k_2$  is calculated from (A5-15).

$$\begin{aligned}
 k_2 &= (s+1+j_2)F(s) = \frac{10}{(s+2)(s+1-j_2)} \Big|_{s=-1-j_2} \\
 &= \frac{10}{(-1-j_2+2)(-1-j_2+1-j_2)} = \frac{10}{(1-j_2)(-j_4)} \\
 &= \frac{10}{(2.236\angle(-63.4^\circ))(4\angle(-90^\circ))} = 1.118\angle 153.4^\circ = R\angle\theta
 \end{aligned}$$

Therefore, using (A5-16),

$$f(t) = 2e^{-2t} + 2.236e^{-t} \cos(2t - 153.4^\circ)$$

## PROPERTIES OF THE LAPLACE TRANSFORM

The Laplace transform was defined in the last section. For the analysis and design of control systems, however, we require several properties of the Laplace transform. As an example, we derive the final-value property.

Suppose that we wish to calculate the final value of  $f(t)$ , that is,  $\lim_{t \rightarrow \infty} f(t)$ . However, we wish to calculate this final value directly from the Laplace transform  $F(s)$  without finding the inverse Laplace transform. The final-value property allows us to do this. To derive this property, it is first necessary to find the Laplace transform of the derivative of a general function  $f(t)$ .

$$\mathcal{L}\left[\frac{df}{dt}\right] = \int_0^\infty e^{-st} \frac{df}{dt} dt \tag{A5-17}$$

This expression can be integrated by parts, with

$$u = e^{-st}, \quad dv = \frac{df}{dt}$$

Thus

$$\begin{aligned}
 \mathcal{L}\left[\frac{df}{dt}\right] &= uv|_0^\infty - \int_0^\infty vdu = f(t)e^{-st}|_0^\infty - s \int_0^\infty e^{-st} f(t) dt \\
 &= 0 - f(0) + sF(s) = sF(s) - f(0)
 \end{aligned} \tag{A5-18}$$

To be mathematically correct, the initial-condition term should be  $f(0^+)$  [1], where

$$f(0^+) = \lim_{t \rightarrow 0} f(t), \quad t > 0 \tag{A5-19}$$

However, we will use the notation  $f(0)$ .

Now the final-value property can be derived. From (A5-17),

$$\begin{aligned} \lim_{s \rightarrow 0} \left[ \mathcal{L} \left( \frac{df}{dt} \right) \right] &= \lim_{s \rightarrow 0} \int_0^{\infty} \varepsilon^{-st} \frac{df}{dt} dt \\ &= \int_0^{\infty} \frac{df}{dt} dt = \lim_{t \rightarrow \infty} f(t) - f(0) \end{aligned} \tag{A5-20}$$

Then, from (A5-18) and (A5-20),

$$\lim_{t \rightarrow \infty} f(t) - f(0) = \lim_{s \rightarrow 0} [sF(s) - f(0)] \tag{A5-21}$$

or,

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s) \tag{A5-22}$$

provided that the limit on the left side of this relationship exists. The right-side limit may exist without the existence of the left-side limit.

Table A5.1 lists several useful properties of the Laplace transform. No further proofs of these properties are given here; interested readers should see [3, 4]. An example of the use of these properties is given next.

**TABLE A5-1 Laplace Transform Properties**

Name	Theorem
Derivative	$\mathcal{L} \left[ \frac{df}{dt} \right] = sF(s) - f(0^+)$
$n$ th-order derivative	$\mathcal{L} \left[ \frac{d^n f}{dt^n} \right] = s^n F(s) - s^{n-1} f(0^+) - \dots - f^{(n-1)}(0^+)$
Integral	$\mathcal{L} \left[ \int_0^t f(\tau) d\tau \right] = \frac{F(s)}{s}$
Shifting	$\mathcal{L} [f(t - t_0) u(t - t_0)] = \varepsilon^{-t_0 s} F(s)$
Initial value	$\lim_{t \rightarrow 0} f(t) = \lim_{s \rightarrow \infty} sF(s)$
Final value	$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s)$
Frequency shift	$\mathcal{L} [\varepsilon^{-at} f(t)] = F(s + a)$
Convolution integral	$\mathcal{L}^{-1} [F_1(s) F_2(s)] = \int_0^t f_1(t - \tau) f_2(\tau) d\tau = \int_0^t f_1(\tau) f_2(t - \tau) d\tau$

### EXAMPLE A5.5

As an example of applying the properties, consider the time function  $\cos(at)$ .

$$F(s) = \mathcal{L}[f(t)] = \mathcal{L}[\cos at] = \frac{s}{s^2 + a^2}$$

Then, from Table A5.1,

$$\mathcal{L}\left[\frac{df}{dt}\right] = \mathcal{L}[-a \sin at] = sF(s) - f(0) = \frac{s^2}{s^2 + a^2} - 1 = \frac{-a^2}{s^2 + a^2}$$

which agrees with the transform from Appendix VI. Also,

$$\mathcal{L}\left(\int_0^t f(\tau) d\tau\right) = \mathcal{L}\left(\frac{\sin at}{a}\right) = \frac{F(s)}{s} = \frac{1}{s^2 + a^2}$$

which also agrees with Appendix VI. The initial value of  $f(t)$  is

$$f(0) = \lim_{s \rightarrow \infty} sF(s) = \lim_{s \rightarrow \infty} \left[ \frac{s^2}{s^2 + a^2} \right] = 1$$

which, of course, is correct. If we carelessly apply the final-value property, we obtain

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sF(s) = \lim_{s \rightarrow 0} \left[ \frac{s^2}{s^2 + a^2} \right] = 0$$

which is incorrect, since  $\cos(at)$  does not have a final value; the function continues to vary between 1 and  $-1$  as time increases towards infinity. This exercise emphasizes the point that the final-value property does not apply to functions that have no final value.

### EXAMPLE A5.6

As a second example, we consider the time function  $f(t) = \varepsilon^{-0.5t}$ , which is then delayed by 4 s. Thus the function that we consider is

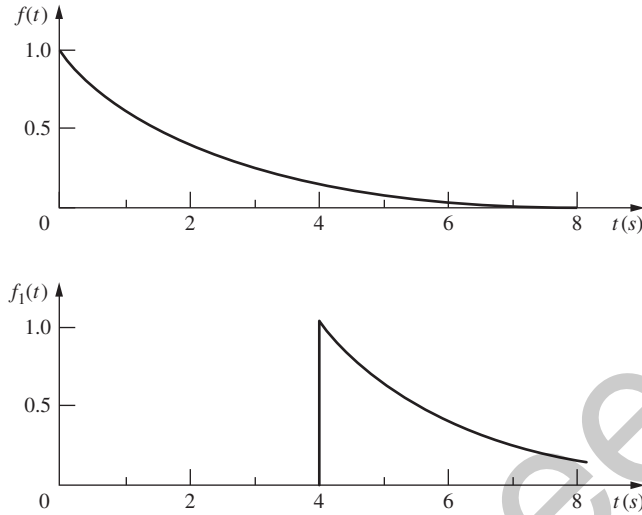
$$f_1(t) = f(t - 4)u(t - 4) = \varepsilon^{-0.5(t-4)}u(t - 4)$$

Both  $f(t)$  and  $f_1(t)$  are shown in Fig. A5.1. Note that  $f(t)$  is delayed by 4 s and that the value of the delayed function is zero for time less than 4 s (the amount of the delay). Both of these conditions are necessary in order to apply the shifting property of Table A5.1. From this property

$$\mathcal{L}[f(t - t_0)u(t - t_0)] = \varepsilon^{-t_0 s} F(s), \quad F(s) = \mathcal{L}[f(t)]$$

For this example, the unshifted function is  $\varepsilon^{-0.5t}$ , and thus  $F(s) = 1/(s + 0.5)$ . Hence

$$\mathcal{L}[\varepsilon^{-0.5(t-4)}u(t - 4)] = \frac{\varepsilon^{-4s}}{s + 0.5}$$



**FIGURE A5.1** Delayed time function.

Note that for the case that the time function is delayed, the Laplace transform is not a ratio of polynomials in  $s$  but contains the exponential function.

## DIFFERENTIAL EQUATIONS AND TRANSFER FUNCTIONS

In control system analysis and design, the Laplace transform is used to transform constant-coefficient linear differential equations into algebraic equations. Algebraic equations are much easier to manipulate and analyze than are differential equations. Whenever possible, we model analog physical systems with linear differential equations with constant coefficients. The Laplace transform, therefore, is a very helpful tool for simplifying the analysis and design of linear time-invariant systems.

An example of a linear differential equation modeling a physical phenomenon is Newton's law,

$$M \frac{d^2 x(t)}{dt^2} = f(t) \quad (\text{A5-23})$$

where  $f(t)$  is the force applied to a mass  $M$ , with the resulting displacement  $x(t)$ . It is assumed that the units in (A5-23) are consistent. Assume that we know the mass  $M$  and the applied force  $f(t)$ . The Laplace transform of (A5-23) is, from Table A5-1,

$$M[s^2 X(s) - sx(0) - \dot{x}(0)] = F(s) \quad (\text{A5-24})$$

where  $\dot{x}(t)$  denotes the derivative of  $x(t)$ . Thus to solve for the displacement of the mass, we must know the applied force, the initial displacement,  $x(0)$ , and the initial velocity,  $\dot{x}(0)$ . Then we can solve this equation for  $X(s)$  and take the inverse Laplace transform to find the displacement  $x(t)$ . We now solve for  $X(s)$ :

$$X(s) = \frac{F(s)}{Ms^2} + \frac{x(0)}{s} + \frac{\dot{x}(0)}{s^2} \quad (\text{A5-25})$$

For example, suppose that the applied force  $f(t)$  is zero. Then the inverse transform of (A5-25) is

$$x(t) = x(0) + \dot{x}(0)t, \quad t \geq 0 \quad (\text{A5-26})$$

If the initial velocity,  $\dot{x}(0)$ , is also zero, the mass will remain at its initial position  $x(0)$  for all time. If the initial velocity is not zero, the displacement of the mass will increase at a constant rate equal to that initial velocity.

Note that if the initial conditions are all zero, (A5-25) becomes

$$X(s) = \frac{1}{Ms^2}F(s) \quad (\text{A5-27})$$

Consider a physical phenomenon (system) that can be modeled by a linear differential equation with constant coefficients. The Laplace transform of the response (output) of this system can be expressed as the product of the Laplace transform of the forcing function (input) times a function of  $s$  (provided all initial conditions are zero), which we refer to as the *transfer function*. We usually denote the transfer function by  $G(s)$ . For the example given above we see from (A5-27) that the transfer function is

$$G(s) = \frac{1}{Ms^2} \quad (\text{A5-28})$$

Another example is shown next.

### EXAMPLE A5.7

Suppose that a system is modeled by the differential equation

$$\frac{d^2x(t)}{dt^2} + 3\frac{dx(t)}{dt} + 2x(t) = 2f(t)$$

In this equation,  $f(t)$  is the forcing function (the input) and  $x(t)$  is the response function (the output). If we take the Laplace transform of this equation, we have

$$s^2X(s) - sx(0) - \dot{x}(0) + 3[sX(s) - x(0)] + 2X(s) = 2F(s)$$

Solving this equation for the response  $X(s)$ ,

$$X(s) = \frac{2F(s) + (s + 3)x(0) + \dot{x}(0)}{s^2 + 3s + 2}$$

The transfer function is obtained by ignoring initial conditions.

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

Suppose that we wish to find the response with no initial conditions and with the system input equal to a unit step function. Then  $F(s) = 1/s$ , and

$$X(s) = G(s)F(s) = \left[ \frac{2}{s^2 + 3s + 2} \right] \left[ \frac{1}{s} \right]$$

or

$$X(s) = \frac{2}{s(s+1)(s+2)} = \frac{1}{s} + \frac{-2}{s+1} + \frac{1}{s+2}$$

by partial-fraction expansion. The inverse transform of this expression is then

$$x(t) = 1 - 2e^{-t} + e^{-2t}, \quad t \geq 0$$

Note that after a very long time,  $x(t)$  is approximately unity. The final-value property yields this same result:

$$\lim_{t \rightarrow \infty} x(t) = \lim_{s \rightarrow 0} sX(s) = \lim_{s \rightarrow 0} \frac{2}{s^2 + 3s + 2} = 1$$

In the last example, the response  $X(s)$  can be expressed as

$$X(s) = G(s)F(s) + \frac{(s+3)x(0) + \dot{x}(0)}{s^2 + 3s + 2} = X_f(s) + X_{ic}(s) \quad (\text{A5-29})$$

The term  $X_f(s)$  is the *forced* (also called the *zero-state*) *response*, and the term  $X_{ic}(s)$  is the *initial-condition* (*zero-input*) *response*. This result is general. We see then that the total response is the sum of two terms. The forcing-function term is independent of the initial conditions, and the initial-condition term is independent of the forcing function. This characteristic is a property of linear equations, and is referred to as the superposition property.

The concept of a transfer function is of fundamental importance to the study of linear feedback control systems. To generalize the results of the preceding paragraphs, let a system having an output  $c(t)$  and an input  $r(t)$  be described by the  $n$ th-order differential equation

$$\begin{aligned} \frac{d^n c}{dt^n} + a_{n-1} \frac{d^{n-1} c}{dt^{n-1}} + \cdots + a_1 \frac{dc}{dt} + a_0 c \\ = b_m \frac{d^m r}{dt^m} + b_{m-1} \frac{d^{m-1} r}{dt^{m-1}} + \cdots + b_1 \frac{dr}{dt} + b_0 r \end{aligned} \quad (\text{A5-30})$$

If we ignore all initial conditions, the Laplace transform of (A5-30) yields

$$\begin{aligned} (s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0)C(s) \\ = (b_ms^m + b_{m-1}s^{m-1} + \cdots + b_1s + b_0)R(s) \end{aligned} \quad (\text{A5-31})$$

Ignoring the initial conditions allows us to solve for  $C(s)/R(s)$  as a rational function of  $s$ , namely,

$$\frac{C(s)}{R(s)} = \frac{b_ms^m + b_{m-1}s^{m-1} + \cdots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0} \quad (\text{A5-32})$$

Note that the denominator polynomial of (A5-32) is the coefficient of  $C(s)$  in (A5-31). The reader will recall from studying classical methods for solving linear differential equations that

this same polynomial set equal to zero is the *characteristic equation* of the differential equation (A5-30).

Since most of the physical systems that we encountered were described by differential equations, we frequently referred to the *characteristic equation* of the *system*, or equivalently as the characteristic equation of the differential equation that described the system. The  $a_i$  coefficients in (A5-30) are parameters of the physical system described by the differential equation, such as mass, friction coefficient, spring constant, inductance, and resistance. It follows, therefore, that the characteristic equation does indeed *characterize* the system, since its roots are dependent only upon the system parameters; these roots determine that portion of the system's response (solution) whose form does not depend upon the form of input  $r(t)$ . This part of the solution is therefore the complementary solution of the differential equation.

## References

- [1] G. Doetsch, *Guide to the Applications of the Laplace and z-Transforms*. New York: Van Nostrand Reinhold, 1971.
- [2] J. D. Irwin, *Basic Engineering Circuit Analysis*, 3d ed. New York: Macmillan Publishing Company, 1990.
- [3] W. Kaplan, *Operational Methods for Linear Systems*. Reading, MA: Addison-Wesley Publishing Company, Inc., 1962.
- [4] R. V. Churchill, *Operational Mathematics*, 2d ed. New York: McGraw-Hill Book Company, 1972.

## Problems

- A5-1.** Using the defining integral for the Laplace transform, (A5-1), derive the Laplace transform of (a)  $f(t) = u(t - 2.5)$ ; (b)  $f(t) = e^{-4t}$ ; (c)  $f(t) = t$ .
- A5-2.** (a) Use the Laplace-transform tables to find the transform of each function given.  
 (b) Take the inverse transform of each  $F(s)$  in part (a) to verify the results.
- (i)  $f(t) = 3te^{-t}$                       (ii)  $f(t) = -5 \cos t$   
 (iii)  $f(t) = 2e^{-t} - e^{-2t}$               (iv)  $f(t) = 7e^{-0.5t} \cos 3t$   
 (v)  $f(t) = 5 \cos(4t + 30^\circ)$               (vi)  $f(t) = 6e^{-2t} \sin(t - 45^\circ)$
- A5-3.** (a) Find the inverse Laplace transform  $f(t)$  for each function given.  
 (b) Verify the results in part (a) by taking the Laplace transform of each  $f(t)$ , using the Laplace-transform tables.

(i)  $F(s) = \frac{1}{s(s+1)}$                       (ii)  $F(s) = \frac{3}{(s+1)(s+2)}$

(iii)  $F(s) = \frac{2s+1}{s^2+s-2}$                       (iv)  $F(s) = \frac{10s}{s^2+5s+4}$

- A5-4.** (a) Find the inverse Laplace transform  $f(t)$  for each function given.  
 (b) Verify the results in part (a) by taking the Laplace transform of each  $f(t)$ , using the Laplace-transform tables.

(i)  $F(s) = \frac{5}{s(s+1)(s+2)}$                       (ii)  $F(s) = \frac{1}{s^2(s+1)}$

(iii)  $F(s) = \frac{2s+1}{s^2+2s+5}$                       (iv)  $F(s) = \frac{s-30}{s(s^2+4s+29)}$

**A5-5.** Given the Laplace transform

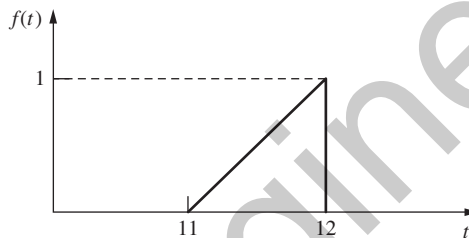
$$F(s) = \frac{s + 5}{s^2 + 4s + 13}$$

- Express the inverse transform as a sum of two complex exponential functions.
- Using Euler's relation, manipulate the result in part (a) into the form  $f(t) = Be^{-at} \sin(bt + \phi)$ .
- Express the inverse transform as  $f(t) = Ae^{-at} \cos(bt + \theta)$ .
- Take the Laplace transform of  $f(t)$  in part (c) to verify your result.

**A5-6.** (a) Plot  $f(t)$  if its Laplace transform is given by

$$F(s) = \frac{\epsilon^{-11s} - \epsilon^{-12s}}{s}$$

- The time function in part (a) is a rectangular pulse. Find the Laplace transform of the triangular pulse shown in Fig. PA5-6.



**FIGURE PA5-6** Triangular pulse.

**A5-7.** Given that  $f(t) = 4e^{-2(t-3)}$ .

- Find  $\mathcal{L}[df(t)/dt]$  by differentiating  $f(t)$  and then using the Laplace-transform tables.
- Find  $\mathcal{L}[df(t)/dt]$  by finding  $F(s)$  and using the differentiation property.
- Repeat parts (a) and (b) for  $f(t) = 4e^{-2(t-3)}u(t-3)$ .

**A5-8.** The Laplace transform of a function  $f(t)$  is given by

$$F(s) = \frac{3s + 4}{s^2 + 3s + 2}$$

- Without first solving for  $f(t)$ , find  $df(\tau)/d\tau$ .
- Without first solving for  $f(t)$ , find  $\int_0^t f(\tau)d\tau$ .
- Verify the results of parts (a) and (b) by first solving for  $f(t)$  and then performing the indicated operations.

**A5-9.** For the functions of Problem A5-4:

- Which of the inverse transforms do not have final values; that is, for which of the inverse transforms do the  $\lim_{t \rightarrow \infty} f(t)$  not exist?
- Find the final values for those functions that have final values.
- Find the inverse transform  $f(t)$  for each function in part (b) and verify your result.

**A5-10.** Given  $f_1(t) = u(t)$  and  $f_2(t) = \sin 10t$ .

- Find  $\mathcal{L}[f_1(t)]\mathcal{L}[f_2(t)]$ .
- Find  $\mathcal{L}[f_1(t)f_2(t)]$ .
- Is  $\mathcal{L}[f_1(t)]\mathcal{L}[f_2(t)]$  equal to  $\mathcal{L}[f_1(t)f_2(t)]$ ?
- Use the convolution integral of Table A5-1 to find the inverse transform of the results in part (a).
- Verify the results of part (d) by finding  $\mathcal{L}^{-1}[F_1(s)F_2(s)]$  directly.



**A5-11.** Given the differential equation

$$\frac{d^2x(t)}{dt^2} + 5\frac{dx(t)}{dt} + 4x(t) = 10u(t)$$

- Find  $x(t)$  for the case that the initial conditions are zero. Show that your solution yields the correct initial conditions; that is, solve for  $x(0)$  and  $\dot{x}(0)$  using your solution.
- Show that your solution in part (a) satisfies the differential equation by direct substitution.
- Find  $x(t)$  for the case that  $x(0) = 1$  and  $\dot{x}(0) = 1$ . Show that your solution yields the correct initial conditions; that is, solve for  $x(0)$  and  $\dot{x}(0)$  using your solution.
- Show that your solution in part (c) satisfies the differential equation by direct substitution.
- Verify all partial-fraction expansions by computer.

**A5-12.** Given the differential equation

$$\frac{d^2x(t)}{dt^2} + 2\frac{dx(t)}{dt} + x(t) = 5, \quad t \geq 0$$

- Find  $x(t)$  for the case that the initial conditions are zero. Show that your solution yields the correct initial conditions; that is, solve for  $x(0)$  and  $\dot{x}(0)$  using your solution.
- Show that your solution in part (a) satisfies the differential equation by direct substitution.
- Find  $x(t)$  for the case that  $x(0) = 1$  and  $\dot{x}(0) = 2$ . Show that your solution yields the correct initial conditions; that is, solve for  $x(0)$  and  $\dot{x}(0)$  using your solution.
- Show that your solution in part (c) satisfies the differential equation by direct substitution.
- Verify all partial-fraction expansions by computer.

**A5-13.** Find the transfer function  $C(s)/R(s)$  for each of the systems described by the given differential equation, where  $\ddot{c}(t)$  denotes the second derivative of  $c(t)$  with respect to  $t$ , and so on.

- $\dot{c}(t) + 2c(t) = r(t)$
- $\ddot{c}(t) + 2\dot{c}(t) = r(t - t_0)u(t - t_0) + 3\dot{r}(t)$
- $\ddot{c}(t) + 3\dot{c}(t) + 2c(t) + c(t) = \dot{r}(t) + 3\dot{r}(t)$

**A5-14.** For each of the systems, find the system differential equation if the transfer function  $G(s) = C(s)/R(s)$  is given by

- $G(s) = \frac{60}{s^2 + 10s + 60}$
- $G(s) = \frac{3s + 20}{s^3 + 4s^2 + 8s + 20}$
- $G(s) = \frac{s + 1}{s^2}$
- $G(s) = \frac{7e^{-0.2s}}{s^2 + 6s + 32}$

**A5-15.** (a) Write the characteristic equation for the system of Problem A5-11.

(b) Write the characteristic equation for the system of Problem A5-12.

(c) Write the characteristic equations for the systems of Problem A5-13.

(d) Write the characteristic equations for the systems of Problem A5-14.

**A5-16.** Equations (A5-4) and (A5-5) illustrate the linear properties of the Laplace transform. This problem illustrates that these linear properties do not carry over to nonlinear operations.

- Given  $f_1(t) = \varepsilon^{-t}$ , find  $F_1(s) = \mathcal{L}[f_1(t)]$  and  $\mathcal{L}[f_1^2(t)]$ .
- In part (a), is  $\mathcal{L}[f_1^2(t)] = F_1^2(s)$ ; that is, is the Laplace transform of a squared time function equal to the square of the Laplace transform of that function?
- Given  $f_1(t) = \varepsilon^{-t}$  and  $f_2(t) = \varepsilon^{-2t}$ , find  $F_1(s)$ ,  $F_2(s)$ , and  $\mathcal{L}[f_1(t)/f_2(t)]$ .
- In part (c), is  $\mathcal{L}[f_1(t)/f_2(t)] = F_1(s)/F_2(s)$ ; that is, is the Laplace transform of the quotient of two time functions equal to the quotient of the Laplace transforms of these functions?

**A5-17.** (a) Write the terms that appear in the natural response for a system described by the differential equation in Problem A5-11.

(b) Write the terms that appear in the natural response for a system described by the differential equation in Problem A5-12.

(c) Write the terms that appear in the natural response for a system described by the transfer function in Problem A5-14(d).

## z-Transform Tables

Controlengineers.ir

Laplace transform $E(s)$	Time function $e(t)$	$z$ -Transform $E(z)$	Modified $z$ -transform $E(z, m)$
$\frac{1}{s}$	$u(t)$	$\frac{z}{z-1}$	$\frac{1}{z-1}$
$\frac{1}{s^2}$	$t$	$\frac{Tz}{(z-1)^2}$	$\frac{mT}{z-1} + \frac{T}{(z-1)^2}$
$\frac{1}{s^3}$	$\frac{t^2}{2}$	$\frac{T^2z(z+1)}{2(z-1)^3}$	$\frac{T^2}{2} \left[ \frac{m^2}{z-1} + \frac{2m+1}{(z-1)^2} + \frac{2}{(z-1)^3} \right]$
$\frac{(k-1)!}{s^k}$	$t^{k-1}$	$\lim_{a \rightarrow 0} \frac{\partial^{k-1}}{\partial a^{k-1}} \left[ \frac{z}{z-\epsilon^{-aT}} \right]$	$\lim_{a \rightarrow 0} \frac{\partial^{k-1}}{\partial a^{k-1}} \left[ \frac{\epsilon^{-amT}}{z-\epsilon^{-aT}} \right]$
$\frac{1}{s+a}$	$\epsilon^{-at}$	$\frac{z}{z-\epsilon^{-aT}}$	$\frac{\epsilon^{-amT}}{z-\epsilon^{-aT}}$
$\frac{1}{(s+a)^2}$	$t\epsilon^{-at}$	$\frac{Tz\epsilon^{-aT}}{(z-\epsilon^{-aT})^2}$	$\frac{T\epsilon^{-amT}[\epsilon^{-aT} + m(z-\epsilon^{-aT})]}{(z-\epsilon^{-aT})^2}$
$\frac{(k-1)!}{(s+a)^k}$	$t^k\epsilon^{-at}$	$\lim_{a \rightarrow 0} \left[ \frac{\partial^k}{\partial a^k} \left[ \frac{z}{z-\epsilon^{-aT}} \right] \right]$	$\lim_{a \rightarrow 0} \left[ \frac{\partial^k}{\partial a^k} \left[ \frac{\epsilon^{-amT}}{z-\epsilon^{-aT}} \right] \right]$
$\frac{a}{s(s+a)}$	$1 - \epsilon^{-at}$	$\frac{(aT)\epsilon^{-aT}}{z(1-z)}$	$\frac{1}{z-1} - \frac{\epsilon^{-amT}}{z-\epsilon^{-aT}}$
$\frac{a}{s^2(s+a)}$	$t - \frac{1-\epsilon^{-at}}{a}$	$\frac{z[(aT+1+\epsilon^{-aT})z+(1-\epsilon^{-aT})] - aT\epsilon^{-aT}}{a(z-1)^2(z-\epsilon^{-aT})}$	$\frac{T}{(z-1)^2} + \frac{amT}{a(z-1)} + \frac{\epsilon^{-amT}}{a(z-\epsilon^{-aT})}$
$\frac{a^2}{s(s+a)^2}$	$1 - (1+at)\epsilon^{-at}$	$\frac{z}{z-\epsilon^{-aT}} - \frac{aT\epsilon^{-aT}z}{(z-\epsilon^{-aT})^2}$	$\frac{1}{z-1} - \frac{aT\epsilon^{-aT}}{a(z-\epsilon^{-aT})}$
$\frac{b-a}{(s+a)(s+b)}$	$\epsilon^{-at} - \epsilon^{-bt}$	$\frac{(\epsilon^{-aT}-\epsilon^{-bT})z}{(z-\epsilon^{-aT})(z-\epsilon^{-bT})}$	$\frac{\epsilon^{-bmT}}{z-\epsilon^{-bT}} - \frac{\epsilon^{-amT}}{z-\epsilon^{-aT}}$

(continued)

Laplace transform $E(s)$	Time function $e(t)$	$z$ -Transform $E(z)$	Modified $z$ -transform $E(z, m)$
$\frac{a}{s^2 + a^2}$	$\sin(at)$	$\frac{z \sin(aT)}{z^2 - 2z \cos(aT) + 1}$	$\frac{z \sin(amT) + \sin(1 - m)aT}{z^2 - 2z \cos(aT) + 1}$
$\frac{s}{s^2 + a^2}$	$\cos(at)$	$\frac{z(z - \cos(aT))}{z^2 - 2z \cos(aT) + 1}$	$\frac{z \cos(amT) - \cos(1 - m)aT}{z^2 - 2z \cos(aT) + 1}$
$\frac{1}{(s + a)^2 + b^2}$	$\frac{1}{b} \frac{\epsilon^{-at} \sin bt}{b}$	$\frac{1}{b} \left[ \frac{z \epsilon^{-aT} \sin bT}{z^2 - 2z \epsilon^{-aT} \cos bT + \epsilon^{-2aT}} \right]$	$\frac{1}{b} \left[ \frac{\epsilon^{-amT} [z \sin bmT + \epsilon^{-aT} \sin(1 - m)bT]}{z^2 - 2z \epsilon^{-aT} \cos bT + \epsilon^{-2aT}} \right]$
$\frac{s + a}{(s + a)^2 + b^2}$	$\epsilon^{-at} \cos bt$	$\frac{z^2 - z \epsilon^{-aT} \cos bT}{z^2 - 2z \epsilon^{-aT} \cos bT + \epsilon^{-2aT}}$	$\frac{\epsilon^{-amT} [z \cos bmT + \epsilon^{-aT} \sin(1 - m)bT]}{z^2 - 2z \epsilon^{-aT} \cos bT + \epsilon^{-2aT}}$
$\frac{a^2 + b^2}{s[(s + a)^2 + b^2]}$	$1 - \epsilon^{-at} \left( \cos bt + \frac{a}{b} \sin bt \right)$	$\frac{z(Az + B)}{(z - 1)(z^2 - 2z \epsilon^{-aT} \cos bT + \epsilon^{-2aT})}$	$\frac{1}{z - 1}$
$\frac{1}{s(s + a)(s + b)}$	$\frac{1}{ab} + \frac{\epsilon^{-at}}{a(a - b)} + \frac{\epsilon^{-bt}}{b(b - a)}$	$A = 1 - \epsilon^{-aT} \left( \cos bT + \frac{a}{b} \sin bT \right)$ $B = \epsilon^{-2aT} + \epsilon^{-aT} \left( \frac{a}{b} \sin bT - \cos bT \right)$	$-\frac{\epsilon^{-amT} [z \cos bmT + \epsilon^{-aT} \sin(1 - m)bT]}{z^2 - 2z \epsilon^{-aT} \cos bT + \epsilon^{-2aT}}$ $+\frac{a}{b} \frac{\{\epsilon^{-amT} [z \sin bmT + \epsilon^{-aT} \sin(1 - m)bT]\}}{z^2 - 2z \epsilon^{-aT} \cos bT + \epsilon^{-2aT}}$

$$A = \frac{b(1 - \epsilon^{-aT}) - a(1 - \epsilon^{-bT})}{ab(b - a)}$$

$$B = \frac{a \epsilon^{-aT}(1 - \epsilon^{-bT}) - b \epsilon^{-bT}(1 - \epsilon^{-aT})}{ab(b - a)}$$

# INDEX

## A

Ackermann's equation, 355  
 Ackermann's formula, 374  
     for current observer, 370  
     for gain matrix, 350  
 adjoint of matrix, 503  
 admissible control, 419  
 admittance, 485  
 aircraft lateral control system, 14  
 aircraft lateral position, 13  
 algebraic loop, 150  
 algebraic Riccati equation, 437–438  
 analog simulation of continuous systems, 60  
 analog-to-digital (A/D) converter, 35, 100, 134, 170  
 antenna pointing system, 20–21  
 antialiasing filter, 116  
*a priori*, 394, 407  
 armature inductance, 19  
 asymptotes, 245  
 automatic aircraft landing system, 13, 466–474  
     chamber behavior for normal operating conditions, 467  
     compensated-system Nyquist diagram, 471  
     design, 468–471  
     disturbances to be modeled, 470–471  
      $\alpha$ -filter frequency response, 473–474  
     F4J lateral frequency response, 469  
     lateral control loop, 469  
     lateral control system, 470–471  
     Nyquist diagram, 468, 470  
     plant model, 468  
      $\alpha$ - $\beta$  tracking filter, 471–472  
     typical frequency response, 469  
 autoregressive moving-average (ARMA) model, 402  
 azimuth angle of antenna, 290

## B

bank command, 13  
     mathematical relationships between wind input, 14  
 basis functions, 391  
 batch least squares, 409  
 Bellman's principle, 421, 433  
 bilinear form, 507  
 black-box identification, 394–401  
 Bode diagrams, 258–259, 287  
     summary of terms employed, 257  
     breakaway points, 245–248

## C

cascade compensation, 285  
 Cauchy's principle of argument, 249

causal system, 109  
 chamber temperature control hardware diagram, 462  
 characteristic equation  
     characteristic values of matrix, 74  
     of a matrix, 74  
     of the system, 519  
 characteristic vector, 77, 79  
 closed-loop digital control system, 170, 173  
 closed-loop discrete-time systems, 167  
 closed-loop frequency response, 261–270  
     resonance in, 268  
 closed-loop physical systems, 11  
     mathematical solutions for, 16  
     pilot's concept of landing an aircraft, 11  
 closed-loop system, 100  
     characteristic polynomial, 346  
     matrix, 350  
     sampled-data, 169  
 closed-loop transfer function  $T(z)$ , 199, 205  
 CO<sub>2</sub> control system, 462–463  
 cofactor of matrix, 503  
 compensation on system, 285–287  
     on digital control systems, 286  
     integration and differentiation filters, 307–309  
     lag-lead, 303–307  
     phase-lag, 287–294  
     phase-lead, 294–295  
 compensator, 11  
     transfer function, 286–287  
 complementary strip, 110–111  
 complex power, 25  
 complex variable theory, 56  
 conditionally stable system, 289  
 conservation of energy, 23  
 constant damping loci, 209  
 constant frequency loci, 209  
 constant magnitude locus (constant M circle), 262–263  
 constant N (phase) circles, 263–264  
 constraint, 422  
 continuous-time (analog) system, 219  
 continuous-time signal, 133  
 continuous-time state variables, 146–147  
     model, 153  
     for SMIB system, 26  
 control actuator, 11  
 control canonical form, 68  
     state matrices for, 76  
 control energy, 419  
 controllability, concepts of, 374–378  
 controller, 11  
 controlling unit, 13

control problem, 15–16  
 control software packages, 223  
 control system designer, task of, 15  
 control system specifications  
     constraints on control effort, 285  
     disturbance rejection, 284–285  
     relative stability measurements, 282–283  
     sensitivity of system characteristics, 283–284  
     steady-state accuracy, 280  
     transient response, 280–282  
 convex function, 486  
 convolution. *See* discrete convolution  
     costate, 433–434  
 cost function, 418, 429  
     terms in, 422  
 Cramer's rule, 172, 176  
 critically damped, 324, 328–332  
 crossover point, 247, 268  
 current estimator, 442  
 current observers, 369–374  
 current phasor, expression for, 25

## D

damping factor for linear friction, 147  
 data hold, 102  
 data reconstruction, 113–121  
     first-order hold, 118–119  
     fractional-order hold, 119–121  
     reconstructed version of  $e(t)$ , 113  
     using polynomial extrapolation, 113  
     zero-order hold, 114–118  
 data-reconstruction device, 101  
 DC gain, 131, 203  
 dc motor system, 18  
 decimal-to-binary conversion algorithms, 293  
 delayed  $z$ -transform, 136–137  
 derivation procedure, 171–172  
 derivative of a matrix, 505–506  
 diagonal matrix, 502  
 differentiator transfer function, 309  
 digital computer, 35–36  
 digital controllers, 12  
     with nonzero computation time, 141  
      $n$ th-order linear, 140  
 digital control system, 12–15  
 digital filter, 37  
     differentiation of a function, 308  
     in U.S. Navy aircraft carriers, 37  
 digital-to-analog (D/A) converter, 35, 100, 134  
 discrete convolution technique, 57–58  
 discrete Riccati equation, 434, 439–440  
 discrete state equations of a sampled-data system, 150–154

discrete state matrices, 180–181  
 discrete state models for digital control systems, 183–188  
 discrete state-space model for the closed loop, 187  
 discrete-time systems, 12, 35–37  
     with time delays, 139–142  
 discrete unit impulse function, 43, 256  
 discrete unit step function, 41  
 disturbances, 13  
 double-sided  $z$ -transform, 38  
 dynamic systems, identifying, 394  
     batch least squares, 409  
     black-box identification, 394–401  
     choice of input, 412–413  
     least-squares system identification, 401–407  
     practical factors for identification, 412–414  
     recursive least-squares system identification, 409–412  
     sampling frequency, 413  
     signal scaling, 413–414

## E

eigenvalues, 74, 435–438, 502  
 eigenvectors, 435–438, 503  
 electric circuit law, 25  
 electric power, 26  
 electric power system, 484  
 electric power system models, topology identification in, 484–488  
 environmental chamber control system, 461–466  
 error-control condition, 382  
 error signal, 15, 18, 21  
 $E^*(s)$ , 169, 208  
     amplitude of the discontinuity of  $e(t)$ , 500  
     for  $e(t) = \epsilon^{-t}$ , 105  
     for  $e(t) = u(t)$ , 104–105  
     evaluation of, 105–108, 496–500  
     Laplace transform of, 105  
     properties of, 110–113  
     relationship between  $E(z)$  and, 126–127  
     theorem of residues, 498–499  
     zeros of, 110  
 Euler method, 219  
 Euler's identity, 511  
 Euler's relation, 54, 107, 110

## F

feedback, parallel, or minor-loop compensation, 286  
 feedback path, 103  
 filter transfer function, 298, 302, 310  
 final-value theorem, 204, 470, 515  
 first-order hold, 118–119  
     frequency response of, 119–120  
 first-order linear differential equation, 23, 36  
 flow graphs, 59–62

forward path, 492  
 Fourier transform  
     of  $e(t)$ , 111, 115  
     results from, 108–110  
 fractional-order hold, 119–121  
     frequency response magnitudes for, 121  
     impulse response of, 120  
     transfer function of, 119  
 frequency aliasing, 116  
 frequency foldover, 116  
 frequency response, interpretation of, 110, 259–261  
 frequency spectrum of  $e(t)$ , 109  
 full-order current observer, 369–370  
 fundamental matrix, 85

## G

gain margin, 255  
 general rational function, 510  
 generating function, defined, 38  
 grey-box identification, 391  
 $G(z)$ , 198

## H

Hankel matrix, 396, 399–400  
 hardware configuration of system, 349  
 high frequency gain, 287  
 high-order systems, computations for, 154–155

## I

ideal filter, 112  
 ideal sampler, 102–104  
     defined, 104  
 ideal time delays, systems with, 139–142  
 identity matrix, 501  
 IEEE 39-bus power system, 486, 487  
 impulse functions, 134  
 impulse modulator, 103  
 inertia, 485  
 infinite bus, 24  
 infinite-horizon linear-quadratic-Gaussian (IH-LQG) design, 442, 444–446  
 initial-condition (zero-input) response, 518  
 input space of system, 63  
 integral of a matrix, 506  
 integrator transfer function, 308  
 inverse Laplace transform, 103, 149  
 inverse  $z$ -transform, 200, 206–207  
     discrete convolution technique, 57–58  
     inversion-formula method, 56  
     partial-fraction expansion method, 52  
     power series method, 51  
 inversion-formula method, 56

## K

Kalman filters, 374, 440–444  
 Kirchoff's law, 25  
 Kronecker delta function, 440

## L

Laplace transform, 17, 24, 37, 52, 57, 102–103, 508–519  
     of constant-coefficient linear differential equations into algebraic equations, 516–519  
     convolution property of, 133  
     definition of, 508  
     of exponential function, 508  
     inverse, 508, 511–513  
     of linear time-invariant continuous-time systems, 37–38  
     properties, 513–514  
     for system response, 218  
     transfer function, 169  
 lateral control system, 13  
 least squares estimation, 392, 401, 486  
 least-squares minimization, 446  
 least-squares system identification, 401–407  
 linear quadratic (LQ) optimal control, 424–428  
 linear time-invariant difference equations, solving, 48–51, 59  
 linear time-invariant (LTI) discrete-time systems, 12  
     bilinear transformation, 234–238  
     characteristic equation of, 234  
     Jury stability test, 239–243, 245–246  
     Nyquist criterion for, 248–256  
     root locus for, 244–247  
     Routh–Hurwitz criterion, 236–241, 246  
     stability, 230–233  
 linear time-invariant (LTI) systems, 167, 391  
 linear time-varying discrete system, 89–90  
 loop, 492  
 loop gain, 492  
 low-order single-input single-output systems, 378–380

## M

marginally stable, 231  
 Marine Air Traffic Control and Landing System (MATCALs), 466, 468  
 Mason's gain formula, 61–62, 68, 80, 150, 172, 174, 176–177, 491–493  
 MATLAB *pidtool*, 319–321, 477–484  
 MATLAB *sisotool*, 332–333  
 matrix, 18, 501–507  
     adjoint of, 503  
     algebra of, 505–507  
     cofactor of, 503  
     derivative of a, 505–506  
     determinant of, 504  
     diagonal, 502  
     identity, 501  
     inverse of, 504  
     inversion lemma, 504

minor of, 503  
 multiplication of  $a$ , 502, 505  
 partitioned, 502  
 symmetric, 502  
 trace of  $a$ , 502  
 transpose of, 502  
 McDonnell-Douglas Corporation F4 aircraft, 14  
 mechanical power, 26  
 memory locations (shift registers), 60  
 minimum-cost function, 424  
 minimum principle, 433–434  
 modal matrix, 77–78  
 modified  $z$ -transform, 136–139, 499  
 properties of, 137  
 Moore-Penrose pseudo-inverse of  $\Phi$ , 392  
 motor back emf, 18  
 multiplication of matrices, 505  
 multiplication of scalars, 505  
 multiplication of vectors, 505

## P

neonatal fractional inspired oxygen, PID feedback controllers for  
   MATLAB *pidtool* PIDF controllers, 477–484  
   plant transfer function, 474–476  
   Taube's PID controller, 476–477  
 Newton's laws, 390, 516  
   second law of motion, 25  
 Nichols chart, 264–266  
 ninth-order ordinary nonlinear differential equation, 14–15  
 nonsynchronous sampling, 142–145  
 nontouching loops, 492  
 $n$ th-order continuous-time system, 37  
 $n$ th-order differential equation, 518  
 $n$ th-order linear difference equation, 37  
 $n$ th-order linear digital controller, 140  
 numerical integration algorithm, 219–222  
 Nyquist criterion for discrete-time systems, 248–256, 311  
   characteristic equation, 249  
   frequency response for  $G(z)$ , 253  
   gain and phase margins, 255  
   MATLAB program to plot Nyquist diagram, 254–255  
   Nyquist diagram, 250–251  
   Nyquist path, 249–250  
   pulse transfer functions, 255–256  
    $s$ -plane Nyquist diagram, 249–250  
   theorem, 249  
   transfer function, 248  
    $z$ -plane Nyquist diagram, 251–252

## O

observability, concepts of, 374–378  
 observer-based control systems, 369–374  
 observer canonical form, 68  
 open-loop dc gain, 203

open-loop sampled-data systems, 168  
 open-loop systems containing digital filters, 133–134  
   model, 134  
 open-loop transfer function, 234  
 optimal control law for system, 428–429  
 optimality, principle of, 421–424  
 original signal flow graph, 171, 173–174, 176–177  
 output-feedback controller, 26  
 overshoot, 206

## P

parameter Estimation, 391  
 partial-fraction expansion method, 52, 510, 512  
 path, 492  
 path gain, 492  
 peak overshoot, 280–282  
 percent peak overshoot, 26  
 performance index, 418  
 persistency of excitation, 412  
 phase-lag compensator, 287–294  
   advantages, 303  
 phase-lead compensation, 294–295  
   advantages of, 303  
   closed-loop frequency responses, 299, 301  
   design procedure, 295–298  
   disadvantages of, 303  
   MATLAB program, 291–292, 299  
   open-loop frequency responses, 299–300, 302–303  
   step responses, 300  
 phase-lead filter, 299  
 phase margin, 255  
 phase margin of the compensated system, 289  
 phase variable canonical form, 68  
 physical sampler, 103  
 pitch angle, 21  
 plant  
   defined, 11  
   dynamics of, 12  
 pole assignment/pole placement, 343–346  
 pole-zero cancellations, 391  
 pole-zero locations, 110  
 positive definite quadratic form, 507  
 positive semidefinite quadratic form, 507  
 power amplifier, 101  
 power series method, 51  
 prediction errors, covariance of, 442  
 prediction observer, 353  
 predictor-corrector algorithm, 221–222  
 primary strip, 110–111  
 proportional-integral (PI) compensator, 35  
 proportional-plus-derivative (PD) controller, 247  
 proportional-plus-integral (PI) compensator, 243

proportional-plus-integral-plus-derivative (PID) controller, 37, 279, 309–313, 463  
   analog version of, 463  
   block diagram, 464  
   design process, 313–315  
   frequency response for, 310–311  
   MATLAB program, 316–318  
   step response behavior, 464–465  
   transfer function, 309–310, 312  
 pseudo inverse, 392. *See also*, Moore Penrose pseudo inverse  
 pulse transfer function, 127–133

## Q

quadratic cost function, 419–421  
 quadratic form, 419, 506

## R

radar-noise disturbances, 14  
 radar unit, 13  
 rectangular rule for numerical integration, 36, 219  
 recursive least-squares system identification, 409–412  
 reduced-order observer, 364–369  
 regulator control system, 378  
 repeated-root terms, 510  
 residue of a function, 106  
 residue of  $F(s)$ , 510  
 resonance, 267–268  
 response, defined, 15  
 resultant system stability margins, 293  
 Riccati equation, 434, 437  
 rise time, 280–282  
 robotic control system, 21–22  
 root locus for a system, 321–334  
   characteristic equation, 321–322  
   filter dc gain equal unity, 322  
   MATLAB program for, 324–326, 330–333  
   phase-lag design, 322–324  
   phase-lead design, 326–328  
 round-off errors in computer, 220  
 Runge-Kutta rule, 223

## S

sampled-data control systems, 100–103, 113  
   sampled signal flow graph, 172  
   sample period, 13  
   sampler-data hold device, 101–103  
   satellite model, 16–18  
   second-order differential equation, 17  
   second-order system, 441  
   second-order transfer function, 17  
   series compensation, 285  
   sensitivity, 283  
   servomechanism, 20–21  
   servomotor system, 18–22, 429, 454–461  
   computer data format, 456  
   phase-lag filter, 460–461

servomotor system (*continued*)
 

- phase lead design, 459–461
- system frequency response, 458–460
- system hardware, 455
- system model, 456–459

 settling time, 280–282
   
 Shannon's sampling theorem, 112–113
   
 shifting theorem, 138
   
 signal flow graph, 60
 

- corresponding state equations of, 68–69

 similarity transformations, 73
 

- properties, 74

 simulation diagrams
 

- of analog plant, 154
- for discrete-time systems, 59–62

 single-machine infinite bus (SMIB)
 

- power system, 24–26
- continuous-time state-variable model for, 26
- current phasor, expression for, 25
- set of symbols, 25

 single-sided  $z$ -transform, 38
   
 single-valued function relationship, 64
   
 singular value decomposition (SVD), 396
   
 sink node, 492
   
 SNR (signal to noise ratio), 413
   
 source node, 492
   
 specific heat of liquid, 23
   
 stabilizable, 439
   
 starred transform, 102, 111
   
 state equations
 

- numerical method via digital computer, 87
- recursive solution, 84–86
- $z$ -transform of, 81–84, 86–87

 state estimator
 

- closed-loop state equations, 363–364
- closed-loop system characteristic equation, 362–363
- controller transfer function, 359–362
- error dynamics, 354–356
- errors in, 354
- example, 356–359
- observer model, 352–353
- plant–observer system, 355

 state transition matrix, 85, 149
 

- computer method for finding, 87
- properties of, 88

 state-variable formulations, 71
 

- converting continuous state equations, 178–183
- for diagonal elements, 76–78
- for digital controller, 183–188
- examples, 71–80
- for mechanical system, 147–150
- of open-loop sampled-data systems, 145–146
- using linear-transformation matrix, 75–76
- using partial-fraction expansion, 72–73
- using similarity transformations, 73–74

state-variable model of a system, 63–64
 

- example, 64–68
- multivariable discrete system, state equations for, 70–71
- transfer function, state equations for, 69–70

 static systems, 391–393
   
 steady-state accuracy, 215–218
   
 steady-state optimal control, 434–438
   
 stiffness factor, 147
   
 summing junction, 491
   
 symmetric matrix, 502
   
 synchronous generator, 484
   
 system characteristic equation, 207–208
   
 system identification, 390
   
 system time response, 198–200
 

- for all instants of time of sampled-data system, 201
- analog system unit-step response, 200
- effects of sampling, 200–202
- mapping  $s$ -plane into  $z$ -plane, 208–215
- simulation of, 218–222

**T**

Taube's PID controller, 476–477
   
 temperature control system, 22–24, 463–466
   
 Texas Instruments TI9900
 

- microprocessor system, 455

 thermal capacity of liquid, 23
   
 thermal system. *see* temperature control system
   
 three-axis control of satellite, 16–18
   
 thrusters, 16
   
 time-delayed function, 41
   
 time-invariant analog filter, 37
   
 time-invariant system, 12
   
 time to peak overshoot, 280
   
 TMS 9900 microcomputer system, 462
   
 topology, 484, 485
   
 trace of a matrix, 502
   
 tracking error  $e(t)$ , 100
   
 transfer function, 59–61, 80, 168, 170, 293, 407, 517
 

- closed-loop, 205
- of continuous-time system, 204
- controller, 359–362
- controller-estimator, 371
- of controller estimator, 373–374
- differentiator, 309
- of a digital PID controller, 309–310, 312
- discrete-time, 400
- filter, 302, 310
- of a first-order hold, 118
- of fractional-order hold, 119
- integrator, 308
- partial-fraction expansion of, 72–73
- poles or zeros of, 407
- for a pure delay of  $T$  seconds, 68
- for sampled-data systems, 171
- set of discrete state-variable equations from, 65–66

state equations, 69–70
 

- of zero-order hold, 115
- zeros of system, 379–380

 transform methods, 37–38
   
 transient response, 207
   
 transmission line, 24–25
   
 transpose of a matrix, 502
   
 trapezoidal rule for numerical integration, 221–223, 307

**U**

uncontrollable, 375
   
 unilateral Laplace transform, 109
   
 unit circle, 231
   
 unit sample function, 43
   
 unit-step function, 102, 509
   
 unit-step response of the sampled-data system, 202–203
   
 unity-dc-gain phase-lead compensator, 298
   
 U.S. Navy aircraft carriers, 12

**V**

vector-matrix form, 18, 501–507
   
 voltage equation for armature circuit, 19

**W**

weighting matrix, 419
   
 wind-noise disturbances, 14

**Y**

yaw-axis control systems, 16, 20–21

**Z**

zero-order hold, 102, 119, 134, 152, 169, 178, 180–181, 202–203, 223
 

- frequency response of, 116
- input and output signals for, 114
- transfer function of, 115

 zero-order-hold transfer function, defined, 104
   
 $z$ -transform, 38, 203, 234, 247, 308, 379
 

- of delayed time function, 136
- double-sided, 38
- examples, 38–40
- finding, 45–46
- generated using MATLAB, 46
- modified, 136–139
- region of existence in complex plane, 40
- single-sided, 38
- of state equations, 81–84
- transfer function, 145

 $z$ -transform, properties of, 45
 

- addition and subtraction, 40
- complex translation, 43
- of  $e(k)$ , 41
- final-value property, 44, 135
- initial-value property, 44
- inversion integral, 41
- multiplication by a constant, 40–41
- real translation, 41–42