CONTROL SYSTEM DESIGN
A State-Space Approach

Richard J. Vaccaro
University of Rhode Island

To the memory of my parents: Robert and Marion Vaccaro

PREFACE

This book is intended as the text for a senior-level course on the analysis and design of digital control systems. It is assumed that the reader has had a first course in control theory.

The systems that we want to control, for example servo positioning systems, chemical systems, or aircraft, operate in continuous-time. The digital control system, on the other hand, operates in discrete time. In digital control, the measured feedback signals are sampled with A/D converters and processed in some way to produce a control signal. The control signal is passed through a D/A converter connected to the input to the plant. Digital control systems are sometimes called *sampled data control systems* to emphasize the fact that the measured feedback signals are sampled. Our approach to digital control systems design has two steps: the first step is to obtain a discrete-time model for the plant. This model describes the behavior of the analog plant at sampling instants. The second step is to design a discrete-time compensator for the discrete-time plant model. The digital control system is then obtained by connecting the discrete-time compensator to the analog plant with A/D and D/A converters. We show how these two steps can be performed using state-space models.

A primary goal for this book is to be useful for an undergraduate course. I believe that the commitment to undergraduate accessibility will be appreciated not only by undergraduates, but also by graduate students and practicing engineers who are learning state-space design techniques for the first time. A comment on the motivation for the approach taken in this book is in order.

The use of state-space models became synonymous with "modern control theory" in the 1960's. Although other useful approaches to control systems design (robust, adaptive, etc.) were developed in the '70s and '80s, these approaches have not been integrated into

the undergraduate curriculum. A premise of this book is that state-space control theory is a useful tool in solving a variety of control problems, and that this tool can be mastered at the undergraduate level. By focusing on one set of tools (state space) it is possible to give a detailed development of the theory.

State space control theory is usually taught at the graduate level using continuous-time models. In my experience, state-space theory can be mastered at the undergraduate level if discrete-time models are used. This is possible because the discrete-time theory is much easier to develop that the corresponding continuous-time theory. For example, the solution to the discrete-time state equations can be done trivially by induction (as opposed to differential equation theory needed in continuous time). Also, the derivation of the rank test for controllability of discrete-time systems is based on the theory of linear equations. The derivation is constructive in the sense that an input sequence that demonstrates controllability can be obtained as the solution to a system of linear equations.

This book shows how linear algebra and state-space system theory are used to design digital control systems. In addition, classical frequency-domain tools such as Bode and Nyquist plots are used to analyze digital control systems designed by state-space techniques. It is assumed that the reader has had a first course in control theory and a signals/systems course that has introduced Laplace and z transforms. In addition, some prior exposure to linear algebra and matrices would be helpful. A computer disk is included with this book containing the **Digital Control Toolbox**. This toolbox is a collection of MATLAB programs that implement all of the design and analysis tools covered in this book. The first version of the programs and documentation was written by Tom Flint. Many of the homework problems require the use of the toolbox, or equivalent software.

A brief description of each chapter follows.

Chapter 1 provides an introduction and motivation for the book.

Chapter 2 presents the linear algebra and matrix theory needed for the rest of the book. Most of the book relies only on the theory of linear equations and eigenvalues/vectors. Chapters 9 and 10 require more advanced concepts such as row and column spaces, their orthogonal complements, and projections. It is not necessary to cover Chapter 2 in its entirety before proceeding with the rest of the book.

Chapter 3 describes state-space and transfer function models for continuous-time and discrete-time systems. In addition, state-space and transfer function models for several example systems are derived.

Chapter 4 lays the foundation for the design techniques presented in the remainder of the book. It is shown that when a linear, continuous-time plant is operating under digital control, it is possible to replace the continuous-time model of the plant by a discrete-time model that describes the behavior of the plant *exactly* at sampling instants. This chapter also shows how to obtain exact discrete-time models for plants with time delay at the input. These models are used in Chapter 6 to design digital regulators and to assess the effect of computational time delay.

Chapter 5 presents tools for frequency-domain analysis – Bode plots and Nyquist plots for discrete-time systems – and discusses stability and gain/phase margins. These tools are used to assess the robustness of control systems designed in later chapters. The topic of steady-state error and system type is also covered.

Chapter 6 shows how to design regulators using digital state feedback. Normalized Bessel polynomials are introduced as a simple way to choose closed-loop pole locations to

achieve a desired speed of response for the closed-loop system. The robustness of the system is assessed by looking at a Nyquist plot of the loop transfer function. The relationship between state feedback and PID control is presented.

Chapter 7 considers the design of observers to estimate the value of the state vector of the plant from measurements of its input and output. Both full and reduced-order observers are presented, and particular attention is given to the case in which some but not all of the state variables can be measured. We show that if at least half of the state variables can be measured, then an observer can be designed by solving a system of linear equations (i.e. *without* having to solve an eigenvalue placement problem). We also show how to design reduced-order observers when less than half of the state variables are measured. A problem can arise when using observers in a control system. If the observer is not designed properly it is possible that the stability margins of an observer-based regulator are much worse than the stability margins of the state-feedback regulator. In this chapter we show how to design robust observers that recover much if not all of the stability margins of the state-feedback regulator.

Chapter 8 shows how to introduce a reference input into a regulator structure to obtain a tracking system. The tracking and regulation problems are unified by introducing the concept of *zero-input state trajectories*. It is seen that tracking with zero steady-state error often requires the use of additional dynamics in the compensator, as opposed to simply using state feedback for regulation. A complete algorithm to design tracking systems that are robust with respect to persistent disturbances and model inaccuracies is given. The algorithm is a solution to the "robust servomechanism" problem introduced in the '70s, and uses the "internal model principle." This algorithm provides a straightforward way to design tracking systems, and is rooted in classical concepts of system type and steady-state error. The tracking systems in this chapter are designed using full state feedback as well as observers.

Chapter 9 considers multivariable regulation and tracking systems. For multivariable systems, the matrix of feedback gains that places the poles of the closed-loop system in desired locations is not unique. This chapter presents a recently-developed algorithm that finds the matrix of feedback gains that optimizes a robustness criterion. This chapter also includes the multivariable version of the tracking algorithm from Chapter 8.

Chapter 10 is an introduction to optimal control. The linear quadratic regulator for multivariable systems is derived. Only the steady-state (infinite horizon) case is considered, and so the solution to the optimal regulation problem is constant state feedback. The feedback gain matrix is computed directly from the eigenstructure of the Hamiltonian matrix without explicitly using the Riccati equation.

RICHARD J. VACCARO

Kingston, RI
September, 2018

ACKNOWLEDGMENTS

My interest in systems and control began as an undergraduate student at Drexel University. I would like to thank Professors Bruce Eisenstein, John D'Alessandro, Richard Klafter, and Hun Sun for providing not only facts and information but also motivation and encouragement for further study. As a graduate student at Princeton University I benefited greatly from courses and discussions with Professors Bede Liu, Stuart Schwartz, and especially my Major Professor, Bradley Dickinson.

The Department of Electrical, Computer, and Biomedical Engineering at the University of Rhode Island is a wonderful place to work. Louis Scharf was department chair when I came to URI and he has provided advice and encouragement ever since. I have had many helpful discussions with my colleagues at URI including Professors Peter Swaszek, Faye Boudreaux-Bartels, Ramdas Kumaresan, Leland Jackson, Al Lindgren, Don Tufts, and Chuck Turtle. My students in ELE 458 and ELE 503 have provided feedback on preliminary versions of this book. I would particularly like to acknowledge two of my graduate students. Simon Hill set up our first digital control laboratory. Tom Flint wrote the first version of the Matlab toolbox that accompanies this book.

Finally, I want to acknowledge a great personal debt to my family. My wife, Sarah, and children, Robert, Bethany, Anna, John, and Nathaniel, are a constant source of joy and inspiration. They are God's gifts to me. My parents, Bob and Marion Vaccaro, taught me by word and example what is truly important. I have dedicated this book to their memory. They set an example I try to follow by living out the admonition in the following Scripture:

> Thus says the Lord:
> "Let not the wise man glory in his wisdom,
> Let not the mighty man glory in his might,
> Nor let the rich man glory in his riches;
> But let him who glories glory in this,

That he understands and knows Me,
That I am the Lord,
exercising lovingkindness, judgement,
and righteousness in the earth.
For in these I delight," says the Lord.
(Jeremiah 9:23-24)

-RJV

CHAPTER 1

INTRODUCTION

1.1 A Motivating Example

Consider the problem of trying to balance a yardstick on the palm of your hand. It is clear that some type of *control* has to be exerted to keep the stick from falling over. If proper control is used, the stick will remain balanced in the vertical position. This is an example of a *regulation* problem, where the goal is to keep a system near its equilibrium point.

Now consider the problem of walking along a prescribed path while balancing the stick. Keeping your position along the path is an example of a *tracking* problem, which must be accomplished while regulating (balancing) the stick.

A system that incorporates the main features of the stick-balancing-while-walking problem is the pendulum-on-a-cart system shown in Fig. 1.1. The cart is constrained to move in only one dimension along the x axis, and the pendulum can only move in one plane, as indicated by the angle θ. A typical control problem for this system is to regulate θ and x near zero in spite of disturbances on the pendulum and/or cart. This type of stabilization problem is similar to launching a rocket. The rocket is represented by the pendulum, which will fall over unless some type of control is used. In the pendulum-cart system, the stabilizing torque is provided by the motion of the cart, while for a rocket, torque is provided by gimballed thruster engines. One other difference is that the rocket problem has two degrees of freedom – the rocket must be balanced about two orthogonal axes – while the pendulum has only one degree of freedom. Nevertheless, a control system which regulates the pendulum-cart system will contain the essential ingredients of one axis of a rocket control system.

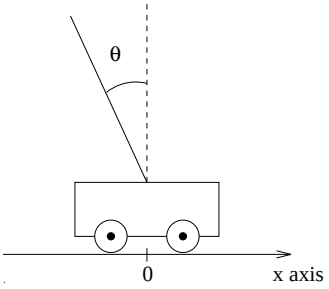


Figure 1.1 Inverted pendulum on a moving cart.

Consider now a related problem called the crane problem. A crane operator would like to move his payload from one location to another in such a way that he reaches the destination without his payload oscillating and perhaps inadvertently demolishing a nearby building! The crane can be represented by a pendulum-cart system in which the pendulum hangs down from the cart. A typical control problem for this system is to move the cart from one position to another while keeping the pendulum nearly vertical. This problem involves both regulation and tracking.

This book presents tools for analyzing and designing control systems. It is assumed that the control will be implemented by a digital computer, and the tools presented here are appropriate for digital control of continuous-time systems. In addition, it will be seen that the mathematical representations of primary importance are state-space models, which are at the foundation of modern control theory. Thus the emphasis of this book is on state-space design techniques for digital control. However, we continue this introduction by first reviewing classical control theory.

1.2 Classical Control Theory

The prototype control system that one encounters in classical control theory is shown in Fig. 1.2, where $G(s)$ is the transfer function of the system (also called the *process* or *plant*) to be controlled. The measured feedback signal is just the plant output $y(t)$. The design problem is to specify the transfer function $C(s)$ so that the closed-loop system has desired characteristics. These characteristics include stability and perhaps some requirements on the step response such as the percent overshoot, settling time, and steady-state error. The signal $d(t)$ in Fig. 1.2 is a disturbance acting on the plant, and another goal of the control system is *disturbance rejection*. That is, the control system should reduce the effect of the disturbance on the output of the plant. In classical control theory, $d(t)$ would be added either to $u(t)$ or $y(t)$. Specific disturbance models are considered in Chapter 8. In addition to stability, step response specifications, and disturbance rejection, a fourth characteristic of a control system is *insensitivity to errors in the plant model*. The model of the plant $G(s)$ may contain parameters whose values are not known precisely. The control system should perform the same despite these uncertain parameter values. The analysis and design of analog control systems with these characteristics is covered in numerous books including [21, 23, 31, 55, 70, 74].

In Fig. 1.2, $r(t)$ is the *reference input*. If $r(t)$ is zero for all time, then $C(s)$ is called a *regulator*, and the design of $C(s)$ is called the *regulation problem*. In this case the purpose

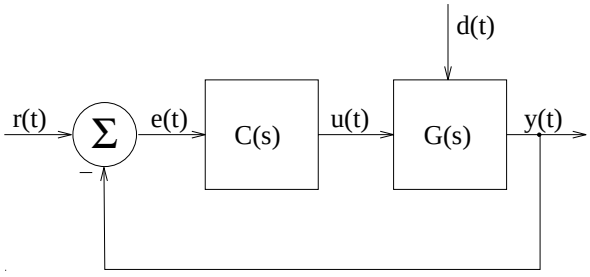


Figure 1.2 An analog control system.

of $C(s)$ is to keep the output $y(t)$ at zero in the face of external disturbances on the system. A generalization of the regulator problem is the *setpoint problem*. In this case $r(t)$ is a constant and the purpose of $C(s)$ is to keep the output at a constant reference value, called the setpoint, in the face of external disturbances on the system. If $r(t)$ is not a constant for all time, then $C(s)$ is called a *compensator* or *controller* and the design of $C(s)$ is called the *servo* design problem. The signal $e(t)$ is the error signal between the reference input and the actual output of the plant, and $u(t)$ is the input signal to the plant. Note that $u(t)$ is generated by the compensator to make the plant behave in a desired way.

Let us now consider how the control of the cart-pendulum system might be formulated using classical control theory. In the usual first undergraduate control course, one learns how to design a compensator to control the behavior of a single variable, the output, as shown in Fig. 1.2. However, for a cart-pendulum system, there are at least two variables of interest – the position of the pendulum and the position of the cart. Suppose one wanted to design a system to control the behavior of both of these variables. Specifically, consider the inverted pendulum, and suppose that the objectives were to keep the pendulum balanced as well as to keep the cart at $x = 0$ in the face of disturbances.

A simple classical approach would be to design separate control loops for each variable considered as an output, and sum the resulting control signals, as shown in Fig. 1.3. $C_1(s)$

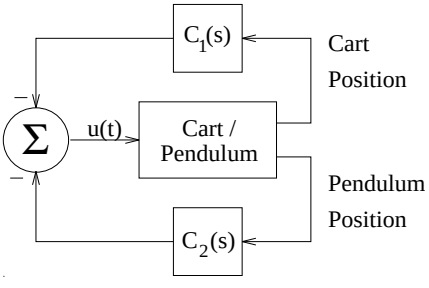


Figure 1.3 A two-loop analog control system for the cart/pendulum.

would be designed using the transfer function from $u(t)$ to pendulum position as the plant, while $C_2(s)$ would be designed using the transfer function from u(t) to cart position as the plant.

Unfortunately, the two control loops (one for the pendulum, the other for the cart) have competing objectives, and they would "fight" each other. This is illustrated by the follow-

ing scenario: suppose the pendulum is balanced with the cart at $x = 0$ and a disturbance occurs which knocks the pendulum to the right (clockwise). In order to balance the pendulum, the pendulum control loop would tell the cart to accelerate to the right. However, as soon as the cart began to move, an error would be introduced in cart position (which was specified to be zero), and the cart control loop would tell the cart to move to the left to reduce this error. This is what was meant by competing objectives – the pendulum control loop tells the cart to move to the right while the cart control loop tells the cart to move to the left!

By a trial and error process, a compensator could be designed which would accommodate the competing objectives. However, classical control theory does not provide convenient mathematical tools for coordinating the control of several variables. State space techniques, on the other hand, will be shown to provide a simple but powerful framework to handle the type of situation just described.

1.2.1 Classical Design Tools

The design of an analog compensator is usually made with the aid of frequency domain plots such as Bode or Nyquist plots, or with s-domain plots such as the root-locus plot. The design techniques are usually graphical procedures for designing simple first- or second-order compensators (such as lead or lead-lag). In addition, many classical design rules involve approximations since they are based on the analysis of the "standard" second-order system, i.e. a system with two poles and no zeros. In spite of what has been said above, the classical design procedures are still useful in the following situations:

1. Whenever the plant is a standard second-order system.

2. Whenever the plant has a pair of "dominant poles", so that it can be approximated by a standard second-order system.

3. Whenever a precise mathematical model of the system to be controlled is not available. In this case, experimental data (such as a Bode plot) can still be used to design a compensator.

4. Graphical tools are useful for analyzing control systems to obtain information regarding relative stability; for example, gain and phase margins.

1.2.2 A Word on Terminology

In this book, the term *analog* is meant to be synonymous with "continuous-time," and the term *digital* is meant to be synonymous with "discrete time." A more correct use of the term "digital" refers both to amplitude quantization of signals, as well as time discretization. However, from the point of view of actual control systems, the use of 12 bit A/D and D/A converters usually results in a negligible amplitude quantization of signals. Hence we can use the terms "digital" and "discrete time" interchangeably.

1.3 State-Space Control Theory

The prototype control system that one encounters in state-space control theory is a regulator structure like the one shown in Fig. 1.4.

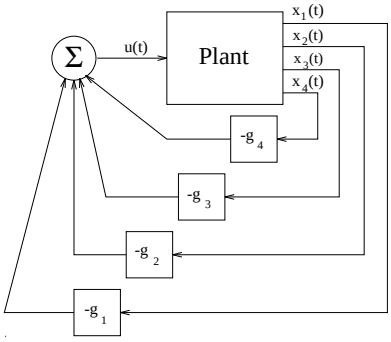


Figure 1.4 An analog state-space control system (regulator) for a fourth-order plant

The system under control (the plant) is characterized by a number of "state variables" – in Fig. 1.4 there are four state variables labeled $x_1(t)$, $x_2(t)$, $x_3(t)$, and $x_4(t)$. The measured feedback signals are the n state variables. The control input $u(t)$ is the sum of the state variables $x_i(t)$ each multiplied by a constant gain g_i. Since this is a regulation problem, there is no external input to the system, or equivalently, the external input is zero. The design problem is to specify the gains $g_1, \cdots, g_4$ which weight the state variables in such a way that the resulting input signal keeps the state variables near zero in the face of disturbances. Note that Fig. 1.4 is an analog control system because the state variables are not sampled. Chapter 6 will discuss the design of regulators using sampled state variables, but sampling is not important for the present discussion.

Recall the cart-pendulum regulation problem discussed in the previous section. Let us now consider how this problem would be formulated using state-space techniques. First a *state-space description* of the cart-pendulum system would have to be obtained. In Fig. 1.4 this would be a set of equations which describe how the input $u(t)$ produces the state variables $x_i(t)$. Such a model will be developed in Chapter 3. It will be seen that the state variables can be chosen as follows: $x_1(t)$ is the angular position of the pendulum, $x_2(t)$ is the angular velocity of the pendulum, $x_3(t)$ is the position of the cart, and $x_4(t)$ is the velocity of the cart. By calculating the feedback gains $g_1 \cdots g_4$, it is possible to design a regulator which will make *all* the state variables go to zero!

In the previous section on classical control theory, it was noted that regulation of cart position is a competing objective with the regulation of pendulum position, and that individual control loops designed for each variable would "fight" each other. How does the state-space approach handle this coordination of variables? State-space theory handles the coordination *automatically* through the mathematical description of the plant, which describes how the state variables are related to each other and to the input. In addition, state-space theory provides a test to determine whether all the state variables can be regulated to zero by a single input (or by several inputs in the case of multivariable systems). Specifically, if the plant can be shown to be *controllable*, then all of the state variables can be regulated to zero. The test for controllability can answer the following question: "Is it possible to balance two pendulums simultaneously on a single cart?" The controllabilty test shows when this is possible, and when it is possible, the state feedback design formulas tell you how to do it! (The answer: it is possible to balance two pendulums simultaneously if they are of different lengths.)

State-space theory can be extended to handle tracking problems where a reference trajectory is specified for the output(s) of the system. It is possible to handle combined reg-

ulation/tracking problems in which some of the state variables are to be regulated to zero, while others are to follow a reference trajectory. Here again, the state-space model provides an automatic coordination of all the variables involved. A dramatic example of this occurs for the inverted pendulum-cart system. Suppose, instead of the pure regulation problem discussed earlier, the problem includes a change in cart position. That is, it is desired to move the cart from point A to point B (where B is to the right of A, for example) while keeping the pendulum balanced. If a control system designed using state-space techniques is employed for this task, and the command is given to move from A to B (to the right), a surprising phenomena will occur. The cart will initially move to the left! The reason, in retrospect, is simple. If the cart initially moved to the right as commanded, the pendulum would begin to fall to the left, and the only way to stop the pendulum from falling over would be for the cart to move to the left. In other words, the cart could never make progress towards point B without the pendulum falling over.

Somehow, the control system is "smart" enough to know this and it moves the cart in the wrong direction long enough to get the pendulum leaning to the right. Then when the pendulum is leaning "just enough," the cart moves to point B in such a way that the pendulum is upright just as the cart arrives. How did the control system get "smart enough" to know to move the cart in the wrong direction and how does it know when the pendulum is leaning "just enough" to move the cart to the right again? The answer to these questions is that the state-variable model of the plant provides a complete description of the interrelationships of all the variables, and the knowledge mentioned above is encoded in the model.

This example illustrates two important points about state-space techniques. The first is that they are quite powerful. They naturally and automatically handle the coordination of many variables. Using the language of linear algebra, the same notation can be employed to describe the control of a second-order system or a tenth-order system. In addition, the design for high-order systems uses the same linear algebraic formulas as for low-order systems. The only difference is that the formulas for high-order systems require a computer to perform the calculations.

The second important point about state-space techniques is that the power that they exhibit is due largely to the state-space model of the plant. If this model is accurate, then the theory provides excellent results. If the model of the plant is not an accurate reflection of the actual system, however, the results could be quite poor. This brings up the subject of *robust control* which provides techniques for designing control systems for plants whose mathematical description is not accurate. In this book the robustness of control systems is analyzed using classical stability margins (gain and phase margins) as well as a state-space perturbation formula. For more information on designing control systems by optimizing a robustness criterion, see [25, 45, 66]. The state-space design tools which will be presented in this book are summarized next.

1.3.1 State-Space Design Tools

Pole Placement by State Feedback The primary tool for designing regulators using state-space models used in this book is pole placement by state feedback. It will be shown that if the plant satisfies a certain mathematical property (controllability), then the poles of the closed-loop system can be placed in arbitrary desired locations by feeding back a constant linear combination of the state variables. Pole placement can be achieved for a system of any order, and the calculation of the feedback gains reduces to the solution of a system of linear equations.

Pole placement can be compared to the root-locus classical design tool. The root locus can be used to calculate the value of a single gain in order to place the closed-loop poles. However, the closed-loop poles cannot be placed in arbitrary locations; they must lie on the root locus. This is to be contrasted with pole placement by state feedback which allows for the calculation of n gains (for an nth-order system) to exactly place all the poles in arbitrarily specified locations.

It turns out that the stability margins for a state-feedback regulator depend on the closed-loop pole locations. One drawback of the pole-placement design procedure is that the designer must pick pole locations without any guarantee on the stability margins of the resulting regulator. Once the regulator is designed these stability margins must be checked (using a Nyquist plot, for example). If the stability margins are not adequate, the designer must pick a new set of closed-loop pole locations and design another regulator.

In Chapter 9 we consider the design of digital control systems for plants which have several inputs and outputs. In this case the specification of desired closed-loop poles does not result in a unique set of feedback gains. This lack of uniqueness is exploited by an algorithm which searches among all the sets of gains corresponding to the desired closed-loop pole locations for the set which optimizes a robustness criterion.

Observers The implementation of a pole-placement regulator requires that all the state-variables of the plant be measured with sensors in order to be used for feedback. This requirement may be overly restrictive: there may be some state variables that are difficult (and therefore expensive) to measure, and there may be some state variables that do not have direct physical significance. In cases where only a subset of the state variables are measured, it is possible to design a system called an *observer* which estimates the values of the unmeasured state variables from the values of the measured state variables and the input(s) to the plant. The combination of an observer together with state feedback gains yields a *dynamic* regulator, i.e. a regulator with poles and zeros, as opposed to a state-feedback regulator which is a set of constant gains.

A problem can arise when using observers in a control system. If the observer is not designed properly it is possible that the stability margins of an observer-based regulator are much worse than the stability margins of the state-feedback regulator. We show in Chapter 7 how to design observers which recover much if not all of the stability margins of the state-feedback regulator.

Tracking Systems

It is possible to introduce a reference input into a regulator structure to obtain a tracking system. Tracking and regulation problems are unified by introducing the concept of *zero-input state trajectories* in Chapter 8. It is seen that tracking with zero steady-state error often requires the use of additional dynamics in the compensator, as opposed to simply using constant state feedback. A complete algorithm to design tracking systems which are robust with respect to persistent disturbances and model inaccuracies is given in Chapter 8. This algorithm provides a straightforward way to design tracking systems, and is rooted in classical concepts of system type and steady-state error. The tracking systems are designed using full state feedback as well as observers.

1.4 Digital Control Theory

A digital control system can be obtained from the analog control system shown in Fig. 1.2 by formally replacing the compensator $C(s)$ by A/D, D/A converters and a discrete-time system $C(z)$ as shown in Fig. 1.5[1]. Note that the signal $\bar{u}(t)$ generated by the digital com-

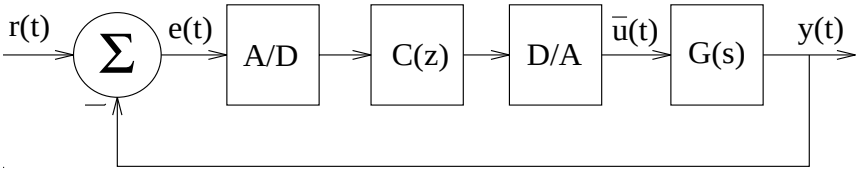


Figure 1.5 A digital control system

pensator in general can never be made exactly equal (for all time) to the signal $u(t)$ generated by the analog compensator in Fig. 1.2. Thus a digital control system will always have different performance than an analog control system. It will be seen that digital control can give essentially equal performance, better performance, or worse performance, depending on the design procedure used. For now, only the following points are emphasized:

1. $C(z)$ is implemented on a computer (perhaps a microprocessor).

2. The input to $C(z)$ is a sequence of numbers (the sampled error signal).

3. The output of C(z) is a sequence of numbers which is converted to a *piecewise constant* control signal, $\bar{u}(t)$, by the D/A converter (i.e. the D/A converter is a zero order hold). Note that the output of the analog compensator $u(t)$ is in general never constant. This difference between $u(t)$ and $\bar{u}(t)$ accounts for much of the difference between analog and digital control. An example of this difference is given in section 1.5.

4. The A/D takes samples at uniformly spaced *sampling instants*. The D/A converter can change its output value only at these same sampling instants. The A/D and D/A converters are synchronized by a clock, which is not shown in Fig. 1.5.

1.4.1 Digital Control Design Tools

In this book, the use of state-space models and linear algebra will be emphasized. A state-space model of the plant (which will always be assumed to be a continuous-time system) will be used to calculate a discrete-time state-space model which specifies the behavior of the plant only at the sampling instants. The primary design tool will be pole placement by state feedback, or by output feedback with an observer. These design procedures are simple applications of linear algebra, and are not graphical. The calculations can be done by hand for low-order systems (order ≤ 3), but for higher order systems, a computer program is necessary. A computer disk containing a set of MATLAB programs called the **Digital Control Toolbox** is available to accompany this book (see Preface). The toolbox contains

[1] A/D and D/A converters, which convert analog signals to digital signals and vice-versa, are discussed in Section 1.4.3.

programs which implement the design and analysis procedures presented in this book, as well as programs for simulating digital control of analog plants.

Note that the compensator only receives samples of the error signal at discrete intervals of time, i.e. only at the sampling instants. Likewise, the compensator only changes, or updates, the input signal to the plant at the sampling instants. The compensator can not respond to anything that happens between the sampling instants. Strictly speaking, then, a digital control design procedure can only specify desired system performance at the sampling instants. Inter-sample behavior is usually examined by simulation. The Digital Control Toolbox contains programs which can simulate the interconnection of continuous- and discrete-time systems.

Digital control offers the following advantages over analog control.

1. The control algorithm is easily changed by reprogramming $C(z)$.
2. Controller performance does not change with changing environment or passage of time.
3. Digital control can give performance which is arbitrarily close to the performance of a given analog controller by sampling "fast enough."
4. Digital control can give performance which is superior to that obtainable by analog control.

1.4.2 Methods of Digital Control Design

The following list gives a brief description of different methods for designing digital control systems. The state-space design procedures developed in this book are an example of the third method listed below.

1. The first method of digital control design to be developed was to discretize a given analog controller [32, 73]. Note that this discretization process is identical to that used in the context of filter design where a digital filter can be obtained as an approximation to a given analog filter by some discretization process [22, 80]. This method is illustrated in Problem 2 in Chapter 4.
2. Obtain a discrete-time transfer function model $G(z)$ for the plant. Map the z plane into another complex plane called the w plane and obtain $G(w)$. Although $G(w)$ is a discrete-time system, The w plane has certain similarities to the s plane. Thus one can use classical control theory (e.g. Bode plots) to design a compensator $C(w)$. Finally, map $C(w)$ into $C(z)$ to obtain the transfer function of the digital controller. This design technique is covered in the following references, [32, 73].
3. Obtain a discrete-time state-space model of the plant. Then do a direct digital design using discrete-time state-space control theory. This method yields the desired system performance at the sampling instants.

From a theoretical point of view, the major difference between the digital control system of Fig. 1.5 and the analog control system of Fig. 1.2 is that the digital control system "mixes" continuous- and discrete-time systems. For the purpose of design, the system must be made all digital or all analog. The third method of digital control design mentioned above makes the system all digital by replacing the plant with an equivalent discrete-time system. It will be shown that this discrete-time system is *not* an approximation to the plant, but it exactly describes the behavior of the plant at the sampling instants.

1.4.3 Sampling and Reconstructing an Analog Signal

Sampling Analog Signals This book deals with digital control of continuous-time systems. Thus the inputs and outputs of the plant are functions of a continuous time variable, while the inputs and outputs of the digital compensator are functions of a discrete time variable. This means that continuous-time signals have to be converted to discrete time and vice versa. These two conversion processes are briefly described below. For a more complete treatment, see [46].

Consider a continuous-time (analog) signal $f_a(t)$ shown in Fig. 1.6. As shown in the

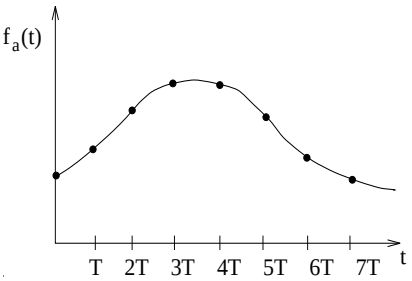


Figure 1.6 A sampled analog signal.

figure, the signal is sampled at uniformly spaced intervals of duration T seconds. The sampled points (represented by the dots in the figure) are just the sequence of numbers $f_a(kT),\ k = 0, 1, \cdots$. This sequence of numbers can be thought of as a discrete-time sequence with time index k; that is, we can *define* the discrete-time sequence $f_d(k)$ as follows:

$$f_d(k) \stackrel{\text{def}}{=} f_a(kT). \tag{1.1}$$

A common abuse of notation is to ignore the distinction between $f_a(\cdot)$ and $f_d(\cdot)$ by calling the analog signal $f(t)$ and calling the discrete-time sequence $f(k)$. One way of explaining this poor notation is to say that the sampling interval T equals 1 second. However, this explanation always leads to the question of what to do when the sampling interval does not equal 1 second! The proper way to view a sampled signal is by the equivalence given in (1.1). In the remainder of this book, a shorthand notation will be used instead of the subscripts "a" and "d" . The notation is as follows: square brackets will be used for discrete-time indices while parentheses will be used for continuous time. If the same letter is used for an analog signal and a digital signal, the digital signal will contain samples of the analog signal. Thus (1.1) will be expressed as

$$f[k] \stackrel{\text{def}}{=} f(kT). \tag{1.2}$$

In order for samples of an analog signal to contain all of the information present in the signal, the sampling interval T must be "small enough." The sampling theorem due to Nyquist and Shannon [46] states that if the highest frequency component in an analog signal is $\omega_0 = 2\pi f_0$ radians per second, then the sampling frequency $f_s = 1/T$ must be greater than $2f_0$. This requirement is equivalent to

$$T < \frac{1}{2f_0} = \frac{\pi}{\omega_0}. \tag{1.3}$$

When choosing the sampling rate for a digital control system, there are other considerations besides (1.3). These other considerations, which are covered in later chapters, insure that (1.3) is satisfied.

In practice, a device which produces samples from an analog signal is called an *analog-to-digital, or A/D, converter*. An A/D converter produces a binary representation with some fixed number of bits, typically 8, 12, or 16. Because an A/D converter only uses a finite number of bits, the samples it produces are *quantized in amplitude*. In the early days of digital control, much effort was expended in calculating the effect of this *quantization noise* on the behavior of the control system. However if either 12 or 16 bit converters are used, the quantization noise is usually negligible. Good treatments of this topic may be found in [45, 51, 68].

Another implication of the finite binary representation is that an A/D converter can only represent an analog signal over a *finite dynamic range*. For instance, an A/D converter may be able to represent an analog signal which takes values between -5 and 5. An analog signal input to this A/D converter must be scaled to fit within this dynamic range. The finite range of an A/D converter is a much more important problem for control system design than the quantization effects.

Reconstructing Analog Signals From Digital Samples The process of converting digital samples to an analog signal which "connects" those samples in some way is called *reconstruction*. A device which performs this reconstruction is called a *digital-to-analog, or D/A, converter*. The most common form of reconstruction used in practice is *zero-order hold (ZOH)* reconstruction. A ZOH reconstruction of a sequence of samples produces a constant output value which is proportional to a given input sample for a fixed amount of time. Then the output changes to a new constant value which is proportional to the value of the next sample. The influence of ZOH reconstruction on the modeling and performance of digital control systems is covered in Chapter 4.

D/A converters also use a finite binary representation which means that they have a *finite range of output values*. For instance, the output of an D/A might range from -5 to +5 volts. If the digital signal that is the input to a D/A converter exceeds this range, the D/A will *saturate* at ± 5 volts.

The operation of a D/A converter which implements ZOH reconstruction is shown in Fig. 1.7. The term *zero order* means that the sample points are interpolated by polynomials

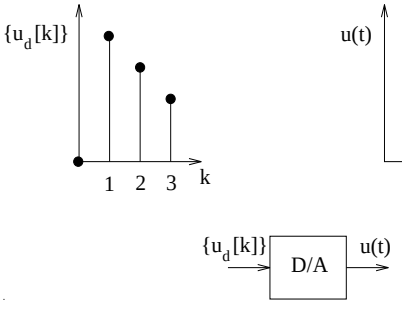


Figure 1.7 A D/A converter that implements zero-order-hold reconstruction.

of degree zero (constants). Higher-order holds are obtained when polynomials of higher degree are used to interpolate the samples. For example, a first-order reconstruction con-

nects the sample points by lines (first-order polynomials). In this book, D/A converters will always be assumed to be implementing zero-order-hold reconstruction, since this type of D/A converter is the most common one used in practice.

1.5 Example

A common control problem is the position control problem. Suppose for instance that we want to control the angular position of a motor shaft. Fig. 1.2 could be a block diagram for this system where $G(s)$ represents the motor, and $C(s)$ is the analog compensator (say a phase lead compensator). Suppose we want this control system to respond to a step change in desired angular position. If we assume that the original angular position was zero, and the desired angular position is unity, then the input to the control system is simply a unit step.

Suppose now we wanted to use a digital compensator to control the same motor, using the system shown in Fig. 1.5. It is possible to design a digital compensator that is essentially equivalent to a given analog compensator by choosing a sampling interval that is "small enough." How might the performances of the analog and digital control systems compare? In particular, what will the step responses look like, and what type of control signal (i.e. input signal to the motor) will be generated?

A typical step response is shown in Fig. 1.8, as well as a typical control signal $u(t)$ generated by an analog compensator in such a situation. The step responses using analog and digital control are indistinguishable in Fig. 1.8. Fig. 1.8 also shows the control signal of the system with a digital compensator and a sampling period of 0.05 secs. It can be seen

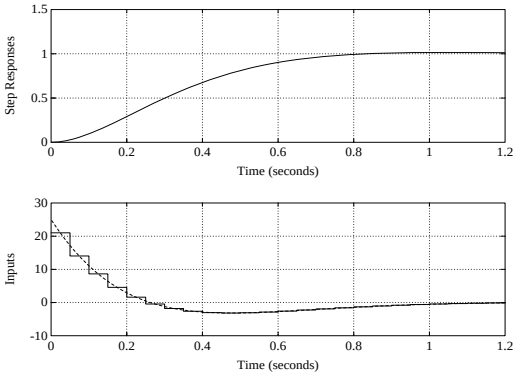


Figure 1.8 Step responses and control inputs of analog (dashed lines) and digital (solid line) control systems. The step responses of the systems with analog and digital compensators are identical in the top plot. The sampling period for the digital compensator is 0.05 secs., as can be seen from the "staircase" digital control signal (solid line) in the bottom figure.

that the digital control system has essentially the same performance as the analog control system.

In addition to matching the performance of analog compensators, digital compensators can be designed to achieve superior performance. This is shown in Fig. 1.9 where the step response and control input are shown for a digital control system designed to give a "deadbeat" response. Note that the output has no overshoot and has a much shorter rise time than the responses shown in Fig. 1.8. In addition, the largest magnitude of the

control input in Fig. 1.9 is smaller than that in Fig. 1.8, so the deadbeat controller is not achieving better performance by using very large control signals. It can be seen that the

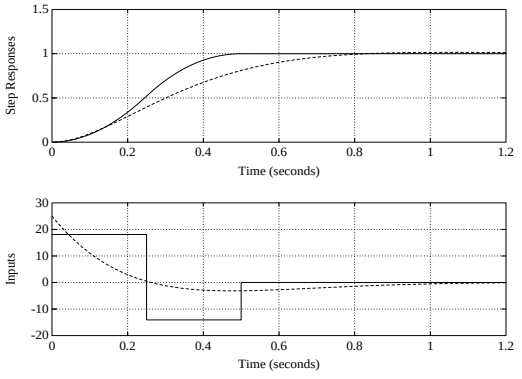


Figure 1.9 Step response and control input of an analog control system (dashed lines), and a digital control system (solid lines) designed for deadbeat response. The sampling period is 0.25 sec.

ability of the digital compensator to "hold" the command input constant for 0.25 seconds enables the motor to move quickly towards its new position. This cannot happen with a linear analog compensator because the control signal it generates will be sums of functions that are products of polynomials and exponential functions. We mention that the deadbeat digital control shown in this example can only be used in special situations satisfying the following requirements: the mathematical model for the plant must be highly accurate, there cannot be any disturbances acting on the plant, and the plant must be able to handle large discontinuities in its input.

1.6 Analog Position Control System

Consider a typical motor-driven positioning system. The system we want to control, called the "plant," is described by the transfer function shown in Fig. 1.10. Suppose that the

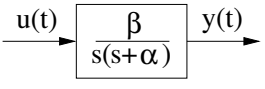


Figure 1.10 Transfer function of the plant for a motor-driven positioning system. The input signal $u(t)$ is the voltage applied to the motor power amplifier. The output signal $y(t)$ is sensor voltage proportional to motor position. β and α are positive real numbers whose values represent the behavior of a particular positioning system.

motor is initially at position zero, $y(0) = 0$, and we want to command it to go to a position corresponding to $y(t) = H$ volts. The reference command signal $r(t)$ is a step signal of height H. That is, $r(t) = 0$ for $t < 0$ and $r(t) = H$ for $t \geq 0$.

The simplest type of analog control system has *negative unity feedback* with *proportional* control, as shown in Fig. 1.11, where $G(s)$ is the transfer function shown in Fig. 1.10. The block diagram in Fig. 1.11 may be redrawn as the equivalent closed-loop system shown in Fig. 1.12. Using the "feedback formula," the system from $r(t)$ to $y(t)$ in Fig. 1.12 is

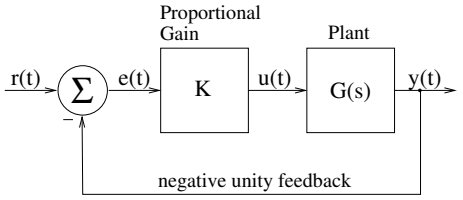


Figure 1.11 Motor-driven positioning system with proportional control. The plant is $G(s) = \beta/[s(s+\alpha)]$. Note that $u(t) = Ke(t)$; that is, the plant input is proportional to the error signal.

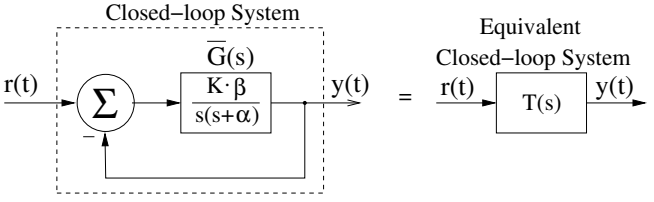


Figure 1.12 Equivalent closed-loop system for the motor-driven positioning system with proportional control shown in Fig. 1.11.

given by

$$T(s) = \frac{\bar{G}(s)}{1+\bar{G}(s)} = \frac{\dfrac{K\beta}{s(s+\alpha)}}{1+\dfrac{K\beta}{s(s+\alpha)}} = \frac{K\beta}{s^2+\alpha s+K\beta}. \tag{1.4}$$

Recall that the reference command signal $r(t)$ is a step signal of height H. So we would like the control system to drive the plant output $y(t)$ to the desired value of H. We now show that the control system shown in Fig. 1.12 has this desirable property. Let $Y(s)$ and $R(s)$ be the Laplace transforms of $y(t)$ and $r(t)$, respectively. Using the fact that $Y(s) = T(s)R(s)$ (see Fig. 1.12) and the fact that the Laplace transform of a step signal of heigh H is H/s (see Table 3.2) we see that $Y(s) = T(s) \cdot H/s$. The Final Value Theorem (see Table 3.1) says that

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} sY(s) = \lim_{s\to 0} sT(s)\frac{H}{s} = H \cdot T(0) = H \cdot \frac{K\beta}{K\beta} = H. \tag{1.5}$$

That is, $y(t) \to H$ as t gets large. Now that we know the system reaches the desired steady-state value, we may ask about the shape of the $y(t)$ signal as it goes from 0 to H. In other words, we are concerned about the *transient response* of the control system.

Recall that $y(t)$ is the step response of $T(s)$ (see (1.4)). The poles of this system are the roots of the denominator polynomial $s^2 + \alpha s + K\beta$. Note that this polynomial will have different roots depending on the value of the proportional gain K. There are three possibilities, shown in Fig. 1.13, for the roots of this denominator polynomial. By choosing the proportional gain, K, we can create a system with either an overdamped, underdamped, or critically damped response.

In order to get a critically damped response for the control system shown in Fig. 1.11 the denominator polynomial must have a repeated root on the negative real axis. The polynomial $(s+p)^2$ has a repeated root located at $-p$. This is the desired denominator polynomial, although we do not yet know the value of p, nor the value of the proportional

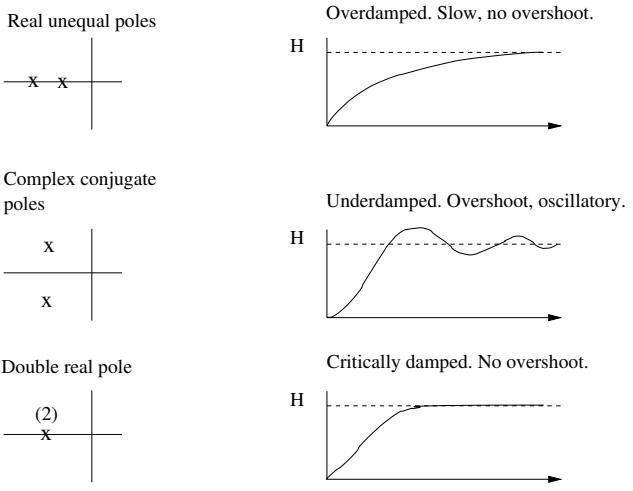


Figure 1.13 Figures on left show poles of 2nd-order system with corresponding step responses on right. All step responses converge to a steady-state value of H.

gain K that gives critical damping. Both of these unknowns are found by equating the actual denominator polynomial given in (1.4) with the desired denominator polynomial:

$$s^s + \alpha s + K\beta = (s+p)^2 = s^2 + 2ps + p^2. \tag{1.6}$$

Equating coefficients of s gives $\alpha = 2p$ or $p = \alpha/2$. Thus, the location of the double root is $-\alpha/2$. Equating the coefficients of s^0 (the constant terms) gives $K\beta = p^2$, or

$$K = \frac{p^2}{\beta} = \frac{\alpha^2}{4\beta}. \tag{1.7}$$

The previous equation shows the proportional gain value that gives a critically damped response. This can be confirmed by calculating the roots of the actual denominator polynomial as a function of K. Using the quadratic formula, the roots of $s^s + \alpha s + K\beta$, which are the poles of the closed-lop system, are given by

$$-\frac{\alpha}{2} \pm \frac{\sqrt{\alpha^2 - 4K\beta}}{2}. \tag{1.8}$$

When $K < \alpha^2/(4\beta)$ the poles are real and unequal, giving an overdamped response. When $K = \alpha^2/(4\beta)$ there is a repeated pole, giving a critically damped response. When $K > \alpha^2/(4\beta)$ there are complex-conjugate poles, giving an underdamped response.

1.7 Relationship Between Pole Locations and Settling Time

Definition The 1% settling time T_S is the time at which the step response enters and stays within 1% of its desired value H.

Consider a first-order transfer function $p/(s+p)$ with a step input of height H. The pole of this system is located at $-p$. The output (step response) of this system is $y(t) =$

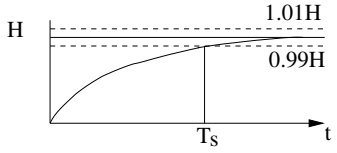


Figure 1.14 Step response of first-order system showing the settling time T_S.

$H(1 - e^{-pt})$. A plot of the step response is shown in Fig. 1.14 with the settling time T_S indicated. Because the step response does not oscillate, the settling time is the time at which the output reaches $0.99H$. The calculation of T_S is as follows:

$$y(t)|_{t=T_S} = 0.99H \Rightarrow 1 - e^{-pT_S} = 0.99 \text{ or } 0.01 = e^{-pT_S}. \tag{1.9}$$

Thus,

$$T_S = \frac{\ln(0.01)}{-p} = \frac{-4.62}{-p} = \frac{4.62}{p}. \tag{1.10}$$

The result may be summarized as follows:

The settling time for a **first-order system with pole at** $-p$ is given by

$$T_S = \frac{4.62}{p}. \tag{1.11}$$

For a 2nd-order system with a **double pole at** $-p$ there is a 45% increase in the settling time compared with the settling time of a first-order system with pole at $-p$. The result is stated as follows:

The settling time for a **second-order system with double pole at** $-p$ is given by

$$T_S = \frac{4.62}{p} * 1.45. \tag{1.12}$$

For complex-conjugate poles located at $-p \pm jq$ it can be shown that the settling time is approximately determined by the real-part of the poles:

The settling time for a **2nd-order system with complex-conjugate poles located at** $-p \pm jq$ is approximately given by

$$T_S = \frac{4.62}{p}. \tag{1.13}$$

For systems with more than one pole, the settling time is *approximately* equal to the settling time of the slowest pole. For example, consider a third-order system with poles

located at -1, -4, and -6. The response due to the poles at -4 and -6 decays to zero more quickly than the response due to the pole at -1. Thus, the settling time of this system is approximately equal to that of a first-order system with pole at -1. In general, for a system with real and complex poles, the settling time is approximately equal to the settling time of the slowest real pole or pair of complex-conjugate poles.

1.8 Design Examples for Analog Position Control Systems

Consider a motor positioning system with transfer function (see Section 1.6 for theoretical developments related to this example)

$$G(s) = \frac{120}{s(s+22)}. \tag{1.14}$$

The input to this transfer function is $u(t)$, the voltage applied to the motor power amplifier. The output signal $y(t)$ is a sensor voltage proportional to motor angular position such that 20 radians of motor position yields a 1-volt output.

In this section we will design three different control systems for this plant. Each of the control systems should drive the motor from an initial position of zero to a final value of 60 radians. Because the output position sensor has a scale factor of 20 rads/volt, the reference command signal $r(t)$ will be a step signal of height 3 volts. All three control system designs will be subject to the following constraints:

$$\begin{aligned} &\textbf{Constraint 1:} \quad y(t) \leq 3 \text{ (no overshoot)} \\ &\textbf{Constraint 2:} \quad |u(t)| \leq 5 \end{aligned} \tag{1.15}$$

1.8.1 Gain Compensator

In the first design we create a closed-loop system around the plant using proportional control. Use (1.7) to calculate the value of the proportional gain that gives critical damping. This design will then satisfy the first constraint in (1.15). The value of K is

$$K = \frac{22^2}{4 * 120} \approx 1. \tag{1.16}$$

Both of the closed loop poles are at -22/2=-11 (see the sentences after (1.6)). This proportional control system is shown in Fig. 1.15. The settling time of this control system is

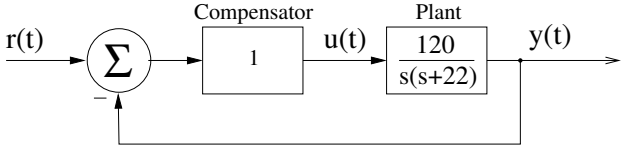


Figure 1.15 Proportional control system with critically damped response. Closed-loop poles are a double pole at -11. The settling time is 0.61 seconds.

given by (see (1.12))

$$T_S = \frac{4.62}{11} * 1.45 \approx 0.61 \text{ sec.} \tag{1.17}$$

The plant input and output signals of the proportional control system with a 3-volt reference input are shown in Fig. 1.16. Notice that the maximum plant input signal is only 3 volts. We

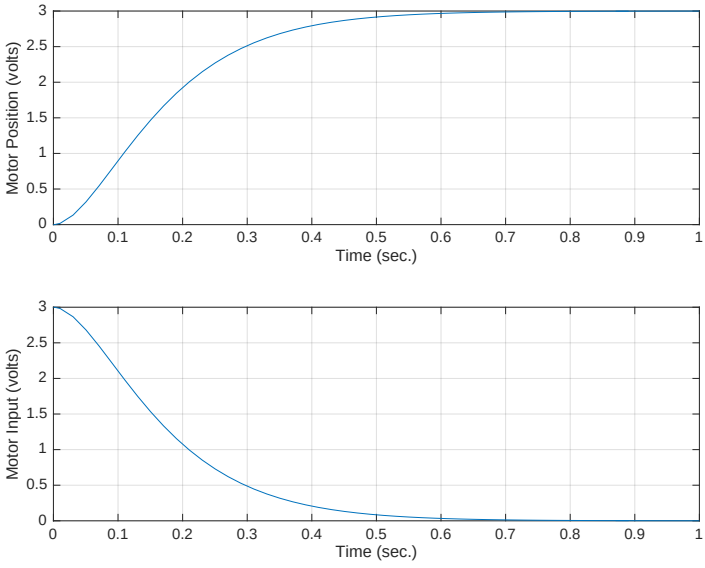


Figure 1.16 Response of proportional control system shown in Fig. 1.15 to a reference step input of 3 volts. Closed-loop poles are a double pole at -11. The settling time is 0.61 seconds.The maximum plant input is 3 volts.

should be able to make the system go faster because the motor input is permitted to be up to 5 volts. One way to make the system go faster is to increase the value of the proportional gain. However, this will result in an underdamped system whose step response overshoots the desired final value. There is a way to make the system go faster without overshoot by using a different type of compensator.

1.8.2 Phase-Lead Compensator

In the second design, shown in Fig. 1.17, we give the compensator a pole and a zero, in addition to a gain. The zero of the compensator is chosen to be -22, which cancels

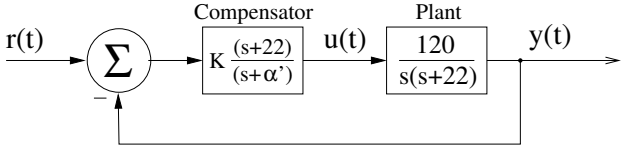


Figure 1.17 Control system with compensator containing a pole and zero. The zero of the compensator is chosen to cancel the plant pole at -22.

the motor pole at that location. The compensator pole is at $-\alpha'$, which will be chosen shortly. Notice that the forward-path transfer function (product of compensator and plant)

of Fig. 1.17 is

$$K\frac{(s+22)}{(s+\alpha')}\frac{120}{s(s+22)} = \frac{120K}{s(s+\alpha')}. \tag{1.18}$$

This is the same as the forward-path transfer function of Fig. 1.15 with the motor pole at -22 replaced by the motor pole at $-\alpha'$. If we choose $\alpha' > 22$ it is as if we have a faster motor! A first-order compensator whose pole is located to the left of its zero is called a phase-lead compensator. The design equation (1.7) that gave the gain value for critical damping in the first design may be used for this design with α replaced by α'. The initial value of the compensator output in Fig. 1.17 will be $u(0) = K(3 - y(0)) = 3K$. Letting $u(0)$ be the maximum allowable value of 5 volts gives $5 = 3K$ or $K = 5/3$. Rearranging (1.7) to solve for α and replacing α by α' yields

$$\alpha' = \sqrt{4\beta K} = \sqrt{4 \cdot 120 \cdot 5/3} = 28.28. \tag{1.19}$$

Thus, the compensator in Fig. 1.17 with $K = 5/3$ and $\alpha' = 28.28$ will give a critically damped step response with a maximum plant input of 5 volts. The closed-loop system will have a double pole at $-28.28/2 = -14.14$ and a settling time of $1.45 \cdot 4.62/14.14 = 0.47$ seconds, which is faster than the settling time of 0.61 seconds for the first design. The response to a 3-volt reference command is shown in Fig. 1.18.

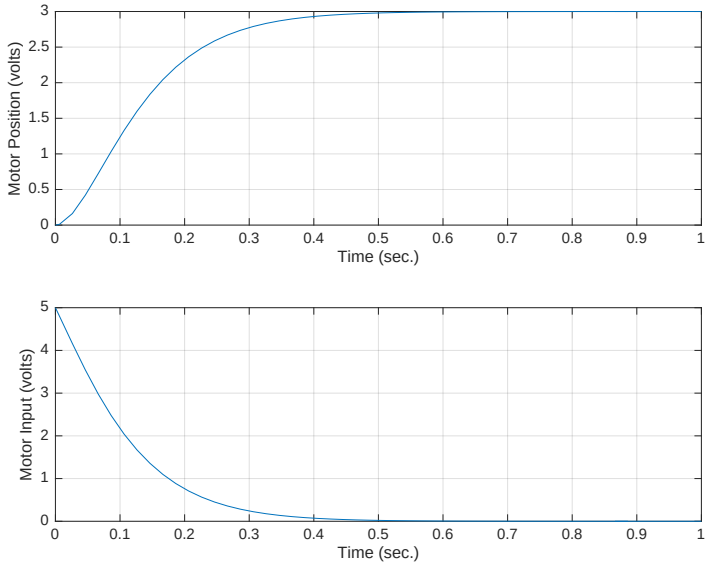


Figure 1.18 Response of control system shown in Fig. 1.17. Closed-loop poles are a double pole at -14.14. The settling time is 0.47 seconds.The maximum plant input is 5 volts.

1.8.3 Phase-Lead Compensator with Prefilter

It may seem that the phase-lead compensator designed in the previous subsection provides the fastest possible response subject to the two constraints given in (1.15). However, it is possible to make the system go faster by adding a prefilter to the control system, as shown in Fig. 1.19. Due to the cancellation between the compensator zero and the plant pole at

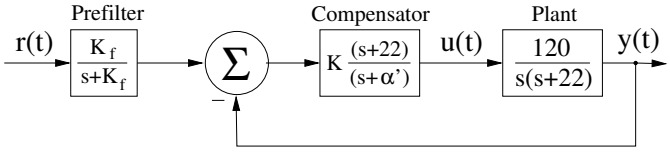


Figure 1.19 Control system with prefilter and a compensator containing a pole and a zero.

-22, the overall system in Fig. 1.19 is third order. Two of the poles of the closed-loop system are the poles of the feedback loop, and the third pole is the prefilter pole at $-K_f$. This control system is designed by choosing values for K, α', and K_f. We know that choosing $K = (\alpha')^2/(4 \cdot 120)$ results in the feedback loop having a double pole at $-\alpha'/2$. In previous designs, a double pole was required to get a critically damped response without overshoot. For the present design, however, there is a third, real pole. It turns out that it is possible for a 3rd-order system to have complex-conjugate poles and have a step response without overshoot. Such a system can be faster than a system with all real poles.

We know that if the gain is increased beyond that used to get a double pole, the resulting poles will be complex. Thus, we choose

$$K = \frac{f \cdot (\alpha')^2}{4 \cdot 120} \tag{1.20}$$

where f is a "factor" giving a 25-50% increase in the value of K (i.e. $1.25 \leq K \leq 1.5$). For any such choice of f, the real parts of the complex poles will be $-\alpha'/2$. It makes sense to choose the prefilter pole so that it is neither faster nor slower than these complex poles. Thus, K_f should be chosen to be

$$K_f = \frac{\alpha'}{2}. \tag{1.21}$$

For a fixed choice of f, a search can be conducted by varying the value of α' and using (1.20) and (1.21) to calculate K and K_f. The simulation results should be checked to see that both constraints are satisfied.

The response for one such system designed in this way are shown in Fig. 1.20. The settling time is 0.32 seconds and the constraints are satisfied. The values of K, K_f, and α' are not given so that the reader will be motivated to do the simulations as an exercise. It should be noted that the results in Fig. 1.20 are not the best possible results!

1.9 Chapter Summary

In this chapter we introduced analog and digital control problems. In analog control the measured feedback signals that the compensator receives are continuous-time signals, and the compensator is a continuous-time system. In digital control, the measured signals are sampled with an A/D converter and the compensator is a discrete-time system. The output of the compensator is passed through a D/A converter connected to the analog input to the plant. Digital control systems are sometimes called sampled data control systems to emphasize the fact that the measured feedback signals are sampled.

We also reviewed classical control theory and state-space control theory. In classical control theory the measured feedback signal is the plant output. The simplest classical compensator is a single gain, $C(s) = g$. On the other hand, state-space control systems

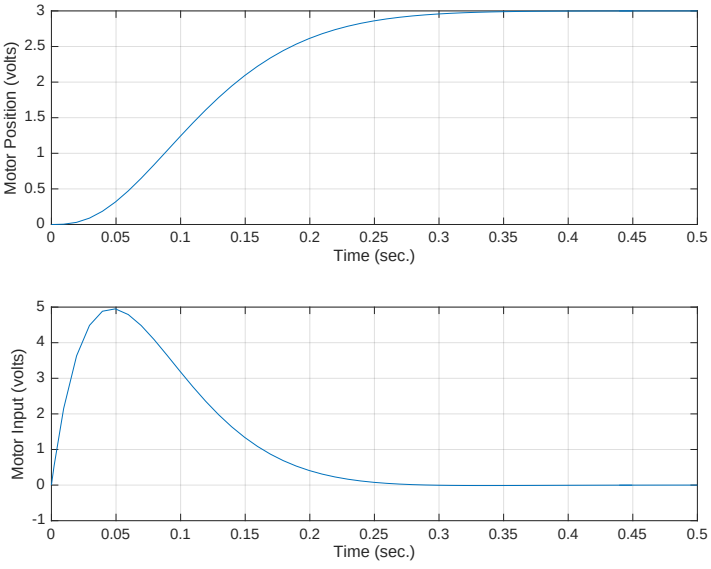


Figure 1.20 Response of control system shown in Fig. 1.19. Closed-loop poles are a pole on the negative real axis (prefilter pole) and a pair of complex-conjugate poles (poles of the feedback loop). The settling time is 0.32 seconds.The maximum plant input is 5 volts.

use n feedback signals which are the plant state variables. The simplest state-space compensator consists of n gains, each of which multiplies a state variable of the plant.

Although there are several methods for designing digital control systems, we will concentrate on one. The first step in this method is to obtain a discrete-time model for the plant. This model describes the behavior of the analog plant at sampling instants. The second step is to design a discrete-time compensator for the discrete-time plant model. The digital control system is then obtained by connecting the discrete-time compensator to the analog plant with A/D and D/A converters.

CHAPTER 2

LINEAR ALGEBRA AND MATRIX THEORY

2.1 Introduction

Vector spaces are central to the development of state-space control theory. They provide a useful generalization of familiar geometric notions regarding lines and planes to more complicated, higher dimensional, settings. In addition, vector spaces provide the tools for developing the theory of linear equations and other aspects of matrix theory. In this chapter, we start with the basic definitions of linear algebra: vector spaces, linear independence, bases, and dimension, and proceed to derive matrix theory results which are used in this book, namely: solutions to linear equations including least-squares and minimum-norm solutions, eigenvalues and eigenvectors, and the singular value decomposition.

This chapter contains a number of useful results from linear algebra. We label these results as "facts" and enclose them in a box. The reason for using this terminology is that we view the elementary theorems from linear algebra as useful facts which will be applied in subsequent work. The proofs of some of the facts are given to convey a sense of the thought processes that go into the development of linear algebra. The reader who is already familiar with linear algebra may simply wish to read the "fact boxes" as a refresher. A summary of the notation which is introduced in this chapter is given in Table 2.1.

Vector Spaces

An abstract vector space consists of two sets $\mathcal{X}$ and $\mathcal{F}$ together with certain properties that the elements of those sets must obey. The elements of the set $\mathcal{X}$ are called *vectors* and the

Symbol	Meaning	Page of First Occurrence
$\mathbf{x} \in \mathcal{X}$	$\mathbf{x}$ is a member of the set $\mathcal{X}$	25
$\forall$	for all	25
$\Longrightarrow$	implies	29
$\Longleftrightarrow$	if and only if	25
$\mathbf{A}^T$	transpose of the matrix $\mathbf{A}$	35
$\lvert\mathbf{A}\rvert$	determinant of the matrix $\mathbf{A}$, also denoted det($\mathbf{A}$)	45
$\mathbf{A}^{-1}$	inverse of the matrix $\mathbf{A}$	45
$\lVert\mathbf{x}\rVert$	norm of the vector $\mathbf{x}$	49
$R^{\perp}$	orthogonal complement of the subspace R	54
row($\mathbf{A}$)	row-space of the matrix $\mathbf{A}$	35
col($\mathbf{A}$)	column-space of the matrix $\mathbf{A}$	34
$\rho(\mathbf{A})$	rank of the matrix $\mathbf{A}$	56
$\nu(\mathbf{A})$	nullity of the matrix $\mathbf{A}$	57

Table 2.1 Summary of mathematical notation.

elements of the set $\mathcal{F}$ are called *scalars*. An addition operation must be defined for the vectors, multiplication and addition must be defined for the scalars, and a multiplication between scalars and vectors must be defined. While all these operations must obey many properties to define a vector space [42, 10], the most important properties are the following:

1. $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}\ ,\ \ \mathbf{x} + \mathbf{y} \in \mathcal{X}$.

2. $\forall \alpha \in \mathcal{F}\ ,\ \ \forall \mathbf{x} \in \mathcal{X}\ ,\ \ \alpha\mathbf{x} \in \mathcal{X}$.

In words, the first property says that a vector space is closed under addition – the sum of any two vectors from the set $\mathcal{X}$ must also be in the set. The second property says that a vector space is closed under scalar multiplication.

In this chapter, linear algebra is used to develop matrix theory, and the matrices consist of real numbers. The rows and columns of an $m \times n$ matrix are, respectively, n-tuples and m-tuples of real numbers. Thus we deal mostly with the vector space consisting of real-valued n-dimensional vectors, which we now define.

2.2 The Vector Space $\mathbf{R}^n$

Let the set $\mathcal{X} = \mathbf{R}^n$, the set of n-tuples of real numbers, where n is some finite integer. We assume that the n-tuples are arranged in a column, so that a typical element of $\mathcal{X}$ looks as follows

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \tag{2.1}$$

Let the set of scalars $\mathcal{F} = \mathbf{R}$. Then it can be shown that these two sets, together with the usual definitions of addition and multiplication, form a vector space [42, 10]. We will use lower-case boldface characters for an n-tuple (vector), and a subscripted normal font to indicate an element of an n-tuple. A subscripted boldface symbol is used to define a vector which is to be distinguished by other vectors with different subscripts. Also, we will refer to the vector space simply as $\mathbf{R}^n$. The associated set $\mathcal{F}$ of real numbers to be used as scalars will be understood.

Two vectors are said to be equal if they are equal element wise; that is

$$\mathbf{x} = \mathbf{y} \Longleftrightarrow x_i = y_i\ ,\ \ i = 1, \cdots, n. \tag{2.2}$$

In a similar way, addition and subtraction of vectors is done element wise

$$\mathbf{x} \pm \mathbf{y} = \begin{bmatrix} x_1 \pm y_1 \\ \vdots \\ x_n \pm y_n \end{bmatrix}. \tag{2.3}$$

The zero vector in $\mathbf{R}^n$ is denoted $\mathbf{0}$ and has the property that $\mathbf{x} \pm \mathbf{0} = \mathbf{x}, \forall \mathbf{x} \in \mathbf{R}^n$. Clearly we must have

$$\mathbf{0} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{2.4}$$

The multiplication of a vector by a scalar is also defined element wise as follows, where $\mathbf{x}$ and $\mathbf{y}$ are vectors in $\mathbf{R}^n$ and α is a scalar (real number):

$$\mathbf{y} = \alpha\mathbf{x} \Longleftrightarrow y_i = \alpha x_i \ , \quad i = 1, \cdots, n. \tag{2.5}$$

The above definition of multiplication of a scalar times a vector is easily shown to satisfy the following properties for any two real numbers α and β, and for any vectors $\mathbf{x}$ and $\mathbf{y}$

1. $(\alpha\beta)\mathbf{x} = \alpha(\beta\mathbf{x})$.
2. $(\alpha + \beta)\mathbf{x} = \alpha\mathbf{x} + \beta\mathbf{x}$.
3. $\alpha(\mathbf{x} + \mathbf{y}) = \alpha\mathbf{x} + \alpha\mathbf{y}$.
4. $1\mathbf{x} = \mathbf{x}$.

It is also easy to see that the two closure properties mentioned in the introduction to this chapter are satisfied.

2.3 Linear Independence, Bases, and Subspaces

Given a collection of m vectors $\mathbf{x}_1, \cdots, \mathbf{x}_m$, consider the vector $\mathbf{y}$ formed from the $\mathbf{x}$'s as follows

$$\mathbf{y} = \alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + \cdots + \alpha_m\mathbf{x}_m. \tag{2.6}$$

The vector $\mathbf{y}$ is said to be a **linear combination** of the vectors $\mathbf{x}_1, \cdots, \mathbf{x}_m$ and the numbers $\alpha_1, \cdots, \alpha_m$ are called the **expansion coefficients of the linear combination**. If all the α's are zero, the linear combination is said to be **trivial**. If at least one α is non-zero, the linear combination is said to be **non-trivial**.

EXAMPLE 2.1

Consider the following set of vectors in $\mathbf{R}^n$

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \ , \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \ , \quad \cdots \ , \quad \mathbf{e}_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

and consider an arbitrary vector $\mathbf{x}$ in $\mathbf{R}^n$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} .$$

It is easy to see that

$$\mathbf{x} = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + \cdots + x_n\mathbf{e}_n.$$

Since $\mathbf{x}$ is arbitrary, we see that any vector in $\mathbf{R}^n$ can be written as a linear combination of $\mathbf{e}_1, \cdots, \mathbf{e}_n$ and that the coefficients of the linear combination are just the elements of $\mathbf{x}$.

Definition 2.1 *Let X be a set of vectors in $\mathbf{R}^n$. The vectors in X are said to be* **linearly independent** *if the only linear combination of them which equals the zero vector is the trivial linear combination (all coefficients equal zero). If a non-trivial linear combination of the vectors equals the zero vector, the vectors in X are said to be* **linearly dependent**.

In light of the above definition, a procedure to check if a given set of vectors is linearly dependent or independent must examine the following equation

$$\alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_m \mathbf{x}_m = \mathbf{0}. \tag{2.7}$$

To prove that the vectors are linearly independent, one must show that the above equation implies that all the α's equal zero. On the other hand, if it is possible to find a set of scalars $\alpha_1, \cdots, \alpha_m$ which are not all zero such that (2.7) is satisfied, then the given vectors are linearly dependent.

EXAMPLE 2.2

Consider the vectors $\mathbf{e}_1, \cdots, \mathbf{e}_n$ introduced in Example 2.1. To test if these vectors are linearly independent or not, we use (2.7) which is written as follows

$$\alpha_1 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \cdots + \alpha_n \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

or

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The above equation says that all the coefficients equal zero, so the vectors $\mathbf{e}_1 \cdots \mathbf{e}_n$ are linearly independent.

Fact 2.1 *Given a set of vectors $\mathbf{x}_1 \cdots \mathbf{x}_m$, if one of the vectors, say $\mathbf{x}_k$, is the zero vector, then the given vectors are linearly dependent.*

PROOF. Equation (2.7) is satisfied with the following choice of coefficients

$$\alpha_i = \begin{cases} 0 & i \neq k \\ 1 & i = k \end{cases}$$

which are not all zero. Thus the vectors are linearly dependent.

Fact 2.2 *A set of vectors is linearly dependent if and only if one of the vectors in the set can be written as a linear combination of the others.*

PROOF. If the vectors are linearly dependent, we know that there exists a set of coefficients which are not all zero such that

$$\alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + \cdots + \alpha_m\mathbf{x}_m = \mathbf{0}.$$

Suppose that α_k is a non-zero coefficient. Then we can solve for $\mathbf{x}_k$ from the above equation as follows

$$\mathbf{x}_k = -\frac{1}{\alpha_k}[\alpha_1\mathbf{x}_1 + \cdots + \alpha_{k-1}\mathbf{x}_{k-1} + \alpha_{k+1}\mathbf{x}_{k+1} + \cdots + \alpha_m\mathbf{x}_m]. \tag{2.8}$$

That is, one of the vectors can be written as a linear combination of the others. On the other hand, if we know that for a given set of vectors $\{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$, one of them (say the kth vector) can be written as a linear combination of the others, then we have

$$\mathbf{x}_k = \alpha_1\mathbf{x}_1 + \cdots + \alpha_{k-1}\mathbf{x}_{k-1} + \alpha_{k+1}\mathbf{x}_{k+1} + \cdots + \alpha_m\mathbf{x}_m$$

for some set of expansion coefficients. Rearranging the previous equation as follows:

$$\alpha_1\mathbf{x}_1 + \cdots + \alpha_{k-1}\mathbf{x}_{k-1} - 1 \cdot \mathbf{x}_k + \alpha_{k+1}\mathbf{x}_{k+1} + \cdots + \alpha_m\mathbf{x}_m = \mathbf{0}$$

shows that (2.7) is satisfied with at least one nonzero coefficient (the coefficient of $\mathbf{x}_k$). Thus, from Definition 2.1 the set of vectors is linearly dependent.

■

The above fact says that the vector $\mathbf{x}_k$ is redundant as far as linear combinations go. In other words, any linear combination of the vectors $\mathbf{x}_1, \cdots, \mathbf{x}_m$ can be written as linear combination of the vectors $\mathbf{x}_1, \cdots, \mathbf{x}_{k-1}, \mathbf{x}_{k+1}, \mathbf{x}_m$ simply by replacing $\mathbf{x}_k$ by its expansion (2.8) in terms of the other vectors. Note that the above fact does *not* say that *every* vector in a linear dependent set of vectors can be written as a linear combination of the others. This is demonstrated in the following example.

EXAMPLE 2.3

Consider the following set of vectors

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{x}_4 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

These vectors are linearly dependent because (2.7) is satisfied with the coefficients $\alpha_1 = 1$, $\alpha_2 = 1$, $\alpha_3 = -1$, and $\alpha_4 = 0$. Note that either $\mathbf{x}_1$, $\mathbf{x}_2$, or $\mathbf{x}_3$ can be written in terms of the other vectors. But $\mathbf{x}_4$ cannot be written as a linear combination of the other vectors.

■

Although Definition 2.1 is the standard definition of a set of linearly independent vectors, it is sometimes more helpful to characterize such set as follows:

Fact N.1 A set of vectors is linearly independent if and only if none of the vectors in the set can be written as a linear combination of the others.

PROOF: Left as a exercise.

■

Linearly independent vectors have an important property of uniqueness of linear combinations. This property is expressed as

Fact 2.3 *Any vector that is a linear combination of linearly independent vectors has unique expansion coefficients.*

PROOF. Suppose a vector $\mathbf{y}$ can be expressed as a linear combination of a linearly independent set of vectors $\mathbf{x}_1, \cdots, \mathbf{x}_m$. Suppose that there are two sets of expansion coefficients for $\mathbf{y}$ as shown below

$$\mathbf{y} = \alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + \cdots + \alpha_m\mathbf{x}_m$$

and

$$\mathbf{y} = \beta_1\mathbf{x}_1 + \beta_2\mathbf{x}_2 + \cdots + \beta_m\mathbf{x}_m.$$

Subtracting the second equation from the first yields

$$\mathbf{0} = (\alpha_1 - \beta_1)\mathbf{x}_1 + (\alpha_2 - \beta_2)\mathbf{x}_2 + \cdots + (\alpha_m - \beta_m)\mathbf{x}_m.$$

In the above equation, the linearly independent vectors $\mathbf{x}_i$ sum to the zero vector, so the expansion coefficients must all equal zero. Thus $\alpha_i = \beta_i,\ i = 1, \cdots, m$. In other words, the two expansions of the vector $\mathbf{y}$ must have identical expansion coefficients.

■

Definition 2.2 *If a vector $\mathbf{x}_{k+1}$ can be written as a linear combination of the vectors $\mathbf{x}_1, \cdots, \mathbf{x}_k$, then we say that $\mathbf{x}_{k+1}$ is* **linearly dependent on** *$\mathbf{x}_1, \cdots, \mathbf{x}_k$. In this case the set of vectors $\mathbf{x}_1, \cdots, \mathbf{x}_{k+1}$ is linearly dependent. If x_{k+1} cannot be written as a linear combination of the vectors $\mathbf{x}_1, \cdots, \mathbf{x}_k$, then we say that $\mathbf{x}_{k+1}$ is* **linearly independent of** *$\mathbf{x}_1, \cdots, \mathbf{x}_k$.*

The elements of the set $\mathbf{R}^n$ satisfy certain properties mentioned in Section 2.2 which show that $\mathbf{R}^n$ is a vector space. There are certain subsets of $\mathbf{R}^n$ whose elements also satisfy the properties of a vector space. Thus these subsets of $\mathbf{R}^n$ are themselves vector spaces, and are called *subspaces* of $\mathbf{R}^n$. As shown in the following definition, there are only two conditions which must be checked to determine if a particular subset of $\mathbf{R}^n$ is a subspace. All of the remaining vector space conditions are satisfied automatically by the fact that the elements of a subset of $\mathbf{R}^n$ are elements of $\mathbf{R}^n$ which is a vector space.

Definition 2.3 *Let S be a nonempty subset of $\mathbf{R}^n$. Then S is a* **subspace** *of $\mathbf{R}^n$ if*

1. $\mathbf{x}, \mathbf{y} \in S \Longrightarrow \mathbf{x} + \mathbf{y} \in S$.

2. $\mathbf{x} \in S,\ \alpha \in \mathbf{R} \Longrightarrow \alpha\mathbf{x} \in S$.

In other words, a subspace is closed under addition and scalar multiplication. Note that S could equal $\mathbf{R}^n$ in which case the above conditions are identical to the two conditions stated at the beginning of this chapter for $\mathbf{R}^n$. If S is a proper subset of $\mathbf{R}^n$, then the above two conditions say that S itself is a vector space. The two conditions also imply that the zero vector is an element of every subspace. The second condition says that if $\mathbf{x} \in S$ then $-\mathbf{x} \in S$ also. The first condition says that the sum of these vectors must be in S, or $\mathbf{x} + (-\mathbf{x}) = \mathbf{0} \in S$. An example of subspaces of $\mathbf{R}^n$ is given after the next fact and definition.

Fact 2.4 *Given $\mathbf{x}_1, \cdots, \mathbf{x}_k$, let S be the set of all possible linear combinations of $\mathbf{x}_1, \cdots, \mathbf{x}_k$. Then S is a subspace of $\mathbf{R}^n$, and S is said to be the subspace* **generated** *by $\mathbf{x}_1, \cdots, \mathbf{x}_k$.*

PROOF. Consider two vectors $\mathbf{z}$ and $\mathbf{w}$ both in S. They can each be written as some linear combination of the vectors $\mathbf{x}_1 \cdots \mathbf{x}_k$

$$\mathbf{z} = \alpha_1\mathbf{x}_1 + \cdots + \alpha_k\mathbf{x}_k$$

$$\mathbf{w} = \beta_1\mathbf{x}_1 + \cdots + \beta_k\mathbf{x}_k.$$

Summing the previous two equations yields

$$\mathbf{z} + \mathbf{w} = (\alpha_1 + \beta_1)\mathbf{x}_1 + \cdots + (\alpha_k + \beta_k)\mathbf{x}_k \in S$$

and the sum of $\mathbf{z}$ and $\mathbf{w}$ is in S because it can be written as a linear combination of $\mathbf{x}_1 \cdots \mathbf{x}_k$. In a similar way, it can be shown that for any real number γ, $\gamma\mathbf{w} \in S$ since it can be written as a linear combination of the vectors $\mathbf{x}_1, \cdots, \mathbf{x}_k$ with expansion coefficients $\gamma\beta_i$.

■

Definition 2.4 *A set of vectors $\mathbf{x}_1, \cdots, \mathbf{x}_k$ is said to* **span** *a subspace S if every vector in S can be written as a linear combination of $\mathbf{x}_1, \cdots, \mathbf{x}_k$.*

EXAMPLE 2.4

Consider the vector space $\mathbf{R}^3$ in which the elements of a vector $\mathbf{x}$ are x, y, z coordinates in Euclidean space

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

The vector

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

generates a subspace of $\mathbf{R}^3$ consisting of the x axis. The vectors

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

generate a subspace of R^3 consisting of the $x - y$ plane. It is not necessary for subspaces to be "parallel" to the coordinate axes. For example, the vector

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

generates a subspace consisting of a line through the origin which is not parallel to a coordinate axis.

■

If the vectors which span a subspace are linearly dependent, then one of them can be removed from the set and the remaining vectors still span the subspace (see Fact 2.2 and the discussion following it). If the remaining set of vectors is still linearly dependent, then another vector can be removed. If all the redundant vectors were removed from the set, we would be left with a set of linearly independent vectors that would still span the subspace. Such a set is called a basis for the subspace. A formal definition of a basis is

Definition 2.5 *Let S be a subspace of $\mathbf{R}^n$ and let $B = \{\mathbf{b}_1, \cdots, \mathbf{b}_k\}$ be a set of vectors in S. Then B is said to be a* **basis for** *S if*

1. *The vectors $\mathbf{b}_1, \cdots, \mathbf{b}_k$ are linearly independent, and*
2. *The vectors $\mathbf{b}_1, \cdots, \mathbf{b}_k$ span the subspace S.*

From (2) of the above definition, we see that any vector $\mathbf{x} \in S$ can be written as a linear combination of the basis vectors

$$\mathbf{x} = \gamma_1 \mathbf{b}_1 + \cdots + \gamma_m \mathbf{b}_m.$$

From (1) of the above definition and Fact 2.3, we know that the expansion coefficients for $\mathbf{x}$ are unique.

Fact 2.5 *The set of vectors $\mathbf{e}_1, \cdots, \mathbf{e}_n$ introduced Example 2.1 is a basis for $\mathbf{R}^n$. This set of vectors is called the* **standard basis for $\mathbf{R}^n$**.

PROOF. From Example 2.1 we see that any vector in $\mathbf{R}^n$ can be written as a linear combination of $\mathbf{e}_1, \cdots, \mathbf{e}_n$. From example 2.2 we know that the vectors $\mathbf{e}_1, \cdots, \mathbf{e}_n$ are linearly independent. Thus the vectors satisfy both requirements for being a basis.

■

Fact 2.6 *Let S be a subspace with a given basis $\mathbf{b}_1 \cdots \mathbf{b}_m$. Then every basis for S must contain exactly m vectors.*

PROOF. The proof of this fact may be found in most linear algebra textbooks. For instance, see [42, 83, 84].

■

Since all bases for a subspace S have the same number of elements, we have the following

Definition 2.6 *The* **dimension** *of a subspace S is the number of elements in a basis for S.*

Using the previous fact and Fact 2.5, we see that **the dimension of $\mathbf{R}^n$ is n.** Thus the name "n-dimensional Euclidean space" is appropriate for $\mathbf{R}^n$. A particular set of basis vectors called the standard basis was introduced in Fact 2.5. However the standard basis is not the only basis for $\mathbf{R}^n$. It turns out that *any* set of n linearly independent vectors is a basis for $\mathbf{R}^n$. This fact is stated next for the general case of m-dimensional subspaces.

Fact 2.7 *Let S be an m-dimensional subspace of $\mathbf{R}^n$. Then no set of linearly independent vectors in S has more than m elements. Furthermore, any set of m linearly independent vectors in S is a basis for S.*

PROOF. The first part of this fact is a corollary to Fact 2.6. To prove the second part of this fact, let $\mathbf{b}_1, \cdots, \mathbf{b}_m$ be linearly independent vectors in S and consider an arbitrary vector $\mathbf{x} \in S$. Then the vectors $\mathbf{b}_1, \cdots, \mathbf{b}_m, \mathbf{x}$ must be linearly dependent by the first part of this fact, since there are more than m vectors, and S has dimension m. Thus there exist coefficients $\alpha_0, \alpha_1, \cdots, \alpha_m$ not all zero such that

$$\alpha_0\mathbf{x} + \alpha_1\mathbf{b}_1 + \cdots + \alpha_m\mathbf{b}_m = \mathbf{0}.$$

We claim that $\alpha_0 \neq 0$, since if $\alpha_0 = 0$ then the above equation reduces to

$$\alpha_1\mathbf{b}_1 + \cdots + \alpha_m\mathbf{b}_m = \mathbf{0}$$

and since $\mathbf{b}_1, \cdots, \mathbf{b}_m$ are linearly independent, this would imply that $\alpha_1 = \alpha_2 = \cdots = \alpha_m = 0$. Thus all the α's (including α_0) would be zero, which we know is not true. So we must have $\alpha_0 \neq 0$. Then we can solve for $\mathbf{x}$ as follows

$$\mathbf{x} = -\frac{1}{\alpha_0}[\alpha_1\mathbf{b}_1 + \cdots + \alpha_m\mathbf{b}_m].$$

Since $\mathbf{x}$ is arbitrary, we see that any vector in S can be written as a linear combination of the vectors $\mathbf{b}_1, \cdots, \mathbf{b}_m$, i.e. these vectors span S. Since these vectors are also linearly independent, they satisfy Definition 2.5 and they are a basis for S.

■

Using the above Fact, we know that the vectors in Example 2.3 must be linearly dependent, because the number of vectors is greater than the dimension of the vector space. The following fact says that a set of linearly independent vectors in a subspace can always be extended to form a basis for the subspace.

Fact 2.8 *Let S be a m-dimensional subspace of $\mathbf{R}^n$ and let $\mathbf{b}_1, \cdots \mathbf{b}_k$ be linearly independent vectors in S for some $k \leq m$. Then additional vectors $\mathbf{b}_{k+1}, \mathbf{b}_{k+2}, \cdots, \mathbf{b}_m$ can be found such that $B = \{\mathbf{b}_1, \cdots, \mathbf{b}_m\}$ is a basis for S.*

PROOF. If $k = m$, then we have m linearly independent vectors in an m-dimensional subspace S, and by Fact 2.7, these vectors are a basis for S. Thus we don't need to add

any vectors to the set to get a basis. Now assume that $k < m$. We claim that there exists a vector $\mathbf{b}_{k+1} \in S$ such that the vectors $\mathbf{b}_1, \cdots, \mathbf{b}_{k+1}$ are linearly independent. If we could not choose such a linearly independent vector, then all vectors $\mathbf{x} \in S$ would have the property that the vectors $\mathbf{x}, \mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_k$ are linearly dependent, and so $\mathbf{x}$ could be written as a linear combination of $\mathbf{b}_1, \cdots, \mathbf{b}_k$ (see the proof of Fact 2.7). Then $\mathbf{b}_1, \cdots, \mathbf{b}_k$ would be k linearly independent vectors which span S, with $k < m$. In other words, these vectors would be a basis for S consisting of fewer than m vectors. This is impossible by Fact 2.6. Thus our claim that we can find a vector $\mathbf{b}_{k+1}$ such that the vectors $\mathbf{b}_1 \cdots, \mathbf{b}_{k+1}$ are linearly independent is indeed true. If $k + 1 < m$, the same argument shows that we can find a vector $\mathbf{b}_{k+2}$ such that the vectors $\mathbf{b}_1, \cdots \mathbf{b}_{k+2}$ are linearly independent. This argument can be repeated until finally we find a vector $\mathbf{b}_m$ such that $\mathbf{b}_1, \cdots, \mathbf{b}_m$ are linearly independent. By Fact 2.7, these vectors are a basis for S, and they also contain the given vectors $\mathbf{b}_1, \cdots, \mathbf{b}_k$.

■

Fact 2.9 *Let S be an m-dimensional subspace of $\mathbf{R}^n$ and let $B = \{\mathbf{b}_1, \cdots, \mathbf{b}_p\}$, $p \geq m$ be a set of vectors which spans S. Then m vectors can be chosen from B to be a basis for S.*

PROOF. Let $\bar{\mathbf{b}}_1 = \mathbf{b}_i$ where $\mathbf{b}_i \in B$ is the non-zero vector from B with the smallest index i. Then let $\bar{\mathbf{b}}_2 = \mathbf{b}_j$ where $\mathbf{b}_j \in B$ with the smallest index j such that the vectors $\bar{\mathbf{b}}_1$, $\mathbf{b}_j$ are linearly independent. Continue in this way to extract a set of linearly independent vectors $\bar{B} = \{\bar{\mathbf{b}}_1, \cdots, \bar{\mathbf{b}}_q\}$ from the given set of vectors B. We will now show that $\bar{B}$ is a basis for S and that $q = m$. To show this, let $\mathbf{b}_k$ be any vector in B that was *not* selected to be in $\bar{B}$. Then the vectors $\mathbf{b}_k, \bar{\mathbf{b}}_1, \cdots, \bar{\mathbf{b}}_q$ must be linearly dependent. Since $\bar{\mathbf{b}}_1, \cdots, \bar{\mathbf{b}}_q$ are linearly independent, then $\mathbf{b}_k$ can be written as a linear combination of them (see the proof of Fact 2.7). This is true for all vectors not selected to be in $\bar{B}$. Thus each $\mathbf{b}_1, \cdots, \mathbf{b}_p$ can be written as a linear combination of vectors in $\bar{B}$, so $\bar{B}$ spans S. Since the vectors in $\bar{B}$ are also linearly independent, $\bar{B}$ is a basis for S. By Fact 2.6, q must equal m, and the above procedure has extracted a basis for S.

■

The previous two facts show that a basis is a maximal independent set of vectors in a given subspace, and at the same time, a minimal spanning set of vectors. A basis cannot be made larger without losing independence, and it cannot be made smaller and still span the subspace.

Fact 2.10 *The set S of all linear combinations of m linearly independent vectors $\mathbf{b}_1, \cdots, \mathbf{b}_m$ is an m-dimensional subspace of $\mathbf{R}^n$.*

PROOF. From Fact 2.4, S is a subspace spanned by the vectors $\mathbf{b}_1, \cdots, \mathbf{b}_m$. Since these vectors are also linearly independent, they form a basis for S (Definition 2.5). Since there are m vectors in this basis, the dimension of S must be m (Fact 2.6).

■

2.4 Matrices

The elements of the vector space $\mathbf{R}^n$ are columns of n real numbers. While column vectors play an important role in state-space system theory, these vectors are often transformed by matrices. In addition, the theory of simultaneous linear equations, which is central to many results in state-space system theory, is developed in terms of subspaces defined by the rows or columns of a matrix. In this section, we give the basic definitions and operations of matrices.

Definition 2.7 *An $m \times n$* **matrix** *is a rectangular array of numbers having m rows and n columns. The numbers m and n are called the* dimensions *of the matrix.*

In this book, matrices are represented by upper-case boldface symbols. The elements of the matrix are written with double subscripts, the first of which indicates the row position of that element, and the second, the column position. For example, a 3×4 matrix $\mathbf{A}$ has the following form.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}.$$

The number a_{ij} is called the (i, j)-*element* of $\mathbf{A}$. An $n \times 1$ matrix is a vector (sometimes called a *column vector*). A $1 \times n$ matrix is called a *row vector*.

Note that each column of an $m \times n$ matrix $\mathbf{A}$ is an element of $\mathbf{R}^m$. For example, the first column of $\mathbf{A}$, which we denote $\mathbf{a}_1$ is

$$\mathbf{a}_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix}.$$

Thus an $m \times n$ matrix may be thought of as a collection of vectors from $\mathbf{R}^m$ indexed according to their column position as follows

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots \mathbf{a}_n \end{bmatrix}.$$

Definition 2.8 *Given an $m \times n$ matrix $\mathbf{A}$, the vector space formed by all linear combinations of the columns of $\mathbf{A}$ is called the* **column space of $\mathbf{A}$***, and is denoted col($\mathbf{A}$). (In other treatments of matrices, the column space is sometimes called the range or image of $\mathbf{A}$).*

Given a matrix $\mathbf{A}$, we can define a new matrix $\mathbf{A}^T$ by setting the rows of $\mathbf{A}^T$ equal to the columns of $\mathbf{A}$. This leads to the following definition.

Definition 2.9 *Given an $m \times n$ matrix*

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

1. $(\mathbf{A}^T)^T = \mathbf{A}$.
2. $(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$.
3. $(\alpha\mathbf{A})^T = \alpha\mathbf{A}^T$.
4. $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$.

Table 2.2 Some identities involving the transpose operation.

the $n \times m$ matrix obtained by interchanging the rows and columns of $\mathbf{A}$ is called the **transpose of the matrix $\mathbf{A}$** *and is denoted by $\mathbf{A}^T$, where*

$$\mathbf{A}^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \cdots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}$$

EXAMPLE 2.5

The transpose of a column vector

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

is a row vector

$$\mathbf{a}^T = [a_1\ a_2\ \cdots\ a_n].$$

■

There are several useful identities involving the transpose operation which are shown in Table 2.2.

Previously we viewed a matrix $\mathbf{A}$ as a collection of column vectors. We can also think of an $m \times n$ matrix $\mathbf{A}$ as a collection of row vectors as follows.

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_m^T \end{bmatrix}$$

where

$$\mathbf{a}_i^T = [a_{i1}\ a_{i2}\ \cdots\ a_{in}]\ ,\ \ i = 1, \cdots, m.$$

Another useful subspace in matrix theory is defined by the rows of a matrix.

Definition 2.10 *Given an $m \times n$ matrix $\mathbf{A}$, the vector space formed by all linear combinations of the rows of $\mathbf{A}$ is called the* **row space of $\mathbf{A}$**, *and is denoted row($\mathbf{A}$).*

1. $(\alpha\beta)\mathbf{A} = \alpha(\beta\mathbf{A})$.
2. $(\alpha + \beta)\mathbf{A} = \alpha\mathbf{A} + \beta\mathbf{A}$.
3. $\alpha(\mathbf{A} + \mathbf{B}) = \alpha\mathbf{A} + \alpha\mathbf{B}$.
4. $1 \cdot \mathbf{A} = \mathbf{A}$.

Table 2.3 Properties of scalar-matrix multiplication.

2.4.1 Operations with Matrices

Definition 2.11 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be* $m \times n$ *matrices. The* **sum of** $\mathbf{A}$ **and** $\mathbf{B}$ *is the matrix* $\mathbf{C}$ *whose elements are given by*

$$c_{ij} = a_{ij} + b_{ij} \ , \ \ i = 1, \cdots, m \ , \ \ j = 1, \cdots, n.$$

We write

$$\mathbf{C} = \mathbf{A} + \mathbf{B}.$$

Note that the sum of two matrices is defined only when they have the same dimensions, and that the addition of matrices is done element wise. Since the addition of real numbers is commutative and associative, it is easy to show that the addition of matrices is also commutative and associative so that the following fact is true.

Fact 2.11

1. $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$.
2. $(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C})$.

Definition 2.12 *Let* $\mathbf{A}$ *be an* $m \times n$ *matrix and let* α *be a real number. The product of* α *and* $\mathbf{A}$*, written* $\alpha\mathbf{A}$ *or* $\mathbf{A}\alpha$ *is the* $m \times n$ *matrix whose elements are given by*

$$\alpha a_{ij} \ , \ \ i = 1, \cdots, m \ , \ \ j = 1, \cdots, n.$$

This definition says that the product of a scalar and a matrix is done element wise. A list of the properties of scalar-matrix multiplication is given in Table 2.3. Note that the properties are also true when the scalars appear to the right of the matrices.

Matrices can be multiplied by scalars as shown above. It is also possible to multiply two matrices together according to the following definition

Definition 2.13 *Let* $\mathbf{A}$ *be a* $p \times m$ *matrix and* $\mathbf{B}$ *be an* $m \times n$ *matrix. The* **product of** $\mathbf{A}$ **and** $\mathbf{B}$ **is the** $p \times n$ **matrix** $\mathbf{C}$ *whose elements are given by*

$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj} \ , \ \ i = 1, \cdots, p; \ j = 1, \cdots, n.$$

We write

$$\mathbf{C} = \mathbf{AB}.$$

1. $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$.
2. $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$.
3. $(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$.
4. $\alpha(\mathbf{AB}) = (\alpha\mathbf{A})\mathbf{B} = \mathbf{A}(\alpha\mathbf{B})$.
5. $\mathbf{AB} \neq \mathbf{BA}$, in general.

Table 2.4 Properties of matrix multiplication.

An example of matrix multiplication is given on page 39 after several interpretations of matrix multiplication are given. Three important aspects of matrix multiplication can be inferred from the definition, and are listed below. Properties of matrix multiplication are given in Table 2.4.

1. A product of two matrices is only defined when the number of columns of the first factor equals the number of rows of the second factor.

2. The product matrix has the same number of rows as the first factor and the same number of columns as the second factor, as shown in the following equation

$$\mathbf{C}_{p\times n} = \mathbf{A}_{p\times m}\mathbf{B}_{m\times n}.$$

3. Matrix multiplication does not commute, in general. The matrices $\mathbf{AB}$ and $\mathbf{BA}$ may not have the same dimensions. Even if the dimensions are the same, it is not necessarily true that $\mathbf{AB} = \mathbf{BA}$.

We now investigate several different expressions of matrix multiplication. To do so, it is useful to first define two types of vector multiplication, and two types of matrix-vector multiplication.

Definition 2.14 *Given two vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *in* $\mathbf{R}^n$*, their* **inner product** γ *is the real number given by*

$$\gamma = \mathbf{x}^T\mathbf{y} = \mathbf{y}^T\mathbf{x} = \sum_{i=1}^{n} x_i y_i.$$

Definition 2.15 *Given two vectors* $\mathbf{x} \in \mathbf{R}^m$ *and* $\mathbf{y} \in \mathbf{R}^n$*, their* **outer product** *results in a matrix. The outer product of* $\mathbf{x}$ *with* $\mathbf{y}$ *gives the* $m \times n$ *matrix* $\mathbf{A}$ *defined by*

$$\mathbf{A} = \mathbf{x}\mathbf{y}^T\ , \quad a_{ij} = x_i y_j.$$

The outer product of $\mathbf{y}$ *with* $\mathbf{x}$ *gives the* $n \times m$ *matrix* $\mathbf{B}$ *defined by*

$$\mathbf{B} = \mathbf{y}\mathbf{x}^T\ , \quad b_{ij} = y_i x_j.$$

By the rules of matrix transposition (see Fact 2.2), $\mathbf{B}^T = (\mathbf{y}\mathbf{x}^T)^T = \mathbf{x}\mathbf{y}^T = \mathbf{A}$ *so that* $\mathbf{A} = \mathbf{B}^T$.

Before presenting different interpretations of matrix multiplication, we first give two interpretations of matrix-vector multiplication. Since a row or column vector can also be

thought of as a matrix with a single row or column, respectively, the next two facts follow from the definition (2.13) of matrix multiplication.

Fact 2.12 *The product of a matrix* $\mathbf{A}$ *and a column vector* $\mathbf{x}$ *can be written as a linear combination of the columns of* $\mathbf{A}$ *in which the expansion coefficients are the elements of the vector* $\mathbf{x}$*, i.e.*

$$\mathbf{A}\mathbf{x} = [\,\mathbf{a}_1 \quad \cdots \quad \mathbf{a}_n\,] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 + \cdots + \mathbf{a}_n x_n.$$

A similar fact is true about the product of a row vector and a matrix.

Fact 2.13 *The product of a row vector* $\mathbf{y}^T$ *and a matrix* $\mathbf{A}$ *can be written as a a linear combination of the rows of* $\mathbf{A}$ *in which the expansion coefficients are the elements of the vector* $\mathbf{y}$*, i.e.*

$$\mathbf{y}^T\mathbf{A} = [\,y_1 \quad \cdots \quad y_m\,] \begin{bmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_m^T \end{bmatrix} = y_1\mathbf{a}_1^T + y_2\mathbf{a}_2^T + \cdots + y_m\mathbf{a}_m^T.$$

We can now state several different interpretations of matrix multiplication.

Elements of a Product Matrix as Inner Products If a $p \times m$ matrix $\mathbf{A}$ is written in terms of its rows, and an $m \times n$ matrix $\mathbf{B}$ is written in terms of its columns

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_p^T \end{bmatrix}, \quad \mathbf{B} = [\,\mathbf{b}_1 \quad \cdots \quad \mathbf{b}_n\,]$$

then the definition of matrix multiplication (Definition 2.13) gives the following

Fact 2.14 *If the matrix* $\mathbf{C} = \mathbf{A}\mathbf{B}$*, then the* (i, j)*-element of* $\mathbf{C}$ *is the inner product of the* i*th row of* $\mathbf{A}$ *with the* j*th column of* $\mathbf{B}$*, i.e.*

$$c_{ij} = \mathbf{a}_i^T\mathbf{b}_j.$$

A Product Matrix as a Sum of Outer Products A less obvious interpretation of matrix multiplication than the one given in Fact 2.14 can also be obtained from Definition 2.13. To get this interpretation, we write $\mathbf{A}$ in terms of its columns and $\mathbf{B}$ in terms of its rows as follows

$$\mathbf{A} = [\,\mathbf{a}_1 \quad \cdots \quad \mathbf{a}_m\,], \quad \mathbf{B} = \begin{bmatrix} \mathbf{b}_1^T \\ \vdots \\ \mathbf{b}_m^T \end{bmatrix}.$$

Then we have the following

Fact 2.15 *If the matrix* $\mathbf{C} = \mathbf{AB}$*, then* $\mathbf{C}$ *can be written as the sum of outer products of columns of* $\mathbf{A}$ *with rows of* $\mathbf{B}$*, i.e.*

$$\mathbf{C} = \mathbf{a}_1\mathbf{b}_1^T + \mathbf{a}_2\mathbf{b}_2^T + \cdots + \mathbf{a}_m\mathbf{b}_m^T. \tag{2.9}$$

Matrix Multiplication as a Weighted Sum of Columns or Rows There are two other interpretations of matrix multiplication that are related to the one just given. We can write the matrix $\mathbf{B}$ in terms of its columns, and then multiply $\mathbf{A}$ by each column of $\mathbf{B}$ as shown below.

Fact 2.16 *The matrix product* $\mathbf{AB}$ *can be written column-wise as* $\mathbf{A}$ *times the columns of* $\mathbf{B}$

$$\mathbf{AB} = \mathbf{A}\begin{bmatrix}\mathbf{b}_1 & \cdots & \mathbf{b}_n\end{bmatrix} = \begin{bmatrix}\mathbf{Ab}_1 & \mathbf{Ab}_2 & \cdots & \mathbf{Ab}_n\end{bmatrix}.$$

From Fact 2.12 the matrix-vector product $\mathbf{Ab}_i$ is a linear combination of columns of $\mathbf{A}$. Thus each column of the product $\mathbf{AB}$ is a linear combination of the columns of $\mathbf{A}$.

A different way of expressing matrix multiplication is to write the matrix $\mathbf{A}$ in terms of its rows, and then multiply each row of $\mathbf{A}$ by the matrix $\mathbf{B}$.

Fact 2.17 *The matrix product* $\mathbf{AB}$ *can be written row-wise as rows of* $\mathbf{A}$ *times* $\mathbf{B}$

$$\mathbf{AB} = \begin{bmatrix}\mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_p^T\end{bmatrix}\mathbf{B} = \begin{bmatrix}\mathbf{a}_1^T\mathbf{B} \\ \vdots \\ \mathbf{a}_p^T\mathbf{B}\end{bmatrix}.$$

From Fact 2.13, each row of the product $\mathbf{AB}$ is a linear combination of the rows of $\mathbf{B}$.

EXAMPLE 2.6

This example illustrates the different interpretations of matrix multiplication mentioned above. Let

$$\mathbf{A} = \begin{bmatrix}1 & 2 & 3 \\ 4 & 5 & 6\end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix}1 & 0 \\ 1 & 1 \\ 0 & 1\end{bmatrix}.$$

Applying Fact 2.14 we compute the elements of the product matrix $\mathbf{C} = \mathbf{AB}$ using inner products

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = 3 \quad \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 5$$

$$\begin{bmatrix} 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = 9 \quad \begin{bmatrix} 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 11$$

resulting in

$$\mathbf{C} = \begin{bmatrix} 3 & 5 \\ 9 & 11 \end{bmatrix}.$$

The same matrix product $\mathbf{AB}$ can be computed in a different way using Fact 2.15 by computing the following outer products

$$\begin{bmatrix} 1 \\ 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 4 & 0 \end{bmatrix} \quad \begin{bmatrix} 2 \\ 5 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 5 & 5 \end{bmatrix}$$

$$\begin{bmatrix} 3 \\ 6 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 3 \\ 0 & 6 \end{bmatrix}$$

The product matrix is obtained by adding the outer products computed above

$$\begin{bmatrix} 1 & 0 \\ 4 & 0 \end{bmatrix} + \begin{bmatrix} 2 & 2 \\ 5 & 5 \end{bmatrix} + \begin{bmatrix} 0 & 3 \\ 0 & 6 \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 9 & 11 \end{bmatrix}.$$

We can also compute the matrix product $\mathbf{AB}$ column wise as in Fact 2.16. The columns of $\mathbf{C}$ are computed as follows

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \end{bmatrix}, \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 11 \end{bmatrix}$$

Notice that the calculations for this method are the same as the method of inner products except that here the rows of $\mathbf{A}$ are grouped together.

Finally, we can compute the matrix product $\mathbf{AB}$ row wise as in Fact 2.17. The rows of $\mathbf{C}$ are computed as follows

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 5 \end{bmatrix}, \quad \begin{bmatrix} 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 9 & 11 \end{bmatrix}.$$

Notice that the calculations for this method are the same as the method of inner products except that here the columns of $\mathbf{B}$ are grouped together.

■

We now define some special matrices which are useful in the sequel.

Definition 2.16 *An $n \times n$ matrix $\mathbf{A}$ is said to be* **lower triangular** *if all elements above the main diagonal are zero. A lower triangular matrix has the following form*

$$\mathbf{A} = \begin{bmatrix} * & 0 & \cdots & 0 \\ * & * & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & * \end{bmatrix}.$$

An **upper triangular** matrix is defined in a similar way – all elements below the main diagonal are zero. A matrix which is both upper triangular and lower triangular has nonzero entries only on the main diagonal. Such a matrix has the following definition.

Definition 2.17 *A matrix is said to be* **diagonal** *if all elements not on the main diagonal are zero as shown below*

$$\mathbf{A} = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_n \end{bmatrix}.$$

We write

$$\mathbf{A} = \mathit{diag}(a_1, a_2, \cdots, a_n).$$

Definition 2.18 *The $n \times n$ diagonal matrix $\mathit{diag}(1, 1, \cdots, 1)$ is called the $n \times n$* identity matrix *and is denoted $\mathbf{I}_n$. The subscript indicating the dimension will often be omitted.*

2.4.2 Operations with Partitioned Matrices*

The matrix operations introduced in the previous section were defined in terms of the addition and multiplication of real numbers, i.e. the elements of the matrices. Suppose we now consider a matrix to consist of **partitioned submatrices**. A **submatrix** is a rectangular array of numbers within a given matrix. For instance, a 4×4 matrix $\mathbf{A}$ can be partitioned into 4 submatrices as follows

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \tag{2.10}$$

where

$$\mathbf{A}_{11} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \mathbf{A}_{12} = \begin{bmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \end{bmatrix}$$

$$\mathbf{A}_{21} = \begin{bmatrix} a_{31} & a_{32} \\ a_{41} & a_{42} \end{bmatrix}, \quad \mathbf{A}_{22} = \begin{bmatrix} a_{33} & a_{34} \\ a_{43} & a_{44} \end{bmatrix}.$$

It is clear that a matrix can be partitioned into submatrices in several different ways. For instance, the partition shown in (2.10) is also true with the following definitions

$$\mathbf{A}_{11} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad \mathbf{A}_{12} = \begin{bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{bmatrix}$$

$$\mathbf{A}_{21} = \begin{bmatrix} a_{41} & a_{42} & a_{43} \end{bmatrix}, \quad \mathbf{A}_{22} = a_{44}.$$

This example shows that when dealing with partitioned submatrices, it is important to know the dimensions of each submatrix.

We now examine the possibility of performing matrix operations with partitioned matrices **as if the submatrices were the "elements" of the matrix**. Of course when the "elements" (submatrices) are added or multiplied, they must be combined according to the rules of matrix addition and multiplication. The results for operations with partitioned matrices are summarized in Fact 2.18 on page 42.

Fact 2.18 *Let the matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *be partitioned as follows*

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \cdots & \mathbf{A}_{1l} \\ \vdots & \vdots & \vdots \\ \mathbf{A}_{k1} & \cdots & \mathbf{A}_{kl} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} & \cdots & \mathbf{B}_{1n} \\ \vdots & \vdots & \vdots \\ \mathbf{B}_{m1} & \cdots & \mathbf{B}_{mn} \end{bmatrix}$$

where $\mathbf{A}_{ij}$ *is a submatrix of* $\mathbf{A}$ *with dimensions* $\alpha_i \times \delta_j$ *and* $\mathbf{B}_{ij}$ *is a submatrix of* $\mathbf{B}$ *with dimensions* $\beta_i \times \gamma_j$*. Then the addition, multiplication, and transpose operations can be performed as follows:*

1. *If* $k = m$*,* $\alpha_i = \beta_i$*, and* $\delta_j = \gamma_j$*, then*

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} \mathbf{C}_{11} & \cdots & \mathbf{C}_{1l} \\ \vdots & \vdots & \vdots \\ \mathbf{C}_{k1} & \cdots & \mathbf{C}_{kl} \end{bmatrix}$$

where

$$\mathbf{C}_{ij} = \mathbf{A}_{ij} + \mathbf{B}_{ij}.$$

2. *If* $l = m$ *and* $\delta_i = \beta_i$ *then*

$$\mathbf{AB} = \begin{bmatrix} \mathbf{C}_{11} & \cdots & \mathbf{C}_{1n} \\ \vdots & \vdots & \vdots \\ \mathbf{C}_{k1} & \cdots & \mathbf{C}_{kn} \end{bmatrix}$$

where

$$\mathbf{C}_{ij} = \sum_{q=1}^{l} \mathbf{A}_{iq}\mathbf{B}_{qj}.$$

3.

$$\mathbf{A}^T = \begin{bmatrix} \mathbf{A}_{11}^T & \cdots & \mathbf{A}_{k1}^T \\ \vdots & \vdots & \vdots \\ \mathbf{A}_{1l}^T & \cdots & \mathbf{A}_{kl}^T \end{bmatrix}.$$

The first two items in Fact 2.18 require that the dimensions of the submatrices of $\mathbf{A}$ and $\mathbf{B}$ are such that addition and multiplication of the submatrices can be performed. Two matrices whose submatrices satisfy this property are said to be partitioned **conformally**. The addition of conformally partitioned matrices is accomplished by simply adding the corresponding submatrices. Multiplication is accomplished by taking an "inner product" with the i-th block row and the j-th block column of the matrices $\mathbf{A}$ and $\mathbf{B}$, respectively. The **i-th block row of $\mathbf{A}$** is shown in the rectangle below

$$\begin{bmatrix} \mathbf{A}_{11} & \cdots & \mathbf{A}_{1l} \\ \vdots & \vdots & \vdots \\ \boxed{\mathbf{A}_{i1} \quad \cdots \quad \mathbf{A}_{il}} \\ \vdots & \vdots & \vdots \\ \mathbf{A}_{k1} & \cdots & \mathbf{A}_{kl} \end{bmatrix}$$

and the **j-th block column of $\mathbf{B}$** is shown in the rectangle below

$$\begin{bmatrix} \mathbf{B}_{11} \cdots \boxed{\mathbf{B}_{1j}} \cdots \mathbf{B}_{1n} \\ \vdots \quad \vdots \quad \boxed{\vdots} \quad \vdots \quad \vdots \\ \mathbf{B}_{m1} \cdots \boxed{\mathbf{B}_{mj}} \cdots \mathbf{B}_{mn} \end{bmatrix}.$$

The "inner product" of the i-th block row of $\mathbf{A}$ and the j-th block column of $\mathbf{B}$ is defined by the summation shown in item 2 of Fact 2.18. Compare this summation with the one given in Definition 2.13 for ordinary matrix multiplication.

EXAMPLE 2.7

Consider the following block matrices

$$\mathbf{A} = \left[\begin{array}{cc|c} 1 & 2 & 3 \\ 4 & 5 & 6 \\ \hline 7 & 8 & 9 \end{array}\right] = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix},$$

and

$$\mathbf{B} = \left[\begin{array}{c|c} 1 & 0 \\ 1 & 1 \\ \hline 0 & 1 \end{array}\right] = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix}.$$

These matrices are each 2×2 block matrices, and so the product matrix $\mathbf{C} = \mathbf{AB}$ can be computed by 4 block inner products as shown below.

$$\mathbf{C}_{11} = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 3 \\ 6 \end{bmatrix} [0] = \begin{bmatrix} 3 \\ 9 \end{bmatrix}$$

$$\mathbf{C}_{12} = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 3 \\ 6 \end{bmatrix} [1] = \begin{bmatrix} 5 \\ 11 \end{bmatrix}$$

$$\mathbf{C}_{21} = [\,7 \quad 8\,] \begin{bmatrix} 1 \\ 1 \end{bmatrix} + [\,9\,][\,0\,] = [\,15\,]$$

$$\mathbf{C}_{22} = [\,7 \quad 8\,] \begin{bmatrix} 0 \\ 1 \end{bmatrix} + [\,9\,][\,1\,] = [\,17\,]$$

The resulting product matrix is then

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 9 & 11 \\ 15 & 17 \end{bmatrix}.$$

■

We also note that the definition of multiplication for partitioned matrices given in Fact 2.18 can also be used to define the multiplication of a partitioned matrix and a partitioned vector simply by having the matrix $\mathbf{B}$ have one column. To more clearly illustrate partitioned matrix-vector multiplication, we use different notation for the partitioned vector as shown below. Let

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \cdots & \mathbf{A}_{1l} \\ \vdots & \vdots & \vdots \\ \mathbf{A}_{k1} & \cdots & \mathbf{A}_{kl} \end{bmatrix}$$

where $\mathbf{A}_{ij}$ is a submatrix of $\mathbf{A}$ with dimensions $\alpha_i \times \delta_j$ and let

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_l \end{bmatrix}$$

where $\mathbf{x}_j$ is a subvector of $\mathbf{x}$ with dimensions $\delta_j \times 1$. Then the matrix-vector product can be written as

$$\mathbf{A}\mathbf{x} = \sum_{j=1}^{l} \mathbf{A}_{ij}\mathbf{x}_j.$$

To illustrate the transpose operation on a partitioned matrix, we show a matrix $\mathbf{A}$ below with its i-th block row enclosed in a rectangle. When $\mathbf{A}$ is transposed, this block row becomes the i-th block column of $\mathbf{A}^T$ and the individual submatrices are each transposed as follows

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \cdots & \mathbf{A}_{1l} \\ \vdots & \vdots & \vdots \\ \boxed{\begin{matrix} \mathbf{A}_{i1} & \cdots & \mathbf{A}_{il} \end{matrix}} \\ \vdots & \vdots & \vdots \\ \mathbf{A}_{k1} & \cdots & \mathbf{A}_{kl} \end{bmatrix}, \quad \mathbf{A}^T = \begin{bmatrix} \mathbf{A}_{11}^T \cdots & \boxed{\mathbf{A}_{i1}^T} & \cdots \mathbf{A}_{k1}^T \\ \vdots \quad \vdots & \vdots & \vdots \quad \vdots \quad \vdots \\ \mathbf{A}_{1l}^T \cdots & \boxed{\mathbf{A}_{il}^T} & \cdots \mathbf{A}_{kl}^T \end{bmatrix}.$$

2.4.3 Determinants and Matrix Inverses

Given an $n \times n$ matrix $\mathbf{A}$, the $n \times n$ matrix $\mathbf{B}$ is called the *inverse of* $\mathbf{A}$ if it satisfies

$$\mathbf{A}\mathbf{B} = \mathbf{B}\mathbf{A} = \mathbf{I}. \tag{2.11}$$

1. If $\mathbf{A}$ and $\mathbf{B}$ are both $n \times n$, then $|\mathbf{AB}| = |\mathbf{A}||\mathbf{B}|$.
2. $|\mathbf{A}| = |\mathbf{A}^T|$.
3. If $\mathbf{A}$ is nonsingular then $|\mathbf{A}^{-1}| = 1/|\mathbf{A}|$.
4. Multiplying all elements of any one row (or column) of a matrix $\mathbf{A}$ by a scalar α yields a matrix whose determinant is $\alpha|\mathbf{A}|$.
5. Any multiple of a row (column) can be added to any other row (column) without changing the value of the determinant.
6. The determinant of a triangular matrix equals the product of its diagonal elements.
7. The determinant of a block triangular matrix equals the product of the determinants of the matrices on the main diagonal.

Table 2.5 Some properties of determinants.

The inverse of the matrix $\mathbf{A}$ is denoted $\mathbf{A}^{-1}$. Note that only square matrices can have an inverse, because the product matrices in (2.11) ($\mathbf{AB}$ and $\mathbf{BA}$) will have the same dimensions only when $\mathbf{A}$ and $\mathbf{B}$ are square. Not all square matrices possess an inverse, however.

One way to test whether or not a given matrix $\mathbf{A}$ has an inverse is to compute the *determinant* of $\mathbf{A}$, denoted $\det(\mathbf{A})$ or $|\mathbf{A}|$. The determinant of a 2×2 matrix is given by the following formula

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc. \tag{2.12}$$

The determinants of larger matrices may be computed by combining determinants of smaller submatrices [10, 84, 83]. In this book we will not be concerned with the hand calculation of determinants.

The formula for the inverse of a 2×2 matrix is

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}. \tag{2.13}$$

This formula can be verified by substituting it into (2.11). Note that the determinant of the matrix appears in the denominator of the expression for the inverse. This means that the inverse of a 2×2 matrix is defined only when its determinant is nonzero. This turns out to be true in general, and an $n \times n$ matrix $\mathbf{A}$ has an inverse if and only if $|\mathbf{A}|$ is not equal to zero. If $|\mathbf{A}| \neq 0$ we say that $\mathbf{A}$ is *nonsingular or invertible*. If $|\mathbf{A}| = 0$ we say that $\mathbf{A}$ is *singular*. Some useful properties of determinants are given in Table 2.5. We conclude this

section with two important facts involving matrix inverses.

Fact 2.19 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be nonsingular matrices having the same dimensions. Then*

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}.$$

The following fact says that the inverse and transpose operations can be performed in either order on a nonsingular matrix with the same result.

Fact 2.20 *If* $\mathbf{A}$ *is a nonsingular matrix then*

$$(\mathbf{A}^{-1})^T = (\mathbf{A}^T)^{-1} \stackrel{\text{def}}{=} \mathbf{A}^{-T}.$$

2.4.4 The Gram Matrix Test for Linear Independence

It is often important to know when a given set of vectors is linearly independent. The definition of linear independence (Definition 2.1) does not provide a convenient way to test a given set of vectors for linear independence. In this section, we state without proof a computational test based on the determinant of a Gram matrix. A proof may be found in [10].

Definition 2.19 *The* **Gram Matrix** $\mathbf{G}$ *associated with the* k *vectors* $\mathbf{x}_1, \cdots, \mathbf{x}_k \in \mathbf{R}^n$ *is a* $k \times k$ *matrix whose* (i, j) *element is given by* $g_{ij} = \mathbf{x}_i^T \mathbf{x}_j$. *If the given vectors are arranged as columns in a matrix as follows*

$$\mathbf{X} = [\, \mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_k \,]$$

then the Gram matrix can be written as

$$\mathbf{G} = \mathbf{X}^T\mathbf{X}.$$

The Gram matrix can be used to test whether or not a given set of vectors is linearly independent, as shown by the following

Fact 2.21 *A set of vectors* $\mathbf{x}_1, \cdots, \mathbf{x}_k \in \mathbf{R}^n$ *is linearly independent if and only if the determinant of the Gram matrix associated with these vectors is not equal to zero.*

EXAMPLE 2.8

Let us use the Gram determinant to test if the following vectors are linearly independent

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}.$$

The Gram matrix is

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

The determinant of $\mathbf{G}$ is $|\mathbf{G}| = 3$, and so the vectors are linearly independent.

■

The Gram matrix can also be used to test if a given vector is in a certain subspace. For example, if $\mathbf{x}_1$ and $\mathbf{x}_2$ are linearly independent vectors, then Fact 2.10 says that the set of all linear combinations of $\mathbf{x}_1$ and $\mathbf{x}_2$ is a 2-dimensional subspace of $\mathbf{R}^3$. Let us denote this subspace as

$$R = \{ \text{ all linear combinations of } \mathbf{x}_1 \text{ and } \mathbf{x}_2 \}. \tag{2.14}$$

Suppose we are given a vector $\mathbf{y}$. How could we check if $\mathbf{y} \in R$? The answer is as follows. If $\mathbf{y} \in R$, it can be expressed as a linear combination of $\mathbf{x}_1$ and $\mathbf{x}_2$. In this case the vectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}$ are linearly dependent (see Definition 2.2 on page 29). On the other hand, if the vectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}$ are linearly independent, then $\mathbf{y}$ cannot be expressed as a linear combination of $\mathbf{x}_1, \mathbf{x}_2$ and so $\mathbf{y} \notin R$. This procedure to test if a given vector belongs to a subspace is summarized in the following fact.

Fact 2.22 *Let R be an m-dimensional subspace of $\mathbf{R}^n$ and let $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m$ be a basis for R. Let $\mathbf{y}$ be a vector in $\mathbf{R}^n$ and define $\mathbf{W}$ to be the following matrix*

$$\mathbf{W} \stackrel{\text{def}}{=} [\, \mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \mathbf{x}_m \quad \mathbf{y} \,].$$

If the columns of $\mathbf{W}$ are linearly independent then $\mathbf{y} \notin R$. If the columns of $\mathbf{W}$ are linearly dependent the $\mathbf{y} \in R$.

The test for linear independence can be done by computing the determinant of the Gram matrix $\mathbf{G} = \mathbf{W}^T\mathbf{W}$.

EXAMPLE 2.9

Let R be the subspace defined by

$$R = \{ \text{ all linear combinations of } \mathbf{x}_1 \text{ and } \mathbf{x}_2 \}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are defined in the previous example. Check whether the following vectors are in R

$$\mathbf{y} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix}.$$

We first form the Gram matrix for $\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}$.

$$\mathbf{G}_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 2 \\ 2 & 2 & 3 \end{bmatrix}.$$

The determinant of $\mathbf{G}_1$ can be calculated to be 1. Thus $\mathbf{y} \notin R$.

The Gram matrix for $\mathbf{x}_1, \mathbf{x}_2, \mathbf{z}$ is

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 3 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 3 \\ 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 4 \\ 1 & 2 & 5 \\ 4 & 5 & 14 \end{bmatrix}.$$

The determinant of $\mathbf{G}_2$ can be calculated to be zero, and so $\mathbf{z} \in R$. Since $\mathbf{z} \in R$, it should be possible to express $\mathbf{z}$ as a linear combination of $\mathbf{x}_1$ and $\mathbf{x}_2$. It is easy to see that

$$\mathbf{z} = \mathbf{x}_1 + 2\mathbf{x}_2.$$

■

The previous example talked about a subspace R as the set of all linear combinations of two vectors. We now show a more convenient representation for such a subspace. Let the m-dimensional subspace R be defined as follows

$$R = \{ \text{ all linear combinations of } \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m \}. \tag{2.15}$$

Using Fact 2.12 we can write any linear combination of $\mathbf{x}_1, \cdots, \mathbf{x}_m$ as $\mathbf{X}\boldsymbol{\alpha}$, where

$$\mathbf{X} = [\, \mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_m \,]$$

and

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix}$$

is a vector of expansion coefficients. Thus every vector in the subspace R can be represented as $\mathbf{X}\boldsymbol{\alpha}$ for some $\boldsymbol{\alpha}$, and a definition of R which is equivalent to (2.15) is

$$R = \{\mathbf{x} = \mathbf{X}\boldsymbol{\alpha} \text{ for some } \boldsymbol{\alpha}\}. \tag{2.16}$$

We conclude this section with a corollary to Fact 2.21. Consider the case when $k = n$ so that the matrix $\mathbf{X}$ is $n \times n$. The Gram matrix is $\mathbf{G} = \mathbf{X}^T\mathbf{X}$ and its determinant is

$$\begin{aligned} |\mathbf{G}| &= |\mathbf{X}^T\mathbf{X}| \\ &= |\mathbf{X}^T||\mathbf{X}| \\ &= |\mathbf{X}|^2 \end{aligned}$$

where we have used some of the properties of determinants given in Table 2.5 on page 45. Since $|\mathbf{G}| = |\mathbf{X}|^2$, then $|\mathbf{G}| = 0$ if and only if $|\mathbf{X}| = 0$. Thus to test the linear independence of n vectors in $\mathbf{R}^n$ we do not need to form the Gram matrix. The determinant of the matrix formed from the given vectors can be used. In this discussion, the n vectors were the columns of the matrix $\mathbf{X}$. A similar development is possible when n vectors in $\mathbf{R}^n$ are placed as rows of a matrix. This gives the following

Fact 2.23 *The columns (or rows) of a $n \times n$ matrix $\mathbf{A}$ are linearly independent if and only if* $\det(\mathbf{A}) \neq 0$.

2.5 Matrix Subspaces and Projections

Many of the design formulas used in state-space control theory reduce to the solution of linear equations. The theory of linear equations is in turn rooted in the foundations of linear algebra. Previous sections of this chapter have reviewed basic facts from linear algebra and matrices. Before dealing specifically with linear equations, we first develop the four fundamental subspaces associated with a matrix. We also develop in this section the notions of orthogonality and projections. With these tools, the next section then develops the theory of linear equations.

2.5.1 Orthogonality

In the previous section, we defined the inner product of two vectors in $\mathbf{R}^n$ (Definition 2.14). The inner product has a simple geometric interpretation in $\mathbf{R}^2$. To develop this interpretation, we first define the length (norm) of a vector.

Definition 2.20 *The* **norm**[1] *of a vector* $\mathbf{x} \in \mathbf{R}^n$*, denoted by* $\|\mathbf{x}\|$ *is given by the positive square root*

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}^T\mathbf{x}}.$$

The above definition of a norm generalizes the notion of Euclidean distance in $\mathbf{R}^3$ to arbitrary dimensions. Note in Fig. 2.1 that the length of the vector $\mathbf{x} \in \mathbf{R}^2$ is equal to 5 (by the Pythagorean theorem), and this result is also obtained using the definition of the norm of $\mathbf{x}$.

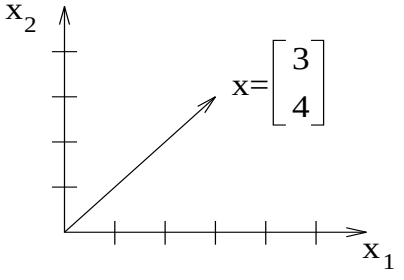


Figure 2.1 The length of the vector $\mathbf{x}$ is given by $\|\mathbf{x}\| = \sqrt{3^2 + 4^2} = 5$.

We can expand the norm squared of $\mathbf{x}$ in terms of the components of $\mathbf{x}$ as follows

$$\|\mathbf{x}\|^2 = \mathbf{x}^T\mathbf{x} = x_1^2 + x_2^2 + \cdots + x_n^2.$$

[1]This norm is called the Euclidean norm or 2-norm. There are many other norms for vectors in $\mathbf{R}^n$, but only the 2-norm will be used in this book

Because the right-hand side of the above equation consists of a sum of squares, it can only equal zero if each x_i is zero. This establishes the following fact.

Fact 2.24 $\|\mathbf{x}\| = 0 \Longleftrightarrow \mathbf{x} = \mathbf{0}$.

Suppose now that $\mathbf{x}$ and $\mathbf{y}$ are two vectors in $\mathbf{R}^2$ as shown in Fig. 2.2, and let $\mathbf{c} = \mathbf{x} - \mathbf{y}$. By the law of cosines for the triangle shown in Fig. 2.2, we have

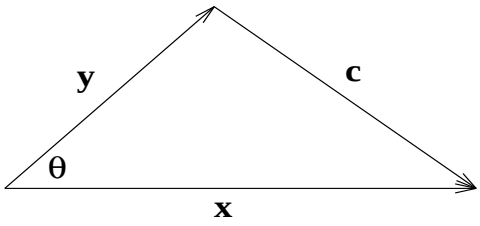


Figure 2.2 A triangle formed by the vectors $\mathbf{x}$ and $\mathbf{y}$. $\mathbf{c} = \mathbf{x} - \mathbf{y}$ and so $\|\mathbf{c}\| = \|\mathbf{x} - \mathbf{y}\| = \|\mathbf{y} - \mathbf{x}\|$.

$$\|\mathbf{c}\|^2 = \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\|\mathbf{x}\|\|\mathbf{y}\| \cos\theta. \tag{2.17}$$

The square of the norm of $\mathbf{c} = \mathbf{x} - \mathbf{y}$ can be expanded as follows

$$\begin{aligned} \|\mathbf{x} - \mathbf{y}\|^2 &= (\mathbf{x} - \mathbf{y})^T(\mathbf{x} - \mathbf{y}) \\ &= \mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{y} - \mathbf{y}^T\mathbf{x} + \mathbf{y}^T\mathbf{y} \\ &= \|\mathbf{x}\|^2 - 2\mathbf{x}^T\mathbf{y} + \|\mathbf{y}\|^2. \end{aligned} \tag{2.18}$$

Substituting the above equation into (2.17) and solving for $\cos\theta$ yields

$$\cos\theta = \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}. \tag{2.19}$$

In words, the above equation says that the cosine of the (acute) angle between the vectors $\mathbf{x}$ and $\mathbf{y}$ is given by the inner product $\mathbf{x}^T\mathbf{y}$ normalized by the lengths of $\mathbf{x}$ and $\mathbf{y}$. The development of (2.19) assumed the vector space was $\mathbf{R}^2$, but (2.19) can be evaluated for vectors in $\mathbf{R}^n$ for arbitrary n. For $n \geq 3$, we take (2.19) as the *definition* of the angle between vectors. This definition is then used to define the notion of orthogonal vectors. The vectors $\mathbf{x}$ and $\mathbf{y}$ in Fig. 2.2 are orthogonal if the angle $\theta = 90°$. In this case, $\cos\theta = 0$, and from (2.19) this means that $\mathbf{x}^T\mathbf{y} = 0$. The inner product is used to define orthogonality for two vectors as follows.

Definition 2.21 *Two vectors* $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$ *are said to be* **orthogonal** *if their inner product* $\mathbf{x}^T\mathbf{y}$ *is zero.*

By defining orthogonality in terms of the inner product, the definition applies to arbitrary dimensions. The definition of orthogonality between two vectors can be used to define a notion of orthogonality for sets of vectors, and for subspaces.

Definition 2.22 *A set of nonzero vectors* $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k\}$ *is said to be an* **orthogonal set of vectors** *if* $\mathbf{x}_i$ *is orthogonal to* $\mathbf{x}_j$ *for all* $i \neq j$. *If, in addition,* $\|\mathbf{x}_i\| = 1,\ i = 1, \cdots, k$ *then the set of vectors is said to be* **orthonormal**.

Fact 2.25 *An orthogonal (or orthonormal) set of nonzero vectors is linearly independent.*

PROOF. To establish this result with a proof by contradiction, assume that the given set of vectors is linearly dependent. Then by Fact 2.2, it is possible to express one vector, say $\mathbf{x}_i$, as a linear combination of the others. That is

$$\mathbf{x}_i = \alpha_1 \mathbf{x}_1 + \cdots + \alpha_{i-1}\mathbf{x}_{i-1} + \alpha_{i+1}\mathbf{x}_{i+1} + \cdots + \alpha_k \mathbf{x}_k.$$

If we multiply both sides of this equation by $\mathbf{x}_i^T$ and use the fact that all the inner products on the right-hand side of the equation are zero because of orthogonality, the result is

$$\|\mathbf{x}_i\|^2 = 0.$$

From Fact 2.24 this implies that $\mathbf{x}_i = \mathbf{0}$ which contradicts the fact that the given vectors are not zero.

■

Definition 2.23 *Two subspaces* S_1 *and* S_2 *of* $\mathbf{R}^n$ *are said to be* **orthogonal subspaces** *if, for any* $\mathbf{x} \in S_1$ *and any* $\mathbf{y} \in S_2$, *the vectors* $\mathbf{x}$ *and* $\mathbf{y}$ *are orthogonal. If* $\mathbf{x}_1, \cdots, \mathbf{x}_p$ *is a basis for* S_1 *and* $\mathbf{y}_1, \cdots, \mathbf{y}_q$ *is a basis for* S_2 *then* S_1 *and* S_2 *are orthogonal if* $\mathbf{x}_i^T \mathbf{y}_j = 0, i = 1, \cdots, p;\ j = 1, \cdots, q$.

2.5.2 Orthogonal Projections

The definition of orthogonality given in the previous section is a generalization of the familiar geometric concept of perpendicular lines. The importance of orthogonality comes from its use in deriving optimal approximations. From geometry, it is known that the shortest distance from a point to a line occurs along the line segment which is perpendicular to the given line. Similarly, the shortest distance from a point to a plane occurs along the line segment which is perpendicular to the plane.

The optimal solutions mentioned above can be expressed in the language of linear algebra as follows: the shortest distance between a vector and a subspace occurs along a vector which is orthogonal to the subspace. Fig. 2.3 shows a vector $\mathbf{y}$ (represented by an arrow) and a subspace R (represented by a plane). Of all vectors which connect the plane and the vector $\mathbf{y}$, the vector $\mathbf{y}_e$ is the one with the smallest norm. The vector $\mathbf{y}_e$ is *orthogonal* to the subspace R. The vector $\mathbf{y}_p$ is the vector in R which is the best approximation to $\mathbf{y}$. The vector $\mathbf{y}_p$ is called *the projection of* $\mathbf{y}$ *onto* R.

Assume that the subspace R is defined as the subspace spanned by a given set of linearly independent vectors $\mathbf{x}_1, \cdots, \mathbf{x}_r$. From Fact 2.10, the given vectors are a basis for R. The question we would like to answer now is: given a vector $\mathbf{y}$ and a basis for a subspace R, how can we compute $\mathbf{y}_p$, the projection of $\mathbf{y}$ onto R?

The answer is quite simple, and it comes from the fact that the "error vector"

$$\mathbf{y}_e = \mathbf{y} - \mathbf{y}_p \tag{2.20}$$

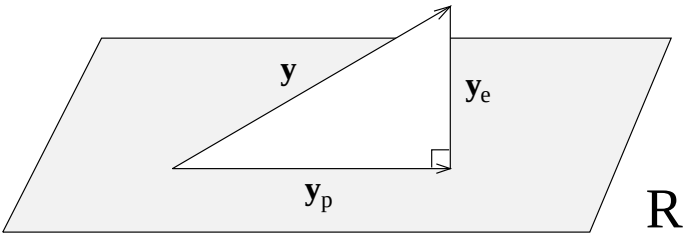


Figure 2.3 $\mathbf{y}_p$ is the best approximation to $\mathbf{y}$ of any vector in R. $\mathbf{y}_p$ is obtained by projecting $\mathbf{y}$ onto R.

is orthogonal to the subspace R, which means that the error vector must be orthogonal to every vector in a basis for R (see Definition 2.23), or

$$\mathbf{y}_e^T \mathbf{x}_i = 0\,, \quad i = 1, \cdots, r. \tag{2.21}$$

However these orthogonality equations cannot be evaluated directly because $\mathbf{y}_e$ is a function of $\mathbf{y}_p$ (see (2.20)), which is the vector we are trying to find! We do know that $\mathbf{y}_p$ is in the subspace R, and from (2.16) on page 48 we know that any vector in R can be represented as $\mathbf{X}\boldsymbol{\alpha}$ for some vector $\boldsymbol{\alpha}$ of expansion coefficients, i.e.

$$\mathbf{y}_p = \mathbf{X}\boldsymbol{\alpha} \text{ for some } \boldsymbol{\alpha} \text{ where } \mathbf{X} = [\,\mathbf{x}_1 \quad \cdots \quad \mathbf{x}_r\,]\,. \tag{2.22}$$

We can substitute this expression into the orthogonality equations and *solve for the expansion coefficients* $\boldsymbol{\alpha}$. Once we find the coefficients $\boldsymbol{\alpha}$ which satisfy the orthogonality conditions, (2.22) is used to find $\mathbf{y}_p$.

To be specific, we substitute (2.22) and (2.20) into (2.21) to obtain the following equations, which are called called *the normal equations*

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\alpha}^*)^T \mathbf{x}_i = 0\,, \quad i = 1, \cdots r, \tag{2.23}$$

or

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\alpha}^*)^T \mathbf{X} = \mathbf{0}$$

or

$$\mathbf{y}^T \mathbf{X} = \boldsymbol{\alpha}^{*T} \mathbf{X}^T \mathbf{X}.$$

We put a $*$ on $\boldsymbol{\alpha}$ to show that this vector of coefficients corresponds to the optimal approximating vector $\mathbf{y}_p$. We can take the transpose of the above equation and solve for the coefficient vector $\boldsymbol{\alpha}^*$ which satisfies the normal equations as follows

$$\boldsymbol{\alpha}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \tag{2.24}$$

Note that the $r \times r$ matrix $\mathbf{X}^T\mathbf{X}$ is the Gram matrix for the vectors $\mathbf{x}_1, \cdots, \mathbf{x}_r$. Since these vectors are linearly independent, this Gram matrix has a non-zero determinant and hence is invertible. Recall that the vector $\mathbf{y}_p$ that we are looking for is expressed as $\mathbf{y}_p = \mathbf{X}\boldsymbol{\alpha}$, and we now know that $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*$ satisfies the normal equations (orthogonality conditions). Thus using (2.24) we have

$$\begin{aligned} \mathbf{y}_p &= \mathbf{X}\boldsymbol{\alpha}^* \\ &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{P}_R\,\mathbf{y}, \end{aligned} \tag{2.25}$$

where the matrix $\mathbf{P}_R$ is called *the orthogonal projection matrix onto the subspace* R. To project any vector $\mathbf{y}$ onto the subspace R, simply multiply $\mathbf{y}$ by $\mathbf{P}_R$. From (2.25) we see that the projection matrix is built from the matrix $\mathbf{X}$, whose columns are a basis for the subspace R, as follows

$$\mathbf{P}_R = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T. \tag{2.26}$$

We note in passing that all orthogonal projection matrices share two properties, each of which is easy to verify using (2.26)

1. A projection matrix is *symmetric* : $\mathbf{P}_R = \mathbf{P}_R^T$.
2. A projection matrix is *idempotent* : $\mathbf{P}_R\,\mathbf{P}_R = \mathbf{P}_R$.

EXAMPLE 2.10

In Example 2.9 we showed that the vector $\mathbf{y}$ is not in the subspace R. In this example we compute the projection matrix P_R and the vector $\mathbf{y}_p$, the projection of $\mathbf{y}$ onto R. Recall that the basis vectors for R, arranged in the matrix $\mathbf{X}$ are

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

Using (2.26) we compute P_R as

$$P_R = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2/3 & 1/3 & -1/3 \\ 1/3 & 2/3 & 1/3 \\ -1/3 & 1/3 & 2/3 \end{bmatrix}.$$

We can now project $\mathbf{y}$ onto R as follows

$$\mathbf{y}_p = P_R\mathbf{y} = \begin{bmatrix} 2/3 & 1/3 & -1/3 \\ 1/3 & 2/3 & 1/3 \\ -1/3 & 1/3 & 2/3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2/3 \\ 4/3 \\ 2/3 \end{bmatrix}.$$

The error between $\mathbf{y}$ and its projection onto R is

$$\mathbf{y}_e = \mathbf{y} - \mathbf{y}_p = \begin{bmatrix} 1/3 \\ -1/3 \\ 1/3 \end{bmatrix}.$$

We can show that the error vector is in fact orthogonal to the subspace R by showing that it is orthogonal to the basis vectors $\mathbf{x}_1$ and $\mathbf{x}_2$.

$$\mathbf{x}_1^T\mathbf{y}_e = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ -1/3 \\ 1/3 \end{bmatrix} = 0$$

and

$$\mathbf{x}_2^T\mathbf{y}_e = [\,0 \quad 1 \quad 1\,] \begin{bmatrix} 1/3 \\ -1/3 \\ 1/3 \end{bmatrix} = 0.$$

■

2.5.3 Orthogonal Complements and the Four Fundamental Subspaces

Suppose we are given r linearly independent vectors in $\mathbf{R}^n$, $\mathbf{x}_1, \cdots, \mathbf{x}_r$, where $r < n$. Then from Fact 2.10, we have that

$$R = \{\text{all linear combinations of } \mathbf{x}_1 \cdots \mathbf{x}_r\}$$

is a subspace of $\mathbf{R}^n$. Given any subspace of $\mathbf{R}^n$, we can always define its **orthogonal complement**. For example, the orthogonal complement of the subspace R, which we denote by $R^\perp$ is defined as

$$R^\perp = \{\mathbf{y} \in \mathbf{R}^n \text{ such that } \mathbf{y}^T\mathbf{x} = 0 \text{ for } \textit{all } \mathbf{x} \in R\}.$$

That is, elements of $R^\perp$ are orthogonal to *every* vector in R. This definition of $R^\perp$ is somewhat cumbersome, because it requires that we check for orthogonality with *every vector* in R. Thus we now develop an equivalent definition of $R^\perp$.

Recall that if a basis is defined for some vector space, then every vector in that space can be (uniquely) expressed as a linear combination of basis vectors. For the space R defined above, the linearly independent vectors $\mathbf{x}_1 \cdots \mathbf{x}_r$ constitute a basis. If a vector $\mathbf{y}$ were orthogonal to each of the basis vectors in R, then $\mathbf{y}$ would be orthogonal to every vector in R since every vector in R is expressible in terms of the basis. We can write an equation to express this fact as follows. Assume $\mathbf{y}^T\mathbf{x}_i = 0, i = 1, \cdots r$, and let $\mathbf{x}$ be an arbitrary vector in R which is expressed in terms of the basis vectors as $\mathbf{x} = \sum_{i=1}^{r} \alpha_i \mathbf{x}_i$. Then

$$\begin{aligned} \mathbf{y}^T\mathbf{x} &= \mathbf{y}^T \sum_{i=1}^{r} \alpha_i \mathbf{x}_i \\ &= \sum_{i=1}^{r} \alpha_i \mathbf{y}^T\mathbf{x}_i \\ &= 0. \end{aligned}$$

Thus an equivalent definition for the orthogonal complement of a subspace R with basis $\mathbf{x}_1, \cdots, \mathbf{x}_r$ is

$$R^\perp = \{\mathbf{y} \in \mathbf{R}^n \text{ such that } \mathbf{y}^T\mathbf{x}_i = 0, i = 1, \cdots r\}.$$

If all the vectors in $R^\perp$ are orthogonal to basis vectors for R, then it is clear that basis vectors for $R^\perp$ are orthogonal to basis vectors for R. In other words, if $\{\mathbf{x}_1, \cdots, \mathbf{x}_r\}$ is a basis for R and $\{\mathbf{y}_1, \cdots, \mathbf{y}_s\}$ is a basis for $R^\perp$ then

$$\mathbf{x}_i^T\mathbf{y}_j = 0\,, \quad i = 1, \cdots, r \text{ and } j = 1, \cdots, s. \tag{2.27}$$

Using (2.27) it can be shown that the vectors $\mathbf{x}_1, \cdots, \mathbf{x}_r, \mathbf{y}_1, \cdots, \mathbf{y}_2$ are linearly independent (compare Fact 2.25). Since these vectors belong to $\mathbf{R}^n$, an n-dimensional vector

space, we must have $r + s \leq n$ (Fact 2.7). With a little more work it is possible to show that $r + s = n$ [84].

A useful property of $\mathbf{R}^n$ is that it can be decomposed into a *direct sum of orthogonal subspaces*. In particular, $\mathbf{R}^n$ can be written as a direct sum of the subspaces R and $R^\perp$. This direct sum of subspaces is written as follows

$$\mathbf{R}^n = R \oplus R^\perp$$

and is explained by the following definition: any vector $\mathbf{v} \in \mathbf{R}^n$ can be decomposed *uniquely* as the sum of two orthogonal vectors $\mathbf{v}_R$ and $\mathbf{v}_{R^\perp}$ where $\mathbf{v}_R \in R$ and $\mathbf{v}_{R^\perp} \in R^\perp$. Notice that $\mathbf{v}_R$ *is the projection of* $\mathbf{v}$ *onto the subspace* R, and $\mathbf{v}_{R^\perp}$ *is the projection of* $\mathbf{v}$ *onto the subspace* $R^\perp$.

EXAMPLE 2.11

Recall from Example 2.10 that

$$\mathbf{y} = \mathbf{y}_p + \mathbf{y}_e$$

where $\mathbf{y}_p \in R$ and $\mathbf{y}_e$ is orthogonal to R so that $\mathbf{y}_e \in R^\perp$. We have seen that $\mathbf{y}_p$ is the projection of $\mathbf{y}$ onto R and we now want to show that $\mathbf{y}_e$ is the projection of $\mathbf{y}$ onto $R^\perp$. In order to do this, we need a basis for $R^\perp$. In this example, the vector space is $\mathbf{R}^3$ ($n = 3$) and the subspace R has dimension 2 ($r = 2$). Thus the dimension of the orthogonal complement $R^\perp$ must be $n - r = s = 1$. A basis for $R^\perp$ consists of a single vector which is orthogonal to the basis vectors for R. Such a vector is[2]

$$\mathbf{y}_1 = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}.$$

We use $\mathbf{y}_1$ to form the projection matrix onto $R^\perp$ as follows.

$$P_{R^\perp} = \mathbf{y}_1(\mathbf{y}_1^T\mathbf{y}_1)^{-1}\mathbf{y}_1 = \begin{bmatrix} 1/3 & -1/3 & 1/3 \\ -1/3 & 1/3 & -1/3 \\ 1/3 & -1/3 & 1/3 \end{bmatrix}.$$

Then

$$P_{R^\perp}\mathbf{y} = \begin{bmatrix} 1/3 \\ -1/3 \\ 1/3 \end{bmatrix}$$

which is the vector $\mathbf{y}_e$ calculated in Example 2.10.

Now that orthogonal complements have been defined, we proceed to define the four fundamental subspaces associated with a matrix.

Given a matrix $\mathbf{A}$, there are four subspaces that can be defined in terms of the rows and columns of $\mathbf{A}$. Two of these subspaces have already been defined in Section 2.4, namely the row space and column space of a matrix (Definitions 2.8 and 2.10, respectively). The orthogonal complements of the row and column spaces are the other two fundamental

[2]A tool for computing basis vectors for the orthogonal complement of a subspace is given in Section 2.7.

subspaces. We deal first with the orthogonal complement of the row space, which is called the null-space.

Definition 2.24 *The* **null space of a matrix $\mathbf{A}$, denoted null$(\mathbf{A})$** *is the set of all vectors* $\mathbf{x}$ *satisfying* $\mathbf{A}\mathbf{x} = \mathbf{0}$. *The dimension of the null-space of* $\mathbf{A}$ *is called the* **nullity** *of* $\mathbf{A}$ *and is denoted* $\nu(\mathbf{A})$.

Notice that any vector $\mathbf{x}$ which satisfies $\mathbf{A}\mathbf{x} = \mathbf{0}$ is orthogonal to all the rows of $\mathbf{A}$. Thus

$$\mathbf{x} \in \text{null}(\mathbf{A}) \Longrightarrow \mathbf{x} \in \text{row}(\mathbf{A})^{\perp}.$$

It can be shown that the reverse implication is also true, namely

$$\mathbf{x} \in \text{row}(\mathbf{A})^{\perp} \Longrightarrow \mathbf{x} \in \text{null}(\mathbf{A}).$$

Thus we have

Fact 2.26 *null($\mathbf{A}$) = row$(\mathbf{A})^{\perp}$.*

The fourth fundamental subspace is the orthogonal complement of the column space of $\mathbf{A}$. There is no special notation for this subspace.

2.5.4 The Rank of a Matrix

An important property of a matrix which is used to characterize solutions to linear equations is its rank. The rank of a matrix is also used in Chapters 6 and 7 to test for system theoretic properties such as controllability and observability.

The **row rank** of a matrix $\mathbf{A}$ is defined to be the maximum number of linearly independent rows in $\mathbf{A}$. The **column rank** of a matrix $\mathbf{A}$ is defined to be the maximum number of linearly independent columns in $\mathbf{A}$.

Fact N.2 Given any $m \times n$ matrix $\mathbf{A}$, the row rank of $\mathbf{A}$ is always equal to the column rank of $\mathbf{A}$.

PROOF: See [84]. ■

The surprising thing about the previous fact is that it is true even when m is not equal to n. Thus we can define the rank of a matrix as follows

Definition 2.25 *The* **rank** *of an $m \times n$ matrix $\mathbf{A}$, denoted $\rho(\mathbf{A})$, is equal to the number of linearly independent rows of $\mathbf{A}$, or equivalently, the number of linearly independent columns of $\mathbf{A}$.*

It is clear that an $m \times n$ matrix $\mathbf{A}$ cannot have more than m linearly independent rows – it only has m rows! Similarly, $\mathbf{A}$ cannot have more than n linearly independent columns. Thus we obtain the simple but useful fact

Fact 2.27 *The rank of an $m \times n$ matrix $\mathbf{A}$ is less than or equal to min(m, n).*

There is a useful fact relating the rank of a matrix and the dimension of its null space. Recall that the rank of a matrix equals the dimension of its row space, while the nullity is the dimension of the orthogonal complement of the row space. In the previous subsection we mentioned that the dimensions of a subspace and its orthogonal complement add up to the dimension of the vector space. Since the rows of an $m \times n$ matrix are elements of $\mathbf{R}^n$, we have the following fact.

Fact 2.28 *Given an $m \times n$ matrix $\mathbf{A}$, rank$(\mathbf{A})$ + nullity$(\mathbf{A}) = n$.*

A proof of this fact can be found in [84].

2.6 Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors are defined only for square matrices. Throughout this section we assume that $\mathbf{A}$ is an $n \times n$ matrix.

Definition 2.26 *Given a matrix $\mathbf{A}$, a nonzero vector $\mathbf{x}$ is said to be an* **eigenvector** *of $\mathbf{A}$ corresponding to* eigenvalue λ *if*

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}. \tag{2.28}$$

It is easy to see from this definition that eigenvectors are not unique. Indeed if $\mathbf{x}$ is an eigenvector of $\mathbf{A}$ corresponding to eigenvalue λ, then so is $\alpha\mathbf{x}$ for any $\alpha \neq 0$. We can derive three useful facts about eigenvalues directly from the definition. The first fact shows how eigenvalues are affected when a matrix $\mathbf{A}$ is multipied by a scalar α. The second fact shows how eigenvalues are affected when a scalar multiple of the identity matrix is added to $\mathbf{A}$. The third fact says that the eigenvalues are unchanged when a matrix is transformed in a certain way.

Fact 2.29 *If λ is an eigenvalue of $\mathbf{A}$ then $\alpha\lambda$ is an eigenvalue of $\alpha\mathbf{A}$.*

PROOF. From Definition 2.26 we know that there is an eigenvector $\mathbf{x}$ such that

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}.$$

If we let $\mathbf{M} = \alpha\mathbf{A}$ then

$$\begin{aligned}\mathbf{M}\mathbf{x} &= (\alpha\mathbf{A})\mathbf{x} \\ &= \alpha(\mathbf{A}\mathbf{x}) \\ &= \alpha\lambda\mathbf{x}\end{aligned}$$

which shows that $\alpha\lambda$ is an eigenvalue of $\mathbf{M} = \alpha\mathbf{A}$.

■

The proof of the next fact is left as an exercise.

Fact 2.30 *If λ is an eigenvalue of $\mathbf{A}$ then $\alpha + \lambda$ is an eigenvalue of $\alpha\mathbf{I} + \mathbf{A}$.*

Before presenting the next fact we must first have the following definition.

Definition 2.27 *Let $\mathbf{A}$ be an $n \times n$ matrix, let $\mathbf{T}$ be a nonsingular $n \times n$ matrix, and let $\bar{\mathbf{A}} \stackrel{\text{def}}{=} \mathbf{T}\mathbf{A}\mathbf{T}^{-1}$. Then $\mathbf{A}$ and $\bar{\mathbf{A}}$ are said to be* **similar** *matrices. They are related by the* **similarity transformation** *matrix $\mathbf{T}$.*

Fact 2.31 *Similar matrices have the same eigenvalues.*

PROOF. We first show that if λ is an eigenvalue of $\mathbf{A}$ then it must also be an eigenvalue of $\bar{\mathbf{A}}$. If λ is an eigenvalue of $\mathbf{A}$ then we have

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

$$\mathbf{T}\mathbf{A}\mathbf{x} = \lambda\mathbf{T}\mathbf{x}$$

$$\mathbf{T}\mathbf{A}(\mathbf{T}^{-1}\mathbf{T})\mathbf{x} = \lambda\mathbf{T}\mathbf{x}$$

$$\bar{\mathbf{A}}(\mathbf{T}\mathbf{x}) = \lambda(\mathbf{T}\mathbf{x}).$$

The last line shows that λ is an eigenvalue of $\bar{\mathbf{A}}$. We also see that the corresponding eigenvector is $\mathbf{T}\mathbf{x}$.

We now show that if λ is an eigenvalue of $\bar{\mathbf{A}}$ then it must also be an eigenvalue of $\mathbf{A}$. We have

$$\bar{\mathbf{A}}\mathbf{y} = \lambda\mathbf{y}$$

$$\mathbf{T}\mathbf{A}\mathbf{T}^{-1}\mathbf{y} = \lambda\mathbf{y}$$

$$\mathbf{A}(\mathbf{T}^{-1}\mathbf{y}) = \lambda(\mathbf{T}^{-1}\mathbf{y}).$$

The last line shows that λ is an eigenvalue of $\mathbf{A}$ corresponding to eigenvector $\mathbf{T}^{-1}\mathbf{y}$.

■

We now derive three more facts about eigenvalues which involve the use of the determinant of a special matrix.

Fact 2.32 *The scalar λ is an eigenvalue of $\mathbf{A}$ if and only if $det(\lambda\mathbf{I} - \mathbf{A}) = 0$.*

PROOF. If λ is an eigenvalue of $\mathbf{A}$ then there is an eigenvector $\mathbf{x}$ such that

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}.$$

We can rewrite $\lambda\mathbf{x}$ as $\lambda\mathbf{I}\mathbf{x}$ and rearrange the above equation to yield

$$(\lambda\mathbf{I} - \mathbf{A})\mathbf{x} = \mathbf{0}.$$

Notice that the eigenvector $\mathbf{x}$ is in the null space of the matrix $\lambda\mathbf{I} - \mathbf{A}$. Since $(\lambda\mathbf{I} - \mathbf{A})$ has at least one nonzero vector in its null space, the nullity of $(\lambda\mathbf{I} - \mathbf{A})$ is at least one, and so the rank of $(\lambda\mathbf{I} - \mathbf{A})$ must be less than or equal to $n - 1$ (see Fact 2.28). Since the rank of $(\lambda\mathbf{I} - \mathbf{A})$ is less than n, this means that its rows (and columns) are linearly dependent. From Fact 2.23 this means that $\det(\lambda\mathbf{I} - \mathbf{A}) = 0$.

To show the reverse implication, assume that $\det(\lambda\mathbf{I} - \mathbf{A}) = 0$. Then by Fact 2.23 the columns of $(\lambda\mathbf{I} - \mathbf{A})$ are linearly dependent. That means that there exists a nonzero vector $\mathbf{x}$ such that $(\lambda\mathbf{I} - \mathbf{A})\mathbf{x} = \mathbf{0}$ or $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$. Thus λ is an eigenvalue of $\mathbf{A}$.

■

The following fact is given without proof.

Fact 2.33 *Let $\mathbf{A}$ be an $n \times n$ matrix. Then $\det(\lambda\mathbf{I} - \mathbf{A})$ is a monic polynomial in λ of degree n. It is denoted $a(\lambda)$ and called the* **characteristic polynomial** *of $\mathbf{A}$. We have*

$$a(\lambda) \stackrel{\text{def}}{=} \det(\lambda\mathbf{I} - \mathbf{A}) = \lambda^n + a_1\lambda^{n-1} + \cdots + a_n.$$

In Table 3.5 on page 114 we give an algorithm to compute the characteristic polynomial of a matrix. Putting the two previous facts together we get

Fact 2.34 *An $n \times n$ matrix $\mathbf{A}$ has n eigenvalues which are the roots of the characteristic polynomial $a(\lambda)$.*

The above facts suggest a way to calculate the eigenvalues and eigenvectors of a matrix. The first step is to form the characteristic polynomial and find its roots $\lambda_1, \cdots, \lambda_n$. These are the eigenvalues of $\mathbf{A}$. For each eigenvalue, form the matrix $(\lambda_i\mathbf{I} - \mathbf{A})$ and find a nonzero vector in its null space. This vector will be an eigenvector of $\mathbf{A}$ (see the proof of Fact 2.32). If the eigenvalues of $\mathbf{A}$ are distinct it turns out that the nullity of $(\lambda\mathbf{I} - \mathbf{A})$ is equal to one, and each eigenvector is picked as any nonzero vector from a one-dimensional null space. The situation is more complicated for repeated eigenvalues. Algorithms for computing eigenvectors can be found in [22, 37]. The case of repeated eigenvalues can be found in [10, 42].

EXAMPLE 2.12

Consider the following matrix

$$\mathbf{A} = \begin{bmatrix} 1.3 & -0.4 \\ -0.2 & 1.1 \end{bmatrix}.$$

We can compute the characteristic polynomial as follows

$$\begin{aligned} a(\lambda) &= |\lambda \mathbf{I} - \mathbf{A}| \\ &= \left| \begin{bmatrix} \lambda - 1.3 & 0.4 \\ 0.2 & \lambda - 1.1 \end{bmatrix} \right| \\ &= (\lambda - 1.3)(\lambda - 1.1) - 0.8 \\ &= \lambda^2 - 2.4\lambda + 1.35. \end{aligned}$$

The eigenvalues of $\mathbf{A}$ are the roots of the $a(\lambda)$ can be calculated using the quadratic formula to be $\lambda_1 = 1.5$ and $\lambda_2 = 0.9$.

To compute the eigenvector corresponding to λ_1, we look for a vector in the null space of $\lambda_1 \mathbf{I} - \mathbf{A}$.

$$1.5\,\mathbf{I} - \mathbf{A} = \begin{bmatrix} 0.2 & 0.4 \\ 0.2 & 0.4 \end{bmatrix}.$$

A null-space vector for this matrix is found (by inspection) to be

$$\mathbf{x}_1 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}.$$

The second eigenvector is computed by finding a vector in the null space of $\lambda_2 \mathbf{I} - \mathbf{A}$.

$$0.9\,\mathbf{I} - \mathbf{A} = \begin{bmatrix} -0.4 & 0.4 \\ 0.2 & -0.2 \end{bmatrix}.$$

A null-space vector for this matrix is found (by inspection) to be

$$\mathbf{x}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

■

In the above example, the null-space vectors were found by inspection. There is a systematic procedure which can be carried out by hand to compute null-space vectors [10, 84]. However we do not stress hand computations in this book. We assume that the reader has access to a computer program such as MATLAB to perform the necessary calculations.

We now give another fact about eigenvalues without proof. The proof of the following fact may be found in [10, 42]

Fact 2.35 *Eigenvectors corresponding to distinct eigenvalues are linearly independent.*

This fact can be used to derive a useful representation of a matrix. Suppose that the eigenvalues of a matrix $\mathbf{A}$ are distinct. Then we have

$$\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i\ ,\ \ i = 1, \cdots, n.$$

The above equations may be written as a single matrix equation as follows

$$\mathbf{A} \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}. \qquad (2.29)$$

Now let

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix} \text{ and } \mathbf{D} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}.$$

From Fact 2.35 the n columns of $\mathbf{X}$ are linearly independent, and so by Fact 2.23 on page 48 $\mathbf{X}$ is nonsingular. Using the definitions of $\mathbf{X}$ and $\mathbf{D}$ we can rewrite (2.29) as

$$\mathbf{AX} = \mathbf{XD}$$

or

$$\mathbf{A} = \mathbf{XDX}^{-1}. \tag{2.30}$$

The above equation says that the matrix $\mathbf{A}$ is similar to a diagonal matrix consisting of its eigenvalues. The columns of the similarity transformation matrix are the eigenvectors of $\mathbf{A}$.

Equation (2.30) is useful for computing powers of a matrix. A matrix $\mathbf{A}$ raised to an integer power k is just the matrix $\mathbf{A}$ multiplied by itself k times as shown below

$$\mathbf{A}^k \stackrel{\text{def}}{=} \overbrace{\mathbf{A}\,\mathbf{A}\cdots\mathbf{A}}^{k \text{ factors}}.$$

Substituting (2.30) into the above equation yields

$$\begin{aligned} \mathbf{A}^k &= (\mathbf{XDX}^{-1})(\mathbf{XDX}^{-1})\cdots(\mathbf{XDX}^{-1}) \\ &= \mathbf{XD}^k\mathbf{X}^{-1}. \end{aligned} \tag{2.31}$$

Notice that all the intermediate $\mathbf{XX}^{-1}$ terms cancel out in the expansion of $\mathbf{A}^k$. For example, consider the matrix $\mathbf{A}$ from Example 2.12. Using (2.31) we have

$$\begin{bmatrix} 1.3 & -0.4 \\ -0.2 & 1.1 \end{bmatrix}^k = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1.5^k & 0 \\ 0 & 0.9^k \end{bmatrix} \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}^{-1}.$$

The above equation is a useful computational tool to compute $\mathbf{A}^k$ for large values of k, but it also gives useful information about $\mathbf{A}^k$ for large k. For instance, because 1.5^k increases without bound as k increases, we see that the entries in the matrix $\mathbf{A}^k$ will increase without bound. However, if both of the eigenvalues had magnitudes less than one, then $\mathbf{A}^k$ would approach the zero matrix as k became large. Furthermore, the rate of increase or decrease of the elements in $\mathbf{A}^k$ depends on the magnitudes of the eigenvalues.

We conclude this section with four additional facts about eigenvalues.

Fact 2.36 *The determinant of a matrix equals the product of its eigenvalues.*

PROOF. We give the proof only for the case in which the matrix has distinct eigenvalues. However, the fact is true for any square matrix.

Let $\mathbf{A}$ be a matrix with distinct eigenvalues, let $\mathbf{X}$ be a matrix whose columns are the eigenvectors of $\mathbf{A}$ and let the corresponding eigenvalues be placed in a diagonal matrix

D. Let the determinant of the matrix $\mathbf{X}$ be denoted α. Then using the properties of determinants shown in Table 2.5 on page 45, we have

$$\begin{aligned}|\mathbf{A}| &= |\mathbf{X}\mathbf{D}\mathbf{X}^{-1}| \\ &= |\mathbf{X}||\mathbf{D}||\mathbf{X}^{-1}| \\ &= \alpha|\mathbf{D}|\tfrac{1}{\alpha} \\ &= |\mathbf{D}| \\ &= \lambda_1\, \lambda_2 \cdots \lambda_n.\end{aligned}$$

■

Since a matrix is nonsingular if and only if its determinant is nonzero, the above fact gives

Fact 2.37 *A matrix is nonsingular if and only if all its eigenvalues are nonzero.*

From Fact 2.32 we know that eigenvalues are the roots of $\det(\lambda\mathbf{I} - \mathbf{A})$ and from item 2 in Table 2.5 on page 45, we know that a matrix and its transpose have the same determinant. Thus the following fact is true.

Fact 2.38 *A matrix and its transpose have the same eigenvalues.*

The derivation of the following result is left as an exercise.

Fact 2.39 *The eigenvalues of a triangular matrix are equal to the diagonal elements of the matrix.*

An extension of the above fact to block-triangular matrices is

Fact 2.40 *The eigenvalues of a block-triangular matrix* $\mathbf{A}$ *are the union of the eigenvalues of the matrices on the main diagonal of* $\mathbf{A}$.

2.7 The Singular Value Decomposition

Any $m \times n$ matrix $\mathbf{A}$ of rank r has a singular value decomposition (SVD) which is a product of three matrices as follows

$$\underbrace{\mathbf{A}}_{m\times n} = \underbrace{\mathbf{U}}_{m\times m}\,\underbrace{\mathbf{\Sigma}}_{m\times n}\,\underbrace{\mathbf{V}^T}_{n\times n}, \tag{2.32}$$

where $\mathbf{U}^T\mathbf{U} = \mathbf{I}_m$, and $\mathbf{V}^T\mathbf{V} = \mathbf{I}_n$. The matrix $\mathbf{\Sigma}$ takes the following form

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & 0 & \cdots & 0 \\ 0 & \cdots & \sigma_r & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

The numbers σ_i are called the *singular values of* $\mathbf{A}$ and they are non-negative real numbers ordered in decreasing order, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$. Note that the matrix $\mathbf{A}$ may also possess a number of zero singular values. It can be shown that the squares of the nonzero singular values of $\mathbf{A}$ are the nonzero eigenvalues of $\mathbf{A}\mathbf{A}^T$ or $\mathbf{A}^T\mathbf{A}$ (both of these matrices have the same nonzero eigenvalues). It can also be shown that the columns of $\mathbf{U}$ are eigenvectors of $\mathbf{A}\mathbf{A}^T$, and columns of $\mathbf{V}$ are eigenvectors of $\mathbf{A}^T\mathbf{A}$.

We now partition the $\mathbf{U}$ and $\mathbf{V}$ matrices given by the SVD in (2.32) according to the rank r of $\mathbf{A}$:

$$\mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_r \mid \mathbf{u}_{r+1} \cdots \mathbf{u}_m] = [\mathbf{U}_1 \mid \mathbf{U}_2], \tag{2.33}$$

and

$$\mathbf{V} = [\mathbf{v}_1 \cdots \mathbf{v}_r \mid \mathbf{v}_{r+1} \cdots \mathbf{v}_n] = [\mathbf{V}_1 \mid \mathbf{V}_2]. \tag{2.34}$$

The vectors $\mathbf{u}_i$ are called the *left singular vectors of* $\mathbf{A}$, and it can be show that they are eigenvectors of $\mathbf{A}\mathbf{A}^T$. The vectors $\mathbf{v}_i$ are called the *right singular vectors of* $\mathbf{A}$, and it can be shown that they are eigenvectors of $\mathbf{A}^T\mathbf{A}$.

Note that zero rows in $\mathbf{\Sigma}$ multiply columns of $\mathbf{U}$ and zero columns of $\mathbf{\Sigma}$ multiply rows of $\mathbf{V}^T$. These zero rows and columns of $\mathbf{\Sigma}$ can be deleted, along with the corresponding columns and rows of $\mathbf{U}$ and $\mathbf{V}^T$ to produce the "economy size" SVD of $\mathbf{A}$:

$$\mathbf{A} = \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^T,$$

where

$$\mathbf{U}_1^T\mathbf{U}_1 = \mathbf{I}_r\,, \quad \mathbf{V}_1^T\mathbf{V}_1 = \mathbf{I}_r,$$

and $\mathbf{U}_1$ and $\mathbf{V}_1$ contain singular vectors corresponding to nonzero singular values as shown in (2.33). Recall that r in (2.33) is the number of nonzero singular values. Thus the rank of $\mathbf{A}$ is the number of nonzero singular values.

SVD and the Four Fundamental Subspaces

The SVD shows the rank of a matrix as the number r of non-zero singular values. When the matrices $\mathbf{U}$ and $\mathbf{V}$ are partitioned as shown in (2.33) and (2.34), the partitions provide orthonormal bases for the four fundamental subspaces of a matrix:

1. $\mathbf{U}_1$ is an orthonormal basis for the column space of $\mathbf{A}$.
2. $\mathbf{U}_2$ is an orthonormal basis for the orthogonal complement of the column space of $\mathbf{A}$.
3. $\mathbf{V}_1$ is an orthonormal basis for the row space of $\mathbf{A}$.
4. $\mathbf{V}_2$ is an orthonormal basis for the null-space of $\mathbf{A}$ (recall that the null space of a matrix is the same as the orthogonal complement of the row space).

EXAMPLE 2.13

In Examples 2.10 and 2.11 we considered a matrix $\mathbf{X}$ which we show below as the matrix $\mathbf{A}$.

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

MATLAB gives the following SVD of $\mathbf{A}$ (to 4 decimal places)

$$\mathbf{U} = \begin{bmatrix} 0.4082 & -0.7071 & 0.5774 \\ 0.8165 & 0.0000 & -0.5774 \\ 0.4082 & 0.7071 & 0.5774 \end{bmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} 1.7321 & 0 \\ 0 & 1.0000 \\ 0 & 0 \end{bmatrix},$$

and

$$\mathbf{V} = \begin{bmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{bmatrix}.$$

Since there are two non-zero singular values, the rank of this matrix is 2. The first two columns of $\mathbf{U}$ are a basis for the column space of $\mathbf{A}$ and the third column on $\mathbf{U}$ is a basis for the orthogonal complement of the column space of $\mathbf{A}$. We can compute projection matrices for these subspaces using (2.26). Because the bases from the SVD are orthonormal, the term that is inverted in (2.26) equals an identity matrix, and can be omitted. The projection matrix for the column space of $\mathbf{A}$ is

$$\begin{aligned} P_R &= \mathbf{U}_1\mathbf{U}_1^T \\ &= \begin{bmatrix} 0.4082 & -0.7071 \\ 0.8165 & 0.0000 \\ 0.4082 & 0.7071 \end{bmatrix} \begin{bmatrix} 0.4082 & 0.8165 & 0.4082 \\ -0.7071 & 0.0000 & 0.7071 \end{bmatrix} \\ &= \begin{bmatrix} 0.6667 & 0.3333 & -0.3333 \\ 0.3333 & 0.6667 & 0.3333 \\ -0.3333 & 0.3333 & 0.6667 \end{bmatrix}. \end{aligned}$$

This is the same projection matrix that was obtained in Example 2.10.

In a similar way we can use the SVD to calculate the projection matrix onto the orthogonal complement of the column space of $\mathbf{A}$. The result is

$$\begin{aligned} P_{R^\perp} &= \mathbf{U}_2\mathbf{U}_2^T \\ &= \begin{bmatrix} 0.5774 \\ -0.5774 \\ 0.5774 \end{bmatrix} \begin{bmatrix} 0.5774 & -0.5774 & 0.5774 \end{bmatrix} \\ &= \begin{bmatrix} 0.3333 & -0.3333 & 0.3333 \\ -0.3333 & 0.3333 & -0.3333 \\ 0.3333 & -0.3333 & 0.3333 \end{bmatrix} \end{aligned}$$

which is the same projection matrix that was obtained in Example 2.11.

■

SVD and Rank

Another important use of the SVD has to do with determining the rank of a matrix. The SVD shows if a small perturbation of a matrix $\mathbf{A}$ results in a matrix whose rank is less than that of $\mathbf{A}$. This can occur if the smallest nonzero singular value of $\mathbf{A}$ is much smaller that the largest singular value, as illustrated by the following example. Suppose $\mathbf{A}$ is a 2×2 matrix given by

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 1/\epsilon & 1 \end{bmatrix}$$

where ϵ is a small positive number. We note that the determinant of $\mathbf{A}$ is 1, so $\mathbf{A}$ is nonsingular (rank = 2 in this case). Now we add a very small matrix to $\mathbf{A}$ to get

$$\tilde{\mathbf{A}} = \mathbf{A} + \begin{bmatrix} 0 & \epsilon \\ 0 & 0 \end{bmatrix}.$$

Note that the determinant of $\tilde{\mathbf{A}} = 0$ so $\tilde{\mathbf{A}}$ is singular (rank =1), even though it is just a small perturbation away from a matrix of rank 2. The singular values of $\mathbf{A}$ show that $\mathbf{A}$ is close to a matrix of rank 1. The singular values of $\mathbf{A}$ can be computed to give

$$\sigma_1^2 = 1 + \frac{1 + \sqrt{1 + 4\epsilon^2}}{2\epsilon^2}, \quad \sigma_2^2 = 1 + \frac{1 - \sqrt{1 + 4\epsilon^2}}{2\epsilon^2}.$$

As ϵ becomes small, the first singular value becomes large, while the second singular value goes to zero. For instance, when ϵ equals 0.001, $\sigma_1 = 1,000$ and $\sigma_2 = 10^{-6}$. Thus the matrix $\mathbf{A}$ has only one singular value significantly different from zero, and so $\mathbf{A}$ is very close to a matrix of rank 1.

2.8 Linear Equations

Consider a set of m linear equations in n unknowns written in matrix form as

$$\underbrace{\mathbf{A}}_{m \times n} \underbrace{\mathbf{x}}_{n \times 1} = \underbrace{\mathbf{y}}_{m \times 1}. \tag{2.35}$$

In this equation, $\mathbf{A}$ and $\mathbf{y}$ are known and $\mathbf{x}$ is unknown. It is useful to define a matrix $\mathbf{W}$ which consists of the matrix $\mathbf{A}$ augmented with the vector $\mathbf{y}$; that is, $\mathbf{W}$ is the $m \times (n+1)$ matrix given by

$$\mathbf{W} \stackrel{\text{def}}{=} [\mathbf{A}\ \mathbf{y}]. \tag{2.36}$$

In the next two sections, we characterize solutions to (2.35).

2.8.1 Characterization by Rank

If $\mathbf{y} \in \text{col}(\mathbf{A})$ then there exists a vector of expansion coefficients $\mathbf{x}$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$. In other words, $\mathbf{y} \in \text{col}(\mathbf{A})$ is a sufficient condition for (2.35) to have a solution. It is also a necessary condition because if $\mathbf{y} \notin \text{col}(\mathbf{A})$, this means that $\mathbf{y}$ cannot be expressed as a linear combination of the columns of $\mathbf{A}$. In this case there does not exist a vector $\mathbf{x}$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$.

A test for whether or not $\mathbf{y} \in \text{col}(\mathbf{A})$ can be expressed in terms of the ranks of the matrices $\mathbf{A}$ and $\mathbf{W}$ ($\mathbf{W}$ is defined in (2.36)). The rank of $\mathbf{A}$, denoted $\rho(\mathbf{A})$, is the number of linearly independent columns of $\mathbf{A}$. The matrix $\mathbf{W}$ contains the columns of $\mathbf{A}$ plus one additional column, the vector $\mathbf{y}$. There are only two ways that $\rho(\mathbf{W})$ and $\rho(\mathbf{A})$ can be related. Either $\rho(\mathbf{W}) = \rho(\mathbf{A}) + 1$, which means that $\mathbf{y}$ is linearly independent of the columns of $\mathbf{A}$, or $\rho(\mathbf{W}) = \rho(\mathbf{A})$, which means that $\mathbf{y}$ is linearly dependent on the columns of $\mathbf{A}$.

The discussion in the previous two paragraphs can be summarized as follows

$$\rho(\mathbf{W}) = \rho(\mathbf{A}) \Longrightarrow \mathbf{y} \in \text{col}(\mathbf{A}) \Longrightarrow \text{(2.35) has a solution.}$$

$$\rho(\mathbf{W}) \neq \rho(\mathbf{A}) \Longrightarrow \mathbf{y} \notin \text{col}(\mathbf{A}) \Longrightarrow \text{(2.35) does not have a solution.}$$

These results characterize the existence of solutions for linear equations. A complete characterization of the existence and uniqueness of solutions is given in the following fact.

Fact 2.41 *1. If $\rho(\mathbf{W}) \neq \rho(\mathbf{A})$ then no solutions exist.*

2. If $\rho(\mathbf{W}) = \rho(\mathbf{A})$ then at least one solution exists.

(a) If $\rho(\mathbf{W}) = \rho(\mathbf{A}) = n$, then there is a unique *solution for $\mathbf{x}$.*

(b) If $\rho(\mathbf{W}) = \rho(\mathbf{A}) = r < n$, then there is an infinite set of solution vectors parameterized by $n - r$ free variables.

3. If $\rho(\mathbf{A}) = m$, a solution exists for every possible right hand side vector $\mathbf{y}$. This case can only occur if $m \leq n$. From 2., the solution is unique if $m = n$ and non-unique if $m < n$.

While Fact 2.41 is theoretically complete, it falls short in two areas: first, it doesn't tell you *how to find solutions, or how to parameterize an infinite set of solutions*, and second, it does not give any information about obtaining *approximate solutions in cases when an exact solution does not exist*. We address these two areas in the following sections. We begin by characterizing solutions in terms of the dimensions of the coefficient matrix.

2.8.2 Characterization by Matrix Dimension

We consider three cases: when $m = n$ and the matrix $\mathbf{A}$ is square, when $m > n$ so that the matrix $\mathbf{A}$ has more rows than columns, and finally the case when $m < n$ so that the matrix $\mathbf{A}$ has more columns than rows. We first characterize the types of solutions that can occur with these three cases using the two conditions in Fact 2.41. Then we show how solutions can be calculated.

1. If $m = n$ (square system of equations), then:

   (a) If $\det(\mathbf{A}) \neq 0$ then a unique solution exists and is given by

   $$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}.$$

   The reasoning for this case goes as follows: $\det(\mathbf{A})$=n implies by the Gram test that the columns of $\mathbf{A}$ are linearly independent, and so $\rho(\mathbf{A}) = n$. Then $\rho(\mathbf{W})$

can't be larger than n because $\mathbf{W}$ has only n rows, and $\rho(\mathbf{W})$ can't be smaller than n because the n columns of $\mathbf{A}$ (which are known to be independent) are also columns of $\mathbf{W}$. Thus this case fits into case 2(a) of the theoretical characterization given in the previous subsection.

(b) If det($\mathbf{A}$)=0 then solutions do not exist for an arbitrary vector $\mathbf{y}$. However, if $\mathbf{y}$ is such that $\rho(\mathbf{W}) = \rho(\mathbf{A}) = r < n$, then an infinite number of solutions exist. (Note that r will be less than n in this case because det($\mathbf{A}$) $= 0$.)

2. If $m > n$ (overdetermined system of equations)

(a) The most common case of overdetermined equations occurs when $\rho(\mathbf{A}) = n$. In this case the unique least-squares solution (the one that minimizes the norm of $\mathbf{y} - \mathbf{A}\mathbf{x}$ over all possible $\mathbf{x}$) is given by projecting $\mathbf{y}$ onto the column space of $\mathbf{A}$, and the solution will shown in the next section to be given by

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}.$$

(b) If $\rho(\mathbf{A}) < n$, then solutions exist if and only if $\rho(\mathbf{A}) = \rho(\mathbf{W}) = r$ (where r is some number less than n). In this case, there are an infinite number of solutions parameterized by $n - r$ variables. This case is examined later using the SVD.

3. If $m < n$ (underdetermined system of equations)

(a) The most common case of underdetermined equations occurs when $\rho(\mathbf{A}) = m$. In this case an infinite number of solutions exists, but we can obtain the unique solution which has the smallest norm of any solution vector by projecting any solution onto the row space of $\mathbf{A}$. The solution will be shown to be given by

$$\mathbf{x}_{\mathrm{min-norm}} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{y}.$$

(b) If $\rho(\mathbf{A}) < m$, then solutions may or may not exist, depending on whether or not $\rho(\mathbf{W})$ equals $\rho(\mathbf{A})$. This case is examined using the SVD later in this section.

2.8.3 Least Squares and Min-Norm Solutions – Full Rank Case

2.8.3.1 Least-Squares Solutions Consider the case when $\rho(\mathbf{W}) \neq \rho(\mathbf{A})$ so that no solutions exist to the equation $\mathbf{A}\mathbf{x} = \mathbf{y}$. In this case, we would like to find the vector $\mathbf{x}_{\mathrm{LS}}$ which minimizes the squared norm between $\mathbf{y}$ and $\mathbf{A}\mathbf{x}$ for any vector $\mathbf{x}$. In other words, we would like to solve the following minimization problem

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2. \tag{2.37}$$

Since the vector $\mathbf{A}\mathbf{x}$ is an element of col($\mathbf{A}$) then an equivalent problem is to find the vector in col($\mathbf{A}$) which is the best approximation to a given vector $\mathbf{y}$. We know the answer to this second problem: the best approximation is obtained by projecting $\mathbf{y}$ onto the subspace col($\mathbf{A}$). See Fig. 2.3 on page 52. The result of the projection is the vector $\mathbf{y}_p$.

Once $\mathbf{y}_p$ is known then any vector $\mathbf{x}_{\mathrm{LS}}$ which satisfies

$$\mathbf{A}\mathbf{x}_{\mathrm{LS}} = \mathbf{y}_p \tag{2.38}$$

will minimize the squared error in (2.37) We know that the above equation has a solution because $\mathbf{y}_p \in$ col($\mathbf{A}$) which implies that $\rho(\mathbf{W}) = \rho(\mathbf{A})$. If we make the further

assumption that $\rho(\mathbf{A}) = n$ (this is the most common case), then by 2(a) of section 4.1, the least-squares solution will be unique, and the matrix which projects onto col$(\mathbf{A})$ can be constructed using (2.26). In other words,

$$\mathbf{y}_p = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$$

and we now solve for the least-squares solution from

$$\mathbf{A}\mathbf{x}_{\text{LS}} = \mathbf{y}_p = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}.$$

The above equation can be solved by multiplying both sides on the left by $\mathbf{A}^T$ and then by $(\mathbf{A}^T\mathbf{A})^{-1}$ to obtain

$$\mathbf{x}_{\text{LS}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}. \tag{2.39}$$

This is the unique least-squares solution when the columns of $\mathbf{A}$ are linearly independent (so that the rank of $\mathbf{A}$ is n). In order for $\mathbf{A}$ to have rank n, it is necessary that $m \geq n$. If $m = n$ then the right-hand side of (2.39) reduces to $\mathbf{A}^{-1}\mathbf{y}$ and the least-squares solution gives an error norm of zero in (2.37).

When the columns of $\mathbf{A}$ are dependent (an uncommon case), the least squares solution is not unique, and solutions can be calculated using the SVD as shown later in this section.

EXAMPLE 2.14

Consider the linear equations $\mathbf{A}\mathbf{x} = \mathbf{y}$ where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Using (2.39) we compute the least-squares solution (to 4 decimal places) to be

$$\mathbf{x}_{\text{LS}} = \begin{bmatrix} 0.6667 \\ 0.6667 \end{bmatrix}.$$

Equation (2.38) says that $\mathbf{A}\mathbf{x}_{\text{LS}}$ should equal $\mathbf{y}_p$, the projection of $\mathbf{y}$ onto the column space of $\mathbf{A}$. We have previously computed this projection in Example 2.10, and the result is

$$\mathbf{y}_p = \begin{bmatrix} 2/3 \\ 4/3 \\ 2/3 \end{bmatrix}.$$

We get the same result by multiplying $\mathbf{A}\mathbf{x}_{\text{LS}}$ as shown below (to 4 decimal places)

$$\mathbf{A}\mathbf{x}_{\text{LS}} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0.6667 \\ 0.6667 \end{bmatrix} = \begin{bmatrix} 0.6667 \\ 1.3333 \\ 0.6667 \end{bmatrix}.$$

■

2.8.3.2 Min-Norm Solutions We now turn our attention to underdetermined equations ($m < n$) and min-norm solutions in the case when the matrix $\mathbf{A}$ has rank m (that is, its rows are linearly independent). From 3 of Fact 2.41 we know that a solution exists for every possible right hand side vector $\mathbf{y}$. In addition, we know that there will be an infinite

number of solutions, since $\rho(\mathbf{A}) = m < n$. We would like to find the solution which has the minimum norm. A method for parameterizing all solutions using the SVD will be given later in this section.

In order to find the min-norm solution, it is useful to consider the row-space of $\mathbf{A}$, denoted row$(\mathbf{A})$, and its orthogonal complement, denoted row$(\mathbf{A})^\perp$. Recall that a subspace and its orthogonal complement form a direct sum decomposition of $\mathbf{R}^n$, so that

$$\mathbf{R}^n = \text{row}(\mathbf{A}) \oplus \text{row}(\mathbf{A})^\perp.$$

Now a solution to $\mathbf{Ax} = \mathbf{y}$ will be a vector $\mathbf{x} \in \mathbf{R}^n$, and so $\mathbf{x}$ can be decomposed uniquely into its projection onto row$(\mathbf{A})$ and its projection onto row$(\mathbf{A})^\perp$ as follows

$$\mathbf{x} = \mathbf{x}_{\mathbf{A}} + \mathbf{x}_{\mathbf{A}^\perp}. \tag{2.40}$$

Thus for any solution $\mathbf{x}$ we can write

$$\begin{aligned} \mathbf{Ax} &= \mathbf{y} \\ \mathbf{A}(\mathbf{x}_A + \mathbf{x}_{A^\perp}) &= \mathbf{y} \\ \mathbf{Ax}_A &= \mathbf{y}. \end{aligned}$$

We used the fact that $\mathbf{Ax}_{A^\perp} = 0$ since vectors in row$(\mathbf{A})^\perp$ are orthogonal to rows of $\mathbf{A}$. We now use the fact that $\mathbf{x}_A$ is in row$(\mathbf{A})$, and so it has a representation as a linear combination of the rows of $\mathbf{A}$. Since the rows of $\mathbf{A}$ are the columns of $\mathbf{A}^T$, we can write this representation as

$$\mathbf{x}_A = \mathbf{A}^T\boldsymbol{\alpha}. \tag{2.41}$$

If we multiply the above equation by $\mathbf{A}$ and use the fact that $\mathbf{Ax}_A = \mathbf{y}$ we get

$$\mathbf{Ax}_A = \mathbf{AA}^T\boldsymbol{\alpha} = \mathbf{y}.$$

Now we notice that $\mathbf{AA}^T$ is the Gram matrix for the *rows of* $\mathbf{A}$, which are assumed to be linearly independent, so we can multiply through by $(\mathbf{AA}^T)^{-1}$ to obtain the unique solution for $\boldsymbol{\alpha}$

$$\boldsymbol{\alpha} = (\mathbf{AA}^T)^{-1}\mathbf{y}.$$

and using (2.41) we get that

$$\mathbf{x}_A = \mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{y}$$

where $\mathbf{x}_A$ is the unique vector in row$(\mathbf{A})$ which satisfies $\mathbf{Ax}_A = \mathbf{y}$. Notice that $\mathbf{y} = \mathbf{Ax}$ so that the above equation can be written as

$$\mathbf{x}_A = \mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{Ax}$$

where the matrix $\mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{A}$ is *the orthogonal projection matrix onto row*$(\mathbf{A})$*, the row-space of the matrix* $\mathbf{A}$ (compare with the formulas for projection onto the space formed from column vectors in Section 2.5.2). We can interpret the above equation as saying that $\mathbf{x}_A$ is obtained by *projecting* any solution $\mathbf{x}$ onto row$(\mathbf{A})$.

Recall from (2.40) that any solution can be expressed as $\mathbf{x} = \mathbf{x}_A + \mathbf{x}_{A^\perp}$. However, since $\mathbf{x}_A$ is orthogonal to $\mathbf{x}_{A^\perp}$ it is easy to show that

$$\|\mathbf{x}\|^2 = \|\mathbf{x}_A\|^2 + \|\mathbf{x}_{A^\perp}\|^2 \geq \|\mathbf{x}_A\|^2$$

and that the solution of minimum norm occurs when we choose $\mathbf{x} = \mathbf{x}_A$. In other words,

$$\mathbf{x}_{\min-\text{norm}} = \mathbf{x}_A = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{y}. \tag{2.42}$$

Finally, notice that $\text{row}(\mathbf{A})^\perp$ is just the null-space of $\mathbf{A}$, which is of dimension $n - m$ when $\mathbf{A}$ has rank m. Thus we can find $n - m$ vectors which are a basis for $\text{row}(\mathbf{A})^\perp$. Any linear combination of these vectors can be added to $\mathbf{x}_{\min-\text{norm}}$ to generate other solutions to $\mathbf{A}\mathbf{x} = \mathbf{y}$.

EXAMPLE 2.15

Consider the equation $\mathbf{A}\mathbf{x} = \mathbf{y}$ where

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}.$$

The min-norm solution is computed using (2.42) as follows (results shown to 4 decimal places)

$$\mathbf{x}_{\min-\text{norm}} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{y} = \begin{bmatrix} 0.6667 & -0.3333 \\ 0.3333 & 0.3333 \\ -0.3333 & 0.6667 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 0.6667 \\ 2.3333 \\ 1.6667 \end{bmatrix}.$$

It can be verified that $\mathbf{A}\mathbf{x}_{\min-\text{norm}}$ does in fact equal $\mathbf{y}$, but there are other solutions to the equation. From Examples 2.10 and 2.11 we know that a basis for the null-space of $\mathbf{A}$ is

$$\mathbf{v} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}.$$

Thus all solutions to $\mathbf{A}\mathbf{x} = \mathbf{y}$ can be written as

$$\mathbf{x} = \mathbf{x}_{\min-\text{norm}} + \alpha\mathbf{v}$$

for some value of α. Notice that $\mathbf{x}_{\min-\text{norm}} \perp \mathbf{v}$ so that

$$\|\mathbf{x}\|^2 = \|\mathbf{x}_{\min-\text{norm}} + \alpha\mathbf{v}\|^2 = \|\mathbf{x}_{\min-\text{norm}}\|^2 + \alpha^2\|\mathbf{v}\|^2 \geq \|\mathbf{x}_{\min-\text{norm}}\|^2.$$

■

2.8.4 Using SVD to Solve Linear Equations

The SVD was introduced in Section 2.7. The SVD provides orthonormal bases for the four fundamental subspaces of a matrix, and these subspaces contain the information necessary to solve linear equations in any situation: unique solution, multiple solutions, least-squares solutions, min-norm solutions, etc. Thus the SVD can be used to solve all of the types of linear equations mentioned in the previous section! A description of how this is done is given in Table 2.6.

Consider the equation $\mathbf{Ax} = \mathbf{y}$ where the matrix $\mathbf{A}$ is $m \times n$. Let the SVD of $\mathbf{A}$ be partitioned as follows

$$\mathbf{A} = [\,\mathbf{U}_1 \quad \mathbf{U}_2\,] \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix}.$$

The $r \times r$ matrix $\mathbf{\Sigma}_1$ contains the r non-zero singular values of $\mathbf{A}$. The matrices $\mathbf{U}_1$ and $\mathbf{V}_1$ each contain r columns. If $r = m$ then $\mathbf{U}_2$ is not present (all the columns of $\mathbf{U}$ appear in $\mathbf{U}_1$), and if $r = n$ then $\mathbf{V}_2$ is not present.
If $\mathbf{y} = \mathbf{0}$ the equation of interest reduces to $\mathbf{Ax} = \mathbf{0}$ and we are interested in the null space of $\mathbf{A}$. Since $\mathbf{V}_2$ provides a basis for the null-space of $\mathbf{A}$ all solutions can be written as

$$\mathbf{x} = \mathbf{V}_2\boldsymbol{\alpha}$$

where $\boldsymbol{\alpha}$ is a vector containing $n - r$ free parameters.
If $\mathbf{y} \neq \mathbf{0}$ we compute the following vector

$$\mathbf{z} = \mathbf{V}_1\mathbf{\Sigma}_1^{-1}\mathbf{U}_1^T\mathbf{y}.$$

The vector $\mathbf{z}$ has the following properties.

1. If $\mathbf{Ax} = \mathbf{y}$ has a unique solution, $\mathbf{z}$ is the solution.
2. If $\mathbf{Ax} = \mathbf{y}$ does not have a solution, $\mathbf{z}$ gives a least-squares solution. If the least-squares solution is not unique, $\mathbf{z}$ is the least-squares solution with minimum norm. Other least-squares solutions can be generated by adding vectors from the null-space of $\mathbf{A}$ to $\mathbf{z}$. All solutions can be written as

   $$\mathbf{x} = \mathbf{z} + \mathbf{V}_2\boldsymbol{\alpha}$$

   where $\boldsymbol{\alpha}$ is a vector containing $n - r$ free parameters.
3. If $\mathbf{Ax} = \mathbf{y}$ has an infinite number of solutions, $\mathbf{z}$ is the min-norm solution. Other solutions can be generated by adding vectors from the null-space of $\mathbf{A}$ to $\mathbf{z}$ as shown in the equation in item 2.

Table 2.6 Using SVD to solve linear equations.

EXAMPLE 2.16

In this example we use the SVD to compute all least-squares solutions to a rank-deficient least-squares problem. Consider the following matrix:

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

MATLAB gives the singular value decomposition of $\mathbf{A}$ as follows

```
>> [u,s,v]=svd(A)

u =

    0.2764   -0.7236    0.5499   -0.3124
    0.4472    0.4472    0.6243    0.4586
    0.7236   -0.2764   -0.5499    0.3124
   -0.4472   -0.4472    0.0744    0.7710

s =

    2.6900         0         0         0
         0    1.6625         0         0
         0         0         0         0
         0         0         0         0

v =

    0.3717   -0.6015    0.7071    0.0000
   -0.3717    0.6015    0.7071   -0.0000
    0.6015    0.3717   -0.0000   -0.7071
    0.6015    0.3717   -0.0000    0.7071
```

Since there are 2 non-zero singular values, the rank of $\mathbf{A}$ is 2. Furthermore, since $\mathbf{A}$ has 4 columns, the nullity of $\mathbf{A}$ is also 2 (see Fact 2.28). Suppose we want to parameterize all least-squares solutions to $\mathbf{Ax} = \mathbf{y}$ where

$$\mathbf{y} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}.$$

Since the columns of $\mathbf{A}$ are *not* linearly independent, the full-rank solution in (2.39) cannot be used. However using the formula in Table 2.6 we can compute

$$\begin{aligned}\mathbf{z} &= \mathbf{V}_1\boldsymbol{\Sigma}_1^{-1}\mathbf{U}_1^T\mathbf{y} \\ &= \begin{bmatrix} 0.3717 & -0.6015 \\ -0.3717 & 0.6015 \\ 0.6015 & 0.3717 \\ 0.6015 & 0.3717 \end{bmatrix} \begin{bmatrix} 2.6900 & 0 \\ 0 & 1.6625 \end{bmatrix}^{-1} \begin{bmatrix} 0.2764 & 0.4472 & 0.7236 & -0.4472 \\ -0.7236 & 0.4472 & -0.2764 & -0.4472 \end{bmatrix} \mathbf{y} \\ &= \begin{bmatrix} 1.1 \\ -1.1 \\ -0.2 \\ -0.2 \end{bmatrix}.\end{aligned}$$

The vector $\mathbf{z}$ is the least-squares solution of minimum norm. All least-squares solutions to $\mathbf{Ax} = \mathbf{y}$ can be written as

$$\mathbf{x}_{\mathrm{LS}} = \mathbf{z} + \mathbf{V}_2\boldsymbol{\alpha}$$

where $\boldsymbol{\alpha}$ is a vector of two coefficients and

$$\mathbf{V}_2 = \begin{bmatrix} 0.7071 & 0.0000 \\ 0.7071 & -0.0000 \\ -0.0000 & -0.7071 \\ -0.0000 & 0.7071 \end{bmatrix}.$$

Notice that

$$\|\mathbf{Az} - \mathbf{y}\|^2 = \|\mathbf{A}(\mathbf{z} + \mathbf{V}_2\boldsymbol{\alpha}) - \mathbf{y}\|^2$$

since $\mathbf{AV}_2 = \mathbf{0}$.

■

2.9 Chapter Summary

In this chapter we presented results from linear algebra and matrix theory which are used in the design of state-space control systems. We developed the concept of linear independence of a set of vectors and used it to define the rank of a matrix as the number of linearly independent rows or columns. Simple, but useful, interpretations of matrix multiplication were shown.

We introduced the four fundamental subspaces of a matrix: the row space, the column space, and their orthogonal complements. Important facts about the eigenvalues and eigenvectors of a square matrix were presented. We characterized the solution of simultaneous linear equations. Conditions for existence and uniqueness of solutions were established. We showed how to compute min-norm solutions when solutions are not unique, and least-squares approximations when no solutions exist. We also showed how the singular value decomposition could be used to solve any type of linear equation.

2.10 Problems

1. Indicate whether the following statements are true or false. If a statement is true, give the reason why it is true. (Use any Fact from Chapter 2.) If a statement is false, give an example in $\Re^2$ showing that it is false.

(a) If two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ are linearly independent then they must be orthogonal.

(b) Let $\mathbf{A}$ be a matrix whose columns are $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, i.e.

$$\mathbf{A} = [\,\mathbf{x}_1 \quad \mathbf{x}_2 \quad \mathbf{x}_3\,].$$

If the rank of $\mathbf{A}$ is 2, then $\mathbf{x}_1$ and $\mathbf{x}_2$ must be linearly independent.

(c) For the matrix $\mathbf{A}$ given in part (b), if the rank of $\mathbf{A}$ is 2, then any two linearly independent columns are a basis for the column space of $\mathbf{A}$.

2. Given p linearly independent vectors $\mathbf{x}_1, \cdots, \mathbf{x}_p$ and p other vectors $\mathbf{y}_1, \cdots, \mathbf{y}_p$, form vectors $\mathbf{z}_i$ by concatenating the vectors $\mathbf{x}_i$ and $\mathbf{y}_i$ as follows:

$$\mathbf{z}_i = \begin{bmatrix} \mathbf{x}_i \\ \mathbf{y}_i \end{bmatrix}, \quad i = 1, \cdots, p.$$

Show that the vectors $\mathbf{z}_1, \cdots, \mathbf{z}_p$ are linearly independent regardless of the values of $\mathbf{y}_1, \cdots, \mathbf{y}_p$.

3. Suppose you are given a matrix $\mathbf{A}_{m\times n}$ with $m > n$ and you are told that the rank of $\mathbf{A}$ is n.

(a) Does there exist a matrix $\mathbf{B}$ such that $\mathbf{AB} = \mathbf{I}_{m\times m}$? If yes, give a formula for $\mathbf{B}$ in terms of $\mathbf{A}$; if not, state why not.

(b) Does there exist a matrix $\mathbf{B}$ such that $\mathbf{BA} = \mathbf{I}_{n\times n}$? If yes, give a formula for $\mathbf{B}$ in terms of $\mathbf{A}$; if not, state why not.

4. Given $\mathbf{X}$ and $\mathbf{y}$ below, show that $\mathbf{y} \in$ col($\mathbf{X}$). Find the expansion coefficients that express $\mathbf{y}$ as a linear combination of columns of $\mathbf{X}$.

$$\mathbf{X} = \begin{bmatrix} 1 & -1 \\ 2 & 0 \\ 1 & 2 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 2 \\ 6 \\ 5 \end{bmatrix}$$

5. Suppose it is true that $\mathbf{Ax} = \mathbf{Ay}$. Under what conditions can you conclude that $\mathbf{x} = \mathbf{y}$?

6. Under what conditions could it be that $\mathbf{x} \neq \mathbf{y}$ but $\mathbf{Ax} = \mathbf{Ay}$?

7. Given $\mathbf{x} \in$ row($\mathbf{A}$) and $\mathbf{y} \in$ null($\mathbf{A}$), show that $\mathbf{x}^T\mathbf{y} = 0$.

8. Let $\mathbf{x}_1, \cdots, \mathbf{x}_m$ be a basis for an m-dimensional subspace X of $\mathbf{R}^n$. Let $\mathbf{y}_1, \cdots, \mathbf{y}_{n-m}$ be a basis for the orthogonal complement of X, $X^\perp = \{\mathbf{y}$ s.t. $\mathbf{y}^T\mathbf{x}_i = 0,\ i = 1, \cdots, m\}$. Let $\mathbf{P}_X$ be the $n \times n$ orthogonal projection matrix onto X.

(a) What does $\mathbf{P}_X\mathbf{x}_i$ equal for $i = 1, \cdots, m$?

(b) What does $\mathbf{P}_X\mathbf{y}_i$ equal for $i = 1, \cdots, n - m$?

(c) $\mathbf{P}_X$ has n eigenvalues. Show using (a) and (b) that the eigenvalues will either equal zero or one. How many eigenvalues will be 1 and how many will be 0?

9. Let $\mathbf{P}_X$ be the orthogonal projection matrix defined in the previous problem. Use (2.26) to show that $\mathbf{P}_X$ is a symmetric matrix, and also that $\mathbf{P}_X$ is idempotent ($\mathbf{P}_X\mathbf{P}_X = \mathbf{P}_X$).

10. Let $\mathbf{x}_1, \cdots, \mathbf{x}_m$ be linearly independent vectors in $\mathbf{R}^n$ with $m < n$, and let $\mathbf{A} = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_m]$.

    (a) Is $\mathbf{A}^T\mathbf{A}$ an invertible matrix? Why or why not.

    (b) Is $\mathbf{A}\mathbf{A}^T$ an invertible matrix? Why or why not.

11. Given a $p \times r$ matrix $\mathbf{A}$ and an $r \times q$ matrix $\mathbf{B}$, show that

$$\text{rank}(\mathbf{AB}) \leq \min\{\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B})\}.$$

    (*Hint:* Use Facts 2.16 and 2.17 and the sentences following them, as well as Definition 2.25.)

12. Let $\mathbf{w}_i$ be the i-th column of the matrix $\mathbf{W}$ shown below

$$\mathbf{W} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 \end{bmatrix}.$$

    (a) What is the dimension of the subspace $S = \text{col}(\mathbf{W})$?

    (b) Find a basis for S whose elements are columns of W.

    (c) Find a basis for S whose elements are columns of W and is different from the basis found in (b).

13. State whether the following statements are true or false and give an explanation for your answer.

    (a) Given $\mathbf{A}_{m\times n}$, if the rank of $\mathbf{A} = n$ then the only vector $\mathbf{x}$ such that $\mathbf{Ax} = \mathbf{0}$ is $\mathbf{x} = \mathbf{0}$.

    (b) Given 3 non-zero vectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, if $\mathbf{Ax}_1 = \mathbf{0}, \mathbf{Ax}_2 = \mathbf{0}$, and $\mathbf{Ax}_3 = \mathbf{0}$ then the nullity of $\mathbf{A}$ is greater than or equal to 3.

    (c) Let $\mathbf{x}$ and $\mathbf{y}$ be n-vectors. If $\mathbf{A} = \mathbf{xy}^T$ then $\mathbf{x}$ is an eigenvector of $\mathbf{A}$.

    (d) If 3 linearly independent vectors $\mathbf{x}, \mathbf{y}$, and $\mathbf{z}$ are elements of a vector space S then the dimension of S is greater than or equal to 3.

14. Suppose that $\mathbf{x}_1, \cdots, \mathbf{x}_k$ are basis vectors for a subspace S of $\Re^n$. These vectors can be placed as columns of a matrix $\mathbf{X}$. Let $\mathbf{T}$ be a $k \times k$ nonsingular matrix and let

$$\mathbf{Y} = \mathbf{XT}.$$

    (a) Show that the columns of $\mathbf{Y}$ are linearly independent. (Hint: form the Gram matrix and use properties of determinants.)

    (b) Calculate the projection matrix for the column space of $\mathbf{Y}$. Show that this projection matrix is equal to the projection matrix for the column space of $\mathbf{X}$.

15. Consider the following vectors

$$\mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \quad \mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

Let S be the subspace generated by $\mathbf{v}_1$ and $\mathbf{v}_2$.

(a) Show that $\mathbf{x}$ is not in the subspace S.

(b) Compute $\mathbf{y}$ = the projection of $\mathbf{x}$ onto S.

(c) Show that $\mathbf{y}$ is in S.

(d) Let $\mathbf{e} = \mathbf{x} - \mathbf{y}$. Show that $\mathbf{e}$ is orthogonal to S.

(e) Express $\mathbf{y}$ as a linear combination of the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$.

16. Given

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix}, \quad \mathbf{y}_1 = \begin{bmatrix} 6 \\ 9 \\ 12 \end{bmatrix}, \quad \mathbf{y}_2 = \begin{bmatrix} 5 \\ 6 \\ 9 \end{bmatrix}$$

(a) Show that $\mathbf{Ax} = \mathbf{y}_1$ has an infinite number of solutions parameterized by 1 free variable. Choose two different values of this variable, and generate two different solution vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. Show by direct substitution that $\mathbf{Ax}_i = \mathbf{y}_1,\ i = 1, 2$.

(b) Show that $\mathbf{Ax} = \mathbf{y}_2$ does not have any solutions, but it does have a 1-parameter family of least-squares solutions. Calculate the min-norm solution $\mathbf{x}_1$ and any other solution $\mathbf{x}_2$ and show that they yield the same error: $\|\mathbf{Ax}_1 - \mathbf{y}_2\| = \|\mathbf{Ax}_2 - \mathbf{y}_2\|$.

17. Consider

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 4 \\ 7 \end{bmatrix}.$$

(a) Calculate the min-norm solution.

(b) Generate another solution $\mathbf{x}_1$ by adding a non-zero vector from the null-space of $\mathbf{A}$ to $\mathbf{x}_{min-norm}$. Show that $\|\mathbf{x}_1\| > \|\mathbf{x}_{min-norm}\|$.

18. Consider

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

Show that a 2-parameter family of solutions exists for the above equation. Find 2 different solutions.

19. Let $\mathbf{A}$ be an $m \times p$ matrix and $\mathbf{B}$ be a $p \times m$ matrix. Suppose $\mathbf{x}$ is an eigenvector of $\mathbf{AB}$ corresponding to the non-zero eigenvalue λ. Show that λ is also an eigenvalue of $\mathbf{BA}$, and find the corresponding eigenvector. (*Hint*: Use the definition of an eigenvector given in (2.28).)

20. Show that if λ is an eigenvalue of $\mathbf{A}$, then $\alpha + \lambda$ is an eigenvalue of $\alpha\mathbf{I} + \mathbf{A}$.

21. (a) Show that the eigenvalues of a triangular matrix are equal to the elements on the main diagonal. (*Hint:* Use Fact 2.32 as well as a property of determinants given in Table 2.5 on page 45.)

(b) Show that the eigenvalues of a block triangular matrix are equal to the union of the eigenvalues of the matrices on the main diagonal. (*Hint:* Use the hint given for part (a).)

22. Test the following sets of vectors for linear dependence. If a set is linearly dependent, express one of the vectors in terms of the others in that set.

    (a)

$$\begin{bmatrix}1\\0\end{bmatrix}, \quad \begin{bmatrix}1\\1\end{bmatrix}, \quad \begin{bmatrix}3\\4\end{bmatrix}$$

    (b)

$$\begin{bmatrix}1\\0\\0\end{bmatrix}, \quad \begin{bmatrix}1\\1\\0\end{bmatrix}, \quad \begin{bmatrix}1\\1\\1\end{bmatrix}$$

    (c)

$$\begin{bmatrix}1\\0\\1\end{bmatrix}, \quad \begin{bmatrix}0\\1\\1\end{bmatrix}, \quad \begin{bmatrix}2\\1\\1\end{bmatrix}$$

23. What is the dimension of the subspace generated by the sets of vectors in the previous problem? Give a basis for each subspace.

24. Show that, for any $m \times n$ matrix $\mathbf{A}$, the rank of $\mathbf{A}$ plus the nullity of $\mathbf{A}^T$ equals m. (*Hint*: use the definitions of rank and nullity.)

25. Using the matrix $\mathbf{A}$ and the vector $\mathbf{y}$ shown below, decompose $\mathbf{y}$ as $\mathbf{y} = \mathbf{y}_A + \mathbf{y}_{A^\perp}$ where $\mathbf{y}_A \in \text{col}(\mathbf{A})$ and $\mathbf{y}_{A^\perp} \in$ orthogonal complement of col($\mathbf{A}$).

$$\mathbf{A} = \begin{bmatrix}1 & 1\\1 & 0\\1 & 1\end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix}1\\2\\3\end{bmatrix}$$

26. This problem shows how the singular value decomposition of a matrix $\mathbf{A}$ is related to an eigenvalue decomposition of the "squared" matrices $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$.

    (a) Show that if $\mathbf{u}$ is a left singular vector of $\mathbf{A}$ then it is an eigenvector of $\mathbf{A}\mathbf{A}^T$.

    (b) Show that if $\mathbf{v}$ is a right singular vector of $\mathbf{A}$ then it is an eigenvector of $\mathbf{A}^T\mathbf{A}$.

    (c) Show that if σ is a nonzero singular value of $\mathbf{A}$ then $\lambda = \sigma^2$ is a nonzero eigenvalue of $\mathbf{A}\mathbf{A}^T$ or $\mathbf{A}^T\mathbf{A}$.

27. This problem considers some special properties of eigenvectors and singular vectors of symmetric matrices. Let $\mathbf{A}$ be a (square) symmetric matrix so that $\mathbf{A} = \mathbf{A}^T$.

    (a) Let $\mathbf{u}_1$ and $\mathbf{u}_2$ be eigenvectors of $\mathbf{A}$ corresponding to distinct eigenvalues λ_1 and λ_2 ($\lambda_1 \neq \lambda_2$). Show that $\mathbf{u}_1$ and $\mathbf{u}_2$ are orthogonal. (Hint: show that $\lambda_2\mathbf{u}_2^T\mathbf{u}_1 = \lambda_1\mathbf{u}_2^T\mathbf{u}_1$.)

    (b) Let $\mathbf{A}$ be a symmetric matrix with distinct eigenvalues. The eigenvalue decomposition of $\mathbf{A}$ can always be written as $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$. However, because of the result in part (a), the eigenvalue decomposition of a symmetric matrix can be written as

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}.$$

i. Show that the left singular vectors of $\mathbf{A}$ can be chosen to be the columns of $\mathbf{U}$.

ii. Show that the right singular vectors of $\mathbf{A}$ can be chosen to be the columns of $\mathbf{U}$.

(Hint: Use the fact that the left singular vectors of $\mathbf{A}$ are eigenvectors of $\mathbf{A}\mathbf{A}^T$ and right singular vectors of $\mathbf{A}$ are eigenvectors of $\mathbf{A}^T\mathbf{A}$.)

28. Given the vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ shown below, construct the matrix $\mathbf{P}$ which projects onto the subspace generated by $\mathbf{x}_1, \mathbf{x}_2$. First calculate $\mathbf{P}$ by definition, then compare with the result obtained using the SVD.

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \ , \ \mathbf{x}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

29. Let S_1 be the subspace generated by $\mathbf{x}_1, \mathbf{x}_2$, and let the subspace S_2 be the subspace generated by $\mathbf{y}_1, \mathbf{y}_2$.

(a) Develop a test to determine if $S_1 = S_2$. Use the test on the subspaces generated by the vectors in (b) and (c) below.

(b)

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \ , \ \mathbf{x}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \ , \ \mathbf{y}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \ , \ \mathbf{y}_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} .$$

(c)

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \ , \ \mathbf{x}_2 = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \ , \ \mathbf{y}_1 = \begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix} \ , \ \mathbf{y}_2 = \begin{bmatrix} 3 \\ -2 \\ -1 \end{bmatrix}$$

30. Calculate the least-squares solution to the following problem using the "full-rank" formula. Show that the same result is obtained using the SVD.

$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \mathbf{x} \approx \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

31. The least-squares solution for a full-rank least-squares problem is given by

$$\mathbf{x}_{\text{LS}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}.$$

(a) Show that $\mathbf{\Sigma}_1$ and $\mathbf{V}_1$ from an SVD of $\mathbf{A}$ are both square, invertible matrices. Also show that $\mathbf{V}_1^{-1} = \mathbf{V}_1^T$.

(b) Derive the alternate expression for $\mathbf{x}_{\text{LS}}$ in terms of the SVD of $\mathbf{A}$:

$$\mathbf{x}_{\text{LS}} = \mathbf{V}_1\mathbf{\Sigma}_1^{-1}\mathbf{U}_1^T\mathbf{y}.$$

32. Consider the linear equations $\mathbf{A}\mathbf{x} = \mathbf{y}$ where $\mathbf{A}$ is an $m \times n$ matrix. Suppose $\mathbf{y} \notin$ col($\mathbf{A}$) so that the equations do not have an exact solution. A least-squares solution

can be found using the SVD. Suppose further that the matrix $\mathbf{A}$ has a non-trivial null-space. The SVD gives an orthonormal basis for this null space as the columns in $\mathbf{V}_2$. The least-squares solution is not unique in this case, but all least-squares solutions are given by

$$\mathbf{x}_{\mathrm{LS}} = \mathbf{z} + \mathbf{V}_2\boldsymbol{\alpha}$$

where $\boldsymbol{\alpha}$ is a vector of q free parameters and $\mathbf{V}_2$ is an $n \times q$ matrix. Different choices for $\boldsymbol{\alpha}$ will result in least-squares solutions with different properties. For example, if we choose $\boldsymbol{\alpha} = \mathbf{0}$, then $\mathbf{x}_{\mathrm{LS}} = \mathbf{z}$ is the least-squares solution of minimum norm.

Suppose we want to use the degrees of freedom given by $\boldsymbol{\alpha}$ to satisfy additional constraints on $\mathbf{x}_{\mathrm{LS}}$ where the constraints are specified as the following linear equations

$$\mathbf{M}\mathbf{x}_{\mathrm{LS}} = \mathbf{w}$$

and $\mathbf{M}$ is a $q \times n$ matrix with $\mathbf{w} \in \mathrm{col}(\mathbf{M})$.

(a) Show that we can always choose $\boldsymbol{\alpha}$ to obtain a least-squares solution which satisfies $\mathbf{M}\mathbf{x}_{\mathrm{LS}} = \mathbf{w}$.

(b) Under what conditions will $\boldsymbol{\alpha}$ be unique?

CHAPTER 3

STATE SPACE AND TRANSFER FUNCTION MODELS

As suggested by the subtitle of this book, primary emphasis is placed on state-space techniques for the design of control systems. This means that the mathematical models for the plant and for the compensator will be in state-space form. Thus we need to develop state-space models for continuous-time systems to represent the plant, which is assumed to be a continuous-time system. In addition, we need to develop state-space models for discrete-time systems to represent the compensator, since we are interested in digital control. Hopefully the reader will appreciate the great degree of algebraic similarity (as well as some differences) in the continuous- and discrete-time theories. Although state-space techniques are at the heart of this book, it is still necessary to recall transfer functions and transforms, and their relationships with state-space models.

3.1 Transforms and Transfer Functions

It is assumed that the reader has a good background in the use of transforms for both continuous and discrete-time systems. In this section, we present basic facts that will be used later. A more complete treatment can be found in [13],[72].

3.1.1 Continuous-Time Systems

Before a control system can be designed for a plant using the techniques presented in this book, a mathematical model for the plant must be obtained. The problem of obtaining a

model is called the *system identification problem*, and is beyond the scope of this book. Interested readers may refer to [57]. In this book we assume that a mathematical model of the plant is available and consider the design of a control system. In this section, we recall many of the basic results and formulas regarding transforms and transfer functions for continuous-time systems. A later section will develop state-space models for continuous-time systems.

We will assume that the plant can be modeled as a *linear, time-invariant system*. It may be that the actual plant is a nonlinear system. In that case, it may be possible to linearize the system about an equilibrium point, and the compensator can be designed for the linearized system. For the remainder of this chapter, we will work with linear, time-invariant(LTI) systems.

An LTI system with input $u(t)$ and output $y(t)$ can be represented by an n^{th}-order differential equation relating the input and output. A general form of such an equation is

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots + a_n y(t) = b_0 u^{(n)}(t) + b_1 u^{(n-1)}(t) + \cdots + b_n u(t) \quad (3.1)$$

where $y^{(n)}(t)$ stands for the n^{th} derivative of $y(t)$ with respect to time. It is customary to assume that (3.1) describes the behavior of the system starting at time zero, and that the input $u(t)$ and output $y(t)$ are zero for $t < 0$. In this section, we make the further assumption that the *initial conditions* for (3.1), $y(0), y^{(1)}(0), \cdots, y^{(n-1)}(0)$ are all equal to zero. It will be seen that initial conditions are handled conveniently with state-space models.

With the above assumptions, we can use the one sided Laplace transform which is defined below

$$Y(s) = L\{y(t)\} \stackrel{\text{def}}{=} \int_0^\infty y(t)e^{-st}dt. \quad (3.2)$$

The above integral exists for values of s in a certain region in the complex plane called the *region of convergence* [46]. Among the many useful properties of the Laplace transform, we recall that the formula for the transform of the derivative of a function (with zero initial conditions) is

$$L\{y^{(k)}(t)\} = s^k Y(s). \quad (3.3)$$

A list of useful properties of Laplace transforms is given in Table 3.1 on page 83.

Using (3.2) and (3.3), we can take the Laplace transform of the differential equation in (3.1) to obtain

$$(s^n + a_1 s^{n-1} + a_2 s^{n-2} + \cdots + a_n)Y(s) = (b_0 s^n + b_1 s^{n-1} + \cdots + b_n)U(s).$$

This equation may be rearranged to obtain the *transfer function* of the system represented by the differential equation in (3.1)

$$H(s) \stackrel{\text{def}}{=} \frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_n}{s^n + a_1 s^{n-1} + \cdots + a_n}. \quad (3.4)$$

The numerator and denominator polynomials are denoted as

$$b(s) \stackrel{\text{def}}{=} b_0 s^n + b_1 s^{n-1} + \cdots + b_n$$

$$a(s) \stackrel{\text{def}}{=} s^n + a_1 s^{n-1} + \cdots + a_n.$$

Two conventions used in this book are i) the coefficients of a polynomial are labeled starting with the highest power and proceeding to the constant term, and ii) the denominator polynomial will always be *monic*, i.e. its leading coefficient will be unity.

Property	Time Function	Laplace Transform
Definition	$f(t)$	$F(s)$
Linearity	$a_1 f_1(t) + a_2 f_2(t)$	$a_1 F_1(s) + a_2 F_2(s)$
Differentiation	$\frac{d^k}{dt^k} f(t)$	$s^k F(s) - s^{k-1} f(0) - s^{k-2} f^{(1)}(0) - \cdots - f^{(k-1)}(0)$
Integration	$\int_0^t f(\tau) d\tau$	$F(s)/s$
Time Shift	$f(t - t_0)$	$e^{-t_0 s} F(s)$
Time Scaling	$f(at)$	$\frac{1}{\lvert a \rvert} F\left(\frac{s}{a}\right)$
Convolution Integral	$\int_0^t f_1(t) f_2(t-\tau) d\tau = \int_0^t f_1(t-\tau) f_2(t) d\tau$	$F_1(s) F_2(s)$
Exponential Modulation	$e^{-at} f(t)$	$F(s + a)$
Initial Value Theorem	$f(0+) = \lim_{s \to \infty} sF(s)$ if df/dt is Laplace transformable.	
Final Value Theorem	$\lim_{t \to \infty} f(t) = \lim_{s \to 0} sF(s)$ if the poles of $sF(s)$ are in the left half plane.	

Table 3.1 Properties of Laplace Transforms. All time functions are assumed to equal zero for $t < 0$.

$f(t)$	$F(s)$
Impulse Function $\delta(t)$	1
Step Function $f(t) = 1$	$\frac{1}{s}$
e^{-at}	$\frac{1}{s+a}$
$\sin \omega t$	$\frac{\omega}{s^2+\omega^2}$
$\cos \omega t$	$\frac{s}{s^2+\omega^2}$
t^n	$\frac{n!}{s^{n+1}}$
$1 - e^{-at}$	$\frac{a}{s(s+a)}$
$t^n e^{-at}$	$\frac{n!}{(s+a)^{n+1}}$

Table 3.2 Some Laplace Transform Pairs. All time functions are assumed to equal zero for $t < 0$.

From the definition of the transfer function, we recall that the Laplace transform of the output of a system is just the product of the Laplace transform of the input to the system and the transfer function:

$$Y(s) = H(s)U(s). \tag{3.5}$$

Taking inverse Laplace transforms of this equation yields (see Table 3.1)

$$y(t) = \int_0^t h(t-\tau)u(\tau)d\tau = \int_0^t h(\tau)u(t-\tau)d\tau \tag{3.6}$$

where the $h(t)$ is the inverse Laplace transform of the transfer function $H(s)$. The function $h(t)$ is called the *impulse response* of the system for the following reason. If $U(s) = 1$ in (3.5), then $Y(s) = H(s)$ and taking inverse Laplace transforms yields $y(t) = h(t)$. Thus $h(t)$ is the output of the system when the input $u(t)$ has Laplace transform $U(s) = 1$. From Table 3.2 this means that $u(t) = \delta(t)$, the impulse function.

Equation (3.6) expresses the output of the system in terms of the *convolution integral* of the impulse response and the input. Note that (3.6) assumes that both the input and the impulse response are equal to zero for $t < 0$, otherwise, the integrals should range from $-\infty$ to ∞. A system whose impulse response equals zero for $t < 0$ is said to be *causal*.

The convolution representation may be used to define a notion of stability for a system. Namely, if the input to the system is bounded, $|u(t)| \leq M$, it is easy to show that the output of the system will be bounded if and only if

$$\int_0^\infty |h(t)|\, dt < \infty. \tag{3.7}$$

A system whose impulse response satisfies (3.7) is said to be BIBO (bounded input bounded output) stable.

Consider the following transfer function

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

where the degree of $b(s)$ is less than the degree of $a(s)$. (If they have equal degree, use synthetic division to express $H(s)$ as a constant plus a ratio of polynomials with the required degree property.) $H(s)$ can be expanded in a *partial fraction expansion* as follows

$$H(s) = \sum_i \frac{A_i}{s - p_i} + \sum_r \sum_{j=1}^{m_r} \frac{B_{rj}}{(s - p_r)^j} \tag{3.8}$$

where the summation over index i includes all distinct roots p_i of $a(s)$, and the summation over index r includes all repeated roots p_r of $a(s)$, where root p_r has multiplicity $m_r > 1$. The roots of $a(s)$ are the poles of the system. The formulas for the expansion coefficients in (3.8) are

$$A_i = H(s)(s - p_i)|_{s=p_i} \tag{3.9}$$

and

$$B_{rj} = \frac{1}{(m_r - j)!} \left\{ \frac{d^{m_r - j}}{ds^{m_r - j}} (s - p_r)^{m_r} H(s) \right\} |_{s=p_r}. \tag{3.10}$$

If the denominator polynomial $a(s)$ has complex roots, then they must occur in complex-conjugate pairs. Suppose p_i is a complex root and $p_j = p_i^*$ is the corresponding conjugate root (the superscript $*$ refers to complex conjugation). We can show that the coefficients A_i and A_j corresponding to p_i and p_j in the partial fraction expansion of $H(s)$ are also related by conjugation. The coefficient A_i is given by (3.8). The same equation is used to evaluate A_j as shown below

$$\begin{aligned} A_j &= H(s)(s - p_j)|_{s=p_j} \\ &= H(s)(s - p_i^*)|_{s=p_i^*} \\ &= [H(s)(s - p_i)|_{s=p_i}]^* \\ &= A_i^*. \end{aligned} \tag{3.11}$$

The coefficients A_i and B_{rj} can be computed by hand for low-order systems. An example of this calculation is given in the first section of Chapter 5. An algorithm for computing these coefficients by computer is given in [9].

The inverse Laplace transform of (3.8) can be taken term by term since the Laplace transform is linear, and the inverse transform of each term in the sum can be easily calculated (see in Tables 3.1 and 3.2). The resulting inverse transform is

$$h(t) = L^{-1}[H(s)] = \sum_{i=1} A_i e^{p_i t} + \sum_r \sum_{j=1}^{m_r} B_{rj} \frac{t^{(j-1)}}{(j-1)!} e^{p_r t}.$$

It is easy to show that the above equation will satisfy the stability condition in (3.7) if and only if the real parts of all the pole locations are less than zero, i.e. $\mathrm{Re}(p_i) < 0$. **This result says that a stable continuous-time system has all its poles in the left half of the s- plane.**

3.1.2 Discrete-Time Systems

Digital control is accomplished when the compensator is a discrete-time system. Thus a treatment of digital control requires a thorough understanding of discrete-time systems.

Property	Sequence	z-Transform
Definition	$f[k]$	$F(z)$
Linearity	$a_1 f_1[k] + a_2 f_2[k]$	$a_1 F_1(z) + a_2 F_2(z)$
Delay	$f[k-n]$	$z^{-n}F(z) \quad + \sum_{i=1}^{n} f[-i]z^{i-n}$
Advance	$f[k+n]$	$z^n F(z) - \sum_{i=0}^{n-1} f[i]z^{n-i}$
Convolution Sum	$\sum_{i=0}^{k} f_1[k-i]f_2[i] = \sum_{i=0}^{k} f_1[i]f_2[k-i]$	$F_1(z)F_2(z)$
Exponential Modulation	$a^k f[k]$	$F(a^{-1}z)$
Initial Value Theorem	$f[0] = \lim_{\|z\|\to\infty} F(z)$ if the limit exists.	
Final Value Theorem	$\lim_{k\to\infty} f[k] = \lim_{z\to 1}(z-1)F(z)$ if the poles of $(z-1)F(z)$ are inside the unit circle.	

Table 3.3 Properties of z-transforms. All sequences are assumed to equal zero for $k < 0$.

This section provides a brief review of the z-transform and transfer functions for discrete-time systems. A later section on state-space system theory provides more information about discrete-time state-space models.

A linear, time-invariant discrete-time system with input sequence $u[k]$ and output sequence $y[k]$ can be represented by an n^{th}-order difference equation relating the input and output. A general form of such an equation is

$$y[k] + a_1 y[k-1] + \cdots + a_n y[k-n] = b_0 u[k] + b_1 u[k-1] + \cdots + b_n u[k-n]. \quad (3.12)$$

In the above equation the time variable is k, and n is a fixed integer which is called the order of the difference equation. We assume in this section that the *initial conditions* $y[-n], y[-n+1], \cdots, y[-1]$ are all equal to zero. The case of non-zero initial conditions will be treated using state-space models.

We use the one-sided z-transform which is defined below

$$Y(z) = Z\{y[k]\} \stackrel{\text{def}}{=} \sum_{k=0}^{\infty} y[k]z^{-k}. \quad (3.13)$$

Among the many useful properties of the z-transform, (see Table 3.3), recall the formula for the transform of a shifted sequence when the sequence has zero initial conditions

$$Z\{y[k-n]\} = z^{-n}Y(z). \quad (3.14)$$

Using (3.13) and (3.14), we can take the z-transform of the difference equation (3.12) to obtain

$$(1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n})Y(z) = (b_0 + b_1 z^{-1} + \cdots + b_n z^{-n})U(z).$$

We can multiply both sides of this equation by z^n and rearrange it to obtain the transfer function of the discrete-time system represented by the difference equation (3.12)

$$H(z) \stackrel{\text{def}}{=} \frac{Y(z)}{U(z)} = \frac{b_0 z^n + b_1 z^{n-1} + \cdots + b_n}{z^n + a_1 z^{n-1} + \cdots + a_n}. \tag{3.15}$$

The numerator and denominator polynomials are defined as

$$b(z) \stackrel{\text{def}}{=} b_0 z^n + b_1 z^{n-1} + \cdots + b_n$$

$$a(z) \stackrel{\text{def}}{=} z^n + a_1 z^{n-1} + \cdots + a_n.$$

From the definition of the transfer function, we recall that the z-transform of the output of a system is just the product of the z-transform of the input and the transfer function: $Y(z) = H(z)U(z)$. Taking inverse z-transforms yields (see Table 3.3)

$$y[k] = \sum_{i=0}^{k} h[k-i]u[i] = \sum_{i=0}^{k} h[i]u[k-i] \tag{3.16}$$

where $h[k]$ is the inverse z-transform of the transfer function $H(z)$. The sequence $h[k]$ is called the *impulse response* of the discrete-time system. It is the system output when the input is the delta (impulse) sequence

$$\delta[k] = \begin{cases} 1, & k = 0 \\ 0, & \text{otherwise.} \end{cases}$$

If $u[k] = \delta[k]$, then $U(z) = 1$ (see Table 3.4). In this case, $Y(z) = H(z)$, and the system output $y[k]$ equals the impulse response $h[k]$.

Equation (3.16) expresses the output of the system in terms of the *convolution sum* of the impulse response and the input. Note that (3.16) assumes that both the input and the impulse response are zero for $k < 0$. The convolution representation may be used to define a notion of stability for a discrete-time system. Namely, if the input to the system is bounded, $|u[k]| < M$, it is easy to show that the output of the system will be bounded if and only if

$$\sum_{k=0}^{\infty} |h[k]| < \infty.$$

Finally, we recall that the transfer function can be decomposed into a *partial fraction expansion* as

$$H(z) = \sum_i \frac{zA_i}{z - z_i} + \sum_r \sum_{j=1}^{m_r} \frac{zB_{rj}}{(z - z_r)^j} \tag{3.17}$$

where the summation over index i includes all distinct roots z_i of $a(z)$, and the summation over index r includes all repeated roots z_r of $a(z)$, where root z_r has multiplicity $m_r > 1$. The formulas for the expansion coefficients in (3.17) are

$$A_i = \frac{H(z)(z - z_i)}{z}\Big|_{z=z_i} \tag{3.18}$$

$f[k]$	$F(z)$
Delta Sequence $\delta[k] = \begin{cases} 1, & k = 0 \\ 0, & \text{otherwise} \end{cases}$	1
Step Sequence $f[k] = 1$	$\frac{z}{z-1}$
Ramp Sequence $f[k] = k$	$\frac{z}{(z-1)^2}$
Parabolic Sequence $f[k] = k^2$	$\frac{z(z+1)}{(z-1)^3}$
$r^k \sin(k\omega T + \theta)$	$\frac{rz[z\sin\theta + \sin(\omega T - \theta)]}{z^2 - 2rz\cos\omega T + r^2}$
$r^k \cos(k\omega T + \theta)$	$\frac{z[z\cos\theta - r\cos(\omega T - \theta)]}{z^2 - 2rz\cos\omega T + r^2}$
a^k	$\frac{z}{z-a}$
$\binom{k}{j-1} a^{k-j+1}$, $\binom{n}{m} = \frac{n!}{m!(n-m)!}$,	$\frac{z}{(z-a)^j}$

Table 3.4 Some z-Transform Pairs. All sequences are assumed to equal zero for $k < 0$.

and

$$B_{rj} = \frac{1}{(m_r - j)!} \left\{ \frac{d^{m_r - j}}{dz^{m_r - j}} \frac{(z - z_r)^{m_r}}{z} H(z) \right\} |_{z=z_r}. \tag{3.19}$$

The coefficients A_i and B_{rj} can be calculated by hand for low-order systems. An example is given in the first section of Chapter 5.

The inverse z-transform of (3.17) can be taken term by term since the z-transform is linear, and the inverse transform of each term in the sum can be easily calculated (see Table 3.4). The resulting inverse transform is

$$h[k] = Z^{-1}[H(z)] = \sum_i A_i p_i^k + \sum_r \sum_{j=1}^{m_r} B_{rj} \binom{k}{j-1} p_r^{k-j-1}$$

It is easy to show that the above equation will satisfy the stability condition in (3.7) if and only if the magnitudes of all the poles are less than one, i.e. $|p_i| < 1$. **This result says that a stable discrete-time system has all its poles inside the unit circle of the z-plane.**

3.2 Relationship Between Continuous and Discrete Step Responses

The previous section covered transforms and transfer functions for continuous- and discrete-time systems. Using these results we can derive a relationship between the pole locations of a continuous-time control system and the pole locations of a corresponding *sampled-data* (digital) control system. The relationship is derived for arbitrary systems in Chapter 4. In the following, we consider the step response of the second-order system

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

This system is sometimes called the *standard second-order system.* The parameter ζ is the *damping ratio* and ω_n is the *natural frequency*. In what follows, we assume that $0 < \zeta \leq 1$.

A typical step response of $G(s)$ is shown in Fig. 3.1. Also shown in the figure are the

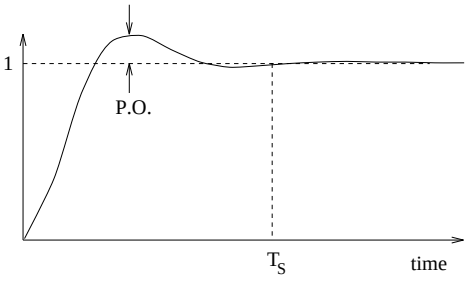


Figure 3.1 Typical step response of a second-order system.

percent overshoot (P.O.) and the settling time T_S. The poles of $G(s)$ are located at

$$s_{1,2} = -\sigma \pm j\omega_d \tag{3.20}$$

where

$$\sigma = \zeta\omega_n$$

$$\omega_d = \omega_n\sqrt{1-\zeta^2}.$$

It is an exercise in Laplace transforms to show that the step response of $G(s)$ is [31]

$$y(t) = 1 - e^{-\sigma t}\left(\cos(\omega_d t) + \frac{\sigma}{\omega_d}\sin(\omega_d t)\right) \tag{3.21}$$

where

$$\omega_d = \omega_n\sqrt{1-\zeta^2}\ , \ \text{ and } \ \sigma = \zeta\omega_n.$$

Using (3.21), the percent overshoot and settling time can be expressed as functions of ζ and ω_n. For example [31]

$$\begin{aligned} T_S &\approx \frac{4.5}{\sigma} \\ \text{P.O.} &= 100\cdot\exp\left(\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}\right)\ , \ 0<\zeta<1. \end{aligned} \tag{3.22}$$

These expressions show that the speed of response (T_s) is governed by the real part of the pole locations – a fast response requires poles "far" into the left half-plane – and that the overshoot is only a function of ζ.

For digital control, we will only be able to specify the behavior of the closed-loop system at sampling instants. The sampled-data (digital) control system is described by a discrete-time transfer function which we will call $G(z)$. We would like the step response $y[k]$ of $G(z)$ to be a sampled version of the step response given by (3.21) and shown in Fig. 3.1. In other words, we would like

$$y[k] = y(t)\mid_{t=kT}. \tag{3.23}$$

The question we would like to answer now is: where should the poles of $G(z)$ be located so that (3.23) is satisfied? The answer to this question is developed as follows. The step

response of $G(z)$ is obtained from samples of the step response in (3.21)

$$y[k] \stackrel{\text{def}}{=} y(kT) = 1 - (e^{-\sigma T})^k \left(\cos(\omega_d kT) + \frac{\sigma}{\omega_d} \sin(\omega_d kT) \right). \tag{3.24}$$

In order for $y[k]$ to be the step response of $G(z)$, the z-transform of $y[k]$ must be the product of the z-transform of the step input and $G(z)$. Thus

$$Z\{y[k]\} = \frac{z}{z-1} G(z). \tag{3.25}$$

If we take the z-transform of $y[k]$ in (3.24) and put the results in the form of (3.25), we will be able to find $G(z)$.

Using Table 3.4, the z-transform of (3.24) is

$$Z\{y[k]\} = \frac{z}{z-1} - z \left[\frac{n_1(z) + (\sigma/\omega_d) n_2(z)}{z^2 - 2rz\cos(\omega_d T) + r^2} \right] \tag{3.26}$$

where

$$r = e^{-\sigma T}$$

and the polynomials $n_1(z)$ and $n_2(z)$ are not important for this derivation. Simplifying the expression in (3.26) yields

$$\begin{aligned} Z\{y[k]\} &= \frac{z}{z-1} \left[1 - \frac{(z-1)[n_1(z) + (\sigma/\omega_d) n_2(z)]}{z^2 - 2rz\cos(\omega_d T) + r^2} \right] \\ &= \frac{z}{z-1} \frac{N(z)}{z^2 - 2rz\cos(\omega_d T) + r^2} \end{aligned}$$

which is in the form of (3.25). Thus the poles of the sampled-data system are the roots of

$$z^2 - 2rz\cos(\omega_d T) + r^2$$

which are

$$\begin{aligned} z_{1,2} &= r\cos(\omega_d T) \pm jr\sqrt{1 - \cos^2(\omega_d T)} \\ &= r[\cos(\omega_d T) \pm j\sin(\omega_d T)] \\ &= \left(e^{-\sigma T}\right)\left(e^{\pm j\omega_d T}\right) \\ &= e^{s_{1,2} T} \end{aligned} \tag{3.27}$$

where the last line follows from (3.20). From (3.27) we see that

$$|z_{1,2}| = e^{-\sigma T}$$

so that $-\sigma$, the real part of the s-plane poles, directly controls the magnitude of the z-plane poles. Thus a "fast" step response requires poles "far" into the left half s-plane, or equivalently z-plane poles of "small" radius. The percent overshoot of the step response depends on ζ, and the effect of ζ on z-plane pole locations is not immediately clear from (3.27). Nevertheless, using (3.27), we can plot contours of constant ζ and ω_n in the z-plane. Such a plot is shown in Fig. 3.2 which identifies regions of the z-plane corresponding to different values of ζ and ω_n.

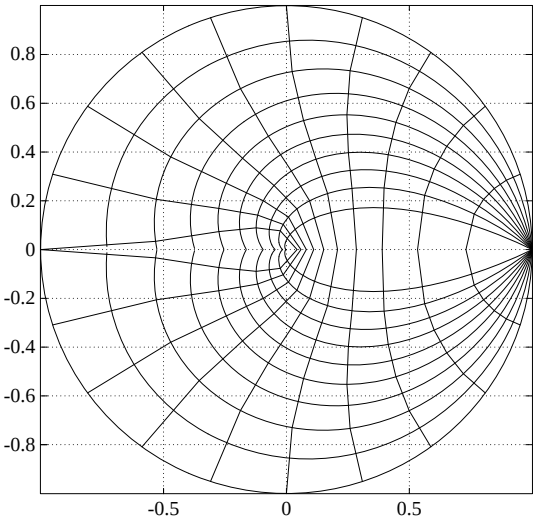


Figure 3.2 Contours of constant ζ and ω_n for a sampled-data control system with sampling interval T seconds.

EXAMPLE 3.1

Consider the following system

$$G(s) = \frac{4}{s^2 + 3.2s + 4}$$

which corresponds to $\zeta = 0.8$ and $\omega_n = 2$. From (3.22) we know that the step response should have a settling time of $4.5/1.6 \approx 2.8$ seconds, and a percent overshoot of 1.5%. The poles of $G(s)$ are located at

$$s_{1,2} = -1.6 \pm j1.2.$$

Let us choose a sampling period of $T = 0.2$ seconds. In order for a sampled-data control system $G(z)$ to have a step response which matches that of $G(s)$ at sampling intervals, the poles of $G(z)$ must be located at (see (3.27))

$$z_{1,2} = e^{s_{1,2}T} = 0.7053 \pm j0.1726.$$

This means that the denominator of $G(z)$ must be

$$(z - 0.7053 - j0.1726)(z - 0.7053 + j0.1726) = z^2 - 1.4107z + 0.5273.$$

Using results developed in Chapter 4, we find that $G(z)$ is in fact given by

$$G(z) = \frac{0.0645(z + 0.8075)}{z^2 - 1.4107z + 0.5273}$$

which has the expected denominator polynomial. The step responses of $G(s)$ and $G(z)$ are shown in Fig. 3.3.

■

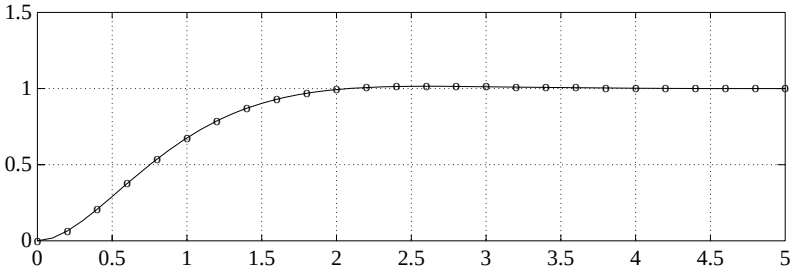


Figure 3.3 Step responses of $G(s)$ (solid line) and $G(z)$ (circles).

3.3 State-Space Models

We have seen that linear, time-invariant systems can be described by differential (or difference) equations as well as by transfer functions. These systems can also be described by state-space models which are introduced in this section. It will be seen in later chapters that the use of state-space models allows tools from linear algebra to be used in the design of digital control systems.

In order to design a control system using state-space techniques, we will need a state-space model for the plant. The plant model will be a continuous-time system, and canonical forms for such systems are derived in this section. The transfer function of an n^{th}-order system contains $2n+1$ parameters: the numerator coefficients $b_0, \cdots, b_n$ and the denominator coefficients $a_1, \cdots, a_n$. (Note that some of these parameters may be zero for a given system.) An n^{th}-order state-space model is a "canonical form" if it is written in terms of $2n+1$ parameters and can represent an arbitrary n^{th}-order transfer function. The parameters in the canonical state-space models used in this book are the transfer function coefficients of the system. Since the compensator will be a discrete-time system, we will also derive canonical forms for discrete-time systems. In addition, we develop some linear algebraic tools for handling state-space models.

Thus far we have considered various representations of linear, time-invariant systems. We now introduce a representation for a more general class of systems which includes nonlinear and time-varying effects. This representation is the so-called *state-space model* for a system with input $u(t)$ and output $y(t)$. Of central importance in such models is the *state vector* $\mathbf{x}(t)$ which is a vector of n time functions

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix}$$

for continuous-time systems, or a vector of n sequences

$$\mathbf{x}[k] = \begin{bmatrix} x_1[k] \\ \vdots \\ x_n[k] \end{bmatrix}$$

for discrete-time systems. We use lower case boldface symbols to represent vectors throughout this book. Let us deal with the continuous-time case first.

A state-space description for a continuous-time system consists of a first-order vector differential equation for $\mathbf{x}(t)$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t), t) \tag{3.28}$$

and an output equation

$$y(t) = g(\mathbf{x}(t), u(t), t). \tag{3.29}$$

The vector-valued function $\mathbf{f}(\cdot, \cdot, \cdot)$ consists of n functions

$$\mathbf{f}(\mathbf{x}(t), u(t), t) = \begin{bmatrix} f_1(\mathbf{x}(t), u(t), t) \\ \vdots \\ f_n(\mathbf{x}(t), u(t), t) \end{bmatrix}$$

each of which is a possibly nonlinear function of all of the state variables in $\mathbf{x}(t)$, $u(t)$ and t. Equation (3.28) is a vector form of n coupled differential equations. The i^{th} such equation is

$$\dot{x}_i(t) = f_i(\mathbf{x}(t), u(t), t). \tag{3.30}$$

An example of a system which has a nonlinear state-space description of the form shown above is the cart/pendulum system discussed in Chapter 1. A derivation of the model for that system is given in Section 3.6.6.

Note that the differential equation for $\dot{x}_i(t)$ in (3.30) depends on *all* the state variables in $\mathbf{x}(t)$. Thus we cannot solve for $x_i(t)$ independently of the other state variables. This is what is meant by coupled differential equations. We also note that the function f_i in (3.30) may be an explicit function of time which means that the system is *time varying*. Thus the simple notation in (3.28) and (3.29) can represent a high-order, nonlinear, time-varying system! However, for the rest of this chapter (and for most of the book), we are interested in the form that the state-space equations take for linear, time-invariant systems.

What does it mean for the functions f_i to be linear? It means that $\dot{x}_i(t)$ can be expressed as a *linear* function of all the state variables and the input. That is

$$\dot{x}_i(t) = a_{i1}x_1(t) + a_{i2}x_2(t) + \cdots + a_{in}x_n(t) + b_i u(t) \tag{3.31}$$

for some set of coefficients $\{a_{ij}\}$ and $\{b_i\}$, $i = 1, \cdots, n$ and $j = 1, \cdots, n$. If, in addition to being linear, the functions f_i are also time invariant, then the coefficients a_{ij} and b_i do not change with time; that is, they are constants. If we integrate both sides of (3.31) we get

$$x_i(t) = x_i(t_0) + \sum_{j=1}^{n} a_{ij} \int_{t_0}^{t} x_j(\tau) d\tau + b_j \int_{t_0}^{t} u(\tau) d\tau. \tag{3.32}$$

If $x_i(t_0)$ is zero, we see that $x_i(t)$ is obtained by a *sum of integrals involving state variables and the input.* If $x_i(t_0)$ is not zero, it is simply added to this integration. We will return to this point when we develop canonical forms in the next section.

For linear, time-invariant systems, we can write (3.31) for all the functions f_i in vector/matrix notation as

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

where

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

In a similar way, if the output equation (3.29) is linear and time invariant then there exists a set of coefficients $\{c_i\}$, $i = 1, \cdots, n$ and a coefficient d such that

$$y(t) = c_1 x_1(t) + \cdots + c_n x_n(t) + Du(t) = \mathbf{C}\mathbf{x}(t) + Du(t)$$

where the output vector $\mathbf{C} = [\, c_1 \quad \cdots \quad c_n \,]$. Thus the general form for a linear, time-invariant, n^{th}-order state-space description is

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,u(t) \\ y(t) &= \mathbf{C}\,\mathbf{x}(t) + D\,u(t) \end{aligned} \tag{3.33}$$

where $\mathbf{A}$ is the $n \times n$ *system matrix*, $\mathbf{B}$ is the $n \times 1$ *input vector*, $\mathbf{C}$ is the $1 \times n$ *output vector*, and D is a scalar called the *feedthrough coefficient*.

The state-space equations are initialized at some time t_0 (often taken to equal 0) by a vector of numbers $\mathbf{x}_0$ called the *initial condition vector*. That is, $\mathbf{x}(t_0) = \mathbf{x}_0$. The relationship between the initial condition vector of a state-space model and the initial conditions of a corresponding differential equation is given in Section 3.3.6.

In Chapter 9, we consider *multivariable systems*; that is, systems with more than one input and more than one output. A state-space description of a multivariable system takes the form of (3.33) with the following changes. The input vector is replaced by an input matrix $\mathbf{B}$, where the i^{th} column of $\mathbf{B}$ is the input vector for the i^{th} input to the system. Similarly, the output vector in (3.33) is replaced by an output matrix $\mathbf{C}$ where the j^{th} row of $\mathbf{C}$ is the output vector for the j^{th} output. The feedthrough coefficient is replaced by a matrix $\mathbf{D}$ whose ij entry is the feedthrough coefficient from input j to output i.

The development of discrete-time state-space models is very similar to the continuous-time case. A state-space description for a discrete-time system consists of a first-order vector difference equation for $\mathbf{x}[k]$

$$\mathbf{x}[k+1] = \mathbf{f}(\mathbf{x}[k], u[k], k) \tag{3.34}$$

and an output equation

$$y[k] = g(\mathbf{x}[k], u[k], k). \tag{3.35}$$

If the functions f_i in (3.34) are linear and time invariant, then $x_i[k+1]$ can be expressed as a *linear* function of all the state variables and the input. That is

$$x_i[k+1] = a_{i1}x_1[k] + a_{i2}x_2[k] + \cdots + a_{in}x_n[k] + b_i u[k] \tag{3.36}$$

for some set of coefficients $\{a_{ij}\}$ and $\{b_i\}$, $i = 1, \cdots, n$ and $j = 1, \cdots, n$. We can then write (3.36) for all the functions f_i in vector/matrix notation as

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}u[k]. \tag{3.37}$$

The matrix $\mathbf{A}$ is called the *system matrix* and the vector $\mathbf{B}$ is called the *input vector*. We call (3.37) the *state-update equation* for a discrete-time system. This equation expresses the value of the state vector at time $k+1$ in terms of the state vector and input at time k.

If the output equation (3.35) is linear and time invariant then there exists a set of coefficients $\{c_i\}$, $i = 1, \cdots, n$ and a coefficient d such that

$$y[k] = c_1 x_1[k] + \cdots + c_n x_n[k] + du[k] = \mathbf{C}x[k] + Du[k].$$

Thus the general form for a discrete-time, linear, time-invariant, n^{th}-order state-space description is

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{A}\,\mathbf{x}[k] + \mathbf{B}\,u[k] \\ y[k] &= \mathbf{C}\,\mathbf{x}[k] + D\,u[k] \end{aligned} \tag{3.38}$$

where $\mathbf{A}$ is an $n \times n$ matrix, $\mathbf{B}$ is an $n \times 1$ vector, $\mathbf{C}$ is a $1 \times n$ vector, and D is a scalar. The state-space model is initialized at some time index k_0 by an initial condition vector $\mathbf{x}_0$; that is $\mathbf{x}[k_0] = \mathbf{x}_0$.

EXAMPLE 3.2

In this example we obtain a state-space model for a dc motor. In addition we show how the motor parameters can be obtained from a step response. A physical block diagram for a motor is shown in Fig. 3.4. The motor is driven by a power amplifier

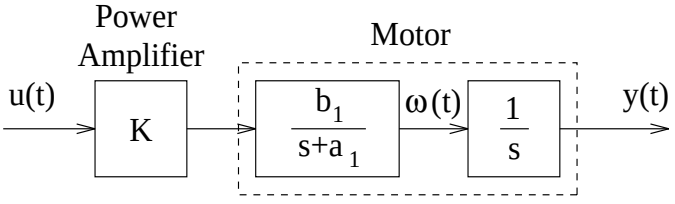


Figure 3.4 Block diagram of a dc motor for Example 3.2.

which produces a current proportional to the input voltage $u(t)$. The output $y(t)$ of this system is the angular position $\theta(t)$ of the motor shaft. The angular velocity $\omega(t)$ is also an important variable for this system. The form of the motor transfer function shown in Fig. 3.4 can be derived from the laws of physics [15].

It is interesting to note that the parameter values for an actual motor system can be determined from a step response. Consider what happens when the step input

$$u(t) = 1 \text{ volt}$$

is applied to the system when the motor shaft is initially at zero velocity. The velocity will start to increase and then remain constant at the value for which the damping torque due to viscous friction is equal to the driving torque due to the input current. From Fig. 3.4 the transfer function from input voltage to angular velocity is

$$\frac{\boldsymbol{\omega}(s)}{U(s)} = \frac{Kb_1}{s + a_1}$$

and so the angular velocity step response is

$$\omega(t) = L^{-1}\left[\frac{1}{s}\frac{Kb_1}{s+a_1}\right] = Kb_1(1 - e^{-a_1 t}).$$

A typical step response is shown in Fig. 3.5. The parameters a_1 and Kb_1 can be obtained directly from the step response.

In order to obtain a state-space model of a motor, suppose we define the following state variables:

$$\begin{aligned} x_1(t) &= y(t) \\ x_2(t) &= \omega(t) \;=\; \dot{y}(t) \end{aligned} \quad \text{and } \mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}.$$

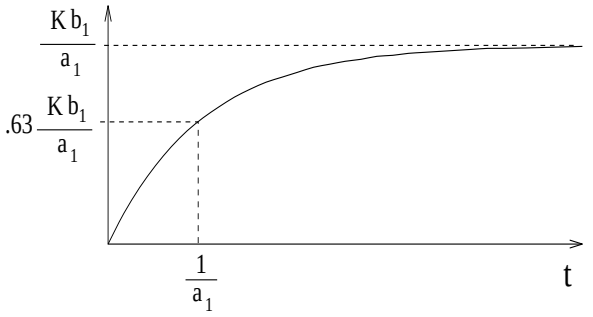


Figure 3.5 Motor velocity response to a step input for Example 3.5.

Using the transfer function from $u(t)$ to $\omega(t)$ we can write

$$(s + a_1)\boldsymbol{\omega}(s) = Kb_1U(s).$$

Taking inverse Laplace transforms and recalling that $x_2(t) = \omega(t)$ yields

$$\dot{x}_2(t) + a_1x_2(t) = Kb_1u(t).$$

From the definition of the state variables we know that

$$\dot{x}_1(t) = x_2(t).$$

The two previous equations can be written together as

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -a_1 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ Kb_1 \end{bmatrix} u(t),$$

and from the definition of the state variables we have

$$y(t) = [\,1 \quad 0\,]\,\mathbf{x}(t) + 0\; u(t).$$

The previous two equations are a state-space model for the motor system written in the form given by (3.33) with

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -a_1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ Kb_1 \end{bmatrix}, \quad \mathbf{C} = [\,1 \quad 0\,], \quad D = 0.$$

For future reference we note from Fig. 3.4 that the transfer function of the complete system from $u(t)$ to $y(t)$ is

$$G(s) = \frac{Kb_1}{s^2 + a_1 s}.$$

■

The derivation of the state-space model in the previous example started by choosing the state variables $x_1(t)$ and $x_2(t)$. A natural question to ask is how the state variables are chosen for an arbitrary system. Instead of answering this question directly, we answer a related question: namely, can we derive a state-space description for an arbitrary system if we know the system transfer function or equivalently, the n^{th}-order differential (or difference) equation? The answer to this question is given for both continuous and discrete-time systems in the following sections.

3.3.1 A Continuous-Time Canonical Form

We have previously seen that a linear, time-invariant system can be described in a variety of equivalent ways: by a differential equation, a transfer function, or an impulse response. In this section, we show how an n^{th}-order differential equation can be converted into n first-order equations which are a *state-space* model for the system. The fundamental tool for deriving such a representation is the cascade of summers and integrators shown in Fig. 3.6. Notice that we use the Laplace transform symbol s^{-1} to denote an integrator (see Table 3.1 on page 83).

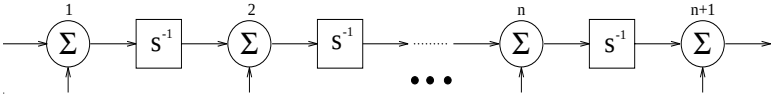


Figure 3.6 Cascade of n integrators and $n + 1$ summing junctions.

Signals which are introduced at different summing junctions in Fig. 3.6 will experience a different number of integrations before reaching the output. In particular, signals which are introduced at summing junction 1 will be integrated n times before reaching the output. Signals which are introduced at summing junction 2 will be integrated $n - 1$ times before reaching the output. Finally, signals which are introduced at summing junction n will be integrated once before reaching the output. Thus the system shown in Fig. 3.6 can produce sums of integrated signals with the number of integrations of any signal ranging from 1 to n. This is all that is needed to solve a general n^{th}-order differential equation.

Recall that such an equation is given by

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots + a_n y(t) = b_0 u^{(n)}(t) + b_1 u^{(n-1)}(t) + \cdots + b_n u(t). \quad (3.39)$$

Suppose we now integrate both sides of the above equation n times. If we assume as before that the initial conditions are all zero, we obtain

$$y(t) + a_1 y^{[1]}(t) + \cdots + a_n y^{[n]}(t) = b_0 u(t) + b_1 u^{[1]}(t) + \cdots + b_n u^{[n]}(t), \quad (3.40)$$

where the superscript $[j]$ stands for j integrations of the function. Solving for $y(t)$, we obtain

$$y(t) = -a_1 y^{[1]}(t) - \cdots - a_n y^{[n]}(t) + b_0 u(t) + b_1 u^{[1]}(t) + \cdots + b_n u^{[n]}(t). \quad (3.41)$$

From the above equation, it can be seen that $y(t)$ is formed from a linear combination of signals that are integrated up to n times. Thus $y(t)$ could be the output of the system shown in Fig. 3.6 if the signals are introduced at the appropriate summing junctions. In particular, if the signals $b_n u(t)$ and $-a_n y(t)$ are introduced at summing junction 1, they will be integrated n times as required by (3.41). If the signals $b_{n-1} u(t)$ and $-a_{n-1} y(t)$ are introduced at summing junction 2, they will be integrated $n - 1$ time times as required by (3.41). This process is repeated, until we come to the term $b_0 u(t)$ which is not integrated, and so it gets added in at summing junction $n + 1$. The resulting system is shown in Fig. 3.7.

In order to write the state-variable equations corresponding to Fig. 3.7, we need to express the derivatives of the state variables in terms of the state variables and the input. These equations can be obtained from Fig. 3.7. Recall from (3.32) that the state variables are obtained by integration. Thus it is always possible to choose the state variables to be

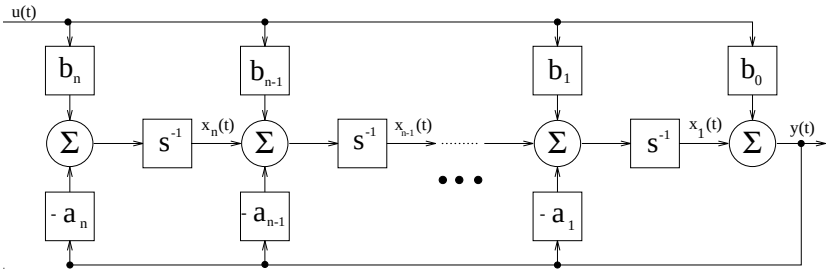


Figure 3.7 A system which solves a general n^{th}-order differential equation. State variable $x_1(t) \cdots x_n(t)$ are shown as the outputs of the integrators.

the outputs of the integrators in Fig. 3.7. With this choice of state variables, the inputs to the integrators are the derivatives of the state variables. We also notice from Fig. 3.7 that the output equation is simply

$$y(t) = x_1(t) + b_0 u(t). \tag{3.42}$$

To obtain the state equations, we start with the last integrator, which from Fig. 3.7 gives

$$\dot{x}_1(t) = -a_1 y(t) + x_2(t) + b_1 u(t), \tag{3.43}$$

and substituting (3.42) into (3.43), we obtain

$$\dot{x}_1(t) = -a_1 x_1(t) + x_2(t) + (b_1 - a_1\, b_0) u(t).$$

A similar procedure can be followed for all the integrators in Fig. 3.7, resulting in the following state-variable equations

$$\begin{aligned}
\dot{x}_1(t) &= -a_1 x_1(t) + x_2(t) + (b_1 - a_1\, b_0) u(t) \\
\dot{x}_2(t) &= -a_2 x_1(t) + x_3(t) + (b_2 - a_2\, b_0) u(t) \\
\vdots \quad &= \qquad\qquad \vdots \\
\dot{x}_n(t) &= -a_n x_1(t) + (b_n - a_n\, b_0) u(t).
\end{aligned} \tag{3.44}$$

The above equation together with (3.42) constitutes a state-space description of a general n^{th}-order differential equation. This description can be written in the standard matrix form of (3.33) with the $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D)$ = $(\mathbf{A}_o,\ \mathbf{B}_o,\ \mathbf{C}_o,\ D_o)$, where

$$\mathbf{A}_o = \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ -a_3 & 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 \\ -a_n & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \mathbf{B}_o = \begin{bmatrix} b_1 - a_1 b_0 \\ b_2 - a_2 b_0 \\ b_3 - a_3 b_0 \\ \vdots \\ b_n - a_n b_0 \end{bmatrix},$$
$$\mathbf{C}_o = [1\ 0\ \cdots\ 0], \qquad D_o = b_0. \tag{3.45}$$

The state-space model in (3.45) form is called *observable canonical form* for reasons that will be explained later. The coefficients a_i and b_i in this canonical form are the same coefficients that appear in the differential equation (3.39) and in the transfer function (3.15).

EXAMPLE 3.3

Find the state-space model in observable canonical form for the following transfer function

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

We simply read the coefficients from the numerator and denominator polynomials of $H(s)$ using the notation given in (3.4) to obtain

$$b_0 = 2\,,\;\; b_1 = 4\,,\;\; b_2 = 4$$

and

$$a_1 = 2\,,\;\; a_2 = 0.$$

These coefficients are substituted into (3.45) to obtain

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -2 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 4 \end{bmatrix} u(t)$$

$$y(t) = [1 \;\; 0]\, \mathbf{x}(t) + 2\, u(t).$$

■

EXAMPLE 3.4

Find the observable canonical form state-space model for the following transfer function

$$H(s) = \frac{s + 2}{s^3 + 7s^2 + 16s + 10}.$$

We read off the following coefficients from the transfer function

$$b_0 = 0\,,\;\; b_1 = 0,\;\; b_2 = 1\,,\;\; b_3 = 2$$

and

$$a_1 = 7\,,\;\; a_2 = 16\,,\;\; a_3 = 10.$$

These coefficients are substituted into (3.45) to obtain

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -7 & 1 & 0 \\ -16 & 0 & 1 \\ -10 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} u(t)$$

$$y(t) = [1 \;\; 0 \;\; 0]\, \mathbf{x}(t).$$

Notice that when the numerator coefficient $b_0 = 0$, the input vector $\mathbf{b}_o$ is comprised of the numerator coefficients $b_1, \cdots, b_n$ of the transfer function, some of which may be zero.

■

EXAMPLE 3.5

The transfer function from input voltage $u(t)$ to output angular position $y(t)$ for a dc motor is given in Example 3.2 on page 95 and is shown below:

$$G(s) = \frac{Kb_1}{s^s + a_1 s}.$$

The observable canonical from state-space model corresponding to this transfer function is

$$\mathbf{A}_o = \begin{bmatrix} -a_1 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{b}_o = \begin{bmatrix} 0 \\ Kb_1 \end{bmatrix}, \quad \mathbf{c}_o = [1 \quad 0], \quad d_o = 0.$$

The careful reader will note that the matrix $\mathbf{A}_o$ given above and the $\mathbf{A}$ matrix given in Example 3.2 are not the same. This is not a mistake! To begin to reconcile this apparent discrepancy let us find the relationship between the state variables of observable canonical form and the state variables used in Example 3.2. The state variables used in Example 3.2 were

$$\begin{aligned} x_1 &= \text{ angular position} \\ x_2 &= \text{ angular velocity.} \end{aligned}$$

The state variables for observable canonical form will be denoted by $\bar{x}_1, \bar{x}_2$. It is always true for observable canonical form that the output is equal to the first state variable. The output of the motor system is angular position, and so

$$\bar{x}_1(t) \;=\; \text{angular position.}$$

The following equation for $\dot{\bar{x}}_1(t)$ can be obtained from the first state equation (the first row of $\mathbf{A}_o$ and $\mathbf{b}_o$):

$$\dot{\bar{x}}_1(t) = -a_1 \bar{x}_1(t) + \bar{x}_2(t).$$

We can solve for $\bar{x}_2(t)$ and use the fact that $\bar{x}_1(t) =$ angular position to get

$$\bar{x}_2(t) = a_1(\text{angular position}) + \text{ angular velocity.}$$

Thus the state variables for observable canonical form are not the same as the state variables used in Example 3.2 and so it is not surprising that the state-space models are different.

For future reference we note that the following relationship exists between the state variables of the two models:

$$\begin{bmatrix} \bar{x}_1(t) \\ \bar{x}_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ a_1 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}.$$

The matrix which relates the two sets of state variables is nonsingular. The general relationship between two state-space models whose state variables are related by a nonsingular matrix is derived in Section 3.3.4.

■

3.3.2 A Discrete-Time Canonical Form

In this section we derive the observable canonical form for discrete-time systems. It will be seen that the construction of this canonical form is in direct analogy to the continuous-time derivation given in the last section.

The fundamental operation for constructing a system which solves difference equations is the *unit delay* element, which is represented by the z-transform symbol z^{-1} (see Table 3.3 on page 86). In other words, with zero initial conditions

$$Z\{y[k-1]\} = z^{-1}Z\{y[k]\}.$$

The fundamental tool to be used in deriving the observable canonical form is the string of n summers and delay elements shown in Fig. 3.8.

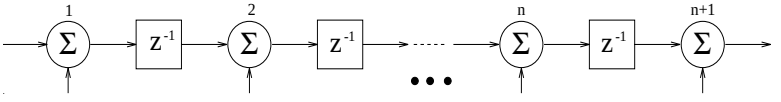


Figure 3.8 Cascade of n delay elements and $n+1$ summing junctions.

Notice that signals which are introduced at different summing junctions will experience different delays before reaching the output. In particular, signals which are introduced at summing junction 1 will be delayed n times before reaching the output. Signals which are introduced at summing junction 2 will be delayed $n-1$ times before reaching the output. Finally, signals which are introduced at summing junction n will be delayed once before reaching the output. Thus the system shown in Fig. 3.8 can produce sums of delayed signals with delays ranging from 1 to n. This is all that is needed to solve a general n^{th}-order difference equation. Such an equation is

$$y[k] + a_1 y[k-1] + \cdots + a_n y[k-n] = b_0 u[k] + b_1 u[k-1] + \cdots + b_n u[k-n], \quad (3.46)$$

and solving for $y[k]$, we obtain

$$y[k] = -a_1 y[k-1] - \cdots - a_n y[k-n] + b_0 u[k] + b_1 u[k-1] + \cdots + b_n u[k-n]. \quad (3.47)$$

From the above equation, it can be seen that $y[k]$ is formed from a linear combination of signals that are delayed up to n steps. Thus $y[k]$ could be the output of the system shown in Fig. 3.8 if the signals are introduced at the appropriate summing junctions. In particular, if the signals $b_n u[k]$ and $-a_n y[k]$ are introduced at summing junction 1, they will be delayed n steps as required by (3.47). If the signals $b_{n-1}u[k]$ and $-a_{n-1}y[k]$ are introduced at summing junction 2, they will be delayed $n-1$ time steps as required by (3.47). This process is repeated, until we come to the term $b_0 u[k]$ which is not delayed at all, and so it gets added in at summing junction $n+1$. The resulting system is shown in Fig. 3.9.

In order to write the state-variable equations corresponding to Fig. 3.9, we need to express the state variables at time $k+1$ in terms of the state variables at time k and the input at time k. These equations can be obtained from Fig. 3.9. For continuous-time systems the state variables were defined to be the outputs of the integrators. The analogous choice for a discrete-time system is to choose the state variables at time k to be the outputs of the delay elements. With this choice of state variables, the inputs to the delay elements are the state variables at time $k+1$. We also notice from Fig. 3.9 that the output equation is simply

$$y[k] = x_1[k] + b_0 u[k]. \quad (3.48)$$

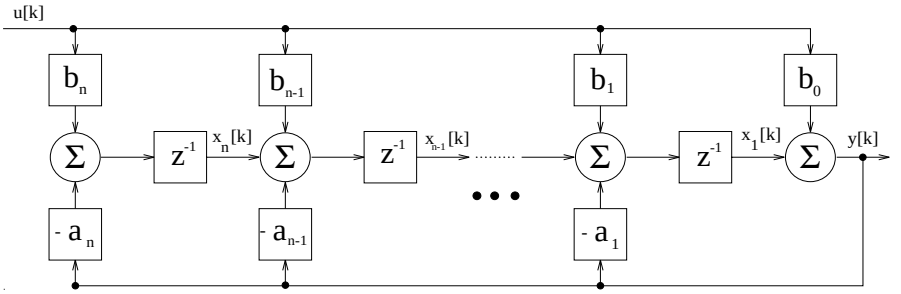


Figure 3.9 A system which solves a general n^{th}-order difference equation. State variable $x_1[k] \cdots x_n[k]$ are shown as the outputs of the delay elements.

To obtain the state equations, we start with the last delay element, which from Fig. 3.9 gives

$$x_1[k+1] = -a_1 y[k] + x_2[k] + b_1 u[k], \tag{3.49}$$

and substituting (3.48) into (3.49), we obtain

$$x_1[k+1] = -a_1 x_1[k] + x_2[k] + (b_1 - a_1\, b_0)u[k].$$

A similar procedure can be followed for all the delay elements in Fig. 3.9, resulting in the following state-variable equations

$$\begin{aligned} x_1[k+1] &= -a_1 x_1[k] + x_2[k] + (b_1 - a_1\, b_0)u[k] \\ x_2[k+1] &= -a_2 x_1[k] + x_3[k] + (b_2 - a_2\, b_0)u[k] \\ \vdots \quad &= \qquad \vdots \\ x_n[k+1] &= -a_n x_1[k] + (b_n - a_n\, b_0)u[k]. \end{aligned} \tag{3.50}$$

The above equation together with (3.48) constitutes a state-space description of a general n^{th}-order difference equation. This description can be written in the standard matrix form of (3.38) with the $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D)$ = $(\mathbf{A}_o,\ \mathbf{B}_o,\ \mathbf{C}_o,\ D_o)$, where

$$\mathbf{A}_o = \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ -a_3 & 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 \\ -a_n & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \mathbf{B}_o = \begin{bmatrix} b_1 - a_1 b_0 \\ b_2 - a_2 b_0 \\ b_3 - a_3 b_0 \\ \vdots \\ b_n - a_n b_0 \end{bmatrix} \tag{3.51}$$

$$\mathbf{C}_o = [1\, 0\, \cdots\, 0]\,, \qquad D_o = b_0.$$

The discrete-time and continuous-time observable canonical forms are expressed identically in terms of the transfer function coefficients a_i and b_i.

EXAMPLE 3.6

Find the observable canonical form state-space model for the following transfer function

$$H(z) = \frac{2z^2 + 4z + 4}{z^2 + 2z}.$$

We simply read the coefficients from the numerator and denominator polynomials of $H(z)$ using the notation given in (3.15) to obtain

$$b_0 = 2\,,\;\; b_1 = 4\,,\;\; b_2 = 4$$

and

$$a_1 = 2\,,\;\; a_2 = 0.$$

These coefficients are substituted into (3.51) to obtain

$$\mathbf{x}[k+1] = \begin{bmatrix} -2 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} 0 \\ 4 \end{bmatrix} u[k]$$

$$y[k] = [1 \;\; 0]\,\mathbf{x}[k] + 2\,u[k].$$

Note the similarity between this example and Example 3.3 on page 99.

■

3.3.3 Other Canonical Forms

It will be shown in section 3.3.4 that state-space models corresponding to a given transfer function, or equivalently, to a given differential or difference equation, are not unique. In the previous section, we derived the observable canonical form state-space description. In this section, we derive a different canonical form for a state-space model of an n^{th}-order system.

Continuous-Time Systems Consider the following n^{th}-order differential equation description of a continuous-time system. Assume that the equation is true for $t \geq 0$, and that all initial conditions are equal to zero.

$$w^{(n)}(t) = -a_1 w^{(n-1)}(t) - \cdots - a_n w(t) + u(t). \tag{3.52}$$

Note that this is not the most general form of a differential equation, because it does not contain derivatives of the input. The general case will be considered shortly. If this equation could be represented as n first-order differential equations (a state-space model), then each first-order equation could be solved by a single integration. Thus to derive a state-space model from (3.52), we consider how to express the solution in terms of n single integrations.

We can draw a block diagram of a system which solves (3.52) using integrators. First, assume that the function $w^{(n)}(t)$ is known. Then this signal can be integrated n times, using n integrating devices in series. The outputs of the n successive integrators are the functions $w^{(n-1)}(t)$, $\cdots$, $w(t)$. These functions can be combined with the input $u(t)$ according to (3.52) to produce $w^{(n)}(t)$, thus validating the assumption that it is known. This discussion leads to the block diagram in Fig. 3.10 for a system which solves (3.52).

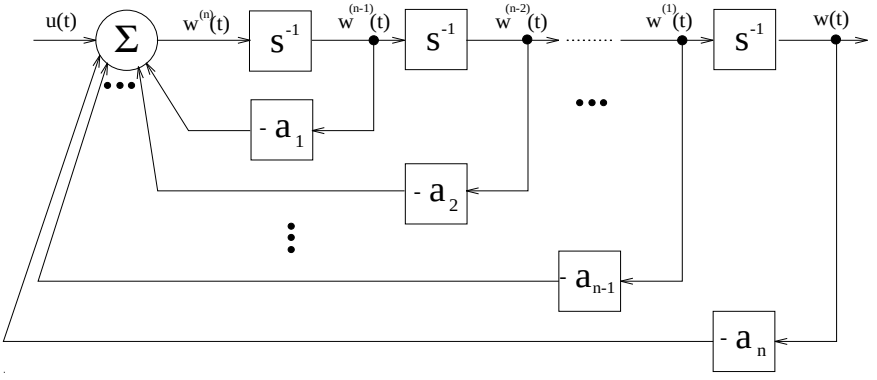


Figure 3.10 A system which solves equation (3.52) using integrating devices.

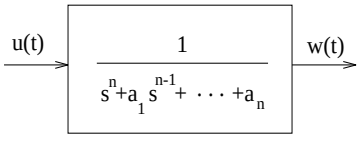


Figure 3.11 Transfer function of system described by equation (3.52).

Recall that (3.52) is not the most general form of a differential equation. We can also see this by looking at the transfer function corresponding to (3.52), which is shown in the Fig. 3.11. It can be seen from Fig. 3.11 that (3.52) corresponds to a system with no finite zeros, since the numerator of the transfer function is a constant. (The transfer function does have zeros at infinity, since when $s \to \infty$, the transfer function goes to zero.) Let us now consider the case of the general n^{th}-order differential equation

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots + a_n y(t) = b_0 u^{(n)}(t) + b_1 u^{(n-1)}(t) + \cdots + b_n u(t). \quad (3.53)$$

The transfer function corresponding to this equation is shown in Fig. 3.12, along with a useful decomposition of the transfer function.

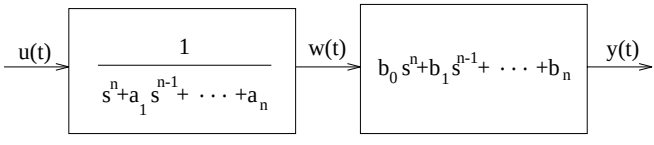


Figure 3.12 Transfer function of system described by equation (3.11).

Note that the first half of Fig. 3.12 is identical to Fig. 3.11, which means that $w(t)$ can be obtained from $u(t)$ by the system shown in Fig. 3.10. The transfer function in Fig. 3.12 from $w(t)$ to $y(t)$ consists of a chain of differentiators (recall from Table 3.1 that the Laplace transform of a differentiator is simply s). Putting these two observations together, we obtain a system which solves the general differential equation. This system, which is shown in Fig. 3.13, contains both integrators and differentiators.

The use of differentiators would be undesirable from a practical point of view if Fig. 3.13 were an actual circuit used to solve a differential equation, with $u(t)$ and $y(t)$ represented

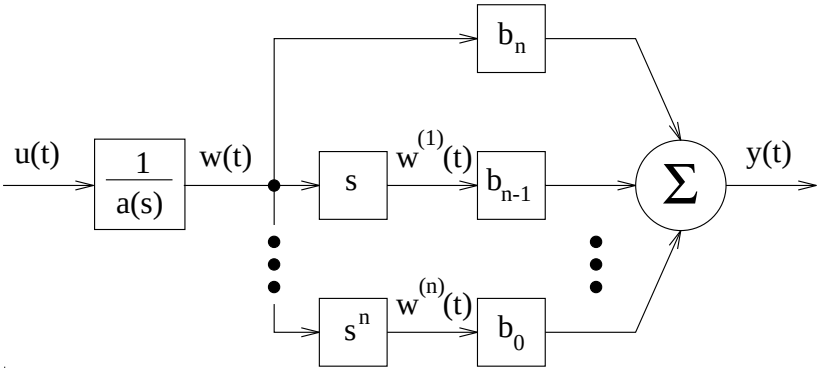


Figure 3.13 A system which solves a general differential equation using differentiators and integrators.

as voltage waveforms, for example. The reason is that the differentiators would accentuate any high frequency noise in the signal. From a theoretical point of view, the differentiators should be unnecessary because an n^{th}-order differential equation should only require n integrations for its solution.

By close inspection of Fig. 3.13 and Fig. 3.10, we can see that there is a way to eliminate the differentiators. Notice that the outputs of the differentiators are derivatives of $w(t)$. But these derivatives have already been constructed as the *inputs* to the integrators. Thus the differentiators can be eliminated, and we can just put lines into the system which take the derivatives of $w(t)$ from the inputs to the integrators to the required places in the second half of the system. The resulting system is shown in Fig. 3.14.

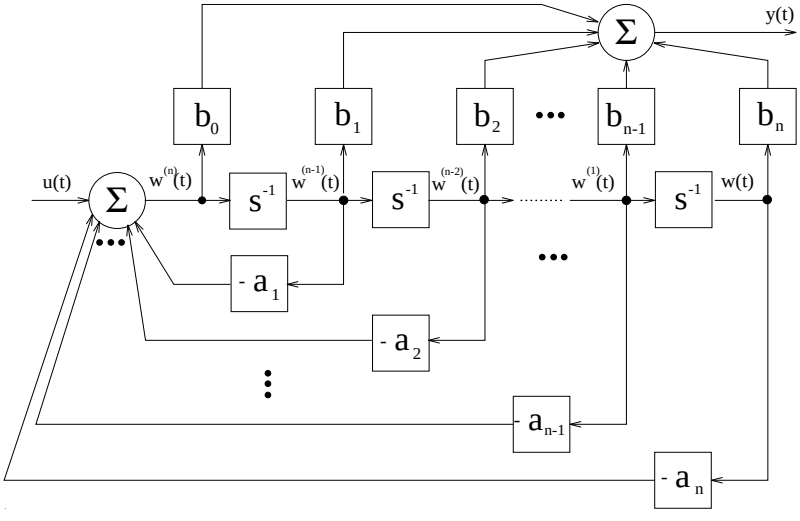


Figure 3.14 A system which solves a general differential equation using only integrators.

A system which solves a differential equation using only integrators is equivalent to a state-space model for that equation. It only remains to define the state variables, which we

take to be the outputs of the integrators. Thus in Fig. 3.14, the state variables are defined as follows:

$$x_1(t) = w^{(n-1)}(t),\ x_2(t) = w^{(n-2)}(t),\ \cdots,\ x_n(t) = w(t). \tag{3.54}$$

Using the "dot" notation for the derivative with respect to time, the definitions in (3.54) can be rewritten as

$$\dot{x}_n(t) = x_{n-1}(t),\ \dot{x}_{n-1}(t) = x_{n-2}(t), \cdots,\ \dot{x}_2(t) = x_1(t), \tag{3.55}$$

and from Fig. 3.14, the equations for $\dot{x}_1(t)$ and $y(t)$ can be obtained:

$$\begin{aligned} \dot{x}_1(t) &= -a_1x_1(t) - a_2x_2(t) - \cdots - a_nx_n(t) + u(t) \\ y(t) &= (b_1 - a_1b_0)x_1(t) + \cdots + (b_n - a_nb_0)x_n(t) + b_0u(t). \end{aligned} \tag{3.56}$$

The last two equations can be combined and rewritten in the standard matrix form of (3.33) where

$$A_c = \begin{bmatrix} -a_1 & -a_2 & -a_3 & \cdots & -a_n \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \qquad \mathbf{B}_c = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.57}$$

$$\mathbf{C}_c = [b_1 - a_1b_0 \ \ b_2 - a_2b_0 \ \ b_3 - a_3b_0 \ \ \cdots \ \ b_n - a_nb_0], \qquad D_c = b_0.$$

Note that (3.57) is a canonical form (called *controllable canonical form* for reasons to be explained later) because $(A_c, \mathbf{b}_c, \mathbf{c}_c, d_c)$ are functions of the $2n + 1$ transfer function (or differential equation) coefficients. Thus (3.57) can represent any n^{th}-order system by appropriately choosing the coefficients.

Discrete-Time Systems We now consider the derivation of the controllable canonical form for discrete-time systems.

We begin with a special difference equation with input $\{u[k]\}$ and output $\{w[k]\}$ which does not contain shifts of the input sequence

$$w[k] = -a_1w[k-1] - a_2w[k-2] - \cdots - a_nw[k-n] + u[k]. \tag{3.58}$$

A system which solves this difference equation is shown in Fig. 3.15. The z-transform of (3.58) shows that

$$\frac{W(z)}{U(z)} = \frac{1}{1 + a_1z^{-1} + \cdots + a_nz^{-n}} = \frac{1}{A(z)} \tag{3.59}$$

and thus the system in Fig. 3.15 may be represented by the transfer function $1/A(z)$.

Now consider the general difference equation

$$y[k] + a_1y[k-1] + \cdots + a_ny[k-n] = b_0u[k] + b_1u[k-1] + \cdots + b_nu[k-n]. \tag{3.60}$$

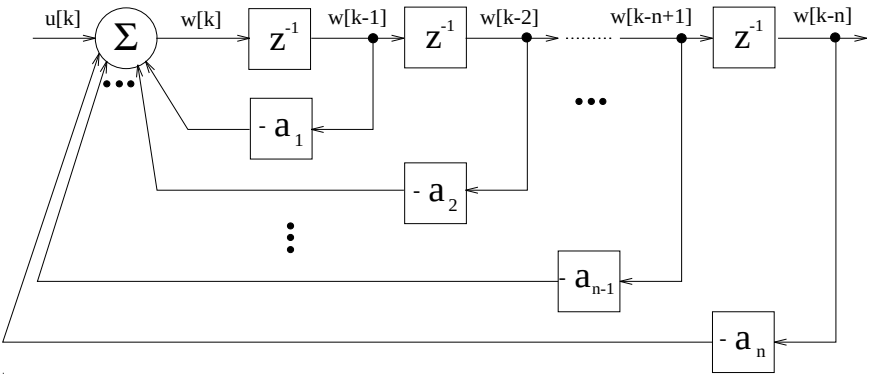


Figure 3.15 A system which solves the difference equation (3.58).

The z-transform of this equation leads to the transfer function

$$\frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_n z^{-n}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}} = \frac{B(z)}{A(z)} = \frac{1}{A(z)} B(z). \tag{3.61}$$

From (3.59) we have that

$$W(z) = \frac{1}{A(z)} U(z)$$

and using (3.61) we have

$$Y(z) = B(z)W(z) = (b_0 + b_1 z^{-1} + \cdots + b_n z^{-n}) W(z).$$

Taking the inverse z-transform of the above equation yields

$$y[k] = b_0 w[k] + b_1 w[k-1] + \cdots + b_n w[k-n]. \tag{3.62}$$

Thus $y[k]$, the solution to the general difference equation (3.60), can be formed as a linear combination of $w[k], \cdots, w[k-n]$. Notice that the system in Fig. 3.15 produces the required $w[\cdot]$ sequences. We simply have to form the linear combination shown in (3.62) to obtain a system which solves the general n^{th}-order difference equation. The resulting system is shown in Fig. 3.16.

A system which solves a difference equation using only delays is equivalent to a state-space model for that equation. We define the state variables to be the outputs of the delays. Thus in Fig. 3.16, the state variables are defined as follows:

$$x_1[k] = w[k-1],\ x_2[k] = w[k-2],\ \cdots,\ x_n[k] = w[k-n]. \tag{3.63}$$

From this definition of the state variables, it is clear that the variables at time $k+1$ can be obtained in terms of the variables at time k in a simple manner, except for the first one.

$$x_n[k+1] = x_{n-1}[k],\ x_{n-1}[k+1] = x_{n-2}[k],\ \cdots,\ x_2[k+1] = x_1[k]. \tag{3.64}$$

From Fig. 3.16, the equations for $x_1[k+1]$ and $y[k]$ can be obtained

$$\begin{aligned} x_1[k+1] &= -a_1 x_1[k] - a_2 x_2[k] - \cdots - a_n x_n[k] + u[k] \\ y[k] &= (b_1 - a_1 b_0) x_1[k] + \cdots + (b_n - a_n b_0) x_n[k] + b_0 u[k]. \end{aligned} \tag{3.65}$$

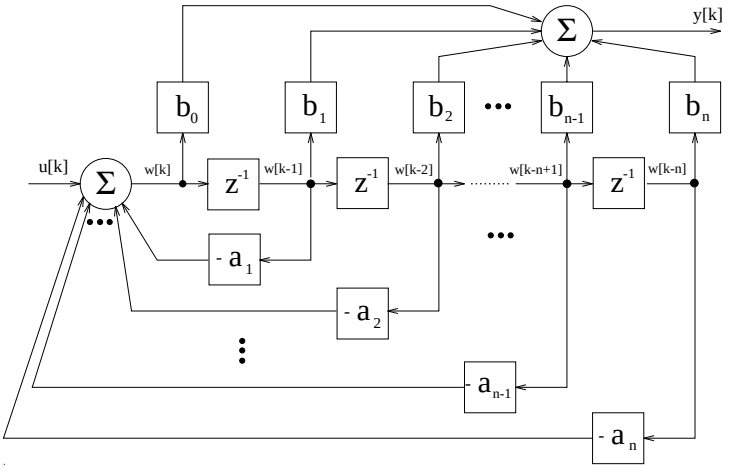


Figure 3.16 A system which solves equation (3.60) using only delays.

The last two equations can be combined and rewritten in the standard matrix form of (3.38). It turns out that the matrices for the discrete-time controllable canonical form have identical forms to those for the continuous-time controllable canonical form shown in (3.57) on page 106.

3.3.4 Linear Transformations and Transfer Functions

A few basic facts about state-space models are collected in this section. This section should be referred to as the need arises when reading later chapters. The results derived in this section are summarized in Table 3.7 on page 132.

Linear Transformations In the previous section, two different canonical forms were derived for continuous- and discrete-time systems. This shows that there is not a unique state-space model associated with a given differential (or difference) equation. There is, however, a unique transfer function associated with such an equation, if pole-zero cancellations are not allowed.

The lack of uniqueness of state-space models raises the following questions. How many different n^{th}-order state-space models can describe the same system? Given a state-space model for a system, how can different state-space models be obtained? In this section we show that there are an infinite number of state-space models that can describe a given system, and they are all related to one another by a linear algebraic transformation. This transformation is described below.

Consider the following discrete-time system (the development for continuous-time systems is completely analogous and is left as an exercise).

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{A}\,\mathbf{x}[k] + \mathbf{B}\,u[k] \\ y[k] &= \mathbf{C}\,x[k] + D\,u[k]. \end{aligned} \tag{3.66}$$

Let $\mathbf{T}$ be any nonsingular $n \times n$ matrix. Define a new state vector $\bar{\mathbf{x}}[k]$ from the state vector $\mathbf{x}[k]$ in (3.66) as follows

$$\bar{\mathbf{x}}[k] = \mathbf{T}\mathbf{x}[k], \text{ or } \mathbf{x}[k] = \mathbf{T}^{-1}\bar{\mathbf{x}}[k]. \tag{3.67}$$

Let us now briefly consider what the above equation means. The matrix $\mathbf{T}$ has elements

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ t_{n1} & t_{n2} & \cdots & t_{nn} \end{bmatrix}.$$

Thus if we look at the i^{th} row of the first equation in (3.67), we see that the new state variable $\bar{x}_i$ is just a linear combination of the old state variables:

$$\bar{x}_i = \sum_{j=1}^{n} t_{ij} x_j,$$

where the coefficients t_{ij} come from the i^{th} row of the matrix $\mathbf{T}$. The new state variables are just linear combinations of the old state variables. The reason why $\mathbf{T}$ must be a nonsingular matrix is the following. If $\mathbf{T}$ were a singular matrix, then at least one of its rows, say the j^{th} one, could be expressed as a linear combination of the other rows (see Chapter 2). In this case, the new state variable $\bar{x}_j[k]$ could be expressed by the same linear combination of the other $\bar{\mathbf{x}}[k]$ variables. In other words, the new set of state variables would contain a redundant variable ($\bar{x}_j[k]$) even if the original set of state variables were independent. Thus the original state variables could not be obtained from the transformed state variables. To avoid this situation, the matrix $\mathbf{T}$ must be nonsingular. We now return to the development of the linear transformation.

Substituting (3.67) into (3.66) yields

$$\begin{aligned} \mathbf{T}^{-1}\bar{\mathbf{x}}[k+1] &= \mathbf{A}\mathbf{T}^{-1}\bar{\mathbf{x}}[k] + \mathbf{B}u[k] \\ y[k] &= \mathbf{C}\,\mathbf{T}^{-1}\bar{\mathbf{x}}[k] + Du[k] \end{aligned}$$

or

$$\begin{aligned} \bar{\mathbf{x}}[k+1] &= \mathbf{T}\mathbf{A}\mathbf{T}^{-1}\bar{\mathbf{x}}[k] + \mathbf{T}\,\mathbf{B}u[k] \\ y[k] &= \mathbf{C}\,\mathbf{T}^{-1}\bar{\mathbf{x}}[k] + Du[k]. \end{aligned} \tag{3.68}$$

It can be seen that the above equation is a new state-space description $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{D})$ for the original system which relates the same input sequence $\{u[k]\}$ and output sequence $\{y[k]\}$ as the state-space description given in (3.66), where

$$\begin{aligned} \bar{\mathbf{A}} &= \mathbf{T}\mathbf{A}\mathbf{T}^{-1} \\ \bar{\mathbf{B}} &= \mathbf{T}\,\mathbf{B} \\ \bar{\mathbf{C}} &= \mathbf{C}\,\mathbf{T}^{-1} \\ \bar{D} &= D. \end{aligned} \tag{3.69}$$

The above equation follows from the relationship between $\mathbf{x}[k]$ and $\bar{\mathbf{x}}[k]$ given in (3.67). Equation (3.69) is also true for continuous-time systems whose state vectors are related by $\bar{\mathbf{x}}(t) = \mathbf{T}\mathbf{x}(t)$.

EXAMPLE 3.7

Two different state-space models for a dc motor were given in Example 3.2 on page 95 and Example 3.5 on page 100. The models are shown below. The model from Example 3.2 is $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ where

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -a_1 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ Kb_1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(t).$$

The model from Example 3.5 is $(\mathbf{A}_o, \mathbf{B}_o, \mathbf{C}_o)$ where

$$\dot{\bar{\mathbf{x}}}(t) = \begin{bmatrix} -a_1 & 1 \\ 0 & 0 \end{bmatrix} \bar{\mathbf{x}}(t) + \begin{bmatrix} 0 \\ Kb_1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \bar{\mathbf{x}}(t).$$

The relationship between $\bar{\mathbf{x}}(t)$ and $\mathbf{x}(t)$ was found in Example 3.5 to be

$$\bar{\mathbf{x}}(t) = \begin{bmatrix} 1 & 0 \\ a_1 & 1 \end{bmatrix} \mathbf{x}(t).$$

The reader can verify that

$$\mathbf{A}_o = \mathbf{TAT}^{-1}$$

$$\mathbf{B}_o = \mathbf{TB}$$

$$\mathbf{C}_o = \mathbf{CT}^{-1}.$$

■

EXAMPLE 3.8

Consider the following state-space description

$$\mathbf{x}[k+1] = \begin{bmatrix} -2 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u[k]$$

$$y[k] = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}[k] + 1\, u[k].$$

Suppose a new vector of state variables is defined as

$$\bar{\mathbf{x}}[k] = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \mathbf{x}[k].$$

We first compute

$$\mathbf{T}^{-1} = \begin{bmatrix} -2 & 1 \\ 1.5 & -0.5 \end{bmatrix}.$$

Then we use the formulas in (3.69) to obtain the new state-space description

$$\bar{\mathbf{x}}[k+1] = \begin{bmatrix} 5.5 & -2.5 \\ 16.5 & -7.5 \end{bmatrix} \bar{\mathbf{x}}[k] + \begin{bmatrix} 3 \\ 7 \end{bmatrix} u[k]$$

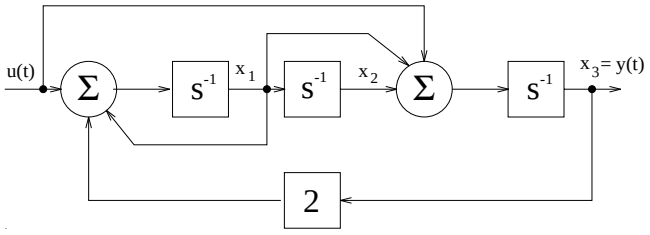
$$y[k] = \begin{bmatrix} -2 & 1 \end{bmatrix} \bar{\mathbf{x}}[k] + 1\, u[k].$$

It is important to note that both state-space descriptions describe the same system taking a given input sequence $u[k]$ to the same output sequence $y[k]$, but each has a different choice of state variables.

■

EXAMPLE 3.9

Consider the system described by the block diagram below

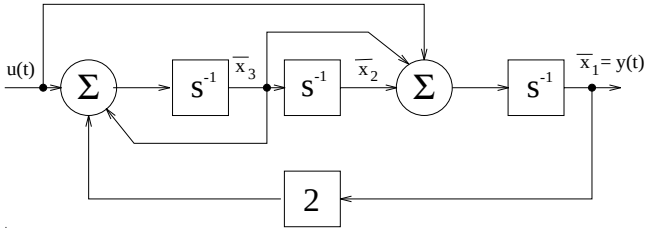


Suppose we define state variables for this system as the outputs of the integrators numbered left to right. That is, the output of the leftmost integrator is $x_1(t)$ while the output of the rightmost integrator is $x_3(t)$. With this choice of state variables we can obtain a state-space description directly from the diagram as follows

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} u(t) \quad = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t)$$

$$y(t) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \quad = \mathbf{C}\mathbf{x}(t).$$

Now suppose we define a new set of state variables by numbering the state variables from right to left; that is, let $\bar{x}_1(t)$ be the output of the rightmost integrator, and let $\bar{x}_3(t)$ be the output of the leftmost integrator as shown in the diagram below.



With this choice of state variables we can obtain a state-space description directly from the diagram as follows

$$\begin{bmatrix} \dot{\bar{x}}_1(t) \\ \dot{\bar{x}}_2(t) \\ \dot{\bar{x}}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{x}_1(t) \\ \bar{x}_2(t) \\ \bar{x}_3(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} u(t) \quad = \bar{\mathbf{A}}\bar{\mathbf{x}}(t) + \bar{\mathbf{B}}u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_1(t) \\ \bar{x}_2(t) \\ \bar{x}_3(t) \end{bmatrix}. \quad = \bar{\mathbf{C}}\bar{\mathbf{x}}(t).$$

Notice that the relationship between the first set of state variables and the second set is

$$\bar{x}_1(t) = x_3(t)\ ,\ \ \bar{x}_2(t) = x_2(t)\ ,\ \ \bar{x}_3(t) = x_1(t).$$

The above equations can be written as a single matrix equation which describes the new state vector in terms of the original state vector

$$\begin{bmatrix} \bar{x}_1(t) \\ \bar{x}_2(t) \\ \bar{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

or

$$\bar{\mathbf{x}}(t) = \mathbf{T}\mathbf{x}(t).$$

The above equation tells us that the new system is a linear transformation of the original system, and so the formulas in (3.69) can be used. In particular, it can be verified that with the matrices defined above, the following equations are true

$$\begin{aligned} \bar{\mathbf{A}} &= \mathbf{T}\mathbf{A}\mathbf{T}^{-1} \\ \bar{\mathbf{B}} &= \mathbf{T}\ \mathbf{B} \\ \bar{\mathbf{C}} &= \mathbf{C}\ \mathbf{T}^{-1} \end{aligned}$$

■

3.3.4.1 From State-Space Representations to Transfer Functions Consider the following transfer function

$$H(w) = \frac{b_0 w^n + b_1 w^{n-1} + \cdots + b_n}{w^n + a_1 w^{n-1} + \cdots + a_n} \tag{3.70}$$

where $w = s$ for continuous-time systems and $w = z$ for discrete-time systems. From previous results, we know that the corresponding differential (or difference) equation relating the input and output can be written directly in terms of the coefficients a_i and b_i. Using these same coefficients, a canonical state-space representation can be written down by inspection. In this section, we consider the reverse problem of starting with a state-space description, and deriving the corresponding transfer function. Of course, if the given state-space description is in a canonical form, then we can just read off the coefficients of the transfer function directly. However, the question remains of how to obtain the transfer function if the given state-space representation is not in a canonical form.

The answer lies in the fact that the transfer function is equal to the ratio of the output transform to the input transform (Laplace or Z). It is possible to take the transform of state-space equations. To see this, we recall the general equations for continuous and discrete time.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,u(t) \quad \mathbf{x}[k+1] = \mathbf{A}\,\mathbf{x}[k] + \mathbf{B}\,u[k]$$
$$y(t) = \mathbf{C}\,x(t) + D\,u(t) \qquad y[k] = \mathbf{C}\,\mathbf{x}[k] + D\,u[k].$$

We are assuming zero initial conditions for these equations, which is required when computing the transfer function. If we take the Laplace transform of the first set of equations and the z-transform of the second set of equations, we get

$$\begin{aligned} w\mathbf{X}(w) &= \mathbf{A}\mathbf{X}(w) + \mathbf{B}\,U(w) \\ Y(w) &= \mathbf{C}\,\mathbf{X}(w) + D\,U(w) \end{aligned} \tag{3.71}$$

where $w = s$ for continuous-time systems and $w = z$ for discrete-time systems. Recall that the transform of a vector of functions is simply the transform of each element of the vector.

We now see a reason for the similarity between the continuous- and discrete-time results regarding transfer functions. In the transform domains, the Laplace variable s represents a derivative, while the z-transform variable z represents a shift. Since the only difference in form between continuous- and discrete-time state equations is that the continuous-time equations have a derivative while the discrete-time equations have a shift, this difference disappears when the appropriate transforms are taken.

We can solve (3.71) for $\mathbf{X}(w)$ in terms of $U(w)$ as $\mathbf{X}(w) = (w\mathbf{I} - \mathbf{A})^{-1}\,\mathbf{B}\,U(w)$ and substitute this into the output equation to get $Y(w)$. Dividing through by $U(w)$ yields

$$H(w) = \frac{Y(w)}{U(w)} = \mathbf{C}(w\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + D. \tag{3.72}$$

The above equation shows how the transfer function $H(w)$ can be computed from a state-space description $(\mathbf{A}, \mathbf{b}, \mathbf{c}, d)$. This formula is mainly of theoretical interest; it is not a convenient computational formula. The reason it is not good for computation is that it involves inverting the matrix $w\mathbf{I} - \mathbf{A}$ which contains the variable w. Computer programs for inverting matrices are usually written to invert matrices of numbers. There is an algorithm for calculating $(w\mathbf{I} - \mathbf{A})^{-1}$ using only matrices of numbers. The Fadeev algorithm shown in Table 3.5 recursively computes the coefficients of the characteristic polynomial in terms of the traces of certain matrices. The trace of a matrix $\mathbf{M}$, denoted $\mathrm{tr}(\mathbf{M})$, is just the sum of the diagonal elements of $\mathbf{M}$. The coefficients and matrices which the algorithm computes can be used to find $(w\mathbf{I} - \mathbf{A})^{-1}$.

Using the fact that the inverse of a matrix equals its adjoint divided by its determinant, we can rewrite (3.72) as

$$H(w) = \frac{\mathbf{C}\ \mathrm{adj}(w\mathbf{I} - \mathbf{A})\mathbf{B}}{\det(w\mathbf{I} - \mathbf{A})} + D.$$

Thus the poles of $H(w)$ are equal to the roots of $\det(w\mathbf{I} - \mathbf{A})$. Recall that these roots are by definition the *eigenvalues* of the matrix $\mathbf{A}$. This means that the *poles of a system are equal to the eigenvalues of the* $\mathbf{A}$ *matrix*. Since the denominator polynomial of $(w\mathbf{I} - \mathbf{A})^{-1}$ is the characteristic polynomial of $\mathbf{A}$, the Fadeev algorithm can also be used to calculate the characteristic polynomial of a matrix.

$a_1 = -\text{tr}(\mathbf{A})$	$\mathbf{S}_1 = \mathbf{I}$
$a_2 = -\frac{1}{2}\text{tr}(\mathbf{S}_2\mathbf{A})$	$\mathbf{S}_2 = \mathbf{S}_1\mathbf{A} + a_1\mathbf{I}$
$a_3 = -\frac{1}{3}\text{tr}(\mathbf{S}_3\mathbf{A})$	$\mathbf{S}_3 = \mathbf{S}_2\mathbf{A} + a_2\mathbf{I}$
$\vdots$	$\vdots$
$a_{n-1} = -\frac{1}{n-1}\text{tr}(\mathbf{S}_{n-1}\mathbf{A})$	$\mathbf{S}_n = \mathbf{S}_{n-1}\mathbf{A} + a_{n-1}\mathbf{I}$
$a_n = -\frac{1}{n}\text{tr}(\mathbf{S}_n\mathbf{A})$	$\mathbf{0} = \mathbf{S}_n\mathbf{A} + a_n\mathbf{I}$

The coefficients a_i and matrices $\mathbf{S}_i$ can also be used to find

$$(w\mathbf{I} - \mathbf{A})^{-1} = \frac{\mathbf{S}_1 w^{n-1} + \mathbf{S}_2 w^{n-2} + \cdots + \mathbf{S}_n}{w^n + a_1 w^{n-1} + \cdots + a_n}$$

Table 3.5 The Fadeev algorithm for computing the characteristic polynomial of $\mathbf{A}$ and also $(w\mathbf{I} - \mathbf{A})^{-1}$. The zero matrix computed in the last line of the algorithm is used as a check on the correctness of the calculations.

Given an arbitrary state-space description, the corresponding transfer function is given by (3.72). However, it was shown earlier that a state-space description could be transformed by a linear transformation matrix $\mathbf{T}$ without changing the external description of the system. Since a transfer function is an external description (relating input and output transforms), this means that different state-space descriptions which are related by linear transformation must have the same transfer function. This can be shown using (3.69) and (3.72) as follows. The transfer function of the original system $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D)$ is

$$H_1(w) = \mathbf{C}(w\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + D.$$

The transformed system is $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{D}) = (\mathbf{TAT}^{-1}, \mathbf{TB}, \mathbf{CT}^{-1}, D)$ which has transfer function

$$H_2(w) = \mathbf{CT}^{-1}(w\mathbf{I} - \mathbf{TAT}^{-1})^{-1}\mathbf{TB} + D.$$

This expression can be simplified as follows (recall $\mathbf{I} = \mathbf{TT}^{-1}$)

$$\begin{aligned} H_2(w) &= \mathbf{CT}^{-1}(w\mathbf{I} - \mathbf{TAT}^{-1})^{-1}\mathbf{TB} + D \\ &= \mathbf{CT}^{-1}[\mathbf{T}(w\mathbf{I} - \mathbf{A})\mathbf{T}^{-1}]^{-1}\mathbf{TB} + D \\ &= \mathbf{CT}^{-1}[\mathbf{T}(w\mathbf{I} - \mathbf{A})^{-1}\mathbf{T}^{-1}]\mathbf{TB} + D \\ &= \mathbf{C}(w\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + D \\ &= H_1(w) \end{aligned}$$

where we used the fact that if $\mathbf{A}_1$ and $\mathbf{A}_2$ are nonsingular matrices then $(\mathbf{A}_1\mathbf{A}_2)^{-1} = \mathbf{A}_2^{-1}\mathbf{A}_1^{-1}$. Thus the transfer function of the system obtained by linear transformation is identical to the transfer function of the original system.

One additional conclusion we can reach from this result regards the eigenvalues of $\bar{\mathbf{A}}$. Since the poles of a system are the eigenvalues of its $\mathbf{A}$ matrix and since $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D)$

and $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{D})$ have the same poles (they have the same transfer function, so they must have the same poles), we conclude that $\mathbf{A}$ and $\bar{\mathbf{A}} = \mathbf{TAT}^{-1}$ have the same eigenvalues. In other words *a similarity transformation on a matrix* $\mathbf{A}$ *does not change its eigenvalues*.

EXAMPLE 3.10

A state-space model for a dc motor was derived in Example 3.2 on page 95 and is shown below:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -a_1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ Kb_1 \end{bmatrix}, \quad \mathbf{C} = [\,1 \quad 0\,], \quad D = 0.$$

The transfer function can be calculated as follows

$$(s\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} s & -1 \\ 0 & s + a_1 \end{bmatrix}^{-1}$$

$$= \frac{1}{s^2 + a_1 s} \begin{bmatrix} s + a_1 & 1 \\ 0 & s \end{bmatrix}$$

and

$$H(s) = \frac{1}{s^2 + a_1 s} [\,1 \quad 0\,] \begin{bmatrix} s + a_1 & 1 \\ 0 & s \end{bmatrix} \begin{bmatrix} 0 \\ Kb_1 \end{bmatrix}$$

$$= \frac{Kb_1}{s^s + a_1 s}.$$

This result agrees with the transfer function given in Example 3.2. The reader can also verify that the observable canonical state-space model for the motor given in Example 3.5 on page 100 has the identical transfer function.

■

EXAMPLE 3.11

Consider the two state-space models derived in Example 3.9. We can compute their transfer functions using (3.72). For instance, the transfer function of $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is

$$H(s) = [\,0 \quad 0 \quad 1\,] \begin{bmatrix} s - 1 & 0 & -2 \\ -1 & s & 0 \\ -1 & -1 & s \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$= \frac{s^2 + 1}{s^3 - s^2 - 2s - 2}.$$

The matrix inverse could be computed using the Fadeev algorithm. The details of this calculation are given in the next example where we compute the transfer function of $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$ from Example 3.9.

$$\bar{\mathbf{A}} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 2 & 0 & 1 \end{bmatrix}, \quad \bar{\mathbf{B}} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad \bar{\mathbf{C}} = [\,1 \quad 0 \quad 0\,].$$

The Fadeev algorithm gives the following results

$$a_1 = -\text{tr}(\mathbf{A}) = -1$$

$$\mathbf{S}_2 = \mathbf{A} + a_1\mathbf{I} = \begin{bmatrix} -1 & 1 & 1 \\ 0 & -1 & 1 \\ 2 & 0 & 0 \end{bmatrix}$$

$$\mathbf{S}_2\mathbf{A} = \begin{bmatrix} 2 & -1 & 1 \\ 2 & 0 & 0 \\ 0 & 2 & 2 \end{bmatrix}$$

$$a_2 = -\frac{1}{2}\text{tr}(\mathbf{S}_2\mathbf{A}) = -2$$

$$\mathbf{S}_3 = \mathbf{S}_2\mathbf{A} + a_2\mathbf{I} = \begin{bmatrix} 0 & -1 & 1 \\ 2 & -2 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

$$\mathbf{S}_3\mathbf{A} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$$a_3 = -\frac{1}{3}\text{tr}(\mathbf{S}_3\mathbf{A}) = -2$$

$$\mathbf{0} = \mathbf{S}_3\mathbf{A} + a_3\mathbf{I} \quad \text{(it checks)}$$

So the characteristic polynomial is

$$a(s) = s^3 - s^2 - 2s - 2.$$

The matrices $\mathbf{S}_i$ can be used to form the numerator of $(s\mathbf{I} - \mathbf{A})^{-1}$ as shown in Table 3.5. Thus the coefficients in the numerator polynomial $b(s)$ of the transfer function are computed as

$$b_i = \bar{\mathbf{C}}\mathbf{S}_i\bar{\mathbf{B}} \ , \quad i = 1, 2, 3.$$

The results of these calculations are

$$b_1 = 1 \ , \ \ b_2 = 0 \ , \ \ b_3 = 1.$$

Thus the transfer function is

$$\frac{s^2 + 1}{s^3 - s^2 - 2s - 2}$$

which is the same as the transfer function of $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ given at the beginning of this example.

■

3.3.5 Obtaining State-Space Models

This subsection contains four examples of procedures for obtaining an overall state-space model for a system given some other type of description. The first two examples show how to obtain an overall state-space model from an interconnection of transfer functions. The given system in the third example is an interconnection of state-space models. Finally, the fourth example shows how to convert a set of differential equations into a state-space model.

EXAMPLE 3.12

Consider the block diagram of interconnected transfer functions shown in Fig. 3.17 The objective is to obtain a state-space model for the overall system from u to y. That is, a model for the system with input u and output y. Notice that the block on the right requires two state variables because the denominator polynomial is a second-order polynomial. The block on the the left requires one state variable. The figure shows that the right block uses the state variables $\{x_1, x_2\}$ while the left block uses $\{x_3\}$. This choice of state variables will lead to a specific state-space model derived below. However, there are other acceptable ways to assign the state variables. For example, the left block could use $\{x_1\}$ and the right block $\{x_2, x_3\}$. Another acceptable choice would be to use $\{x_2\}$ for the left block and $\{x_1, x_3\}$ for the right block (although this would be a strange choice!). Any acceptable choice of state variables will lead to a correct state-space model but each of these models will have different $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ matrices. This is due to the fact that state space models for a particular system are not unique. A particular system has an infinite number of state-space models, all of the same order, whose state-space matrices are related by the formulas given in Section 3.3.4.

The procedure for obtaining an overall state-space model from an interconnection of transfer functions has two steps. The first step is to use a canonical form (e.g OCF) to obtain state-space models for the individual blocks. The second step is to combine the individual state-space models into a single model for the overall system. The state vector for the overall system is just the concatenation (stacking together) of the state vectors for the individual blocks, as shown below.

Left Block

In order to use the *observable canonical form* (OCF - see (3.45)) for the left block in Fig. 3.17 we must first use the variable names from the generic transfer function shown in (3.4) for the numerical transfer function of the block. For the left block the result is as follows:

$$\frac{5}{s+5} \Longrightarrow \frac{b_0 = 0, b_1 = 5}{a_1 = 5}. \tag{3.73}$$

The OCF formulas yield the following state-space matrices for this block:

$$\mathbf{A} = -5, \;\; \mathbf{B} = 5, \;\; \mathbf{C} = 1, \;\; \mathbf{D} = 0. \tag{3.74}$$

Note that it is not sufficient to simply obtain $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ for each block. Rather, the complete state-space model for each block should be written out using the chosen state variables as well as the name of the input (and possibly output) of each block. For the left block the state vector is simply x_3 and the input is u; thus, the OCF state equation is

$$\dot{x}_3 = -5x_3 + 5u \tag{3.75}$$

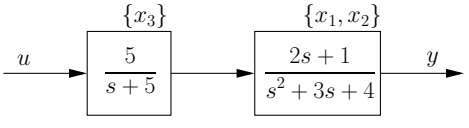


Figure 3.17 A cascade connection of two transfer functions for Example 3.12. The state-variables are indicated at the top of each block.

Because of the form of the OCF $\mathbf{C}$ and $\mathbf{D}$ matrices, the output of any OCF block equals the first state variable plus $\mathbf{D}$ times the block input. Whenever $\mathbf{D} = 0$ (which means $b_0 = 0$), the output of an OCF block is simply the first state variable of that block. Thus, the unlabelled signal between the two blocks in Fig. 3.17 should be labelled x_3.

Right Block

The numerical coefficients of the transfer function in the right block correspond to the variable names from the generic transfer function in (3.4) as follows:

$$\frac{2s+1}{s^2+3s+4} \Longrightarrow \frac{b_0 = 0, b_1 = 2, b_2 = 1}{a_1 = 3, a_2 = 4}. \tag{3.76}$$

The OCF formulas yield the following state-space matrices for this block:

$$\mathbf{A} = \begin{bmatrix} -3 & 1 \\ -4 & 0 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \ \mathbf{D} = 0. \tag{3.77}$$

From the analysis of the left block, we know that its output is x_3 so x_3 is the input to the right block; thus, the OCF state equation is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -3 & 1 \\ -4 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} x_3. \tag{3.78}$$

Because $b_0 = 0$ the output of the right block is its first state variable, which is x_1. The diagram shows that this signal is also the output y of the overall system; that is,

$$y = x_1. \tag{3.79}$$

In order to obtain the overall state-space model we use the state vector $\mathbf{x}$ that is the concatenation of the state vectors of the two blocks:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{3.80}$$

Using this state vector with (3.78), (3.75), and (3.79), the state-space model for the overall system from u to y is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -3 & 1 & 2 \\ -4 & 0 & 1 \\ 0 & 0 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} u \tag{3.81}$$

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

EXAMPLE 3.13

Consider the block diagram of interconnected transfer functions shown in Fig. 3.18. The objective is to obtain a state-space model for the overall system from r to y.

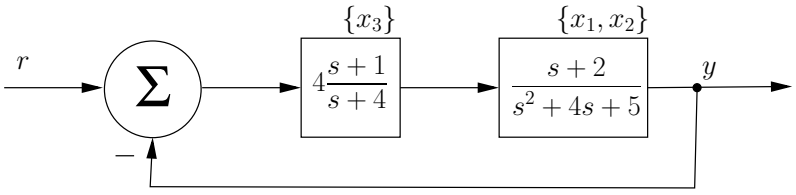


Figure 3.18 A cascade connection of two transfer functions for Example 3.13. The state-variables are indicated at the top of each block.

Right Block

The transfer function coefficients of the right block in Fig. 3.18 are associated with the variable names in the generic transfer function (3.4) as follows:

$$\frac{s+2}{s^2+4s+5} \Longrightarrow \frac{b_0 = 0, b_1 = 1, b_2 = 2}{a_1 = 4, a_2 = 5}. \tag{3.82}$$

The state-space matrices for the OCF state-space model (see (3.45)) of this block are

$$\mathbf{A} = \begin{bmatrix} -4 & 1 \\ -5 & 0 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \ \mathbf{D} = 0. \tag{3.83}$$

The input to the right block is the signal between the two blocks, which is not labelled. We can use the temporary variable w to as the name of the right-block input. The state equation for the right block is then

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -4 & 1 \\ -5 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} w. \tag{3.84}$$

The output of the right block is x_1 (OCF block with $b_0 = 0$), which is also the signal y, the output of the overall system. Thus,

$$y = x_1. \tag{3.85}$$

Left Block

The transfer function for the left block is first modified by multiplying the numerator coefficients by the number four before associating the resulting coefficients with the variable names from the generic transfer function in (3.4). The result is

$$\frac{4s+4}{s+4} \Longrightarrow \frac{b_0 = 4, b_1 = 4}{a_1 = 4}. \tag{3.86}$$

The state-space matrices for the OCF state-space model (see (3.45)) of this block are

$$\mathbf{A} = -4, \ \mathbf{B} = b_1 - a_1 \cdot b_0 = -12, \ \mathbf{C} = 1, \ \mathbf{D} = b_0 = 4 \tag{3.87}$$

The state variable for this block is x_3 and the input to the block is $r - y = r - x_1$. The state equation for the left block is then

$$\dot{x}_3 = -4x_3 - 12(r - x_1) \tag{3.88}$$

and the output of this block, which was called w when analyzing the right block, is obtained using the OCF output equation (3.45) with $\mathbf{C}$ and $\mathbf{D}$ given by (3.87). The result is

$$w = x_3 + 4(r - x_1) = -4x_1 + x_3 + 4r. \tag{3.89}$$

Substituting (3.89) into (3.84), the state-space model for the right block may be rewritten as follows:

$$\begin{aligned}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} &= \begin{bmatrix}-4 & 1\\ -5 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}1\\ 2\end{bmatrix}(-4x_1 + x_3 + 4r)\\ &= \begin{bmatrix}-8 & 1 & 1\\ -13 & 0 & 2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}4\\ 8\end{bmatrix} r.\end{aligned} \tag{3.90}$$

Combining (3.88) with the previous equation yields the overall state-space model

$$\begin{aligned}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} &= \begin{bmatrix}-8 & 1 & 1\\ -13 & 0 & 2\\ 12 & 0 & -4\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}4\\ 8\\ -12\end{bmatrix} r\\ y &= \begin{bmatrix}1 & 0 & 0\end{bmatrix}\mathbf{x}.\end{aligned} \tag{3.91}$$

EXAMPLE 3.14

Consider the block diagram of interconnected state-space models shown in Fig. 3.19. The objective is to obtain a state-space model for the overall system from r to y. The procedure for doing this is to first write the state equations for each block using the connections shown in the block diagram to find the input signals to each block. Then simplify the results and combine them into the overall state-space model. The state vector of the overall model will be the concatenation (stacking together) of all of the individual state vectors.

Block 2

From the diagram, we see that the input signal to Block 2 is $y_1 = \mathbf{C}_1\mathbf{x}_1 + \mathbf{D}_1 e$. The input to Block 1 is $e = r - y = r - y_2 = r - \mathbf{C}_2\mathbf{x}_2$. Substituting this into the previous equation gives $y_1 = \mathbf{C}_1\mathbf{x}_1 + \mathbf{D}_1(r - \mathbf{C}_2\mathbf{x}_2)$, which is the input to Block 2. Thus, the

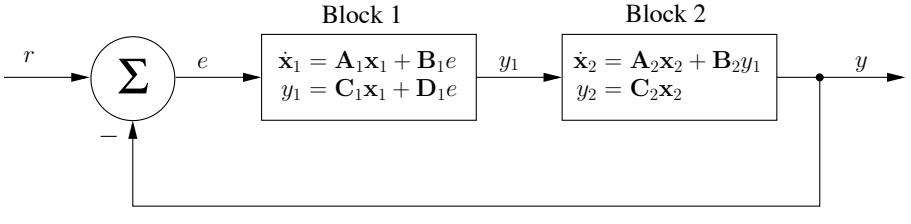


Figure 3.19 An interconnection of two state-space models for Example 3.14.

state-equation for Block 2 is

$$\begin{aligned}\dot{\mathbf{x}}_2 &= \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_2[\mathbf{C}_1\mathbf{x}_1 + \mathbf{D}_1(r - \mathbf{C}_2\mathbf{x}_2)] \\ &= \mathbf{B}_2\mathbf{C}_1\mathbf{x}_1 + (\mathbf{A}_2 - \mathbf{B}_2\mathbf{D}_1\mathbf{C}_2)\mathbf{x}_2 + \mathbf{B}_2\mathbf{D}_1 r.\end{aligned} \tag{3.92}$$

Block 1
The input to Block 2 is $r - y = r - \mathbf{C}_1\mathbf{x}_1$. Thus the state equation for Block 1 is

$$\begin{aligned}\dot{\mathbf{x}}_1 &= \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1(r - \mathbf{C}_2\mathbf{x}_2) \\ &= \mathbf{A}_1\mathbf{x}_1 - \mathbf{B}_1\mathbf{C}_2\mathbf{x}_2 + \mathbf{B}_1 r.\end{aligned} \tag{3.93}$$

Combining (3.92) and (3.93) yields the state-space model of the overall system from r to y:

$$\begin{aligned}\begin{bmatrix}\dot{x}_1 \\ \dot{\mathbf{x}}_2\end{bmatrix} &= \begin{bmatrix}\mathbf{A}_1 & -\mathbf{B}_1\mathbf{C}_2 \\ \mathbf{B}_2\mathbf{C}_1 & (\mathbf{A}_2 - \mathbf{B}_2\mathbf{D}_1\mathbf{C}_2)\end{bmatrix}\begin{bmatrix}\mathbf{x}_1 \\ \mathbf{x}_2\end{bmatrix} + \begin{bmatrix}\mathbf{B}_1 \\ \mathbf{B}_2\mathbf{D}_1\end{bmatrix} r \\ y &= \begin{bmatrix}\mathbf{0} & \mathbf{C}_2\end{bmatrix}\begin{bmatrix}\mathbf{x}_1 \\ \mathbf{x}_2\end{bmatrix}.\end{aligned} \tag{3.94}$$

EXAMPLE 3.15

In the previous three examples an interconnection of transfer functions or state-space models was given and the goal was to find an overall state-space model. In this example, the system of interest is described by a set of two differential equations. A simpler example would contain only one differential equation. In addition to the differential equations a choice of state variables is also given. The choice of acceptable state variables for such problems is not always obvious. See Section 9.1.2 for a method for choosing state variables. To keep this example simple the choice of state variables is given.

Consider a system with input u that is defined by the following two differential equations in terms of the functions z_1 and z_2. Note that u, z_1 and z_2 are all functions of time.

$$\begin{aligned}\ddot{z}_1 + 2(\dot{z}_1 - \dot{z}_2) + 3z_2 &= u \\ \dot{z}_2 - \dot{z}_1 + 4z_1 &= 0\end{aligned} \tag{3.95}$$

This system can be represented by a state-space model using the following three state variables:

$$\begin{aligned}x_1 &= z_1 \\ x_2 &= \dot{z}_1 \\ x_3 &= z_2.\end{aligned} \tag{3.96}$$

The procedure for finding the state-space model is to take the first derivative with respect to time of each of the state variables. The resulting expressions must be simplified until the only terms they contain are state variables and/or the input function without any derivatives. We begin this procedure with x_1. The first derivative is $\dot{x}_1$,

which, from (3.96) equals $\dot{z}_1$. The second equation in (3.96) shows that $\dot{z}_1 = x_2$. Thus,

$$\dot{x}_1 = x_2. \tag{3.97}$$

Proceeding to the second state variable we see from (3.96) that $\dot{x}_2 = \ddot{z}_1$. The function $\ddot{z}_1$ does not appear in the definition of state variables (3.96). However, it does appear in the first differential equation of (3.95). Solving for $\ddot{z}_1$ from that equation and setting the result equal to $\dot{x}_2$ yields

$$\begin{aligned} \dot{x}_2 &= 2(\dot{z}_2 - \dot{z}_1) - 3z_2 + u \\ &= 2(\dot{x}_3 - x_2) - 3x_3 + u \end{aligned} \tag{3.98}$$

where the second line was obtained with the help of (3.96). Note that this expression for $\dot{x}_2$ is in terms of state variables and input but contains a derivative, $\dot{x}_3$. In order to deal with this problem we simply continue the process by computing the derivative of x_3 using (3.96) and (3.95)

$$\begin{aligned} \dot{x}_3 &= \dot{z}_2 \\ &= \dot{z}_1 - 4z_1 \\ &= x_2 - 4x_1 \\ &= -4x_1 + x_2. \end{aligned} \tag{3.99}$$

Note that it is useful to arrange the right-hand side of the equation with the state variable number increasing from left to right. The reason is that this is the order of appearance of the coefficients in the state-space matrix $\mathbf{A}$ (for example, compare the last line of (3.99) with the last line of the matrix in (3.101)). Substituting (3.99) into (3.98) removes the derivative on the right-hand side of (3.98) giving

$$\begin{aligned} \dot{x}_2 &= 2(-4x_1 + x_2 - x_2) - 3x_3 + u \\ &= -8x_1 - 3x_3 + u. \end{aligned} \tag{3.100}$$

Combining (3.97), (3.100), and (3.99) yields a state equation of the form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$ for the overall system:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -8 & 0 & -3 \\ -4 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u \tag{3.101}$$

3.3.6 Initial Conditions for Continuous-Time Systems*

Recall that nonzero initial conditions are easily specified for a state-space model by having a nonzero initial condition vector $\mathbf{x}_0$. We show in this section how initial conditions for a differential equation can be expressed as an appropriate initial condition vector for a state-space model.

To be specific, consider a general n^{th}-order differential equation

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots + a_n y(t) = b_0 u^{(n)}(t) + b_1 u^{(n-1)}(t) + \cdots + b_n u(t) \tag{3.102}$$

and let us assume that the initial conditions $y(0-), y^{(1)}(0-), \cdots, y^{(n-1)}(0-)$ are known. The initial time $t = 0-$ is taken to mean a time just before $t = 0$. By evaluating conditions

at $t = 0-$ we can ignore discontinuities in $u(t)$ and its derivatives at $t = 0$. We will assume that $u(t) = 0, t < 0$ so that $u(0-), u^{(1)}(0-), \cdots, u^{(n-1)}(0-)$ are all equal to zero. Let (3.102) be represented by the following state-space model

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,u(t) \\ y(t) &= \mathbf{C}\,\mathbf{x}(t) + D\,u(t) \end{aligned} \tag{3.103}$$

where $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D)$ could be a canonical form, for instance. Using (3.103) and the fact that initial conditions in $u(t)$ are zero, we can express the initial conditions on $y(t)$ in terms of the state-space model as follows

$$\begin{aligned} y(0-) &= \mathbf{C}\mathbf{x}(0-) + Du(0-) &&= \mathbf{C}\mathbf{x}(0-) \\ y^{(1)}(0-) &= \mathbf{C}\mathbf{x}^{(1)}(0-) + Du^{(1)}(0-) &&= \mathbf{C}\mathbf{A}\mathbf{x}(0-) \\ y^{(2)}(0-) &= \mathbf{C}\mathbf{A}\mathbf{x}^{(1)}(0-) + Du^{(2)}(0-) &&= \mathbf{C}\mathbf{A}^2\mathbf{x}(0-) \\ \vdots &= \vdots &&= \vdots \\ y^{(n-1)}(0-) &= \mathbf{C}\mathbf{A}\mathbf{x}^{(n-2)}(0-) + Du^{(n-1)} &&= \mathbf{C}\mathbf{A}^{(n-1)}\mathbf{x}(0-). \end{aligned} \tag{3.104}$$

If we introduce the following notation

$$\mathbf{y}_0 = \begin{bmatrix} y(0-) \\ y^{(1)}(0-) \\ \vdots \\ y^{(n-1)}(0-) \end{bmatrix}, \quad \mathbf{W}_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix}$$

then (3.104) can be rewritten as

$$\mathbf{y}_0 = \mathbf{W}_o \mathbf{x}_0.$$

Note that $\mathbf{W}_o$ is an $n \times n$ matrix. If it is nonsingular, we can multiply through by its inverse to obtain

$$\mathbf{x}_0 = \mathbf{W}_0^{-1} \mathbf{y}_0(0). \tag{3.105}$$

The matrix $\mathbf{W}_o$ is called the *observability* matrix, and its importance will be seen again in Chapter 7. For now we just note that the observability matrix must be nonsingular in order to be able to calculate a unique initial condition vector for a state-space model which corresponds to given initial conditions of a differential equation.

EXAMPLE 3.16

Consider the following differential equation

$$\ddot{y}(t) + \dot{y}(t) + y(t) = u(t)$$

with initial conditions $y(0) = \dot{y}(0) = 1$. The controllable canonical form state-space model for this differential equation is

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \begin{bmatrix} -1 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{x}(t). \end{aligned}$$

We want to find the initial condition vector $\mathbf{x}(0)$ which corresponds to the given initial conditions of the differential equation. We begin by calculating the observability matrix

$$\mathbf{W}_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

We now use (3.105) to calculate the initial condition vector

$$\mathbf{x}_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

■

3.3.7 Initial Conditions for Discrete-Time Systems*

Consider a general n^{th}-order difference equation

$$y[k] + a_1 y[k-1] + \cdots + a_n y[k-n] = b_0 u[k] + b_1 u[k-1] + \cdots + b_n u[k-n]. \quad (3.106)$$

Assume that the initial conditions $y[-1], \cdots, y[-n]$ are known, and that $u[-1], \cdots, u[-n]$ are all zero. Suppose (3.106) is represented by the following state-space model

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{A}\,\mathbf{x}[k] + \mathbf{B}\,u[k] \\ y[k] &= \mathbf{C}\,\mathbf{x}[k] + D\,u[k]. \end{aligned}$$

Using the above equations we can write the first n samples of the output in terms of $\mathbf{x}[0]$ and the input as follows (compare (3.115) in the next section)

$$\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[n-1] \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \mathbf{x}[0] + \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ \mathbf{CB} & D & 0 & \cdots & 0 \\ \mathbf{CAB} & \mathbf{CB} & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{n-2}\mathbf{B} & \mathbf{CA}^{n-3}\mathbf{B} & \mathbf{CA}^{n-4}\mathbf{B} & \cdots & D \end{bmatrix} \begin{bmatrix} u[0] \\ u[1] \\ \vdots \\ u[n-1] \end{bmatrix}$$

$$\stackrel{\text{def}}{=} \mathbf{W}_o \mathbf{x}[0] + \mathbf{M}\mathbf{u}. \quad (3.107)$$

Using the difference equation (3.106) these same output samples can be written in terms of past output samples (initial conditions) and input samples as follows

$$\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[n-1] \end{bmatrix} = \mathbf{P} \begin{bmatrix} y[-1] \\ y[-2] \\ \vdots \\ y[-n] \end{bmatrix} + \mathbf{M}\mathbf{u} \quad (3.108)$$

where $\mathbf{P}$ and $\mathbf{M}$ are constructed from coefficients of the difference equation. The next example shows how the matrix $\mathbf{P}$ is constructed. It turns out that the matrix $\mathbf{M}$ in (3.108)

is identical to the matrix $\mathbf{M}$ in (3.107). If we equate (3.107) and (3.108), the term $\mathbf{Mu}$ cancels and we obtain

$$\mathbf{W}_o\mathbf{x}[0] = \mathbf{P}\mathbf{y}^-$$

where $\mathbf{y}^-$ is a vector of initial conditions

$$\mathbf{y}^- = \begin{bmatrix} y[-1] \\ y[-2] \\ \vdots \\ y[-n] \end{bmatrix}.$$

If the matrix $\mathbf{W}_o$ is nonsingular, we can find a unique initial state vector corresponding to any set of initial conditions of the difference equation as

$$\mathbf{x}[0] = \mathbf{W}_o^{-1}\mathbf{P}\mathbf{y}^-. \tag{3.109}$$

EXAMPLE 3.17

Consider the following difference equation

$$y[k] + 2y[k-1] + y[k-2] = u[k] \tag{3.110}$$

with initial conditions $y[-1] = y[-2] = 1$. The controllable canonical form state-space model for this differential equation is

$$\mathbf{x}[k+1] = \begin{bmatrix} -2 & -1 \\ 1 & 0 \end{bmatrix}\mathbf{x}[k] + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u[k]$$

$$y[k] = \begin{bmatrix} -2 & -1 \end{bmatrix}\mathbf{x}[k] + u[k].$$

We want to find the initial condition vector $\mathbf{x}[0]$ which corresponds to the given initial conditions of the differential equation. We begin by calculating the observability matrix

$$\mathbf{W}_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix} = \begin{bmatrix} -2 & -1 \\ 3 & 2 \end{bmatrix}.$$

Next we use (3.110) to express $y[0]$ and $y[1]$ in terms of $y[-1]$ and $y[-2]$ and the input as follows

$$y[0] = -2y[-1] - y[-2] + u[0]$$

$$\begin{aligned} y[1] &= -2y[0] - y[-1] + u[1] \\ &= -2(-2y[-1] - y[-1] + u[0]) - y[-1] + u[1] \\ &= 3y[-1] + 2y[-2] - 2u[0] + u[1]. \end{aligned}$$

The above equations can be written in matrix form as follows

$$\begin{bmatrix} y[0] \\ y[1] \end{bmatrix} = \begin{bmatrix} -2 & -1 \\ 3 & 2 \end{bmatrix}\begin{bmatrix} y[-1] \\ y[-2] \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix}\begin{bmatrix} u[0] \\ u[1] \end{bmatrix}$$

$$= \mathbf{P}\mathbf{y}^- + \mathbf{M}\mathbf{u}.$$

The matrix $\mathbf{P}$ is all that we need from the above equation. We evaluate (3.109) to obtain the initial state vector

$$\mathbf{x}[0] = \mathbf{W}_o^{-1}\mathbf{P}\mathbf{y}^- = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

■

3.4 Solving the State Equations

The previous section has introduced state-space models for continuous- and discrete-time systems. Various canonical forms have been introduced, and certain manipulations involving linear transformations and transfer functions have been discussed. All this has been done without any mention of the solution to the state equations. In this section, we develop formulas for the solution to both discrete-time and continuous-time systems. A solution to the state equations is simply a formula which expresses the state vector at the current time in terms of the state vector at some initial time and the input. We begin with discrete-time state equations because they are easier to solve than continuous-time state equations.

3.4.1 Solving Discrete-Time State Equations

Consider first the homogeneous vector difference equation

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k]\,, \quad \mathbf{x}[0] = \mathbf{x}_0. \tag{3.111}$$

We can determine the values of $\mathbf{x}[k]$ in terms of $\mathbf{x}_0$ for $k = 1, 2, \cdots$ by repeated application of (3.111) as follows

$$\begin{aligned} \mathbf{x}[1] &= \mathbf{A}\mathbf{x}_0 \\ \mathbf{x}[2] &= \mathbf{A}\mathbf{x}_1 \\ &= \mathbf{A}^2\mathbf{x}_0 \\ &\vdots \quad = \quad \vdots \\ \mathbf{x}[k] &= \mathbf{A}^k\mathbf{x}_0. \end{aligned} \tag{3.112}$$

The last line is the solution to the homogeneous equation (3.111).

Now suppose that an input sequence is added to (3.111) to obtain the following discrete-time state equation

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}u[k]. \tag{3.113}$$

Suppose we know $\mathbf{x}[k_0]$ and the input $\{u[k]\}, k \geq k_0$. How do we solve for $\mathbf{x}[k], k > k_0$? The answer is obtained by recursively using (3.113) starting from the known value:

$$\begin{aligned}
\mathbf{x}[k_0+1] &= \mathbf{A}\mathbf{x}[k_0] + \mathbf{B}u[k_0] \\
\mathbf{x}[k_0+2] &= \mathbf{A}\mathbf{x}[k_0+1] + \mathbf{B}u[k_0+1] \\
&= \mathbf{A}\left\{\mathbf{A}\mathbf{x}[k_0] + \mathbf{B}u[k_0]\right\} + \mathbf{b} == Bbu[k_0+1] \\
&= \mathbf{A}^2\mathbf{x}[k_0] + \mathbf{A}\mathbf{B}u[k_0] + \mathbf{B}u[k_0+1] \\
\mathbf{x}[k_0+3] &= \mathbf{A}\mathbf{x}[k_0+2] + \mathbf{B}u[k_0+2] \\
&= \mathbf{A}^3\mathbf{x}[k_0] + \mathbf{A}^2\mathbf{B}u[k_0] + \mathbf{A}\mathbf{B}u[k_0+1] + \mathbf{B}u[k_0+2] \\
\vdots &= \qquad \vdots \\
\mathbf{x}[k] &= \mathbf{A}^{k-k_0}\mathbf{x}[k_0] + \mathbf{A}^{k-k_0-1}\mathbf{B}u[k_0] + \cdots + \mathbf{B}u[k-1].
\end{aligned}$$

The above recursions simplify to the following general formula

$$\mathbf{x}[k] = \begin{cases} \mathbf{x}[k_0] & k = k_0 \\ \mathbf{A}^{k-k_0}\mathbf{x}[k_0] + \displaystyle\sum_{j=k_0}^{k-1} \mathbf{A}^{k-j-1}\mathbf{B}u[j] & k > k_0. \end{cases} \tag{3.114}$$

If the output equation for the discrete-time system is $y[k] = \mathbf{C}\mathbf{x}[k] + Du[k]$, then a formula for the output of the system in terms of the input and initial state vector is obtained using (3.114) as follows

$$y[k] = \begin{cases} \mathbf{C}\mathbf{x}[k_0] + Du[k_0] & k = k_0 \\ \mathbf{C}\mathbf{A}^{k-k_0}\mathbf{x}[k_0] + \displaystyle\sum_{j=k_0}^{k-1} \mathbf{C}\mathbf{A}^{k-j-1}\mathbf{B}u[j] + Du[k] & k > k_0. \end{cases} \tag{3.115}$$

Discrete-Time Impulse Response We can immediately put (3.115) to work by using it to derive a formula for the impulse response of a discrete-time system in terms of its state-space matrices. Recall that the impulse response $h[k]$ of a discrete-time system is the response (with zero initial conditions) to the delta sequence $\delta[k]$ defined as

$$\delta[k] = \begin{cases} 1, & k = 0 \\ 0, & \text{otherwise.} \end{cases}$$

If we let $u[k] = \delta[k]$ in (3.115), take the initial time to be $k = 0$, and let the initial state vector equal zero, we obtain

$$h[k] = \begin{cases} D\delta[k] & k = 0 \\ \displaystyle\sum_{j=0}^{k-1} \mathbf{C}\mathbf{A}^{k-j-1}\mathbf{B}\delta[j] & k > 0 \end{cases}$$

or

$$h[k] = \begin{cases} D & k = 0 \\ \mathbf{C}\mathbf{A}^{k-1}\mathbf{B} & k > 0. \end{cases} \tag{3.116}$$

Recall that the impulse response and the transfer function are a z-transform pair. We can verify this by taking the z-transform of (3.116) and seeing if we end up with (3.72). The z-transform is

$$\begin{aligned} Z\{h[k]\} &= \sum_{k=0}^{\infty} h[k] z^{-k} \\ &= \sum_{k=1}^{\infty} \mathbf{C}\mathbf{A}^{k-1}\mathbf{B} z^{-k} + D \end{aligned}$$

where the term D comes out of the sum because it was multiplied by $\delta[k]$, which is only nonzero when $k = 0$. Since the vectors $\mathbf{C}$ and $\mathbf{B}$ are constant (do not depend on k), we can pull them out of the summation to yield

$$\begin{aligned} H(z) &= \mathbf{C}\left[\sum_{k=1}^{\infty} \mathbf{A}^{k-1} z^{-k}\right]\mathbf{B} + D \\ &= \mathbf{C}\left[z^{-1}\sum_{k=0}^{\infty} (z^{-1}\mathbf{A})^{k}\right]\mathbf{B} + D. \end{aligned} \tag{3.117}$$

We now use the matrix identity for a matrix $\mathbf{M}$ whose eigenvalues lie inside the unit circle

$$\sum_{k=0}^{\infty} \mathbf{M}^k = (\mathbf{I} - \mathbf{M})^{-1}$$

with $\mathbf{M} = z^{-1}\mathbf{A}$ to complete the derivation. (The proof of the identity is left as an exercise.) Since the eigenvalues of $\mathbf{M}$ must lie inside the unit circle, this means that z lies outside a circle whose radius is the magnitude of the largest eigenvalue of $\mathbf{A}$ (see Fact 2.29 on page 57). Such values of z constitute the region of convergence for $H(z)$ in (3.117). Then we have

$$\begin{aligned} H(z) &= \mathbf{C}z^{-1}(I - z^{-1}\mathbf{A})^{-1}\mathbf{B} + D \\ &= \mathbf{C}(zI - \mathbf{A})^{-1}\mathbf{B} + D. \end{aligned}$$

Thus we see that $h[k]$ and $H(z)$ indeed form a z-transform pair. It is left as an exercise to take the inverse z-transform of $H(z)$ directly and see that the result is $h[k]$.

3.4.2 Solving Continuous-Time State Equations

Let us begin with an example of the first-order differential equation

$$\dot{x}(t) = a\, x(t)\ , \quad x(0) = x_0 \tag{3.118}$$

where $x(t)$ is a scalar function. The solution to this equation is

$$x(t) = e^{at} x_0, \tag{3.119}$$

which can be verified by substituting (3.119) into (3.118). Note that (3.119) also satisfies the initial condition $x(0) = x_0$.

The exponential function e^{at} is the solution to (3.118) because it possesses the following "reproducing" property under differentiation

$$\frac{d(e^{at})}{dt} = a\, e^{at}. \tag{3.120}$$

This property is easily derived by considering the power series expansion for the exponential function

$$e^{at} = 1 + at + \frac{a^2t^2}{2} + \frac{a^3t^3}{3!} + \cdots. \tag{3.121}$$

The reproducing property in (3.120) is obtained by differentiating both sides of the power series expansion in (3.121), as shown below

$$\begin{aligned} \frac{d(e^{at})}{dt} &= 0 + a + a^2t + \frac{a^3t^2}{2} + \cdots \\ &= a[1 + at + \frac{a^2t^2}{2} + \cdots] \\ &= ae^{at}. \end{aligned} \tag{3.122}$$

Let us now consider the homogeneous vector differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)\ ,\ \ t \geq 0\ ,\ \ \mathbf{x}(0) = \mathbf{x}_0, \tag{3.123}$$

where $\mathbf{x}(t)$ is an $n \times 1$ vector and $\mathbf{A}$ is an $n \times n$ matrix. The equation is called "homogeneous" because there is no external input (i.e. no $\mathbf{b}\,u(t)$ term). If we could define a "matrix exponential" $e^{\mathbf{A}t}$ which had the reproducing property (3.120) that scalar exponentials have, then solution to (3.123) would be $e^{\mathbf{A}t}\mathbf{x}_0$, in analogy with the scalar case. It is clear that the reproducing property would be true for the matrix exponential if it were defined by the same power series as the scalar exponential function. This is, in fact, what is done. The matrix exponential is defined by

$$e^{\mathbf{A}t} \stackrel{\text{def}}{=} I + \mathbf{A}t + \frac{\mathbf{A}^2t^2}{2} + \frac{\mathbf{A}^3t^3}{3!} + \cdots. \tag{3.124}$$

The above equation is a matrix power series where the notation "$\mathbf{A}^k$" denotes the matrix $\mathbf{A}$ multiplied by itself k times. From (3.124) it is easy to see that the matrix exponential satisfies

$$e^{\mathbf{A}0} = \mathbf{I},\ \text{and}$$

$$\frac{d}{dt}e^{\mathbf{A}t} = \mathbf{A}e^{\mathbf{A}t}.$$

Thus $\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0$ is the solution to (3.123).

For a particular value of t, the matrix exponential could be calculated by truncating the power series after a large but finite number of terms. However, reliable numerical procedures that computers use to calculate the matrix exponential are based on more sophisticated algorithms. Many of the methods proposed to calculate the matrix exponential are not numerically reliable [63]. See [37] for a reliable numerical procedure to compute the matrix exponential. The matrix exponential shares some properties with the scalar exponential, such as those shown in Table 3.6, which can be derived using the series representation for $e^{\mathbf{A}t}$.

1. $\dfrac{d(e^{\mathbf{A}t})}{dt} = \mathbf{A}e^{\mathbf{A}t} = e^{\mathbf{A}t}\mathbf{A}$.
2. $\displaystyle\int_0^t e^{\mathbf{M}\tau}d\tau = \mathbf{M}^{-1}\left(e^{\mathbf{M}t} - \mathbf{I}\right) = \left(e^{\mathbf{M}t} - \mathbf{I}\right)\mathbf{M}^{-1}$ for any invertible matrix $\mathbf{M}$.
3. exp $[\mathbf{A}(t_1 + t_2)]$ = exp $(\mathbf{A}t_1)$ exp $(\mathbf{A}t_2)$.
4. exp $(\mathbf{A}t)$ is a nonsingular matrix and $[\text{exp }(\mathbf{A}t)]^{-1}$ = exp $(-\mathbf{A}t)$.
5. exp $(\mathbf{A})$ exp $(\mathbf{B})$ = exp $(\mathbf{A} + \mathbf{B})$ if $\mathbf{AB} = \mathbf{BA}$.

Table 3.6 Some properties of the matrix exponential.

Thus far we have only considered the homogeneous vector differential equation. We now consider the equation with an input term and with initial conditions at some time t_0

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\,u(t)\,, \quad \mathbf{x}(t_0) = \mathbf{x}_0. \tag{3.125}$$

By analogy with the corresponding scalar equation (or by using Laplace transforms), we can write the solution as

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\,u(\tau)\,d\tau. \tag{3.126}$$

This equation is derived in [33], for example. When $t = t_0$ the integral equals zero and the matrix exponential in the first term equals the identity matrix. Thus $\mathbf{x}(0) = \mathbf{x}_0$, and (3.126) satisfies the initial conditions of (3.125). We verify that (3.126) satisfies the differential equation of (3.125) by differentiating (3.126). We will make use of the product rule for differentiation

$$\frac{d[f(t)g(t)]}{dt} = f(t)\frac{d[g(t)]}{dt} + \frac{d[f(t)]}{dt}g(t).$$

Note that this rule is valid even for matrix- and vector-valued functions as long as the order of f and g is preserved as shown above. The reason that the order must be preserved is that matrix-vector multiplication is not commutative. We also need to recall the rule for differentiating under an integral when the lower limit is a constant:

$$\frac{d}{dt}\int_a^t f(\tau)d\tau = f(t). \tag{3.127}$$

We want to differentiate (3.126) and show that it satisfies (3.125). We begin by rewriting (3.126) by pulling $e^{\mathbf{A}t}$ out of the integral, since it does not depend of the variable of integration τ

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + e^{\mathbf{A}t}\int_{t_0}^{t} e^{-\mathbf{A}\tau}\mathbf{B}\,u(\tau)\,d\tau. \tag{3.128}$$

Note that the term with the integral is the product of two functions of time. The first function, $e^{\mathbf{A}t}$ can be differentiated to yield $\mathbf{A}e^{\mathbf{A}t}$. The second function is the integral itself which can be differentiated using (3.127). These derivatives are combined using the

product rule. Thus the derivative of (3.128) is

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + e^{\mathbf{A}t}e^{-\mathbf{A}t}\mathbf{B}u(t) + \mathbf{A}e^{\mathbf{A}t}\int_{t_0}^{t} e^{-\mathbf{A}\tau}\mathbf{B}u(\tau)d\tau \\ &= \mathbf{A}\{e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + e^{\mathbf{A}t}\int_{t_0}^{t} e^{-\mathbf{A}\tau}\mathbf{B}\,u(\tau)\,d\tau\} + \mathbf{B}u(t) \\ &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t).\end{aligned}$$

By differentiating (3.128), we have obtained (3.125), and so (3.128) (or equivalently (3.126)) is the solution to (3.125).

If the output equation for the continuous-time system is $y(t) = \mathbf{C}\mathbf{x}(t) + Du(t)$, then a formula for the output of the system in terms of the input and initial state vector is obtained using (3.126) as follows

$$y(t) = \mathbf{C}e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)D\tau + Du(t). \tag{3.129}$$

Continuous-Time Impulse Response We can use (3.129) to derive a formula for the impulse response of a continuous-time system in terms of its state-space matrices. If we let $u(t) = \delta(t)$ in (3.129), take the initial time to be $t_0 = 0$, and let the initial state vector equal zero, we obtain

$$\begin{aligned}h(t) &= \int_0^t \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{b}\delta(\tau)d\tau + d\delta(t) \\ &= \mathbf{c}e^{\mathbf{A}t}\mathbf{b} + d\delta(t)\,, \quad t \geq 0.\end{aligned} \tag{3.130}$$

Recall that the impulse response and the transfer function are a Laplace transform pair. We can check this by taking the Laplace transform of (3.130) and seeing that we end up with (3.72). The Laplace transform is

$$\begin{aligned}L[h(t)] &= \int_0^\infty h(t)e^{-st}dt \\ &= \int_0^\infty [\mathbf{C}e^{\mathbf{A}t}\mathbf{B} + D\delta(t)]e^{-st}dt \\ &= \mathbf{C}\int_0^\infty e^{\mathbf{A}t}e^{-st}dt\,\mathbf{B} + D \\ &= \mathbf{C}\int_0^\infty e^{-(sI-\mathbf{A})t}dt\,\mathbf{B} + D \\ &= \mathbf{C}(sI - \mathbf{A})^{-1}\mathbf{B} + D.\end{aligned} \tag{3.131}$$

The last equality follows from the matrix identity

$$\int_0^\infty e^{-\mathbf{M}t}dt = \mathbf{M}^{-1}, \tag{3.132}$$

where $\mathbf{M}$ is any matrix whose eigenvalues all have positive real parts. The proof of this identity is left as an exercise. To use (3.132) set $\mathbf{M} = (s\mathbf{I} - \mathbf{A})$. Since the eigenvalues of $\mathbf{M}$ must have positive real parts we see that the Laplace transform in (3.131) is defined

Continuous Time	Discrete Time
$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t)$ $y(t) = \mathbf{C}\mathbf{x}(t) + Du(t)$	$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}u[k]$ $y[k] = \mathbf{c}\mathbf{x}[k] + du[k]$
$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)d\tau$	$\mathbf{x}[k] = \mathbf{A}^k\mathbf{x}(0) + \sum_{j=0}^{k-1}\mathbf{A}^{k-j-1}\mathbf{B}u(j)$
$H(s) = \mathbf{C}(sI - \mathbf{A})^{-1}\mathbf{B} + D$	$H(z) = \mathbf{C}(zI - \mathbf{A})^{-1}\mathbf{B} + D$
$h(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{B} + D\delta(t)$	$h[k] = \begin{cases} D & k = 0 \\ \mathbf{C}\mathbf{A}^{k-1}\mathbf{B} & k > 0. \end{cases}$
$(sI - \mathbf{A})^{-1} \underset{L}{\overset{L^{-1}}{\rightleftarrows}} e^{\mathbf{A}t}$	$(zI - \mathbf{A})^{-1} \underset{Z}{\overset{Z^{-1}}{\rightleftarrows}} \mathbf{A}^{k-1}$
If $\bar{\mathbf{x}}(t) = T\mathbf{x}(t)$ then $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D) \xrightarrow{T} (\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{D})$ where $\bar{\mathbf{A}} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}$ $\bar{\mathbf{B}} = \mathbf{T}\mathbf{B}$ $\bar{\mathbf{C}} = \mathbf{C}\mathbf{T}^{-1}$ $\bar{D} = D$	If $\bar{\mathbf{x}}[k] = T\mathbf{x}[k]$ then $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D) \xrightarrow{T} (\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{D})$ where $\bar{\mathbf{A}} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}$ $\bar{\mathbf{B}} = \mathbf{T}\mathbf{B}$ $\bar{\mathbf{C}} = \mathbf{C}\mathbf{T}^{-1}$ $\bar{D} = D$

Table 3.7 Summary of results from continuous- and discrete-time system theory.

for values of s whose real parts are greater than the real parts of all the eigenvalues of $\mathbf{A}$ (see Fact 2.30 on page 58). Such values of s constitute the region of convergence for the Laplace transform in (3.131).

Thus we see that $h(t)$ and $H(s)$ indeed form a Laplace transform pair. It is left as an exercise to take the inverse Laplace transform of $H(s)$ directly and see that the result is $h(t)$.

3.5 Interconnected Systems

We have shown previously how state-space models of a system can be obtained. In this section we assume that state-space models for two systems are given. A new system can be formed by interconnecting the given systems in some way: parallel, cascade, or feedback. We show how to obtain a state-space description for the resulting system, and we also give the corresponding transfer function relationships. Discrete-time systems are used in the derivations, but identical results are obtained for continuous-time systems.

Let the state-space model for the first system be denoted as

$$\begin{aligned} \mathbf{x}_1[k+1] &= \mathbf{A}_1\,\mathbf{x}_1[k] + \mathbf{B}_1\,u_1[k] \\ y_1[k] &= \mathbf{C}_1\,\mathbf{x}_1[k] + D_1\,u_1[k] \end{aligned} \tag{3.133}$$

and the state-space model for the second system as

$$\begin{aligned} \mathbf{x}_2[k+1] &= \mathbf{A}_2\,\mathbf{x}_2[k] + \mathbf{B}_2\,u_2[k] \\ y_2[k] &= \mathbf{C}_2\,\mathbf{x}_2[k] + D_2\,u_2[k]. \end{aligned} \tag{3.134}$$

The inputs to the two systems could be different (as well as the outputs) and so subscripts are used to distinguish the input to the first system from the input to the second system. We denote the transfer functions of the two systems as $H_1(z)$ and $H_2(z)$.

The two systems will be interconnected to form a single system. The input and output of this new system will be denoted $r[k]$ and $y[k]$, respectively. The state vector for the new system will be denoted $\mathbf{z}[k]$, and is chosen to be the concatenation of the state vectors for the two given systems as follows

$$\mathbf{z}[k] = \begin{bmatrix} \mathbf{x}_1[k] \\ \mathbf{x}_2[k] \end{bmatrix}.$$

The state-space model for the new system will have the form

$$\begin{aligned} \mathbf{z}[k+1] &= \begin{bmatrix} \bar{\mathbf{A}}_{11} & \bar{\mathbf{A}}_{12} \\ \bar{\mathbf{A}}_{21} & \bar{\mathbf{A}}_{22} \end{bmatrix} \mathbf{z}[k] + \begin{bmatrix} \bar{\mathbf{B}}_1 \\ \bar{\mathbf{B}}_2 \end{bmatrix} r[k] \\ y[k] &= \begin{bmatrix} \bar{\mathbf{C}}_1 & \bar{\mathbf{C}}_2 \end{bmatrix} \mathbf{z}[k] + \bar{D} r[k]. \end{aligned} \tag{3.135}$$

The transfer function of the interconnected system will be denoted by $\bar{H}(z)$. In the next three subsections we show how the matrices of this state-space model can be obtained from the matrices of the state-space models for the two original systems. The results are presented for single-input, single-output systems, but they are valid for multiple-input, multiple-output systems with the following change: $\mathbf{b}$, $\mathbf{c}$, and d would all be upper case boldface letters to indicate that they are matrices.

Parallel Connection

A parallel connection of two systems is shown in Fig. 3.20, and is characterized by the fact that $u_1[k] = u_2[k] = r[k]$ and $y[k] = y_1[k] + y_2[k]$.

Because the given systems are both driven by the same input, the state-update equation for the interconnected system is

$$\mathbf{z}[k+1] = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{bmatrix} \mathbf{z}[k] + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} r[k]. \tag{3.136}$$

The upper half of this equation is the state-update equation for the first system, and the bottom half is the state-update equation for the second system. The output of the interconnected system can be written as

$$\begin{aligned} y[k] &= y_1[k] + y_2[k] \\ &= \mathbf{C}_1\mathbf{x}_1[k] + D_1 r[k] + \mathbf{C}_2\mathbf{x}_2[k] + D_2 r[k] \\ &= \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix} \mathbf{z}[k] + (D_1 + D_2) r[k]. \end{aligned} \tag{3.137}$$

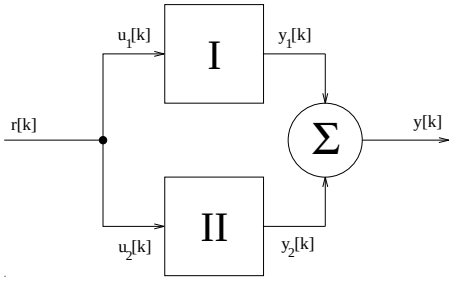


Figure 3.20 A parallel interconnection of two systems.

Equations (3.136) and (3.137) are a state-space model of the form of (3.135) for the parallel interconnection of two systems. The transfer function relationship is

$$\bar{H}(z) = H_1(z) + H_2(z).$$

Cascade Connection

A cascade connection of two systems is shown in Fig. 3.21, and is characterized by the fact that $u_1[k] = r[k]$, $u_2[k] = y_1[k]$ and $y[k] = y_2[k]$. In order to derive the state-update

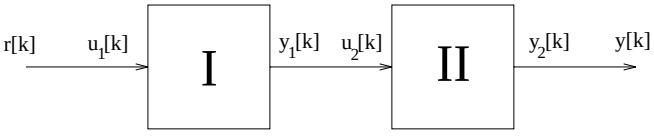


Figure 3.21 A cascade interconnection of two systems.

equation for the overall system, we substitute the output equation of the first system into the state-update of the second system as follows

$$\begin{aligned}\mathbf{x}_2[k+1] &= \mathbf{A}_2\mathbf{x}_2[k] + \mathbf{B}_2 u_2[k] \\ &= \mathbf{A}_2\mathbf{x}_2[k] + \mathbf{B}_2(\mathbf{C}_1\mathbf{x}_1[k] + D_1 r[k]) \\ &= \mathbf{B}_2\mathbf{C}_1\mathbf{x}_1[k] + \mathbf{A}_2\mathbf{x}_2[k] + \mathbf{B}_2 D_1 r[k].\end{aligned} \tag{3.138}$$

The state-update equation for the interconnected system is obtained from the above equation and the state-update equation for the first system. The result is

$$\mathbf{z}[k+1] = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{B}_2\mathbf{C}_1 & \mathbf{A}_2 \end{bmatrix} \mathbf{z}[k] + \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 D_1 \end{bmatrix} r[k]. \tag{3.139}$$

The output of the interconnected system can be shown to be

$$y[k] = \begin{bmatrix} D_2\mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix} \mathbf{z}[k] + D_1 D_2 r[k]. \tag{3.140}$$

The transfer function relationship is

$$\bar{H}(z) = H_2(z)H_1(z).$$

Feedback Connection

A feedback connection of two systems is shown in Fig. 3.22, and is characterized by the fact that $u_1[k] = r[k] - y_2[k]$, $u_2[k] = y_1[k]$, and$y[k] = y_1[k]$. The derivation of the

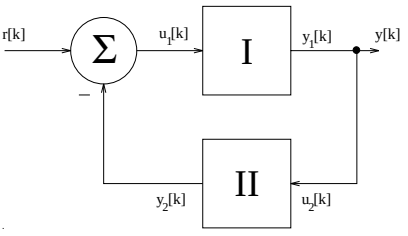


Figure 3.22 A feedback interconnection of two systems.

equations for the feedback connection of two systems is left as an exercise. The results are given below by specifying the submatrices that appear in the general form of the state equations given in (3.135).

$$\begin{aligned} \bar{\mathbf{A}}_{11} &= \mathbf{A}_1 - \mathbf{B}_1(1 + D_2D_1)^{-1}D_2\mathbf{C}_1, \quad & \bar{\mathbf{A}}_{12} &= -\mathbf{B}_1(1 + D_2D_1)^{-1}\mathbf{C}_2 \\ \bar{\mathbf{A}}_{21} &= \mathbf{B}_2(1 + D_1D_2)^{-1}\mathbf{C}_1, & \bar{\mathbf{A}}_{22} &= \mathbf{A}_2 - \mathbf{B}_2(1 + d_1D_2)^{-1}D_1\mathbf{C}_2 \end{aligned} \tag{3.141}$$

$$\bar{\mathbf{B}}_1 = \mathbf{B}_1(1 - (1 + D_2D_1)^{-1}D_2D_1)$$

$$\bar{\mathbf{B}}_2 = \mathbf{B}_2(1 + D_1D_2)^{-1}D_1)$$

$$\bar{\mathbf{C}}_1 = (1 + D_1D_2)^{-1}\mathbf{C}_1, \quad \bar{\mathbf{C}}_2 = -(1 + D_1D_2)^{-1}D_1\mathbf{C}_2$$

$$\bar{D} = (1 + D_1D_2)^{-1}D_1.$$

Note that the feedback connection of two systems is not defined if $(1 + D_1D_2)^{-1} = 0$. For multivariable systems the "1" in the expression $(1 + D_1D_2)^{-1}$ would be an identity matrix. The transfer function relationship is

$$\bar{H}(z) = \frac{H_1(z)}{1 + H_2(z)H_1(z)}.$$

3.6 Models of Example Systems

In this section, transfer functions and state-space models are given for several different systems. These systems have a single input and output. Examples of systems with more than one input and output such as a chemical reactor or an airplane are given in Chapter 9.

3.6.1 A One Time-Constant System

Consider a system with input $u(t)$ and output $y(t)$ described by the differential equation

$$\dot{y}(t) + p\,y(t) = Ku(t) \tag{3.142}$$

and by the corresponding transfer function

$$H(s) = \frac{K}{s+p}. \tag{3.143}$$

This first-order system is characterized by a single time constant which is related to the pole location $-p$. The step response of this system is

$$\frac{K}{p}(1 - e^{-pt}).$$

Recall that the "time constant" of a system is defined to be the time τ at which the step response reaches 63% (or 1-$e^{-1} = .63$) of its steady-state value. Since the formula for the step response contains the factor $1 - e^{-pt}$, it can be seen that the time constant is $\tau = 1/p$ seconds. A sketch of a step response is shown in Fig. 3.23.

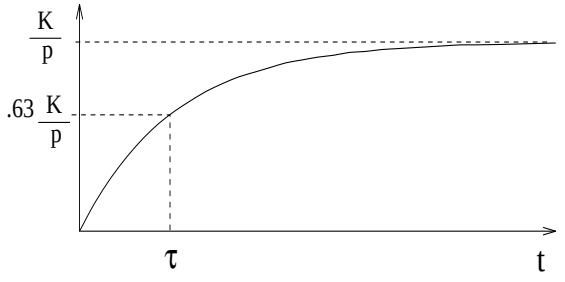


Figure 3.23 Step response of a one time-constant system.

A state-space model for this system can be obtained by choosing the state variable $x(t) = y(t)$. Then the state variable equations are

$$\begin{aligned} \dot{x}(t) &= -p\,x(t) + Ku(t) \\ y(t) &= x(t) \end{aligned} \tag{3.144}$$

This model is appropriate for any physical process which exhibits a single time-constant.

3.6.2 Double Integrator

The double integrator is described by the differential equation

$$\ddot{y}(t) = u(t) \tag{3.145}$$

and has the corresponding transfer function

$$H(s) = \frac{1}{s^2}. \tag{3.146}$$

This system has a double pole at $s = 0$. A state-space model can be obtained by introducing the state variables $x_1(t) = y(t)$ and $x_2(t) = \dot{y}(t)$. In situations where $y(t)$ represents an output position, the state variables are then position and velocity of the output, which

are physically meaningful variables. With this choice of state variables, the differential equation can be written as

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix} u(t)$$

$$y(t) = [1\ 0]\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}. \tag{3.147}$$

This model can be used to describe the motion of an undamped inertia in response to a torque. For example, the angle (attitude) of a satellite can be described by the equation

$$J\ddot{\theta}(t) = \tau(t) \tag{3.148}$$

where θ is the attitude angle, τ is the net torque on the satellite, and J is the moment of inertia. The double integrator model corresponds to a normalized inertia of unity. In this case the state variables are $x_1 = \theta$, the angular position of the satellite, and $x_2 = \dot{(\theta)}$, the angular velocity of the satellite.

3.6.3 Type-1 Servo System

Many physical systems are type-1 (having a single integrator – a pole at $s = 0$) with an additional time constant. Such a system is described by the differential equation

$$\ddot{y}(t) + p\,\dot{y}(t) = Au(t) \tag{3.149}$$

and transfer function

$$H(s) = \frac{A}{s(s+p)} \tag{3.150}$$

where $y(t)$ represents an output position. For example, an armature controlled dc motor with $u(t)$ = current into the motor and $y(t)$ = angular position of the motor. This model can be considered as an extension of the one time-constant model in which the output is passed through an integrator. This is shown in Fig. 3.24, where the state variables $x_1(t) = y(t)$ and $x_2(t) = \dot{y}(t)$ are labeled.

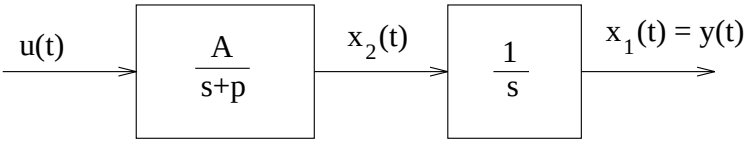


Figure 3.24 Block diagram of a Type-1 Servo System.

A state-space model for this system corresponding to this choice of state variables is

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -p\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}0\\ A\end{bmatrix} u(t)$$

$$y(t) = [1 \quad 0]\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}. \tag{3.151}$$

3.6.4 Harmonic Oscillator

A harmonic oscillator is a second-order system whose poles lie on the imaginary axis in the s-plane. The unforced response of such a system results in harmonic motion – a sinusoid whose frequency depends on the pole location. This model describes the motion of a pendulum near its stable equilibrium point. We will consider the full nonlinear pendulum model later. This model is appropriate for any system which exhibits oscillatory behavior.

The differential equation describing the harmonic oscillator is

$$\ddot{y}(t) + \omega^2 y(t) = \omega^2 u(t) \tag{3.152}$$

where $y(t)$ represents output position, and ω represents the frequency of oscillation in radians per second. The transfer function of the system is

$$H(s) = \frac{\omega^2}{s^2 + \omega^2}. \tag{3.153}$$

Note that the poles are located on the imaginary axis at $s = \pm j\omega$ radians per second.

By defining the state variables as $x_1(t) = y(t)$ and $x_2(t) = \dot{y}(t)$, the differential equation can be written in the following state space form

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix} u(t)$$
$$y(t) = [1\ 0] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}. \tag{3.154}$$

3.6.5 One Time-Constant Systems with Time Delay

Many industrial processes are characterized by a time delay between the time at which inputs are introduced into the system and the time at which they begin to take effect. For instance, consider a chemical process-control system with long input pipes. When an input reagent is introduced into a pipe, it takes a certain time for the reagent to travel down the pipe into the reaction vessel. The travel time is called the *transportation delay*. Another industrial process that exhibits time delay is paper making.

We will consider the simplest case where the dynamics of the process are first-order (characterized by a single time constant) when the time delay is taken into account. The differential equation for such a process is

$$\dot{y}(t) + py(t) = u(t - D). \tag{3.155}$$

where D is the time delay and $\tau = 1/p$ is the time constant. The transfer function for this process is

$$H(s) = \frac{e^{-sD}}{s + p}.$$

A model which includes time delay is important when considering a digital control system. This is true even when the system being controlled does not exhibit time delay. The reason is that the digital compensator takes a certain amount of time to compute output values given new input values. This computation time is a time delay which can be incorporated into the model of the plant, as in (3.155).

In most cases, the computational delay introduced by the digital compensator will be very small compared to the time constant of the system (i.e. $D \ll 1/p$ in (3.155)), and in this case, the computational delay can usually be neglected. Nevertheless, it is important to perform a careful analysis of the effects of computational delay on a digital control system and see when it is and when it is not important to take the delay into account. Such an analysis is given in Chapter 4.

If we define a variable $x(t) = y(t)$, we can rewrite (3.155) as

$$\dot{x}(t) = -px(t) + u(t - D). \tag{3.156}$$

This equation looks like a state-space equation, but it is not one because the input term is not evaluated at time t. In fact, it is not possible to give a continuous-time state-space description of a system with time delay. The reason is that such a model is *infinite dimensional*, which means that it requires an infinite number of state variables. The state variables are needed to store past values of the input signal for a duration of D seconds. Since this is a continuum of values, it results in an infinite number of state variables.

This point will be clarified in Chapter 4, where a discrete-time state-space model will be developed for systems with time delay. In that case, it is only necessary to store a finite number of previous input samples. Nevertheless, the dimension of the model could be large: the number of state variables required to model the time delay is about D/T where D is the time delay and T is the sampling interval.

3.6.6 A Pendulum-on-a-Cart System

This section gives a brief development of a model for a pendulum-on-a-cart system. A model is developed for an actual hardware system which was built at the University of Rhode Island. A diagram of the system is shown in Fig. 3.25. The input to this system is

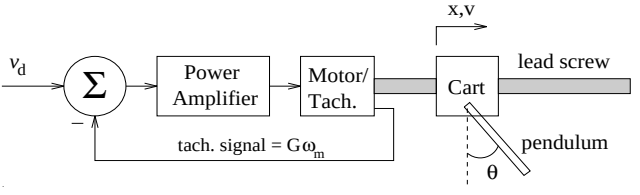


Figure 3.25 Diagram of a Cart/Pendulum System.

v_d, the voltage from the D/A converter which is specified by the computer controlling the system (the computer is not shown). The voltage v_d is proportional to the desired velocity of the motor. The *actual* motor velocity is measured by a tachometer which is attached to the motor. The tachometer signal, a voltage which is proportional to the angular velocity of the motor, is compared to the desired motor velocity v_d, and the difference is multiplied by a gain K in the power amplifier to produce a current into the motor.

The tachometer signal provides feedback for an *analog control loop* to compensate for the friction (deadzone) in the motor. Because of friction in the motor, if a small current is put into the motor, the motor will not turn at all. This means that the motor is not actually a linear system in which the angular velocity is proportional to the input current. However, by comparing the *actual* motor velocity with the *desired* motor velocity and multiplying their difference by a large number K, a current is produced which makes the motor move

Quantity	Symbol	Units
Desired motor velocity	v_d	volts
Angular position of motor	θ_m	radians
Angular velocity of motor	$\omega_m = \dot{\theta}_m$	radians/second
Angular position of pendulum	θ	radians
Angular velocity of pendulum	$\omega = \dot{\theta}$	radians/second
Lead-screw ratio	n	radians/meter
Cart position	$x = \theta_m/n$	meters
Cart velocity	$v = \dot{x} = \omega_m/n$	meters/second
Gain of power amplifier	K	amps/volt
Torque constant of motor	K_T	Newton-meter/amp
Tachometer constant	G	volts/radian/sec

Table 3.8 Definition of variables for the cart/pendulum system

with the desired velocity. This means that the input to our system is demanded motor velocity.

The pendulum system we are considering consists of a dc motor which drives a lead screw which drives a cart. The pendulum hangs off the side of the cart, and is free to rotate 360 degrees. The pendulum is mounted to a resolver which senses its angular position, and the resolver-to-digital converter circuit provides the angular velocity of the pendulum. The motor is equipped with an optical encoder and tachometer, which measure its angular position and velocity, respectively. The sensors measure all the state variables of the system. A list of all the important variables of the system and their symbols is given in Table 3.8.

The equations for the cart and pendulum may be found using Newton's laws to write force/torque balance equations as in [31]. The resulting equations are

$$(I_1 + ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta \tag{3.157}$$

and

$$(M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \tag{3.158}$$

where the variables have the following definitions:

I_1 is the moment of inertia of the pendulum about its center of mass
m is the mass of the pendulum
l is the distance from the pivot point to the pendulum's center of mass
θ is the angular position of the pendulum
g is the acceleration due to gravity
x is the position of the cart
M is the mass of the cart
F is the force exerted on the cart.

It will turn out that most of the parameters listed above do not have to be known, but only certain combinations of them. To see this, we first rewrite (3.157) as

$$\ddot{\theta} + A \sin\theta = -\frac{A}{g}\ddot{x}\cos\theta \tag{3.159}$$

where $A = (mgl)/(I_1 + ml^2)$. If we set $\ddot{x} = 0$ by holding the cart in a fixed position, and linearize the above equation about $\theta = 0$ we get

$$\ddot{\theta} + A\theta = 0. \tag{3.160}$$

This is just the harmonic oscillator discussed earlier, and for any nonzero initial value of θ, the solution to the above equation is that $\theta(t)$ is sinusoidal with radian natural frequency of $\sqrt{A}$ radians per second. Thus we do not need to know the values of I_1, m, and l. It is sufficient to let the pendulum swing freely and measure its natural frequency in radians per second. The square of this number is the parameter A in (3.159).

The equations for the motor are as follows. The torque output τ which the motor exerts on the lead screw is

$$\tau = \tau_{\text{in}} - J\dot{\omega}_m - B\omega_m \tag{3.161}$$

where τ_{in} is the torque produced within the motor, and J and B are the inertia and damping of the motor, respectively [15]. The torque produced within the motor is proportional to the current through the motor, with a proportionality constant K_T, so that (see Table 3.8 and Fig. 3.25)

$$\tau_{\text{in}} = K_T K(v_d - G\omega_m). \tag{3.162}$$

Finally, the force exerted on the cart is related to the torque exerted on the lead screw by the transduction ratio for the lead screw n:

$$F = n\tau. \tag{3.163}$$

Substituting (3.161)–(3.163) into (3.158) together with the fact that $\omega_m = n\dot{x}$ and $\dot{\omega}_m = n\ddot{x}$ yields

$$\begin{aligned} nK_T K v_d \;=\; & (\frac{M+m}{n} + nK_T KJ)\dot{\omega}_m + (\frac{b}{n} + nB + nK_T KG)\omega_m + \\ & ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta. \end{aligned} \tag{3.164}$$

If we divide both sides of this equation by the coefficient of $\dot{\omega}_m$, it turns out that the terms involving θ are negligible. This is another advantage of using the analog velocity

$A = 23.1$	rad/sec^2
$C = 25.0$	sec^{-1}
$D = 2,633.$	rad/(volt-sec^2)
$n = 495$	rad/m
$g = 9.81$	m/sec^2

Table 3.9 The parameter values for (3.166) and (3.167) corresponding to the hardware system at the University of Rhode Island.

loop around the motor: it not only reduces the effect of friction on the motor, but it also decouples the pendulum and motor dynamics.

The resulting equation is then

$$\dot{\omega}_m + C\omega_m = Dv_d \tag{3.165}$$

for some constants C and D. This is just a one time-constant system. The step response of this system has a time constant equal to $1/C$ and a steady-state value of DV_d/C for a voltage step equal to V_d volts. Thus the parameters C and D can be obtained from a step response of the cart. The parameters A, B, C, n and g (gravity), are all that is needed to specify the behavior of the cart/pendulum system.

In fact, (3.159) and (3.165) can be used to write down a state-space description of the cart/pendulum system if the input and state variables are chosen as follows:

u = D/A voltage from computer
$x_1 = \theta$ (pendulum position)
$x_2 = \dot{\theta}$ (pendulum velocity)
$x_3 = \theta_m$ (motor position)
$x_4 = \omega_m$ (motor velocity)

With these definitions, (3.159) and (3.165) can be written as

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -A\sin(x_1) - \frac{A}{ng}\cos(x_1)(-Cx_4 + Du) \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= -Cx_4 + Du.
\end{aligned} \tag{3.166}$$

The numerical values for the parameters in (3.166) which correspond to the hardware system used at the University of Rhode Island are shown in Table 3.9.

Linearized models can be obtained from (3.166) for the case when the pendulum is hanging down, and also when it is pointing up. When the pendulum is hanging down, we define the first state variable to be $x_1 = \theta$, and consider only small deviations about zero. A general linearization technique is given in the Appendix to this Chapter. Here, we simply note that for values of x_1 near zero, $\sin x_1 \approx x_1$ and $\cos x_1 \approx 1$.

When the pendulum is pointing up, we define the first state variable to be $x_1 = \theta - \pi$, and consider only small deviations about the vertical $\theta = \pi$. For this range of θ, $\sin x_1 \approx$

$-x_1$ and $\cos x_1 \approx 0$. Using these approximations, (3.166) becomes a linear state-space model for either the hanging or inverted pendulum. These linear equations are shown below,

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \pm A & 0 & 0 & \mp\frac{AC}{ng} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -C \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ \pm\frac{AD}{ng} \\ 0 \\ D \end{bmatrix} u(t) \tag{3.167}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t)$$

where the top plus or minus signs are used to describe the inverted pendulum, and the bottom signs are used to describe the hanging pendulum. For future reference, we give the linearized state-space model for the inverted pendulum using the parameter values from Table 3.9:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 23.1 & 0 & 0 & -0.1189 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -25.0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 12.5 \\ 0 \\ 2633 \end{bmatrix} u(t) \tag{3.168}$$

3.7 Chapter Summary

In this chapter we considered transfer function and state-space models for continuous- and discrete-time systems. We briefly reviewed Laplace and z transforms. These results were used to find a necessary condition for the step response of a standard, second-order, continuous-time system and the step response of a discrete-time sampled data system to be equal at sampling instants. The condition was that the poles of the discrete-time system had to be related to the poles of the continuous-time system by

$$z_{1,2} = e^{s_{1,2}T}$$

where T is the sampling interval. This result is shown to be true in general in Chapter 4.

Canonical form state-space models were introduced as a way to obtain state-space models from arbitrary transfer functions. A given transfer function has an infinitely many different state-space models. Given one state-space model, another state-space model can be obtained using a linear transformation matrix. A linear transformation of a state-space model results in a new state-space model which has the same input, output, and transfer function as the original model.

We presented solutions to the state equations for continuous- and discrete-time systems and used these solutions to develop formulas for the impulse responses in terms of the state-space matrices. When two state-space models are connected, the resulting system can be described by a state-space model. We presented state-space models for parallel, cascade, and feedback connections of systems. Finally, state-space and transfer function models for some simple systems were presented, and a state-space model for a pendulum-on-a-cart system was derived.

3.8 Appendix: Linearization of Nonlinear Systems

Recall that a state-space description of a nonlinear system takes the form

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), u(t)) \\ y(t) &= g(\mathbf{x}(t), u(t)), \end{aligned} \tag{3.169}$$

where $\mathbf{x}(t)$ is a vector of n state variables, $u(t)$ is the input and $y(t)$ is the output. In many cases, it is desired to control a nonlinear system's behavior near an *operating point*. An operating point consists of a constant state vector $\mathbf{x}_o$ and a constant input u_o which satisfy the following equation

$$\mathbf{0} = \mathbf{f}(\mathbf{x}_o, u_o). \tag{3.170}$$

Comparing (3.170) with (3.169), we see that if the state vector equals $\mathbf{x}_o$ and the input is held constant at $u(t) = u_o$, then the state vector will remain at $\mathbf{x}(t) = \mathbf{x}_o$, since the derivative $\dot{\mathbf{x}}(t) = 0$. If an operating point exists with the input value $u_o = 0$, we refer to that operating point as an *equilibrium point*. That is, an equilibrium point is a constant state vector $\mathbf{x}_e$ such that

$$\mathbf{0} = \mathbf{f}(\mathbf{x}_e, 0). \tag{3.171}$$

In many cases the behavior of a nonlinear system near an operating point can be described by a *linear time-invariant system*! To see this, let us consider state and input trajectories which are small perturbations away from the operating point

$$\begin{aligned} \mathbf{x}(t) &= \mathbf{x}_o + \boldsymbol{\delta x}(t) \\ u(t) &= u_o + \delta u(t). \end{aligned} \tag{3.172}$$

An example of the nominal input u_o, the perturbation input $\delta u(t)$, and the actual input $u(t) = u_o + \delta u(t)$ is shown in Fig. 3.26. The input and state vector must obey the differ-

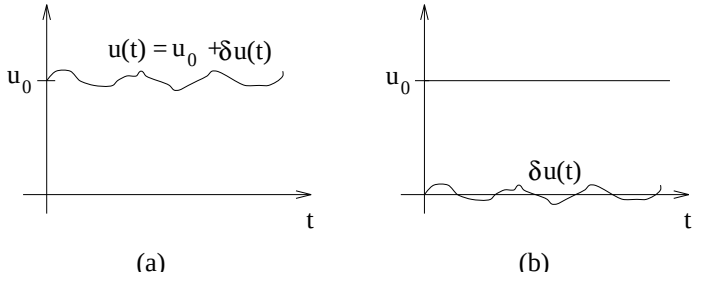


Figure 3.26 (a) Actual input. (b) Nominal and perturbation inputs.

ential equation (3.169), and substituting (3.172) into (3.169) yields

$$\dot{\boldsymbol{\delta x}}(t) = \mathbf{f}(\mathbf{x}_o + \boldsymbol{\delta x}(t), u_o + \delta u(t)). \tag{3.173}$$

Expanding the right-hand side of the above equation in a Taylor series about $(\mathbf{x}_o, u_o)$ and only keeping the linear terms yields

$$\dot{\boldsymbol{\delta x}}(t) = \mathbf{f}(\mathbf{x}_o, u_o) + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{(\mathbf{x},u)=(\mathbf{x}_o,u_o)} \boldsymbol{\delta x}(t) + \frac{\partial \mathbf{f}}{\partial u}\Big|_{(\mathbf{x},u)=(\mathbf{x}_o,u_o)} \delta u(t). \tag{3.174}$$

Notice that $\mathbf{f}(\mathbf{x}_o, u_o) = \mathbf{0}$ by the definition of an operating point. The partial derivatives in the above equation have the following definitions. The derivative of the vector-valued function $\mathbf{f}$ with respect to the vector $\mathbf{x}$ yields a matrix of derivatives

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}. \tag{3.175}$$

The derivative of the vector-valued function $\mathbf{f}$ with respect to the scalar function u yields a vector of derivatives

$$\frac{\partial \mathbf{f}}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \\ \vdots \\ \frac{\partial f_n}{\partial u} \end{bmatrix}. \tag{3.176}$$

When the constant vector $\mathbf{x}_o$ and the constant input u_o are substituted into the partial derivatives on the right-hand side of equation (3.174), the results are a matrix of numbers, call it A, and a vector of numbers, call it $\mathbf{b}$, respectively. Thus (3.174) is a *linear, time-invariant system* which describes how a *perturbation* $\delta u(t)$ around the nominal input u_o causes a *perturbation* around the nominal state $\mathbf{x}_o$. We can rewrite (3.174) as

$$\dot{\boldsymbol{\delta x}}(t) = \mathbf{A}\, \boldsymbol{\delta x}(t) + \mathbf{B}\, \delta u(t). \tag{3.177}$$

Since this is a linear system, we can design a digital controller for it using the theory developed in this book. This digital controller can be used to keep the nonlinear system near its operating point. It is important to remember that $\boldsymbol{\delta x}(t)$ is not the state vector of the nonlinear system, and $\delta u(t)$ is not the input to the nonlinear system. Rather, these quantities are offset by the nominal state vector and nominal input, respectively, as shown in (3.172). In order for the nonlinear system to be regulated by the digital controller designed for the linearized system, the nominal state vector must be subtracted from the actual state vector, and the nominal input must be added to the perturbation input. This is shown in Fig. 3.27. The addition and subtraction of u_0 and $\mathbf{x}_0$ could be done inside the digital controller if

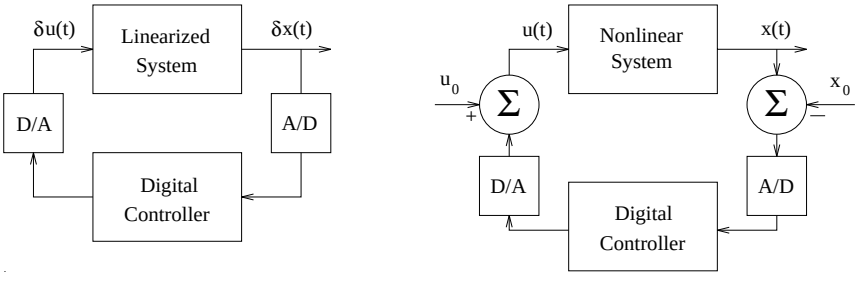


Figure 3.27 (a) Linearized system under digital control. (b) Nonlinear system under digital control. The digital controller is the same as in (a); it is designed for the linearized system.

desired. We also remark that the linearized system shown in Fig. 3.27(a) is a mathematical model of the form (3.177); it is not a physical system.

EXAMPLE 3.18

A nonlinear state-space model for the cart/pendulum system is shown in (3.166) of section 3.6.6. A linearized model was obtained by a "common sense" procedure in Section 3.6.6. In this example, we show that the linearization procedure introduced in this section gives the same linearized model as that obtained earlier.

From (3.166), we see that the elements of the vector function $\mathbf{f}(\mathbf{x}(t), u(t))$ are given by

$$f_1(\mathbf{x}(t), u(t)) = x_2(t)$$

$$f_2(\mathbf{x}(t), u(t)) = -A\sin(x_1(t)) - \tfrac{A}{ng}\cos(x_1(t))(-Cx(t)_4 + Du(t))$$

$$f_3(\mathbf{x}(t), u(t)) = x_4(t)$$

$$f_4(\mathbf{x}(t), u(t)) = -Cx_4(t) + Du(t).$$

The equilibrium point corresponding to the inverted pendulum is specified by

$$\mathbf{x}_e = \begin{bmatrix} \pi \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

It is easy to see that $\mathbf{f}(\mathbf{x}_e, \mathbf{0}) = 0$, so $\mathbf{x}_e$ is indeed an equilibrium point. To find the linear, time-invariant system which describes the behavior of the cart/pendulum system near this equilibrium point, we need to evaluate the derivatives in (3.175) and (3.176). We will suppress the explicit dependence on t for the rest of this example.

It can be seen that the derivative of f_1 with respect to x_i is zero except when $i = 2$, and then it equals one. Thus the first row of the matrix in (3.175) is $[0\ 1\ 0\ 0]$. Also, the derivative of f_1 with respect to u equals zero, so the first element of the vector in (3.176) is zero.

The derivative of f_2 with respect to x_1 is

$$\frac{\partial f_2}{\partial x_1} = -A\cos(x_1) + \frac{A}{ng}\cos(x_1)(-Cx_4 + Du)$$

and the derivative of f_2 with respect to x_4 is

$$\frac{\partial f_2}{\partial x_4} = \frac{AC}{ng}\cos(x_1).$$

The derivatives of f_2 with respect to x_2 and x_3 are both zero, and the derivative of f_2 with respect to u is

$$\frac{\partial f_2}{\partial u} = -\frac{DA}{ng}.$$

When the derivatives in the previous three equations are evaluated at the equilibrium point, the result is that the second row of the matrix in (3.175) equals $[A\ 0\ 0\ (-AC)/(ng)]$, and the second element in the input vector equals $(AD)/ng)$.

The derivatives of f_3 and f_4 with respect to $\mathbf{x}$ and u are easy since f_3 and f_4 are linear in $\mathbf{x}$ and u. If we define a state vector $\mathbf{z} = \boldsymbol{\delta x}$ and substitute the derivatives

calculated above into (3.175) and (3.176), then (3.174) becomes

$$\dot{\mathbf{z}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ A & 0 & 0 & -\frac{AC}{ng} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -C \end{bmatrix} \mathbf{z}(t) + \begin{bmatrix} 0 \\ \frac{AD}{ng} \\ 0 \\ D \end{bmatrix} u(t)$$
$$y(t) = [\,1 \quad 0 \quad 0 \quad 0\,]\,\mathbf{x}(t)$$

■

EXAMPLE 3.19

Consider a nonlinear system described by

$$\dot{x}_1 = x_1 u - 1$$
$$\dot{x}_2 = -x_1^2 + x_2 + u.$$

Find the operating point corresponding to a nominal input of $u_o = 1$, and obtain the linear differential equations which describe the behavior of this system near the operating point.

We begin by using (3.170) to find the the operating point. Let the operating point be given by

$$\mathbf{x}_o = \begin{bmatrix} x_{o1} \\ x_{o2} \end{bmatrix}.$$

Then (3.170) with $u_o = 1$ yields

$$0 = x_{o1} - 1$$
$$0 = -x_{o1}^2 + x_{o2} + 1.$$

Notice that the operating point is found by solving for the roots of nonlinear equations, which may have multiple solutions. In this example, the above equation has the unique solution

$$x_{o1} = 1\,, \quad x_{o2} = 0.$$

The linear differential equation shown in (3.174) can now be found by taking partial derivatives and substituting in the values of the nominal input and operating point. The result is

$$A = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \left.\begin{bmatrix} u & 0 \\ -2x_1 & 1 \end{bmatrix}\right|_{\substack{u=1 \\ x_1=1 \\ x_2=0}} = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix}.$$

and

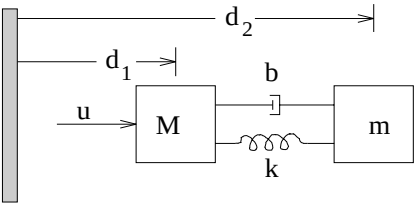
$$\mathbf{B} = \frac{\partial \mathbf{f}}{\partial u} = \left.\begin{bmatrix} x_1 \\ 1 \end{bmatrix}\right|_{\substack{u=1 \\ x_1=1 \\ x_2=0}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Thus the linear differential equations which describes the behavior of the nonlinear system near the given operating point is

$$\dot{\boldsymbol{\delta x}} = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix} \boldsymbol{\delta x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \delta u.$$

3.9 Problems

1. Many mechanical systems exhibit some degree of structural flexibility. The robot arm on the space shuttle, for example, is not a perfectly rigid structure. A simple one degree-of-freedom model for such a system is shown below



m,M = masses
b = damping coefficient
k = spring constant
d_1 = position of mass M
d_2 = position of mass m
u = applied control force

The input is $u(t)$ and the output is $d_2(t)$. This system contains structural flexibility between the sensor and the actuator. The behavior of this system is described by the following coupled differential equations

$$M\ddot{d}_1 + (\dot{d}_1 - \dot{d}_2)b + (d_1 - d_2)k = u$$

$$m\ddot{d}_2 + (\dot{d}_2 - \dot{d}_1)b + (d_2 - d_1)k = 0.$$

In order to obtain a state-space description for this system we define the following state variables

$$\begin{aligned} x_1 &= d_1 \\ x_2 &= d_2 \\ x_3 &= \dot{d}_1 \\ x_4 &= \dot{d}_2 \end{aligned}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}.$$

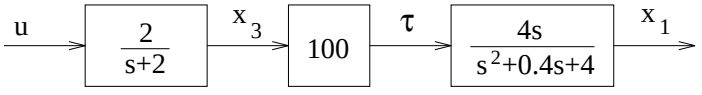
A state-space description is of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

$$d_2 = \mathbf{C}\mathbf{x}.$$

Find $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$.

2. Consider the following block diagram

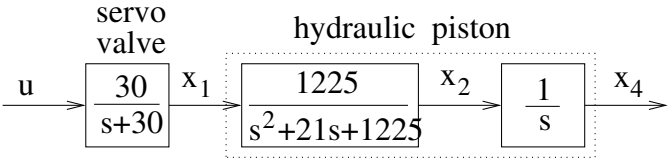


(a) Write a first-order differential equation relating u and x_3.

(b) Use a canonical form to get a state-space description for the system shown in the second block (i.e. the system with τ as input and x_1 as output).

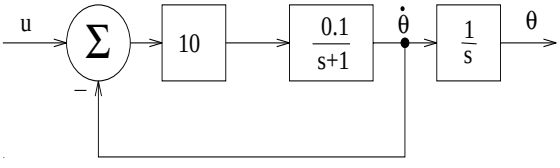
(c) Combine the results of (a) and (b) to obtain a state-space description for the complete system shown in the figure (i.e. the system with u as input and x_1 as output). This state-space description will be third order, and two of the state variables should be x_1 and x_3 shown in the diagram.

3. A hydraulic positioning system can be represented by the following block diagram where the variables are defined as follows: x_1 is valve position, x_2 is flow, and x_4 is



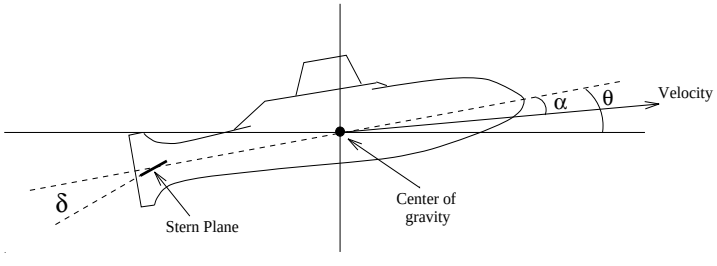
the position of the piston. Obtain a state-space model for this system which uses x_1, x_2, and x_4 as 3 of the 4 state variables.

4. Consider the system described by the following block diagram:

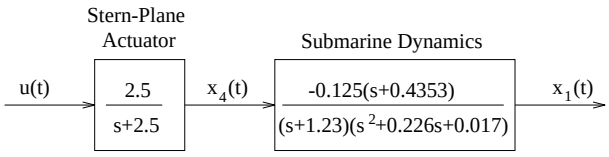


Let the state variables for this system be $x_1 = \theta$ and $x_2 = \dot{\theta}$. Obtain the state-space model which uses these state variables.

5. The dynamics of a submarine in the vertical plane are described using the variables shown in the following picture



The angle θ is the *pitch angle*, α is the *angle of attack*, and δ is the *deflection of the stern plane*. If θ is small and the velocity of the submarine is constant, the motion of the submarine can be described by a linear system model. A block diagram for this model is shown below:



(a) Obtain the state-space model corresponding to the block diagram shown above. Let two of the four state variables be $x_1 = \delta$ and $x_4 = \theta$ shown in the figure. Use observable canonical form for the submarine dynamics block.

(b) Suppose we want a state-space model for the submarine dynamics block which has the pitch angle θ as a state variable (call it $\bar{x}_1$) and also has pitch rate $\dot{\theta}$ as a state variable (call it $\bar{x}_2$). Find a linear transformation matrix $\mathbf{T}$ which transforms the model found in (a) to the desired form. Calculate the transformed model $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}})$ and verify that it has the correct form. *Hints:* let the state variables of the transformed model be given by

$$\bar{\mathbf{x}}(t) = T\mathbf{x}(t).$$

If pitch angle is to be the first state variable, and also the output of the system, then the output vector $\bar{\mathbf{c}}$ must equal $[\,1 \quad 0 \quad 0\,]$ or

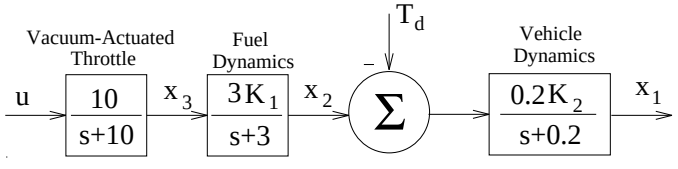
$$\mathbf{C}\mathbf{T}^{-1} = [\,1 \quad 0 \quad 0\,].$$

Also, in order for $\dot{\bar{x}}_1$ to equal $\bar{x}_2$ the first row of $\bar{\mathbf{A}}$ must equal $[\,0 \quad 1 \quad 0\,]$ or

$$[\,1 \quad 0 \quad 0\,]\,\mathbf{T}\mathbf{A}\mathbf{T}^{-1} = [\,0 \quad 1 \quad 0\,].$$

The previous two equations can be used to calculate the first two rows of the transformation matrix $\mathbf{T}$. The only requirement on the third row of $\mathbf{T}$ is that it make $\mathbf{T}$ a nonsingular matrix.

6. A block diagram of the forward path of an automotive cruise control system is shown below: Obtain a state-space models using the indicated state variables. T_d is a load

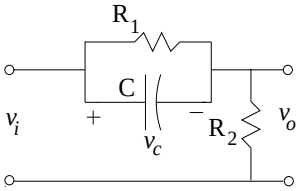


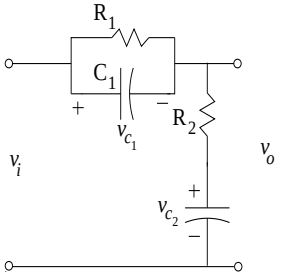
torque which would occur when the car went up a hill. This system has two inputs, u and T_d, and so a state-space model will have two input vectors, one for each input.

7. The following figure shows a passive lead network that could be used as an analog compensator

Choose the state variable to be the voltage v_c and write a state-space description for the circuit with input $u(t) = v_i(t)$ and output $y(t) = v_o(t)$.

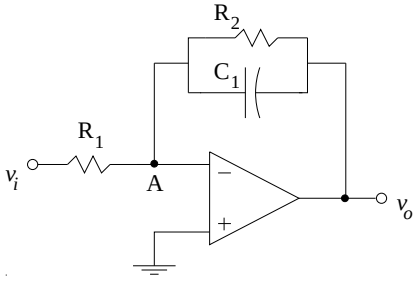
8. The following figure shows a passive lead-lag filter Choose the state variables to be





the voltages v_{c1} and v_{c2} and write a state-space description for the circuit with input $u(t) = v_i(t)$ and output $y(t) = v_o(t)$.

9. Op-amps are used to synthesize active filters like the one shown below.

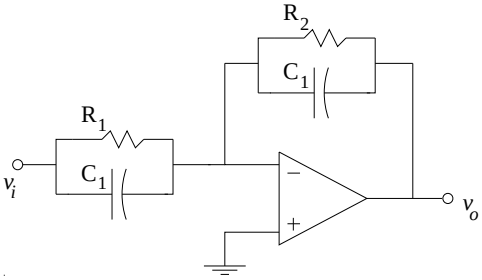


State-space descriptions can be obtained for many active filters by using two facts: (i) the current into the inverting input of the op-amp equals zero, and (ii) the potential at the inverting input equals the potential at the non-inverting input. When the non-inverting input is grounded, as in the above circuit, the inverting input is also effectively grounded. Thus for the above circuit, we can sum the currents into node A and set the result equal to zero

$$\frac{v_i}{R_1} + \frac{v_o}{R_2} + C_1 \dot{v}_o = 0.$$

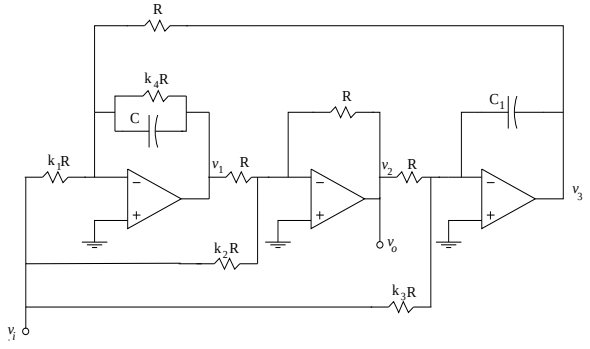
This first-order differential equation is effectively a state-space description of the circuit with state variable v_o.

(a) Obtain a state-space description for the filter shown here



Show that any real-valued pole and zero locations in the left half plane can be obtained by choosing appropriate values for the resistors and capacitors.

(b) By cascading two of the filters shown in (a) together, a filter with two real poles and two real zeros can be obtained. Obtain a state-space description for the following filter, and show that the filter can realize a second-order transfer function with complex poles and zeros.



10. Show that a continuous-time system is BIBO stable if and only if its impulse response $h(t)$ satisfies the following condition

$$\int_0^\infty |h(t)|\, dt < \infty.$$

11. Show that a discrete-time system is BIBO stable if and only if its impulse response $h[k]$ satisfies the following condition

$$\sum_0^\infty |h[k]| < \infty.$$

12. Show that the following identity is true for any matrix $\mathbf{M}$ which has all its eigenvalues inside the unit circle

$$\sum_{k=0}^{\infty} \mathbf{M}^k = (\mathbf{I} - \mathbf{M})^{-1}.$$

You can assume (it can be shown) that $\lim_{n\to\infty} \sum_{k=0}^{n} \mathbf{M}^k$ exists. Hint: multiply the summation by $(\mathbf{I} - \mathbf{M})$.

13. Derive the following identity for any matrix M whose eigenvalues all have positive real parts

$$\int_0^\infty e^{-\mathbf{M}t}\,dt = \mathbf{M}^{-1}.$$

Hint: start by taking the derivative of $e^{-\mathbf{M}t}$. Multiply both sides by dt and integrate from 0 to ∞.

14. Consider the following transfer function for a discrete-time system

$$H(z) = \frac{z^2 - 0.5z}{z^2 - 1.85z + 0.855}.$$

(a) Write a state-space description in controllable canonical form corresponding to the above transfer function.

(b) Assume $\mathbf{x}[0] = \mathbf{0}$. If $u[0] = 1$, $u[1] = 2$, and $u[2] = 3$, what will the state vector $\mathbf{x}[3]$ equal?

(c) Assume $\mathbf{x}[0] = \mathbf{0}$. Find input values $u[0]$ and $u[1]$ so that the state vector $\mathbf{x}[2]$ takes on the same value as the vector $\mathbf{x}[3]$ found in part (b).

15. Derive the results given in (3.141) for a state-space representation of the feedback interconnection of two systems.

16. Consider the following system

$$\begin{aligned}\mathbf{x}[k+1] &= \mathbf{A}\mathbf{x}[k] + \mathbf{B}u[k], \quad \mathbf{x}(0) = \mathbf{x}_0\\ y[k] &= \mathbf{C}\mathbf{x}[k] + Du[k], \quad k \geq 0.\end{aligned}$$

(a) Suppose the input is zero for all time and D is arbitrary. Can the initial condition vector $\mathbf{x}_0$ and D be chosen so that the output $y[k]$ equals $h[k]$, the impulse response of the system? Explain your answer.

(b) Find the transfer function of the above system when $A = -2$, $b = 1$, $c = 1$, and $d = -2$.

17. Many mechanical systems can be described by second-order vector differential equations. An example of such an equation is

$$\begin{bmatrix}\ddot{x}_1\\ \ddot{x}_2\end{bmatrix} + \begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} + \begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} = \begin{bmatrix}1\\ 0\end{bmatrix}u(t).$$

In order to use state-space techniques for control system design, a state-space description must be obtained, which is a first-order vector differential equation. We can obtain a state-space description which is equivalent to the second-order differential equation shown above by defining the following state variables

$$z_1 = x_1\,, \quad z_2 = x_2\,, \quad z_3 = \dot{x}_1\,, \quad z_4 = \dot{x}_2.$$

Note that x_i and z_i are all functions of time.

(a) Find matrices $\mathbf{A}$ and $\mathbf{B}$ so that the state-space description $\dot{\mathbf{z}}(t) = \mathbf{A}\mathbf{z}(t) + \mathbf{B}\mathbf{u}(t)$ describes the same system as the second-order equation given at the beginning of

this problem, where

$$\mathbf{z}(t) = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}.$$

(b) Now consider a more general form of second-order vector differential equations shown below

$$\ddot{\mathbf{x}} + \mathbf{A}_1\dot{\mathbf{x}} + \mathbf{A}_2\mathbf{x} = \mathbf{B}\mathbf{u}(t),$$

where $\mathbf{x}$ is an $n \times 1$ vector, $\mathbf{A}_1$ and $\mathbf{A}_2$ are $n \times n$ matrices, and $\mathbf{B}$ is an $n \times 1$ matrix. Define the following $n \times 1$ vectors: $\mathbf{z}_1 = \mathbf{x}$ and $\mathbf{z}_2 = \dot{\mathbf{x}}$. Also define the following $2n \times 1$ vector

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix}.$$

Find $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ in terms of $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{B}$ so that $\dot{\mathbf{z}} = \bar{\mathbf{A}}\mathbf{z} + \bar{\mathbf{B}}\mathbf{u}(t)$.

18. Consider a single-input system described by the state-space equations

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u\,, \quad \mathbf{x}(0) = \mathbf{0} \\ y &= \mathbf{C}\mathbf{x}. \end{aligned}$$

Suppose that this system is stable so that the eigenvalues of $\mathbf{A}$ all have negative real parts. Then it can be shown that the step response of this system (i.e. the response to $u(t) = 1$ for $t > 0$) results in a well-defined steady-state value for the state vector. That is, $\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}_{ss}$, where $\mathbf{x}_{ss}$ is a vector of numbers (a constant).

(a) Explain why $\mathbf{A}$ is an invertible matrix.

(b) Find an expression for $\mathbf{x}_{ss}$ by simply plugging a constant vector $\mathbf{x}_{ss}$ into the state equations with u set equal to 1.

(c) Use the results of (b) and (3.126) to show that the following identity is true when $\mathbf{A}$ is a stable matrix.

$$\lim_{t\to\infty} \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{b}d\tau = -\mathbf{A}^{-1}\mathbf{B}.$$

19. Consider a single-input system described by the state-space equations

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{A}\mathbf{x}[k] + \mathbf{B}u[k]\,, \quad \mathbf{x}[o] = \mathbf{0} \\ y[k] &= \mathbf{C}\mathbf{x}[k]. \end{aligned}$$

Suppose that this system is stable so that the eigenvalues of $\mathbf{A}$ are all within the unit circle. Then it can be shown that the step response of this system (i.e. the response to $u(k) = 1$ for $k \geq 0$) results in a well-defined steady-state value for the state vector. That is, $\lim_{k\to\infty} \mathbf{x}[k] = \mathbf{x}_{ss}$, where $\mathbf{x}_{ss}$ is a vector of numbers (a constant).

(a) Explain why $\mathbf{I} - \mathbf{A}$ is an invertible matrix, where $\mathbf{I}$ is the identity matrix with dimensions identical to $\mathbf{A}$.

(b) Find an expression for $\mathbf{x}_{ss}$ by simply plugging a constant vector $\mathbf{x}_{ss}$ into the state equations with u set equal to 1.

(c) Use the results of (b) and (3.114) to show that the following identity is true when A is a stable matrix.

$$\lim_{k\to\infty} \sum_{j=1}^{k} \mathbf{A}^{k-j}\mathbf{B} = (\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}.$$

Hint: If λ is an eigenvalue of a matrix $\mathbf{M}$, then $1+\lambda$ is an eigenvalue of the matrix $\mathbf{I}+\mathbf{M}$.

20. Consider the following system

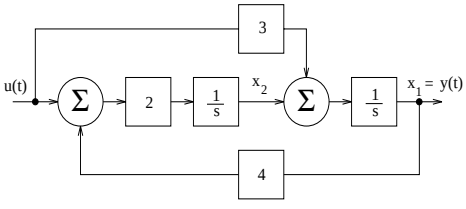
$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}u[k]$$
$$y[k] = \mathbf{C}\mathbf{x}[k]$$

for $k > 0$ with $\mathbf{x}[0] = \mathbf{x}_0$.

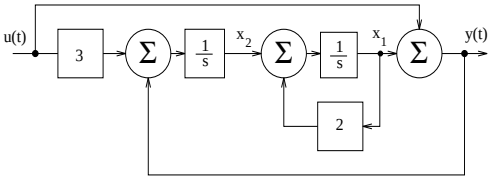
(a) Assuming that $\mathbf{x}_0 = \mathbf{0}$, find the system's impulse response sequence $\{h[k]\}$. (Note – this is the output sequence $y[k]$ when the input is $u[0] = 1$, $u[k] = 0$, $k > 0$.)

(b) Let $u[k] = 0$ for all k. Can x_0 be chosen so that the resulting output equals the impulse response $\{h[k]\}$? Explain.

21. For each block diagram shown below, obtain the state-space description and the transfer function.



(a)



(b)

22. Use the definition of the matrix exponential to verify the properties listed in Table 3.6 on page 130.

23. A state-space model of the cart-pendulum system is given in (3.168). This model uses to the following state variables

$x_1 = \theta$ (pendulum position)
$x_2 = \dot{\theta}$ (pendulum velocity)
$x_3 = \theta_m$ (motor position)
$x_4 = \omega_m$ (motor velocity).

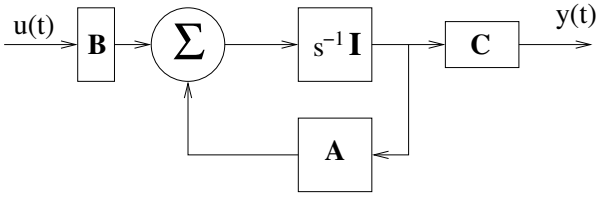
Find the state-space model which uses the state variables shown below

$\bar{x}_1 = \theta$ (pendulum position)
$\bar{x}_2 = \theta_m$ (motor position)
$\bar{x}_3 = \dot{\theta}$ (pendulum velocity)
$\bar{x}_4 = \omega_m$ (motor velocity).

24. Use the Fadeev algorithm to calculate the transfer function of the following system

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t)$$
$$y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x}(t).$$

25. A continuous-time state-space model can be represented by the block diagram shown below.



The block diagram can be manipulated to yield a new system called the *transpose* system using the following rules:

(i) Reverse the directions of all the arrows;

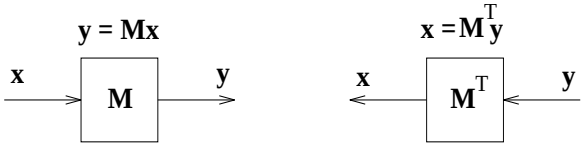
(ii) Change summing junctions to nodes and nodes to summing junctions;

(iii) Interchange the input and output terminals.

Rule (i) is meant to be applied to lines carrying a single variable, but the block diagram shown above has lines represented vector-valued variables. The first part of this problem shows how to handle vector-valued lines.

(a) Consider the block diagram corresponding to $\mathbf{y} = \mathbf{M}\mathbf{x}$ shown in Figure (a) below, where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad \mathbf{M} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}.$$



(a) (b)

Re-draw block diagram (a) with two scalar inputs and two scalar outputs and show that rules (i) and (ii) of block diagram transposition yield the block diagram in Figure (b).

(b) Use the rules of block diagram transposition together with the result from (a) to transpose the block diagram of the state-space model shown at the beginning of this problem.

(c) Let the state-space matrices for the transposed system be denoted $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{D})$. Find these matrices in terms of the original system matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D)$.

(d) Show that a system and its transpose have the same transfer function $H(s)$. (*Hint*: Take the matrix transpose $H^T(s)$ of the 1×1 matrix $H(s)$. Since $H(s)$ is a scalar, $H^T(s) = H(s)$.)

(e) Show that controllable canonical form is the transpose system of observable canonical form.

26. It is well known that the poles of a linear system are equal to the eigenvalues of the matrix $\mathbf{A}$ of state-space model. In this problem we show that the zeros of the system can be computed as the eigenvalues of a certain matrix provided the feedthrough coefficient is not equal to zero.

Let $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D)$ be a state-space model for the transfer function $H(s) = b(s)/a(s)$ with input $u(t)$ and output $y(t)$. Assume that $d \neq 0$ so that $b(s)$ and $a(s)$ have the same degree. In this case we can define the *inverse system* with input $y(t)$, output $u(t)$, and transfer function $1/H(s) = a(s)/b(s)$. Notice that the zeros of $H(s)$ are the poles of the inverse system.

(a) Obtain a state-space model for the inverse system from the given state-space model for $H(s)$. [*Hint:* Solve the output equation for $u(t)$ in terms of $y(t)$. Substitute this expression into the state equation.]

(b) Show that the zeros of $H(s)$ are the eigenvalues of the matrix $\mathbf{A} - \mathbf{B}D^{-1}\mathbf{C}$.

27. Let $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D)$ be a state-space model for the transfer function $H(s)$ with input $u(t)$ and output $y(t)$. The purpose of this problem is to show that with an input of the form $u(t) = e^{\gamma t}$, if γ is a zero of $H(s)$, then we can choose the initial state vector $\mathbf{x}(0)$ so that $y(t) = 0, t \geq 0$. We assume that there are no pole-zero cancellations in $H(s)$ so that if γ is a zero of the system, then γ is not an eigenvalue of $\mathbf{A}$.

(a) We start with a first-order example. Suppose

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

Write a state-space model for $H(s)$ and compute $y(t)$ corresponding to the input $u(t) = e^{-t}$. Show that the (scalar) initial state $x(0)$ can be chosen to make $y(t) = 0,\ t \geq 0$.

(b) Consider a general state-space model and let γ be a zero of the system ($H(\gamma) = 0$). Show that if $u(t) = e^{\gamma t}$ and $\mathbf{x}(0) = (\gamma\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$, then the output of the system will be zero for all $t \geq 0$. [*Hint:* Table 3.6 on page 130 shows how to integrate a matrix exponential. Also note that a scalar exponential times a vector can be written as a matrix exponential times a vector as follows: $e^{\gamma}\mathbf{B} = e^{\gamma}\mathbf{I}\mathbf{B} = e^{\gamma\mathbf{I}}\mathbf{B}$.]

28. Problem 26 showed that the zeros of a system can be computed as the eigenvalues of a certain matrix for the special case in which the numerator and denominator polynomials have the same degree, or equivalently, the state-space model has a non-zero feedthrough coefficient. The purpose of this problem is to derive a more general result about system zeros. We use the same notation as the previous problem, and start by defining the matrix

$$\mathbf{S}(\gamma) \stackrel{\text{def}}{=} \begin{bmatrix} \gamma\mathbf{I} - \mathbf{A} & -\mathbf{B} \\ \mathbf{C} & D \end{bmatrix}.$$

It is easy to verify that the following result is true for any number γ that is not an eigenvalue of $\mathbf{A}$

$$\begin{bmatrix} \gamma\mathbf{I} - \mathbf{A} & -\mathbf{B} \\ \mathbf{C} & D \end{bmatrix} \begin{bmatrix} (\gamma\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ H(\gamma) \end{bmatrix}.$$

The above result shows that if γ is a zero of the system, then the matrix $\mathbf{S}(\gamma)$ has a non-trivial null-space vector. In other words, $H(\gamma) = 0$ implies that $\mathbf{S}(\gamma)$ is rank deficient. The steps (a) and (b) of this problem show that the reverse implication is also true, namely, $\mathbf{S}(\gamma)$ rank deficient implies that $H(\gamma) = 0$. The two implications taken together provide the following characterization of system zeros: **The number γ is a zero of the system if and only if $\mathbf{S}(\gamma)$ is rank deficient**.

(a) Assume that for some given number γ which is not an eigenvalue of $\mathbf{A}$, the matrix $\mathbf{S}(\gamma)$ is rank deficient. Then there exists a non-trivial null-space vector as shown below

$$\begin{bmatrix} \gamma\mathbf{I} - \mathbf{A} & -\mathbf{B} \\ \mathbf{C} & D \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ f \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}. \tag{3.178}$$

Show that f cannot equal zero. [*Hint:* Assume that $f = 0$ and use a proof by contradiction.]

(b) Using the same assumptions as part (a), solve for $\mathbf{e}$ in terms of f from the first block row of (3.178) Substitute this expression into the equation $\mathbf{C}\mathbf{e} + Df = 0$ obtained from the last row of (3.178). Show that $H(\gamma) = 0$.

29. Let $(\mathbf{A}_i,\ \mathbf{B}_i,\ \mathbf{C}_i)$, $i = 1, 2, 3$ be three n^{th}-order systems. System i is

$$\begin{aligned} \mathbf{x}_i[k+1] &= \mathbf{A}_i\mathbf{x}_i[k] + \mathbf{B}_i u_i[k], \quad x(0) = x_0 \\ y_i[k] &= \mathbf{C}_i x_i[k] + D_i u_i[k], \quad k \geq 0. \end{aligned}$$

Suppose system 2 is related to system 1 by the linear transformation $\mathbf{T}_1$ and system 3 is related to system 2 by the linear transformation $\mathbf{T}_2$. Show that systems 1 and 3 are related by a linear transformation. Give a formula for system 3 in terms of system 1.

30. Let

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

(a) Obtain a state-space description for $H(s)$ in observable canonical form. Let the state vector be called $\mathbf{x}(t)$, and the system matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D)$.

(b) Define a new state vector

$$\bar{\mathbf{x}}(t) = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \mathbf{x}(t)$$

and calculate the state space description $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{D})$ corresponding to the state vector $\bar{\mathbf{x}}$.

(c) Show by direct calculation that the transfer function of $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{D})$ is the given $H(s)$.

31. Let $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ be a continuous-time system and $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{E}, G)$ be a discrete-time system as shown below

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \quad & \mathbf{x}[k+1] &= \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k] \\ y(t) &= \mathbf{C}\mathbf{x}(t) & y[k] &= \mathbf{E}\mathbf{x}[k] + Gu[k]. \end{aligned}$$

Let $h_c(t)$ be the impulse response of the continuous-time system, and let $h_d[k]$ be the impulse response of the discrete-time system. Suppose we want $h_d[k]$ to equal $h_c(kT)$, $k = 0, 1, \cdots$ for some sampling period T. A discrete-time system whose impulse response equals the sampled impulse response of a continuous-time system is said to be *impulse invariant*. Find expressions for $\mathbf{\Phi}$, $\mathbf{\Gamma}$, $\mathbf{E}$, and G in terms of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and T to obtain the impulse invariant discrete-time system.

32. Consider a discrete-time system $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C}, D)$ and a continuous-time system $(\mathbf{A}, \mathbf{B}, \mathbf{E}, F)$, both with zero initial conditions.

(a) Show that the step response of the discrete system can be written as

$$S_d[k] = \begin{cases} D & k = 0 \\ \sum_{j=0}^{k-1} \mathbf{C}\mathbf{\Phi}^{k-j-1}\mathbf{\Gamma} + D & k \geq 1. \end{cases}$$

(b) Show that the step response of the continuous system can be written as

$$S_a(t) = \mathbf{E}\int_0^t e^{\mathbf{A}(t-\tau)}d\tau\ \mathbf{B} + F.$$

The purpose of the rest of the problem is to derive the so-called "step-invariant" discrete-time model. That is, we want to find the discrete-time system with the property that $S_d[k] = S_a(kT)$. The following steps lead you through a derivation of this model. You can use a different argument to obtain the answer if you like.

(c) Using the results of part (b), show that $S_a(kT)$ can be written as

$$S_a(kT) = \begin{cases} F & k = 0 \\ \mathbf{E}\sum_{j=0}^{k-1}\int_{jT}^{(j+1)T} e^{\mathbf{A}(kT-\tau)}d\tau\ \mathbf{B} + F & k \geq 1. \end{cases}$$

(d) In the integral, make the following change of variables: $\gamma = (j+1)T - \tau$, and pull terms outside the integrand that do not depend on γ. (Hint: $e^{\mathbf{A}\alpha}e^{\mathbf{A}\beta} = e^{\mathbf{A}(\alpha+\beta)}$, where $\mathbf{A}$ is a matrix and α and β are real numbers.)

(e) Show that the step invariant model is defined by

$$\mathbf{\Phi} = e^{\mathbf{A}T}\ ,\ \ \mathbf{\Gamma} = \int_0^T e^{\mathbf{A}\gamma}d\gamma\ \mathbf{B}\ ,\ \ \mathbf{C} = \mathbf{E}\ ,\ \ D = F.$$

33. Consider the following system

$$\mathbf{x}[k+1] = \begin{bmatrix} 1.5 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} 1 \\ -.5 \end{bmatrix} u[k]$$
$$y[k] = [\,1 \quad 0\,]\, x[k].$$

(a) Assume that the state vector at time zero is the zero vector. Calculate the input sequence (as a function of γ) which will cause $\mathbf{x}[2]$ to equal $[1 \ \gamma]^T$, where γ is a real number and the superscript "T" stands for transpose.

(b) If $u[0] = 2$, what is the value of $u[1]$ that will cause the system output to equal 1 at $k = 2$?

(c) How many different input sequences will cause the system output to equal 1 at $k = 2$?

CHAPTER 4

DISCRETE-TIME MODELS OF ANALOG PLANTS

As mentioned in Chapter 1, the analysis and design of a digital control system is complicated by the fact that the plant is a continuous-time system while the compensator is a discrete-time system. One way of dealing with this problem to replace the plant with a discrete-time model which specifies its behavior at sampling instants. In this way, the entire control system becomes discrete-time, and standard discrete-time system theory can be used to design the compensator. In this chapter we show how to obtain discrete-time models of linear, continuous-time plants. The discrete-time models derived in this chapter are *not* approximations. Rather, they describe *exactly* the behavior of the plants at sampling instants. The reason that the discrete-time models are exact is that the input to the plant is a piecewise-constant signal – it is the output of a D/A converter (i.e. a zero-order hold, ZOH) from the digital compensator. This special form of the plant input makes it possible to derive an exact discrete-time model called the *ZOH equivalent model* for the plant. The ZOH model is sometimes called a *sampled-data* model for the plant.

4.1 ZOH Equivalent Models for Analog Systems

In order to motivate the development of the ZOH equivalent model, consider the computer controlled regulation system shown in Fig. 4.1. This regulator may be thought of as a special case of the general digital control system shown in Fig. 1.5 with the reference input equal to zero and the minus sign on the summing junction absorbed into the $C(z)$.

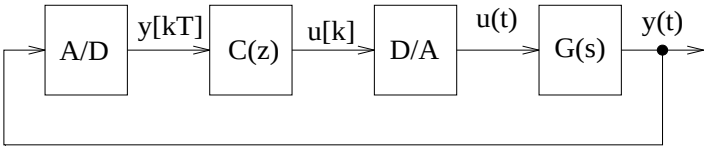


Figure 4.1 A computer controlled regulation system

Note that the digital compensator $C(z)$ only requires the sampled values $\{y(kT)\}$ and not the function $y(t)$ for all t. Similarly, the compensator produces the control sequence $\{u(kT)\}$. The input to the plant, $u(t)$, is a ZOH version of $\{u(kT)\}$, so $u(t)$ cannot change its value between the sampling instants. Thus, as far as the digital compensator is concerned, all that needs to be described is the relationship between $\{u(kT)\}$ and $\{y(kT)\}$. That is, the relationship between a control sequence $\{u(kT)\}$ and the corresponding sampled output sequence $\{y(kT)\}$ needs to be described. The system that relates these two sequences is the system which is "seen" by the digital compensator. This is shown in Fig. 4.2.

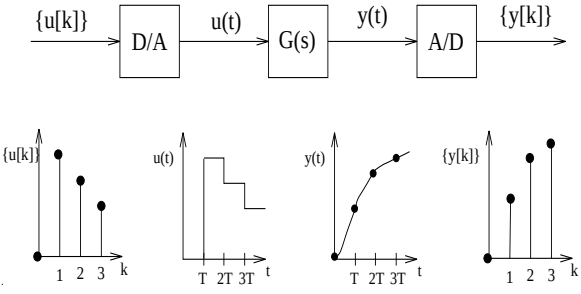


Figure 4.2 The system seen by the digital compensator $C(z)$

4.1.1 Model Development

The discrete-time sequences are related to the continuous-time signals as follows. The continuous-time input to the plant is a zero-order hold of the compensator output

$$u(t) = u[k],\ kT \le t < kT + T, \tag{4.1}$$

and the output of the plant is sampled by an A/D converter

$$y[k] = y(kT). \tag{4.2}$$

Assume that we have a state-space model $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D)$ for the plant $G(s)$; that is, the behavior of the plant is governed by the following equations

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,u(t) \\ y(t) &= \mathbf{C}\,\mathbf{x}(t) + D\,u(t). \end{aligned} \tag{4.3}$$

Since (4.3) is a first-order (vector) differential equation, if the value of $\mathbf{x}(t)$ is known at some time t_0, then the value of $\mathbf{x}(t)$ at future times can be found by (3.128) on page 130.

The resulting formula is shown below

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\,u(\tau)\,d\tau \tag{4.4}$$

where the symbol "$e^{\mathbf{A}t}$" stands for the matrix exponential function defined in Chapter 3.

If $t_0 = kT$ and $t = kT + T$, where T is the sampling period, then (4.4) gives an update formula for the state vector at sampling instants. That is, integrating the state equations over one sample period yields:

$$\mathbf{x}(kT+T) = e^{\mathbf{A}(kT+T-kT)}\mathbf{x}(kT) + \int_{kT}^{kT+T} e^{\mathbf{A}(kT+T-\tau)}\mathbf{B}\,u(\tau)\,d\tau. \tag{4.5}$$

Now recall from (4.1) that in the interval of integration, the function $u(t)$ is equal to $u[k]$, a constant. This constant can be taken outside of the integral as follows

$$\mathbf{x}(kT+T) = e^{\mathbf{A}T}\mathbf{x}(kT) + \left[\int_{kT}^{kT+T} e^{\mathbf{A}(kT+T-\tau)}\mathbf{B}\,d\tau\right] u[k]. \tag{4.6}$$

The above formula is only computing the value of the state vector $\mathbf{x}(t)$ at sampling instants $t = kT$. Thus if we define a discrete-time state vector sequence by

$$\mathbf{x}[k] = \mathbf{x}(kT) \tag{4.7}$$

and we let the matrix $e^{\mathbf{A}T}$ be called $\mathbf{\Phi}$ and the vector in the brackets be called $\mathbf{\Gamma}$, then (4.6) becomes the following discrete-time state-space equation:

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k]. \tag{4.8}$$

Note that $\mathbf{\Gamma}$ in the above equation is a constant vector, but from its definition in (4.6), it looks as though $\mathbf{\Gamma}$ might depend on the discrete-time index k. It will be shown shortly that $\mathbf{\Gamma}$ is in fact a constant vector. Also, using the output equation of (4.3) we can write

$$y[k] = \mathbf{C}\mathbf{x}[k] + Du[k]. \tag{4.9}$$

Equations (4.8) and (4.9) are a discrete-time system whose output, by construction, *exactly* matches the output of the analog system if its input is piecewise constant. That is, these two equations describe the ZOH equivalent for $G(s)$. Note that if $G(s)$ is a linear time-invariant system, then its ZOH equivalent, which we will denote by $G_d(z)$ will also be linear and time-invariant. The relationship between $G(s)$ and $G_d(z)$ is shown in Fig. 4.3, which is just a different way of showing Fig. 4.2. There are two crucial points to be made about Fig. 4.3:

1. When the input to a plant is piecewise constant, its ZOH equivalent is an *exact* model of the plant at sampling instants.

2. In a digital control system, the input to the plant is *always* piecewise constant, since it is the output of a D-A converter (i.e. a ZOH).

There is one mathematical detail that must be considered to finish this treatment of ZOH equivalent systems. That is to show that $\mathbf{\Gamma}$ is a constant vector. Recall that $\mathbf{\Gamma}$ is defined as

$$\mathbf{\Gamma} = \int_{kT}^{kT+T} e^{\mathbf{A}(kT+T-\tau)}\mathbf{B}\,d\tau. \tag{4.10}$$

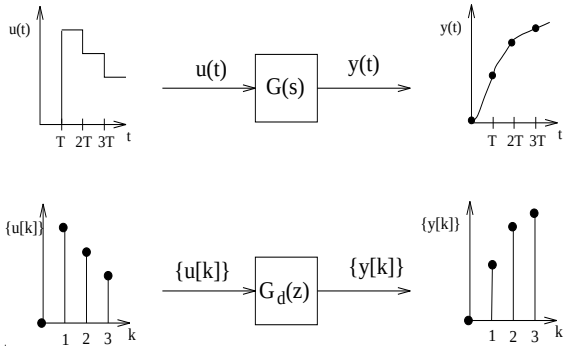


Figure 4.3 Relationship between $G(s)$ and its ZOH equivalent $G_d(z)$

Consider now the change of variables $\gamma = kT + T - \tau$. The new limits of integration are calculated as follows: when $\tau = kT$, $\gamma = T$, and when $\tau = kT + T$, $\gamma = 0$. In addition, the differentials are related by $d\tau = -d\gamma$, and so the upper and lower limits of integration can be interchanged to take care of the negative sign. With these substitutions, (4.10) can be rewritten as

$$\mathbf{\Gamma} = \int_0^T e^{\mathbf{A}\gamma}\mathbf{B}\, d\gamma \tag{4.11}$$

which is a constant vector, independent of k.

To summarize the above development, given a state-space description of a plant $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D)$ and a sampling interval T, the ZOH equivalent discrete-time system is given by $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C}, D)$, where

$$\mathbf{\Phi} = e^{\mathbf{A}T}\ , \quad \mathbf{\Gamma} = \int_0^T e^{\mathbf{A}\tau}\mathbf{B} d\tau. \tag{4.12}$$

The matrix exponential can be computed numerically for a fixed value of T [37], or it can be computed analytically as a function of T using a procedure described in section 4.2.1. It would appear that the vector $\mathbf{\Gamma}$ must be found by integration as shown in (4.12). However, it is interesting to note that both $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ can be computed simultaneously using a single matrix exponential. The result, due to Van Loan, is as follows: define the matrix $\mathbf{M}$ as

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & 0 \end{bmatrix}. \tag{4.13}$$

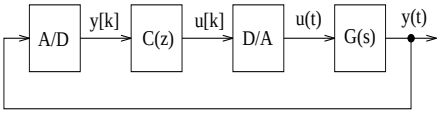
Note that $\mathbf{M}$ is a partitioned matrix in which the zero in the lower left hand corner represents a row-vector of n zeros. Then it can be shown that [89]

$$e^{\mathbf{M}T} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{4.14}$$

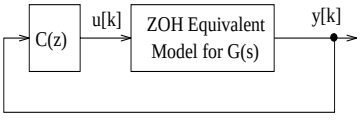
That is, $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ can be computed by finding the matrix exponential of $\mathbf{M}T$ and partitioning the result. Example 4.3 on page 171 illustrates the use of this formula. Problem 10 at the end of this chapter shows how to derive this formula.

4.1.2 The ZOH Equivalent as a Design Model

Before continuing the discussion of computing ZOH equivalent models, we digress in this section to consider how the ZOH model will be used for digital control system design. Recall that the plant to be controlled operates in continuous time. However, a digital compensator only acquires the value of the output of the plant at sampling instants, and can only change the value of the control signal it is producing at those same sampling instants. Since the *compensator* operates in discrete time, it makes sense to design the compensator so that the resulting control system has acceptable behavior at sampling instants. In other words, using the ZOH design model, we have a discrete-time design problem to specify the compensator $C(z)$. This compensator is designed so that the output sequence $\{y[k]\}$ of the ZOH model is acceptable.



(a)



(b)

Figure 4.4 (a) A digital control system with an analog plant. (b) A digital control system in which the analog plant has been replaced by its ZOH equivalent. By construction, $u[k] = u(kT)$ and $y[k] = y(kT)$.

Once $C(z)$ has been designed, it is actually used to control a continuous-time system as shown in Fig. 4.4(a). The output of the plant is $y(t)$, a continuous-time signal. From the properties of ZOH equivalence, we are guaranteed that the signal $y(t)$ will pass through the points $y[k]$ at sampling instants; that is, $y(kT) = y[k]$. Thus by designing $C(z)$ so that the sequence $\{y[k]\}$ is acceptable, we are guaranteed that the output of the plant will be acceptable *at sampling instants*. However, the use of the ZOH model for control system design means that the behavior in between sampling instants cannot be directly specified. The behavior between sampling instants can be examined by simulating the digital control system in Fig. 4.4(a). This behavior can also be analyzed using the modified z-transform [44] or state-space techniques [45, 43].

4.2 Matrix Functions

For systems of low order, it is possible to compute the matrix exponential analytically for an arbitrary sampling interval T. The advantage of an analytical computation is that the result is expressed in terms of the sampling interval T and possibly other parameters, and it is possible to examine the effect of changing these parameters analytically. When numerical techniques are used with a fixed value of T, the exponential is computed as a matrix of numbers. In this section we consider two different methods for computing

an arbitrary matrix function. The resulting formulas can be used to calculate the matrix exponential with arbitrary T, as well as a variety of other matrix functions.

4.2.1 Matrix Functions via Cayley-Hamilton

Recall that the matrix exponential is defined from the power series expansion of the scalar exponential function. Consider an arbitrary function $f(\tau)$ which is defined by a power series expansion

$$f(\tau) = \sum_{j=0}^{\infty} c_j \, \tau^j \tag{4.15}$$

where c_j are the expansion coefficients. The development and results in this section are also valid for functions defined by the power series

$$f(\tau) = \sum_{j=0}^{\infty} c'_j \, (\tau - \alpha)^j \tag{4.16}$$

for some real number α. For simplicity we only give details for (4.15). Many functions can be defined by a power series using an appropriate choice of expansion coefficients. For example, $\sin x$, $\cos x$, and e^x all have power series expansions of the form shown in (4.15), and $\log x$ has an expansion of the form shown in (4.16). In general, such series converge for certain values of τ, $\tau \in R$, where R is called the *region of convergence* of the power series. For the real-valued exponential function, $R = (-\infty, \infty)$.

When a scalar function is defined by a power series, it is always possible to define a corresponding matrix function by a matrix power series with the same expansion coefficients. For example, if $\mathbf{A}$ is an $n \times n$ matrix, then the function $f(\mathbf{A})$ is defined as

$$f(\mathbf{A}) = \sum_{j=0}^{\infty} c_j \, \mathbf{A}^j. \tag{4.17}$$

This power series is well-defined (convergent) if all the eigenvalues of $\mathbf{A}$ are elements of the region of convergence of the scalar power series for $f(\tau)$.

We now make use of the Cayley-Hamilton theorem from linear algebra to show how matrix functions defined by power series can be computed analytically. Recall that the characteristic polynomial of an $n \times n$ matrix $\mathbf{A}$ is defined as

$$a(\lambda) = \det(\lambda \mathbf{I} - \mathbf{A}) = \lambda^n + a_1 \lambda^{n-1} + \cdots + a_n. \tag{4.18}$$

The theorem is as follows

Cayley-Hamilton Theorem

An $n \times n$ matrix $\mathbf{A}$ satisfies $a(\mathbf{A}) = \mathbf{0}$, or

$$\mathbf{A}^n + a_1 \mathbf{A}^{n-1} + \cdots + a_n \mathbf{I} = \mathbf{0}$$

where $a(\lambda)$ is the characteristic polynomial of $\mathbf{A}$.

In other words, the matrix $\mathbf{A}^n$ can be expressed as a linear combination of lower powers of $\mathbf{A}$, where the coefficients of the combination are the negatives of the coefficients of the characteristic polynomial of $\mathbf{A}$.

If we multiply the equation shown in the Cayley-Hamilton Theorem by $\mathbf{A}$, we see that $\mathbf{A}^{n+1}$ can be expressed as a linear combination of the matrices $\mathbf{A}^n$, $\mathbf{A}^{n-1}$, $\cdots$, $\mathbf{A}$. But since $\mathbf{A}^n$ can be expressed as a linear combination of lower powers of $\mathbf{A}$, it is clear that $\mathbf{A}^{n+1}$ can be written as a linear combination of $\mathbf{A}^{n-1}, \cdots, \mathbf{I}$. Proceeding inductively, it is clear that the following is true

Corollary to Cayley-Hamilton

Given any $n \times n$ matrix $\mathbf{A}$, $\mathbf{A}^k$, $k \geq n$ can be written as a linear combination of $\mathbf{A}^{n-1}, \mathbf{A}^{n-2}, \cdots, \mathbf{I}$.

Suppose we now take a linear combination of matrices $\mathbf{A}^k$ for several different values of k. Since *each* matrix $\mathbf{A}^k$ can be expressed in terms of $\mathbf{A}^{n-1}, \cdots, \mathbf{I}$, then a linear combination of $\mathbf{A}^k$

$$\sum_{j=0}^{k} c_j \mathbf{A}^j \tag{4.19}$$

can also be expressed in terms of $\mathbf{A}^{n-1}, \cdots, \mathbf{I}$. This is true even if we let k range to infinity provided that the sum in (4.19) exists as k goes to infinity. Thus we have a remarkable fact about matrices that a convergent infinite power series can always be represented by a finite series

$$f(\mathbf{A}) = \sum_{j=0}^{\infty} c_j \mathbf{A}^j = \alpha_1 \mathbf{A}^{n-1} + \alpha_2 \mathbf{A}^{n-2} + \cdots + \alpha_n \mathbf{I}. \tag{4.20}$$

The coefficients c_j in the above equation come from the power series expansion of $f(\cdot)$, which could be the exponential function, for example. From the Cayley-Hamilton theorem, we know that there exists a finite number of coefficients $\alpha_1, \cdots, \alpha_n$ which define a finite power series equivalent to the infinite power series. We now show how to calculate this finite set of coefficients, and thus evaluate the matrix function $f(\mathbf{A})$ as a *finite* matrix polynomial.

We start with the defining equation for eigenvalues and eigenvectors of the matrix $\mathbf{A}$

$$\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i \ , \quad i = 1, \cdots, n \tag{4.21}$$

where $\mathbf{x}_i$ is an eigenvector of $\mathbf{A}$ corresponding to eigenvalue λ_i. If we multiply the above equation by $\mathbf{A}$, we obtain

$$\begin{aligned} \mathbf{A}(\mathbf{A}\mathbf{x}_i) &= \mathbf{A}(\lambda_i \mathbf{x}_i) \\ &= \lambda_i \mathbf{A}\mathbf{x}_i \\ &= \lambda_i^2 \mathbf{x}_i. \end{aligned} \tag{4.22}$$

If we continue to multiply the above equation by $\mathbf{A}$, we obtain the following result

$$\mathbf{A}^k \mathbf{x}_i = \lambda_i^k \mathbf{x}_i \ , \quad i = 1, \cdots, n \ , \quad k \geq 0. \tag{4.23}$$

We now multiply the matrix power series (4.17) by an eigenvector $\mathbf{x}_i$ of the matrix $\mathbf{A}$, and use the result of the previous equation to express powers of $\mathbf{A}$ times the eigenvector as

powers of the eigenvalue times the eigenvector

$$\begin{aligned} f(\mathbf{A})\mathbf{x}_i &= \sum_{j=0}^{\infty} c_j \mathbf{A}^j \mathbf{x}_i \\ &= \sum_{j=0}^{\infty} c_j \lambda_i^j \, \mathbf{x}_i \\ &= f(\lambda_i)\mathbf{x}_i \, , \quad i = 1, \cdots, n. \end{aligned} \tag{4.24}$$

Notice in the last line of the above equation, the infinite power series is applied to the number λ_i yielding the scalar function value $f(\lambda_i)$ as defined in (4.15). This implies that the function $f(\cdot)$ must exist at the eigenvalues of $\mathbf{A}$ in order to evaluate the matrix function $f(\mathbf{A})$. The power series in the matrix $\mathbf{A}$ became a power series in the scalar λ_i because of the multiplication by the eigenvector $\mathbf{x}_i$.

The trick of converting a matrix series to a scalar series can also be used on the finite polynomial whose coefficients we are trying to calculate. That is, if we multiply (4.20) by $\mathbf{x}_i$, we obtain

$$\begin{aligned} f(\mathbf{A})\mathbf{x}_i &= \alpha_1 \mathbf{A}^{n-1}\mathbf{x}_i + \alpha_2 \mathbf{A}^{n-2}\mathbf{x}_i + \cdots \alpha_n \mathbf{I}\mathbf{x}_i \\ &= \alpha_1 \lambda_i^{n-1}\mathbf{x}_i + \alpha_2 \lambda_i^{n-2}\mathbf{x}_i + \cdots + \alpha_n \mathbf{x}_i \\ &= \alpha(\lambda_i)\mathbf{x}_i \, , \quad i = 1, \cdots, n, \end{aligned} \tag{4.25}$$

where $\alpha(\lambda_i)$ is the polynomial

$$\alpha(\lambda) \stackrel{\text{def}}{=} \alpha_1 \lambda^{n-1} + \cdots + \alpha_n \tag{4.26}$$

evaluated at $\lambda = \lambda_i$. Equating the last lines of (4.24) and (4.25), we obtain

$$f(\lambda_i)x_i = \alpha(\lambda_i)\mathbf{x}_i \, , \quad i = 1, \cdots, n, \tag{4.27}$$

or since $\mathbf{x}_i$ is nonzero (it is an eigenvector),

$$f(\lambda_i) = \alpha(\lambda_i) \, , \quad i = 1, \cdots, n. \tag{4.28}$$

In the above equation, $f(\cdot)$ is a known function (such as the exponential function), and so the left hand side of the equation is a number. The right hand side of the equation contains the unknown polynomial coefficients $\alpha_1, \cdots, \alpha_n$. Since there are n equations, one for each eigenvalue, we can form a system of n equations in n unknowns and solve for the unknown coefficients. If the eigenvalues λ_i are all distinct, then (4.28) will yield n independent equations which can be written in matrix form as shown below.

$$\begin{bmatrix} f(\lambda_1)) \\ f(\lambda_2) \\ \vdots \\ f(\lambda_n) \end{bmatrix} = \begin{bmatrix} \lambda_1^{n-1} & \lambda_1^{n-2} & \cdots & 1 \\ \lambda_2^{n-1} & \lambda_2^{n-2} & \cdots & 1 \\ \vdots & \vdots & \cdots & \vdots \\ \lambda_n^{n-1} & \lambda_n^{n-2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}. \tag{4.29}$$

If one of the eigenvalues (say λ_1) is complex, then one of the other eigenvalues (say λ_2) must be the complex conjugate of λ_1. In order to avoid forming complex linear equations,

the equations for $f(\lambda_1)$ and $f(\lambda_2)$ in (4.29) can be replaced by the following equations involving real and imaginary parts:

$$\begin{bmatrix} \text{real}(f(\lambda_1)) \\ \text{imag}(f(\lambda_1)) \end{bmatrix} = \begin{bmatrix} \text{real}(\lambda_1^{n-1}) & \text{real}(\lambda_1^{n-2}) & \cdots & 1 \\ \text{imag}(\lambda_1^{n-1}) & \text{imag}(\lambda_1^{n-2}) & \cdots & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}. \tag{4.30}$$

If there is a repeated eigenvalue (which corresponds to a multiple pole in a systems context), then the equations in (4.28) will not be independent. In fact, the equations corresponding to the repeated eigenvalue will be identical to each other, since the same value of λ would be used in each equation. In the case where an eigenvalue λ has multiplicity m, it can be shown that the required m equations can be generated by differentiation with respect to λ as follows

$$\begin{aligned} f(\lambda_j) &= \alpha(\lambda_j) \\ f^{(1)}(\lambda_j) &= \alpha^{(1)}(\lambda_j) \\ \vdots &= \vdots \\ f^{(m-1)}(\lambda_j) &= \alpha^{(m-1)}(\lambda_j). \end{aligned} \tag{4.31}$$

EXAMPLE 4.1

The type-1 servo system introduced in Chapter 3 has the following state-space description

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -p \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

where p is a pole of the system (the other pole is at $s = 0$). We will use the procedure described in this section to compute the ZOH equivalent $\mathbf{\Phi}$ and $\mathbf{\Gamma}$. Since $\mathbf{A}$ is an upper triangular matrix, its eigenvalues are the diagonal elements so $\lambda_1 = 0$ and $\lambda_2 = -p$. The system is second order ($n = 2$) and so the polynomial $\alpha(\lambda)$ has degree $n - 1 = 1$

$$\alpha(\lambda) = \alpha_1 \lambda + \alpha_2.$$

The scalar function of interest here is the exponential function

$$f(\lambda) = e^{\lambda T}$$

and from (4.20), the corresponding matrix function is given by the first-order matrix polynomial

$$f(\mathbf{A}) = e^{\mathbf{A}T} = \alpha_1 \mathbf{A} + \alpha_2 I.$$

The coefficients α_1 and α_2 are computed using (4.28) which yields

$$\begin{aligned} e^{0T} &= \alpha_1 \cdot 0 + \alpha_2 \\ e^{-pT} &= \alpha_1(-p) + \alpha_2. \end{aligned}$$

The first of the above equations gives $\alpha_2 = 1$ and the second equation minus the first equation gives $-p\alpha_1 = e^{-pT} - 1$ or $\alpha_1 = (1 - e^{-pT})/p$. Note that the above equations could be written in the form of (4.29) as follows:

$$\begin{bmatrix} e^{0T} \\ e^{-pT} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -p & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}.$$

The coefficients α_1 and α_2 are then used in the first-order matrix polynomial

$$e^{\mathbf{A}T} = \alpha_1 \begin{bmatrix} 0 & 1 \\ 0 & -p \end{bmatrix} + \alpha_2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha_2 & \alpha_1 \\ 0 & \alpha_2 - p\alpha_1 \end{bmatrix}.$$

When the values of α_1 and α_2 are substituted in, the final result is

$$\mathbf{\Phi} = e^{\mathbf{A}T} = \begin{bmatrix} 1 & (1 - e^{-pT})/p \\ 0 & e^{-pT} \end{bmatrix}.$$

The vector $\mathbf{\Gamma}$ is computed using (4.12) which is the integral of $e^{\mathbf{A}\tau}\mathbf{b}$. We first form this product

$$e^{\mathbf{A}\tau}\mathbf{b} = \begin{bmatrix} 1 & (1 - e^{-p\tau})/p \\ 0 & e^{-p\tau} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} (1 - e^{-p\tau})/p \\ e^{-p\tau} \end{bmatrix}$$

and then integrate

$$\mathbf{\Gamma} = \int_0^T \begin{bmatrix} (1 - e^{-p\tau})/p \\ e^{-p\tau} \end{bmatrix} d\tau = \begin{bmatrix} T/p - (1 - e^{-pT})/p^2 \\ (1 - e^{-pT})/p \end{bmatrix}.$$

Notice that the formulas for $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ derived above can be evaluated for any value of the sampling interval T to get a particular ZOH model. In this example, $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ were computed separately. It is left as an exercise to show that the same answers could be obtained using (4.13) and (4.14).

■

EXAMPLE 4.2

The harmonic oscillator system introduced in Chapter 3 has the following state-space model:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ \omega^2 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(t).$$

The eigenvalues of $\mathbf{A}$ are $\pm j\omega$. In order to compute $e^{\mathbf{A}t}$, the scalar function of interest is

$$f(\lambda) = e^{\lambda t}$$

and the corresponding matrix function can be written as a first-order matrix polynomial

$$f(\mathbf{A}) = e^{\mathbf{A}t} = \alpha_1 \mathbf{A} + \alpha_2 \mathbf{I}.$$

Since the eigenvalues of $\mathbf{A}$ are complex, the coefficients α_1 and α_2 are computed using (4.30) which, for this example , is:

$$\begin{bmatrix} \cos \omega t \\ \sin \omega t \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \omega & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}.$$

The solution is $\alpha_2 = \cos \omega t$, $\alpha_1 = (\sin \omega t)/\omega$. Then

$$e^{\mathbf{A}t} = \begin{bmatrix} \alpha_2 & \alpha_1 \\ \omega^2 \alpha_1 & \alpha_2 \end{bmatrix} = \begin{bmatrix} \cos \omega t & \frac{\sin \omega t}{\omega} \\ -\omega^2 \sin \omega t & \cos \omega t \end{bmatrix}.$$

■

EXAMPLE 4.3

In this example, we consider the double integrator system introduced in Chapter 3. We will use (4.13) and (4.14) to compute $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ simultaneously. The matrix $\mathbf{C}$ defined in (4.13) is

$$\mathbf{C} = \left[\begin{array}{cc|c} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 0 & 0 & 0 \end{array}\right].$$

This matrix has an eigenvalue $\lambda = 0$ with multiplicity $m = 3$. Since $\mathbf{C}$ is a 3×3 matrix, the polynomial $\alpha(\lambda)$ will be of degree 2, and the matrix exponential can be written as a matrix polynomial

$$e^{\mathbf{C}T} = \alpha_1 \mathbf{C}^2 + \alpha_2 \mathbf{C} + \alpha_3 \mathbf{I}. \tag{4.32}$$

The coefficients of this polynomial can be found using (4.31) with $\lambda_j = 0$ and $m = 3$. Equation (4.31) requires the function $f(\lambda)$ and the polynomial $\alpha(\lambda)$, as well as their first and second derivatives with respect to λ. These are shown below:

$$f(\lambda) = e^{\lambda T} \qquad \alpha(\lambda) = \alpha_1 \lambda^2 + \alpha_2 \lambda + \alpha_3$$

$$f^{(1)}(\lambda) = T e^{\lambda T} \qquad \alpha^{(1)}(\lambda) = 2\alpha_1 \lambda + \alpha_2$$

$$f^{(2)}(\lambda) = T^2 e^{\lambda T} \qquad \alpha^{(2)}(\lambda) = 2\alpha_1.$$

When the above equations are evaluated at the eigenvalue $\lambda = 0$, the result is

$$\begin{aligned} \alpha_3 &= 1 \\ \alpha_2 &= T \\ \alpha_1 &= T^2/2. \end{aligned}$$

These coefficients are then used in (4.32) to find the matrix exponential

$$e^{\mathbf{C}T} = \alpha_1 \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \alpha_3 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

which results in

$$e^{\mathbf{C}T} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}.$$

Finally, $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ are extracted from the matrix C according to the partitions shown in (4.14)

$$\mathbf{\Phi} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix}.$$

■

4.2.2 Matrix Functions via Eigenvalue Decomposition

The method for computing matrix functions $f(\mathbf{A})$ given in the previous subsection will work even if the matrix $\mathbf{A}$ has repeated eigenvalues. In this subsection we present another method for computing matrix functions. This method is simple to use but it applies in general only to matrices with distinct eigenvalues.

From Chapter 2 we know that the eigenvectors of a matrix with distinct eigenvectors are linearly independent (Fact 2.35). If $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues of $\mathbf{A}$ as its diagonal elements and if $\mathbf{X}$ is a matrix whose columns are the corresponding eigenvectors of $\mathbf{A}$ then (see (2.30) in Chapter 2)

$$\mathbf{AX} = \mathbf{X\Lambda}$$

or

$$\mathbf{A} = \mathbf{X\Lambda X}^{-1}. \tag{4.33}$$

For future reference, recall the following notation for a diagonal matrix

$$\text{diag}(d_1, d_2, \cdots, d_n) = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix}.$$

The decomposition of $\mathbf{A}$ in (4.33) can be used to compute the matrix function $f(\mathbf{A})$ where the scalar function $f(\tau)$ is defined by the power series

$$f(\tau) = \sum_{k=0}^{\infty} c_k \tau^k.$$

By writing out the first few terms of the corresponding matrix power series it is easy to show that if $\mathbf{D}$ is a diagonal matrix $\mathbf{D} = \text{diag}(d_1, \cdots, d_n)$ then the matrix function $f(\mathbf{D})$ is also a diagonal matrix given by

$$f(\mathbf{D}) = \text{diag}(f(d_1), \cdots, f(d_n)).$$

Using this result and the power series definition of $f(\mathbf{A})$, it is also easy to show that

$$\begin{aligned} f(\mathbf{A}) &= \mathbf{X} f(\mathbf{D}) \mathbf{X}^{-1} \\ &= \mathbf{X}\ \text{diag}(f(\lambda_1), \cdots, f(\lambda_n)) \mathbf{X}^{-1}. \end{aligned} \tag{4.34}$$

This equation shows that the matrix function $f(\mathbf{A})$ can be evaluated by first computing the scalar function $f(\cdot)$ at the n eigenvalues of $\mathbf{A}$ and then multiplying by the eigenvector matrices in (4.34).

As an example of the above results, suppose we want to compute $e^{\mathbf{A}T}$ where T is an unknown sampling period and $\mathbf{A}$ is a matrix with distinct eigenvalues. Using (4.34) we have

$$e^{\mathbf{A}T} = \mathbf{X}\ \text{diag}(e^{\lambda_1 T}, \cdots, e^{\lambda_n T}) \mathbf{X}^{-1}. \tag{4.35}$$

It would be useful to express $e^{\mathbf{A}T}$ as a weighted sum of scalar exponentials as follows

$$e^{\mathbf{A}T} = \mathbf{M}_1 e^{\lambda_1 T} + \cdots + \mathbf{M}_n e^{\lambda_n T}. \tag{4.36}$$

This can be done by partitioning $\mathbf{X}$ into columns and $\mathbf{X}^{-1}$ into rows as follows

$$\mathbf{X} = [\,\mathbf{x}_1 \quad \cdots \quad \mathbf{x}_n\,]\ ,\ \mathbf{X}^{-1} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_n^T \end{bmatrix}. \tag{4.37}$$

By applying the rules of matrix multiplication to the right-hand side of (4.35), it can be shown that the matrices $\mathbf{M}_i$ in (4.36) are given by

$$\mathbf{M}_i = \mathbf{x}_i \mathbf{y}_i^T. \tag{4.38}$$

EXAMPLE 4.4

In this example we use the eigenvalue decomposition to compute $e^{\mathbf{A}T}$ where T is arbitrary and

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix}.$$

The eigenvalues and eigenvectors of $\mathbf{A}$ are shown in the following matrices:

$$\mathbf{\Lambda} = \begin{bmatrix} 0 & 0 \\ 0 & -2 \end{bmatrix}\ ,\ \mathbf{X} = \begin{bmatrix} 1 & -.4472 \\ 0 & .8944 \end{bmatrix}.$$

Using (4.35) we have

$$\begin{aligned} e^{\mathbf{A}T} &= \mathbf{X} \begin{bmatrix} e^{0T} & 0 \\ 0 & e^{-2T} \end{bmatrix} \mathbf{X}^{-1} \\ &= \begin{bmatrix} 1 & -.4472 \\ 0 & .8944 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{-2T} \end{bmatrix} \begin{bmatrix} 1 & 0.5000 \\ 0 & 1.1180 \end{bmatrix} \end{aligned}$$

and using (4.37) and (4.38) we get

$$\begin{aligned} e^{\mathbf{A}T} &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} [\,1 \quad 0.5\,] \cdot 1 + \begin{bmatrix} -.4472 \\ .8944 \end{bmatrix} [\,0 \quad 1.1180\,]\, e^{-2T} \\ &= \begin{bmatrix} 1 & .5000 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -.5 \\ 0 & 1 \end{bmatrix} e^{-2T} \\ &= \begin{bmatrix} 1 & \frac{1}{2}(1 - e^{-2T}) \\ 0 & e^{-2T} \end{bmatrix}. \end{aligned}$$

This result checks with Example 4.1.

■

4.3 Discretizing a System with Time Delay

As mentioned in Chapter 3, there are many industrial processes that can be modeled by a linear system with a time delay. In addition, the use of digital control always results in a time delay in the control loop. The reason is that it takes a certain amount of time for the compensator to compute its output value given a new input value. This computational

delay can be included in the description of the plant for the purpose of design. If the computational delay is very small, it probably does not have to be included in the design model. However, the results of this section can be used to assess the effect of time delay on the closed-loop system, and to decide whether or not it is negligible.

A system with time delay has a transfer function of the following form (see Table 3.1 on page 83 and Section 3.6.5)

$$G(s) = e^{-sD} G_1(s). \tag{4.39}$$

If $G_1(s)$ has a state-space description $(\mathbf{A}, \mathbf{B}, \mathbf{C})$, then $G(s)$ can be described by the following equations

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t-D) \\ y(t) &= \mathbf{C}\mathbf{x}(t). \end{aligned} \tag{4.40}$$

Note that the above equation is *not* a finite-dimensional state-space description for $G(s)$ because the input is delayed by D seconds. As we show in the following paragraphs, the state vector $\mathbf{x}(t_1)$ of a state-space model contains all the information necessary to compute the output for $t \geq t_1$. In other words, the state vector $x(t_1)$ summarizes the effect of past inputs ($t \leq t_1$) on future outputs ($t \geq t_1$). However, a delay in the input means that the vector $\mathbf{x}(t_1)$ does *not* summarize the effect of past inputs.

Consider first the system *without* time delay. If the value of the state vector is known at time t_1, and if the input equals zero for $t \geq t_1$, then the output of the system can be calculated as follows:

$$\begin{aligned} &\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t)\ , \ \ \mathbf{x}(t_1) \text{ given} \\ &\mathbf{x}(t) = e^{\mathbf{A}(t-t_1)}\mathbf{x}(t_1)\ , \ \ t \geq t_1 \\ &y(t) = \mathbf{C}e^{\mathbf{A}(t-t_1)}\mathbf{x}(t)\ , \ \ t \geq t_1. \end{aligned} \tag{4.41}$$

In the second line of the above equations, the state vector for all time greater than t_1 can be computed in terms of the state vector at time t_1. In other words, the n state variables $x_1(t_1), \cdots, x_n(t_1)$ contain all the information about the future output of the system in response to inputs up to time t_1.

Now consider the system with time delay. If the input is zero for $t \geq t_1$, then the output of the system must be computed from

$$\begin{aligned} &\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t-D)\ , \ \ \mathbf{x}(t_1) \text{ given}\ , \ \ u(t) \text{ given for } t_1 \leq t \leq t_1 + D \\ &\mathbf{x}(t) = e^{\mathbf{A}(t-(t_1+D))}\mathbf{x}(t_1+D)\ , \ \ t \geq t_1 + D \\ &y(t) = \mathbf{c}\mathbf{x}(t)\ , \ \ t \geq t_1. \end{aligned} \tag{4.42}$$

So $\mathbf{x}(t_1)$ does not contain all the information about the system even if the input is set to zero at time t_1. In addition to the n state variables $x_1(t_1), \cdots, x_n(t_1)$, the input function $u(t)$ has to be known over the interval $t_1 \leq t \leq t_1 + D$ in order to solve the first equation shown in (4.42) for $\mathbf{x}(t)$ in the interval $t_1 \leq t \leq t_1 + D$. The input signal over this interval contains a continuum of values in general, and so an *infinite* number of state variables are needed to summarize the information about the system at time t_1. Thus a system with time delay is *infinite dimensional*.

It is clear that infinite dimensional systems are much more difficult to deal with than finite dimensional systems. Nevertheless, we now show that it is possible to obtain a finite

dimensional ZOH equivalent model of a linear system with time delay. The ZOH model will *exactly* describe the output of such a system in response to a piecewise-constant input. The intuitive reason why it is not difficult to handle systems with time delay when the input is piecewise-constant is that such an input signal can be represented by a *finite* number of values over the time-delay interval. This will be shown in the following development.

We begin with the vector differential equation model of a system with time delay

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t-D) \\ y(t) &= \mathbf{C}\mathbf{x}(t). \end{aligned} \tag{4.43}$$

Assume this system is driven by an input which is piecewise constant over some sampling interval T; that is,

$$u(t) = u[k]\,, \quad kT \le t < kT + T. \tag{4.44}$$

If we integrate the differential equation over one sample period, we obtain in a similar way as before

$$\mathbf{x}(kT+T) = e^{\mathbf{A}T}\mathbf{x}(kT) + \int_{kT}^{kT+T} e^{\mathbf{A}(kT+T-\tau)}\mathbf{B}\,u(\tau - D)\,d\tau. \tag{4.45}$$

In the previous derivation without time delay, we made use of the fact that the input was constant over the region of integration, and so we could pull it out of the integral. In the present case with time delay, it is not necessarily true that the delayed input is constant over the interval kT to $kT+T$. (It is only true in the special case in which the time delay is an integer multiple of the sampling interval.) This is most easily seen be looking at Fig. 4.5.

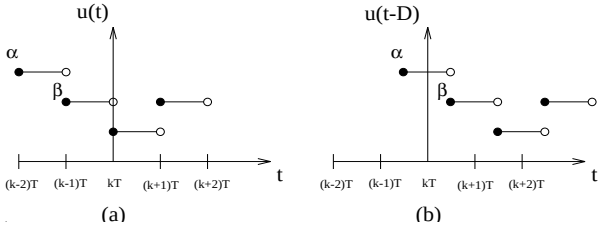


Figure 4.5 (a) A piecewise-constant signal $u(t)$. (b) The delayed signal $u(t-D)$. In this figure, $D = (1+\frac{1}{2})T$. From (a), we see that α has the value $u(kT-2T)$ and β has the value $u(kT-T)$.

We see that if the time delay D is not an integer multiple of the sampling period T, then the delayed input signal will take two different values in the interval kT to $kT+T$. Thus to perform the integration in (4.45), we can split the integral into two parts, where the delayed input signal is constant over each part. The input can then be taken outside of each integral. Fig. 4.5 shows the situation when $D = T + \frac{1}{2}T$. In general, the time delay will equal some integer number (possibly zero) of sampling intervals plus a fractional part of a sampling interval. That is, we can represent any delay D as $D = qT + \gamma$ where $q \ge 0$ and $0 < \gamma \le T$. In Fig. 4.5 we have $q = 1$ and $\gamma = \frac{1}{2}T$. Note that when D is equal to an integer multiple of the sampling period, say $D = mT$, we let $\gamma = T$ and $q = m - 1$. We could also let $q = m$ and $\gamma = 0$, but this is not advised as will be explained shortly.

In order to perform the integration in (4.45), we need to know the value of $u(t-D)$ over the interval kT to $kT+T$. In the general case, the result is

$$u(t-D) = \begin{cases} u(kT-(q+1)T)\,, & kT \le t < kT+\gamma \\ u(kT-qT)\,, & kT+\gamma \le t < kT+T. \end{cases} \tag{4.46}$$

The reader can verify that this expression is correct for the example shown in Fig. 4.5. We can now substitute the above expression for the delayed input into (4.45) and split the integral into two parts

$$\begin{aligned} \mathbf{x}(kT+T) = e^{\mathbf{A}T}\mathbf{x}(kT) \quad &+\int_{kT}^{kT+\tau} e^{\mathbf{A}(kT+T-\tau)}\mathbf{B}\,d\tau\; u(kT-qT-T) \\ &+\int_{kT+\tau}^{kT+T} e^{\mathbf{A}(kT+T-\tau)}\mathbf{B}\,d\tau\; u(kT-qT). \end{aligned} \tag{4.47}$$

The integrals in the above equation, which we label $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_0$, respectively, turn out to be independent of the time index k. Making the substitution $\gamma = kT+T-\tau$ in the first integral yields

$$\mathbf{\Gamma}_1 = \int_{T-\gamma}^{T} e^{\mathbf{A}\tau}\mathbf{B}\,d\tau, \tag{4.48}$$

and making the same substitution in the second integral yields

$$\mathbf{\Gamma}_0 = \int_{0}^{T-\gamma} e^{\mathbf{A}\tau}\mathbf{B}d\tau. \tag{4.49}$$

Note that $\mathbf{\Gamma}_0+\mathbf{\Gamma}_1 = \mathbf{\Gamma}$ where $\mathbf{\Gamma}$ is the input vector for the ZOH model without time delay. Substituting $\mathbf{\Gamma}_0$ and $\mathbf{\Gamma}_1$ into (4.47) and changing to discrete-time notation yields

$$\mathbf{x}[k+1] = e^{\mathbf{A}T}\mathbf{x}[k] + \mathbf{\Gamma}_1 u[k-q-1] + \mathbf{\Gamma}_0 u[k-q]. \tag{4.50}$$

We assume temporarily that $q \ge 1$. The case when $q=0$ will be handled separately.

Equation (4.50) describes how the variables in the vector $\mathbf{x}(t)$ of the system with time delay behave at sampling instants. It is an *exact* equation – no approximations were made – but it is *not* in the form of a state-update equation for a discrete-time system. Recall that the right hand side of a state-update equation must consist of a linear combination of state variables at time k and the input at time k. In (4.50), the input term does not appear at time k. We can remedy this situation by *defining* two new state variables as follows

$$\begin{aligned} z_1[k] &= u[k-q-1] \\ z_2[k] &= u[k-q]. \end{aligned} \tag{4.51}$$

If these new state variables are substituted into (4.50), then the right hand side of (4.50) consists of a linear combination of state variables at time k as desired. However, if $z_1[k]$ and $z_2[k]$ are state variables, then we need to write state-update equations for them. Using (4.51) we have

$$\begin{aligned} z_1[k+1] &= u[k-q] = z_2[k] \\ z_2[k+1] &= u[k-q+1]. \end{aligned} \tag{4.52}$$

The first equation above is a valid state-update equation since the right hand side consists of the state variable z_2 at time k. The second equation above is a valid state update equation only if $q = 1$ so that the input term appears at time k. If $q > 1$, however, we can define $z_3[k] = u[k-q]$ and proceed as before. In general, we define as many new state variables as are needed so that the right hand side of the state-update equation for the last state variable consists of the input at time k.

To summarize the above discussion, equation (4.50) can be written as a valid state-update equation if additional state variables are used to store past values of the input. It turns out that exactly $q+1$ additional state variables are needed

$$\begin{aligned} z_1[k] &= u[k-q-1] \\ z_2[k] &= u[k-q] \\ \vdots &= \vdots \\ z_{q+1}[k] &= u[k-1]. \end{aligned} \tag{4.53}$$

Since these state variables are consecutive shifts of the input sequence, it is easy to write update equations for them. The update equations are

$$\begin{aligned} z_1[k+1] &= u[k-q] &&= z_2[k] \\ z_2[k+1] &= u[k-q+1] &&= z_3[k] \\ \vdots &= \vdots &&= \vdots \\ z_q[k+1] &= u[k-1] &&= z_{q+1}[k] \\ z_{q+1}[k+1] &= u[k]. \end{aligned} \tag{4.54}$$

Notice that the update equation for z_{q+1} is written in terms of the input at time k, which is a valid update equation. The update equation (4.50) for the vector $x[k]$ can now be written with the help of the additional state variables $z_1[k]$ and $z_2[k]$. Using (4.53) and (4.50) we can write

$$\mathbf{x}[k+1] = e^{\mathbf{A}T}\mathbf{x}[k] + \mathbf{\Gamma}_1 z_1[k] + \mathbf{\Gamma}_0 z_2[k]\ ,\ \ q \geq 1, \tag{4.55}$$

which is a valid state-update equation which does not depend explicitly on the input. Note that the above equation is only valid when $q \geq 1$ because we have assumed that $z_2[k]$ is defined as a state variable. If $q = 0$ so that the time delay is less than or equal to the sample period, then only one additional state variable is defined, and then (4.50) can be written as

$$\mathbf{x}[k+1] = e^{\mathbf{A}T}\mathbf{x}[k] + \mathbf{\Gamma}_1 z_1[k] + \mathbf{\Gamma}_0 u[k]\ ,\ \ q = 0. \tag{4.56}$$

A complete state-update equation for a system with time delay can now be obtained by combining (4.55) or (4.56) with (4.54) to obtain

$$\begin{bmatrix} \mathbf{x}[k+1] \\ z_1[k+1] \\ z_2[k+1] \\ \vdots \\ z_q[k+1] \\ z_{q+1}[k+1] \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma}_1 & \mathbf{\Gamma}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & 0 & 1 & 0 & \cdots & 0 \\ \mathbf{0} & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & 0 & 0 & 0 & \cdots & 1 \\ \mathbf{0} & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}[k] \\ z_1[k] \\ z_2[k] \\ \vdots \\ z_q[k] \\ z_{q+1}[k] \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u[k]\ ,\ \ q \geq 1, \tag{4.57}$$

or

$$\begin{bmatrix} \mathbf{x}[k+1] \\ z_1[k+1] \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma}_1 \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}[k] \\ z_1[k] \end{bmatrix} + \begin{bmatrix} \mathbf{\Gamma}_0 \\ 1 \end{bmatrix} u[k] \ , \quad q = 0. \tag{4.58}$$

Notice that the above equations contain partitioned matrices. For instance, each zero below the matrix $\mathbf{\Phi}$ represents a row vector of n zeros.

For any value of q, the output equation is obtained from (4.40) to be

$$y[k] = [\mathbf{C}\ 0\ \cdots\ 0] \begin{bmatrix} \mathbf{x}[k] \\ z_1[k] \\ \vdots \\ z_{q+1}[k] \end{bmatrix} . \tag{4.59}$$

Notice that the state vector contains $n + q + 1$ state variables. The vector $\mathbf{x}[k]$ contains n state variables corresponding to the system without time delay, and there are $q + 1$ additional state variables which are used to model the time delay. Recall that the time delay D is represented as $D = qT + \gamma$. Thus if the time delay is large compared to the sampling interval then q will be a large number, and many additional state variables will have to be used to store samples of the input sequence.

Let us now consider the special case when the time delay is an integer multiple of the sampling period; that is, $D = mT$. We have stated previously that in this case, one should choose $\gamma = T$ and $q = m - 1$. With this choice, $\mathbf{\Gamma}_0 = 0$ (see (4.49)) and m state variables have to be added to model the time delay. The alternative when $D = mT$ is to choose $q = m$ and $\gamma = 0$. In this case, $\mathbf{\Gamma}_1 = 0$ (see (4.48)) and $m + 1$ state variables have to be added to the model. But inspection of (4.57) shows that when $\mathbf{\Gamma}_1 = 0$, the state variable $z_1[k]$ is not used in updating the state vector, and hence this variable can be deleted from the equations. Deleting this variable leaves m additional state variables, and the equations reduce to those obtained by choosing $\gamma = T$ and $q = m - 1$.

Before giving an example of the computation of a ZOH equivalent for a system with time delay, we remark that the special case $D = mT$ occurs frequently when modeling digital control systems. The reason has to do with a common way of implementing digital control which is described next. An input to the compensator is read at time kT. The output value is calculated in a time which is less than the sampling period T; however, the output value is actually sent to the D/A converter at the *next* sampling instant $kT + T$. This procedure results in a "computational" time delay of exactly T seconds, as shown in Fig. 4.6. If this procedure is used to control a linear, time invariant plant described by the

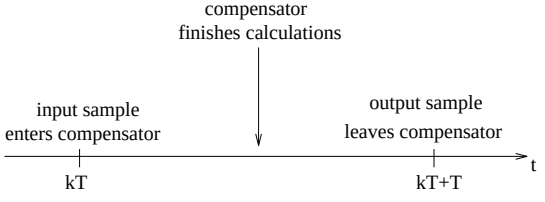


Figure 4.6 Timing sequence of a single input/output pair of samples of a digital compensator using a typical implementation procedure. The output of the compensator is delayed by exactly one sample period.

state-space model $(\mathbf{A}, \mathbf{b}, \mathbf{c})$, then a time delay of $\gamma = T$ seconds (with $q = 0$) should be

included in the model of the plant. A digital compensator designed using the plant model with time delay will properly account for the computational delay.

EXAMPLE 4.5

Consider a one time-constant system with time delay that was introduced in Section 3.6.5. Let the time constant be 1 second and the time delay be 0.66 seconds. A differential equation for this system of the type shown in (4.43) is

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{x}(t) + u(t - 0.66)\\ y(t) &= \mathbf{x}(t).\end{aligned}$$

Note for this system the matrix $A = 1$ and $b = 1$. Suppose that we want to use a sampling rate of $T = 0.2$ seconds. In this case, the delay can be decomposed as

$$0.66 = 3(0.2) + 0.06$$

so that $q = 3$ and $\gamma = 0.06$. We compute $\mathbf{\Phi} = e^{AT} = e^{0.2} = 1.2214$, and use (4.48) and(4.49) to compute Γ_1 and Γ_0 as follows

$$\Gamma_1 = \int_{0.2-0.006}^{0.2} e^{\gamma}\, d\gamma = e^{\gamma}\,|_{0.14}^{0.2} = 0.0711,$$

and

$$\Gamma_0 = \int_{0}^{0.14} e^{\gamma}\, d\gamma = e^{\gamma}\,|_{0}^{0.14} = 0.1503.$$

Recall that we have to define $q + 1 = 4$ additional state variables $z_1[k], \cdots, z_4[k]$. Then the ZOH model from (4.57) is

$$\begin{bmatrix} x[k+1] \\ z_1[k+1] \\ z_2[k+1] \\ z_3[k+1] \\ z_4[k+1] \end{bmatrix} = \begin{bmatrix} 1.2214 & 0.0711 & 0.1503 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x[k] \\ z_1[k] \\ z_2[k] \\ z_3[k] \\ z_4[k] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u[k]$$

$$y[k] = [1\ 0\ 0\ 0\ 0] \begin{bmatrix} x[k] \\ z_1[k] \\ z_2[k] \\ z_3[k] \\ z_4[k] \end{bmatrix}.$$

■

EXAMPLE 4.6

Consider a type-1 servo system (e.g. a dc motor) described by poles at $s = 0$ and $s = -10$. Suppose that the sampling period is $T = 0.01$ seconds and that a digital compensator will introduce a time delay of one period. We can incorporate this time

delay into the ZOH model of the plant as follows:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -10 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t - 0.01)$$

$$y(t) = [1\ 0]\mathbf{x}(t).$$

A ZOH equivalent model can be obtained by computing $\mathbf{\Phi}$, $\mathbf{\Gamma}_0$ and $\mathbf{\Gamma}_1$ and substituting these into (4.58). Example 4.1 derives the ZOH equivalent model for this system without time delay. The matrix $\mathbf{\Phi}$ will be the same in both examples, and so we evaluate the expression for $\mathbf{\Phi}$ in example 4.1 with $T = 0.01$ and $p = 10$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.9048 \end{bmatrix}.$$

Recall that $\mathbf{\Gamma}_0$ defined in (4.49) equals $\mathbf{0}$, and since $\gamma = T$, we have from (4.48) that $\mathbf{\Gamma}_1 = \mathbf{\Gamma}$, where $\mathbf{\Gamma}$ is given in Example 4.1

$$\mathbf{\Gamma}_1 = \begin{bmatrix} 0.0000 \\ 0.0095 \end{bmatrix}.$$

We can now use (4.58) to obtain the ZOH equivalent model

$$\begin{bmatrix} \mathbf{x}[k+1] \\ z_1[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0.0952 & 0 \\ 0 & 0.9048 & 0.0095 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}[k] \\ z_1[k] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u[k]$$

$$y[k] = [\,1 \quad 0 \quad 0\,] \begin{bmatrix} \mathbf{x}[k] \\ z_1[k] \end{bmatrix}.$$

This model can be used to assess the effect of time delay on a closed-loop control system. This is done in the next chapter.

■

4.4 The ZOH Pole-Mapping Formula

When the plant is replaced by its ZOH equivalent, the design model is a discrete-time system. The performance specifications for the design model are thus discrete-time specifications. However the actual control system operates in continuous time and it is natural to use continuous-time performance specifications. A convenient way to achieve performance specifications is to specify the pole locations that the closed-loop system should have. For continuous-time specifications, we would place closed-loop poles in the s-plane. Since the design model is a discrete-time system, however, the desired closed-loop s-plane poles must be mapped into an equivalent set of desired z-plane poles. In this section we show that the poles of the ZOH design model are related in a simple way to the poles of the continuous-time plant. This relationship will be used in the next chapter to map desired s-plane pole locations into the z-plane.

It turns out that there is no simple relationship between the zeros of a continuous-time system and its ZOH equivalent [4]. However, there is a formula which relates the respective pole locations. Recall that the system matrices for the state-space descriptions of $G(s)$

and $G_d(z)$ are related by the formula $\mathbf{\Phi} = e^{\mathbf{A}T}$ (see (4.12)). Also recall from Chapter 3 that the poles of a system are given by the eigenvalues of its system matrix. So the question of how the poles map under the ZOH transformation reduces to the question of how the eigenvalues of a matrix are transformed by the matrix exponential. This question can be answered by the following remarkable fact about functions of matrices and their eigenvalues:

Fact. Let $\mathbf{B} = f(\mathbf{A})$, where $\mathbf{A}$ and $\mathbf{B}$ are matrices. If λ_i is an eigenvalue of $\mathbf{A}$ then $f(\lambda_i)$ is an eigenvalue of $\mathbf{B}$.

The proof of this fact is based on the Cayley-Hamilton theorem, and can be established by comparing (4.25) and (4.28).

Since $\mathbf{\Phi}$ and $\mathbf{A}$ are related by $\mathbf{\Phi} = e^{\mathbf{A}T}$, the eigenvalues of $\mathbf{\Phi}$ and $\mathbf{A}$ are related by the same exponential function. That is, from the definition of $\mathbf{\Phi}$ and the above fact, a given s-plane pole location s_0 maps into the z-plane by the ZOH mapping to the location

$$z_0 = e^{s_0 T}. \tag{4.60}$$

Note that the exponential function maps the left half of the s-plane into the interior of the unit circle in the z plane. This means that stable analog systems map to stable discrete-time systems.

It is sometimes useful to invert (4.60); that is, to solve for s_0 in terms of z_0. This is accomplished by taking logarithms of both sides of (4.60) to obtain

$$s_0 = \frac{1}{T} \ln(z_0). \tag{4.61}$$

When z_0 is a complex number, the above formula requires the logarithm of a complex number. The result can be expressed in terms of the logarithm of a real number as follows:

$$s_0 = \frac{1}{T}[\, \ln(|z_0| + j \arg(z_0)] \tag{4.62}$$

where $|z_0|$ is the magnitude of the complex number z_0, and $\arg(z_0)$ is the argument (angle in radians) of z_0. If $s_0 = \sigma + j\omega_0$, we assume that the sampling interval satisfies

$$\omega_0 T < \pi$$

so that the principal argument of z_0 (in the range $(-\pi, \pi)$) gives the correct value of s_0.

EXAMPLE 4.7

In Example 4.6, the $\mathbf{A}$ matrix of the type-1 servo was given as

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

and the matrix $\mathbf{\Phi}$ for the ZOH equivalent model with sampling period $T = 0.01$ was given as

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0.0952 \\ 0 & 0.9048 \end{bmatrix}.$$

Both $\mathbf{A}$ and $\mathbf{\Phi}$ are upper-triangular matrices which means that their eigenvalues are simply the elements on the main diagonal (Fact 2.39). The eigenvalues of $\mathbf{A}$ (poles of the analog system) are

$$s_1 = 0 \text{ and } s_2 = -10.$$

The corresponding eigenvalues of $\mathbf{\Phi}$ (poles of the ZOH equivalent model) are

$$z_1 = 1 = e^{0T} \text{ and } z_2 = 0.9048 = e^{-10T}.$$

Thus the s-plane and z-plane pole locations satisfy (4.60).

■

The following is a list of how certain s-plane pole locations map into the z-plane under the ZOH pole-mapping formula (4.60). These mappings are illustrated in Fig. 4.7.

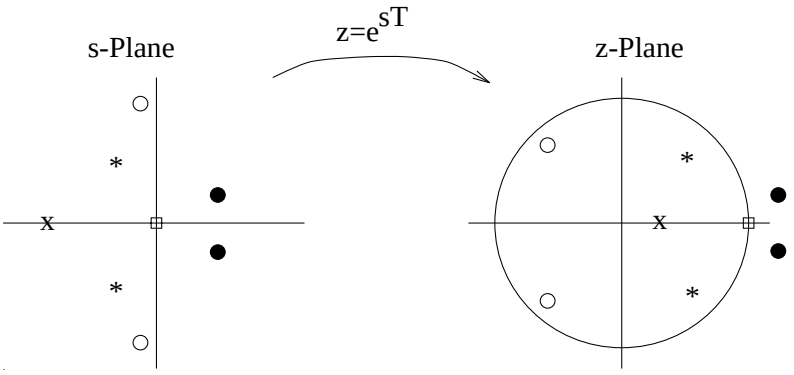


Figure 4.7 Examples of the ZOH pole-mapping formula.

- A pole at the origin of the s-plane, $s_0 = 0$, maps to a pole at $z_0 = e^{0T} = 1$ in the z-plane.
- A pole on the negative real axis of the s-plane, $s_0 = -p$, maps to a pole on the positive real axis in the z-plane, $z_0 = e^{-pT}$. The z-plane pole location will be inside the unit circle because the exponential of a negative number lies between 0 and 1.
- Complex conjugate poles in the left half of the s-plane, $s_{0,1} = -p \pm jq$, map to complex conjugate poles inside the unit circle in the z-plane, $z_{0,1} = e^{(-p \pm jq)T} = e^{-pT}[\cos(qT) + j\sin(qT)]$. The magnitude of the z-plane pole locations is $|z_{0,1}| = e^{-pT} \leq 1$. The angle of the z-plane pole locations is qT radians.
- Complex conjugate poles on the $j\omega$ axis in the s-plane, $s_{0,1} = \pm j\omega_0$, map to complex conjugate poles on the unit circle in the z-plane, $z_{0,1} = e^{\pm j\omega_0 T}$.
- Poles in the right half of the s-plane map to poles outside the unit circle in the z-plane.

4.5 Calculating Intersample Response

Consider a linear system driven by a piecewise-constant input, as shown in Fig. 4.8(a). Assume that the D/A converter holds input samples at a constant value for T seconds. The state-vector $\mathbf{x}(t)$ of the system will be a continuous function of t, even when the input is piecewise-constant (discontinuous). If we want to compute samples of the state vector a

integer multiples of T, we can use the ZOH equaivalent system show in Fig. 4.8(b). From the results in Section 4.1 we know that

$$\mathbf{x}[k] = \mathbf{x}(t)|_{t=kT}.$$

Suppose now that we want to compute samples of the state vector at integer multiples of T_1 where $T_1 < T$, while the input signal is still piecewise constant over intervals of length T. In other words, we want to compute samples of $\mathbf{x}(t)$ *in between* the samples $\mathbf{x}(kT)$ considered above. There are several reasons why we might be interested in calculating this *intersample response*. One reason is to obtain a plot of the continuous function $\mathbf{x}(t)$ by connecting sample values with straight lines. Samples separated by T seconds may not give an accurate plot of $\mathbf{x}(t)$ when connected by straight lines, while samples separated by $T_1 < T$ can be used to obtain a better plot. Another reason for computing intersample response is to check for *intersample ripple*, which occurs when $\mathbf{x}(t)$ is well behaved at samples $\mathbf{x}(kT)$, but $\mathbf{x}(t)$ oscillates between these samples. Intersample ripple can occur in improperly designed digital control systems. The design procedures given in this book will result in digital control systems which are free from intersample ripple.

The calculation of intersample response is based on the following observation: *an input signal $u(t)$ which is piecewise constant over intervals of length T seconds is also piecewise constant over synchronized intervals of length $T_1 = T/N$ seconds, for any positive integer N.* An example of this fact with $N = 3$ is shown in Fig. 4.9. The input to the system is generated by passing $u[k]$ through a D/A converter which holds each value constant for T seconds. However, the input is also constant over intervals of length T_1 seconds, and we can use a ZOH equivalent model to calculate samples of the state vector every T_1 seconds. The ZOH equivalent is calculated for the sampling interval T_1

$$\mathbf{\Phi}_1 = e^{\mathbf{A}T_1}\ , \quad \mathbf{\Gamma}_1 = \int_0^{T_1} e^{\mathbf{A}\tau}\mathbf{B}\, d\tau.$$

The state-vector samples are then computed as follows:

$$\mathbf{x}[k+1] = \mathbf{\Phi}_1\mathbf{x}[k] + \mathbf{\Gamma}\bar{u}[k]$$

where the input sequence is defined below:

$$\bar{u}[k] = u[\text{int}(k/N)]\ , \quad k = 0, 1, \cdots$$

and int$(\cdot)$ means "integer part of." The state-vector samples are exact at multiples of T_1; that is

$$x[k] = x(kT_1).$$

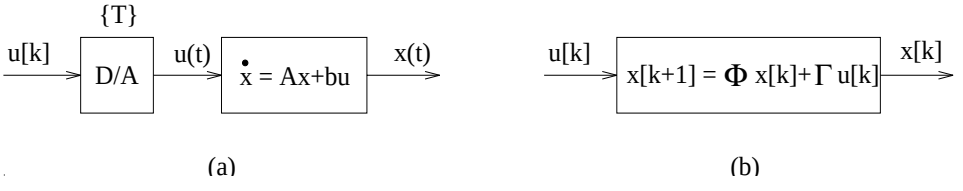


Figure 4.8 (a) A linear system driven by an input signal that is piecewise constant over intervals of length T seconds. (b) The ZOH equivalent that computes samples of the state vector for sampling interval T.

```
function [t,x]=intersamp(A,B,x_0,T_1,N,u)
[phi,gamma]=c2d(A,B,T_1); %Calculate ZOH equivalent
                          %at sampling interval T_1
k_1=length(u);  %The vector u contains input samples
         % which are held constant for N T_1 seconds.
kf= k_1*N;
n=length(B);
x=zeros(n,kf);
x(:,1)=x_0;      % x_0 is the initial state vector
for k=1:k_1
  for j=1:N      % Hold input constant for N samples
    x(:,(k-1)*N+j+1)=phi*x(:,(k-1)*N+j)+gamma*u(k);
  end
end
t=[0:kf]*T_1;    % Time axis for plotting
return
```

Table 4.1 MATLAB program to compute the intersample response of a linear system driven by a piecewise-constant input. `T1` is the time interval at which state-vector samples are to be calculated. The vector `u` contains input samples which are held constant for `N T1` seconds.

A MATLAB program to compute the intersample response of a linear system $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t)$, driven by a piecewise-constant input, is shown in Table 4.1. The samples of the state vector are calculated every T_1 seconds, while the input to the system is held constant for NT_1 seconds.

EXAMPLE 4.8

Consider a Type-1 servo with the following state-space model:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t).$$

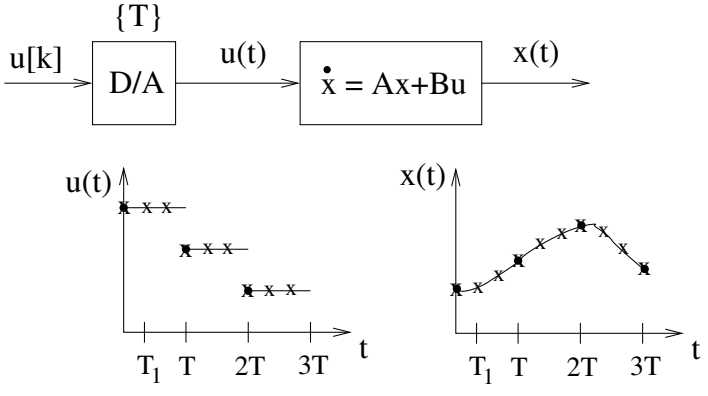


Figure 4.9 A linear system driven by an input signal that is piecewise constant over intervals of length T seconds. Samples of the input and state vector at sampling interval T are shown by dots, while inout and state vector samples at sampling interval $T_1 = T/3$ are shown by x's.

Suppose that the input to this system is generated by a D/A converter which holds samples for 1 second. That is, the sampling interval is $T = 1$. If the input sequence to the D/A converter is

$$u[0] = 3\ ,\ \ u[1] = 2\ ,\ \ u[2] = 1,$$

then the input to the plant is

$$u(t) \quad = \quad \begin{cases} 3,\ 0 \leq t < 1 \\ 2,\ 1 \leq t < 2 \\ 1,\ 2 \leq t < 3 \end{cases}$$

Since the input to the plant is piecewise-constant over intervals of length 1 second, we could compute a ZOH equivalent model at this sampling interval. This ZOH model could then be used to compute samples of the state vector at 1-second intervals. However, to obtain a smooth plot of $\mathbf{x}(t)$ we need to compute samples at a smaller sampling interval. In order to illustrate the procedure, we choose the smaller sampling interval to be $T_1 = 1/3$. We use the MATLAB program shown in Table 4.1 with $N = 3$. The results are shown in Fig. 4.10.

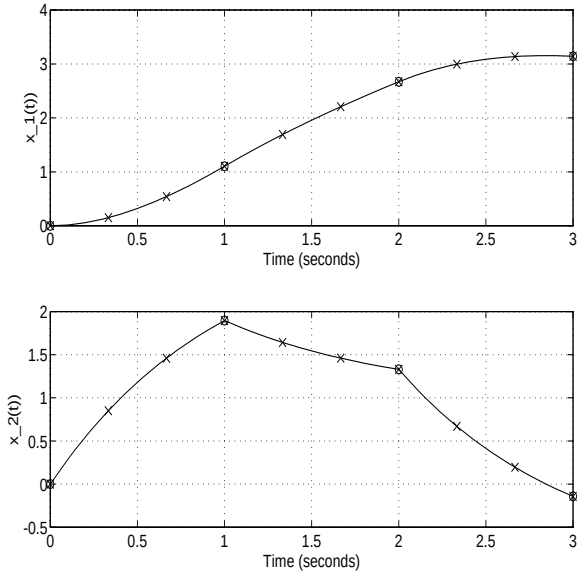


Figure 4.10 Examples of the ZOH pole-mapping formula.

4.6 Chapter Summary

In this chapter we showed that a linear system (plant) driven by a piecewise-constant input can be exactly represented at sampling instants by a discrete-time model called the ZOH

equivalent model, or ZOH for short. ZOH equivalents can be found even for systems with a time delay on the input.

Given a state-space model for the plant, the ZOH is obtained by computing a matrix exponential. Two procedures were given for computing a matrix exponential $e^{\mathbf{A}t}$. The first procedure is based on the Cayley-Hamilton theorem, while the second uses the eigenvectors of $\mathbf{A}$. We showed that the eigenvalues z_i of the ZOH model were related to the eigenvalues s_i of a given continuous-time model by the ZOH pole mapping formula

$$z_i = e^{s_i T}$$

where T is the sampling interval.

4.7 Problems

1. Given a continuous-time system

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

$$y(t) = \mathbf{C}\mathbf{x}(t)$$

the discrete-time ZOH equivalent model for sampling interval T is

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}u[k]$$

$$y[k] = \mathbf{C}\mathbf{x}[k]$$

where

$$\mathbf{\Phi} = e^{\mathbf{A}T} \text{ , and } \mathbf{\Gamma} = \int_0^T e^{\mathbf{A}\tau}\mathbf{B}d\tau.$$

The theory developed in this chapter states that if the input to the continuous-time system is piecewise constant with $u(kT) = u[k]$, then the ZOH model is exact at sampling instants as shown below:

$$\mathbf{x}[k] = \mathbf{x}(kT)$$

$$y[k] = y(kT).$$

The purpose of this problem is to verify these results numerically for some simple systems. In all cases the initial state vectors are zero we define the inputs to the discrete- and continuous-time systems as follows:

$$u[0] = 1 \text{ , } \; u[1] = 2 \text{ , } \; u[2] = 3,$$

and

$$u(t) = u[k] \text{ , } \; kT < t \leq kT + T$$

where $T = 0.1$ seconds. The continuous-time systems are:

(a)

$$\mathbf{A} = 0, \; \mathbf{B} = 1, \; \mathbf{C} = 1, \; D = 0.$$

(b)

$$\mathbf{A} = -1,\ \mathbf{B} = 1,\ \mathbf{C} = 1,\ D = 0.$$

(c)

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix},\ D = 0.$$

(d)

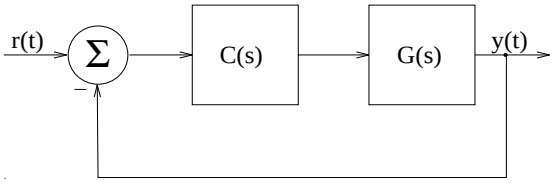
$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix},\ \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix},\ D = 0.$$

For each of the systems given above,

i. Calculate $e^{\mathbf{A}t}$ as a function of t.
ii. Use (4.4) to calculate the continuous-time state vector $\mathbf{x}(t)$ at $t = 0.1,\ 0.2,\ 0.3$.
iii. Calculate $\mathbf{\Phi}$ and $\mathbf{\Gamma}$.
iv. Use (4.8) to calculate the discrete-time state vector $\mathbf{x}[k]$ at $k = 1,\ 2,\ 3$ and verify that $\mathbf{x}[k] = \mathbf{x}(kT)$.

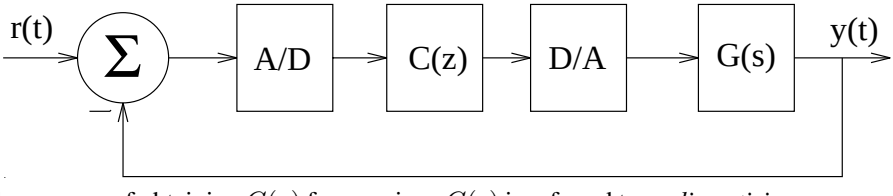
2. *Discretizing an Analog Compensator*

Consider the following analog control system:



Note that the input to the analog compensator $C(s)$ is $r(t) - y(t)$ which is not a piecewise-constant signal. Thus the ZOH equivalent of $C(s)$ will not describe its behavior exactly at sampling instants in this application. Nevertheless the ZOH of $C(s)$ can provide a useful *digital approximation* to $C(s)$ if the sampling interval T is small enough.

Let $C(z)$ be the transfer function of the ZOH equivalent of $C(s)$. Then the analog control system shown above can be approximated by the digital control system shown below:



The process of obtaining $C(z)$ from a given $C(s)$ is referred to as *discretizing an analog compensator*. A rule-of-thumb for choosing the sampling interval T is developed in Problem 4 of Chapter 5.

Suppose that the plant is a dc motor described by the transfer function

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

A phase-lead compensator which can be used to improve the transient response of the motor is

$$C(s) = 2\frac{s+2}{s+8}.$$

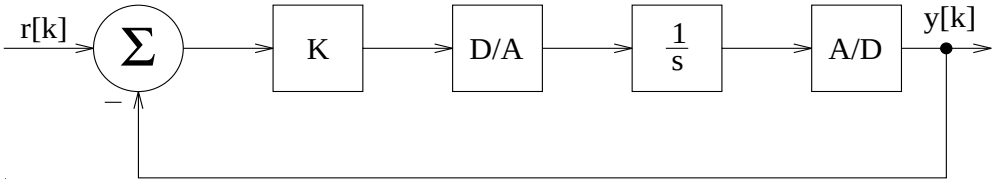
(a) Discretize $C(s)$ using its ZOH equivalent to obtain $C(z)$. Use the sampling interval $T = 0.1$

(b) Run a simulation and compare the step responses of the analog control system using $C(s)$ and the digital control system using $C(z)$.

3. Calculate the ZOH equivalent model for the following system with a sampling period of $T = 0.1$ sec.

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -2 & 1 \\ -2 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t).$$

4. Consider the following system under digital control. Replace the plant by its ZOH equivalent as a function of an unspecified sampling period T to obtain a completely digital model.



(a) Show that the resulting closed-loop digital system can go unstable for large values of K.

(b) If $K = 1$ show that the closed-loop digital system can go unstable for large values of T.

(c) If the digital system considered in parts (a) or (b) above is unstable, is the actual system shown in the figure stable or unstable? Explain your answer.

5. Let the sampling period for this problem be $T = 0.1$ seconds.

(a) Use the ZOH pole mapping formula to map the following S-plane poles into the Z-plane:

$$s_0 = -1 + j2$$
$$s_1 = -1 - j2.$$

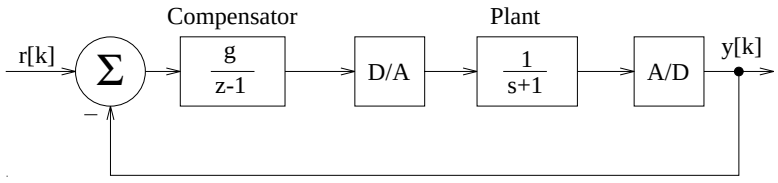
Show all work.

(b) Suppose the following z-plane pole locations have been obtained from some s-plane pole locations using the ZOH pole mapping formulas with $T = 0.1$:

$$z_0 = 0.8868 + j0.1798$$
$$z_1 = 0.8868 - j0.1798.$$

Calculate the s-plane pole locations. Show all work.

6. Consider the digital control system shown below

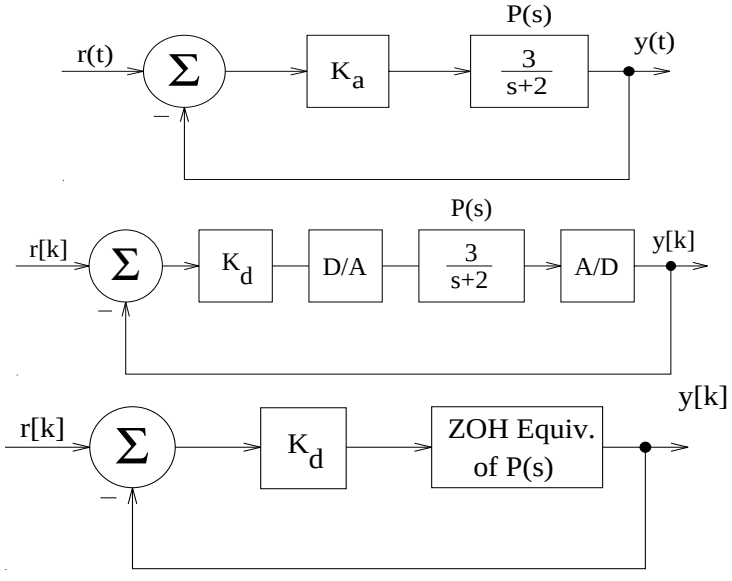


(a) Obtain the ZOH equivalent of the plant for a sampling period of $T = 0.2$ sec.

(b) Obtain the transfer function of the ZOH model.

(c) For what value of the gain $g > 0$ does the closed-loop system go unstable?

7. Consider the three systems shown below.



(a) Show that the closed-loop system in the first figure is stable for any value of the gain $K_a > 0$.

(b) Find the poles of the closed-loop system shown in the second figure as a function of K_d and T.

(c) Show that K_d can always be made large enough so that the pole location for the closed-loop system computed in part (b) will be less than -1 (i.e. the closed-loop system can be made unstable if K_d is large enough, regardless of how small T is).

(d) If $K_a = 1$ and $T = 0.01$, calculate K_d so that the pole location z_0 for the system in the third figure and the pole location s_0 for the system in the first figure are related by $z_0 = e^{s_0 T}$.

(e) Using the values of K_a and K_d from the previous part, show that the closed-loop system in the second figure can go unstable for large values of T. Calculate the upper bound on T for which the system will remain stable.

(f) The value of K_d was calculated in part (d) so that, for the specified values of K_a and T, $z_0 = e^{s_0 T}$. Suppose that K_d remains fixed, but $2T$ is used as the sampling rate inadvertently instead of T. The system in the second figure will behave like an analog system with pole location s_1. Calculate the value of s_1.

(g) If the discrete-time system in the third figure is unstable, will the digital control system in the second figure be stable or unstable? Explain your answer.

8. The matrices $\mathbf{A}$ and $\mathbf{\Phi}$ shown below are related by $\mathbf{\Phi} = e^{\mathbf{A}T}$, where $T = 0.1$.

$$\mathbf{A} = \begin{bmatrix} 1 & 3 \\ 0 & 2 \end{bmatrix}, \quad \mathbf{\Phi} = \begin{bmatrix} 1.1052 & 0.3486 \\ 0 & 1.2214 \end{bmatrix}.$$

Calculate the matrix function $\ln(\mathbf{\Phi})$ (natural logarithm) using the Cayley-Hamilton theorem. Show all work.

9. It is known that matrix multiplication is not commutative in general, so that $\mathbf{AB} \neq \mathbf{BA}$ for arbitrary matrices $\mathbf{A}$ and $\mathbf{B}$. However, suppose $\mathbf{B}$ is the matrix function $\mathbf{B} = f(\mathbf{A})$, where the scalar function $f(\tau)$ is represented by the power series $f(\tau) = \sum_{i=0}^{\infty} c_i \tau^i$, and $\mathbf{A}$ and $\mathbf{B}$ are square matrices. Show that $\mathbf{AB} = \mathbf{BA}$ in this case.

10. This problem leads you through a derivation of Van Loan's formula, (4.14).

Given a continuous-time system $(\mathbf{A}, \mathbf{b})$, the ZOH equivalent $(\mathbf{\Phi}, \mathbf{\Gamma})$ at sampling interval T is defined by

$$\mathbf{\Phi} = e^{\mathbf{A}T}, \quad \mathbf{\Gamma} = \int_0^T e^{\mathbf{A}\tau} d\tau \, \mathbf{b}.$$

The matrix exponential is defined by the power series

$$e^{\mathbf{A}\tau} = \mathbf{I} + \mathbf{A}\tau + \frac{\mathbf{A}^2\tau^2}{2!} + \frac{\mathbf{A}^3\tau^3}{3!} + \cdots + \frac{\mathbf{A}^k\tau^k}{k!} + \cdots. \tag{4.63}$$

(a) Using (4.63) and the definition of $\mathbf{\Gamma}$, show that the power series expansion for $\mathbf{\Gamma}$ is

$$\mathbf{\Gamma} = \sum_{k=0}^{\infty} \frac{1}{(k+1)!} \mathbf{A}^k T^{k+1} \mathbf{b}.$$

(b) Consider the $(n+1) \times (n+1)$ matrix

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & 0 \end{bmatrix}.$$

Show by induction that

$$\mathbf{M}^k = \begin{bmatrix} \mathbf{A}^k & \mathbf{A}^{k-1}\mathbf{B} \\ \mathbf{0} & 0 \end{bmatrix}$$

for $k = 1, 2, 3, \cdots$. In other words, first show that the expression for $\mathbf{M}_k$ is true for $k = 1$; then, assuming it is true for some k, show that it is true for $k+1$.

(c) Using the results from (a) and (b), show that

$$e^{\mathbf{M}T} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{0} & 1 \end{bmatrix}.$$

11. Table 3.6 on page 130 lists the following result

$$\int_0^t e^{\mathbf{M}\tau}d\tau = \mathbf{M}^{-1}\left(e^{\mathbf{M}t} - \mathbf{I}\right) = \left(e^{\mathbf{M}t} - \mathbf{I}\right)\mathbf{M}^{-1} \text{ for any invertible matrix } \mathbf{M}.$$

The purpose of this problem is to consider what happens when $\mathbf{M}$ is *not* an invertible matrix.

Recall that the matrix exponential is defined by the infinite power series

$$e^{\mathbf{M}\tau} = \mathbf{I} + \mathbf{M}\tau + \frac{\mathbf{M}^2\tau^2}{2!} + \frac{\mathbf{M}^3\tau^3}{3!} + \cdots = \sum_{k=0}^{\infty}\frac{\mathbf{M}^k\tau^k}{k!}. \tag{4.64}$$

(a) Substitute (4.64) into the following integral

$$\int_0^t e^{\mathbf{M}\tau}d\tau$$

and perform the integration from 0 to t. Express the answer as an infinite power series.

(b) Using your result from (a) calculate $\int_0^t e^{\mathbf{M}\tau}d\tau$ for the following singular matrix

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

Notice that $\mathbf{M}^k = \mathbf{0}$ for $k \geq 2$.

(c) Repeat (b) for the following singular matrix

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

Notice that $\mathbf{M}^k = \mathbf{M}$ for $k \geq 1$.

CHAPTER 5

FREQUENCY DOMAIN ANALYSIS

Frequency domain tools are very useful for analyzing the behavior of linear systems, and in particular, linear control systems. The frequency response of a system describes the steady-state output in response to a sinusoidal input. The variation of the system response as a function of the frequency of the input sinusoid gives the frequency response of the system. The frequency response of a linear system is useful in determining the response of the system to other inputs besides sinusoids. The reason is that an input signal $u(t)$ can be represented as a sum of sinusoids by Fourier analysis. The responses of the system to each of these sinusoids can be added together to obtain the response to $u(t)$ (this is the superposition property of linear time-invariant systems).

In control system analysis, the frequency response of the combined compensator/plant system is of particular importance. The frequency response can be used to compute gain and phase margins, which are a measure of how close a given control system is to being unstable. Because a digital control system has both a continuous-time part (the plant) and a discrete-time part (the compensator), it is necessary to discuss the frequency response of both continuous-time and discrete-time systems. In addition, it is necessary to show the relationship between the frequency response of a continuous-time system, and the frequency response of a discrete-time system that operates using samples of the continuous-time signals.

We begin this chapter by deriving expressions for the frequency responses of continuous- and discrete-time systems and show plots of these responses (Bode plots). Then Nyquist plots for discrete-time systems are introduced and gain and phase margins are defined for

digital control systems. Finally, we show how to scale a transfer function to achieve a desired speed of response, and consider the steady-state error of a control system.

5.1 Sinusoidal Response of Linear Systems

5.1.1 Continuous-Time Systems

Consider a stable transfer function $G(s)$ with input $u(t)$ and output $y(t)$. If we denote the Laplace transforms of $u(t)$ and $y(t)$ by $U(s)$ and $Y(s)$, respectively, then we have

$$Y(s) = G(s)U(s). \tag{5.1}$$

We now derive the response of $G(s)$ to an input sinusoid. The derivation is a simple application of the partial fraction expansion introduced in Chapter 3. If $u(t)$ is the sinusoidal signal $U_0 \sin \omega t$, then from Table 3.2 we see that

$$U(s) = \frac{U_0\omega}{s^2 + \omega^2} = \frac{U_0\omega}{(s - j\omega)(s + j\omega)}. \tag{5.2}$$

Assume for simplicity that the poles of the transfer function $G(s)$ are distinct at locations $p_1, \cdots, p_n$. Then from (5.1), a partial fraction expansion for $Y(s)$ can be written in terms of the poles of $G(s)$ and $U(s)$. Since all the poles are distinct, the partial fraction expansion will take the form

$$Y(s) = \sum_{i=1}^{n} \frac{A_i}{s - p_i} + \frac{\alpha}{s - j\omega} + \frac{\alpha^*}{s + j\omega}. \tag{5.3}$$

If the poles of $G(s)$ were *not* distinct, then the summation term in the above equation would be slightly more complicated (cf. (3.8)), and the results that follow would be altered only in minor detail. Thus it is sufficient to consider only the case of distinct poles.

The inverse Laplace transform of the right-hand side of (5.3) can be found using the third entry in Table 3.2 to be

$$y(t) = \sum_{i=1}^{n} A_i e^{p_i t} + \alpha e^{j\omega t} + \alpha^* e^{-j\omega t}. \tag{5.4}$$

Because the poles p_i of $G(s)$ are assumed to be in the left half plane, the first term of the right-hand side of (5.4) is a *transient* term which will go to zero as t increases. The second and third terms will give the steady-state sinusoidal response as shown below.

The residue α can be computed using (3.9) as follows

$$\begin{aligned} \alpha &= Y(s)\ (s - j\omega)\ |_{s=j\omega} \\ &= G(s)\ U(s)\ (s - j\omega)\ |_{s=j\omega} \\ &= G(s)\ U_0\ \frac{\omega}{s + j\omega}\ |_{s=j\omega} \\ &= G(j\omega)\ U_0\ \frac{1}{2j}. \end{aligned} \tag{5.5}$$

The complex residue α can be written in polar form (magnitude and angle) as

$$\alpha = |\alpha| e^{j\gamma} \tag{5.6}$$

where

$$|\alpha| = \frac{U_0}{2}|G(j\omega)|\,, \quad \text{and} \;\; \gamma = \angle(G(j\omega)) - \pi/2, \tag{5.7}$$

and "$\angle$" refers to the argument (or angle) of a complex number. Recall that the complex conjugate of α in (5.6) is $\alpha^* = |\alpha|e^{-j\gamma}$ so that (5.4) can be re-written as

$$y(t) = \sum_{i=1}^{n} A_i e^{p_i t} + |\alpha|e^{j(\omega t+\gamma)} + |\alpha|e^{-(\omega t+\gamma)}. \tag{5.8}$$

Steady-State Response

If we define the sum of the second and third terms on the right-hand side of equation (5.8) to be $y_S(t)$, the steady-state response, then

$$\begin{aligned} y_S(t) &= |\alpha|\left(e^{j(\omega t+\gamma)} + e^{-j(\omega t+\gamma)}\right) \\ &= 2|\alpha|\cos(\omega t + \gamma) \end{aligned} \tag{5.9}$$

where we have made use of *Euler's identity*

$$e^{j\theta} = \cos\theta + j\sin\theta. \tag{5.10}$$

Finally, we substitute (5.7) into (5.8) to obtain

$$y_S(t) = U_0|G(j\omega)|\cos(\omega t + \angle(G(j\omega)) - \pi/2), \tag{5.11}$$

and using the fact that $\cos(\theta - \pi/2) = \sin(\theta)$, we get

$$y_S(t) = A\sin(\omega t + \phi) \tag{5.12}$$

where

$$A = U_0|G(j\omega)|\,, \quad \text{and} \;\; \phi = \angle G(j\omega).$$

Thus if the sinusoid $U_0 \sin\omega t$ is the input to a stable linear system $G(s)$, then the output will be, after a transient, a sinusoid of the same frequency ω. The amplitude of the output sinusoid will be $|G(j\omega)|$ times the amplitude of the input sinusoid, and the output sinusoid will have a phase of $\angle(G(j\omega))$ relative to the input sinusoid.

The magnitude and phase plots which display the frequency response of a system are referred to as a *Bode plot* in honor of H. Bode who first used such plots for system analysis. The frequency axis of a Bode plot has a logarithmic scale, and the magnitude is plotted in units of decibels (dB). The conversion of a number x to decibels is accomplished by

$$y = 20\log(x)\ \text{dB} \tag{5.13}$$

where $\log$ is the base 10 logarithm, and the above formula may be inverted as follows

$$x = 10^{(y/20)}. \tag{5.14}$$

Transient Response

Before giving an example of the above results, we first consider the duration of the transient term. From (5.7), we define the transient term to be

$$y_T(t) = \sum_{i=1}^{n} A_i e^{p_i t}. \tag{5.15}$$

The pole locations can be written in terms of their real and imaginary parts as

$$p_i = c_i + jd_i$$

and

$$\begin{aligned} |e^{p_i t}| &= |e^{c_i t} e^{jd_i t}| \\ &= |e^{c_i t}| \\ &= e^{c_i t}. \end{aligned}$$

The magnitude of $y_T(t)$ can be bounded as follows

$$\begin{aligned} |y_T(t)| &\leq \textstyle\sum_{i=1}^{n} |A_i||e^{p_i t}| \\ &= \textstyle\sum_{i=1}^{n} |A_i| e^{c_i t}. \end{aligned} \tag{5.16}$$

The amount of time it takes for the transient term to go to zero depends on the values of c_i, the real parts of the pole locations, and on the residues A_i. An approximate analysis only depends on the c_i with the smallest magnitude, call it c. Then we can compute the value of t at which e^{ct} is a suitably small number. For example,

$$e^{ct} = 0.01 \text{ when } t = T_T = \frac{1}{c}\ln(0.01).$$

We can solve for T_T, which is the approximate duration of the transient response, as follows

$$T_T \approx \frac{-4.6}{c} \tag{5.17}$$

and the result is that

$$y_T(t) \approx 0 \text{ for } t > T_T.$$

EXAMPLE 5.1

Consider the following transfer function

$$G(s) = \frac{150.5693}{s^3 + 12.9429s^2 + 69.7997s + 150.5693}$$

which has poles at the following locations: -5.0093, $-3.9668 \pm j3.7854$. This is a normalized Bessel transfer function which is discussed in Chapter 6 and shown to have several desirable properties. One disadvantage is that the coefficients are somewhat messy!

The frequency response (magnitude and phase) $G(j\omega)$ is shown in Fig. 5.1. This

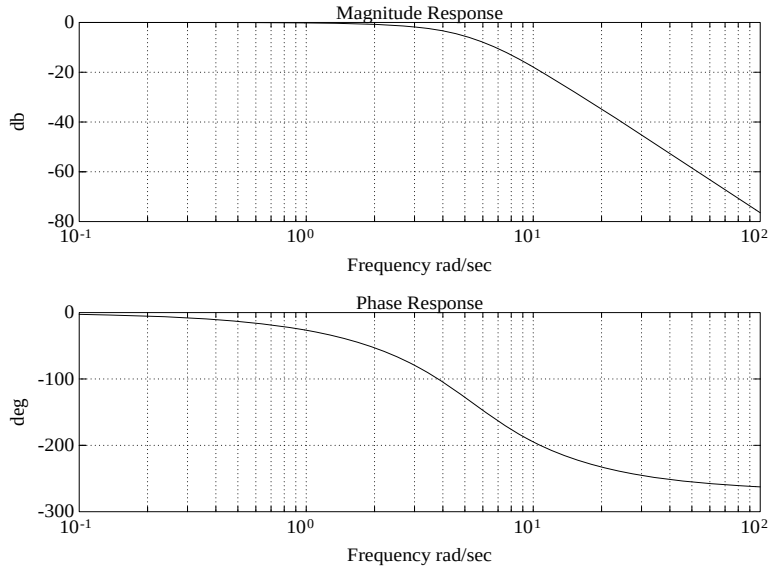


Figure 5.1 (a) Magnitude of $G(j\omega)$ for Example 5.1. (b) Phase of $G(j\omega)$ for Example 5.1.

figure gives the magnitude and phase of the steady-state output with respect to a sinusoidal input. For example, at 5 radians/second, the magnitude is about -6 dB $= 10^{-6/20} \approx 0.5$ and the phase is about -125 degrees (the exact numbers can be calculated to be 0.53 and -128). We should be able to see the effect of these numbers when the system input is a sinusoid of 5 radians/second. Fig. 5.2 shows this input sinusoid and the resulting output. The output has a transient part lasting about 1 second

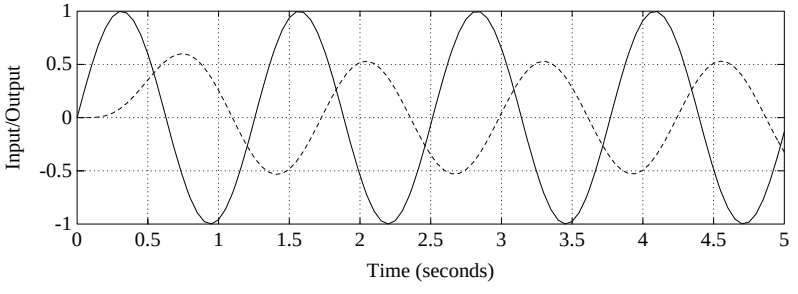


Figure 5.2 A sinusoid of frequency 5 radians/second (solid line) used as input to $G(s)$. The corresponding output of $G(s)$ is the dashed line.

before the steady-state sinusoidal response is achieved. The duration of the transient part agrees with equation (5.17) because the smallest real part of a pole location is $c = -3.9668$ so that $T_T \approx 4.5/3.9668 \approx 1.1$. The amplitude of the output sinusoid is indeed 0.53 times the amplitude of the input sinusoid. How can we check if the phase of the output sinusoid is about -125° with respect to the input sinusoid? The answer is that we can use the relationship between time shift of d seconds and phase shift of

ϕ radians/second for sinusoidal signals. It can be shown that

$$d = \frac{\phi}{\omega}$$

where ω is the frequency of input and output sinusoids (radians/second), ϕ is the relative phase (radians), and d is the time shift between the two sinusoids. A negative number for d indicates a time delay. For this example we convert the phase to radians and obtain

$$d = \frac{-125 * \pi/180}{5} \approx -.44.$$

Thus the output sinusoid should be 0.44 seconds delayed with respect to the input sinusoid. This can be seen from Fig. 5.4. Note, for example, that the input sinusoid has a peak at $t \approx 1.6$ seconds while the corresponding peak in the output sinusoid is at $t \approx 2$ seconds.

■

Suppose $G(s)$ in the previous example is the transfer function of a control system. How could we characterize the tracking performance of this control system? From the Bode plot, we see that if the input signal is a sinusoid of frequency 1 radian/second or less, then the output will be a sinusoid of the same frequency and amplitude with a slight phase lag. In other words, the control system will do a good job of tracking sinusoidal inputs with frequency less than 1 radian/second. Taking this further, we can say that the control system will do a good job of tracking a non-sinusoidal input $u(t)$ as long as the frequency content of $u(t)$ is below 1 radian/second.

Again from the Bode plot in Fig. 5.1 we see that the the control system cannot track a sinusoidal input of frequency 10 radians/second or higher. At 10 radians/second the output sinusoid would be about one tenth the magnitude of the input sinusoid. Thus the Bode plot shows the range of frequencies over which a control system will have good tracking performance. The highest such frequency is referred to as the *bandwidth* of the control system.

The bandwidth ω_B of a control system is defined (somewhat arbitrarily) as the frequency ω_B at which

$$|G(j\omega_B)| = \frac{1}{\sqrt{2}}|G(j0)|.$$

In decibels the factor of $1/\sqrt{2}$ is

$$20 \log \frac{1}{\sqrt{2}} \approx -3 \text{ dB}$$

so that at $\omega = \omega_B$ the Bode magnitude plot is 3 dB down from its value at zero frequency. For example, the control system in Example 5.1 has a bandwidth of about 4 radians/second (see Fig. 5.1). It is clear that a control system does not do a good job of tracking a reference sinusoid at ω_B rad/sec because of the factor of $1/\sqrt{2}$ difference in the desired and actual outputs. In addition, the output sinusoid is delayed by some amount with respect to the input sinusoid. Nevertheless, the frequency ω_B is used as a dividing line: the tracking performance for reference sinusoids with frequency less than ω_B is considered adequate, while the tracking performance for reference sinusoids with frequency greater than ω_B is considered inadequate. Thus the bandwidth of a control system is a measure of its speed of response. A control system with a large bandwidth is said to be "fast" because it can

track a quickly varying input (a high-frequency sinusoid). On the other hand, a system with small bandwidth is "slow" because it can only track slowly varying inputs.

The bandwidth of an analog control system can be used to determine the sampling interval T for a digital control system. Bandwidth calculations will be used in Chapters 6 and 8 to choose sampling intervals for state-space regulators and tracking systems.

5.1.2 Discrete Time

Now consider a stable discrete-time system $G(z)$ with input sequence $u[k]$ and output $y[k]$. Let $U(z)$ and $Y(z)$ denote the z-transforms of $u[k]$ and $y[k]$, respectively. Then

$$Y(z) = G(z)U(z). \tag{5.18}$$

Suppose $u[k]$ is obtained from a sinusoid of frequency ω sampled every T seconds. That is,

$$u[k] = U_0 \sin(\omega kT). \tag{5.19}$$

The z-transform of the sequence $\{u[k]\}$ obtained from Table 3.4 is

$$U(z) = \frac{U_0 z \sin(\omega T)}{z^2 - 2z\cos(\omega T) + 1} = \frac{U_0 z \sin(\omega T)}{(z - e^{j\omega T})(z - e^{-j\omega T})}. \tag{5.20}$$

As in the continuous-time case, we assume that the poles of $G(z)$ are distinct and write a partial fraction expansion for $Y(z)$ in terms of the poles of $G(z)$ and $U(z)$ as follows

$$Y(z) = \sum_{i=1}^{n} A_i \frac{z}{z - p_i} + \frac{\alpha z}{(z - e^{j\omega T})} + \frac{\alpha^* z}{(z - e^{-j\omega T})}. \tag{5.21}$$

The inverse z-transform can be found using Table 3.4 to be

$$y[k] = \sum_{i=1}^{n} A_i p_i^k + \alpha e^{j\omega kT} + \alpha^* e^{-j\omega kT}. \tag{5.22}$$

The first term on the right-hand side of this equation is the transient term which decays to zero. A simple analysis of the number of samples it takes for this term to approximately go to zero is as follows. Let p be the pole of largest magnitude, and let the number N_T be such that p^{N_T} is suitably small. For example

$$|p|^{N_T} = 0.01 \text{ when } N_T = \ln(0.01)/\ln(|p|) \approx -\frac{4.6}{\ln(|p|)}. \tag{5.23}$$

The number N_T is the approximate duration in samples of the transient response. We now proceed to compute the steady-state part of (5.22).

The residue α in (5.21) can be computed using (3.18) to be

$$\begin{aligned}
\alpha &= \frac{Y(z)(z - e^{j\omega T})}{z} \Big|_{z=e^{j\omega T}} \\
&= \frac{G(z)U_0 z \sin(\omega T)(z - e^{j\omega T})}{z(z - e^{j\omega T})(z - e^{-j\omega T})} \Big|_{z=e^{j\omega T}} \\
&= \frac{U_0 G(e^{j\omega T}) \sin(\omega T)}{(e^{j\omega T} - e^{-j\omega T})} \\
&= \frac{1}{2j} U_0 G(e^{j\omega T})
\end{aligned} \tag{5.24}$$

which can be expressed in polar form as

$$\alpha = |\alpha| e^{j\gamma} \tag{5.25}$$

where

$$|\alpha| = \frac{1}{2} U_0 |G(e^{j\omega T})| \text{ and } \gamma = \angle(G(e^{j\omega T}) - \frac{\pi}{2}. \tag{5.26}$$

If we define the sum of the second and third terms on the right-hand side of (5.22) to be $y_S[k]$, the steady-state response, we have

$$\begin{aligned} y_S[k] &= \alpha e^{j\omega kT} + \alpha^* e^{-j\omega kT} \\ &= 2|\alpha| \cos(\omega kT + \gamma) \\ &= A \sin(\omega kT + \phi), \end{aligned} \tag{5.27}$$

where the last equality is obtained by using (5.26) and the definitions

$$A = U_0 |G(e^{j\omega T})| \,, \;\; \phi = \angle(G(e^{j\omega T})). \tag{5.28}$$

Thus if the sinusoid $U_0 \sin(\omega kT)$ is the input to a stable discrete-time system, then the output will be, after a transient, a sinusoid of the same frequency ω. The amplitude of the output sinusoid will be $|G(e^{j\omega T})|$ times the amplitude of the input sinusoid, and the output sinusoid will have a phase of $\angle(G(e^{j\omega T}))$ relative to the input sinusoid. There are several important features of discrete-time frequency responses which we now state.

Remarks

1. Because of the Euler identity (5.10), a discrete-time frequency response $G(e^{j\omega T})$ is periodic in ω with period $2\pi/T$. Thus all of the information in $G(e^{j\omega T})$ is contained in its values on any interval of length $2\pi/T$. A convenient interval to use is $-\pi/T < \omega < \pi/T$.

2. A discrete-time frequency response is conjugate symmetric with respect to zero frequency (as is a continuous-time frequency response). In other words, the frequency response at some frequency $-\omega_0$ is the complex conjugate of the frequency response at $+\omega_0$. The conjugate symmetry is due to the fact that replacing ω by $-\omega$ gives the same result as replacing j by $-j$ in $e^{j\omega T}$. Since a frequency response is conjugate symmetric, it is customary to only plot it in the interval $0 \leq \omega \leq \pi/T$.

3. The periodicity of the frequency response mentioned above is related to the phenomenon of "aliasing" in sampled sinusoids. Aliasing refers to the fact that samples of a sinusoid of some frequency $\omega_2 > \pi/T$ are identical to samples of a sinusoid of a lower frequency ω_1 which is less than π/T. Because the samples of the two different sinusoids are identical, the corresponding output sinusoids must also be identical, giving rise to the same frequency response at two different frequencies.

   The following calculation illustrates the aliasing phenomenon. Consider a value of ω_2 in the range $\pi/T < \omega_2 < 2\pi/T$, and let $\omega_1 = 2\pi/T - \omega_2$. Notice that ω_1 is in

the range $0 < \omega_1 < \pi/T$. Then

$$\begin{aligned} \sin(k\omega_2 T) &= \sin(k(\omega_2 - 2\pi/T)T) \\ &= \sin(-k\omega_1 T) \\ &= -\sin(k\omega_1 T) \end{aligned} \tag{5.29}$$

where the last equality is true because $\sin(-x) = -\sin(x)$. Equation (5.29) says that at sampling instants $t = kT$, samples of the sinusoid $\sin(\omega_2 t)$ are identical to samples of the sinusoid $-\sin(\omega_1 t)$, where $\pi/T < \omega_2 < \pi/T$ and $0 < \omega_1 < \pi/T$.

From Remark 3 above we see that aliasing will not occur for a sinusoid of frequency ω_0 radians/second sampled with a sampling interval of T seconds if $\omega_0 < \pi/T$. This inequality is just a re-arrangement of the condition for the sampling theorem mentioned in Chapter 1. The following example illustrates the main points about discrete-time frequency responses.

EXAMPLE 5.2

Consider the following transfer function

$$G(z) = 55.2514 \frac{z^2 + 2.8833z + 0.5233}{z^3 - 1.8559z^2 + 1.2097z - 0.2741}$$

which has poles at the following locations: 0.6060, $0.6250 \pm j0.2485$. Suppose that the input to this system is obtained by sampling a continuous-time signal at a sampling period of $T = 0.1$ seconds. From Remarks 1 and 2 above we know that the frequency response in the interval $[-\pi/T, 0]$ is conjugate symmetric with the frequency response in $[0, \pi/T]$. If we look only at the magnitude plot, the response will be identical in these two intervals. In addition, the response will be periodic outside of these intervals. For a sampling period of $T = 0.1$ we have $\pi/T = 31.4$. The periodicity of the magnitude plot is shown in Fig. 5.3. As mentioned in Remark 2 above, it is

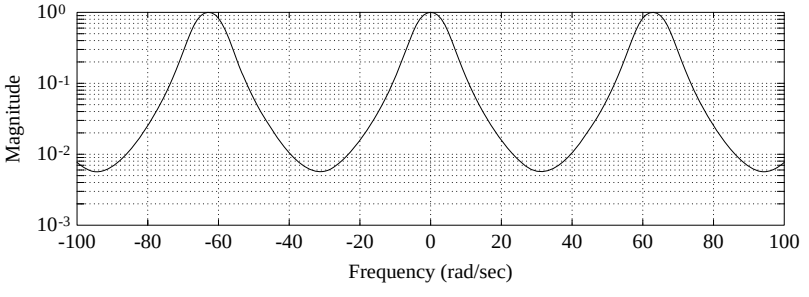


Figure 5.3 Magnitude of $G(e^{j\omega T})$ for Example 5.2.

customary to plot discrete-time frequency responses over the interval $0 \leq \omega \leq \pi/T$. The magnitude and phase of $G(e^{j\omega T})$ are shown for this range of ω in Fig. 5.4. Note that these figures use a logarithmic frequency scale.

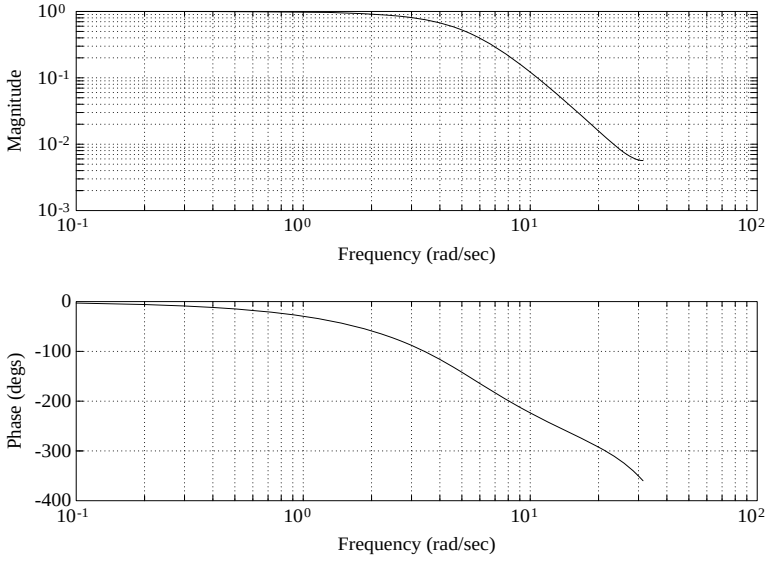


Figure 5.4 (a) Magnitude of $G(e^{j\omega T})$ for Example 5.2. (b) Phase of $G(e^{j\omega T})$ for Example 5.2.

We can also demonstrate the aliasing phenomena for sinusoids sampled at $T = 0.1$ seconds. Using the notation from Remark 3 above, we choose

$$\omega_1 = \frac{1.2\pi}{T} = 37.6991$$
$$\omega_2 = \frac{0.8\pi}{T} = 25.1327.$$

According to the calculations shown in Remark 3, the sinusoids $\sin(\omega_1 t)$ and $-\sin(\omega_2 t)$ should take on identical values at integer multiples of $T = 0.1$ seconds. This is shown in Fig. 5.5.

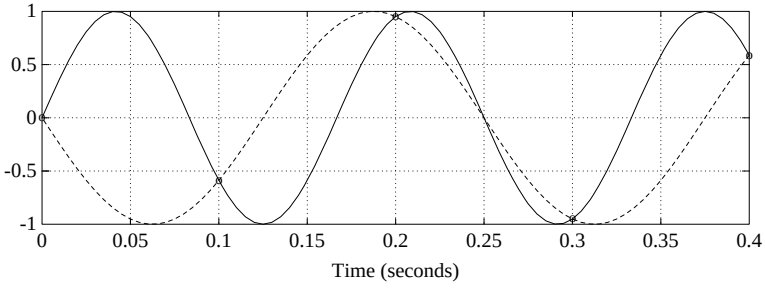


Figure 5.5 Two sinusoids of different frequencies whose values are identical at integer multiples of the sampling period $T = 0.1$ seconds. The solid line is $\sin(\omega_1 t)$ and the dashed line is $-\sin(\omega_2 t)$.

■

5.2 Stability Margins and Nyquist Plots

The frequency response of a system, as displayed by a Bode magnitude plot, is useful for determining the system bandwidth. In this section we show that the frequency response of a system, as displayed by a Nyquist plot, can also be used to determine the *stability margins* of a control system. The definition of stability margins is given in Section 5.3, but it is worthwhile to introduce the idea here.

The methods for designing control systems presented in Chapters 6–10 start with a state-space model of the plant and proceed with a set of calculations to obtain a state-space model of the compensator. If the calculations are done properly, the feedback interconnection of the plant and compensator will be a stable closed-loop system. This system is called the *nominal closed-loop system* because it is based on a *mathematical model* of the plant. The *actual closed-loop system* is the one that is obtained by connecting the *actual hardware plant* to the compensator. An important question is the following: given that the nominal closed-loop system is stable, under what conditions will the actual closed-loop system be stable?

The answer to this question clearly depends on the accuracy of the mathematical model of the plant. Assume for a moment that the plant model represents the hardware system *exactly*. Then the actual closed-loop system will be stable. However, if the plant model does not represent the hardware exactly, then it is possible that the actual closed-loop system will be unstable while the nominal closed-loop system is unstable. This is an unacceptable situation!

The concept of stability margins addresses the dilemma raised above as follows. Suppose we design a compensator for a given plant model. If we keep the compensator fixed, how much can we perturb the plant model before the closed-loop system becomes unstable? If the plant model can be perturbed a great deal before the closed-loop system becomes unstable, then the nominal control system is said to have large stability margins. In this case, the actual closed-loop system will be stable even if the mathematical model of the plant is not exact. On the other hand, if the plant model can only tolerate small perturbations before the closed-loop system becomes unstable (small stability margins) then the mathematical model of the plant will have to be very accurate to insure that the actual closed-loop system is stable.

In this section we show how a certain display of frequency response data can be used to determine the stability margins of a digital control system. In the derivation of the result we will consider the nominal closed-loop system shown in Fig. 5.6. We will show how the state-space control systems designed in Chapters 6–8 can be put in the form of Fig. 5.6. The stability margins developed in this chapter will be used in future chapters to compute the stability margins of state-space control systems.

We begin by deriving a graphical test to determine if the system in Fig. 5.6 is stable. Of course, if this figure represents a properly designed nominal control system then it will be stable. However we show in Section gapmd that this graphical stability test can also reveal the stability margins of the control system.

The derivation of the graphical stability test begins by considering the closed-loop system shown in Fig. 5.6. The transfer function of the closed-loop system from $r[k]$ to $y[k]$ is

$$\frac{KL(z)}{1+KL(z)}. \tag{5.30}$$

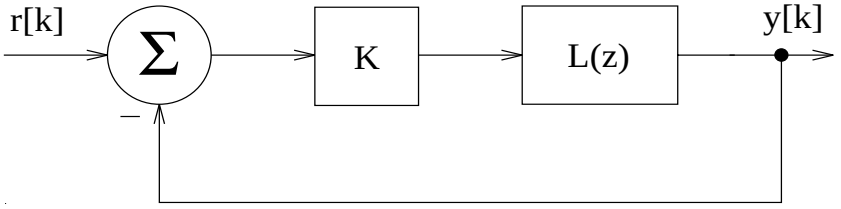


Figure 5.6 A closed-loop system.

The poles of the closed-loop system are values of z for which $1 + KL(z) = 0$, which can be rewritten as

$$L(z) = -\frac{1}{K}. \tag{5.31}$$

In other words, values of z which satisfy (5.31) are poles of the closed-loop system. *Thus if there is some value of z outside the unit circle which satisfies (5.31), then the closed-loop system is unstable.*

We can now state a conceptual test for stability: take every value of z outside the unit circle and compute the corresponding $L(z)$. If, for one of these values of z (5.31) is satisfied, then the closed-loop system is unstable. Consider a graphical implementation of this test. Fig. 5.7(a) shows the set of points outside the unit circle. This set of points will

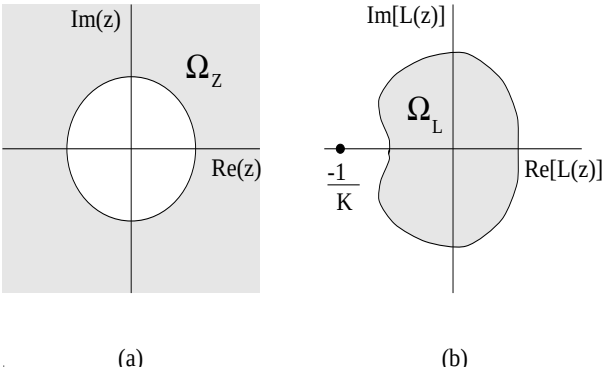


Figure 5.7 (a) The set of points $\Omega_Z = \{z : |z| \geq 1\}$. (b) The set $\Omega_L = \{L(z) : z \in \Omega_Z\}$.

be denoted Ω_Z. Fig. 5.7(b) shows the corresponding $L(z)$ points, which we denote by the set Ω_L. Is the closed-loop system in Fig. 5.6 stable? The answer is *yes* because none of the values of z outside the unit circle satisfy (5.31); in other words, the closed-loop system does not have any poles outside the unit circle. Notice that (5.31) is satisfied for some $z \in \Omega_Z$ if and only if the point $-1/K \in \Omega_L$.

For some other value of gain $K' > K$ the set of points Ω_Z shown in Fig. 5.8 could be obtained. In this case the closed-loop system is unstable because *some* value of z outside the unit circle is a pole of the closed-loop system. In other words, $L(z) = -1/K$ for some value of $z \in \Omega_Z$. Note that Fig. 5.8 does not tell us which value of $z \in \Omega_Z$ is a pole of the closed-loop system.

A question which must be answered is how we can obtain pictures like Fig. 5.7(b) and Fig. 5.8 without having to calculate $L(z)$ for every $z \in \Omega_Z$. The answer is that we only

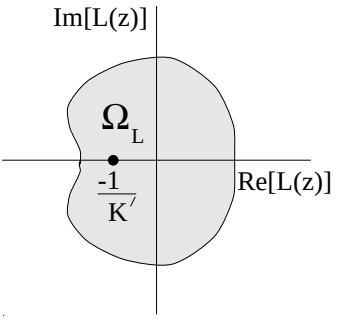


Figure 5.8 The set Ω_L with a different value of gain.

have to calculate $L(z)$ for z on the *boundary* of Ω_Z and the resulting set of points are the *boundary* of the set Ω_L that gives us information about the stability of the closed-loop system. Once the boundary of Ω_L is calculated, we then need to determine which set of points is "inside" the boundary. The situation can be more complicated than what is shown in Fig. 5.7.

In order to make use of results from complex variable theory, we need to define Ω_Z as the set of points "inside" some boundary. The "inside" of an oriented boundary is defined as follows: if you were to walk along the boundary in the direction of the arrows, the points inside the boundary will always be to your right. The set of points inside an oriented boundary is said to be *encircled* by that boundary. Fig. 5.9 shows the unit circle with two different orientations. The definition of encirclement (which points are "inside" a

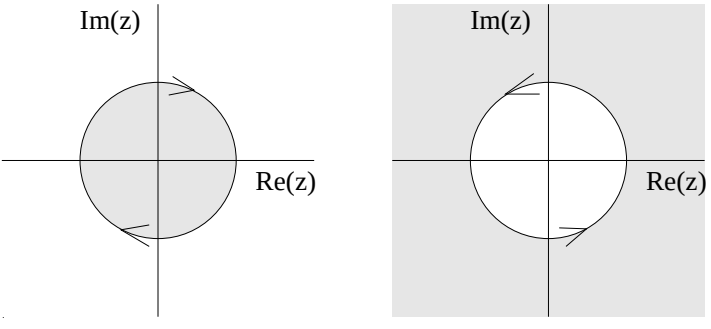


Figure 5.9 The unit circle with two different orientations. The set of points encircled by each boundary is indicated by the shading.

boundary) depends on the direction in which the boundary is traversed as shown in Fig. 5.9. Note that the set of points in Fig. 5.9(b) is Ω_Z.

We now describe a procedure to map the boundary of Ω_Z (i.e. the unit circle with counter-clockwise orientation) using $L(z)$. We begin by noting that if z_0 is a pole of $L(z)$, then $L(z_0)$ is undefined (infinite magnitude and undefined angle). Thus we cannot map z_0 into the $L(z)$ plane. In order to proceed when $L(z)$ has poles on the unit circle, we modify the boundary of Ω_Z slightly to avoid these poles. This modification is shown in Fig. 5.10 for the common case in which $L(z)$ has a pole at $z = 1$.

The boundary of the set shown in Fig. 5.10 is composed of two different parts

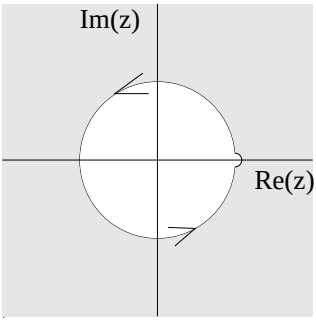


Figure 5.10 A boundary which avoids poles on the unit circle at $z = 1$.

1. The unit circle (excluding poles of $L(z)$), and
2. The small semi-circle(s) around unit-circle poles of $L(z)$.

The small semi-circle can be omitted if $L(z)$ does not have a pole at $z = 1$. We must calculate $L(z)$ for values of z on the unit circle and (possibly) on the small semi-circle.

Mapping the Unit Circle To map the unit circle, we calculate

$$L(z)|_{z=e^{j\omega T}} \ , \ \ 0 < \omega < \frac{2\pi}{T}. \tag{5.32}$$

This set of points, plotted imaginary part versus real part, is called a *polar plot* of $L(z)$. These are the same set of points that would be calculated for a Bode plot of $L(z)$. For a Bode plot, the magnitude and phase of $L(e^{j\omega T})$ are displayed separately as functions of ω. Notice that we don't evaluate $L(z)$ at $z = 1$ if $L(z)$ has a pole there.

Mapping the Semi-Circle Around $z = 1$ Suppose $L(z)$ has a pole of multiplicity m at $z = 1$. That is, $L(z)$ is of the form

$$L(z) = \frac{L'(z)}{(z-1)^m} \ , \ \ \gamma \stackrel{\text{def}}{=} L'(1) \tag{5.33}$$

where $L'(z)$ contains poles and zeros of $L(z)$ except the pole(s) at $z = 1$. This means that $\gamma = L'(1)$, a real, finite number. What values does $L(z)$ take as z takes on values on the small semi-circle near $z = 1$? To answer this question, we first write an equation for the semi-circle as follows

$$z = 1 + \rho e^{j\phi} \ , \ \ -\frac{\pi}{2} < \phi < \frac{\pi}{2}. \tag{5.34}$$

The next step is to evaluate $L(z)$ on the set of points described by (5.34), taking the limit as $\rho \to 0$ so that the semi-circle comes arbitrarily close to the pole(s) at $z = 1$. Using (5.33) we get

$$\begin{aligned} \lim_{\rho\to 0} L(1+\rho e^{j\phi}) &= \lim_{\rho\to 0} \frac{L'(1+\rho e^{j\phi})}{1+\rho e^{j\phi}-1)^m} \ , \ \ -\frac{\pi}{2} < \phi < \frac{\pi}{2} \\ &= \lim_{\rho\to 0} \frac{\gamma}{\rho e^{jm\phi}} \ , \ \ -\frac{\pi}{2} < \phi < \frac{\pi}{2}. \end{aligned} \tag{5.35}$$

ϕ	$\angle(L(z)) = -m\phi$
$-\pi/2$	$m\pi/2$
From $-\pi/2$ to $\pi/2$	m infinite semi-circles in the clockwise direction
$\pi/2$	$-m\pi/2$

Table 5.1 Table showing $\angle L(z)$ as a function of ϕ when z takes on values along the semi-circle $z = 1 + \rho e^{j\phi}$.

As $\rho \to 0$, $L(z)$ approaches a complex number with infinite magnitude and argument $-m\phi$ if γ is positive, $m\phi$ if γ is negative. We can calculate the argument of $L(z)$ as ϕ varies from $-\pi/2$ to $\pi/2$. Table 5.1 gives the results assuming γ is positive. If γ is negative, the signs of $\angle L(z)$ in the first and third rows of Table 5.1 are reversed.

EXAMPLE 5.3

Consider the first-order loop transfer function

$$L(z) = \frac{1}{z - 0.9}.$$

The Nyquist plot is shown in Fig. 5.11 and the set of points encircled is shaded. This plot was produced by the program **nyq** from the Digital Control Toolbox. The program puts an 'x' and an 'o' on the plot to show the direction in which the plot should be traversed (from 'x' to 'o'). These points should not be confused with pole or zero locations. The Nyquist plot intersects the negative real axis at -0.526. Thus the point $-1/K$ will not be in the shaded set of points (the closed-loop system will be stable) provided

$$0 < K < (1/0.526) = 1.9.$$

This bound on K can be verified by computing the pole of the closed-loop transfer

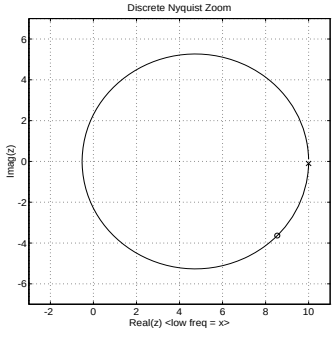


Figure 5.11 The Nyquist plot for Example 5.3.

function for positive values of K. The closed-loop pole is the solution to

$$1 + KL(z) = 1 + \frac{K}{z - 0.9} = 0$$

which is equivalent to the solution to

$$z - 0.9 + K = 0.$$

The solution of the above equation leaves the unit circle when $K > 1.9$.

■

5.3 Gain and Phase Margins

The primary use of Nyquist plots in this book is to illustrate stability margins for a negative unity-feedback control system with loop transfer function $KL(z)$ (see Fig. 5.6). In general we can take $L(z)$ to be the product of the compensator transfer function $C(z)$ and the plant transfer function $G(z)$ so that $K = 1$. If the compensator has been designed properly the closed-loop system will be stable and the point $-1/K = -1$ will not be encircled by the Nyquist plot. However, it may be that -1 is *close to being encircled* in the sense that a small change in the Nyquist plot would result in encirclement. Such changes in the Nyquist plot could be due to errors in modeling the plant.

Suppose that the *actual plant* has transfer function $\tilde{G}(z)$, but due to modeling errors, the mathematical model of the plant is given by $G(z)$. Suppose further that a compensator $C(z)$ is designed using the mathematical model $G(z)$ so that the nominal loop transfer function is $L(z) = C(z)G(z)$. However, the actual loop transfer function is

$$\tilde{L}(z) = C(z)\tilde{G}(z) \tag{5.36}$$

which might correspond to an unstable closed-loop system. Under what conditions will stability of the nominal closed-loop system imply that the actual closed-loop system is stable?

The above question can easily be answered if we restrict the relationship between $\tilde{G}(z)$ and $G(z)$. We will consider two simple relationships.

1. Gain perturbation: $\tilde{G}(z) = \alpha G(z)$ where α is a real-valued gain parameter. This relationship can be expressed in terms of the frequency responses of $\tilde{G}(z)$ and $G(z)$ as follows:
$$|\tilde{G}(e^{j\omega T})| = \alpha |G(e^{j\omega T})|\,, \;\; 0 \le \omega < \pi/T$$
$$\angle \tilde{G}(e^{j\omega T}) = \angle\, G(e^{j\omega T}).$$

2. Phase perturbation: this perturbation is expressed in terms of the frequency responses as follows
$$|\tilde{G}(e^{j\omega T})| = |G(e^{j\omega T})|\,, \;\; 0 \le \omega < \pi/T$$
$$\angle\, \tilde{G}(e^{j\omega T}) = \angle\, G(e^{j\omega T}) - \phi.$$

If $\tilde{G}(z)$ differs from $G(z)$ by either a gain or phase perturbation, then the Nyquist plot of the nominal system can be used to assess how large a perturbation can be tolerated before the actual closed-loop system goes unstable.

5.3.1 Gain Margin

Consider first a gain perturbation. In this case it is easy to see from (5.36) that $\tilde{L}(z) = \alpha L(z)$ so that the Nyquist plot corresponding to $\tilde{L}(z)$ is just a scaled version (real and

imaginary parts multiplied by α) of the nominal Nyquist plot. For stability analysis we only need to determine if the critical point at -1 is encircled by the perturbed Nyquist plot. Instead of multiplying the real and imaginary parts of the nominal Nyquist plot by α and checking for encirclement of -1, it is equivalent to check for encirclement of $-1/\alpha$ using the nominal Nyquist plot. In other words, we scale the critical point instead of the Nyquist plot. From Fig. 5.12 we see that the perturbed system will be stable for all values of α in

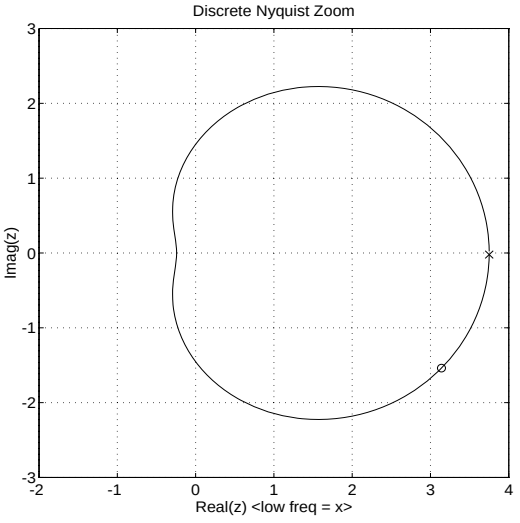


Figure 5.12 A nominal Nyquist plot. The point $-1/\alpha$ is encircled for all values of $\alpha \geq \alpha_{\max}$.

the range

$$0 < \alpha < \alpha_{\max}$$

where $\alpha_{\max}$ is characterized by the fact that $-1/\alpha_{\max}$ equals the point on the Nyquist plot which intersects the negative real axis. The value $\alpha_{\max}$ is called the *gain margin* (GM) of the system and is often expressed in decibels

$$GM = 20\log(\alpha_{\max}) \text{ dB}.$$

For some systems, it is necessary for $\alpha > \alpha_{\min}$ in order for the perturbed system to be stable. In this case the gain margin would be a negative number in dB. Complicated systems may have upper and lower gain margins as illustrated in the Problems.

5.3.2 Phase Margin

Now consider the case of a phase perturbation. The definition of a phase perturbation yields the following relationships

$$|\tilde{L}(e^{j\omega T})| = |L(e^{j\omega T})|$$

and

$$\angle\, \tilde{L}(e^{j\omega T}) = \angle\, L(e^{j\omega T}) - \phi.$$

The above two equations show that the points which make up the perturbed Nyquist plot have the same magnitudes as the corresponding points on the nominal Nyquist plot, but

the phase angles of the points on the perturbed plot are all ϕ degrees less than the corresponding phase angles on the nominal Nyquist plot. This means that in the case of phase perturbation, the perturbed Nyquist plot is obtained by rotating the nominal Nyquist plot $-\phi$ degrees. As in the case of gain perturbation, it is easier to perturb the critical point instead of the entire Nyquist plot. Instead of rotating the Nyquist plot by $-\phi$ degrees, we simply rotate the critical point by $+\phi$ degrees. The original critical point is at -1 which has a magnitude of 1 and a phase angle of 180 degrees. The rotated critical point still has magnitude 1 but has a phase angle of $180 + \phi$ degrees. The equivalence between rotating the Nyquist plot and rotating the critical point is shown in Fig. 5.13. Since the magnitude of

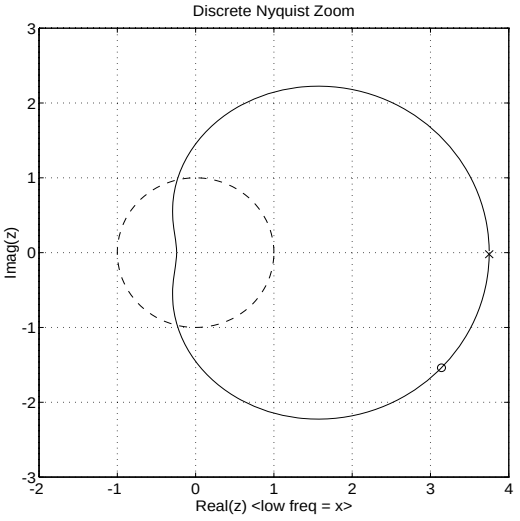


Figure 5.13 A nominal Nyquist plot together with the unit circle. $\phi_{\max}$ is the phase margin.

the critical point is 1 it is convenient to draw a unit circle, which is also shown in Fig. 5.13. From Fig. 5.13 we see that the perturbed system will be stable for all values of ϕ in the range

$$0 < \phi < \phi_{\max}$$

where $\phi_{\max}$ is characterized by the fact that the unit circle intersects the Nyquist plot at a phase angle of $180 + \phi_{\max}$ degrees. The value $\phi_{\max}$ is called the *phase margin* of the system.

EXAMPLE 5.4

Consider the following loop transfer function which has a pole at $z = 1$

$$L(z) = \frac{z}{(z-1)(z-0.905)}.$$

The Nyquist plot is shown in Fig. 5.14. The dotted part of the curve represents an infinite semi-circle. The set of points encircled by the plot is shaded, but because of the scales on the graph, it is difficult to see if -1 is encircled. The portion of the plot near -1 is shown in Fig. 5.15. From Fig. 5.15 we can see that -1 is not encircled and so the closed-loop system is stable. The Nyquist plot intersects the real axis at -0.26

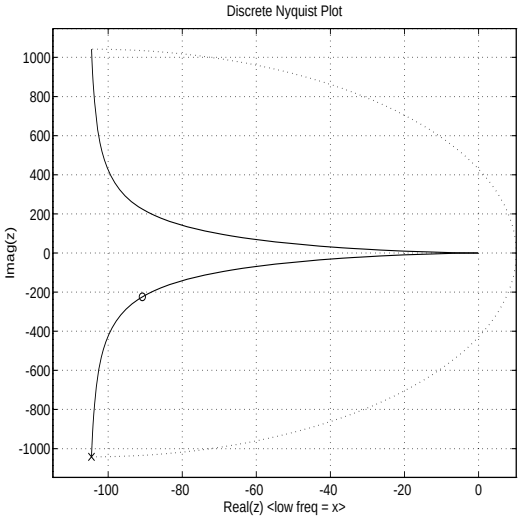


Figure 5.14 The Nyquist plot for Example 5.4.

and so the gain margin is

$$\text{GM} \;=\; 20\log\;\frac{1}{.26} = 11.7\text{ dB}.$$

Fig. 5.15 also shows a portion of the unit circle. The point at which the Nyquist plot

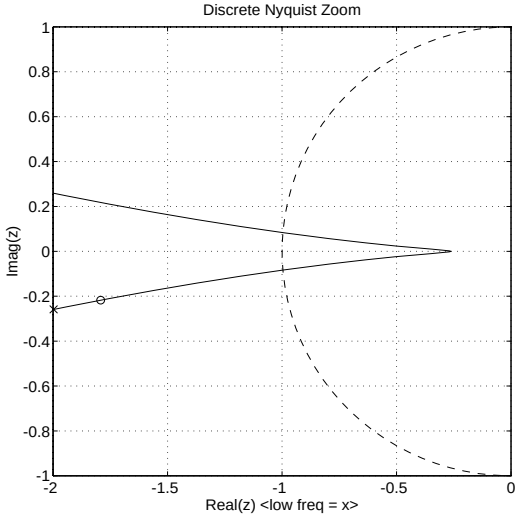


Figure 5.15 A portion of the Nyquist plot for Example 5.4 together with the unit circle (dashed line).

and the unit circle intersect is about 5 degrees away from 180, and so the phase margin is approximately 5 degrees.

EXAMPLE 5.5

Consider the following loop transfer function

$$L(z) = \frac{z(z+0.5)}{(z-1)(z-1.5)}.$$

A portion of the Nyquist plot with the unstable region indicated by shading as well as a unit circle is shown in Fig. 5.16. We can see that the real axis segment from -2 to 0

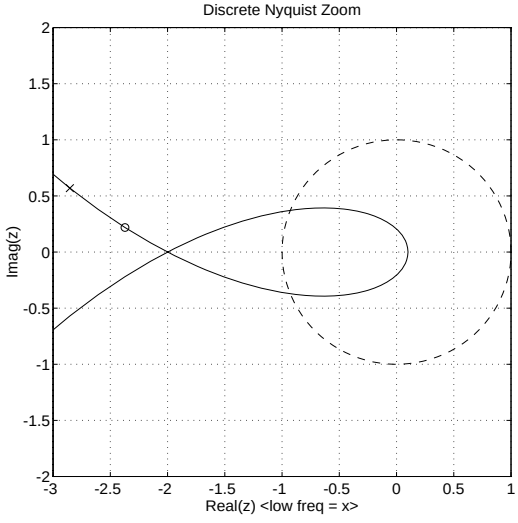


Figure 5.16 Part of Nyquist plot for Example 5.5

is part of the stable region. Thus the closed-loop system is stable, and the gain has to be *decreased* by a factor of 1/2 before the system becomes unstable. The system has an infinite upper gain margin, and the lower gain margin is

$$\text{GM} = 20 \log \frac{1}{2} = -6 \text{ dB}.$$

The intersection of the unit circle and the Nyquist plot occurs at -158° giving a phase margin of 21°.

5.4 Stability Robustness

The design of a control system begins with a mathematical model (e.g. transfer function or state-space model) of the plant to be controlled. Using the plant model, a controller is designed to get a stable closed-loop system with acceptable performance (e.g. settling time, overshoot). A generic (analog or digital) closed-loop control system is shown in Fig. 5.17.

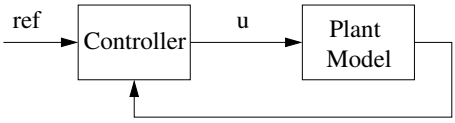


Figure 5.17 Generic (analog or digital) control system. The line coming out of the plant carries all measured signals, which could be outputs and/or state variables. The controller is a mathematical model (e.g. transfer function or state-space) that computes a signal u which is sent to the input of the plant.

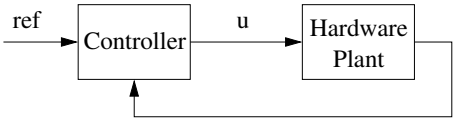


Figure 5.18 Generic (analog or digital) control system. The controller from Fig. 5.17 is connected to the actual hardware plant.

Obtaining the stable closed-loop system in Fig. 5.17 is *not* the only design goal because that system is controlling a mathematical model of the plant. It is not controlling the actual hardware plant. In Fig. 5.18 we show the controller connected to the hardware plant. We know that the closed-loop system in Fig. 5.17 is stable by design. What about the stability of the system in Fig. 5.18? The answer is that it may be stable or unstable! An important design goal is to prevent the latter possibility.

The reason that the control system in Fig. 5.18 may be unstable while the system in Fig. 5.17 is stable is that the plant model might not be an accurate representation of the hardware plant. Of course, if the plant model *is* an accurate representation of the hardware plant we would expect the control system in Fig. 5.18 to be stable.

To get an idea of the *stability robustness* of the control system, we perturb the plant model in a certain mathematical way and see how large the perturbation can be before the closed-loop system goes unstable. We know that the closed-loop system with the mathematical plant model, Fig. 5.17, is stable. If the closed-loop system remains stable when the plant model is perturbed by a "large" amount, we say that the control system has good stability robustness.

In what follows we consider two different ways, classical and modern, of perturbing the plant model. The classical plant perturbation method is used for single-input plants and is used to obtain (classical) *gain and phase margins*. The modern plant-perturbation method may be used for multiple-input systems (with single-input systems as a special case) and is used to obtain *stability robustness bounds*.

5.5 Classical Plant Perturbation Models

A simple way to perturb a single-input plant model is to multiply the plant input by a number, q, as shown in Fig. 5.19. If $q = 1$, the perturbed plant model is identical to the (nominal) plant model. Thus, the closed-loop system in Fig. 5.19 is stable when $q = 1$. Classical stability margins are a measure of how much q can differ from 1 before the closed-loop system becomes unstable. In the classical approach, q is allowed to vary in two special ways, which are presented in the following two subsections.

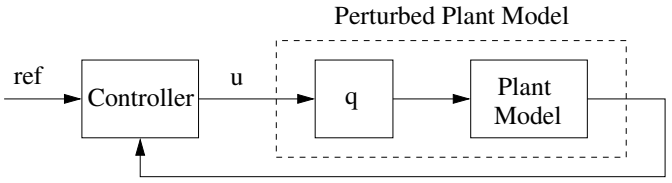


Figure 5.19 Classical plant perturbation model.

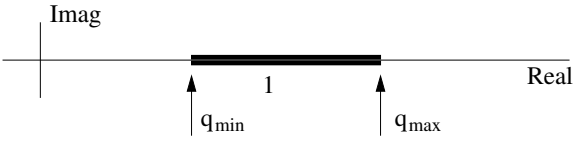


Figure 5.20 The line seqment between $q_{\min}$ and $q_{\max}$ is the range of real-valued q values for which the closed-loop system in Fig. 5.19 is stable. Note that this line segment always includes $q = 1$ because the nominal closed-loop system is stable by design.

5.5.1 Gain Perturbations

Let q be a real number, which represents a gain perturbation. The closed-lop system in Fig. 5.19 will remain stable as q is increased from 1 up to some maximum value $q_{\max}$. The system will be unstable for $q > q_{\max}$. In a similar way the closed-loop system will remain stable as q is decreased from 1 down to a minimum value $q_{\min}$. The range of real-valued q values for which the closed-loop system remains stable is shown in Fig. 5.20. The *upper and lower gain margins* are defined by converting the numbers $q_{\max}$ and $q_{\min}$ to units of decibels as follows.

$$\text{Upper Gain Margin (UGM)} = 20\log_{10}(q_{\max}) \text{ dB}$$
$$\text{Lower Gain Margin (LGM)} = 20\log_{10}(q_{\min}) \text{ dB}$$

5.5.2 Phase Perturbations

Let q be a complex number of unit magnitude, $q = e^{j\phi}$, which represents a pure phase lag of ϕ radians. When $\phi = 0$, then $q = 1$ and the closed-loop system in Fig. 5.19 is stable. The system remains stable for $|\phi| < \phi_{\max}$. The values of q for which the closed-loop system is stable are given by $q = e^{-j\phi}$, $-\phi_{\max} < \phi < \phi_{\max}$. These values are shown in Fig. 5.21.

5.6 Modern Perturbation Models

The modern approach to perturbing plant models may be used with multiple-input plants, which include single-input plants as a special case. Similar to the classical approach, the perturbation model is in cascade with the plant input. Two different perturbation models are particularly useful and are described below. Before describing these models, however, we first give a result about the stability of interconnected systems.

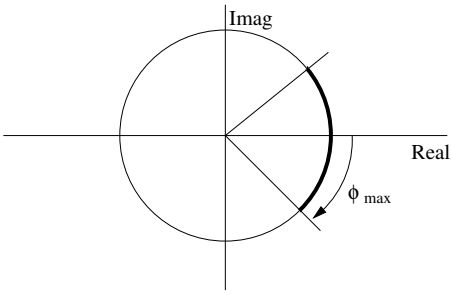


Figure 5.21 The bold arc on a circle of radius 1is the range of complex-valued q values for which the closed-loop system in Fig. 5.19 is stable. Note that this arc always includes $q = 1$ because the nominal closed-loop system is stable by design.

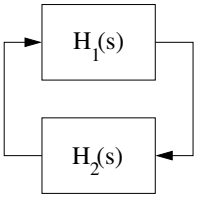


Figure 5.22 A feedback interconnection of two stable systems used to state the small-gain theorem. These systems may have single of multiple inputs.

5.6.1 Small Gain Theorem

The *small gain theorem* gives a sufficient condition for stability for the feedback interconnection of two stable systems. Consider the closed-loop system shown in Fig. 5.22, where $H_1(s)$ and $H_2(s)$ are stable systems. Note that these systems may be multiple-input. They are shown as continuous-time systems but the results described here are identical for discrete-time systems. The small gain theorem says that the closed-loop system in Fig. 5.22 is guaranteed to be stable provided the following inequality is satisfied:

$$\|\mathbf{H}_1(s)\|_\infty \|\mathbf{H}_2(s)\|_\infty < 1. \tag{5.37}$$

In this equation, the norms are the *system infinity norms* defined as follows

$$\|\mathbf{H}(s)\| \stackrel{\text{def}}{=} \sup_\omega \bar{\sigma}[\mathbf{H}(j\omega)] \tag{5.38}$$

where "sup" stands for "supremum" (which may be thought of as a maximum value) and $\sigma(\bar{\mathbf{M}})$ represents the maximum singular value of a matrix $\mathbf{M}$. So the system infinity norm may be thought of as the maximum value of the frequency response gain across all frequencies.

Note that the small gain theorem provides only a sufficient, but not necessary, condition for stability. In other words, if the inequality in (5.37) is satisfied then the closed-loop system in Fig. 5.22 is guaranteed to be stable. However, if the inequality in (5.37) is *not* satisfied, the closed-loop system in Fig. 5.22 may be stable or unstable. The failure to satisfy the inequality in (5.37) does *not* imply instability!

The small-gain theorem is used to derive stability robustness bounds for multiple-input systems using two important perturbation models. These models are described next.

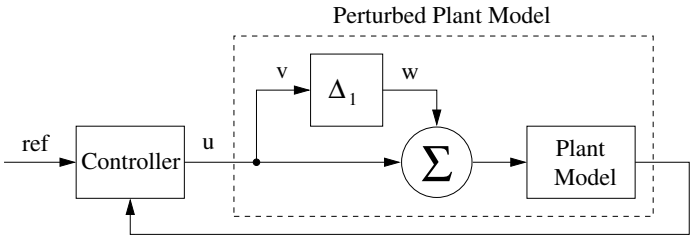


Figure 5.23 A control system with a modern input-multiplicative plant perturbation model. The system $\mathbf{\Delta}_1$ is an unknown stable system that is used to perturb the (nominal) plant model.

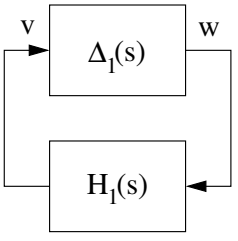


Figure 5.24 The closed-loop system shown here is identical to the closed-loop system in Fig. 5.23 when all external inputs in that figure are set to zero and $\mathbf{H}_1(s)$ is the system from $\mathbf{w}$ to $\mathbf{v}$.

5.6.2 Input-Multiplicative Plant Perturbation

Consider the *input-multiplicative plant perturbation model* shown in Fig. 5.23. Suppose that the plant is a p-input system. Then the system $\mathbf{\Delta}_1$ is p-input p-output stable system. For ease of presentation let the systems in each block of Fig. 5.23 be a continuous-time system. The $\mathbf{\Delta}(s)$ is an unknown, stable system that represents a perturbation to the nominal plant model. Note that if $\mathbf{\Delta}_1(s) = \mathbf{0}$ the perturbed plant model is identical to the (nominal) plant model. The question of *stability robustness* is the following: how "large" can the unknown system $\mathbf{\Delta}_1(s)$ become before the closed-loop system in Fig. 5.23 goes unstable? The "size" of $\mathbf{\Delta}_1(s)$ will be measured by the system infinity norm shown in (5.38). This question will be answered using the small gain theorem. Before using this theorem, the closed-loop system in Fig. 5.23 must be redrawn as the feedback interconnection of two stable systems, as shown in Fig. 5.22.

The first step in redrawing Fig. 5.23 is to set all external inputs to zero. The only such input in Fig. 5.23 is the reference input signal, and it is set to zero. If there were other external inputs such as disturbance signals they would also be set to zero. The reason these signals may be set to zero is that, for an interconnection of linear systems, additive signals inserted anywhere in a feedback loop cannot destabilize the closed-loop system. Another way to say this is that it is only the poles of the closed-loop system that determine its stability and additive signals do not change the pole locations of a feedback loop. The second step is to obtain a model for the system from $\mathbf{w}$ to $\mathbf{v}$ in Fig. 5.23. Call this system $\mathbf{H}_1(s)$. Note that $\mathbf{H}_1(s)$ is a stable system because it has the same poles as the nominal closed-loop control system that is stable by design. The unknown system $\mathbf{\Delta}(s)$ is stable by assumption. With this definition of $\mathbf{H}_1(s)$, the system in Fig. 5.23 may be redrawn as Fig. 5.24. The small-gain theorem says that the stability of the closed-loop system in Fig. 5.24 is guaranteed if the product of the system infinity norms is less than 1 (see (5.37)).

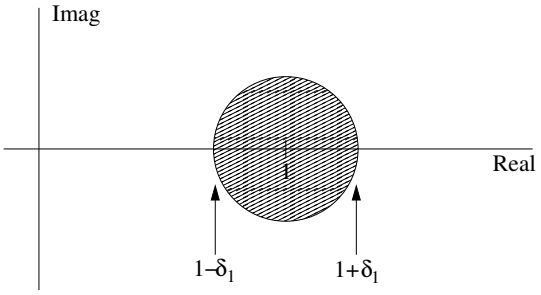


Figure 5.25 The shaded region is a circle of radius δ_1 centered at 1. It represents the set of points $1 + c$ for which the closed-loop system in Fig. 5.23 is stable when $\mathbf{\Delta}_1(s) = c$.

This inequality may be rewritten to obtain the following bound on the size of the unknown perturbation system:

$$\|\mathbf{\Delta}_1(s)\|_\infty < \frac{1}{\|\mathbf{H}_1(s)\|_\infty}. \tag{5.39}$$

The right-hand side of this inequality is defined to be the number δ_1,

$$\delta_1 \stackrel{\text{def}}{=} \frac{1}{\|\mathbf{H}_1(s)\|_\infty}. \tag{5.40}$$

The number δ_1 is the *input-multiplicative stability robustness bound*. The larger this number is, the more tolerant the control system is to errors in the plant model. As a rule of thumb we would like to have a control system for which $\delta_1 > 0.5$. For most useful control systems (e.g. systems in which there is a pole at $s = 0$ in either the plant or controller) δ_1 cannot be larger than 1.

5.6.3 Relationship Between Modern Stability Robustness Bounds and Classical Stability Margins

The stability robustness of control systems with a single-input plant may be assessed using classical gain and phase margins or by using the stability robustness bound δ_1. In this subsection we derive the relationship between these different robustness measures. Note that the classical plant perturbation model in Fig. 5.19 and the modern plant perturbation model in Fig. 5.23 both have a perturbation model in series with the plant model. In the classical case the perturbation is simply a number q. Consider Fig. 5.23 for single-input plants. In this case, the unknown system $\mathbf{\Delta}_1(s)$ is a single-input single-output system. Consider the special case in which $\mathbf{\Delta}_1(s)$ is simply a gain of c, where c is an unknown number. In this case, the signal u in Fig. 5.23 gets multiplied by $1 + c$ before entering the plant model. Thus, the multiplicative perturbation is $1 + c$. We now ask, what are the largest and smallest values of $1 + c$ for which the stability of the closed-loop system can be guaranteed? The answer to this question uses the fact that the closed-loop system is guaranteed to be stable when the system infinity norm of $\mathbf{\Delta}_1(s)$ is less than δ_1. The definition of system infinity norm for any system is given in (5.38). When $\mathbf{\Delta}_1(s)$ is simply a complex number c, this definition reduces to the magnitude of c. Given the condition that $|c| < \delta_1$, the values of $1 + c$ are contained in a circle of radius δ_1 centered at 1 in the complex plane. This region is shown in Fig. 5.25. The closed-loop control system is guaranteed to be stable for all values of the multiplicative perturbation $1 + c$ lying in

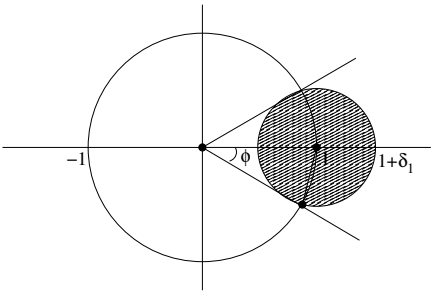


Figure 5.26 The shaded region of Fig. 5.25 is shown here along with a circle of radius 1. The points on the arc of the unit circle for which the closed-loop system in Fig. 5.23 is stable lie between the angles $-\phi$ to ϕ.

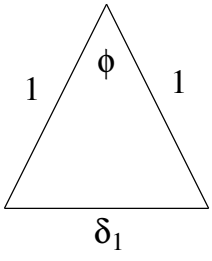


Figure 5.27 The triangle defined by the three bold points in Fig. 5.26.

the shaded region. Note that the shaded region contains both real and complex-valued numbers. The real-axis points in this region lie in the interval from $1 - \delta_1$ to $1 + \delta_1$. This interval may be compared with the classical gain margin interval shown in Fig. 5.20. By definition, the classical gain margin interval is the largest possible interval of real-valued gain perturbations. Thus, the real-valued interval in Fig. 5.25 must be contained within the interval in Fig. 5.20. This gives the following two inequalities:

$$\begin{aligned} q_{\max} &\geq 1 + \delta_1 \\ q_{\min} &\leq 1 - \delta_1 \end{aligned} \tag{5.41}$$

The shaded region in Fig. 5.25 is redrawn on a complex plane in Fig. 5.26 along with the classical phase-margin circle from Fig. 5.21. The shaded region in this figure may be compared with the arc of the circle in Fig. 5.21 defining the classical phase margin. By definition, the classical phase margin arc is the largest arc of a unit circle for which the closed-loop system will remain stable. Thus, the arc on the unit circle in Fig. 5.26 must be contained within the arc in Fig. 5.21. The result is that $\phi_{\max}$ in Fig. 5.21 must be greater than or equal to ϕ in Fig. 5.26. Also, there is a relationship between ϕ and δ_1 in Fig. 5.26. In order to derive this relationship, consider the triangle formed by the three bold points in Fig. 5.26. These points form a triangle, which is rotated and shown in Fig. 5.27. A perpendicular bisector drawn through the angle ϕ creates two right triangles, each of whom have a base of $\delta_1/2$ and an angle of $\phi/2$. The triangle shows that $\phi/2$ is the angle whose sin is $\delta_1/2$. Thus, $\phi = 2\sin^{-1}(\delta_1/2)$. The result is that

$$\phi_{\max} \geq 2\sin^{-1}(\delta_1/2). \tag{5.42}$$

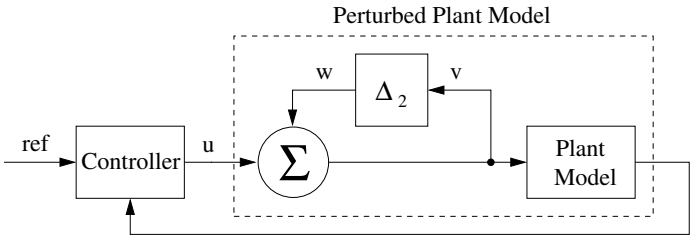


Figure 5.28 A control system with a modern input-feedback plant perturbation model. The system $\mathbf{\Delta}_2$ is an unknown stable system that is used to perturb the (nominal) plant model.

5.6.4 Input-Feedback Plant Perturbation

Another useful plant perturbation model is shown in Fig. 5.28. This figure can be used for both continuous- and discrete-time systems. For ease of presentation we consider continuous-time systems. If the plant model has p inputs, then $\mathbf{\Delta}_2(s)$ is a p-input, p-output unknown system that is assumed to be stable. Let all external inputs be set to zero and let the system from $\mathbf{w}$ to $\mathbf{v}$ in Fig. 5.28 be called $\mathbf{H}_2(s)$. The the system in Fig. 5.28 may be redrawn to look like the system in Fig. 5.24, with $\mathbf{H}_1(s)$ replaced by $\mathbf{H}_2(s)$ and $\mathbf{\Delta}_1(s)$ is replaced by $\mathbf{\Delta}_2(s)$. Using the small-gain theorem, an upper bound on the size of the unknown perturbation $\mathbf{\Delta}_2(s)$ in Fig. 5.28 is given by

$$\|\mathbf{\Delta}_2(s)\|_\infty < \delta_2 \stackrel{\text{def}}{=} \frac{1}{\|\mathbf{H}_2(s)\|_\infty}. \tag{5.43}$$

The number δ_2 is called the *input-feedback stability robustness bound.* The relationship between δ_2 and the classical gain and phase margins may be derived in a similar manner as was done for δ_1. The result is that

$$\begin{aligned} q_{\max} &\geq \frac{1}{1-\delta_2} \\ q_{\min} &\leq \frac{1}{1+\delta_2} \\ \phi_{\max} &\geq 2\sin^{-1}(\delta_2/2). \end{aligned} \tag{5.44}$$

Note that the inequalities in (5.41), (5.42), and (5.44) must be simultaneously satisfied. Thus, combining these inequalities we have

$$\begin{aligned} q_{\max} &\geq \max[1+\delta_1, \frac{1}{1-\delta_2}] \\ q_{\min} &\leq \min[1-\delta_1, \frac{1}{1+\delta_2}] \\ \phi_{\max} &\geq \max[2\sin^{-1}(\delta_1/2), 2\sin^{-1}(\delta_2/2)]. \end{aligned} \tag{5.45}$$

5.7 Scaling a Transfer Function

The step response can be used to measure the speed of response of a control system. This measure is called the *settling time*, and it is defined as the time T_S at which the step response enters and stays within 1% of the reference input. Two examples are given in Fig. 5.29.

A control system with a small value of T_S is said to be "fast" while a system with a large value of T_S is said to be "slow." Thus bandwidth and settling time are two different measures of the speed of response of a control system.

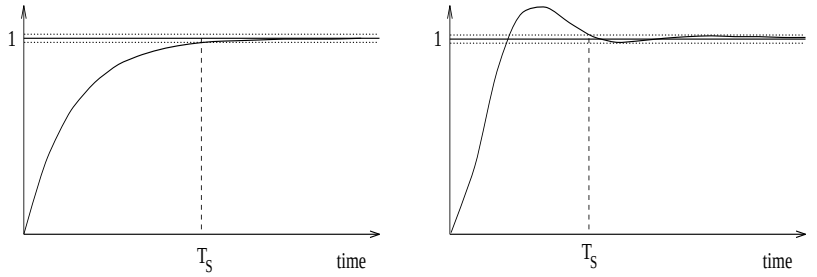


Figure 5.29 Two different step responses with settling times indicated.

Settling time is often used as a specification for the speed of response of a control system. In order to achieve a desired settling time, the closed-loop control system must have poles at appropriate locations. For small settling times, the poles must be far into the left half-plane. For large settling times, the poles can be closer to the $j\omega$ axis. These relationships can be made precise using *prototype transfer functions with scaling*.

A class of prototype transfer functions is introduced in Chapter 6. The step responses of these transfer functions all have a settling time of 1 second. If some other value of settling time is desired, a prototype system can be scaled to achieve the desired value. Scaling can be viewed in the time domain or in the s-domain, as we now discuss.

Suppose $y(t)$ is a step response with a settling time of 1 second. Then the scaled step response $y'_S(t)$ defined by

$$y'_S(t) \stackrel{\text{def}}{=} y(\frac{t}{T_S}) \tag{5.46}$$

has a settling time of T_S seconds (note that $y'_S(T_S) = y(1)$). Thus a step response having a desired settling time of T_S seconds can be obtained by time-domain scaling a prototype step response with a 1 second settling time.

To view the scaling procedure in the s-domain, we take the Laplace transform of both sides of (5.46). Let $H'(s)$ and $H(s)$ represent the transfer functions of the scaled and normalized systems, respectively. Then the Laplace transforms of the step responses in (5.46) can be written using the time scaling property of Laplace transforms (see Table 3.1) as

$$H'(s) = T_S\, H(sT_s). \tag{5.47}$$

The above equation gives the relationship between the poles of the original and scaled systems. Namely, if s_1 is a pole of the original system $H(s)$, then

$$s'_1 = s_1/T_S \tag{5.48}$$

is a pole of the scaled system $H'(s)$. Equation (5.48) shows that the poles of the scaled system are obtained by dividing the poles of the prototype system (real and imaginary parts) by the desired settling time. If $T_S > 1$, the poles of the scaled system will be closer to the $j\omega$ axis than the poles of the prototype transfer function. Conversely if $T_S < 1$ the poles of the scaled system will be farther into the left half-plane than those of the prototype system. This corresponds to our intuition about "fast" and "slow" poles.

EXAMPLE 5.6

Consider a 2nd-order normalized Bessel transfer function from Chapter 6. The poles of this system are given in Table 6.2 on page 257 to be

$$s_{1,2} = -4.0530 \pm j2.3400.$$

The denominator polynomial for this system can be calculated by multiplying out these roots to obtain

$$a(s) = (s - s_1)(s - s_2) = s^2 + 8.1060s + 21.9024.$$

The numerator polynomial is taken to be the constant term of $a(s)$ so that the transfer function unity dc gain

$$H(s) = \frac{21.9024}{s^2 + 8.1060s + 21.9024}.$$

The step response and Bode magnitude plot for this system are shown in Fig. 5.30. It can be seen that the step response has a settling time of 1 second, as promised by Table 6.2. The Bode plot shows that the 3 dB bandwidth is about 3.5 radians/second.

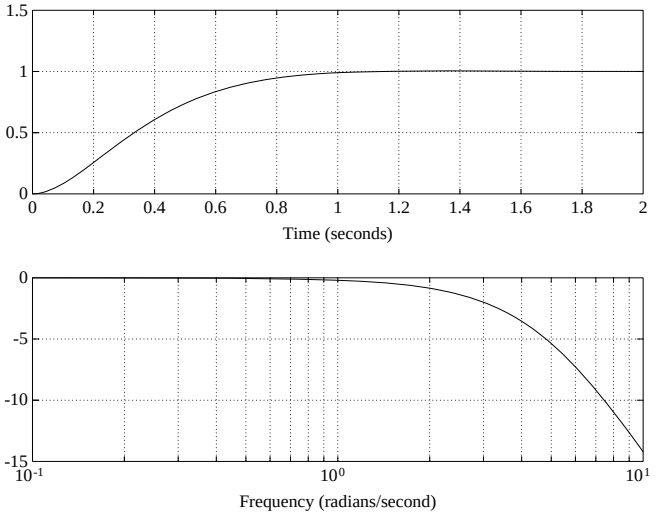


Figure 5.30 Step response and Bode magnitude plot for system in Example 5.6.

Suppose we want to scale this system to obtain a new system with a settling time of 0.5 seconds. Using (5.48) we see that the poles of the new system are obtained by dividing the poles of the original system by 0.5. The result is

$$s'_{1,2} = s_{1,2}/0.5 = -8.1060 \pm j4.6800.$$

The denominator polynomial for the scaled system is

$$a'(s) = (s - s'_1)(s - s'_2) = s^2 + 16.2120s + 87.6096$$

and the scaled transfer function is

$$H'(s) = \frac{87.6096}{s^2 + 16.2120s + 87.6096}$$

The step response and Bode magnitude plot for the scaled system are shown in Fig. 5.31. It can be seen that the step response has a settling time of 0.5 seconds. The Bode plot shows that the 3 dB bandwidth is about 7 radians/second, which is twice the bandwidth of the original system.

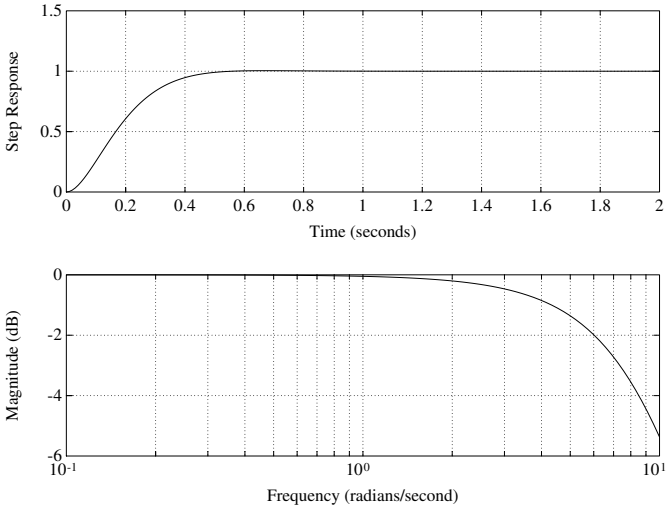


Figure 5.31 Step response and Bode magnitude plot for the scaled system in Example 5.6.

■

5.8 Steady-State Error and System Type

So far in this chapter we have discussed the following aspects of the closed-loop control system shown in Fig. 5.6: stability of the nominal system, stability margins, and speed of response. Another important measure of control system performance is discussed in this section, namely steady-state error.

Let the input sequence to a digital control system be denoted $r[k]$, and the output $y[k]$. We define the *tracking error* to be the difference

$$e[k] \stackrel{\text{def}}{=} r[k] - y[k].$$

The *steady-state error*, denoted SSE, is defined to be the limiting value of the tracking error

$$\text{SSE} = \lim_{k \to \infty} e[k]. \tag{5.49}$$

Note that SSE is only defined when the limit in (5.49) exists.

In order to compute the steady-state error associated with a given control system, we must first specify the input $r[k]$. It is customary to consider the following three input signals, shown below with their z-transforms. In each case, the symbol 'A' is a positive

amplitude parameter.

$$\begin{aligned}
&\text{Step:} && r_0[k] = A, && R_0(z) = \frac{Az}{z-1} \\
&\text{Ramp:} && Ar_1[k] = Ak, && R_1(z) = \frac{Az}{(z-1)^2} \\
&\text{Parabola:} && Ar_2[k] = Ak^2, && R_2(z) = \frac{Az(z+1)}{(z-1)^3}.
\end{aligned}$$

Inputs of the form k^m are used to define the "type" of a system.

Definition 5.1 *A control system is said to be* **Type m** *if it can track a reference input of the form $r_m[k] = k^m$ with finite but non-zero steady-state error.*

The steady-state tracking error is easily computed using z-transforms. $E(z)$, the z-transform of the tracking error for the control system of Fig. 5.6, is derived as follows:

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

It will be seen that the system type 'm' is equal to the order of the pole of $L(z)$ at $z = 1$. In anticipation of this result, we express $L(z)$ as

$$L(z) = \frac{L'(z)}{(z-1)^q}$$

where q is the number of poles of $L(z)$ at $z = 1$. Note that if $q = 0$ then $L'(z) = L(z)$.

The steady-state error is evaluated using the final value theorem from Table 3.3 on page 86. We assume that the poles of the closed-loop system are inside the unit circle.

Step Inputs For a step input with amplitude A, the steady-state error is

$$E(z) = \frac{1}{1+KL(z)} \cdot \frac{Az}{(z-1)}$$

and using the final value theorem for $E(z)$ results in

$$\begin{aligned}
\text{SSE} &= \lim_{z\to 1}(z-1)\frac{1}{1+KL(z)}\frac{Az}{(z-1)} \\
&= \lim_{z\to 1}\frac{Az}{1+KL(z)} \\
&= \lim_{z\to 1}\frac{A}{1+KL'(1)(z-1)^{-q}} \\
&= \begin{cases} \dfrac{A}{1+KL(1)}\,, & q = 0 \\ 0\,, & q \geq 1. \end{cases}
\end{aligned}$$

If the loop transfer function $L(z)$ does not have a pole at $z = 1$, then $L(1) = L'(1)$ is a finite number and SSE is a finite non-zero number. From Definition 5.1 this means that the control system is Type 0. In other words, a control system is Type 0 if and only if the loop transfer function $L(z)$ does not have a pole at $z = 1$. On the other hand, if $q \geq 1$ (i.e. $L(z)$ has at least one pole at $z = 1$), then the steady-state error is zero for a step input.

Ramp Input For a ramp input, the steady-state error is

$$E(z) = \frac{1}{1 + KL(z)} \cdot \frac{Az}{(z-1)^2}$$

and using the final value theorem for $E(z)$ results in

$$\begin{aligned}
\text{SSE} &= \lim_{z \to 1} (z-1) \frac{1}{1 + KL(z)} \frac{Az}{(z-1)^2} \\
&= \lim_{z \to 1} \frac{Az}{(z-1) + \dfrac{KL'(z)}{(z-1)^{q-1}}} \\
&= \lim_{z \to 1} \frac{A}{KL'(1)(z-1)^{1-q}} \\
&= \begin{cases} \infty\,, & q = 0 \\ \dfrac{A}{KL'(1)}\,, & q = 1 \\ 0\,, & q \geq 2. \end{cases}
\end{aligned}$$

If the loop transfer function $L(z)$ does not have a pole at $z = 1$ the steady-state error to a ramp input is infinite. If $L(z)$ has a single pole at $z = 1$, then SSE is finite but non-zero (i.e. $q = 1$ implies a Type-1 system). Finally, if $L(z)$ has more than one pole at $z = 1$ then the steady-state error for a ramp input is zero.

EXAMPLE 5.7

Consider the digital control system shown in Fig. 5.32. The loop transfer function has

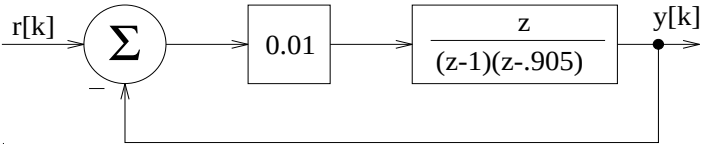


Figure 5.32 A closed-loop control system for Example 5.7.

a single pole at $z = 1$ (i.e. $q = 1$) and so the steady-state error for a step input should be zero. For a ramp input $r[k] = 0.1k$, the steady-state error should be

$$\text{SSE} = \frac{0.1}{KL'(1)}$$

where

$$L'(1) = \frac{0.01}{(1 - 0.905)} \approx 0.105.$$

Thus the steady-state error for a ramp input should be SSE= $0.1/0.105 = 0.95$. The calculated steady-state errors are confirmed by the simulation results in Fig. 5.33.

■

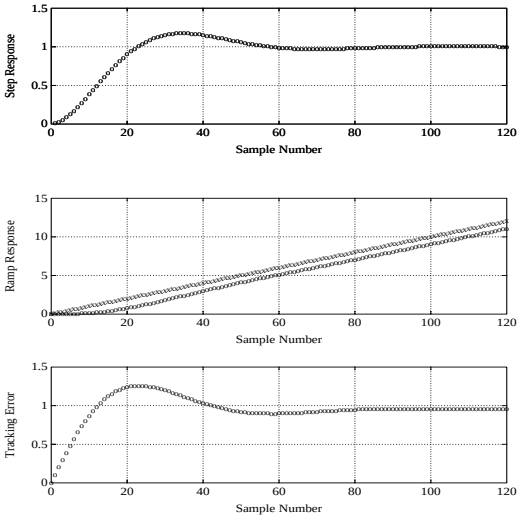


Figure 5.33 (a)Step response for Example 5.7. (b) Ramp response. (c) Tracking error corresponding to the ramp response.

Parabolic Input A similar derivation gives the following result for parabolic inputs

$$\text{SSE} = \begin{cases} \infty\,, & q = 0, 1 \\ \dfrac{2A}{KL'(1)}\,, & q = 2 \\ 0\,, & q \geq 3. \end{cases}$$

The steady-state error expressions for step, ramp, and parabolic inputs confirm the fact that a control system is Type-m if and only if the loop transfer function has m poles at $z = 1$.

5.9 Chapter Summary

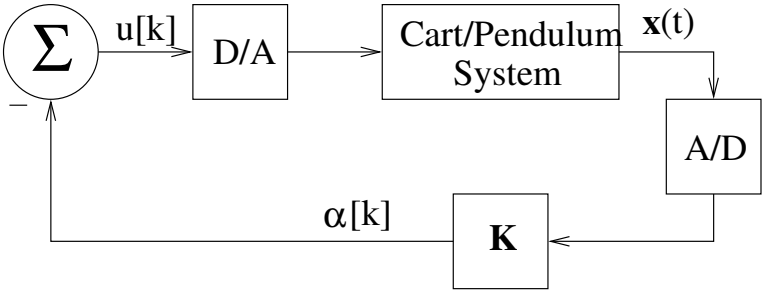
In this chapter we presented frequency-domain tools for analyzing the behavior of linear control systems. One way to display the frequency response of a system is to plot the magnitude and phase of the transfer function separately versus frequency – these are called Bode plots. The Bode magnitude plot reveals the bandwidth of a control system. Bandwidth, which is a measure of the speed of response of a system, is defined as the frequency at which the Bode magnitude plot is 3 dB down from its value at zero frequency.

We introduced Nyquist plots and showed how they can be used to find the stability margins of a control system. Stability margins are a measure of how much a plant model can be perturbed before the closed-loop system becomes unstable. If the stability margins are small, then the mathematical model of the plant will have to be a very accurate representation of the hardware in order for the actual closed-loop plant to be stable. On the other hand, if the stability margins are large, then the actual closed-loop system will be stable even if the mathematical model of the plant does not perfectly represent the hardware. These ideas are made precise by the definitions of gain and phase margins. The

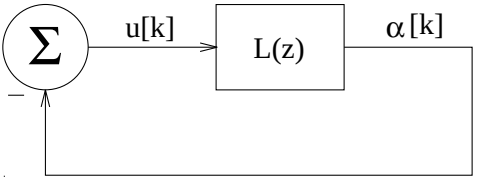
stability margins developed in this chapter are used in future chapters to assess the stability of state-space control systems.

5.10 Problems

1. The cart/pendulum system described in Chapters 1 and 3 can be controlled by feeding back a linear combination of the state variables (see the discussion in Section 1.3). A block diagram of the digital regulation system is shown below



In this figure, $\mathbf{K}$ is a row vector consisting of 4 numbers. A procedure for calculating an $\mathbf{K}$ vector which gives good regulator performance is given in Chapter 6. The system from $u[k]$ to $\alpha[k]$ in the above figure is a discrete-time system. This discrete-time system can be calculated from the continuous-time system $(\mathbf{A}, \mathbf{b})$ (and the vector $\mathbf{K}$) using a procedure given in Chapter 4. The result for the inverted pendulum system with a sampling interval of 0.01 seconds is shown below



where

$$L(z) = -0.0091 \frac{z^3 - 5.4606z^2 + 7.7620z - 3.3035}{z^4 - 3.7811z^3 + 5.3405z^2 - 3.3382z + 0.7788}.$$

(a) Obtain a Nyquist plot of $L(z)$ and find the gain and phase margins for this regulator.

(b) Obtain a Bode plot of $L(z)$ and explain why it is difficult to find the gain and phase margins from this plot.

2. Consider the following loop transfer functions. Assume that $K = 1$, and find the gain and phase margins from a Nyquist plot of $L(z)$.

(a)

$$L(z) = 0.5 \frac{z + 1.5}{z(z - 1)}.$$

(b)

$$L(z) = \frac{z - 0.8}{z^2 - 2z + 0.99}$$

(c)

$$L(z) = 20\frac{z^2 - 1.77z + 0.82}{z^2 - z - 0.75}$$

(d)

$$L(z) = 10\frac{z^2 - 1.25z + 0.375}{z^2 - z - 0.75}$$

3. Find the gain and phase margins for the system shown in Example 5.7.

4. *Choosing the sampling interval for discretizing an analog compensator*

Problem 1 of Chapter 4 showed that the ZOH equivalent of $C(s)$ could be used as a digital approximation to an analog compensator. In this problem we give a rule-or-thumb for choosing the sampling interval to use in the discretization.

It can be shown that a digital approximation of an analog system introduces a time delay of about $T/2$ seconds, where T is the sampling interval. This time delay is equivalent to a phase lag of $\omega T/2$ radians at any frequency ω radians per second. In order for the digital approximation to work properly it is important that the phase lag it introduces is much less than the phase margin of the analog control system. The rule-of-thumb given in the next paragraph says that the phase lag should be less than 20% of the phase margin.

The phase margin of an analog control system can be obtained from an analog Nyquist plot of the loop transfer function $C(s)G(s)$. (The program **anyq** in the Digital Control Toolbox produces analog Nyquist plots.) The phase margin (PM) is obtained by finding the intersection of the Nyquist plot with the unit circle. The program **anyq** also gives the radian frequency ω_0 at which this intersection occurs. A rule-of-thumb for choosing T is:

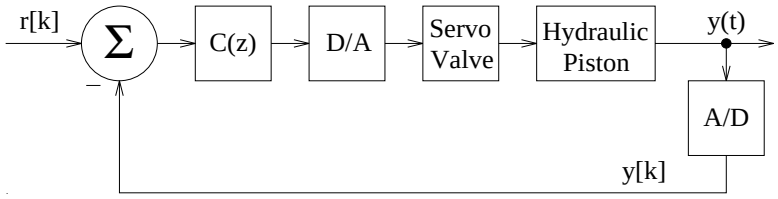
$$\frac{\omega_0 T}{2} \leq \frac{PM(radians)}{5}$$

which is approximately the same as

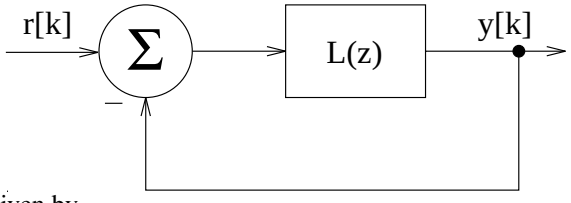
$$T \leq 0.007\frac{PM(degrees)}{\omega_0}.$$

Use **anyq** to get a Nyquist plot of $C(s)G(s)$ from Problem 4.1. Extract the phase margin and ω_0, the frequency at which the Nyquist plot intersects the unit circle. Use the rule-of-thumb to show that a sampling interval of $T = 0.1$ seconds is acceptable for discretizing $C(s)$.

5. Consider the hydraulic positioning system shown below



Such a control system can be designed using procedures given in Chapters 6 and 8, and a discrete-time model can be obtained using results from Chapter 4. The discrete-time model for a certain hydraulic positioning system is shown below



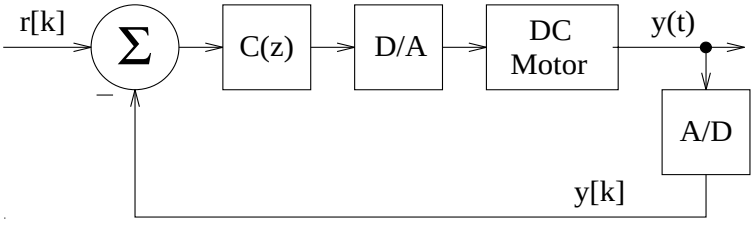
where $L(z)$ is given by

$$0.00048\frac{z^6+7.1372z^5-15.8014z^4-0.0263z^3+12.8628z^2-4.7458z-0.5421}{z^7-5.3258z^6+12.1880z^5-15.5376z^4+11.9205z^3-5.5092z^2+1.4232z-0.1591}.$$

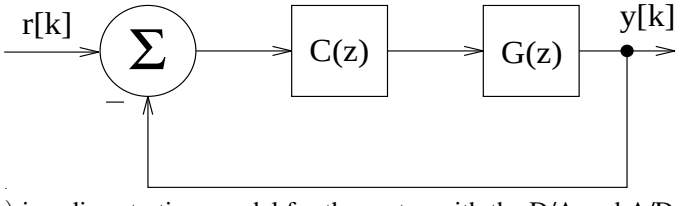
(a) Obtain a Nyquist plot and determine the gain and phase margins of the system.

(b) Calculate the steady-state error to a step input, and to a ramp input.

6. A digital position-control system for a dc motor is shown below.



A discrete-time model is shown below



where $G(z)$ is a discrete-time model for the motor with the D/A and A/D converters at a sampling interval of 0.1 seconds. For a certain motor control system we have

$$G(z)=0.00484\frac{z+0.9672}{z^2-1.9048z+0.9048}\ ,\quad C(z)=107.9568\frac{z-0.7050}{z+0.3390}.$$

(a) Obtain a Nyquist plot and determine the gain and phase margins of the system.

(b) Calculate the steady-state error to a step input, and to a ramp input.

(c) Now suppose that the digital compensator introduces a *1 sample time delay*. This time delay can be modeled by multiplying the denominator polynomial of $C(z)$ by z. Obtain a Nyquist plot for the system with time delay and investigate its stability.

CHAPTER 6

DESIGNING STATE-FEEDBACK REGULATORS

The control problem considered in this chapter is the regulation problem, which is to keep all the state variables (and thus the output) at their equilibrium value (zero) in the face of disturbances acting on the plant. Problems such as balancing an inverted pendulum, keeping an airplane in level flight, or keeping a chemical reaction at a constant temperature are all regulation problems. We examine the regulation problem in detail for two reasons. First, many important control problems are regulation problems, as mentioned above. Second, the more general problem of tracking reference inputs can be solved in part by using an appropriate regulator. Tracking systems designed using state-space techniques require that an associated regulation problem be solved. The design of tracking systems is considered in Chapter 8.

In this book, the primary tool for designing regulators is pole placement by state feedback. It will be shown that if the plant satisfies a certain mathematical property (controllability), then the poles of the closed-loop system can be placed in arbitrary desired locations by computing the input to the plant as a constant linear combination of the state variables. Pole placement can be achieved for a system of any order, and the calculation of the feedback gains reduces to the solution of a system of linear equations.

Pole placement can be compared to the root-locus classical design tool. The root locus can be used to calculate the value of a single gain in order to place the closed-loop poles. However, the closed-loop poles cannot be placed in arbitrary locations; they must lie on the root locus. This is to be contrasted with pole placement by state feedback which allows for the calculation of n gains (for an nth-order system) to exactly place all the poles in arbitrarily specified locations.

It turns out that the stability margins for a state-feedback regulator depend on the closed-loop pole locations. One drawback of the pole-placement design procedure is that the designer must pick pole locations without any guarantee on the stability margins of the resulting regulator. Once the regulator is designed these stability margins must be checked (using a Nyquist plot, for example). If the stability margins are not adequate, the designer must pick a new set of closed-loop pole locations and design another regulator.

In this chapter, the concepts of controllability and pole placement by state feedback are developed in discrete-time.

6.1 Controllability

The key concept for the development of control systems using state-space methods is that of controllability. (Other treatments use the term "reachability" instead of controllability.)

> **Definition:** The state equation $\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k]$ is said to be *controllable* if it is possible to find an input sequence $u[k]$ which takes the system from an arbitrary initial state $\mathbf{x}[0] = \mathbf{z}_1$ to an arbitrary final state $\mathbf{x}[j] = \mathbf{z}_2$ for some finite j.

In other words, a controllable system is one in which it is possible to transfer the system state from an arbitrary initial state to an arbitrary final state in finite time. If a system did *not* have this property, then there would exist some states that were inaccessible in the sense that, for some initial state, no possible combination of input values could transfer the state vector to the inaccessible states. Such a system would clearly be difficult to control, especially if the control objective required the system to reach an inaccessible state!

In order to determine whether or not a given system is controllable, recall the solution to the discrete-time state equations derived in Chapter 3:

$$\mathbf{x}[j] = \mathbf{\Phi}^j\mathbf{x}[0] + \sum_{i=0}^{j-1} \mathbf{\Phi}^{j-i-1}\mathbf{\Gamma}\, u[i]. \tag{6.1}$$

This equation expresses $\mathbf{x}[j]$ in terms of $\mathbf{x}[0]$ and the input sequence. The system is controllable if we can choose $\mathbf{x}[0]$ and $\mathbf{x}[j]$ arbitrarily and find an input sequence $\{u[0], u[1], \cdots, u[j-1]\}$ which moves the system from $\mathbf{x}[0]$ to $\mathbf{x}[j]$. Note that the definition of controllability did not specify how long it should take to make the move. The only requirement is that the time j be finite. However, in order to use (6.1), we need to specify a value for j. To start with, we will choose $j = n$ where n is the order of the system.

When $j = n$ we can write

$$\mathbf{x}[n] = \mathbf{\Phi}^n\mathbf{x}[0] + \mathbf{\Phi}^{n-1}\mathbf{\Gamma}u[1] + \mathbf{\Phi}^{n-2}\mathbf{\Gamma}u(2) + \cdots + \mathbf{\Gamma}u[n-1]. \tag{6.2}$$

We want to write the above equation in matrix form. This is easily done by recalling that multiplication of a matrix times a vector can be expressed as a linear combination of the columns of the matrix. That is, if

$$\mathbf{A} = \begin{bmatrix} | & | & | & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \vdots & \mathbf{a}_n \\ | & | & | & | \end{bmatrix}, \text{ and } \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \tag{6.3}$$

then the product $\mathbf{Ax}$ can be represented as a weighted sum of columns of $\mathbf{A}$

$$\mathbf{Ax} = \sum_{i=1}^{n} \mathbf{a}_i x_i. \tag{6.4}$$

Note that the i^{th} column of $\mathbf{A}$ is weighted by the i^{th} element of the vector $\mathbf{x}$.

If we now return to (6.2) we see that it is written as a weighted sum of columns. We can collect the weighting coefficients into a vector and put the corresponding columns into a matrix to write (6.2) in matrix form. The result is

$$\begin{bmatrix} x_1[n] \\ \vdots \\ x_n[n] \end{bmatrix} - \mathbf{\Phi}^n \begin{bmatrix} x_1[0] \\ \vdots \\ x_n[0] \end{bmatrix} = \begin{bmatrix} \mathbf{\Gamma} & \mathbf{\Phi\Gamma} & \cdots & \mathbf{\Phi}^{n-1}\mathbf{\Gamma} \end{bmatrix} \begin{bmatrix} u[n-1] \\ \vdots \\ u[0] \end{bmatrix} \tag{6.5}$$

or

$$\mathbf{z} = \mathbf{W}_c \mathbf{u} \tag{6.6}$$

where the matrix

$$\mathbf{W}_c = \begin{bmatrix} \mathbf{\Gamma} & \mathbf{\Phi\Gamma} & \cdots & \mathbf{\Phi}^{n-1}\mathbf{\Gamma} \end{bmatrix} \tag{6.7}$$

is called the *controllability matrix* for $(\mathbf{\Phi}, \mathbf{\Gamma})$.

If the initial state and final state vectors are arbitrary, then the vector $\mathbf{z}$ in (6.6) is also arbitrary. The system will be controllable if and only if we can find a vector $\mathbf{u}$ of input values which will satisfy (6.6) for an arbitrary vector $\mathbf{z}$. If the matrix $\mathbf{W}_c$ has an inverse, then we can find such an input sequence, and it is given by

$$\mathbf{u} = \mathbf{W}_c^{-1}\mathbf{z}. \tag{6.8}$$

Recall that an $n \times n$ matrix has an inverse (is nonsingular, or has rank n) if and only if its determinant is different from zero. If the rank of the controllability matrix is less than n (that is, if the determinant of $\mathbf{W}_c$ is 0), the system is said to be *uncontrollable*.

At this point, the reader may object to the choice of $j = n$ in the development of controllability. Recall that the definition of controllability required the system to go from one state to another in some finite time j, but there was no other requirement on j. In the above development, we took $j = n$ where n is the order of the system, and showed that if the rank of $\mathbf{W}_c$ is less than n, then there does not exist an input sequence of length $j = n$ which will move the system to the desired final state. It seems reasonable to ask whether or not input sequences of length *greater than* n might be able to move the system to the desired final state. The answer to this question is as follows: if an input sequence of length n cannot be chosen to move an n^{th}-order system to a desired final state, then *no sequence of length greater than* n will be able to move the system to the desired final state either.

This surprising result is established as follows. If (6.5) is written for $j > n$, the resulting extended controllability matrix will have more than n columns as shown below

$$\mathbf{z} = \begin{bmatrix} \mathbf{\Gamma} & \mathbf{\Phi\Gamma} & \cdots & \mathbf{\Phi}^{j-1}\mathbf{\Gamma} \end{bmatrix} \begin{bmatrix} u[j-1] \\ \vdots \\ u[0] \end{bmatrix}, \quad j > n. \tag{6.9}$$

The additional columns in the extended controllability matrix which are not in the controllability matrix shown in (6.7) are

$$\mathbf{\Phi}^n\mathbf{\Gamma}\,,\ \ \mathbf{\Phi}^{n+1}\mathbf{\Gamma}\,,\ \ \cdots\,,\ \ \mathbf{\Phi}^{j-1}\mathbf{\Gamma}. \tag{6.10}$$

From the corollary to the Cayley-Hamilton theorem presented in Chapter 4, the matrix $\mathbf{\Phi}$ raised to a power greater than or equal to n can be expressed as a linear combination of the matrices $\mathbf{\Phi}^{n-1}, \cdots, \mathbf{I}$. This means that the additional columns shown in (6.10) will be linearly dependent on the columns shown in (6.7), and so they cannot be used to obtain a vector $\mathbf{z}$ in (6.9) unless it can already be achieved with the n columns shown in (6.7). In other words, the set of final states that can be reached from a given initial state in n time steps is not made any larger when $j > n$ because the additional columns that are added to the controllability matrix are linearly dependent on the first n columns. We also note that if $j < n$ we can't solve for $\mathbf{u}$ from (6.6) (overdetermined, inconsistent equations). Thus we have established the following theorem.

Theorem 6.1 *The n^{th} order system $\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k]$ is controllable if and only if the controllability matrix $\mathbf{W}_c = [\mathbf{\Gamma}\ \mathbf{\Phi}\mathbf{\Gamma}\ \cdots\ \mathbf{\Phi}^{n-1}\mathbf{\Gamma}]$ is of rank n.*

An important consequence of the above development is the following: **an n^{th}-order controllable system can go from its current state to any other state in n time steps**.

EXAMPLE 6.1

We want to test whether or not the following system is controllable

$$\mathbf{x}[k+1] = \begin{bmatrix} 1.8 & 1 \\ 0.85 & 0 \end{bmatrix}\mathbf{x}[k] + \begin{bmatrix} 1 \\ 0 \end{bmatrix}u[k].$$

The controllability matrix for this second-order system is calculated to be

$$\mathbf{W}_c = [\mathbf{\Gamma}\ \mathbf{\Phi}\mathbf{\Gamma}] = \begin{bmatrix} 1 & 1.8 \\ 0 & 0.85 \end{bmatrix}.$$

The determinant of the $\mathbf{W}_c$ is 0.85, and so the system is controllable.

We now show that we can find an input sequence to take the system state vector from the origin at time $k = 0$ to an arbitrary state at time $k = 2$. Let the desired state at $k = 2$ be

$$\mathbf{x}[2] = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

From (6.8), the required input sequence is

$$\begin{aligned} \begin{bmatrix} u[1] \\ u[0] \end{bmatrix} &= \mathbf{W}_c^{-1}\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\ &= \begin{bmatrix} 1 & -2.1176 \\ 0 & 1.1765 \end{bmatrix}\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \\ &= \begin{bmatrix} \alpha - 2.1176\beta \\ 1.1765\beta \end{bmatrix}. \end{aligned}$$

We can demonstrate that this input sequence is correct by plugging it into the state update equation as follows

$$
\begin{aligned}
x[1] &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} u[0] = \begin{bmatrix} 1.1765\beta \\ 0 \end{bmatrix} \\
x[2] &= \begin{bmatrix} 1.8 & 1 \\ 0.85 & 0 \end{bmatrix} \begin{bmatrix} 1.1765\beta \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u[1] \\
&= \begin{bmatrix} 2.1176\beta \\ \beta \end{bmatrix} + \begin{bmatrix} \alpha - 2.1176\beta \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.
\end{aligned}
$$

■

EXAMPLE 6.2

Consider the following system

$$
\mathbf{x}[k+1] = \begin{bmatrix} 2 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} u[k].
$$

The controllability matrix for this system is calculated to be

$$
\mathbf{W}_c = [\, \mathbf{\Gamma} \quad \mathbf{\Phi\Gamma} \quad \mathbf{\Phi}^2\mathbf{\Gamma} \,] = \begin{bmatrix} 0 & 1 & 1 \\ 1 & -1 & -1 \\ -1 & 0 & 0 \end{bmatrix}.
$$

The determinant of $\mathbf{W}_c$ turns out to be zero, and so the system is not controllable.

■

The property of controllability is extremely important for control system design. It is a property that all "properly modeled" systems have. By properly modeled we mean that the state-space model of the plant does not have a transfer function with pole-zero cancellations. Such a model will be controllable and it will also be observable. Observability is another important property which is defined in Chapter 7. We summarize this result in the following theorem.

Theorem 6.2 *A state-space model* $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C}, D)$ *with transfer function* $H(z) = \mathbf{C}(z\mathbf{I} - \mathbf{\Phi})^{-1}\mathbf{\Gamma} + d = b(z)/a(z)$ *will be controllable and observable if and only if* $b(z)$ *and* $a(z)$ *have no common roots (i.e. no pole-zero cancellations).*

PROOF: See [50]. This theorem is also true for continuous-time systems $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D)$ with transfer function $b(s)/a(s)$.

■

A useful consequence of this theorem is that if an n^{th}-order system has no pole-zero cancellations in its transfer function, then any n^{th}-order state-space model for that system will be controllable and observable.

Controllability of ZOH Equivalent Models

So far in this chapter we have discussed controllability of discrete-time models. However the actual system to be controlled is a continuous-time system. Suppose the plant has the following state-space description

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}\mathbf{x}(t) + D\,u(t)\end{aligned} \tag{6.11}$$

and the transfer function of (6.11) does not have any pole-zero cancellations. From Theorem 6.2 we know that (6.11) is controllable. However (6.11) is not used directly to design a digital control system. Rather we first find the ZOH equivalent model which describes the behavior of the plant at sampling instants. The state-update equation for the ZOH equivalent model is

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k] \tag{6.12}$$

where

$$\mathbf{\Phi} = e^{\mathbf{A}T} \text{ and } \mathbf{\Gamma} = \int_0^T e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)d\tau$$

and T is the sampling interval. In order to design a digital control system we would like (6.12) to be controllable. We are thus led to the following question: if the continuous-time system (6.2) is controllable, are we guaranteed that the ZOH equivalent (6.12) will also be controllable? The answer is that the ZOH equivalent may or may not be controllable depending on the value of the sampling interval T. Fortunately, the conditions on T to insure controllability are not very restrictive, as we now show.

A simple sufficient condition on T to insure controllability of the ZOH equivalent model is stated in terms of the poles of the continuous-time system (eigenvalues of $\mathbf{A}$ in (6.11)). Suppose the pole locations $p_i, i = 1, \cdots, n$ are written in terms of their real and imaginary parts

$$p_i = \alpha_i + j\beta_i\ ,\ \ i = 1, \cdots, n.$$

We define β_{max} to be the imaginary part with the largest magnitude as follows

$$\beta_{\text{max}} \stackrel{\text{def}}{=} \max_i\ |\beta_i|\ ,\ \ i = 1, \cdots, n. \tag{6.13}$$

Then a sufficient condition on T to insure controllability of the ZOH equivalent model is [81]

$$\frac{1}{2T} > \frac{\beta_{\text{max}}}{2\pi}.$$

This inequality is similar in form to the condition for the sampling theorem mentioned in Chapter 1. The condition for controllability is summarized in the following theorem. This theorem, as well as another theorem giving necessary and sufficient conditions for controllability, is found in [81].

Theorem 6.3 *Given a controllable continuous-time system $(\mathbf{A}, \mathbf{BC}, D)$ with pole locations $p_i = \alpha_i + j\beta_i\ ,\ \ i = 1, \cdots, n$, let β_{max} be defined by (6.13). Then the ZOH equivalent model at sampling period T will be controllable if T satisfies the following inequality*

$$T < \frac{\pi}{\beta_{max}}.$$

Note that if the plant has all its poles on the real axis, then $\beta_{\max} = 0$ and the theorem says that the ZOH equivalent model is controllable for *any* value of T. As we will see later in this chapter, performance considerations will usually lead to a much smaller upper bound than that given in the theorem. As long as the bound given in the theorem is satisfied, we know that the ZOH equivalent will be controllable.

6.2 Transforming a Controllable System to Controllable Canonical Form

In this section we prove that any controllable system can be transformed to controllable canonical form, and show how the appropriate linear transformation can be calculated. This result will be used in the next section where we consider the design of state feedback regulators. The design procedure is derived for a system in controllable canonical form. However the procedure can be applied to *any* controllable system using the transformation derived in this section.

The controllable canonical form was derived in Chapter 3. The state-update equation for this canonical form is shown below. The special canonical form is indicated by putting a bar on top of all matrices and vectors.

$$\begin{aligned} \bar{\mathbf{x}}[k+1] &= \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_n \\ 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & \cdots & 1 & 0 \end{bmatrix} \bar{\mathbf{x}}[k] + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u[k]. \\ &= \bar{\mathbf{\Phi}}\bar{\mathbf{x}}[k] + \bar{\mathbf{\Gamma}}u[k] \end{aligned} \tag{6.14}$$

This canonical form has the following two properties:

1. The controllability matrix of the controllable canonical form is upper triangular, as shown below, where an asterisk represents a matrix element whose value we are not interested in.

$$\begin{aligned} \bar{\mathbf{W}}_c &= \begin{bmatrix} \bar{\mathbf{\Gamma}} & \bar{\mathbf{\Phi}}\bar{\mathbf{\Gamma}} & \cdots & \bar{\mathbf{\Phi}}^{n-1}\bar{\mathbf{\Gamma}} \end{bmatrix} \\ &= \begin{bmatrix} 1 & -a_1 & * & \cdots & * \\ 0 & 1 & * & \cdots & * \\ 0 & 0 & 1 & * & \\ \vdots & \vdots & 0 & \ddots & * \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}. \end{aligned} \tag{6.15}$$

   Thus the determinant of $\bar{\mathbf{W}}_c$ equals one (since the determinant of a triangular matrix is simply the product of its diagonal elements). This means that the system $(\bar{\mathbf{\Phi}}, \bar{\mathbf{\Gamma}})$ is always controllable regardless of the values of $a_1 \cdots a_n$.

2. Any controllable system can be transformed to controllable canonical form by proper choice of linear transformation. (This is proved below.)

The first property listed above is the reason for the name "controllable canonical form." The second property is stated next as a theorem.

Theorem 6.4 *An arbitrary controllable system* $(\mathbf{\Phi}, \mathbf{\Gamma})$ *can be transformed to controllable canonical form by a linear transformation.*

PROOF. The state-update equation for the given system is

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k].$$

The characteristic polynomial of $\mathbf{\Phi}$ is

$$a(z) = \det(z\mathbf{I} - \mathbf{\Phi}) = z^n + a_1 z^{n-1} + \cdots + a_n.$$

The coefficients from $a(z)$ can be used to form the state-update equation for controllable canonical form $(\bar{\mathbf{\Phi}}, \bar{\mathbf{\Gamma}})$ using (6.14). The controllability matrix for the given system $(\mathbf{\Phi}, \mathbf{\Gamma})$ is

$$\mathbf{W}_c = [\,\mathbf{\Gamma} \quad \mathbf{\Phi\Gamma} \quad \cdots \quad \mathbf{\Phi}^{n-1}\mathbf{\Gamma}\,]$$

and the controllability matrix for controllable canonical form is

$$\bar{\mathbf{W}}_c = \left[\,\bar{\mathbf{\Gamma}} \quad \bar{\mathbf{\Phi}}\bar{\mathbf{\Gamma}} \quad \cdots \quad \bar{\mathbf{\Phi}}^{n-1}\bar{\mathbf{\Gamma}}\,\right].$$

We leave it as an exercise to show that if two controllable systems $(\mathbf{\Phi}_1, \mathbf{\Gamma}_1)$ and $(\mathbf{\Phi}_2, \mathbf{\Gamma}_2)$ are related by a linear transformation $\mathbf{x}_1[k] = \mathbf{T}\mathbf{x}_2[k]$, then their controllability matrices are related by $\mathbf{W}_{c2} = \mathbf{T}\mathbf{W}_{c1}$. We can solve for $\mathbf{T}$ as $\mathbf{T} = \mathbf{W}_{c2}\mathbf{W}_{c1}^{-1}$. Using this result we can say that **if** a given controllable system $(\mathbf{\Phi}, \mathbf{\Gamma})$ and controllable canonical form $(\bar{\mathbf{\Phi}}, \bar{\mathbf{\Gamma}})$ are related by the linear transformation $\bar{\mathbf{x}}[k] = \mathbf{T}_c\bar{\mathbf{x}}[k]$, **then**

$$\mathbf{T}_c = \bar{\mathbf{W}}_c\mathbf{W}_c^{-1}. \tag{6.16}$$

From the statement of the theorem we do not know that $(\mathbf{\Phi}, \mathbf{\Gamma})$ and $(\bar{\mathbf{\Phi}}, \bar{\mathbf{\Gamma}})$ are necessarily related by a linear transformation. Nevertheless the matrix $\mathbf{T}_c$ in (6.16) is well defined and we now show that it is indeed the desired linear transformation.

From (6.16) we know that $\bar{\mathbf{W}}_c = \mathbf{T}_c\mathbf{W}_c$ and equating the first column of $\bar{\mathbf{W}}_c$ with the first column of $\mathbf{T}_c\mathbf{W}_c$ yields

$$\bar{\mathbf{\Gamma}} = \mathbf{T}_c\mathbf{\Gamma}. \tag{6.17}$$

The relationship between $\bar{\mathbf{\Phi}}$ and $\mathbf{\Phi}$ is derived as follows. We first note that

$$\begin{aligned}\mathbf{W}_c^{-1}\mathbf{W}_c &= \mathbf{W}_c^{-1}\,[\,\mathbf{\Gamma} \quad \mathbf{\Phi\Gamma} \quad \cdots \quad \mathbf{\Phi}^{n-1}\mathbf{\Gamma}\,] \\ &= \mathbf{I} \\ &= [\,\mathbf{e}_1 \quad \mathbf{e}_2 \quad \cdots \quad \mathbf{e}_n\,]\end{aligned}$$

where $\mathbf{e}_i$ is the i-th column of the $n \times n$ identity matrix. The right-hand sides in the first and third lines above can be equated to yield

$$\mathbf{W}_c^{-1}\mathbf{\Phi}^{n-i}\mathbf{\Gamma} = \mathbf{e}_{n-i+1}\,, \quad i = 1, \cdots, n. \tag{6.18}$$

We can then write

$$\begin{aligned}\mathbf{W}_c^{-1}\mathbf{\Phi}\mathbf{W}_c &= \mathbf{W}_c^{-1}\left[\mathbf{\Phi}\mathbf{\Gamma} \quad \mathbf{\Phi}^2\mathbf{\Gamma} \quad \cdots \quad \mathbf{\Phi}^{n-1}\mathbf{\Gamma} \quad \mathbf{\Phi}^n\mathbf{\Gamma}\right] \\ &= \mathbf{W}_c^{-1}\left[\mathbf{\Phi}\mathbf{\Gamma} \quad \mathbf{\Phi}^2\mathbf{\Gamma} \quad \cdots \quad \mathbf{\Phi}^{n-1}\mathbf{\Gamma} \quad -\sum_{i=1}^{n} a_i\mathbf{\Phi}^{n-i}\mathbf{\Gamma}\right] \\ &= \left[\mathbf{e}_2 \quad \mathbf{e}_3 \quad \cdots \quad \mathbf{e}_n \quad -\sum_{i=1}^{n} a_1\mathbf{e}_{n-i+1}\right].\end{aligned} \tag{6.19}$$

The second line in the above equation is derived using the Cayley-Hamilton theorem, and the third line follows from (6.18). We are now ready to show that $\mathbf{\Phi}$ and $\bar{\mathbf{\Phi}}$ are related by a similarity transformation $\mathbf{T}_c$. From (6.16) and (6.19) we have

$$\begin{aligned}\mathbf{T}_c\mathbf{\Phi}\mathbf{T}_c^{-1} &= \bar{\mathbf{W}}_c[\mathbf{W}_c^{-1}\mathbf{\Phi}\mathbf{W}_c]\bar{\mathbf{W}}_c^{-1} \\ &= \bar{\mathbf{W}}_c\left[\mathbf{e}_2 \quad \mathbf{e}_3 \quad \cdots \quad \mathbf{e}_n \quad -\sum_{i=1}^{n} a_1\mathbf{e}_{n-i+1}\right]\bar{\mathbf{W}}_c^{-1} \\ &= \left[\bar{\mathbf{\Phi}}\bar{\mathbf{\Gamma}} \quad \bar{\mathbf{\Phi}}^2\bar{\mathbf{\Gamma}} \quad \cdots \quad \bar{\mathbf{\Phi}}^{n-1}\bar{\mathbf{\Gamma}} \quad \bar{\mathbf{\Phi}}^n\bar{\mathbf{\Gamma}}\right]\bar{\mathbf{W}}_c^{-1} \\ &= \bar{\mathbf{\Phi}}\left[\bar{\mathbf{\Gamma}} \quad \bar{\mathbf{\Phi}}\bar{\mathbf{\Gamma}} \quad \cdots \quad \bar{\mathbf{\Phi}}^{n-2}\bar{\mathbf{\Gamma}} \quad \bar{\mathbf{\Phi}}^{n-1}\bar{\mathbf{\Gamma}}\right]\bar{\mathbf{W}}_c^{-1} \\ &= \bar{\mathbf{\Phi}}.\end{aligned} \tag{6.20}$$

From this equation and (6.17) we see that $(\mathbf{\Phi}, \mathbf{\Gamma})$ and $(\bar{\mathbf{\Phi}}, \bar{\mathbf{\Gamma}})$ are related by the linear transformation

$$\bar{\mathbf{x}}[k] = \mathbf{T}_c\mathbf{x}[k]. \tag{6.21}$$

■

The transformation matrix $\mathbf{T}_c$ which takes an arbitrary controllable system to controllable canonical form is shown in (6.16) in the proof of the above theorem. $\mathbf{T}_c$ is computed using $\bar{\mathbf{W}}_c$, the controllability matrix of the controllable canonical form. From (6.14), recall that the controllable canonical form is expressed in terms of the coefficients of $a(z)$, the denominator polynomial of the system (or equivalently, the characteristic polynomial of the matrix $\mathbf{\Phi}$). If a system $(\mathbf{\Phi}, \mathbf{\Gamma})$ is given, the characteristic polynomial must be computed in order to form the matrix $\bar{\mathbf{\Phi}}$ of controllable canonical form. There are many different ways of computing the characteristic polynomial of a $\mathbf{\Phi}$, some of which are listed below.

1. Compute $\det(z\mathbf{I} - \mathbf{\Phi})$ by hand. This involves taking the determinant of a matrix which depends on the variable z, which is only practical for small matrices.

2. If the poles of the continuous-time system are known, they can be mapped to the z-plane using the ZOH pole mapping formula. These z-plane poles, call them $z_1, \cdots, z_n$ are the eigenvalues of $\mathbf{\Phi}$. They can be multiplied out to get the characteristic polynomial. That is

$$a(z) = \prod_{i=1}^{n}(z - z_i).$$

3. Use the Fadeev algorithm shown in Table 3.5 on page 114 to recursively compute the coefficients of the characteristic polynomial in terms of the traces of certain matrices.

EXAMPLE 6.3

Consider the following continuous-time state-space model

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t).$$

The poles of the system are 0 and -2. The ZOH equivalent model for a sampling period of 0.1 seconds can be computed to be

$$\begin{aligned} \mathbf{x}[k+1] &= \begin{bmatrix} 1 & .0906 \\ 0 & .8187 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} .0047 \\ .0906 \end{bmatrix} u[k] \\ &= \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k]. \end{aligned}$$

In this example we compute the linear transformation which takes this discrete-time system to controllable canonical form. We begin by finding the characteristic polynomial of $\mathbf{\Phi}$. The eigenvalues of $\mathbf{\Phi}$ may be found using the ZOH pole-mapping formula on the poles of the continuous-time model as follows:

$$z_1 = e^{0T} = 1, \text{ and } z_2 = e^{-2T} = e^{-0.2} = 0.8187.$$

(In this example, the eigenvalues of $\mathbf{\Phi}$ are easily obtained as the diagonal elements of the upper-triangular matrix $\mathbf{\Phi}$. If the poles of the continuous-time model are known, the ZOH pole-mapping formula is the easiest way to obtain the eigenvalues of $\mathbf{\Phi}$, in general.) The characteristic polynomial of $\mathbf{\Phi}$ is found by multiplying out the eigenvalues as follows

$$a(z) = (z-1)(z-0.8187) = z^2 - 1.8187z + 0.8187$$

so that $a_1 = -1.8187$ and $a_2 = 0.8187$. These coefficients are used to form controllable canonical form using (6.14)

$$\bar{\mathbf{\Phi}} = \begin{bmatrix} 1.8187 & -0.8187 \\ 1 & 0 \end{bmatrix}, \quad \bar{\mathbf{\Gamma}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

The controllability matrix of controllable canonical form is computed next

$$\bar{\mathbf{W}}_c = \begin{bmatrix} 1 & 1.8187 \\ 0 & 1 \end{bmatrix}.$$

The controllability matrix for $(\mathbf{\Phi}, \mathbf{\Gamma})$ can be computed to be

$$\mathbf{W}_c = \begin{bmatrix} .0047 & .0129 \\ .0906 & .0742 \end{bmatrix}.$$

Finally (6.16) can be used to compute the desired linear transformation as follows

$$\mathbf{T}_c = \bar{\mathbf{W}}_c \mathbf{W}_c^{-1} = \begin{bmatrix} 110.3331 & 5.3329 \\ 110.3331 & -5.7004 \end{bmatrix}.$$

It can be verified by direct multiplication that the matrix $\mathbf{T}_c$ is correct, since

$$\bar{\mathbf{\Phi}} = \mathbf{T}_c \mathbf{\Phi} \mathbf{T}_c^{-1}, \text{ and } \bar{\mathbf{\Gamma}} = \mathbf{T}_c \mathbf{\Gamma}.$$

■

EXAMPLE 6.4

Consider the continuous-time system $(\mathbf{A}, \mathbf{B})$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -4 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

The ZOH equivalent system for a sampling interval of $T = 0.1$ is

$$\mathbf{\Phi} = \begin{bmatrix} 1.0000 & 0.0952 & 0.0042 \\ 0 & 0.9048 & 0.0782 \\ 0 & 0 & 0.6703 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0001 \\ 0.0042 \\ 0.0824 \end{bmatrix}.$$

In this example we compute the transformation matrix $\mathbf{T}_c$ which takes $(\mathbf{\Phi}, \mathbf{\Gamma})$ to controllable canonical form. We begin by computing the characteristic polynomial of $\mathbf{\Phi}$. This polynomial is obtained by multiplying out the eigenvalues of $\mathbf{\Phi}$ as follows

$$a(z) = (z-1)(z-.9048)(z-.6703) = z^3 - 2.5752z^2 + 2.1817z - .6065.$$

The coefficients of this polynomial are used to form the matrices for controllable canonical form as in (6.14)

$$\bar{\mathbf{\Phi}} = \begin{bmatrix} 2.5752 & -2.1817 & .6065 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \bar{\mathbf{\Gamma}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

The controllability matrix for controllable canonical form is

$$\bar{\mathbf{W}}_c = [\bar{\mathbf{\Gamma}} \ \bar{\mathbf{\Phi}}\bar{\mathbf{\Gamma}} \ \bar{\mathbf{\Phi}}^2\bar{\mathbf{\Gamma}}] = \begin{bmatrix} 1.0000 & 2.5752 & 4.4497 \\ 0 & 1.0000 & 2.5752 \\ 0 & 0 & 1.0000 \end{bmatrix}.$$

The controllability matrix for the given system is

$$\mathbf{W}_c = [\mathbf{\Gamma} \ \mathbf{\Phi}\mathbf{\Gamma} \ \mathbf{\Phi}^2\mathbf{\Gamma}] = \begin{bmatrix} 0.0001 & 0.0009 & 0.0021 \\ 0.0042 & 0.0103 & 0.0136 \\ 0.0824 & 0.0552 & 0.0370 \end{bmatrix}.$$

Using (6.16), the transformation matrix which takes the system to controllable canonical form is given by

$$\mathbf{T}_c = \begin{bmatrix} 1,2750. & 118.4 & 3.7 \\ 1,2750. & -3.2 & -2.1 \\ 1,2750 & -137.6 & 4.8 \end{bmatrix}.$$

It can be verified by direct multiplication that the matrix $\mathbf{T}_c$ is correct, since

$$\bar{\mathbf{\Phi}} = \mathbf{T}_c\mathbf{\Phi}\mathbf{T}_c^{-1}, \quad \text{and } \bar{\mathbf{\Gamma}} = \mathbf{T}_c\mathbf{\Gamma}.$$

■

6.3 Pole Placement by Digital State Feedback

So far in this chapter we have defined the property of controllability and shown how an arbitrary controllable system can be transformed into controllable canonical form. In this section, we show how to design state-feedback regulators for controllable discrete-time systems. Recall that the purpose of a regulator is to keep the state variables of the system at their nominal value (zero) in the face of disturbances acting on the plant. It will be shown that the poles of the closed-loop system can be placed in arbitrary desired locations using a constant linear combination of the state variables. Pole placement can be achieved for a system of any order, and the calculation of the feedback gains reduces to the solution of a system of linear equations.

Consider the regulation system shown in Fig. 6.1. We assume that all of the state vari-

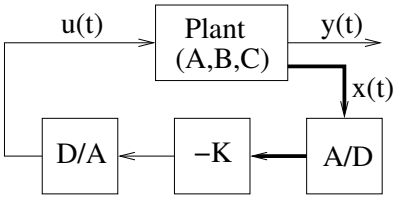


Figure 6.1 A regulation system using digital state feedback

ables are measured and sampled by n A/D converters. The vector $-\mathbf{K}$ is a (row) vector of constant gains which weights each state variable by some amount and adds the resulting contributions together to form the input sequence $u[kT]$, which is then converted to the analog signal $u(t)$ by a single D/A converter.

Assume that the system to be controlled is described by the state equation

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\,u(t). \tag{6.22}$$

Since the system is under digital control, it is possible to replace the continuous-time equation by its ZOH equivalent to get an exact model of the behavior of the plant at sampling instants. The ZOH equivalent system is described by

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k], \tag{6.23}$$

and the input to the plant is given by linear state feedback of the sampled state variables:

$$u[k] = -\mathbf{K}\,\mathbf{x}[k] = [-K_1 \; -K_2 \; \cdots \; -K_n] \begin{bmatrix} x_1[k] \\ \vdots \\ x_n[k] \end{bmatrix}. \tag{6.24}$$

Thus the *design model* for digital state feedback is the discrete-time ZOH equivalent system $(\mathbf{\Phi}, \mathbf{\Gamma})$. The ZOH equivalent of the plant has state feedback with the *same* $\mathbf{K}$ vector as the actual system shown in Fig. 6.1. The closed-loop discrete-time system is shown in Fig. 6.2(a). By the principle of ZOH equivalence, we know that $\mathbf{x}(kT) = \mathbf{x}[k]$. That is, the actual plant will have the desired behavior at sampling instants. The desired behavior is specified by the discrete-time design model. In the previous section we saw that any controllable system could be transformed to controllable canonical form by an appropriate

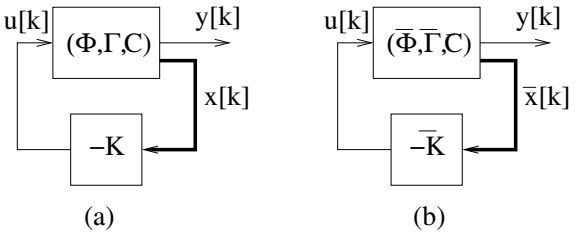


Figure 6.2 (a) ZOH design model for digital state feedback. (b) The transformed design model in controllable canonical form.

linear transformation. In particular, we can transform the ZOH equivalent $(\mathbf{\Phi}, \mathbf{\Gamma})$ to canonical form $(\bar{\mathbf{\Phi}}, \bar{\mathbf{\Gamma}})$ and close the loop with state feedback through the vector of gains $-\bar{\mathbf{K}}$ as shown in Fig. 6.2. It turns out to be a simple matter to calculate the feedback vector $\bar{\mathbf{K}}$ which places the closed-loop poles of the transformed design model in desired locations. Once this is done, it is then possible to transform the vector $\bar{\mathbf{K}}$ into the vector $\mathbf{K}$ which places the closed-loop poles of the ZOH design model in desired locations. The vector $\mathbf{K}$ is the vector of gains that is used in the actual digital regulator shown in Fig. 6.1. In the next two subsections we show how $\bar{\mathbf{K}}$ is computed, and how to obtain $\mathbf{K}$ from $\bar{\mathbf{K}}$.

6.3.1 The Regulation Problem in Controllable Canonical Form

Let us calculate the closed loop pole locations for the system shown in Fig. 6.2(b) (i.e. for controllable canonical form with state feedback). Equation (6.14) describes the open-loop system, and the feedback connection which closes the loop is described by

$$u[k] = -\bar{\mathbf{K}}\bar{\mathbf{x}}[k]. \tag{6.25}$$

Substituting the above equation into (6.14) yields

$$\begin{aligned} \bar{\mathbf{x}}[k+1] &= \bar{\mathbf{\Phi}}\,\bar{\mathbf{x}}[k] + \bar{\mathbf{\Gamma}}(-\bar{\mathbf{K}}\bar{\mathbf{x}}[k]) \\ &= (\bar{\mathbf{\Phi}} - \bar{\mathbf{\Gamma}}\bar{\mathbf{K}})\bar{\mathbf{x}}[k]. \end{aligned} \tag{6.26}$$

Note that $\bar{\mathbf{\Gamma}}\bar{\mathbf{K}}$ is given by

$$\begin{aligned} \bar{\mathbf{\Gamma}}\bar{\mathbf{K}} &= \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} [\,\bar{K}_1 \quad \bar{K}_2 \quad \cdots \quad \bar{K}_n\,] \\ &= \begin{bmatrix} \bar{K}_1 & \bar{K}_2 & \cdots & \bar{K}_n \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{6.27}$$

Substituting (6.27) into (6.26), the closed-loop system can be written as

$$\bar{\mathbf{x}}[k+1] = \begin{pmatrix} -a_1 - \bar{K}_1 & -a_2 - \bar{K}_2 & \cdots & -a_n - \bar{K}_n \\ 1 & 0 & \cdots & 0 \\ & & & \\ 0 & \ddots & \cdots & 0 \\ 0 & \cdots & 1 & 0 \end{pmatrix} \bar{\mathbf{x}}[k]. \tag{6.28}$$

Notice that the above matrix has the same form as the controllable canonical form open-loop system in (6.14), but that the elements in the first row are different. We know that for controllable canonical form, the eigenvalues (poles) are equal to the roots of a polynomial given by the negative of the elements in the first row. Suppose we want the eigenvalues of the matrix in(6.28) to equal the roots of a *specified* polynomial $p(z)$

$$p(z) = z^n + p_1 z^{n-1} + \cdots + p_n. \tag{6.29}$$

Then we should choose the gain parameters $\bar{K}_1, \cdots, \bar{K}_n$ to satisfy the following relationships so that the first row of the matrix in (6.28) will contain the negatives of the coefficients of the polynomial $p(z)$:

$$\begin{aligned} -a_1 - \bar{K}_1 &= -p_1 \\ -a_2 - \bar{K}_2 &= -p_2 \\ \vdots &= \vdots \\ -a_n - \bar{K}_n &= -p_n \end{aligned} \tag{6.30}$$

which results in the following choice for the state feedback vector $\bar{\mathbf{K}}$:

$$\bar{\mathbf{K}} = [(p_1 - a_1) \ (p_2 - a_2) \ \cdots \ (p_n - a_n)]. \tag{6.31}$$

With this choice of gains, the closed-loop transformed design model will have its poles at the roots of the specified polynomial $p(z)$. Observe that the above solution for the feedback vector is based strongly on the special form of the transformed design model $\bar{\mathbf{\Phi}}$ and $\bar{\mathbf{\Gamma}}$.

6.3.2 Pole Placement for an Arbitrary Controllable System

The poles of the ZOH equivalent are the eigenvalues of $\mathbf{\Phi}$ which are the roots of the polynomial

$$a(z) = \det(z\mathbf{I} - \mathbf{\Phi}) = z^n + a_1 z^{n-1} + \cdots + a_n. \tag{6.32}$$

The design problem is to calculate a vector $\mathbf{L}$ of feedback gains so that the closed-loop system shown in Fig. 6.2(a) has desired pole locations. The state equations for the closed-loop system are obtained as follows. The equations for the plant are

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k] \tag{6.33}$$

and the equation for the feedback connection is

$$u[k] = -\mathbf{K}\mathbf{x}[k]. \tag{6.34}$$

Substituting (6.34) into (6.33) yields

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{\Phi}\mathbf{x}[k] - \mathbf{\Gamma}(-\mathbf{K}\mathbf{x}[k]) \\ &= (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})\mathbf{x}[k]. \end{aligned} \tag{6.35}$$

Notice that the closed-loop system has no external input since we are considering the regulation problem (another way to say this is that the reference input equals zero). The mathematical problem which must be solved is to calculate the vector of feedback gains $\mathbf{L}$ so that the eigenvalues of the closed-loop system matrix $\mathbf{\Phi} - \mathbf{\Gamma K}$ shown in (6.35) are at desired values. This problem has already been solved for controllable canonical form. To solve the problem for an arbitrary system, we need to use the transformation $\mathbf{T}_c$ which relates the given system and controllable canonical form. Before doing this, we first note that (6.35) is a homogeneous difference equation, and has the following solution

$$\mathbf{x}[k] = (\mathbf{\Phi} - \mathbf{\Gamma K})^k \mathbf{x}[0]. \tag{6.36}$$

The closed-loop poles are the eigenvalues of $(\mathbf{\Phi} - \mathbf{\Gamma K})$. If these eigenvalues are inside the unit circle, then from (6.36) it can be seen that $\mathbf{x}[k] \Rightarrow \mathbf{0}$ as k gets large (see the discussion following (2.31) in Chapter 2). The rate at which the state vector goes to $\mathbf{0}$ depends on the exact locations of the eigenvalues of $(\mathbf{\Phi} - \mathbf{\Gamma K})$ which are the roots of the polynomial $p(z)$. The problem of choosing the closed loop pole locations is considered in Section 6.4 starting on page 249.

At this point, we have developed enough theory to propose a method for designing a digital compensator based on state-variable feedback. Although a few more details remain to be worked out, it is nevertheless useful to summarize the approach now:

1. Given a continuous-time plant, and continuous-time performance specifications, choose appropriate s-plane closed-loop pole locations and select the sampling interval T. Guidelines for selecting T are given in the next section.

2. Obtain the ZOH equivalent of the plant, and map the s-plane pole locations into the z-plane using the ZOH pole mapping formula discussed in section 4.4. Multiply out these pole locations to obtain the polynomial $p(z)$ which is the desired closed-loop denominator polynomial of the discrete-time system.

3. Transform the system to controllable canonical form by the appropriate similarity transformation.

4. Calculate the feedback vector $\bar{\mathbf{K}}$ for the transformed design model to place the poles at the roots of $p(z)$.

5. Transform back to the original state variables and find $\mathbf{K}$ in terms of $\bar{\mathbf{K}}$.

The last three steps mentioned above are not actually carried out in practice. Rather, they are used to develop a formula which performs all three steps at the same time. This formula is derived below.

In the previous section it was shown that for the transformed design model $(\bar{\mathbf{\Phi}}, \bar{\mathbf{\Gamma}})$ in controllable canonical form, the feedback vector

$$\bar{\mathbf{K}} = [\,(p_1 - a_1) \cdots (p_n - a_n)\,]$$

will place the poles of the closed loop system at the roots of

$$p(z) = z^n + p_1 z^{n-1} + \cdots + p_n.$$

However, the actual plant shown in Fig. 6.1 will be controlled using the gains in the vector $\mathbf{K}$, not $\bar{\mathbf{K}}$. Thus the vector $\mathbf{K}$ which places the poles of the design model must still be

calculated. $\mathbf{K}$ is related to $\bar{\mathbf{K}}$ by the linear transformation matrix $\mathbf{T}_c$ which takes the design model to controllable canonical form (see Fig. 6.2). We now derive the relationship between $\mathbf{K}$ and $\bar{\mathbf{K}}$.

Recall that a linear transformation does not alter the input/output behavior of a system; it only rearranges the state variables. Thus the inputs and outputs of the two systems in Fig. 6.2 are equal. Since the state vectors are related by the similarity transformation matrix $\mathbf{T}_c$, the state feedback which drives the two systems are also related:

$$\begin{aligned} \bar{x}[k] &= \mathbf{T}_c\mathbf{x}[k] \\ u[k] &= -\bar{\mathbf{K}}\bar{x}[k] = -\bar{\mathbf{K}}\mathbf{T}_c\mathbf{x}[k] = -\mathbf{K}\mathbf{x}[k] \end{aligned} \tag{6.37}$$

The last equality must be true for any value of the state vector $\mathbf{x}[k]$, and so we see that

$$\mathbf{K} = \bar{\mathbf{K}}\mathbf{T}_c. \tag{6.38}$$

The above formula tells us how to compute a feedback vector for an arbitrary controllable system. First, we compute the feedback vector for the controllable canonical form which is just the vector $\bar{\mathbf{K}}$ given by (6.31). This vector is transformed by the linear transformation $\mathbf{T}_c$ which relates the given system to controllable canonical form. $\mathbf{T}_c$ is given by (6.16) as

$$\mathbf{T}_c = \bar{\mathbf{W}}_c\mathbf{W}_c^{-1} \tag{6.39}$$

or

$$\mathbf{T}_c = \left[\bar{\mathbf{\Gamma}} \quad \bar{\mathbf{\Phi}}\bar{\mathbf{\Gamma}} \quad \cdots \quad \bar{\mathbf{\Phi}}^{n-1}\bar{\mathbf{\Gamma}}\right]\left[\mathbf{\Gamma} \quad \mathbf{\Phi}\mathbf{\Gamma} \quad \cdots \quad \mathbf{\Phi}^{n-1}\mathbf{\Gamma}\right]^{-1}. \tag{6.40}$$

Note that the controllability matrix $\mathbf{W}_c$ can be computed from $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ of the given system. The matrix $\bar{\mathbf{W}}_c$ is the controllability matrix for controllable canonical form, and it can be calculated using $\bar{\mathbf{\Phi}}$ and $\bar{\mathbf{\Gamma}}$ given in (6.14).

Equation (6.38) yields the desired formula for the feedback vector for an arbitrary controllable system:

$$\mathbf{K} = [\,(p_1 - a_1) \quad (p_2 - a_2) \quad \cdots \quad (p_n - a_n)\,]\,\bar{\mathbf{W}}_c\mathbf{W}_c^{-1}. \tag{6.41}$$

In order to use the above formula to compute a feedback vector, the following quantities must be known:

1. $\mathbf{\Phi}$ and $\mathbf{\Gamma}$, the ZOH equivalent of the system to be controlled.
2. The coefficients $p_1, \cdots, p_n$ of the polynomial $p(z)$ whose roots are the desired pole locations.
3. The coefficients $a_1, \cdots, a_n$ of $a(z)$, the denominator polynomial of the ZOH model.

The third item listed above is redundant, since the matrix $\mathbf{\Phi}$ is already used, and $a(z)$ is just the characteristic polynomial of $\mathbf{\Phi}$. A different formula for computing $\mathbf{K}$ is Ackermann's formula, which does not use $a(z)$.

Ackermann's Formula

$$\mathbf{K} = [0 \ \cdots \ 1]\mathbf{W}_c^{-1}p(\mathbf{\Phi})$$

1. Given a continuous-time system and performance specifications, choose appropriate s-plane pole locations and sampling interval T (see section 6.4 for information on choosing closed-loop pole locations and sampling interval).
2. Obtain the ZOH equivalent of the plant, and calculate the denominator polynomial

$$\det(z\mathbf{I} - \mathbf{\Phi}) = a(z) = z^n + a_1 z^{n-1} + \cdots + a_n$$

of the ZOH model. If the poles s_i of the continuous-time plant are known, then $a(z)$ can be computed as follows

$$a(z) = \prod_{i=1}^{n} (z - e^{s_i T}).$$

3. Map the s-plane pole locations chosen in Step 1 to the z-plane using the ZOH pole-mapping formula $z_i = e^{s_i T}$. Multiply out the z-plane pole locations to obtain $p(z)$

$$\prod_{i=1}^{n} (z - z_i) = z^n + p_1 z^{n-1} + \cdots + p_n \stackrel{\text{def}}{=} p(z).$$

4. Form the controllability matrix $\mathbf{W}_c$ of the plant (ZOH), and form the controllability matrix $\bar{\mathbf{W}}_c$ of controllable canonical form.
5. Calculate the feedback vector

$$\mathbf{K} = [\,(p_1 - a_1) \quad (p_2 - a_2) \quad \cdots \quad (p_n - a_n)\,]\,\bar{\mathbf{W}}_c \mathbf{W}_c^{-1}.$$

Table 6.1 A procedure for calculating the state feedback vector.

where $\mathbf{W}_c$ is the controllability matrix of the ZOH model $(\mathbf{\Phi}, \mathbf{\Gamma})$, and $p(\mathbf{\Phi})$ is the matrix polynomial $\mathbf{\Phi}^n + p_1 \mathbf{\Phi}^{n-1} + \cdots + p_n \mathbf{I}$. The roots of $p(z)$ are the desired closed-loop pole locations.

Ackermann's formula is a useful theoretical tool. For hand calculations, it is about the same amount of work to use Ackermann's formula as it is to evaluate (6.41). If the calculations are done by computer, it is easy to find the characteristic polynomial (find the eigenvalues and multiply them out). However, for best numerical results, the algorithm given in Chapter 9 for multivariable systems should be used with the number of inputs to the plant equal to one.

A complete design procedure for state feedback must include a discussion of the choice of closed-loop pole locations. This topic is covered in the next section. A summary of the procedure to design digital state-feedback regulators is given in Table 6.1. We conclude this section with an example showing how the feedback gains are calculated for a given polynomial $p(z)$ which specifies the closed-loop poles.

EXAMPLE 6.5

Consider the system $(\mathbf{\Phi}, \mathbf{\Gamma})$ given in Example 6.4. Suppose we want to calculate a feedback vector $\mathbf{K}$ which will place the closed-loop poles at the roots of the polynomial

$$p(z) = z^3 - 2.3893z^2 + 1.9265z - 0.5235$$

(methods for choosing $p(z)$ are given in Section 6.4). The roots of $p(z)$ are 0.7784, $0.8055 \pm j0.1543$ which are all within the unit circle. Using the coefficients of $p(z)$ given above and the coefficients of $a(z)$ given in Example 6.4, the vector $\bar{\mathbf{K}}$ can be calculated as

$$\begin{aligned}\bar{\mathbf{K}} &= [\,(-2.3893 + 2.5752) \quad (1.9265 - 2.1817) \quad (-0.5235 + 0.6065)\,] \\ &= [\,0.1858 \quad -0.2551 \quad 0.0830\,].\end{aligned}$$

and using (6.41), we multiply this vector by the matrix $\mathbf{T}_c$ computed in Example 6.4 as follows

$$\mathbf{K} = \bar{\mathbf{K}}\mathbf{T}_c = [\,17.4134 \quad 11.4013 \quad 1.6358\,].$$

The closed-loop system matrix shown in (6.35) is

$$\mathbf{\Phi} - \mathbf{\Gamma K} = \begin{bmatrix} 0.9974 & 0.0935 & 0.0040 \\ -0.0740 & 0.8564 & 0.0712 \\ -1.4352 & -0.9397 & 0.5355 \end{bmatrix}.$$

The characteristic polynomial of the above matrix can be computed using the Fadeev algorithm shown in Table 3.5 to be

$$\det(zI - (\mathbf{\Phi} - \mathbf{\Gamma K})) = z^3 - 2.3893z^2 + 1.9265z - 0.5235$$

which is equal to the given polynomial $p(z)$ as it should be.

■

6.3.3 Regulator Stability Margins

In Chapter 5 we defined gain and phase margins for the closed-loop control system as shown in Fig. 6.3.

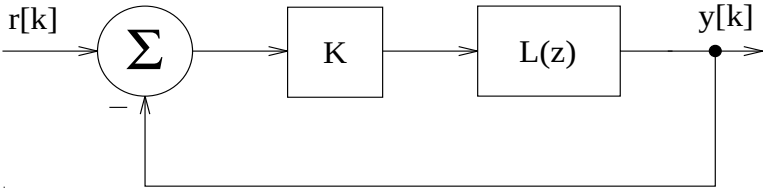


Figure 6.3 A negative unity feedback control system.

The stability margins were determined from the Nyquist plot of the loop transfer function $L(z)$ in conjunction with the critical point $-1/K$. The stability margins show how large a gain or phase perturbation can be tolerated in $L(z)$ before the closed-loop system becomes be unstable.

It is important to be able to find stability margins for a digital regulator such as that shown in Fig. 6.1. In order to do this we need to convert the regulator in Fig. 6.1 to a single-input, single output negative unity feedback system. This is accomplished by "breaking" the regulator loop at the input to the D/A converter and moving the negative sign on $-\mathbf{K}$ to the summing junction, as shown in Fig. 6.4.

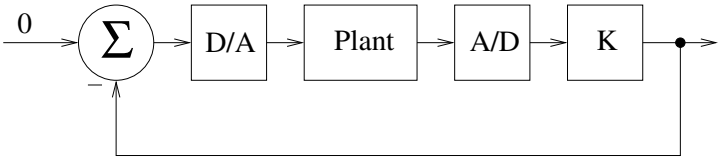


Figure 6.4 A re-arrangement of a digital regulator in the form of a negative unity feedback system.

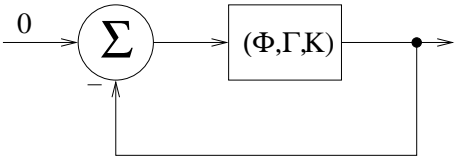


Figure 6.5 The discrete-time equivalent of Fig. 6.4.

The plant surrounded by D/A and A/D converters can be replaced by its ZOH equivalent $(\mathbf{\Phi}, \mathbf{\Gamma})$ which describes its behavior at sampling instants. The result is the discrete-time system shown in Fig. 6.5. This figure is equivalent to Fig. 6.3 with the following minor differences

- The reference input is zero instead of arbitrary. This difference is not important when analyzing the stability of the closed-loop system.
- The state-space model $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{K})$ in Fig. 6.5 represents the loop transfer function $L(z)$ in Fig. 6.3.
- The system in Fig. 6.5 does not show a separate forward-path gain K; equivalently, the forward-path gain equals 1. This means that the critical point of the Nyquist plot for the system in Fig. 6.5 is -1.

Thus to analyze the gain and phase margins of the regulator shown in Fig. 6.1, we look at the Nyquist plot of

$$L(z) = \mathbf{K}(z\mathbf{I} - \mathbf{\Phi})^{-1}\mathbf{\Gamma}$$

with critical point -1.

6.4 Choosing Closed-Loop Pole Locations

The formulas derived in the last section can be used to calculate a feedback vector which will place the poles of the closed-loop system in desired locations. The question that the control system designer must answer is, "Where do I want the closed loop poles to be located?" There are several ways to answer this question.

1. Choose the closed-loop poles to optimize a performance index.
2. Choose s-plane pole locations based on the classical control concept of dominant second-order poles. Map these poles into the z-plane using the ZOH pole mapping formula.
3. Choose all the closed-loop poles to be at the origin in the z-plane (deadbeat control).

4. Use a precomputed set of pole locations with a known prototype step response, such as those of a Bessel filter. Map the s-plane poles into the z-plane using the ZOH pole mapping formula.

Method 1 is the subject of Chapter 10. We briefly elaborate on methods 2–4 below.

6.4.1 Specifying Dominant Poles

Classical control concepts based on the root locus can be used to specify the locations of two dominant closed-loop pole locations. The calculations are based on the desired damping ratio ζ and natural frequency ω_n of the pole locations. The coefficients of the denominator polynomial for a 2nd-order system are written in terms of ζ and ω_n as follows

$$s^2 + 2\zeta\omega_n s + \omega_n^2.$$

The roots of this polynomial are

$$s_{1,2} = -\zeta\omega_n \pm j\sqrt{1-\zeta^2}.$$

One difficulty with the classical design procedure is that if the system to be controlled is greater than second order, the "extra" poles may be too close to the dominant poles so that the design formulas do not accurately predict system performance (since they are exact only for second order systems). However, it is known that poles "far enough" into the left half-plane do not affect the behavior of the dominant poles. If the dominant poles have real part $\sigma = -\zeta\omega_n$, then poles with real parts more than 4σ into the left half plane can be neglected. Thus we can choose to place two poles in locations corresponding to a desired ζ and ω_n. We place any remaining poles so that their real parts are less than $-4\zeta\omega_n$. Recall that if the plant is n^{th}-order then we must specify n closed-loop pole locations. Once the desired pole locations are specified in the s-plane, they must be mapped into the z-plane using the ZOH pole mapping formula. The z-plane pole locations can then be multiplied out to obtain the polynomial $p(z)$ which is used to calculate the feedback vector $\mathbf{L}$.

EXAMPLE 6.6

Suppose we want to design a digital control system for the following plant

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

with step response specifications of 1% overshoot and settling time equal to 2 seconds. We will use the following state-space description for the plant

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

Using results for second order systems (see [23] for example), the step response specifications could be achieved with pole locations specified by the polynomial

$$s^2 + 2\zeta\omega_n s + \omega_n^2$$

with the values $\zeta = 0.83$ and $\omega_n = 2.7$. That is, two desired closed-loop poles are at the roots of

$$s^2 + 4.482s + 7.29$$

or $s = -2.24 \pm j1.5$. In order for these roots to be dominant, the third pole of the system should be placed far enough into the left-half plane, say at $s = -4\zeta\omega_n \approx -9$. Suppose the digital control system will use a sampling period of $T = 0.1$ sec. Then the desired closed-loop s-plane poles are mapped into the z-plane using the ZOH pole mapping formula $z_i = e^{s_i T}$ which gives

$$\begin{aligned} s_1 &= -2.2410 + j1.5060, & z_1 &= e^{0.1s_1} = 0.7902 + j0.1199 \\ s_2 &= -2.2410 - j1.5060, & z_2 &= e^{0.1s_2} = 0.7902 - j0.1199 \\ s_3 &= -9, & z_3 &= e^{0.1s_3} = .4066. \end{aligned}$$

Finally, these z-plane roots can be multiplied out to obtain the polynomial $p(z)$

$$\begin{aligned} p(z) &= (z - z_1)(z - z_2)(z - z_3) \\ &= z^3 - 1.9869z^2 + 1.2813z - 0.2597. \end{aligned}$$

Following the procedure in Table 6.1 we obtain

$$\mathbf{K} = [\,44.1846 \quad 24.8134 \quad 5.7789\,].$$

Suppose the initial state of the plant was

$$\mathbf{x}[0] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

The regulator is designed to drive all of the state variable to zero with a settling time of 2 seconds. Fig. 6.6 shows the state variables of the regulated system as a function of time. The 2 second settling time is achieved. Notice that the second, third, and fourth state variables have zero error at $t = 0$, but as the regulator moves the system to decrease the initial error in the first state variable, the other three state variables take on non-zero values. The motion of the closed-loop system is coordinated in such a way that the error in each of the state variables is driven to zero by $t = 2$ seconds.

From Fig. 6.6 we see that the input signal has an initial value of about -43. This signal could not be the output of a standard D/A converter with an output range of ± 5 volts. In this case, the gains in the feedback vector $\mathbf{K}$ could be divided by 10, and an analog amplifier with a gain of 10 inserted between the D/A converter and the plant. In this way, the loop gain of the regulator would remain the same but the peak output of the D/A converter would now be -4.3 volts.

The stability margins for this regulator can be evaluated from a Nyquist plot of $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{K})$. The full Nyquist plot is shown in Fig. 6.7 (top figure), and an expanded portion of the plot together with the unit circle is shown in Fig. 6.7 (bottom figure).

The Nyquist plot shows that the closed-loop system remains stable if the loop gain is reduced, but since the unstable region intersects the negative real axis at -0.2884, the system has an (upper) gain margin of

$$\text{GM} \;=\; 20\log\,\frac{1}{.2885} = 10.8\text{ dB}.$$

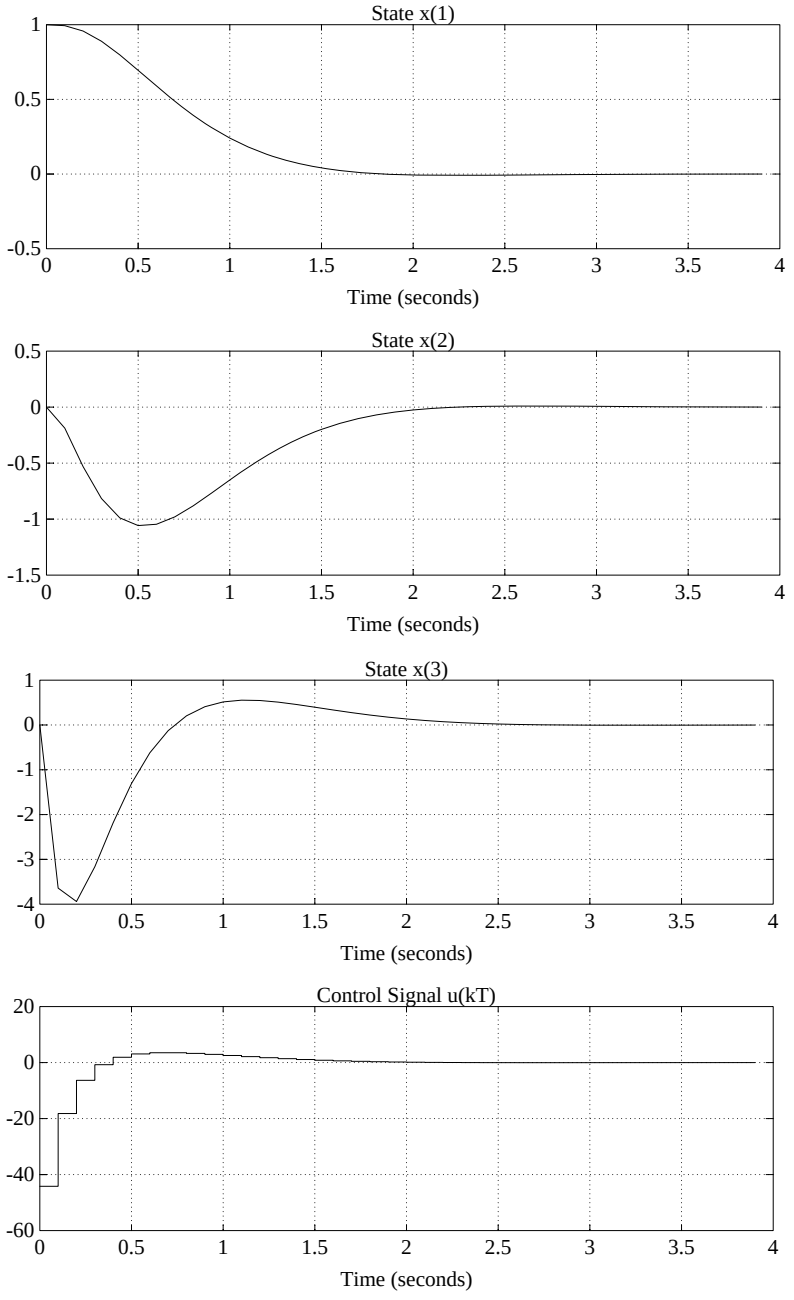


Figure 6.6 State variables and control input for the regulator of Example 6.6.

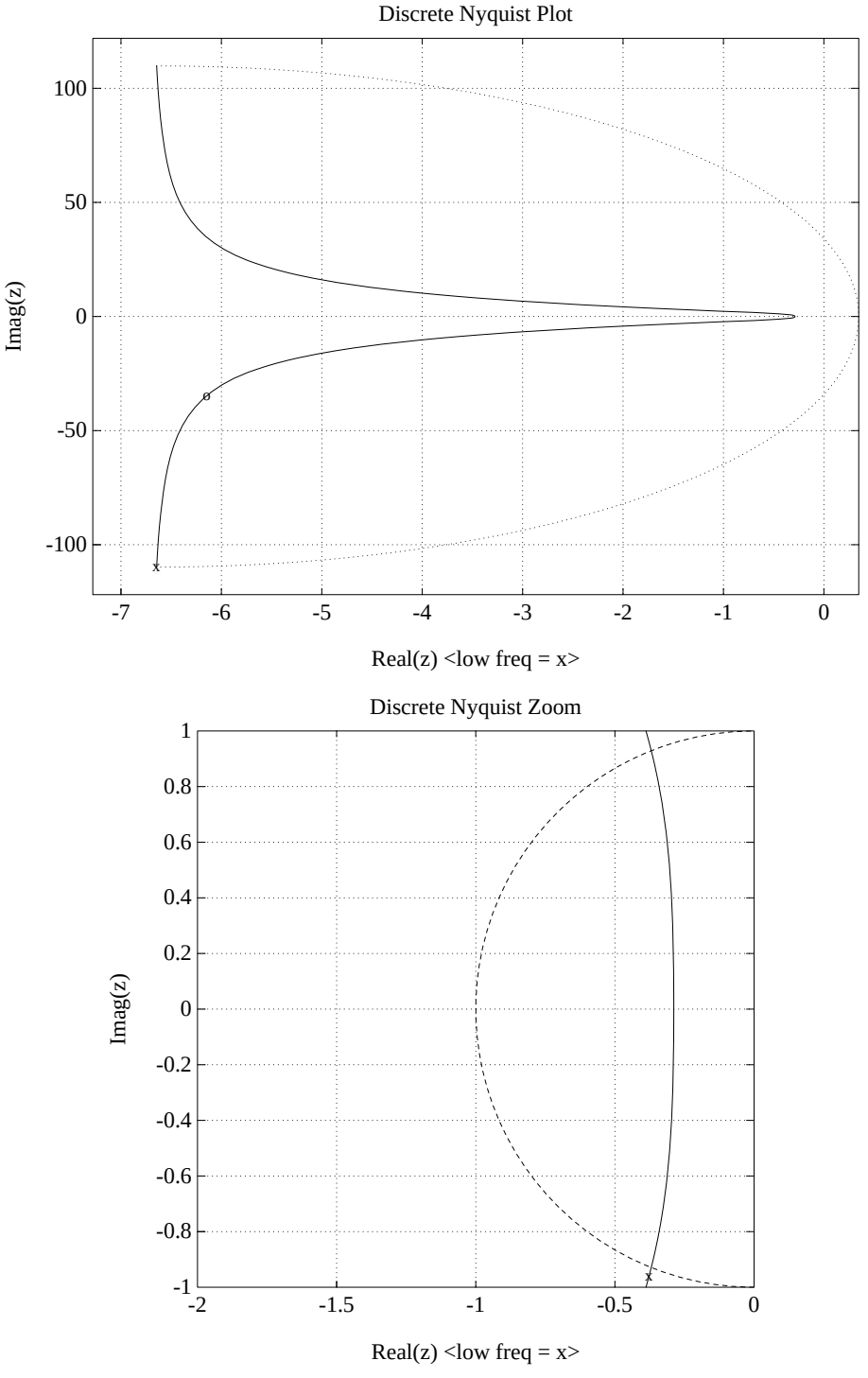


Figure 6.7 Nyquist plot for the regulator in Example 6.6 (top figure). An expanded portion of the Nyquist plot together with the unit circle (dashed line) (bottom figure).

The point at which the Nyquist plot intersects the unit circle is about 68.5° away from -180°, and so the phase margin is

$$\text{PM} \;=\; 68.5^\circ.$$

These stability margins are excellent, and the transient behavior of the regulator (Fig. 6.6) is also good. Thus the design is complete.

■

6.4.2 Deadbeat Control

We know that as the closed-loop poles move further into the left-half s-plane, the system responds faster and faster to command inputs. In the extreme case as s goes to minus infinity (which is not actually possible for an analog system), the system would be "as fast as possible." A pole at $s = -\infty$ maps to $z = 0$ under the ZOH pole mapping formula $z = e^{sT}$. Thus an interesting choice of closed-loop pole locations in the z-plane is to put them all at the origin! The polynomial which specifies these desired pole locations is just $p(z) = z^n$. This choice of closed-loop poles is referred to as *deadbeat control*.

Having all the closed-loop poles at the origin means that the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma K}$ will all be zero, and the characteristic polynomial of $\mathbf{\Phi} - \mathbf{\Gamma K}$ will be z^n. From the Cayley-Hamilton theorem we know that

$$(\mathbf{\Phi} - \mathbf{\Gamma K})^n = \mathbf{0}.$$

If we repeatedly multiply the above equation by $\mathbf{\Phi} - \mathbf{\Gamma K}$, it is clear that

$$(\mathbf{\Phi} - \mathbf{\Gamma K})^k = \mathbf{0}\;, \quad k \geq n.$$

The behavior of the closed-loop system is described by

$$\mathbf{x}[k] = (\mathbf{\Phi} - \mathbf{\Gamma K})^k \mathbf{x}[0].$$

From the previous two equations we see that, with deadbeat control,

$$\mathbf{x}[k] = \mathbf{0}\;, \quad k \geq n \tag{6.42}$$

where n is the dimension of the plant model. Equation (6.42) says that a deadbeat regulator will drive the state variables to zero in n time steps regardless of the value of $\mathbf{x}[0]$. From the discussion of controllability we know that it is possible to find an input which will drive a controllable system to a desired state in n time steps. Deadbeat control provides a way to compute this input in the form of state feedback.

The settling time of a deadbeat regulator will be nT seconds, assuming that the behavior between sampling instants is not oscillatory. If the sampling interval T is small, the settling time will be small. A small settling time might not be achievable in practice because it requires input signals with large magnitude. Since n is fixed, the only way to increase the settling time of a deadbeat regulator is to increase the sampling interval T. In some applications it may be acceptable to have a large sampling interval. However, since a digital regulator can only respond at sampling instants to disturbances, a large sampling interval results in poor disturbance rejection. Thus deadbeat digital control should only be used in special situations satisfying the following requirements: the mathematical model for the plant must be highly accurate, there cannot be any disturbances acting on the plant, and the plant must be able to handle large discontinuities in its input.

EXAMPLE 6.7

Consider the plant that was used in Example 6.6 on page 250. In that example the sampling interval was $T = 0.1$ seconds. Suppose we design a deadbeat regulator with the same sampling interval. Following the procedure given in Table 6.1 with $p(z) = z^3$ yields the following vector of feedback gains

$$\mathbf{K} = [\,1,275 \quad 228.5 \quad 17.2\,].$$

From the theory of deadbeat control we expect this regulator to have a settling time of $T_S = 3(0.1) = 0.3$ seconds. Fig. 6.8(a) shows the state variables of the regulated system as a function of time starting from the initial state vector

$$\mathbf{x}[0] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

The settling time is indeed 0.3 seconds but this is unrealistically small because the

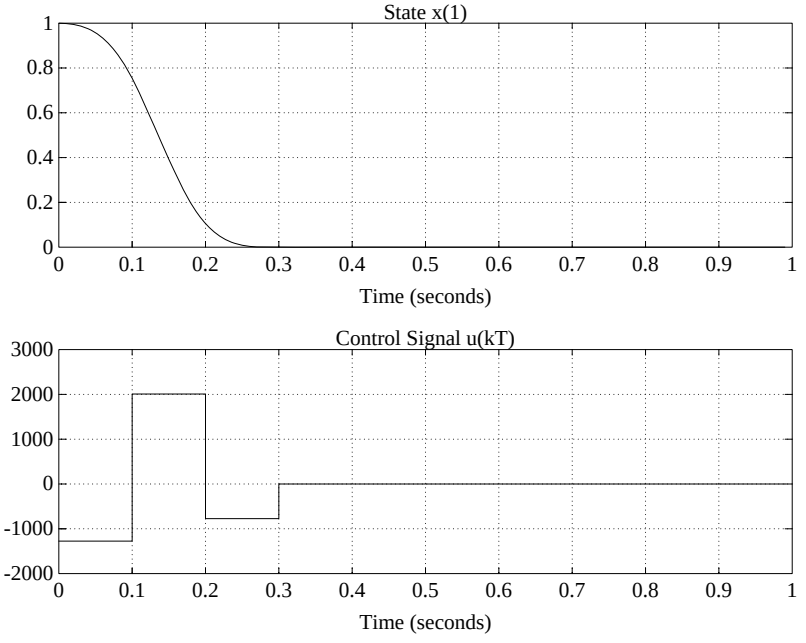


Figure 6.8 State variable x_1 and input u for the regulator of Example 6.7 with $T = 0.1$.

required input signal shown in Fig. 6.8(b) is too large. In other words, this regulator "works" in simulation but could probably not be implemented in hardware.

We know from Example 6.6 that a settling time of 2 seconds can be achieved. If we want to have this settling time using a deadbeat regulator for this 3rd-order plant, we must choose the sampling interval to be

$$T = \frac{T_S}{n} = \frac{2}{3}.$$

After calculating a new ZOH equivalent for this sampling time, the feedback vector which places the closed-loop poles at the origins can be calculated to be

$$\mathbf{K} = [\,13.2517 \quad 9.3889 \quad 2.0288\,].$$

Fig. 6.9 shows the state variable x_1 and the input u of the regulated system as a function of time. Notice that the peak input signal is about -12, as compared to -43 for

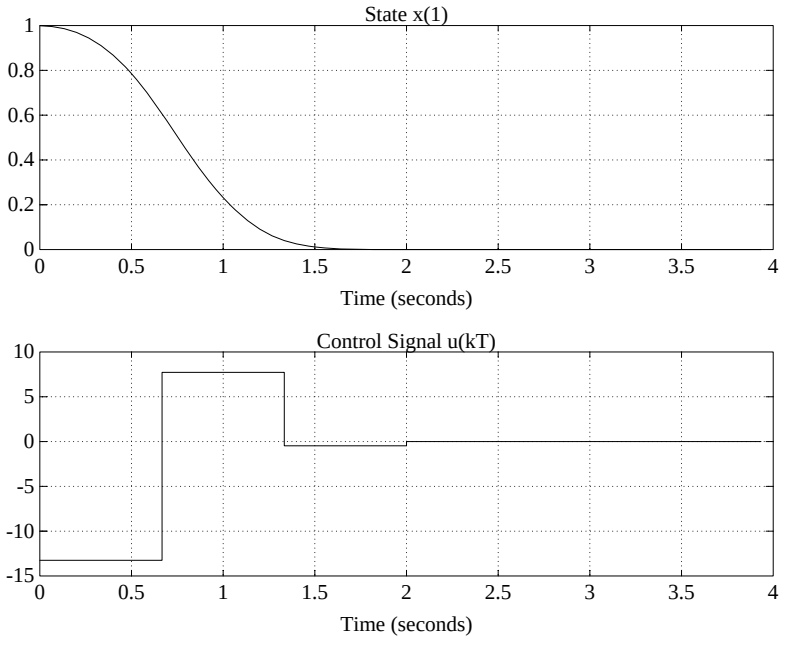


Figure 6.9 State variable x_1 and input u or the regulator of Example 6.7 with $T = 2/3$.

the regulator designed in Example 6.6. However, the disadvantage of the deadbeat regulator is that if a disturbance acts on the system, it might take a long time for the regulator to respond because the sampling interval is so large.

The Nyquist plot for the deadbeat regulator is similar in form to the Nyquist plot for the regulator in Example 6.6. However the stability margins for the deadbeat regulator are smaller: the gain margin is 3.2 dB and the phase margin is $34°$, compared with 10.8 dB and $68.5°$ for the regulator in Example 6.6.

■

6.4.3 Prototype Systems

Another way to choose closed-loop poles locations is to consider the step response of various prototype systems. For instance, let $B_k(s)$ be the k^{th}-order normalized Bessel polynomial. Table 6.2 lists the roots of several of these polynomials. An all-pole system whose poles are the roots of a Bessel polynomial is known to possess a desirable step response. The transfer function of such a system is $B_k(0)/B_k(s)$. The step responses for some of these systems are shown in Fig. 6.10. Notice that the step responses in Fig. 6.10 all have a settling time of 1 second. If a different settling time is desired, then the roots of

k	Pole locations of $B_k(s)$
1	-4.6200
2	-4.0530 $\pm$ j2.3400
3	-5.0093, -3.9668 $\pm$ j3.7845
4	-4.0156 $\pm$ j5.0723, -5.5281 $\pm$ j1.6553
5	-6.4480, -4.1104 $\pm$ j6.3142, -5.9268 $\pm$ j3.0813
6	-4.2169 $\pm$ j7.5300, -6.2613 $\pm$ j4.4018, -7.1205 $\pm$ j1.4540
7	-8.0271, -4.3361 $\pm$ j8.7519, -6.5714 $\pm$ j5.6786, -7.6824 $\pm$ j2.8081
8	-4.4554 $\pm$ j9.9715, -6.8554 $\pm$ j6.9278, -8.1682 $\pm$ j4.1057, -8.7693 $\pm$ j1.3616
9	-9.6585, -4.5696 $\pm$ j 11.1838, -7.1145 $\pm$ j 8.1557, -8.5962 $\pm$ j 5.3655, -9.4013 $\pm$ j 2.6655
10	-4.6835 $\pm$ j -12.4022, -7.3609 $\pm$ j 9.3777, -8.9898 $\pm$ j 6.6057, -9.9657 $\pm$ j 3.9342, -10.4278 $\pm$ j 1.3071

Table 6.2 Roots of normalized Bessel polynomials corresponding to a settling time of 1 second. For a settling time of T_s seconds, divide the real and imaginary parts of each root by T_s.

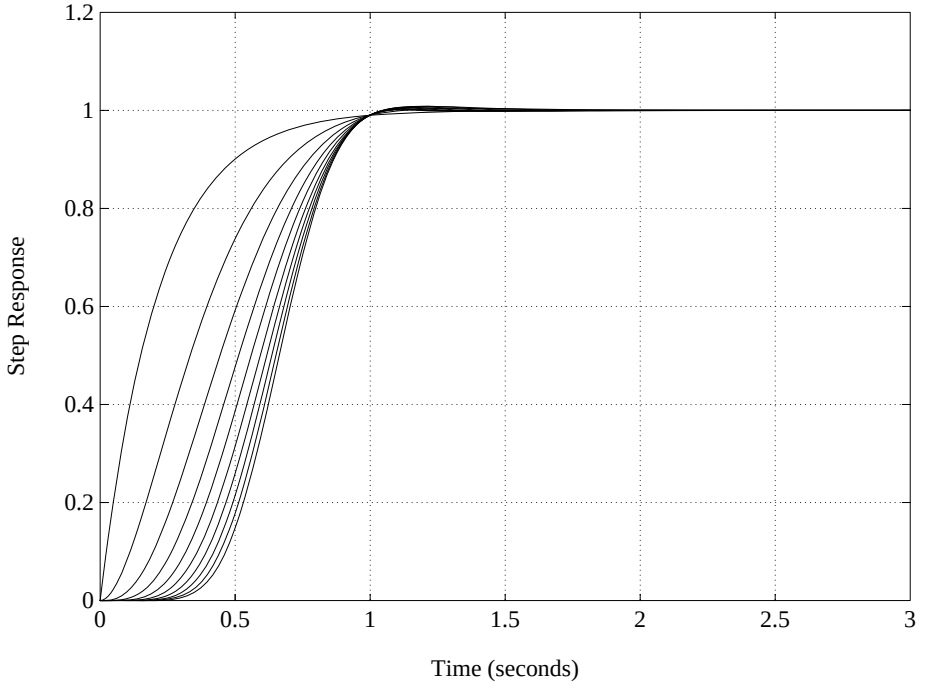


Figure 6.10 Step responses of $B_k(0)/B_k(s)$ for $k = 1, \cdots, 10$.

the Bessel polynomial must be scaled. To get a settling time of T_s seconds, the real and imaginary parts of the roots should be divided by T_s. A description of scaling a prototype system to achieve a desired speed of response is given in Chapter 5 in Section 5.7 on page 219.

To summarize, the procedure for choosing closed-loop pole locations for an n^{th}-order plant using prototype Bessel systems is as follows. Determine the desired settling time of the closed-loop system based on performance specifications, taking into account the limitations of the system hardware. Let the desired settling time be called T_s. Divide the roots of the n^{th}-order polynomial in Table 6.2 by T_s to obtain the desired closed-loop s-plane pole locations. Map these pole locations into the z-plane using the ZOH pole mapping formula. We conclude this section with an example of regulator design using Bessel polynomials. The two subsections following the example give more information on how to choose closed-loop pole locations which satisfy constraints on system hardware, and on how to choose the sampling interval.

EXAMPLE 6.8

Consider the same plant that was used in Examples 6.6 and 6.7. Regulators were designed in those examples to achieve a settling time of $T_s = 2$ seconds. We choose $T = 0.1$ in order to compare results with Example 6.6.

The desired s-plane closed-loop poles are obtained from Table 6.2 after dividing by the desired settling time

$$s_1 = \frac{-5.0093}{2} = -2.5047$$

$$s_{2,3} = \frac{-3.9668 \pm 3.7845}{2} = -1.9834 \pm j1.8922$$

The pole locations are then mapped to the z-plane using the ZOH pole-mapping formula with $T = 0.1$ to obtain

$$z_1 = e^{s_1 T} = .7784$$

$$z_{2,3} = e^{s_{2,3} T} = .8055 \pm j.1543.$$

The pole locations can be multiplied out to obtain the polynomial $p(z)$ used to calculate the feedback vector

$$p(z) = (z - z_1)(z - z_2)(z - z_3) = z^3 - 2.3893z^2 + 1.9265z - .5235.$$

Using the procedure given in Table 6.1 the vector of feedback gains is found to be

$$\mathbf{K} = [\,17.4134 \quad 11.4014 \quad 1.6358\,].$$

Fig. 6.11 shows the state variable x_1 of the regulated system as well as the control input for an initial state vector of

$$\mathbf{x}[0] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Notice that the largest value of the control signal is about -17 as opposed to -43 for the

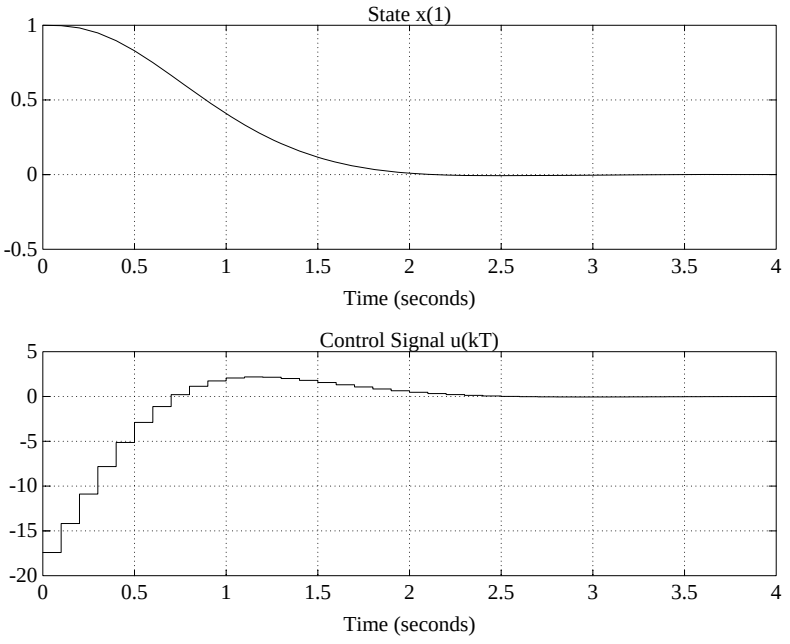


Figure 6.11 (a) State variable x_1 for the regulator of Example 6.8. (b) Control input for the regulator of Example 6.8

regulator of Example 6.6. The Nyquist plot for the regulator designed in this example is similar in form to the Nyquist plot for Example 6.6. However the stability margins for this example are

$$\text{GM} = 21.7 \text{ dB}$$

$$\text{PM} = 60^\circ$$

These are the best results so far.

■

The rules given below specify pole locations for continuous-time systems. For a digital control (discrete-time systems), these numbers, s_i, must be mapped using the ZOH pole-mapping formula, $e^{s_i T}$, where T is the sampling interval in seconds. Let `Ts` be the desired settling time for the regulator. For what follows, we note that given a continuous-time state-space model `(A,B,C,D)`. for a plant (SISO or MIMO), the zeros of the plant are calculated by the command: `tzero(A,B,C,D)`. The rules below refer to a variable `s1`, which is the first-order normalized Bessel pole with 1-second settling time. Its numerical value is `s1=-4.62`.

Regulator Poles

1. Use normalized Bessel poles scaled (divided) by the desired settling time (see Table II).
2. Use sufficiently damped plant poles; that is, plant poles whose real parts lie to the left of `s1/Ts`.
3. If the plant has complex poles that are not sufficiently damped, choose closed-loop poles by replacing the real parts of these poles with `s1/Ts` and keeping the imaginary parts the same. Changing the real parts adds damping to those plant poles, and so these are called "added damping" poles.
4. If the plant has unstable pole locations (real part greater than zero) consider the reflection of these poles about the imaginary axis, i.e. change the sign of the (real-part) of the pole. If the reflected pole is to the left of `s1/Ts` use it as a closed-loop pole.

Table 6.3 Rules for Selecting Regulator Pole Locations

6.4.4 Choosing the Sampling Interval

The development of Theorem 6.3 showed that, given a controllable continuous-time system $(\mathbf{A}, \mathbf{B})$, the sampling interval T must satisfy $T < \pi/\beta_{max}$ in order to insure that the ZOH equivalent discrete-time system is controllable. We choose to have more than the minimum number of samples for the highest frequency component of the plant, resulting in the following requirement on the sampling interval:

$$T < \frac{\pi}{5\beta_{max}}. \tag{6.43}$$

Note that if the plant has all real poles then $\beta_{max} = 0$ and (6.43) does not place any restriction on the choice of sampling interval.

The choice of sampling interval is heavily influenced by the desired regulator seconds, denoted T_s. That is, would like the regulator to drive the state vector of the plant essentially to zero within T_s seconds. The development of Theorem 6.1 showed that it is always possible to drive the state vector of an nth-order discrete-time controllable system from an arbitrary initial state $\mathbf{x}_{init}$ to an arbitrary final state $\mathbf{x}_{final}$ in n or more time steps. If fewer than n time steps are used, it may not be possible to drive the system from $\mathbf{x}_{init}$ to $\mathbf{x}_{final}$.

The purpose of a digital regulator is to drive a system from an arbitrary initial state to a state that is close to zero. That is, $\mathbf{x}_{init}$ is arbitrary and $\mathbf{x}_{final} \approx \mathbf{0}$. Based on the above discussion, we must allow at least n time steps to drive the system from $\mathbf{x}_{init}$ to $\mathbf{x}_{final}$. Suppose that the regulator settling time is specified to be T_s seconds. If the time axis is partitioned into T-second intervals, as it would be for digital control, then there are T_s/T such intervals from $t = 0$ to $t = T_s$. For an nth-order system the number of intervals must be at least n. That is, we must have

$$\frac{T_s}{T} \geq n. \tag{6.44}$$

This inequality will be modified in two ways. First, we insert an engineering "safety factor" by requiring at least $2n$ samples rather than the minimum of n. Second, we multiply by 10 to get an upper bound on the number of samples used. These two modifications yield

$$2n \leq \frac{T_s}{T} \leq 20n. \tag{6.45}$$

These inequalities may be rearranged by taking the reciprocal of every term, which requires reversing the sense of the inequalities, and then multiplying by T_s to get

$$\frac{T_s}{20n} \leq T \leq \frac{T_s}{2n}. \tag{6.46}$$

The recommended value to use from this range is the smallest value; that is,

$$T = \frac{T_s}{20n}. \tag{6.47}$$

This recommended value may be used only if it satisfies the requirement given in (6.43). Combining (6.43) and (6.47) yields the following recommendation for the sampling interval of a digital regulator:

$$T = \min\left(\frac{\pi}{\beta_{max}}, \frac{T_s}{20n}\right). \tag{6.48}$$

Note that under some circumstances (e.g. when the plant is subject to broadband random disturbances) the factor $20n$ in (6.48) may be increased to $30n$ or $40n$, which would result

in a smaller sampling interval for the digital state-feedback regulator. Also, if the control computer that implements the regulator requires a larger sampling interval, the factor $20n$ may be decreased to any value down to $2n$.

In the next example we design a state-feedback regulator for an unstable 4th-order nonlinear system. It is shown that prototype systems (e.g. Bessel polynomials) provide a simple, systematic way to design state-feedback regulators. The simplicity is due to the fact that only a single parameter has to be varied, namely the desired settling time, regardless of the dimension of the plant model. Regulators with different performance are obtained by varying T_S, and an "optimum" choice of T_S can be found by using several different values of T_S and analyzing the resulting regulators.

EXAMPLE 6.9

Consider the inverted pendulum-on-a-cart system described in Chapter 3. The nonlinear state-space description is given in (3.166). These equations can be linearized about the equilibrium point with the pendulum balanced in the upright position $\theta = \pi$ to obtain the following linear model

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 23.1 & 0 & 0 & -0.1189 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -25.0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 12.52 \\ 0 \\ 2633 \end{bmatrix} u(t). \tag{6.49}$$

The input and state variables are

u = D/A voltage from computer
x_1 = pendulum angular position - π (radians)
x_2 = pendulum angular velocity (radians/second)
x_3 = motor angular position (radians)
x_4 = motor angular velocity radians/second).

The numerical values in (6.49) describe an actual hardware system built at the University of Rhode Island. Since the D/A converter in this system has an output range of ± 5 volts, the model in (6.49) is only valid if $|u(t)| \leq 5$.

In order to design a regulator for this system, we assume that the worst-case initial state vector is

$$\mathbf{x}_{\text{WC}} = \begin{bmatrix} 0.17 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

which corresponds to the pendulum at an initial angle of 10° (0.17 radians) from the vertical, and the cart in the center of the track with zero velocity. We can calculate an analog feedback vector $\mathbf{L}_a$ for the system in (6.49) to place the closed-loop poles at the rots of a 4th-order Bessel polynomial scaled to achieve some value of settling time T_S. We then simulate the behavior of the closed-loop analog system with $\mathbf{x}(0) = \mathbf{x}_{\text{WC}}$. The non-linear state-space equations are used in this simulation. For $T_S = 0.95$, the state-feedback vector is

$$\mathbf{K}_a = [\,22.2656 \quad 4.5484 \quad -0.0281 \quad -0.0235\,].$$

Plots of pendulum position and the input signal are shown in Fig. 6.12. Notice that

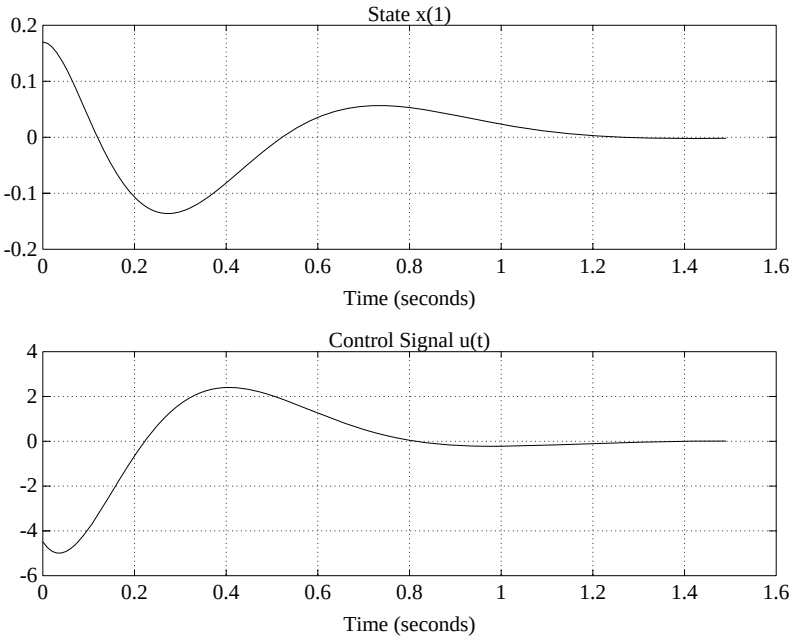


Figure 6.12 (a) Pendulum position for the analog regulator of Example 6.9. (b) Input signal.

the peak magnitude of the input signal is 5. If the closed-loop poles were chosen to achieve $T_S < 0.95$, the peak magnitude of the input signal would be greater than 5. On the other hand, if T_S were chosen to be greater than 0.95, the peak magnitude of the input signal would be less than 5. Thus by calculating feedback vectors corresponding to different values of T_S and simulating the resulting closed-loop systems with $\mathbf{x}(0) = \mathbf{x}_{\text{WC}}$, we find the value $T_S = 0.95$ to be acceptable.

Let $\mathbf{K}_a$ be the analog feedback vector corresponding to $T_S = 0.95$. We need to find the bandwidth of the analog regulator in order to choose the sampling interval for the digital regulator. The magnitude of the closed-loop transfer function

$$H(j\omega) = \mathbf{K}_a(j\omega\mathbf{I} - (\mathbf{A} - \mathbf{b}\mathbf{K}_a))^{-1}\mathbf{b}$$

is shown in Fig. 6.13 (see Section 6.4.4). The closed-loop bandwidth is about 20 radians/second. If we use the rule-of-thumb given in (6.62), the sampling interval should satisfy

$$400 \leq \frac{2\pi}{T} \leq 800 \text{ radians/second}$$

or

$$64 \leq \frac{1}{T} \leq 128 \text{ Hz.}$$

We choose the sampling frequency $1/T = 100$ Hz or $T = 0.01$ seconds.

We are now ready to calculate the feedback vector for the digital regulator using the procedure in Table 6.1. The s-plane pole locations for Step 1 are the roots of a 4th-order Bessel polynomial scaled to achieve $T_S = 0.95$. The sampling interval is 0.01. The feedback vector can be calculated to be

$$\mathbf{K} = [\,23.3255 \quad 4.7691 \quad -0.0288 \quad -0.0240\,].$$

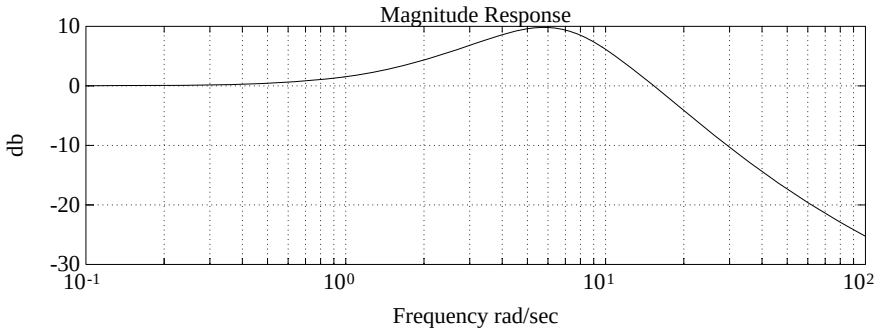


Figure 6.13 Bode magnitude plot for the closed-loop analog regulator of Example 6.9.

We can calculate the stability margins of this regulator by examining the Nyquist plot of $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{L})$. A portion of the Nyquist plot is shown in Fig. 6.14. This regulator has

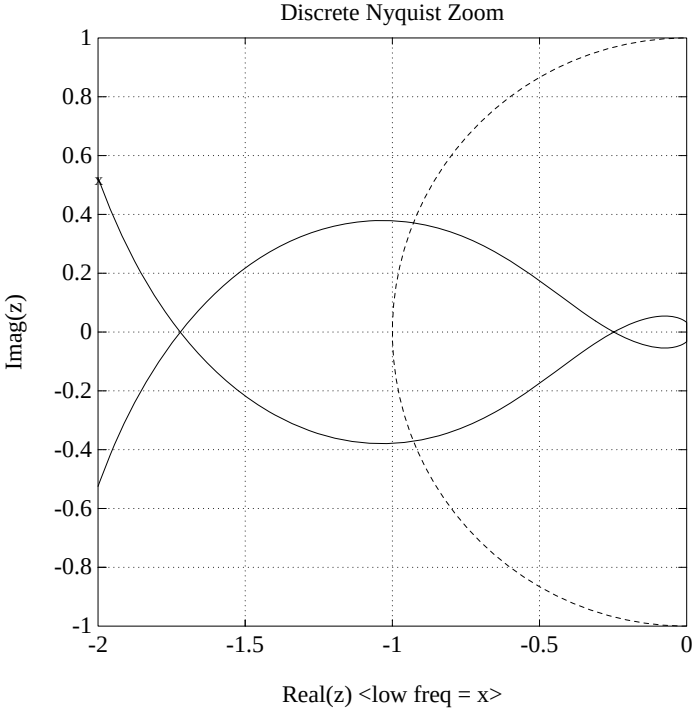
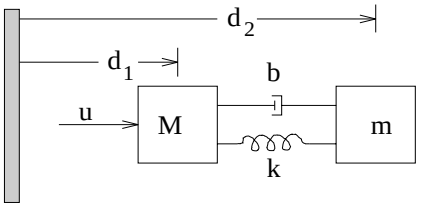


Figure 6.14 A portion of the Nyquist plot for the digital regulator in Example 6.9. The dashed line is the unit circle.

upper and lower gain margins

$$-4.8 \leq \text{ GM } \leq 12.2 \text{ dB}$$

and a phase margin of 22°. These margins are acceptable and the design is complete. ■



m,M = masses
b = damping coefficient
k = spring constant
d_1 = position of mass M
d_2 = position of mass m
u = applied control force

Figure 6.15 The plant for Example 6.10.

.

In the next example, it is shown that the roots of a Bessel polynomial are not always an appropriate choice for closed-loop pole locations. However, insight into the physical meaning of the closed-loop pole locations can be used to guide their selection. A system with a structural resonance is considered in the next example. It is shown that a regulator which adds damping to the resonance, but does not alter its frequency, is superior to a regulator designed using only Bessel polynomials.

EXAMPLE 6.10

Consider the system in Fig. 6.15. The input to the system is the control force u and the output is the position of the mass m. We are trying to control the position of m through a flexible coupling consisting of a spring and damper. If the damping is small ($b \ll k$) then the system will exhibit a structural resonance with natural frequency $\sqrt{k/m}$ radians/second. The state variables for this system are chosen as follows

$$
\begin{aligned}
x_1 &= d_1 \\
x_2 &= \dot{x_1} \\
x_3 &= d_2 \\
x_4 &= \dot{x_3}.
\end{aligned}
$$

With this choice of state variables, the state-space model for the system is

$$
\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{M} & -\frac{b}{M} & \frac{k}{M} & \frac{b}{M} \\ 0 & 0 & 0 & 1 \\ \frac{k}{m} & \frac{b}{m} & -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ 0 \end{bmatrix} u(t)
$$

$$
y(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}(t).
$$

The parameters for the system are

$$
\begin{aligned}
M &= 20 \text{ kg} \\
m &= 1 \text{ kg} \\
k &= 25 \text{ N/m} \\
b &= 0.2 \text{ N-sec/m}.
\end{aligned}
$$

The desired settling time is $T_S = 3$ seconds, the sampling interval is $T = 0.1$ seconds, and the initial state vector is

$$\mathbf{x}(0) = \begin{bmatrix} 0 \\ 0 \\ .01 \\ 0 \end{bmatrix}.$$

In this example, we do not go into detail about the choices of T_S and T as we did in the previous example. The point of this example is to show that for a given value of T_S, the roots of a Bessel polynomial may not be the most appropriate choice for closed-loop pole locations.

The eigenvalues of the $\mathbf{A}$ matrix for this system can be calculated to be

$$0, 0, -0.1050 \pm j5.1224.$$

From Theorem 6.3 on page 236 the sampling interval should satisfy

$$T < \frac{\pi}{5.1224} = 0.61$$

to insure that the ZOH equivalent is controllable. The choice of T given above satisfies this inequality. The eigenvalues of $\mathbf{A}$ show that the plant has complex-conjugate poles near the $j\omega$ axis (i.e. lightly damped) with a natural frequency of 5.1224 radians/second.

With T_S and T chosen above, we can go ahead and design a digital regulator which places the closed-loop poles at the roots of a 4th-order Bessel polynomial scaled to achieve a settling time of 3 seconds. The scaled roots in the s-plane are

$$-1.3385 \pm 1.6908 \text{ and } -1.8427 \pm 0.5518. \tag{6.50}$$

After mapping these poles into the z-plane with the ZOH pole-mapping formula, we obtain the following feedback vector for the digital regulator

$$\mathbf{K} = \begin{bmatrix} -231.3856 & 82.7142 & 241.7470 & -65.9406 \end{bmatrix}.$$

Plots of $y(t)$ and $u(t)$ for the regulated system are shown in Fig. 6.16. The behavior of this regulator will be compared with that of another regulator designed later in this example.

Note that the poles in (6.50) are well damped (large negative real parts) but the imaginary parts are quite different than the imaginary part of the lightly-damped poles of the plant. This means that we are asking the regulator to change the effective resonant frequency of the plant. A more realistic objective is to keep the resonant frequency of the closed-loop system the same as the open-loop system but simply add more damping to the resonant poles. In order to achieve a settling time of 3 seconds, the real part c of the closed-loop resonant poles should satisfy

$$3 \approx \frac{-4.6}{c}$$

or $c \approx -1.5$ (see the discussion following (5.17) on page 196). The imaginary parts of these closed-loop poles will be chosen to be the same as the imaginary parts of the open-loop resonant poles. The remaining two closed-loop poles will be chosen to

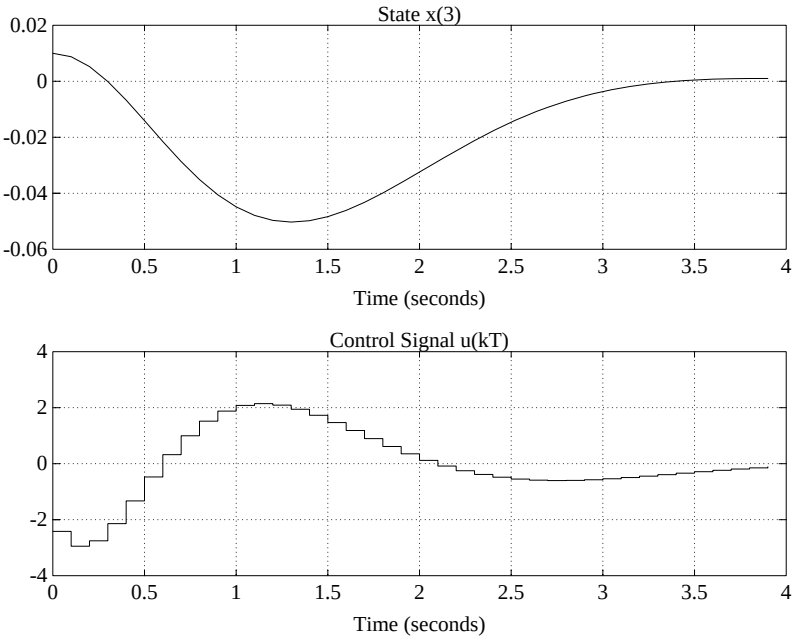


Figure 6.16 (a) The output of the regulated system in Example 6.10 with closed-loop poles specified by a 4th-order Bessel polynomial. (b) The control input.

be the roots of a 2nd-order Bessel polynomial scaled to achieve $T_S = 3$. Thus the complete set of closed-loop s-plane pole locations is

$$\left.\begin{matrix} -1.5 + j5.1224 \\ -1.5 - j5.1224 \end{matrix}\right\} \text{ resonant poles pulled further into the left half-plane}$$

$$\left.\begin{matrix} -1.3510 + j0.7800 \\ -1.3510 - j0.7800 \end{matrix}\right\} \text{ 2nd-order Bessel poles with } T_S = 3 \quad .$$

These poles can be mapped into the z-plane using the ZOH pole mapping formula and the resulting state feedback vector for the digital regulator is

$$\mathbf{K} = [\,130.4705 \quad 92.3476 \quad -88.2016 \quad -38.9959\,]\,.$$

Plots of $y(t)$ and $u(t)$ for the closed-loop system under digital control are shown in Fig. 6.17. Notice that the magnitude of $u(t)$ is much less than the corresponding magnitude in Fig. 6.16. Although $y(t)$ in Fig. 6.17 is more oscillatory than $y(t)$ in Fig. 6.16, the oscillations are well damped and are realistic for a plant which has structural flexibility.

Finally, we can compare the stability margins for the two regulators. The regulator whose closed-loop poles were specified by a 4th-order Bessel polynomial has a gain

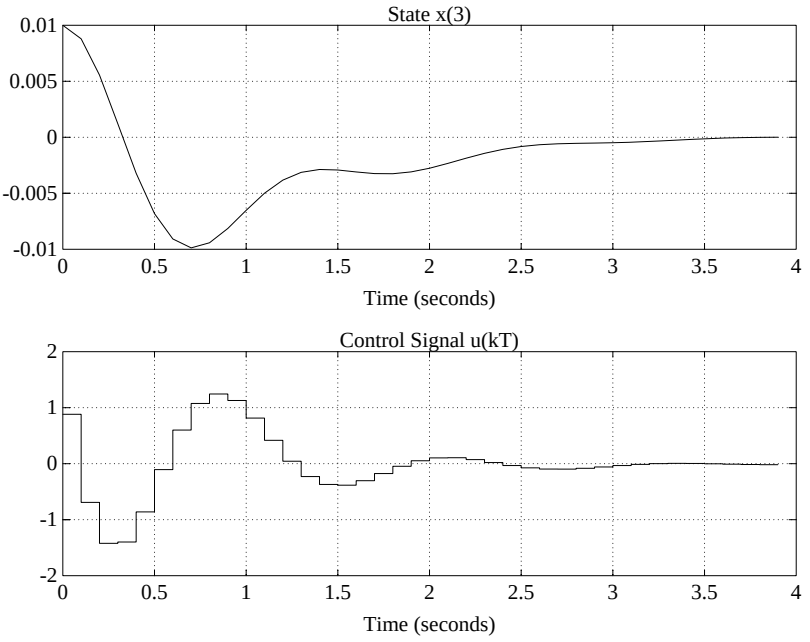


Figure 6.17 (a) The output of the regulated system in Example 6.10 with two closed-loop poles chosen to add damping to the structural resonance, and the other two closed-loop poles chosen as the roots of a 2nd-order Bessel polynomial. (b) The control input.

margin of 3.6 dB and a phase margin of 29°. The regulator based on added damping and a 2nd-order Bessel polynomial has a gain margin of 12.7 dB and a phase margin of 59°. Thus the "added damping" regulator is superior to the "4th-order Bessel" regulator.

■

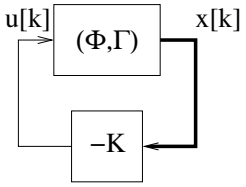
6.5 Adding a Reference Input

This chapter has dealt with the design of regulators which, by definition, have no reference input. In this section, we consider a simple way to modify a regulator to obtain a system in which a single state variable is made to track a reference input. The general tracking problem is developed in Chapter 8.

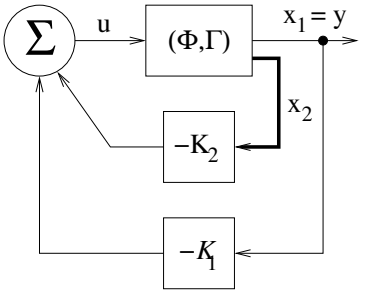
In this section we make the assumption that x_1 is the output of the plant, and that a reference input is available. The modification of the regulator system to include the reference input is shown in a sequence of block diagrams in Fig. 6.18. In those Figures, the feedback vector $\mathbf{K}$ is partitioned into its first element K_1 and the remaining $n-1$ elements $\mathbf{K}_2$ as follows

$$\mathbf{L} = [\, K_1 \quad \mathbf{K}_2 \,].$$

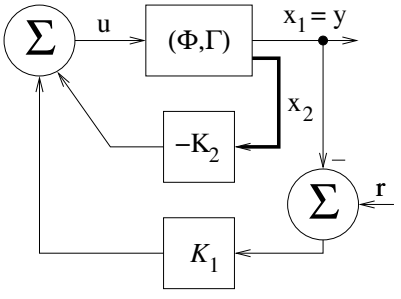
The state vector is also partitioned into its first element x_1 and the remaining elements $\mathbf{x}_2$. Note that in figure (c), the negative sign on l_1 is moved back to the summing junction. The system in Fig. 6.18(d) in the form of a negative unity-feedback system with output $y = x_1$. This system can be modified to track some other state variable besides x_1. For example,



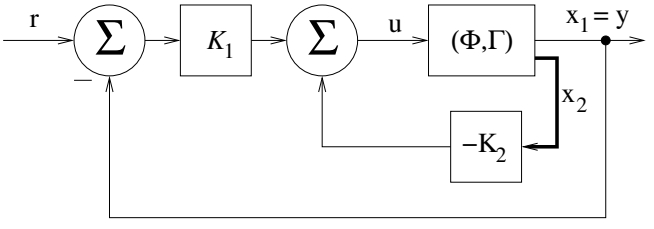
(a)



(b)



(c)



(d)

Figure 6.18 (a) A digital regulator; (b) A regulator with x_1 shown separately from the other state variables; (c) A reference input supplied for x_1; (d) The system in (c) re-drawn in the form of a standard tracking system.

if we want x_j to track a reference input, then the gain K_1 in Fig. 6.18(d) is replaced by l_j and $\mathbf{K}_2$ is defined as the vector $\mathbf{K}$ with the j-th element removed. Similarly, $\mathbf{x}_2$ is defined as the state vector $\mathbf{x}$ with the j-th element removed.

The equations that describe the system in Fig. 6.18(d) are as follows. The ZOH of the plant is

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k]. \tag{6.51}$$

The feedback connection is

$$\begin{aligned} u[k] = & -\mathbf{K}_2\mathbf{x}_2[k] + K_1(r[k] - x_1[k]) \\ = & -\mathbf{K}\mathbf{x}[k] + K_1 r[k]. \end{aligned} \tag{6.52}$$

The equation for the closed-loop system is obtained by substituting (6.52) into (6.51) to get

$$\mathbf{x}[k+1] = (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})\mathbf{x}[k] + K_1\mathbf{\Gamma}r[k]. \tag{6.53}$$

Since the feedback vector $\mathbf{K}$ is chosen to make the closed-loop system stable, (6.53) reaches a finite steady-state value, call is $\mathbf{x}_s$, in response to a unit step input. We can find $\mathbf{x}_s$ by substituting $r[k] = 1$ and $\mathbf{x}[k] = \mathbf{x}[k+1] = \mathbf{x}_s$ into (6.53) to get

$$\mathbf{x}_s = (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})\mathbf{x}_s + K_1\mathbf{\Gamma}. \tag{6.54}$$

We can solve for $\mathbf{x}_s$ from this equation as follows

$$\mathbf{x}_s = K_1(\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{K})^{-1}\mathbf{\Gamma}. \tag{6.55}$$

We know that the matrix in parentheses in the above equation is invertible because the eigenvalues of $(\mathbf{\Phi}-\mathbf{\Gamma}\mathbf{K})$ are within the unit circle, and so the eigenvalues of $[\mathbf{I}-(\mathbf{\Phi}-\mathbf{\Gamma}\mathbf{K})]$ cannot equal zero (see Facts 2.30 and 2.37 in Chapter 2). Thus the solution for $\mathbf{x}_s$ in (6.55) is unique. Using (6.55) we can compute the steady-state value of the output in response to a unit step input to be

$$y_s = \mathbf{c}\mathbf{x}_s = K_1\mathbf{e}_1^T(\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{K})^{-1}\mathbf{\Gamma} \tag{6.56}$$

where

$$\mathbf{c} = \mathbf{e}_1^T = [\,1 \quad 0 \quad \cdots \quad 0\,]$$

because we have assumed that the output of the system is x_1.

We now investigate conditions under which $y_s = 1$ (which implies zero steady-state error for a unit step input). We know from section 5.8 in Chapter 5 that a negative unity-feedback system will have zero steady-state error to a step input if the transfer function in the forward path has a pole a $z = 1$. A commonly occurring case is when the ZOH of the plant has a pole at $z = 1$ (i.e. $\mathbf{\Phi}$ has an eigenvalue equal to 1) and the corresponding eigenvector of $\mathbf{\Phi}$ is $\mathbf{e}_1$ so that

$$\mathbf{\Phi}\mathbf{e}_1 = \mathbf{e}_1. \tag{6.57}$$

This can occur, for example, when the plant is a Type-1 servo system and x_1 represents output position (see the example at the end of this section). In this case the system shown in Fig. 6.18(d) will have zero steady-state error to a step input, as shown below. We first

show that $\mathbf{x}_s = \mathbf{e}_1$ is the solution to (6.55) by substituting into the equivalent equation (6.54)

$$\begin{aligned}\mathbf{e}_1 &= (\mathbf{\Phi} - \mathbf{\Gamma K})\mathbf{e}_1 + K_1\mathbf{\Gamma} \\ &= \mathbf{\Phi e}_1 - \mathbf{\Gamma K e}_1 + K_1\mathbf{\Gamma} \\ &= \mathbf{e}_1 - K_1\mathbf{\Gamma} + K_1\mathbf{\Gamma} \\ &= \mathbf{e}_1\end{aligned}$$

where (6.57) was used to go from the second to third line of the above equation. If $\mathbf{x}_s = \mathbf{e}_1$, then

$$y_s = \mathbf{c}\mathbf{x}_s = \mathbf{e}_1^T\mathbf{e}_1 = 1$$

and the system in Fig. 6.18(d) has zero steady-state error to a unit step input. Achieving zero steady-state error in this case is due to the fact that the plant contains an integrator. It is not due to any special values for the feedback gains.

If the plant does not contain an integrator so that (6.57) is not satisfied then y_s given by (6.56) will not equal 1. In this case we can multiply the reference input by a gain g equal to the reciprocal of the number given by (6.56); that is

$$g = \frac{1}{l_1\mathbf{e}_1^T(\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma K})^{-1}\mathbf{\Gamma}}. \tag{6.58}$$

By scaling the reference input in this way, the system will have zero steady-state error to a step input. Achieving zero steady-state error in this case is due to the special value of g given by (6.58). However, if the mathematical model of the plant is not accurate then the value of g given by (6.58) will not result in zero steady-state error when the control system is connected to the actual plant. A better way to proceed in this case is to use integral control, a topic which is covered in Chapter 8. We conclude this section with an example in which a reference input is added to a regulator to obtain a tracking system.

EXAMPLE 6.11

Consider the plant that was used in Examples 6.6–6.8. The transfer function is

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

and the state-space description is

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -4 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

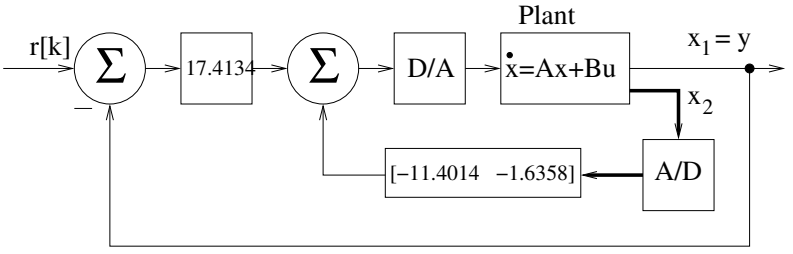
This plant contains an integrator (pole at $s = 0$). The ZOH equivalent of this plant at sampling interval $T = 0.1$ is

$$\mathbf{\Phi} = \begin{bmatrix} 1.0000 & 0.0952 & 0.0042 \\ 0 & 0.9048 & 0.0782 \\ 0 & 0 & 0.6703 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0001 \\ 0.0042 \\ 0.0824 \end{bmatrix}.$$

Notice that $\mathbf{\Phi}$ has an eigenvalue equal to 1 and that the corresponding eigenvector is $\mathbf{e}_1$ so that $\mathbf{\Phi e}_1 = \mathbf{e}_1$. The feedback vector calculated in the regulator design of Example 6.8 is

$$\mathbf{K} = [\,17.4134 \quad 11.4014 \quad 1.6358\,].$$

This vector can be partitioned as shown in Fig. 6.18(d) to obtain the tracking system shown below.



The response of this system to a unit step input is shown in Fig. 6.19.

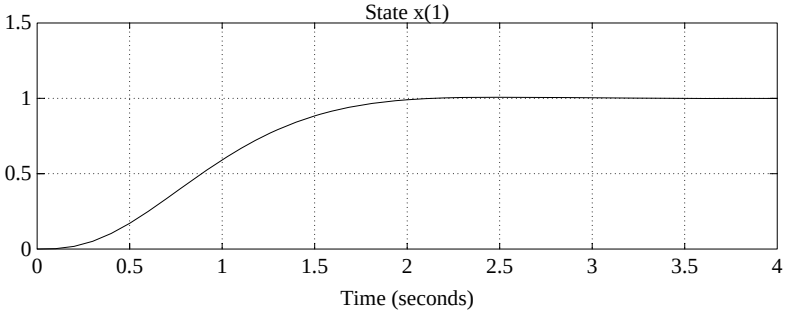


Figure 6.19 Step response for the tracking system of Example 6.11

■

6.6 Relationship Between State Feedback and PID Control*

The previous section showed how a state-feedback regulator could be modified to accept a reference input. In this section we show that the resulting control system is related to a classical PD or PID controller (defined below). The relationship between state feedback and PD or PID control is derived in continuous time for ease of presentation.

A PID control system is shown in Fig. 6.20. The *proportional gain* K_P multiplies the error signal $e(t)$, the *integral gain* K_I multiplies the integral of the error signal, and the *derivative gain* K_D multiplies the derivative of the error signal. The design of PID controllers does not require a mathematical model of the plant. Rather, the gains can be adjusted in accordance with qualitative observations about their effect on the step response. These observations are as follows: increasing K_P decreases the rise time, but may cause oscillations in the step response. Increasing K_D adds damping and reduces overshoot. The integral term insures zero steady-state error to a step input and rejects constant disturbances, however increasing K_I increases the settling time. The implementation of a PD or PID controller requires some care because of the derivative term. For example, a step

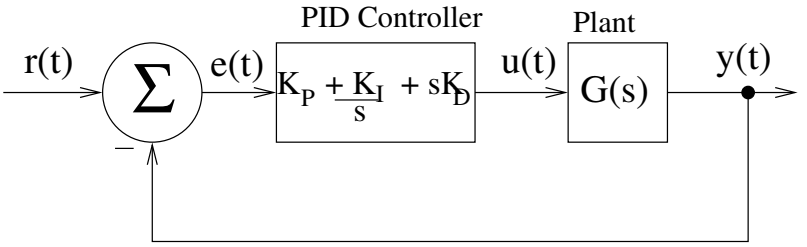


Figure 6.20 A PID control system.

input causes the error signal to be discontinuous at $t = 0$, and the derivative term produces an input to the plant which is theoretically infinite at $t = 0$. In practice the input to the plant is limited to some finite value due to saturation in the circuit which implements the controller.

The PID gains can be adjusted in accordance with the above observations until a satisfactory step response is achieved. There are also "tuning rules" for choosing the PID gains [31]. In this section we show that if the plant is described by a second-order transfer function and if a certain state-space model is used, then the feedback gains of a state-feedback control system are equivalent to the gains of a PID controller.

Suppose that the plant to be controlled is described by the following second-order transfer function relating the input u and the output y:

$$G(s) = \frac{b_2}{s^2 + a_1 s + a_2}. \tag{6.59}$$

This transfer function has a constant numerator, and is a special case of the general form considered in Chapter 3. Many physical systems have a transfer function of this form, e.g. simple electromechanical systems. If we want to obtain a state-space description for the system described by $G(s)$, we can always pick the following state variables:

$$x_1 = y$$

$$x_2 = \dot{x}_1$$

and the resulting state-space description for $G(s)$ is

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ b_2 \end{bmatrix} u(t) \\ y(t) &= [\,1 \quad 0\,]\, \mathbf{x}(t). \end{aligned} \tag{6.60}$$

We can choose desired closed-loop s-plane pole locations and calculate the corresponding vector of feedback gains $\mathbf{K}$ for this state-space model. The analog state-feedback regulator can be re-arranged as in Fig. 6.18 on page 269 to obtain the control system shown in Fig. 6.21. We are assuming that the plant is described by the state-space model shown in (6.60) so that $x_2 = \dot{x}_1 = \dot{y}$. That is, x_2 is the derivative of the system output y. This differentiation can be represented in a block diagram by the Laplace variable s as shown in Fig. 6.22(a). By re-arranging the block diagram to that shown in Fig. 6.22(c), we see that the state feedback controller can be implemented as a PD controller with a unity dc

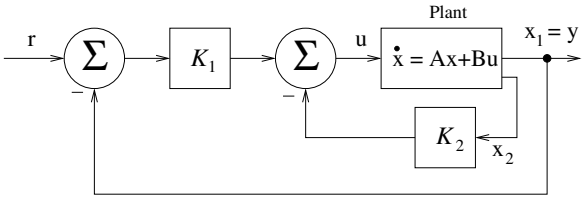


Figure 6.21 An analog state-feedback regulator rearranged as a tracking system for a second-order plant.

gain pre-filter on the reference input. For classical PD control this pre-filter is simply unity. Comparing the PID controller shown in Fig. 6.20 with the PD controller shown in Fig. 6.22(c), we see that the state feedback gains are related to the PD gains as follows:

$$K_P = K_1$$

$$K_D = K_2.$$

Thus if the plant has a second-order transfer function as shown in (6.59) and the state-space model (6.60) is used, then the state feedback gains can be used as the proportional and derivative gains in PD control. This is illustrated in the next example.

EXAMPLE 6.12

The plant is a Type-1 servo with input $u(t)$, output $y(t)$, and transfer function

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

The state-space model corresponding to state variables $x_1 = y, x_2 = \dot{x}_1$ is (see (6.60))

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

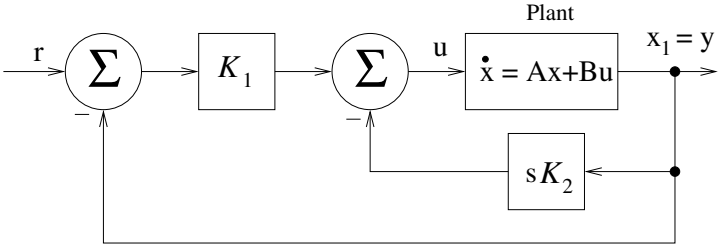
Suppose the desired settling time is 1 second. We choose the closed-loop poles as the roots of a second-order Bessel polynomial from Table 6.2, and calculate a vector of feedback gains $\mathbf{K}$ such that the eigenvalues of $\mathbf{A} - \mathbf{bK}$ are $-4.0530 \pm j2.3400$. The resulting feedback gains are

$$K_1 = 21.9024\,, \quad K_2 = 7.1060.$$

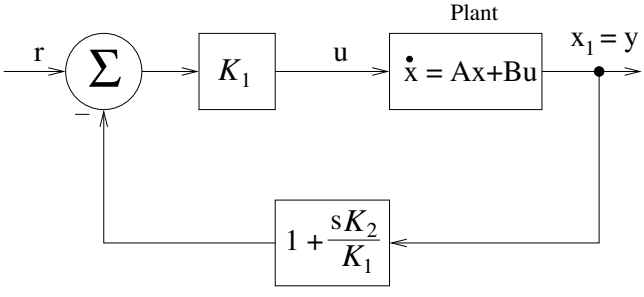
These gains are used in the control system shown in Fig. 6.22(c) with and without the pre-filter. With the pre-filter in place this control system is equivalent to a state-feedback system. Without the pre-filter the system in Fig. 6.22(c) is a classical PD control system. The step responses and plant inputs are shown in Fig. 6.23. Note that the plant input generated by the PD controller is very large (off the scale) for $t < 0.05$ seconds.

■

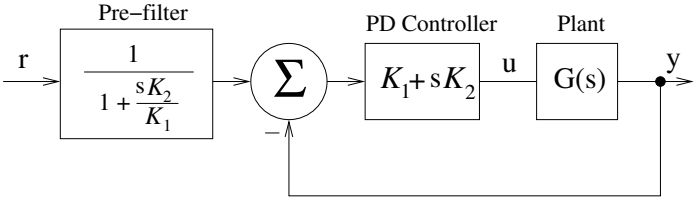
If the plant is described by a transfer function $G(s)$ shown in (6.59) and $a_2 \neq 0$, then the control system shown in Fig. 6.22(c) will have a non-zero steady-state error to a step



(a)



(b)



(c)

Figure 6.22 (a) A control system that is equivalent to the state-feedback system shown in Fig. 6.21. (b) A block diagram equivalent to (a). (c) A rearrangement of block diagram (b) in the form of a PD controller with a prefilter.

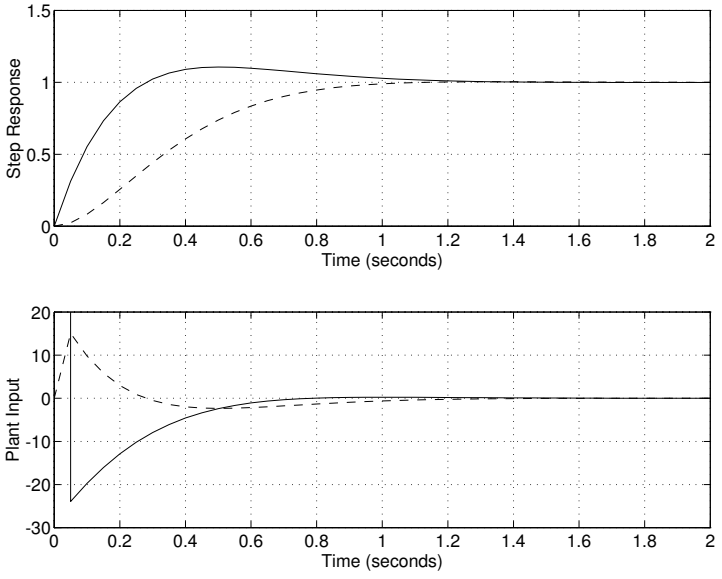


Figure 6.23 Step response and plant input for Example 6.12. The solid lines are for the PD controller, while the dashed lines are for the PD controller with prefilter.

input. From Section 5.8 we know that we require an integrator in the forward path to achieve zero steady-state error for a step input. In order to insure that the forward path contains an integrator we can proceed as follows. First augment the plant transfer function $G(s)$ with an integrator. Then calculate state feedback gains from a state-space model for the augmented third-order system. Finally, implement the integrator as part of the compensator. The augmented system is shown in Fig. 6.24, along with state feedback gains. The transfer function $G(s)$ followed by an integrator is a cascade connection of two systems, and a state-space model for the combined system can be obtained from state-space models of the individual systems using (3.139). In order to derive a PID controller we use the state-space model for $G(s)$ given in (6.60). We use the following state-space model for an integrator: $(\mathbf{A}, \mathbf{B}, \mathbf{C}, D) = (0, 1, 1, 0)$. Then a state-space model for the cascade connection of $G(s)$ and an integrator is (we use the subscript 'D' to denote design model):

$$\mathbf{A}_D = \begin{bmatrix} 0 & 1 & 0 \\ -a_2 & -a_1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_D = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{C}_D = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}. \tag{6.61}$$

Using this design model, we can choose three desired closed-loop s-plane pole locations, and calculate the corresponding feedback vector $\mathbf{K}$. The three gains in $\mathbf{K}$ are shown in Fig. 6.24.

We can introduce a reference input in the system as shown in Fig. 6.25(a), and manipulate the block diagram to obtain the system shown in Fig. 6.25(c). Note that the negative sign on l_3 in Fig. 6.25(a) has been pushed back to the summing junction. This last figure shows that a state-feedback control system with an integrator added to the forward path can be implemented as a PID control system with a pre-filter. For classical PID control this pre-filter is simply unity.

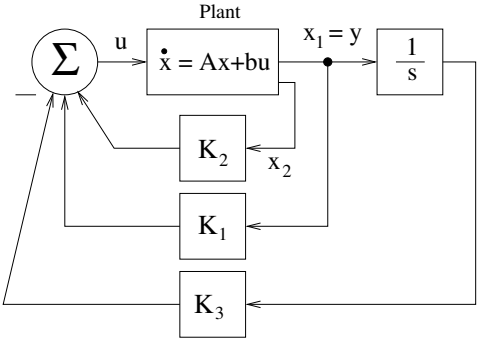
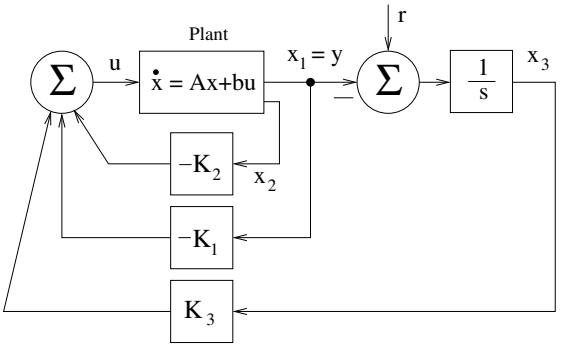
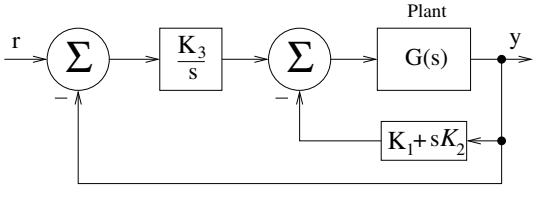


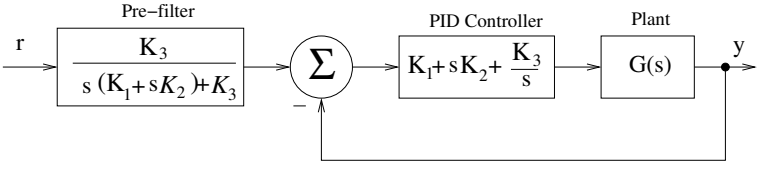
Figure 6.24 An analog state-feedback regulator a second-order plant with an additional integrator.



(a)



(b)



(c)

Figure 6.25 (a) An analog state-feedback regulator with an additional integrator and reference input. (b) Rearrangement of the block diagram in (a). (c) Rearrangement of block diagram (b) in the form of a PID controller with a prefilter.

EXAMPLE 6.13

The plant is a dc motor with input voltage $u(t)$ in volts and output velocity $y(t)$ in radians per second. The transfer function is

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

where the two time constants represent the mechanical and electrical time constants of the motor [15]. Using (6.60) we obtain the following state-space model:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -4 & -5 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

We want to design a control system which causes the motor to achieve a desired constant speed. We add an integrator to the plant in order to have zero steady-state error to a step input, and also to reject a constant disturbance caused, for example, by a load torque. Using (6.61) the state-space design model is

$$\mathbf{A}_D = \begin{bmatrix} 0 & 1 & 0 \\ -4 & -5 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_D = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{C}_D = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

Suppose the desired settling time is 1 second. We choose the closed-loop poles as the roots of a third-order Bessel polynomial from Table 6.2, and calculate a vector of feedback gains $\mathbf{K}$ such that the eigenvalues of $\mathbf{A} - \mathbf{BK}$ are $-5.0093, -3.9668 \pm j3.7845$. The resulting feedback gains are

$$K_1 = 65.7997\,, \quad K_2 = 7.9429\,, \quad K_3 = 150.5688.$$

These gains are used in the control system shown in Fig. 6.25(c) with and without the pre-filter. With the pre-filter in place this control system is equivalent to a state-feedback system. Without the pre-filter the system in Fig. 6.25(c) is a classical PID control system. The step responses and plant inputs are shown in Fig. 6.26. Note that the plant input generated by the PID controller is very large (off the scale) for $t < 0.05$ seconds.

■

6.7 Choosing the Sampling Interval Based on Bandwidth

In the next two subsections we show how the sampling interval of a digital state-feedback regulator may be chosen based on the bandwidth of a corresponding analog regulator.

6.7.1 Satisfying Constraints

The methods for choosing closed-loop pole locations, namely dominant poles and prototype systems, specify poles in the s-plane which are then mapped to the z-plane. Whenever s-plane poles are specified, it is useful to check whether or not an *analog regulator* with the specified closed-loop poles satisfies certain constraints. If the analog regulator does not satisfy the constraints, there is no sense mapping those pole locations to the z-plane for

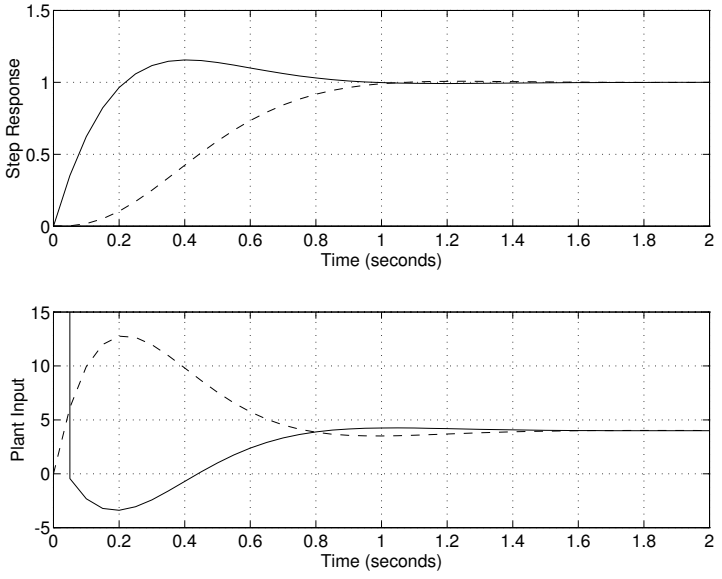


Figure 6.26 Step response and plant input for Example 6.13. The solid lines are for the PID controller, while the dashed lines are for the PID controller with prefilter.

the digital regulator. Rather, the s-plane pole locations must first be modified to satisfy the constraints.

Suppose the plant to be regulated is described by the state-space equations

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

An analog regulator uses continuous-time state feedback $u(t) = -\mathbf{K}_a\mathbf{x}(t)$. The poles of the closed-loop system are the eigenvalues of $\mathbf{A} - \mathbf{b}$
$Bb\mathbf{K}_a$. One constraint is that the magnitude of the input signal should not exceed 5, assuming the digital regulator will use a D/A converter with an output range of $\pm$ 5 volts. This constraint also assumes that the model for the plant includes the largest amplifier we are willing to use between the D/A converter and the input to the actual plant. A second constraint on the analog regulator is that it has acceptable stability margins.

A convenient feature of regulator design using prototype systems (Bessel polynomials) is that possible closed-loop pole locations are parameterized by a single number, T_S, regardless of the order of the plant model. Thus we can design a family of analog regulators, one for each value of T_S in some range, and check if the constraints are satisfied. To check the stability margins, we need to look at the Nyquist plots for the continuous-time system $(\mathbf{A}, \mathbf{B}, \mathbf{K}_a)$, where $\mathbf{L}_a$ is the feedback vector which places the eigenvalues of $\mathbf{A} - \mathbf{B}\mathbf{K}_a$ at the roots of a Bessel polynomial for some value of T_S.

It is not so simple to check the constraint on the magnitude of the input signal. The reason is that the input signal depends on the value of $\mathbf{x}(0)$ and also on disturbances acting on the plant. For simplicity, we only consider the effect of the initial state. If we pick a "worst case" initial state vector $\mathbf{x}_{\mathrm{WC}}$, say one that is as far away from the origin as we

expect the regulator to have to handle[1], then we can simulate an analog regulator designed for some settling time T_S with initial state $\mathbf{x}_{\text{WC}}$. Note that the choice of $\mathbf{x}_{\text{WC}}$ is problem dependent; no general results can be given. If the simulation shows that the control input exceeds 5 volts for a given $\mathbf{x}_{\text{WC}}$, we pick a larger value of T_S, calculate a new feedback vector, and run another simulation. If the simulation shows that the control input is always less than 5, we can pick a smaller value of T_S. In this way we can find an analog regulator whose control input is less than 5 for the initial state $\mathbf{x}_{\text{WC}}$. We can also check the stability margins of each of the analog regulators considered.

Once an acceptable analog regulator is found (i.e. an acceptable value of T_S), we design a digital regulator to have the same T_S. In the next subsection, we show how to choose the sampling interval for this digital regulator. An example which illustrates the procedures for choosing T_S and T is given at the end of the next sub-section.

6.7.2 Choosing the Sampling Interval

Assume that a feedback vector $\mathbf{K}_a$ has been found so that the analog regulator has acceptable stability margins and satisfies the constraint on the magnitude of the control input. When Bessel polynomials are used, $\mathbf{K}_a$ corresponds to some choice of T_S. In order to choose the sampling interval T for a digital regulator, we must first find the *bandwidth* of the analog regulator. Bandwidth has been defined for *tracking systems* in Chapter 5. Since a regulator does not have an external input, it is unclear what the definition of bandwidth should be. The bandwidth of a regulator is a measure of the frequency content of the control input signal which the regulator generates in response to a non-zero initial conditions or disturbances. However, we do not want the definition of bandwidth to depend on the value of a specific initial condition or disturbance. Thus we proceed as follows.

A regulator can be represented by the block diagram shown in Fig. 6.27. This figure is in

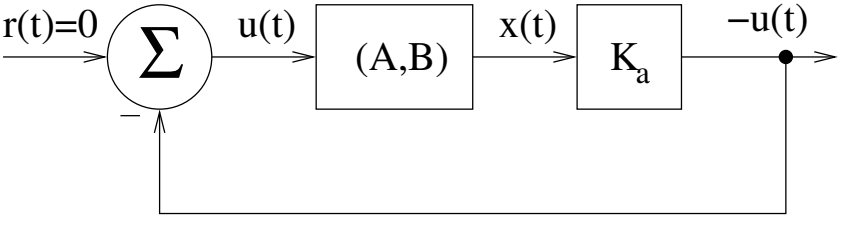


Figure 6.27 A state-feedback regulator shown as a single input, single output tracking system. The bandwidth of the regulator is defined to be the bandwidth of this tracking system.

the form of a tracking system with the signal $-u(t)$ as the output. If we let $r(t) = \sin(\omega t)$ and vary ω, we can see the range of frequencies that the control input $u(t)$ can follow. A qualitative relationship between this frequency range and the choice of sampling period for a digital regulator is as follows: if the analog regulator can generate control inputs with a "high" frequency content then the sampling period has to be "small." Conversely if the analog regulator can only generate "low" frequency signals then the sampling period can be large.

[1]If the system encounters an initial state vector further from the origin than $\mathbf{x}_{\text{WC}}$, the control input will saturate at ± 5 until the regulator brings the system state vector closer to the origin. In some situations, this saturation is acceptable; for instance, in a position control system. See Problem 8 at the end of this chapter.

The system in Fig. 6.27 is described by the state-space model $(\mathbf{A} - \mathbf{B}\mathbf{K}_a, \mathbf{B}, \mathbf{K}_a)$. A Bode plot for this system is used to find the bandwidth of the analog regulator. The bandwidth is defined to be the frequency at which the Bode magnitude plot is 3 dB down from its value at zero frequency. That is, if we let $H(s)$ be the transfer function of the closed-loop system shown in Fig. 6.27

$$H(s) = \mathbf{K}_a(s\mathbf{I} - (\mathbf{A} - \mathbf{B}\mathbf{K}_a))^{-1}\mathbf{B}$$

then the bandwidth ω_B is defined as

$$|H(j\omega_B)| \text{ dB} = |H(0)| - 3 \text{ dB}.$$

This is the definition of bandwidth given in Chapter 5 applied to the system of Fig. 6.27.

Once the bandwidth of an analog regulator has been found we can use one of the following rules-of-thumb to pick the sampling interval T for the digital regulator. We first note that a sampling interval T corresponds to a sampling frequency ω_s given by

$$\omega_s = \frac{2\pi}{T} \text{ radians/second.}$$

The following rules are used to choose ω_s

1. If the plant (or the sensors) is subject to broadband random disturbances choose

$$20 \leq \frac{\omega_s}{\omega_B} \leq 40. \tag{6.62}$$

   By broadband random disturbances, we mean random disturbances whose frequency content exceeds the bandwidth of the control system.

2. If the plant is not subject to broadband random disturbances choose

$$5 \leq \frac{\omega_s}{\omega_B} \leq 10. \tag{6.63}$$

Some discussion of the motivation for these rules may be found in [32]. These rules correspond to upper and lower bounds for the sampling interval T. One may question why lower bounds on T are given. In other words, what is the problem with a very small value of T? The answer is that there is no problem with sampling very fast if you have the hardware to do it. However, there little or no improvement in control system performance if T is chosen smaller than the lower bounds corresponding to the above rules.

6.8 Chapter Summary

In this chapter we showed how to design digital state-feedback regulators. The purpose of a regulator is to keep all state variables as close as possible to zero in the face of disturbances acting on the plant. A state-feedback regulator accomplishes this by feeding a linear combination of the state variables back to the input to the plant. A digital regulator samples the state variables with A/D converters and passes the calculated linear combination through a D/A converter connected to the input to the plant.

We showed how to calculate the feedback gains to place the closed-loop poles in desired locations. It is helpful to specify closed-loop poles in the s-plane, and then map them into

the z-plane using the ZOH pole-mapping formula. One advantage of specifying the poles in the s-plane initially is that the sampling interval for the digital regulator can then be chosen as a function of the bandwidth of the analog regulator corresponding to the given s-plane poles. Normalized Bessel polynomials were introduced as a convenient way of choosing closed-loop pole locations. The roots of these polynomials can be scaled to achieve a desired settling time.

We showed how to find the gain and phase margins for a digital state-feedback regulator. These stability margins can be used to choose the best among several different regulators which have been designed for a given settling time. An example with a plant having complex poles showed that a regulator which added damping to the complex poles had superior stability margins to a regulator designed using Bessel poles.

Finally, we showed how to modify a regulator to obtain a tracking system whose output is one of the state variables. This tracking system will have zero steady-state error to a step input if the plant has a pole at $s = 0$.

6.9 Problems

1. Test whether or not the following systems are controllable. If they are controllable, calculate the transformation that transforms them to controllable canonical form.

   (a)

   $$\mathbf{\Phi} = \begin{bmatrix} 1.3 & 1 \\ -0.4 & 0 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}.$$

   (b)

   $$\mathbf{\Phi} = \begin{bmatrix} 0 & 1 \\ 0.25 & 0 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}.$$

   (c)

   $$\mathbf{\Phi} = \begin{bmatrix} 1 & -.08 \\ 1 & 0.4 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.8 \\ 1 \end{bmatrix}.$$

2. Consider a type-1 servo system described by the equations

   $$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

   $$y(t) = [1\ 0] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}.$$

   The ZOH design model for the plant has been calculated to be

   $$\mathbf{\Phi} = \begin{bmatrix} 1.0000 & 0.0488 \\ 0 & 0.9512 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0012 \\ 0.0488 \end{bmatrix},$$

   where the sampling period is $T = 0.05$.

Calculate the gains for digital state feedback which will place the closed- loop poles at a location corresponding to a double pole at -4 in the s-plane. Show all calculations.

3. Given a continuous-time system $(\mathbf{A}, \mathbf{b}, \mathbf{c})$ we want to use digital state feedback to obtain the following closed-loop pole locations in the s-plane: $-1 \pm j$. Suppose that the ZOH equivalent system for $T = 0.2$ secs. is

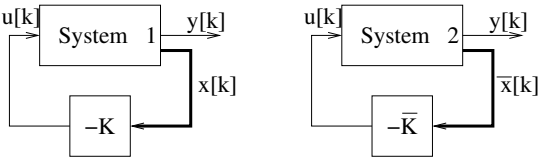
$$\mathbf{x}[k+1] = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[k]$$

$$y[k] = [1 \;\; 0]\mathbf{x}[k].$$

   (a) Determine the feedback vector $\mathbf{K}$ so that the the control $u[k] = -\mathbf{K}\mathbf{x}[k]$ achieves the desired closed-loop pole locations.

   (b) Calculate the poles of the discrete-time closed-loop system and verify that the desired poles have been obtained.

4. (a) Let $(\mathbf{\Phi}, \Gamma, \mathbf{c})$ be the ZOH equivalent of System 1, shown below, which is to be controlled using digital state feedback. Let $\mathbf{x}[k]$ denote the vector of state variables for this system, and Kb denote a feedback vector which has been designed for this system. Suppose now that different sensors are used which do not measure $\mathbf{x}[k]$ directly, but instead measure $\bar{\mathbf{x}}[k] = \mathbf{T}\mathbf{x}[k]$, where $\mathbf{T}$ is a nonsingular matrix. What should the vector $\bar{\mathbf{K}}$ be in order for System 2 to have the same closed-loop poles as System 1?



   (b) As a simple example of the results of part (a), assume that state feedback of a second order system results in the following input to the plant:

$$u[k] = -K_1 x_1[k] - K_2 x_2[k]. \tag{6.64}$$

   Suppose different sensors are used which measure the following variables

$$\begin{aligned} \bar{x}_1[k] &= x_1[k] + x_2[k] \\ \bar{x}_2[k] &= x_1[k] - x_2[k]. \end{aligned} \tag{6.65}$$

   What values should $\bar{K}_1$ and $\bar{K}_2$ take (in terms of K_1 and K_2) so that the state feedback

$$u[k] = -\bar{K}_1 \bar{x}_1[k] - \bar{K}_2 \bar{x}_2[k] \tag{6.66}$$

   results in the same control signal as (6.64)? Solve this problem directly by combining (6.64)–(6.66), and also by applying the general formula derived in part (a).

5. Consider the following state-space description, where α is a real number

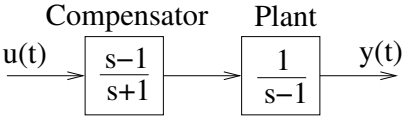
$$\mathbf{x}[k+1] = \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} 1 \\ \alpha \end{bmatrix} u[k]$$

$$y[k] = [1 \;\; 0]\mathbf{x}[k].$$

(a) Find all values of α for which the system is *not* controllable.

(b) Find all values of α for which a pole-zero cancellation occurs in the transfer function.

6. A major reason for the development of state-space control theory in the mid 1950's was the need to explain why compensators for unstable systems which rely on pole-zero cancellation will not work even if the cancellation is *perfect*. Consider the following compensator and plant



The overall transfer function from $u(t)$ to $y(t)$ is $1/(s+1)$ which is stable. A state-space model for the above system is given by the following

$$A = \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \quad c = [1 \;\; 0].$$

(a) Show that the state-space model given above is uncontrollable.

(b) Compute e^{At} for the above model, where t is a variable. Show that when the input to the system is zero and for almost all initial conditions, the state vector has at least one element which goes to infinity. (Note: this problem is *not* a ZOH calculation!)

The conclusion is that the above system has a stable transfer function, but it has an unstable uncontrollable subsystem caused by the pole zero cancellation. Thus the system is actually unstable (elements of its state vector go to infinity) for almost all nonzero initial conditions.

7. A system with transfer function $1/[s(s+1)]$ has the following state-space description

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$
$$y = [\,1 \quad 0\,]\,\mathbf{x}.$$

In order to design a digital regulator for this system, we compute its ZOH equivalent at a sampling interval of $T = 0.1$ seconds

$$\mathbf{\Phi} = \begin{bmatrix} 1.0000 & 0.0952 \\ 0 & 0.9048 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0048 \\ 0.0952 \end{bmatrix}.$$

Calculate a feedback vector $\mathbf{K}$ for a digital regulator which has a settling time of $T_S = 1.5$ seconds. Use the roots of a 2nd-order Bessel polynomial. (*Hint:* $e^{a+jb} = e^a(\cos b + j \sin b)$ where b has units of radians.)

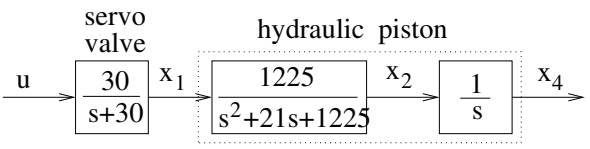
8. *Position Control System with Actuator Saturation*

Consider the position control system designed in Example 6.11. Repeat the simulation

and see that the input signal exceeds the value 5. Now change the simulation so that the input signal saturates at ± 5 and obtain a new step response. Compare the step responses with and without saturation.

9. *Hydraulic Positioning System*

The purpose of this problem is to design a digital regulator for a hydraulic positioning system. The hardware is represented by the following block diagram



where the variables are defined as follows

x_1 = valve position

x_2 = flow

x_4 = position of piston.

The first block can be represented in state-space form as

$$\dot{x}_1 = -30x_1 + 30u.$$

The input and output of the middle block are related by

$$\ddot{x}_2 + 21\dot{x}_2 + 1225x_2 = 1225x_1.$$

If we define the state variable x_3 as follows

$$\dot{x}_2 = x_3,$$

then we have

$$\dot{x}_3 = 1225x_1 - 1225x_2 - 21x_3.$$

We can combine all of these equations into the following state-space model.

$$\dot{\mathbf{x}} = \begin{bmatrix} -30 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1225 & -1225 & -21 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 30 \\ 0 \\ 0 \\ 0 \end{bmatrix} u.$$

The lightly damped complex poles model a resonance caused by the compressibility of oil.

The desired settling time for the system is $T_S = 0.35$ seconds. Choose two different sets of closed-loop s-plane pole locations to achieve this settling time. Let the first set be the roots of a 4th-order Bessel polynomial. Let the second set of roots consist of roots from a 2nd-order Bessel polynomial, and let the other two roots be the resonant poles pulled further into the left half plane (see Example 6.10).

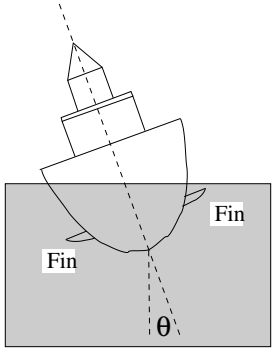
(a) Calculate $\mathbf{K}$ vectors for *analog* regulators corresponding to the two different sets of desired closed-loop pole locations. Obtain a Bode magnitude plot for each of the analog regulators and find their respective bandwidths. Show that a sampling rate of $T = 0.01$ is adequate for both of the corresponding digital regulators.

(b) Compute the ZOH equivalent for $T = 0.01$. Calculate $\mathbf{K}$ vectors for *digital* regulators corresponding to the two different sets of desired closed-loop pole locations. Don't forget to map the pole locations into the z-plane using the ZOH pole-mapping formula! Obtain Nyquist plots for each regulator and find the gain and phase margins.

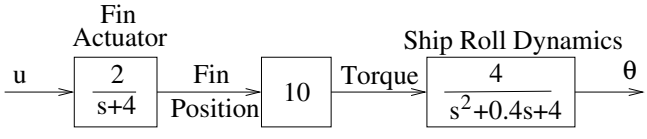
(c) Introduce a reference input as discussed in Section 6.5. Plot the step response of the system and the control input signal.

10. *Ship Stabilization System*

The roll of a ship can be regulated using fins which project into the water to generate a stabilizing torque. The following figure shows a picture of the roll angle θ, and shows the fins extending a certain distance into the water.



The displacement of the fins is controlled by motors (fin actuators) and we assume that the torque generated by the fins is proportional to their displacement. If the roll angle is positive, only one of the fins is moved, and if the roll angle is negative, the other fin is moved. A block diagram of the system is shown below



We choose the following state variables

$$x_1 = \theta \text{ (radians)}$$

$$x_2 = \dot{\theta} \text{ (radians/second)}$$

$$x_3 = \text{fin position (m)}$$

which yields the following state-space description

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ -4 & -.4 & 40 \\ 0 & 0 & -4 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} u.$$

(a) Plot the open-loop response of the ship in response to the following initial state

$$\mathbf{x}(0) = \begin{bmatrix} .34 \\ 0 \\ 0 \end{bmatrix}$$

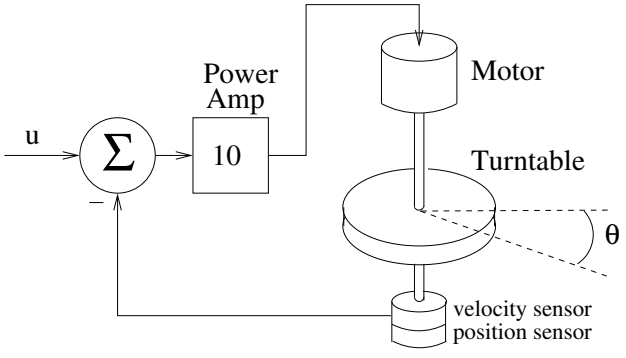
which corresponds to the ship at 20° roll angle. You should see a roll period of about 3 seconds.

(b) Design a digital state feedback regulator which results in a settling time of 4.5 seconds. Plot the roll angle of the closed-loop system in response to the initial state used in (a). Find the stability margins of this regulator.

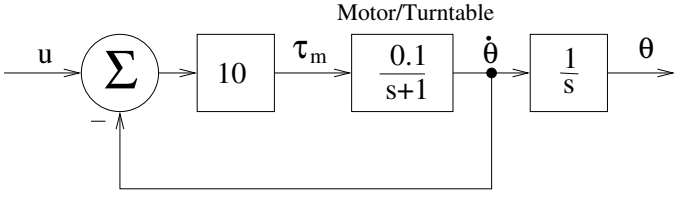
Note: the regulation of the ship's roll angle in the presence of a disturbance torque due to waves is considered in Chapter 8, problem 2.

11. *Large Turntable System*

Consider a position control system for the large turntable shown in system shown below



A block diagram of this system is shown below



The state variables are $x_1 = \theta$ and $x_2 = \dot{\theta}$. τ_m represents the torque produced by the motor. The time constant for the motor is assumed to be much smaller than that for

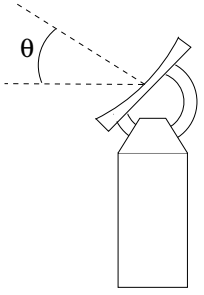
the turntable (even with gearing) so that the motor/turntable system is described by a single time constant.

(a) Design a digital regulator which achieves a settling time of 1 second. Find the stability margins of this regulator.

(b) Introduce a reference input for x_1 and plot the step response of the system for a step input of 0.34 radians.

12. *Antenna Positioning System*

A sketch of an antenna system is shown below where θ is the pointing angle to be controlled.



This system is described by the following differential equation

$$J\ddot{\theta} + B\dot{\theta} = T_c$$

where J and B represent the effective inertia and damping, respectively, of the antenna and motor drive. The state variables, input, and output are defined as follows

$$x_1 = \theta$$

$$x_2 = \dot{\theta}$$

$$u = T_c$$

$$y = x_1$$

and the corresponding state-space model is

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & -1.5 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0.2 \end{bmatrix} u$$

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

A transient wind disturbance at time zero can be modeled as a non-zero initial velocity, for example,

$$\mathbf{x}(0) = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}.$$

(a) Design a digital state-feedback regulator to achieve a settling time of 1 second. Find the stability margins of this regulator. Perform a simulation with the initial state vector given above.

(b) Introduce a reference input and plot the response to a step input of 0.34 radians.

CHAPTER 7

OBSERVERS

7.1 Introduction

We saw in the previous chapter that state feedback can be used to place a system's closed-loop poles at specified locations. The pole-placement formulas gave a simple but powerful method for designing control systems.

A problem in using state feedback comes about when it is difficult (or expensive) to measure all of the state variables of the plant. This situation might arise for a high-order system. For instance, if the plant were modeled with a 25th order state-space model, then 25 sensors would be needed, one to measure each state variable. It might be that the cost of 25 sensors would be too great in an actual system, and the decision might be made to measure only a subset (say half) of the state variables. Another situation that might arise even for low-order systems is that some of the state variables are difficult to measure. For example, consider a two-link pendulum mounted on a cart. It is difficult to mount a sensor at the joint between the two links of the pendulum. Unfortunately, if even a single state variable cannot be measured, then none of the state feedback results from the previous chapter can be used directly!

This unhappy situation is remedied in this chapter where it is shown how to design observers. Observers are systems which estimate some or all of the state variables of the plant. It will be shown that the estimated state variables can be used in conjunction with the state-feedback results of the previous chapter to perform digital control.

7.2 Full-Order Observers

In this section, we derive the equations of a system called an *observer* which estimates the values of all the state variables of the plant. The observer is called *full-order* because it provides estimates of *all* the state variables. Observers which only provide estimates of *some* of the state variables are called *reduced-order observers* and are presented in the Section 7.4.

7.2.1 Open-Loop Observers

We begin our treatment of observers by considering the system shown in Fig. 7.1. The top

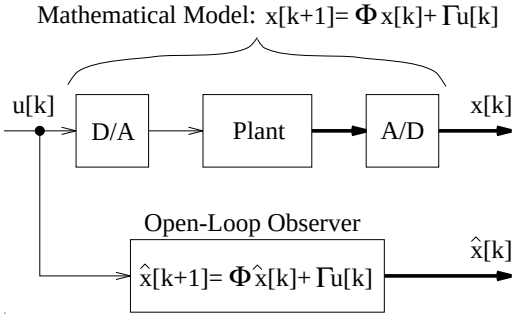


Figure 7.1 A plant under digital control and its open-loop observer.

branch of that figure shows a plant under digital control. It also shows the state variables being sampled by A/D converters. In Chapter 4, we developed the ZOH equivalent model $(\mathbf{\Phi}, \mathbf{\Gamma})$ for a plant surrounded by A/D and D/A converters as in Fig. 7.1. The ZOH model describes how input samples $u[k]$ are transformed to state-vector samples $\mathbf{x}[k]$ by the plant. The lower branch of Fig. 7.1 shows that we can use the ZOH model to construct state-vector estimates $\hat{\mathbf{x}}[k]$. We emphasize that the upper branch of Fig. 7.1 represents a physical system while the lower branch is a state-space model whose equations can be implemented in a computer.

We now ask under what circumstances will the open-loop observer produce state-vector estimates that are exact; that is, when will $\hat{\mathbf{x}}[k] = \mathbf{x}[k]$? The answer is that the following three conditions must be met for equality to occur for all k:

1. The initial conditions of the plant are known so that we can set $\hat{\mathbf{x}}[0] = \mathbf{x}[0]$.
2. The mathematical model of the plant is *perfect*, i.e. the behavior of the physical system is described *exactly* by the ZOH model.
3. There are no disturbances acting on the plant.

If any one of the above conditions is not satisfied, then the state vector of the observer will not be equal to the state vector of the plant at sampling instants, and we will need some way to correct the estimates produced by the observer. In other words, we will need to "close the loop" around the observer. Before deriving the closed-loop observer equations, let us briefly consider the three conditions mentioned above.

If the initial conditions of the observer were not equal to the initial conditions of the plant, the observer would produce incorrect state estimates. But if the ZOH model were

stable, then the effect of the initial conditions would decay with time. Thus it seems that condition 1 is the least important of the three. However, we note that disturbances acting on the plant (condition 3) have the effect of changing the state vector of the plant in a way that is not predicted by the ZOH equations. For example, think of the inverted pendulum system with the rod perfectly balanced. If the cart is motionless, the equations say that the position and velocity of the rod should also be zero (since the rod is balanced). A transient disturbance can be introduced by tapping the rod, and this disturbance will give a nonzero value to two of the state variables of the system: the position and velocity of the rod. Thus disturbances acting on a system are equivalent to altering the values of state variables, and this change in state variables can be modeled as a mismatch in initial conditions. In this sense, conditions 1 and 3 are equivalent for transient disturbances. If an observer has good performance with respect to a mismatch in initial conditions, it will also have good performance with respect to transient disturbances on the plant.

Now consider condition 2 – the requirement for an exact mathematical model. The power of state-space techniques is most evident when the model of the plant is accurate; however, it is too much to expect any mathematical model to be perfect! In this book, we make the assumption that the model is accurate but not perfect. A closed-loop observer will be able to correct for the errors introduced by "small" model inaccuracies. A rigorous analysis of the effects of model inaccuracies is beyond the scope of this book. We give a simplified treatment of "robust observers" later in this chapter. See [33] for a fuller treatment of this topic.

7.2.2 Closed-Loop Observers

In the previous section, we saw that the ZOH model of the plant can be used to generate estimates of the state vector, but these estimates will be good only when three conditions are satisfied. Since all three conditions will never be satisfied in practice, we need some way of detecting and correcting errors that occur. We would like to use some type of feedback to force the error vector $\mathbf{x}[k]-\hat{\mathbf{x}}[k]$ to go to zero, but the problem is that we don't know $\mathbf{x}[k]$! (If we did, we wouldn't need an observer.) Although we can't measure $\mathbf{x}[k]$, we can measure the output of the plant $y[k] = \mathbf{c}\mathbf{x}[k]$ which is a particular linear combination of the state variables. If we use the same linear combination of the estimated state variables, we can generate the following error signal: $e[k] = y[k] - \mathbf{C}\hat{\mathbf{x}}[k] = \mathbf{C}(\mathbf{x}[k] - \hat{\mathbf{x}}[k]))$. If $\hat{\mathbf{x}}[k] = \mathbf{x}[k]$, then the error signal will be zero, which would seem to indicate that the observer is correctly estimating the state vector of the plant. At this point the careful reader will notice that it is possible for $e[k]$ to be zero even if $\hat{\mathbf{x}}[k] \neq \mathbf{x}[k]$. All that is required is that the vector $\hat{\mathbf{x}}[k] - \mathbf{x}[k]$ be orthogonal to the vector $\mathbf{C}$. Thus it might seem that the error signal $e[k]$ will not provide enough information to insure that the observer correctly estimates the state vector of the plant. Nevertheless, the only way to determine whether or not $e[k]$ can be used as a feedback signal is to try it! A closed-loop observer which uses $e[k]$ as a feedback signal is shown in Fig. 7.2. The closed-loop observer has two inputs, $u[k]$ and $e[k]$. The input $u[k]$ is multiplied by the vector $\mathbf{\Gamma}$ which comes from the ZOH model of the plant. The input $e[k]$ is multiplied by an $n \times 1$ vector $\mathbf{L}$ of "observer gains." The question of whether or not the closed-loop observer shown in Fig. 7.2 will work comes down to the following: can we choose the vector $\mathbf{L}$ in such a way that the observer has "good" performance with respect to unknown initial conditions? By good performance, we mean that $\hat{\mathbf{x}}[k] \rightarrow \mathbf{x}[k]$ even as $\mathbf{x}[k]$ is changing. A typical plot of a plant state variable $x_1[k]$ and the estimated variable $\hat{x}_1[k]$ produced by an observer is shown in Fig. 7.3.

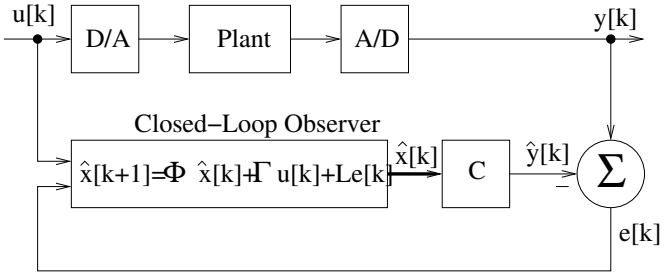


Figure 7.2 A plant under digital control and its closed-loop observer. **L** is a vector of observer gains that must be specified.

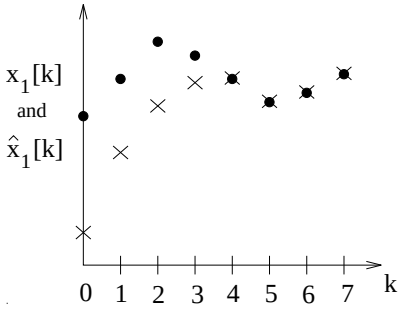


Figure 7.3 A plant under digital control and its closed-loop observer. **L** is a vector of observer gains that must be specified.

In the analysis that follows, we assume that the initial state vector is unknown so that there is a difference between the plant states and the observer states, as shown in Fig. 7.3. An observer has "good performance" if its estimates "quickly" converge to the actual values of the plant state variables. Once the observer is "tracking" the plant state variables as shown in Fig. 7.3 for $k \geq 4$, it is possible that a disturbance acting on the plant will again cause an error between the actual and estimated state variables. In this case, the observer estimates will converge to the true plant states in exactly the same manner as they did in the case of unknown initial conditions.

In order to examine the behavior of the closed-loop observer, we start with an *analysis model* shown in Fig. 7.4, which is obtained from Fig. 7.2 by replacing the plant with its

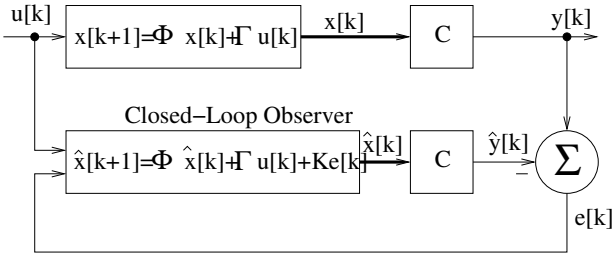


Figure 7.4 Analysis model for a closed-loop observer in which the plant in Fig. 7.2 has been replaced by its ZOH equivalent.

ZOH equivalent. We begin the analysis of the closed-loop observer by defining an error vector $\mathbf{e}[k] = \mathbf{x}[k] - \hat{\mathbf{x}}[k]$ and we assume that initial error $\mathbf{e}[0] \neq \mathbf{0}$. Let us write a difference equation for $\mathbf{e}[k]$ and see how the error vector propagates with time. In particular, we are interested in conditions which will insure that the error vector goes to zero. The system in Fig. 7.4 is described by the ZOH model for the plant

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k] \tag{7.1}$$

and by the following equation for the closed-loop observer

$$\hat{\mathbf{x}}[k+1] = \mathbf{\Phi}\hat{\mathbf{x}}[k] + \mathbf{\Gamma}u[k] + \mathbf{L}\mathbf{C}(\mathbf{x}[k] - \hat{\mathbf{x}}[k]). \tag{7.2}$$

Subtracting (7.2) from (7.1) yields the following equation for the error vector

$$\begin{aligned} \mathbf{e}[k+1] &= \mathbf{\Phi}\mathbf{e}[k] - \mathbf{L}\mathbf{C}\mathbf{e}[k] \\ &= (\mathbf{\Phi} - \mathbf{L}\mathbf{C})\mathbf{e}[k]. \end{aligned} \tag{7.3}$$

The solution to this homogeneous first-order vector difference equation was derived in Chapter 3 and is given by

$$\mathbf{e}[k] = (\mathbf{\Phi} - \mathbf{L}\mathbf{C})^k \mathbf{e}[0]. \tag{7.4}$$

If the matrix $\mathbf{\Phi} - \mathbf{L}\mathbf{C}$ has eigenvalues inside the unit circle, then $(\mathbf{\Phi} - \mathbf{L}\mathbf{C})^k$ approaches the zero matrix as the time index k gets large. From (7.4), this means that the error vector will go to zero regardless of the initial error $\mathbf{e}[0]$. Thus a closed-loop observer can correct errors associated with unknown initial conditions (and also errors caused by transient disturbances acting on the plant) if an observer gains vector $\mathbf{L}$ can be found such that the eigenvalues of the matrix $\mathbf{\Phi} - \mathbf{L}\mathbf{C}$ are within the unit circle. The problem of finding such a vector $\mathbf{L}$ is closely related to the pole-placement problem discussed in the previous chapter. Before we show how to calculate $\mathbf{L}$, however, we first show how to choose the initial state vector for the observer and also obtain a simplified block diagram for the observer.

We can re-write (7.2) as follows

$$\hat{\mathbf{x}}[k+1] = (\mathbf{\Phi} - \mathbf{L}\mathbf{C})\hat{\mathbf{x}}[k] + \mathbf{\Gamma}u[k] + \mathbf{L}y[k]. \tag{7.5}$$

This is a state-update equation for the observer which has two inputs: $u[k]$ and $y[k]$ (the input and output of the plant). In order to implement this equation in a computer, an initial state vector $\hat{\mathbf{x}}[0]$ must be chosen. If we assume that the state vector of the plant is not measured, then the initial state vector $\mathbf{x}[0]$ is unknown. However, the output $y[0]$ provides some information about $\mathbf{x}[0]$. Assume that the output equation for the plant is

$$\mathbf{y}[k] = \mathbf{C}\mathbf{x}[k].$$

For a fixed value of k this equation is a linear matrix equation in which $\mathbf{C}$ is a $1 \times n$ coefficient matrix and $\mathbf{x}[k]$ is a vector of n unknowns. From the theory of linear equations (Chapter 2) this equation has an infinite number of solutions. The minimum-norm solution at $k = 0$ is (see (2.42))

$$\mathbf{x}_{MN}[0] = \mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1}y[0]$$

and we initialize the observer with this vector.

$$\hat{\mathbf{x}}[0] = \mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1}y[0]. \tag{7.6}$$

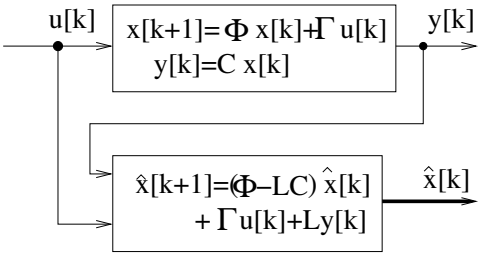


Figure 7.5 The plant ZOH and an observer given by (7.5).

This choice if initialization was suggested in [49]. Using (7.5), the block diagram of Fig. 7.4 can be replaced by the simpler block diagram shown in Fig. 7.5.

We now show how to calculate the observer-gains vector $\mathbf{L}$ to place the eigenvalues of $\mathbf{\Phi} - \mathbf{LC}$ in specified locations.

Recall in the pole-placement problem, we needed to calculate a row vector $\mathbf{L}$ such that the eigenvalues of $(\mathbf{\Phi} - \mathbf{\Gamma K})$ were inside the unit circle at specified locations. To find observer gains, we need to calculate a column vector $\mathbf{L}$ such that the eigenvalues of $(\mathbf{\Phi} - \mathbf{LC})$ are within the unit circle. These problems are clearly similar, and in fact we can calculate observer gains using pole placement formulas by using the fact that a matrix and its transpose have the same eigenvalues. Thus we can think of the observer gains problem as one of specifying a row vector $\mathbf{L}^T$ such that the matrix

$$(\mathbf{\Phi} - \mathbf{LC})^T = \mathbf{\Phi}^T - \mathbf{C}^T\mathbf{L}^T \tag{7.7}$$

has eigenvalues within the unit circle. We are guaranteed that the eigenvalues of $(\mathbf{\Phi} - \mathbf{LC})$ will be identical to those of $(\mathbf{\Phi} - \mathbf{LC})^T$.

By considering the transposed matrix in (7.7), the problem is now identical to a pole placement problem. Briefly, we put $\mathbf{\Phi}^T$ and $\mathbf{C}^T$ into the pole placement formula and solve for the vector $\mathbf{L}^T$. This results in the following formula

$$\mathbf{L}^T = [\,(p_1 - a_1) \quad \cdots \quad (p_n - a_n)\,]\,\bar{\mathbf{W}}_c[\mathbf{C}^T\ \mathbf{\Phi}^T\mathbf{C}^T\ \cdots\ (\mathbf{\Phi}^T)^{n-1}\mathbf{c}^T]^{-1}. \tag{7.8}$$

In the above equation, the coefficients a_i come from the characteristic polynomial of $\mathbf{\Phi}$ (which is identical to the characteristic polynomial of $\mathbf{\Phi}^T$). The coefficients p_i specify a polynomial $p(z)$ whose roots are the desired eigenvalues of $(\mathbf{\Phi} - \mathbf{LC})$. These eigenvalues are called the *observer poles*. The matrix $\bar{\mathbf{W}}_c$ in (7.8) is identical to the one used in the pole placement formula. The reason is that $\bar{\mathbf{W}}_c$ only depends on $a(z)$, the characteristic polynomial of $\mathbf{\Phi}$ or $\mathbf{\Phi}^T$. A more useful formula for the observer gains vector is obtained by transposing (7.8) to obtain

$$\mathbf{L} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C\Phi} \\ \vdots \\ \mathbf{C\Phi}^{n-1} \end{bmatrix}^{-1} \bar{\mathbf{W}}_c^T \begin{bmatrix} p_1 - a_1 \\ p_2 - a_2 \\ \vdots \\ p_n - a_n \end{bmatrix}. \tag{7.9}$$

1. Given $\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C}$ and $p(z)$, a polynomial whose roots are the desired observer poles, first calculate the observability matrix

$$\mathbf{W}_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{C\Phi} \\ \vdots \\ \mathbf{C\Phi}^{n-1} \end{bmatrix}.$$

   A vector of observer gains can be calculated only if $\mathbf{W}_o$ is non-singular.

2. Calculate $a(z)$, the characteristic polynomial of $\mathbf{\Phi}$. Recall that the roots of $a(z)$ can be obtained using the ZOH pole mapping formula on the poles of the (continuous-time) plant.

3. Form the matrices

$$\bar{\mathbf{\Phi}} = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_n \\ 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & \cdots & 1 & 0 \end{bmatrix}, \text{ and } \bar{\mathbf{\Gamma}} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

4. Form

$$\bar{\mathbf{W}}_c = \begin{bmatrix} \bar{\mathbf{\Gamma}} & \bar{\mathbf{\Phi}}\bar{\mathbf{\Gamma}} & \cdots & \bar{\mathbf{\Phi}}^{n-1}\bar{\mathbf{\Gamma}} \end{bmatrix}.$$

5. Calculate the vector of observer gains as follows

$$\mathbf{L} = \mathbf{W}_o^{-1}\bar{\mathbf{W}}_c^T \begin{bmatrix} p_1 - a_1 \\ \vdots \\ p_n - a_n \end{bmatrix}.$$

6. Initialize the observer with

$$\hat{\mathbf{x}}[0] = \mathbf{C}^T(\mathbf{C}\bar{\mathbf{C}}^T)^{-1}y[0].$$

Table 7.1 A procedure for calculating a vector of observer gains.

Notice that the observer gains vector can be calculated only if the *observability matrix*

$$\mathbf{W}_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{C\Phi} \\ \vdots \\ \mathbf{C\Phi}^{n-1} \end{bmatrix} \tag{7.10}$$

is invertible. In general, an n^{th} order system $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C})$ is said to be *observable* if and only if the observability matrix $\mathbf{W}_o$ is of rank n. The observability matrix was used in Chapter 3, Section 3.3.7, to compute the initial conditions of a system. The procedure for designing an observer with specified pole locations is given in Table 7.1. The question of how to choose observer pole locations is deferred until Section 7.3.3.

EXAMPLE 7.1

Consider the following system

$$\mathbf{x}[k+1] = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u[k]$$

$$y[k] = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x}[k].$$

The observability matrix for this system can be calculated to be

$$\mathbf{W}_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{C\Phi} \\ \mathbf{C\Phi}^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & 1 & 0 \end{bmatrix}.$$

The determinant of this matrix equals zero, and so the system is not observable. ■

Observability means that we can estimate the values of n state variables using a single output measurement. Controllability means that we can control the values of n state variables using a single input.

EXAMPLE 7.2

Consider the plant

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

with the following state-space model

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -4 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

The ZOH equivalent model for a sampling period of $T = 0.1$ seconds is

$$\mathbf{\Phi} = \begin{bmatrix} 1.0000 & 0.0952 & 0.0042 \\ 0 & 0.9048 & 0.0782 \\ 0 & 0 & 0.6703 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0001 \\ 0.0042 \\ 0.0824 \end{bmatrix}.$$

The eigenvalues of $\mathbf{\Phi}$ are its diagonal elements, and we can calculate the characteristic polynomial of $\mathbf{\Phi}$ by multiplying out the eigenvalues as follows

$$a(z) = (z-1)(z-0.9048)(z-0.6703) = z^3 - 2.575z^2 + 2.1817z - 0.6065.$$

The matrices for controllable canonical form are

$$\bar{\mathbf{\Phi}} = \begin{bmatrix} 2.575 & -2.1817 & .6065 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \bar{\mathbf{\Gamma}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

and the controllability matrix for controllable canonical form is

$$\mathbf{W}_c = \begin{bmatrix} 1.0000 & 2.5752 & 4.4497 \\ 0 & 1.0000 & 2.5752 \\ 0 & 0 & 1.0000 \end{bmatrix}.$$

The observability matrix for $(\mathbf{\Phi}, \mathbf{\Gamma})$ is

$$\mathbf{W}_o = \begin{bmatrix} \mathbf{C} \\ \mathbf{C\Phi} \\ \mathbf{C\Phi}^2 \end{bmatrix} = \begin{bmatrix} 1.0000 & 0 & 0 \\ 1.0000 & 0.0952 & 0.0042 \\ 1.0000 & 0.1813 & 0.0145 \end{bmatrix}.$$

Suppose we want to specify the observer poles as the roots of a 3rd-order Bessel polynomial scaled to achieve a settling time of 0.5 seconds. From Table 6.2 we get

$$\begin{aligned} p(z) &= (z - e^{-5.0093/0.5})(z - e^{(-3.9668 \pm j3.7845)/0.5}) \\ &= z^3 - 1.024z^2 + 0.4461z - 0.0751. \end{aligned}$$

Then the formula for observer gains yields

$$\mathbf{L} = \mathbf{W}_o^{-1} \bar{\mathbf{W}}_c^T \begin{bmatrix} p_1 - a_1 \\ \vdots \\ p_n - a_n \end{bmatrix} = \begin{bmatrix} 1.5503 \\ 6.9754 \\ 10.0252 \end{bmatrix}.$$

To demonstrate the performance of this observer, we "connect" the observer to the plant as shown in Fig. 7.6. The following sequence is used as the input to the plant

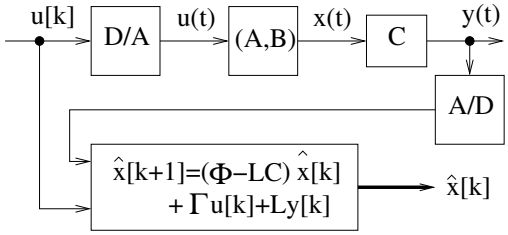


Figure 7.6 The observer designed in Example 7.2 connected to the plant.

$$u[k] = \begin{cases} 1, & k = 0, \cdots, 9 \\ 0, & k = 10, \cdots 19. \end{cases}$$

Because the sampling period is 0.1 seconds, this corresponds to the plant input $u(t) = 1$, $0 \leq t < 1$ and $u(t) = 0$, $t \geq 1$. With this input to the plant, we perform two different simulations. In the first simulation, the initial state vector of the plant and the initial state vector of the observer are both set to the zero vector. This means that

$$\mathbf{e}[0] = \mathbf{x}[0] - \hat{\mathbf{x}}[0] = \mathbf{0}$$

and from (7.4), the error vector $\mathbf{e}[k]$ should be zero for all k (assuming that no disturbances act on the system). This is indeed what we see in Fig. 7.7. In that figure, the vector $\hat{\mathbf{x}}[k]$ is passed through a zero-order hold to produce a piecewise-constant plot. The observer state vector exactly matches the plant state vector at sampling instants.

In the second simulation, the plant was initialized as follows

$$\mathbf{x}(0) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

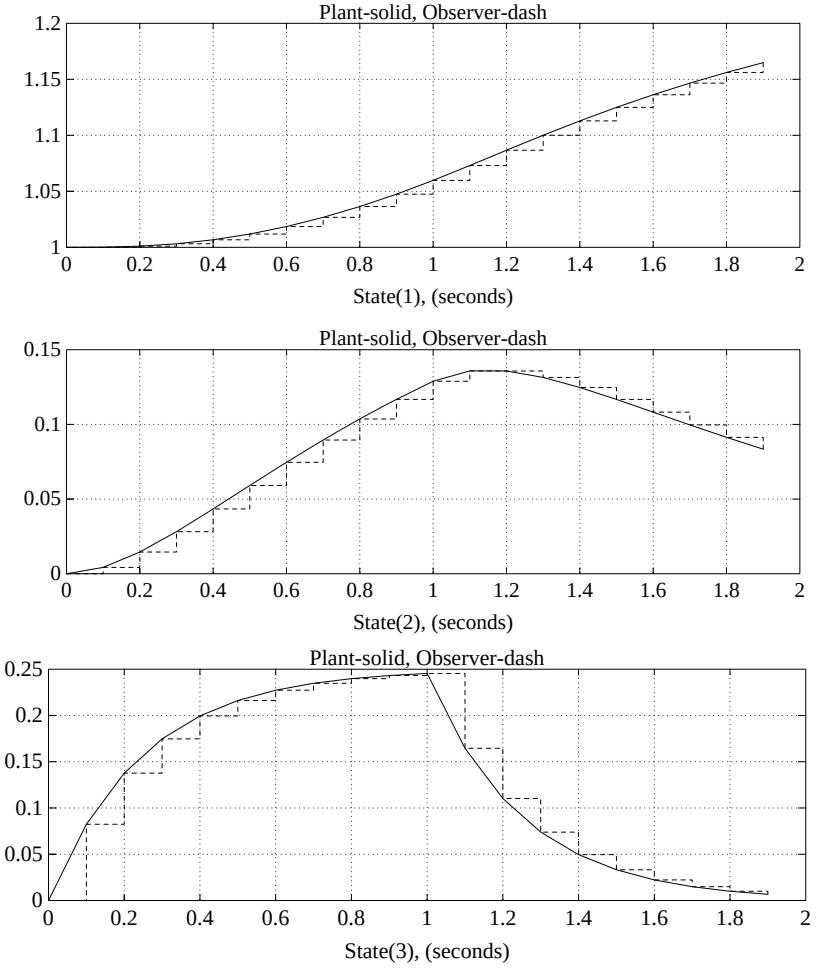


Figure 7.7 Plant state variables (solid line) and observer state variables (dashed line) for Example 7.2. The plant and observer were both initialized to the zero vector.

Since the output corresponding to this state vector is $y[0] = \mathbf{C}\mathbf{x}[0] = 0$, the initialization given by (7.6) is $\hat{\mathbf{x}}[0] = \mathbf{0}$. Thus in this simulation, there is an initial error between the plant state vector and the state vector of the observer. The observer state vector should converge to the plant state vector even as the plant state vector is changing. Moreover, this convergence should take place in about 0.5 seconds, which is the settling time chosen for the observer. The results are shown in Fig. 7.8.

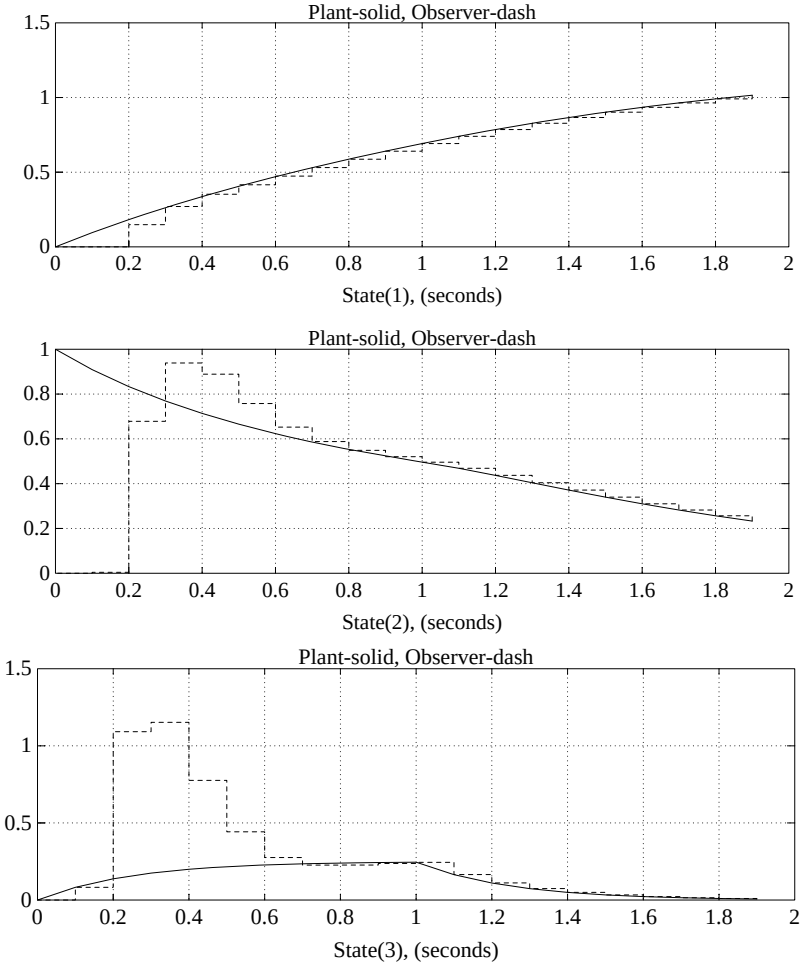


Figure 7.8 Plant state variables (solid line) and observer state variables (dashed line) for Example 7.2. The plant is initialized to the vector $[0 \quad 1 \quad 0]^T$, and the observer is initialized to the zero vector.

■

7.3 Using Observers for Regulation

The motivation for developing the theory of observers was to be able to use state feedback in situations where not all the state variables could be measured. In the previous section,

we showed how to design observers that would estimate state variables in such a way that the error between the estimates and the actual values of the state variables would go to zero. It seems reasonable to use these estimated state variables for the purpose of state feedback. However, there are several important questions that need to be addressed. The most important question is whether or not an $\mathbf{K}$ vector that was designed for full state feedback can be used when some of the state variables are not measured but only estimated. If we do use the same $\mathbf{K}$ vector, how is the behavior of the closed-loop system affected by the estimation errors that are present before the observer is "tracking" the plant state variables? In fact, how do we know that the closed-loop system using an observer is even stable? Another important question is how the stability margins of a full-state feedback regulator are affected when an observer is used to estimate the state variables. The answers to these questions are developed in this section.

To begin the discussion, assume that the plant we want to regulate is described by the ZOH model $(\mathbf{\Phi},\ \mathbf{\Gamma})$. Suppose that we have designed a feedback vector $\mathbf{K}$ using the techniques of Chapter 6 such that the closed-loop system with full state feedback has desired pole locations. In other words, the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma K}$ are the desired closed-loop pole locations. Suppose that the state variables are not measured, and only the system output $y[k]$ is available. Using an observer, an estimate $\hat{\mathbf{x}}[k]$ of the state vector can be computed and used to close the loop through a vector of gains $\mathbf{K}$. The actual closed-loop system is shown in Fig. 7.9(a), and the discrete-time design model is shown in Fig. 7.9(b).

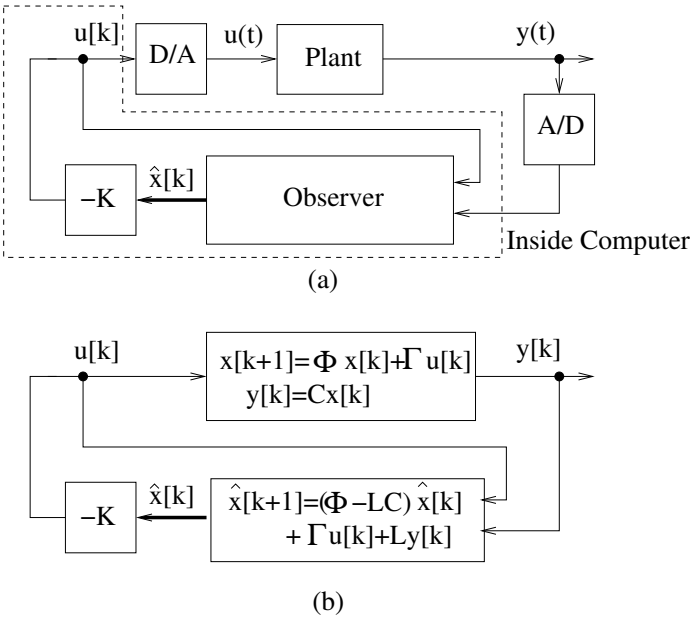


Figure 7.9 (a) A plant under digital control. An observer estimates the state vector for state feedback. (b) The discrete-time model for the system in (a).

In order to write a state-space description for the interconnected system in Fig. 7.9, we define the state vector to be

$$\mathbf{z}[k] = \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{e}[k] \end{bmatrix} \tag{7.11}$$

where $\mathbf{e}[k]$ is the error vector

$$\mathbf{e}[k] = \mathbf{x}[k] - \hat{\mathbf{x}}[k]. \tag{7.12}$$

This choice of state vector is different than that used to describe the interconnection of two systems in Section 3.5. However, the choice in (7.11) is more convenient for the present purpose because an update equation for $\mathbf{e}[k]$ has already been obtained in (7.3). In order to write an update equation for $\mathbf{z}[k]$, we need to get an expression for $\mathbf{x}[k+1]$ in terms of $\mathbf{x}[k]$ and $\mathbf{e}[k]$. From Fig. 7.9(b) we have

$$\begin{aligned} u[k] &= -\mathbf{K}\hat{\mathbf{x}}[k] \\ &= \mathbf{K}(\mathbf{e}[k] - \mathbf{x}[k]). \end{aligned}$$

Substituting this equation into the state update equation for the plant yields

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}\mathbf{K}(\mathbf{e}[k] - \mathbf{x}[k]) \\ &= (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})\mathbf{x}[k] + \mathbf{\Gamma}\mathbf{K}\mathbf{e}[k]. \end{aligned}$$

The above equation and (7.3) give the following equation for the closed-loop regulator system

$$\begin{bmatrix} \mathbf{x}[k+1] \\ \mathbf{e}[k+1] \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K} & \mathbf{\Gamma}\mathbf{K} \\ \mathbf{0} & \mathbf{\Phi} - \mathbf{L}\mathbf{C} \end{bmatrix}}_{\mathbf{\Phi_{CL}}} \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{e}[k] \end{bmatrix}.$$

The poles of the closed-loop system are the eigenvalues of the matrix $\mathbf{\Phi_{CL}}$. Since this matrix is block upper-triangular, its eigenvalues are just the eigenvalues of the diagonal blocks. Thus we have that

$$\text{eig}(\mathbf{\Phi_{CL}}) = \text{eig}(\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}) \cup \text{eig}(\mathbf{\Phi} - \mathbf{L}\mathbf{C}) \tag{7.13}$$

where the symbol $\cup$ denotes union. But the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}$ are the desired closed-loop poles that were specified when the $\mathbf{K}$ vector was designed for full state feedback, and the eigenvalues of $\mathbf{\Phi} - \mathbf{L}\mathbf{C}$ are the observer poles which were chosen to be well within the unit circle.

Equation (7.13) is called the *the separation principle*, which refers to the fact that the state feedback vector and the observer gains vector can be calculated separately. If full state feedback is used with the vector $\mathbf{K}$, then the closed-loop poles will be at the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}$. If the state variables cannot be measured, then an observer can be designed to estimate them. The observer can be introduced into the system using the *same* $\mathbf{K}$ *vector that full state feedback would use, and the observer will not change the poles that are located at the eigenvalues of* $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}$. The closed-loop system will have additional poles at the eigenvalues of $\mathbf{\Phi} - \mathbf{L}\mathbf{C}$. If the eigenvalues of $\mathbf{\Phi} - \mathbf{L}\mathbf{C}$ are chosen to be much less "dominant" than the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}$, then the system with the observer can have almost the same transient behavior as the system with full state feedback. However, the stability margins of an observer-based regulator must also be taken into account when choosing the observer poles. Before discussing stability margins and the choice of observer poles, we first give an example to illustrate the transient behavior of an observer-based regulator.

EXAMPLE 7.3

Consider the observer-based regulator shown in Fig. 7.9 on page 302. Such a regulator is specified by a state feedback vector $\mathbf{K}$ and an observer gains vector $\mathbf{L}$. Suppose the plant in Fig. 7.9 is described by

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -4 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

This plant that was used in Example 6.8 on page 258 where a state feedback vector $\mathbf{K}$ was calculated to give a settling time of 2 seconds. This plant was also used in Example 7.2 where an observer gains vector was calculated to give a "settling time" for the state estimation error $\mathbf{e}[k]$ of about 0.5 seconds. We say that this observer is "4 times faster" than the state feedback regulator. In other words, $\mathbf{e}[k] \rightarrow \mathbf{0}$ (equivalently, $\hat{\mathbf{x}}[k] \rightarrow \mathbf{x}[k]$) 4 times faster than $\mathbf{x}[k] \rightarrow \mathbf{0}$.

The performance of a state-feedback regulator for this plant is shown in Example 6.8. We can compare this performance to that of an observer-based regulator which uses the same $\mathbf{K}$ vector as in Example 6.8, but uses an observer to estimate the state vector instead of measuring the state vector directly. We use the observer gains vector calculated in Example 7.2, and the observer is initialized using (7.6) with $\hat{\mathbf{x}}[0] = \mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1}y[0]$. If the initial state vector of the plant is

$$\mathbf{x}[0] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

then $\hat{\mathbf{x}}[0]$ turns out to equal $\mathbf{x}[0]$. This means that $\mathbf{e}[0] = \mathbf{0}$ and so $\mathbf{e}[k]$ will be zero for all k (in the absence of any disturbances). In this case the transient behavior of the observer-based regulator will be identical to that of the full-state feedback regulator in Example 6.8. However, if the initial state vector of the plant is

$$\mathbf{x}[0] = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},$$

then the initialization $\hat{\mathbf{x}}[0] = \mathbf{C}^T(\mathbf{C}\mathbf{C}^T)^{-1}y[0]$ yields $\hat{\mathbf{x}}[0] = \mathbf{0}$. In this case the state estimates produced by the observer will not be identical to the state vector of the plant until $\mathbf{e}[k] \rightarrow \mathbf{0}$ at about 0.5 seconds (see Example 7.2). Fig. 7.10 shows the transient behavior of full-state feedback regulator and an observer-based regulator. Notice that the observer-based regulator exhibits slightly more overshoot than the full-state feedback regulator.

■

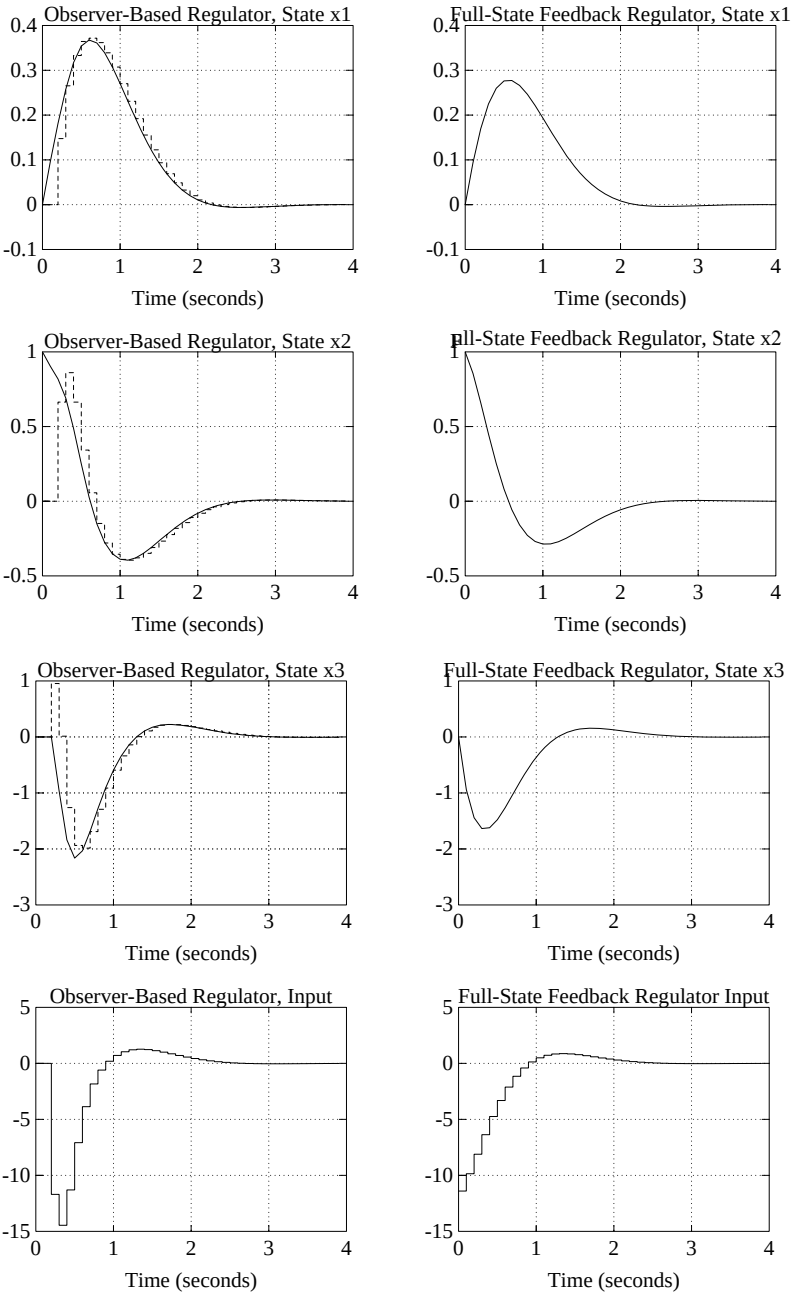


Figure 7.10 Example 7.3 – The graphs in the left-hand column are for the observer-based regulator. The solid lines are the plant state variables and the dashed lines are the estimates produced by the observer. The graphs in the right-hand column are for the full-state feedback regulator.

7.3.1 Choosing the Sampling Interval for an Observer-Based Regulator

The settling time of an observer is defined as the time it takes for the error in the estimated state variables to converge to zero. This time will be denoted T_{so} seconds. The settling time of a digital state-feedback regulator is defined as the time it takes for the regulator to drive the plant state vector to zero. This settling time is denoted T_s seconds. In order for an observer-based regulator to function properly, the observer settling time should be smaller than the regulator settling time. The rule-of-thumb is three to five times smaller. That is,

$$T_{so} = \frac{T_s}{f}, \;\; 3 \leq f \leq 5. \tag{7.14}$$

When this relationship is satisfied for some value of f, the observer is said to be "f-times faster" than the desired regulator settling time.

An f-times faster observer has a settling time of T_s/f seconds. An observer is an nth-order system whose state vector needs to be driven from its initial state fo the state-vector of the plant, and we must allow n time steps for this to happen. See the discussion about the choice of sampling interval for state feedback regulators in Chapter 6. From that discussion, recall that we require the number of samples to settle to be larger than twice the order of the system. That is,

$$\frac{T_s/f}{T} \geq 2n \quad \text{or} \;\; T \leq \frac{T_s}{2 \cdot n \cdot f}. \tag{7.15}$$

If T is chosen to be $T_s/(20n)$ then (7.15) will be satisfied. If a larger sampling interval is needed then $20n$ may be decreased to any value down to $2 \cdot n \cdot f$.

7.3.2 Stability Margins for Observer-Based Regulators

In Chapter 6, the gain and phase margin of a digital regulator were obtained by looking at a Nyquist plot of the regulator with the loop broken at the input to the plant (see Section 6.3.3). In this section we examine the stability margins of a digital regulator which employs an observer to estimate the state vector; for example, the regulator shown in Fig. 7.9. Based on the development of observers given in Section 7.2, we might hope that if the observer is "fast enough," then a regulator using an observer would have the same stability margins as a full state feedback regulator. Unfortunately, this is not the case.

To analyze the stability margins of an observer-based regulator, we re-draw the system in Fig. 7.9 with the loop broken at the input to the plant. Because the Nyquist plot assumes that the loop will be closed with a negative sign, we remove the negative sign from the $\mathbf{K}$ vector, resulting in the system shown in Fig. 7.11. The loop would be closed by defining $u[k] = -\alpha[k]$. Notice that when the negative sign is removed from $\mathbf{K}$, a negative sign must be inserted on the $\mathbf{\Gamma}\alpha[k]$ term in Fig. 7.11. The following equations are obtained from Fig. 7.11

$$\begin{aligned}
\begin{bmatrix} \mathbf{x}[k+1] \\ \hat{\mathbf{x}}[k+1] \end{bmatrix} &= \underbrace{\begin{bmatrix} \mathbf{\Phi} & \mathbf{0} \\ \mathbf{LC} & \mathbf{\Phi} - \mathbf{LC} - \mathbf{\Gamma K} \end{bmatrix}}_{\bar{\mathbf{\Phi}}} \begin{bmatrix} \mathbf{x}[k] \\ \hat{\mathbf{x}}[k] \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{0} \end{bmatrix}}_{\bar{\mathbf{\Gamma}}} u[k] \\
\alpha[k] &= \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{K} \end{bmatrix}}_{\bar{\mathbf{C}}} \begin{bmatrix} \mathbf{x}[k] \\ \hat{\mathbf{x}}[k] \end{bmatrix}.
\end{aligned} \tag{7.16}$$

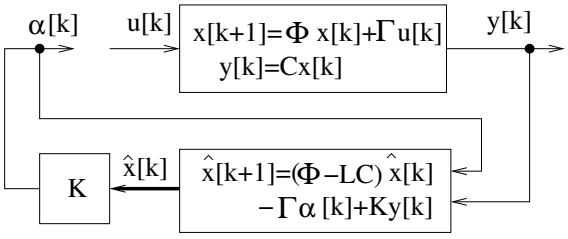


Figure 7.11 An observer-based regulator with the loop broken at the input to the plant.

The Nyquist plot of $(\bar{\mathbf{\Phi}}, \bar{\mathbf{\Gamma}}, \bar{\mathbf{C}})$ can be used to find the stability margins for the observer-based regulator of Fig. 7.9. As an example, we now find the stability margins for the regulator designed in the previous example.

EXAMPLE 7.4

The regulator in Example 7.3 is an observer-based regulator which uses the same state feedback vector $\mathbf{L}$ that was used in a full state feedback regulator in Example 6.8 on page 258. The full-state feedback regulator had the following stability margins

$$\text{GM} = 21.7\,\text{dB}$$

$$\text{PM} = 60^\circ.$$

The Nyquist plot for the loop transfer function of the observer-based regulator of Example 7.3 is shown in Fig. 7.12.

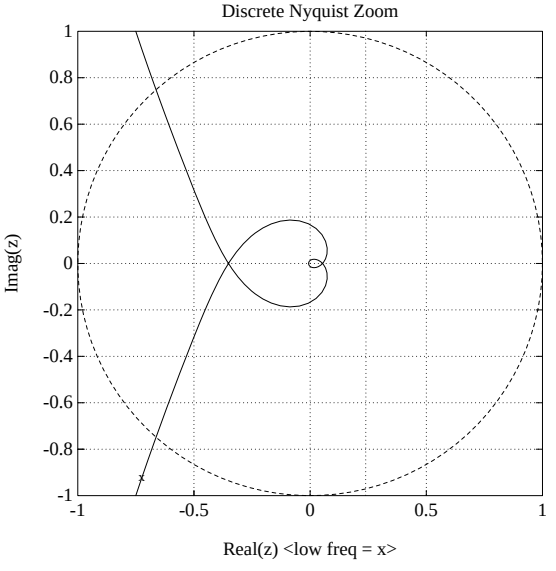


Figure 7.12 A portion of the Nyquist plot for Example 7.4.

The stability margins for the observer-based regulator are

$$\text{GM} = 8.9 \text{ dB}$$

$$\text{PM} = 49^\circ.$$

These stability margins are adequate, but they are not as good as the stability margins for the full state-feedback regulator.

■

The previous example showed that a particular observer-based regulator had worse stability margins than the corresponding full-state feedback regulator. This example raises a number of questions. Does an observer always decrease the stability margins of a full-state feedback regulator? Is is possible to design an observer that preserves stability margins? If stability margins cannot be preserved, what type of observer gives the best possible stability margins? We begin to answer these questions by way of another example.

Doyle and Stein [26] give an example which illustrates that the stability margins of an observer-based regulator can be much worse than the margins of the corresponding full-state regulator. Moreover, the problem cannot be solved by making the observer "faster." The plant in the Doyle-Stein example is

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 2 & 1 \end{bmatrix} \mathbf{x}(t).$$

The transfer function of this plant is

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

In order to control this plant with a digital regulator, we compute the ZOH equivalent with $T = 0.1$. The result is

$$\mathbf{\Phi} = \begin{bmatrix} 0.9868 & 0.0820 \\ -0.2460 & 0.6588 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0044 \\ 0.0820 \end{bmatrix}.$$

Suppose we want a regulator with a settling time of 1 second. One way to do this is to choose the s-plane roots of a normalized 2nd-order Bessel polynomial and map them to the z-plane as follows

$$z_{1,2} = e^{0.1(-4.0530 \pm j2.3400)} = 0.6486 \pm j0.1546. \tag{7.17}$$

The state feedback vector which achieves these closed-loop poles can be calculated to be

$$\mathbf{K} = \begin{bmatrix} 14.9264 & 3.4509 \end{bmatrix}.$$

We can also design observers using Bessel polynomials. A "3-times faster" observer for this example has $T_S = 1/3$ (since the regulator has a settling time of 1 second). Thus we choose the following observer poles

$$z_{1,2} = e^{0.1(-4.0530 \pm j2.3400)/(1/3)} = 0.2263 \pm 0.1914.$$

Similarly, a 6-times faster observer has $T_S = 1/6$, and is specified by the following pole locations

$$z_{1,2} = e^{0.1(-4.0530 \pm j2.3400)/(1/6)} = 0.0146 \pm 0.0867.$$

The observer gains vectors for the 3-times and 6-times faster observers can be calculated to be

$$\mathbf{L}_1 = \begin{bmatrix} 4.8672 & -8.5414 \end{bmatrix}^T \text{, and } \mathbf{L}_2 = \begin{bmatrix} 8.1391 & -14.6617 \end{bmatrix}^T,$$

respectively. Fig. 7.13 shows the Nyquist plot for the full-state feedback regulator. Fig. 7.14 and Fig. 7.15 show the Nyquist plots for the 3-times and 6-times faster observer-based regulators, respectively. The gain and phase margins for these regulators are

	Gain Margin	Phase Margin
Full-state feedback	15.2 dB	83°
3× faster observer	-2.7, 1.6 dB	10°
6× faster observer	-1.3 ,1.1 dB	8°

It can be seen that the stability margins for the observer-based regulators are much worse than for the full-state feedback regulator, and that making the observer faster actually makes the problem worse.

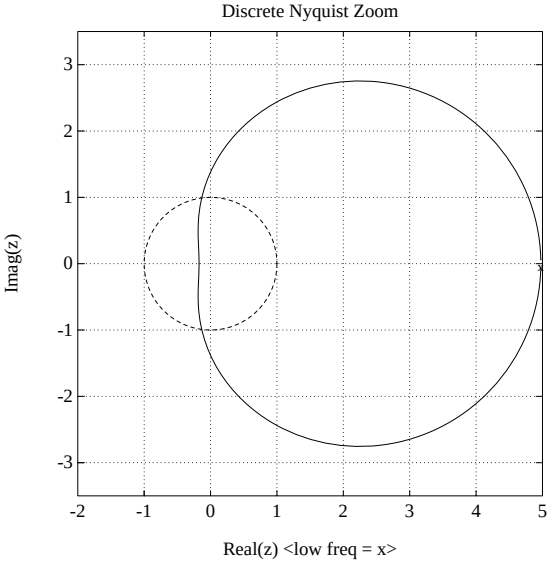


Figure 7.13 The Nyquist plot for the full-state feedback regulator.

The difficulty is caused by the zero location of the plant. Doyle and Stein have shown that any "significant" zeros of the plant must be chosen as observer pole locations in order for an observer-based regulator to have stability margins close to those of the corresponding state-feedback regulator. A significant zero is one which is located near the closed-loop

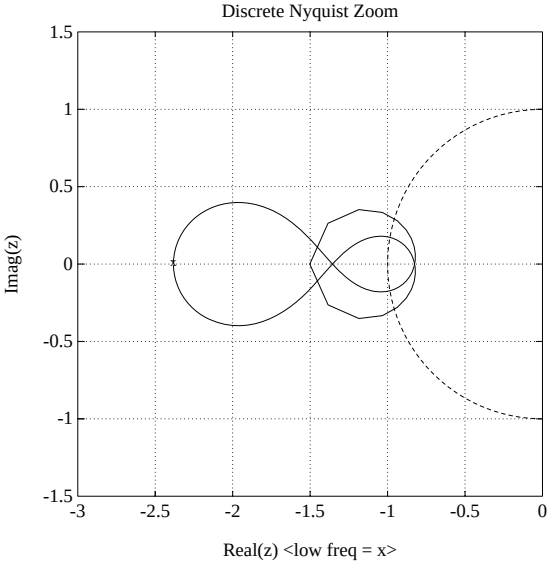


Figure 7.14 The Nyquist plot for the full-state feedback regulator.

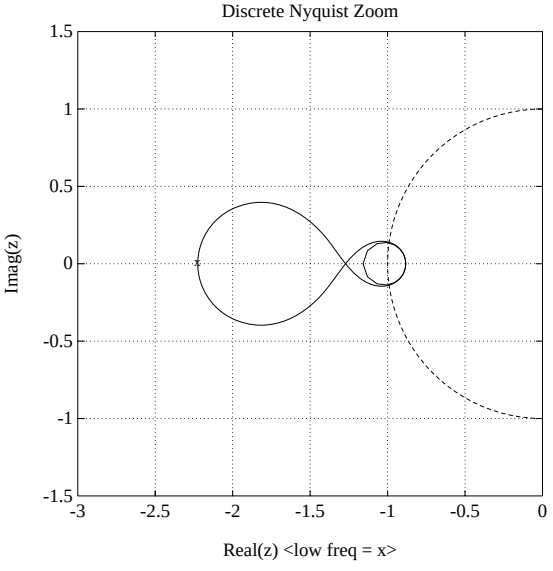


Figure 7.15 The Nyquist plot for the full-state feedback regulator.

poles of the full-state feedback regulator. In the example system considered above, the ZOH model $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C})$ corresponds to the transfer function

$$G(z) = \mathbf{C}(z\mathbf{I} - \mathbf{\Phi})^{-1}\mathbf{\Gamma} = 11.0159\frac{z - 0.8189}{z^2 - 1.6457z + 0.6703}.$$

This transfer function has a single zero located at

$$z = 0.8189.$$

The closed-loop poles of the state-feedback regulator are given in (7.17). Since the zero locations are "close" to the closed-loop pole locations, an observer pole should be placed at $z = 0.8189$. The other observer pole can be placed at the root of a first-order Bessel polynomial scaled for a settling time of 1/3 (i.e. this pole is 3 times faster than the desired closed-loop poles). This pole is at $z = .2501$. The observer gains vector corresponding to this choice of observer poles is

$$\mathbf{L}_3 = \begin{bmatrix} 0.1094 & 0.3578 \end{bmatrix}.$$

The Nyquist plot for the observer-based regulator using $\mathbf{K}_3$ is shown in Fig. 7.16. The

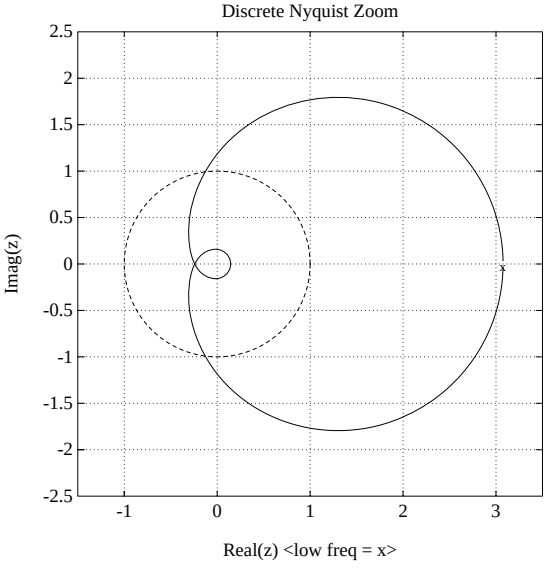


Figure 7.16 The Nyquist plot for the observer-based regulator with observer gains vector $\mathbf{K}_3$.

stability margins for this regulator are

$$\text{GM } = 12.3 \text{ dB}, \text{ and PM } = 83^\circ$$

which are close to the stability margins for the full-state feedback regulator.

It is not always possible to choose observer pole locations to be equal to plant zeros. For instance, if a zero of $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C})$ is outside the unit circle, say at $z = 1.1$, we would not choose 1.1 as an observer pole. The reason is that observer poles are some of the poles of the overall closed-loop system (see (7.13)). Thus if we choose an observer pole at $z = 1.1$

we have actually designed an unstable closed-loop system! When the plant has a zero outside the unit circle, Doyle and Stein recommend using the reciprocal zero location, i.e. $z = 1/1.1 = 0.91$ as an observer pole location.

It frequently happens that a ZOH equivalent for the plant has zeros on the negative real axis. These zeros are usually "far" from the desired closed-loop poles, and they do not have to be chosen as observer poles. Another reason for not choosing observer poles on the negative real axis is that such poles can cause an oscillation in the control input to the plant.

7.3.3 How to Choose Observer Poles

In light of the discussion in the previous subsection, the first step in choosing observer pole locations is to calculate the zero locations $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C})$, the ZOH of the plant. If z_0 is a zero location which is close to the eigenvalues of $(\mathbf{\Phi} - \mathbf{\Gamma K})$ then z_0 should be chosen as one of the observer poles. If z_0 is outside the unit circle and $1/z_0$ is close to the eigenvalues of $(\mathbf{\Phi} - \mathbf{\Gamma K})$, then choose $1/z_0$ as one of the observer poles.

The remaining observer poles can be chosen to be three to five times faster than the eigenvalues of $(\mathbf{\Phi} - \mathbf{\Gamma K})$. For instance, if the $\mathbf{K}$ vector was calculated using the roots of a Bessel polynomial corresponding to settling time T_S, then the remaining observer poles can be chosen as the roots of a Bessel polynomial corresponding to settling time

$$T_S' = T_S/\alpha$$

where

$$3 \leq \alpha \leq 5.$$

Of course, the Bessel roots must be mapped into the z-plane using the ZOH pole-mapping formula before computing $\mathbf{K}$ or $\mathbf{L}$. The complete list of guidelines for choosing the poles of an observer-based regulator is given in the following table.

The rules given below specify pole locations for continuous-time systems. For a digital control (discrete-time systems), these numbers, s_i, must be mapped using the ZOH pole-mapping formula, $e^{s_i T}$, where T is the sampling interval in seconds. Let]tt Ts be the desired settling time of the regulator. For what follows, we note that given a continuous-time state-space model `(A,B,C,D)`. for a plant (SISO or MIMO), the zeros of the plant are calculated by the command: `tzero(A,B,C,D)`. The rules below refer to a variable `s1`, which is the first-order normalized Bessel pole with 1-second settling time. Its numerical value is `s1=-4.62`.

Regulator Poles

1. Use normalized Bessel poles scaled (divided) by the desired settling time (see Table II).
2. Use sufficiently damped plant poles; that is, plant poles whose real parts lie to the left of `s1/Ts`.
3. If the plant has complex poles that are not sufficiently damped, choose closed-loop poles by replacing the real parts of these poles with `s1/Ts` and keeping the imaginary parts the same. Changing the real parts adds damping to those plant poles, and so these are called "added damping" poles.
4. If the plant has unstable pole locations (real part greater than zero) consider the reflection of these poles about the imaginary axis, i.e. change the sign of the (real-part) of the pole. If the reflected pole is to the left of `s1/Ts` use it as a closed-loop pole.

Observer Poles

1. Use normalized Bessel poles scaled (divided) by the desired observer settling time. The observer settling time should be chosen to be one third to one fifth of the desired settling time for the regulator or tracking system. Such an observer is said to be "three to five times faster."
2. If the plant has zeros in the left half-plane, consider using these zero locations as observer poles.
3. If the plant has zeros in the right half-plane, consider using the reflection of these zero locations (replace their real parts with the negatives of their real parts) as observer poles.

Table 7.2 Rules for Selecting Regulator and Observer Pole Locations

7.4 Reduced-Order Observers

In the previous section, we showed how to design an observer which produces estimates of *all* the state variables of the plant. This observer was based on the ZOH equivalent of the plant, and so the order of the observer system was the same order as the plant. Such an observer is called a *full-order observer*. There is an important case where one might not want to use a full-order observer. That is when some (but not all) of the state variables can be measured. In this case, we only need to estimate the state variables that cannot be measured.

There are two differences between the full-order observers derived in the previous section and the reduced-order observers derived in this section. These differences are

- The full-order observers of the previous section are *prediction observers*. That is, the estimate of the state vector at time k depends on the plant output value at time $k-1$. This type of observer allows for one complete sampling interval to perform the required calculations. In situations where the sampling interval is very short (or the control computer is very slow), it may be necessary to allow a complete sampling interval for calculations. The disadvantage of this type of observer, however, is that it causes a one-sample time delay in a feedback loop. For example, compare the plots of the control input in Fig. 7.10. The observer-based regulator results in a zero input for the first sample, while the full-state feedback regulator responds immediately to the disturbance. This one-sample time delay will not result in instability (recall (7.13) showed that the closed-loop system with the observer is stable), but it will result in a degradation in performance as compared with a full-state feedback regulator, as shown in Example 7.3.

  In some cases the observer calculations can be performed in a very small fraction of the sampling interval. In these cases, it is useful design an observer in which the estimate of the state vector at time k depends on the plant output at time k. Such observers, called *current observers*, cannot be realized exactly in practice because they don't allow any time for the calculations to be performed. However they can be approximately realized whenever the calculations can be performed in a small fraction of the sampling interval. The reduced-order observers discussed in this section are all current observers. It is also possible to design full-order current observers, see for example [32].

- The full-order observers of the previous section assumed that a scalar output measurement will be used as an input to the observer. It may be that the observability matrix $\mathbf{W}_o$ defined in (7.10) is singular which implies that the plant is not observable from a scalar output. A vector of observer gains cannot be calculated in this situation. However, it is possible that if *more than one* measurement were available, then an observer could be designed. The reduced-order observers developed in this section explicitly consider the case of multiple measurements, and can be used in situations where the plant is not observable from a scalar output.

  It is also possible to design full-order observers that use multiple measurements. Such observers are complicated to design because they require the solution to a multivariable pole-placement problem (an algorithm to do this is given in Chapter 9). In contrast, there is a useful special case of reduced-order observers with multiple measurements which can be designed simply by solving linear equations – that is, using theory from Chapter 2, not from Chapter 9.

To begin the discussion of reduced-order observers, assume that we measure m linear combinations of the state vector, and we put these measurements at each sampling instant in a vector $\mathbf{y}_m[k]$, where

$$\underbrace{\mathbf{y}_m[k]}_{m\times 1} = \underbrace{\mathbf{C}_m}_{m\times n}\underbrace{\mathbf{x}[k]}_{n\times 1}. \tag{7.18}$$

We assume that the measurements (rows of $\mathbf{C}_m$) are linearly independent so that the matrix $\mathbf{C}_m$ has rank m. It will be shown that if the ZOH of the plant is a system of order n, and if m measurements are available, then a *reduced-order observer* of dimension $n-m$ can be designed to estimate the $n-m$ state variables that are not measured. A special case of this result is easy to verify. Suppose that $m=n$ and the matrix $\mathbf{C}_m$ is invertible. Then $\mathbf{x}[k] = \mathbf{C}_m^{-1}\mathbf{y}_m[k]$. That is, we can recover the complete state vector directly from the output vector without using an observer. In this case we can say that the observer has dimension $n-m=0$. When $\mathbf{C}_m$ is the identity matrix, we are actually measuring all of the state variables directly.

To begin the derivation of reduced-order observers, we will assume that the state vector of the plant has been partitioned in such a way so that the first m state variables can be measured and the remaining $n-m$ state variables must be estimated. That is, we partition $\mathbf{x}[k]$ as

$$\mathbf{x}[k] = \begin{bmatrix} \mathbf{x}_1[k] \\ \mathbf{x}_2[k] \end{bmatrix} \begin{matrix} \} m \\ \} n-m \end{matrix}. \tag{7.19}$$

We make a slight generalization in the above assumptions by saying that the vector of measurements $\mathbf{y}_m[k]$ can be a nonsingular combination of the state variables in $\mathbf{x}_1[k]$ as follows

$$\begin{aligned} \mathbf{y}_m[k] &= \mathbf{C}_m\mathbf{x}[k] \\ &= [\,\mathbf{C}_1 \quad \mathbf{0}\,]\,\mathbf{x}[k] \\ &= \mathbf{C}_1\mathbf{x}_1[k] \end{aligned} \tag{7.20}$$

where $\mathbf{C}_1$ is an $m\times m$ nonsingular matrix. In other words, we assume that the measurement matrix has the special form

$$\mathbf{C}_m = [\,\mathbf{C}_1 \quad \mathbf{0}\,].$$

If $\mathbf{C}_1$ is the identity matrix it means we are measuring the state variables in $\mathbf{x}_1[k]$ directly. However, it is only necessary to assume that $\mathbf{C}_1$ is nonsingular.

The partition of the state vector in (7.19) induces a partitioning of the ZOH matrices as shown below

$$\mathbf{\Phi} = \begin{matrix} m & n-m \\ \end{matrix} \begin{bmatrix} \mathbf{\Phi}_{11} & \mathbf{\Phi}_{12} \\ \mathbf{\Phi}_{21} & \mathbf{\Phi}_{22} \end{bmatrix} \begin{matrix} \} m \\ \} n-m \end{matrix} \tag{7.21}$$

and

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{\Gamma}_1 \\ \mathbf{\Gamma}_2 \end{bmatrix} \begin{matrix} \} m \\ \} n-m \end{matrix}. \tag{7.22}$$

The ZOH equation for the plant, $\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k]$ can now be written as two equations using the partitioned matrices defined above. The result is

$$\begin{aligned} \mathbf{x}_1[k+1] &= \mathbf{\Phi}_{11}\mathbf{x}_1[k] + \mathbf{\Phi}_{12}\mathbf{x}_2[k] + \mathbf{\Gamma}_1 u[k] \\ \mathbf{x}_2[k+1] &= \mathbf{\Phi}_{21}\mathbf{x}_1[k] + \mathbf{\Phi}_{22}\mathbf{x}_2[k] + \mathbf{\Gamma}_2 u[k]. \end{aligned} \tag{7.23}$$

Since the variables in the $\mathbf{x}_1[k]$ vector are measured, the observer does not have to estimate them. The observer only has to estimate the variables in the $\mathbf{x}_2[k]$ vector. Equation (7.23) provides the ZOH model for $\mathbf{x}_2[k]$. If we proceed in direct analogy to the development of the full-order observer, we would augment the ZOH equation for $\mathbf{x}_2[k]$ with an error term that depends on the system outputs as follows:

$$\hat{\mathbf{x}}_2[k+1] = \mathbf{\Phi}_{21}\mathbf{x}_1[k] + \mathbf{\Phi}_{22}\hat{\mathbf{x}}_2[k] + \mathbf{\Gamma}_2 u[k] + \mathbf{L}_1(\mathbf{y}_m[k] - \mathbf{C}_1\mathbf{x}_1[k]). \tag{7.24}$$

However, the "error term" in the above equation always equals zero, since by definition, $\mathbf{y}_m[k] = \mathbf{C}_1\mathbf{x}_1[k]$! Thus the "observer gains" matrix $\mathbf{L}_1$ does not have any effect on (7.24). Nevertheless, (7.24) is indeed a closed-loop observer, because the vector $\mathbf{x}_1[k]$ that appears in the first term of that equation is equivalent to the measurement vector since $\mathbf{x}_1[k] = \mathbf{C}_1^{-1}\mathbf{y}_m[k]$. If we substitute this expression for $\mathbf{x}_1[k]$ into (7.24), we obtain

$$\hat{\mathbf{x}}_2[k+1] = \mathbf{\Phi}_{22}\hat{\mathbf{x}}_2[k] + \mathbf{\Gamma}_2 u[k] + \mathbf{\Phi}_{21}\mathbf{C}_1^{-1}\mathbf{y}_m[k]. \tag{7.25}$$

A block diagram of the above equation is shown in Fig. 7.17. We reiterate that this observer

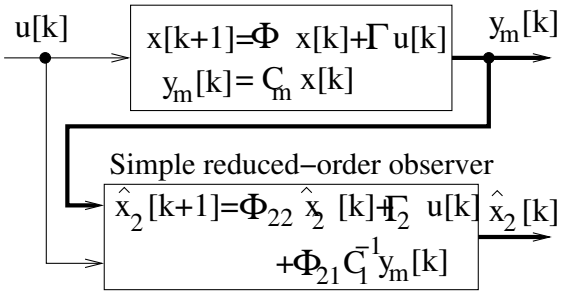


Figure 7.17 A simple reduced-order observer that does not require the calculation of an observer-gains vector.

is a closed-loop observer since the measurement vector $\mathbf{y}_m[k]$ is fed back to the input of the observer. There is still the question of how the reduced-order observer gets information about $\mathbf{x}_2[k]$, the vector it is trying to estimate. A qualitative answer is as follows: if the plant variables $\mathbf{x}_2[k]$ are disturbed, or have a different value from the estimated vector $\hat{\mathbf{x}}_2[k]$, the disturbance will be "felt" by the variables in $\mathbf{x}_1[k]$ as shown in (7.23). If the $\mathbf{x}_1[k]$ variables are changed, these in turn change the value of $\mathbf{y}_m[k]$, which is fed into the observer and helps the observer produce a better estimate of $\mathbf{x}_2[k]$. The fact that changes in the $\mathbf{x}_2[k]$ vector produce changes in the $\mathbf{x}_1[k]$ vector will be translated into an observability condition in the next section. For now, we just mention that if this were not the case, then a reduced-order observer could not be designed.

We remark that the poles of the reduced-order observer shown in Fig. 7.17 are the eigenvalues of $\mathbf{\Phi}_{22}$, and $\mathbf{\Phi}_{22}$ is a submatrix of the matrix $\mathbf{\Phi}$ in the ZOH model of the plant. In the previous section we found that the observer had good performance when the observer poles were within the unit circle. If a similar result is true for reduced-order observers, then the simple observer in Fig. 7.17 will have good performance if the eigenvalues of $\mathbf{\Phi}_{22}$ happen to be within the unit circle. This observer is called simple because it is not necessary to do any calculation to obtain it – it comes from partitioning the ZOH equation of the plant.

We can make the above discussion about reduced-order observers rigorous by writing equations for the error between the actual and the estimated state variables. We will show

that it is true that the simple observer will work well if the eigenvalues of $\mathbf{\Phi}_{22}$ are within the unit circle. When this is not the case, we will show how an observer gains vector can be calculated to yield a reduced-order observer with good performance.

7.4.1 Analysis and Design of a Reduced-Order Observer

The simple reduced-order observer discussed above will be slightly generalized in this section. The performance of this modified reduced-order observer will be analyzed to see what requirements must be met in order for the estimation error to go to zero. The result of this analysis will be a set of design equations that can be used to obtain reduced-order observers with good performance.

The simple reduced-order observer is described by (7.25), which has the form of a state-update equation driven by two inputs. The state vector is $\hat{\mathbf{x}}_2[k]$ and the inputs are $u[k]$ and $\mathbf{y}_m[k]$, where $\mathbf{y}_m[k]$ is a vector-valued input. This reduced-order observer equation will be generalized in two ways. First, the state vector will be re-named $\mathbf{z}[k]$, and arbitrary matrices will be used in the state-update equation. Second, a "feedthrough" term from $\mathbf{y}_m[k]$ will be added to the state variables $\mathbf{z}[k]$ to form the estimates $\hat{\mathbf{x}}_2[k]$. Thus the general form of the reduced-order observer is given by

$$\begin{aligned} \mathbf{z}[k+1] &= \mathbf{F}\mathbf{z}[k] + \mathbf{G}\mathbf{y}_m[k] + \mathbf{H}u[k] \\ \hat{\mathbf{x}}_2[k] &= \mathbf{z}[k] + \mathbf{L}\mathbf{y}_m[k]. \end{aligned} \tag{7.26}$$

In the above equations, all of the matrices need to be specified in designing a reduced-order observer. These matrices are

$$\begin{aligned} \mathbf{F}_{(n-m)\times(n-m)} &= \text{system matrix for reduced-order observer.} \\ &\quad \text{The eigenvalues of } \mathbf{F} \text{ are the observer poles.} \\ \mathbf{G}_{m\times m} &= \text{input matrix for } \mathbf{y}[k]. \\ \mathbf{H}_{m\times 1} &= \text{input vector for } u[k]. \\ \mathbf{L}_{(n-m)\times m} &= \text{observer gains matrix.} \end{aligned} \tag{7.27}$$

The general observer structure in (7.26) can be specialized to the simple reduced-order observer in (7.25) by making the following choices for the matrices:

$$\begin{aligned} \mathbf{F} &= \mathbf{\Phi}_{22} \\ \mathbf{G} &= \mathbf{\Phi}_{21}\mathbf{C}_1^{-1} \\ \mathbf{H} &= \mathbf{\Gamma}_2 \\ \mathbf{L} &= \mathbf{0}. \end{aligned} \tag{7.28}$$

If this simple observer works well, there is no need for the observer gains matrix $\mathbf{L}$ (it is set to zero). However, if the simple reduced-order observer does not work well, then an observer will have to be designed using the equations developed in this section. A general reduced-order observer will have a non-zero observer gains matrix. Note that the simple reduced-order observer is a prediction observer, while the general observer with $\mathbf{L} \neq \mathbf{0}$ is a current observer.

In order to assess the behavior of any reduced-order observer, we have to analyze the equation for the error between the actual and the estimated state variables. We will analyze

the general observer equations given in (7.26) since they include the simple reduced-order observer as a special case. The analysis model is shown in block diagram form in Fig. 7.18.

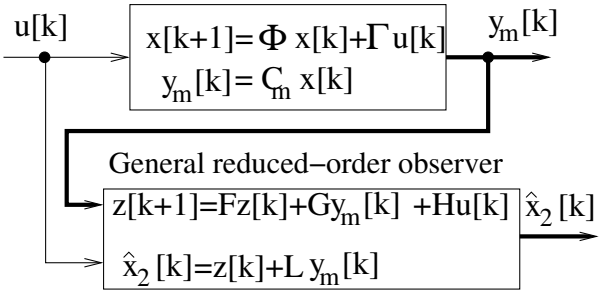


Figure 7.18 Analysis model for the general reduced-order observer.

Since the variables in $\mathbf{x}_1[k]$ are measured directly, they are not estimated and there is no estimation error. However, the variables in $\mathbf{x}_2[k]$ need to be estimated, and we are interested in the behavior of the estimation error

$$\mathbf{e}_2[k] \stackrel{\text{def}}{=} \mathbf{x}_2[k] - \hat{\mathbf{x}}_2[k]. \tag{7.29}$$

We will proceed with the analysis by writing an equation for $\mathbf{e}_2[k+1]$ in terms of $\mathbf{e}_2[k]$, $\mathbf{x}_1[k]$, $\mathbf{x}_2[k]$ and $u[k]$. We would like to end up with the following result

$$\mathbf{e}_2[k+1] = \mathbf{F}\mathbf{e}_2[k] \tag{7.30}$$

with $\mathbf{F}$ having eigenvalues inside the unit circle. If we can indeed obtain the above error equation, it would mean that the estimation error is independent of $\mathbf{x}_1[k]$, $\mathbf{x}_2[k]$, and $u[k]$ since they do not appear in the equation. In addition, the error would decay with time to zero. With (7.30) as a goal, we now proceed to write the error equation.

By definition,

$$\mathbf{e}_2[k+1] = \mathbf{x}_2[k+1] - \hat{\mathbf{x}}_2[k+1]. \tag{7.31}$$

Substituting in (7.23) for $\mathbf{x}_2[k+1]$ and (7.26) for $\hat{\mathbf{x}}_2[k+1]$, we obtain

$$\mathbf{e}_2[k+1] = \mathbf{\Phi}_{21}\mathbf{x}_1[k] + \mathbf{\Phi}_{22}\mathbf{x}_2[k] + \mathbf{\Gamma}_2 u[k] - (\mathbf{z}[k+1] + \mathbf{L}\mathbf{y}_m[k+1]). \tag{7.32}$$

In order to simplify this equation, we need to obtain expressions for $\mathbf{z}[k+1]$ and $\mathbf{y}_m[k+1]$. The expression for $\mathbf{z}[k+1]$ is obtained from (7.26) as follows:

$$\mathbf{z}[k+1] = \mathbf{F}\mathbf{z}[k] + \mathbf{G}\mathbf{y}_m[k] + \mathbf{H}u[k], \tag{7.33}$$

and we use the output equation in (7.26) to obtain

$$\mathbf{z}[k+1] = \mathbf{F}(\hat{\mathbf{x}}_2[k] - \mathbf{L}\mathbf{y}_m[k]) + \mathbf{G}\mathbf{y}_m[k] + \mathbf{H}u[k].$$

Substitute (7.29) into the above equation to obtain

$$\mathbf{z}[k+1] = \mathbf{F}(\mathbf{x}_2[k] - \mathbf{e}_2[k] - \mathbf{L}\mathbf{y}_m[k]) + \mathbf{G}\mathbf{y}_m[k] + \mathbf{H}u[k]$$

and finally, substitute $\mathbf{y}_m[k] = \mathbf{C}_1\mathbf{x}_1[k]$ to obtain

$$\mathbf{z}[k+1] = \mathbf{F}(\mathbf{x}_2[k] - \mathbf{e}_2[k]) - (\mathbf{F}\mathbf{K}\mathbf{C}_1 - \mathbf{G}\mathbf{C}_1)\mathbf{x}_1[k] + \mathbf{H}u[k]. \tag{7.34}$$

$$\mathbf{F} = \mathbf{\Phi}_{22} - \mathbf{L}\mathbf{C}_1\mathbf{\Phi}_{12} \quad \text{(D1)}$$

$$\mathbf{H} = \mathbf{\Gamma}_2 - \mathbf{L}\mathbf{C}_1\mathbf{\Gamma}_1 \quad \text{(D2)}$$

$$\mathbf{G} = (\mathbf{\Phi}_{21} - \mathbf{L}\mathbf{C}_1\mathbf{\Phi}_{11})\mathbf{C}_1^{-1} + \mathbf{F}\mathbf{L} \quad \text{(D3)}$$

Table 7.3 The design equations for a reduced-order observer.

The equation for $\mathbf{y}_m[k+1]$ is obtained from (7.20) and (7.23) as

$$\begin{aligned} \mathbf{y}_m[k+1] &= \mathbf{C}_1\mathbf{x}_1[k+1] \\ &= \mathbf{C}_1(\mathbf{\Phi}_{11}\mathbf{x}_1[k] + \mathbf{\Phi}_{12}\mathbf{x}_2[k] + \mathbf{\Gamma}_1 u[k]). \end{aligned} \tag{7.35}$$

We can now substitute (7.34) and (7.35) into (7.32) to obtain

$$\begin{aligned} \mathbf{e}_2[k+1] \;=\; & \mathbf{\Phi}_{21}[k+1]\mathbf{x}_1[k] + \mathbf{\Phi}_{22}\mathbf{x}_2[k] + \mathbf{\Gamma}_2 u[k] \\ & -(\mathbf{F}\mathbf{x}_2[k] - \mathbf{F}\mathbf{e}_2[k] + (\mathbf{G}\mathbf{C}_1 - \mathbf{F}\mathbf{L}\mathbf{C}_1)\mathbf{x}_1[k] + \mathbf{H}u[k]) \\ & -\mathbf{L}\mathbf{C}_1(\mathbf{\Phi}_{11}\mathbf{x}_1[k] + \mathbf{\Phi}_{12}\mathbf{x}_2[k] + \mathbf{\Gamma}_1 u[k]). \end{aligned} \tag{7.36}$$

This equation can be simplified by collecting terms as follows:

$$\begin{aligned} \mathbf{e}_2[k+1] \;=\; & \mathbf{F}\mathbf{e}_2[k] \;+\; (\mathbf{\Phi}_{21} + \mathbf{F}\mathbf{L}\mathbf{C}_1 - \mathbf{G}\mathbf{C}_1 - \mathbf{L}\mathbf{C}_1\mathbf{\Phi}_{11})\mathbf{x}_1[k] \\ & +\; (\mathbf{\Phi}_{22} - \mathbf{F} - \mathbf{L}\mathbf{C}_1\mathbf{\Phi}_{12})\mathbf{x}_2[k] \\ & +\; (\mathbf{\Gamma}_2 - \mathbf{H} - \mathbf{L}\mathbf{C}_1\mathbf{\Gamma}_1)u[k]. \end{aligned} \tag{7.37}$$

If all the terms in parentheses in the above equation were equal to zero, then the error equation would reduce to the desired expression $\mathbf{e}_2[k+1] = \mathbf{F}\mathbf{e}_2[k]$. Of course, we would still have to insure that the eigenvalues of $\mathbf{F}$ were inside the unit circle in order for the observer to have good performance. Recall that the matrices $\mathbf{F}$, $\mathbf{H}$, $\mathbf{G}$, and $\mathbf{L}$ have not yet been specified – they are arbitrary matrices which describe a general reduced-order observer as shown in (7.26). However, we now set the terms of (7.37) in parentheses to zero to get the desired error equation. Setting the three terms to zero will yield three equations which can be used to design reduced-order observers. These three design equations are shown in Table 7.3. All of the design equations are obtained from (7.37). Equation (D1) is obtained by setting the coefficient of $\mathbf{x}_2[k]$ to zero, (D2) is obtained by setting the coefficient of $u[k]$ to zero, and (D3) is obtained by setting the coefficient of $\mathbf{x}_1[k]$ to zero.

Remarks

1. If the observer gains matrix $\mathbf{L}$ can be found such that the resulting $\mathbf{F}$ matrix has desired eigenvalues, then the design equations (D2)–(D3) in Table 7.3 can be evaluated (multiplied out) to obtain $\mathbf{G}$ and $\mathbf{H}$.

2. If we make the choice $\mathbf{L} = \mathbf{0}$, then the design equations yield the following reduced-order observer matrices: $\mathbf{F} = \mathbf{\Phi}_{22}$, $\mathbf{H} = \mathbf{\Gamma}_2$, and $\mathbf{G} = \mathbf{\Phi}_{21}\mathbf{C}_1^{-1}$. This is exactly the simple reduced-order observer shown in Fig. 7.17. The calculations done in this

section show that the error equation for this observer is $\mathbf{e}_2[k+1] = \mathbf{\Phi}_{22}\mathbf{e}_2[k]$, so we have shown that if the eigenvalues of $\mathbf{\Phi}_{22}$ are within the unit circle, the simple observer will have the right qualitative behavior. If the eigenvalues are not within the unit circle, then $\mathbf{L}$ must be different from zero to get a good reduced-order observer.

3. The design equations in Table 7.3 need to be used whenever the eigenvalues of $\mathbf{\Phi}_{22}$ are not within the unit circle. The manner in which the design equations are used to obtain a reduced-order observer will be divided into three different cases. Each case depends on the relationship between m, the number of state variables measured, and n the dimension of the plant.

7.4.2 Case 1: $m \geq n/2$, rank$(\mathbf{\Phi_{12}}) = n - m$

This case applies when m, the number of measured signals, is greater than or equal to half the number of state variables. We are still assuming that the measurement matrix is

$$\mathbf{C}_m = [\,\mathbf{C}_1 \quad \mathbf{0}\,]$$

where $\mathbf{C}_1$ is an $m \times m$ invertible matrix. The general case in which $\mathbf{C}$ is an arbitrary $m \times n$ matrix is treated in Section 7.6. If the rank of $\mathbf{\Phi}_{12}$ is less than $n - m$, then the general solution (Case 3) must be used.

The design of a reduced-order observer requires that the observer gains matrix $\mathbf{L}$ be somehow calculated, and it is possible to begin our discussion with methods for calculating $\mathbf{L}$. However, it will be shown that for Case 1 we can start by specifying the matrix $\mathbf{F}$. Once $\mathbf{F}$ is specified, the matrix $\mathbf{L}$ can be found as the solution to a set of linear equations, and the other matrices can be calculated with a few matrix multiplications.

The solution procedure for Case 1 begins by specifying $\mathbf{F}$, the system matrix of the reduced-order observer. Since we want the eigenvalues of this matrix to be within the unit circle, we actually specify the eigenvalues of $\mathbf{F}$ first. Let us assume for simplicity that the desired eigenvalues of $\mathbf{F}$ are the real numbers $\lambda_1 \cdots \lambda_m$. Then we would choose $\mathbf{F}$ to be the diagonal matrix $\mathbf{F} = \text{diag}(\lambda_1, \cdots, \lambda_m)$. Note that we not only specify the eigenvalues of $\mathbf{F}$, *but we also specify the matrix* $\mathbf{F}$ *itself.* By specifying $\mathbf{F}$ to be a diagonal matrix, we will obtain a "decoupled" error equation $\mathbf{e}_2[k+1] = \mathbf{F}\mathbf{e}_2[k]$. By decoupled, we mean that the estimation error for a given state variable is not affected by the errors for the other state variables. Furthermore, the rate at which the error for each state variable goes to zero can be independently assigned by choosing λ_i appropriately. Once the matrix $\mathbf{F}$ is specified, design equation (D1) from Table 7.3 is solved for $\mathbf{L}$, and then (D2) and (D3) are used to obtain $\mathbf{H}$ and $\mathbf{G}$.

Let us examine how to solve (D1) for $\mathbf{L}$. If we take the transpose of (D1) we get

$$\underbrace{\mathbf{\Phi}_{12}^T\mathbf{C}_1^T}_{(n-m)\times m}\,\mathbf{L}^T = \mathbf{\Phi}_{22}^T - \mathbf{F}^T, \tag{7.38}$$

In the above equation, $\mathbf{\Phi}_{12}^T\mathbf{C}_1^T$ is an $(n-m) \times m$ coefficient matrix, $\mathbf{\Phi}_{22}^T - \mathbf{F}^T$ is an $(n-m) \times (n-m)$ matrix of right-hand sides, and each of the $n-m$ columns of $\mathbf{L}^T$ is a vector of m unknowns. Under Case 1, we assume that $m \geq n/2$ and rank$(\mathbf{\Phi}_{12}) = n - m$. Consider the special case in which $m = n/2$. In this case,

$$n - m = n - \frac{n}{2} = \frac{n}{2}$$

and so $n - m = m$. Thus the coefficient matrix in (7.38) is a square, nonsingular matrix, and we can solve for $\mathbf{L}^T$ as

$$\mathbf{L}^T = (\mathbf{\Phi}_{12}^T \mathbf{C}_1^T)^{-1}(\mathbf{\Phi}_{22}^T - \mathbf{F}^T).$$

Transposing this equation yields

$$\mathbf{L} = (\mathbf{\Phi}_{22} - \mathbf{F})(\mathbf{C}_1\mathbf{\Phi}_{12})^{-1}, \quad m = \frac{n}{2}. \tag{7.39}$$

In general under Case 1, we have $m \geq n/2$. When $m \geq n/2$, (7.38) contains $n - m$ equations in m unknowns for each column of $\mathbf{L}^T$, and $n - m < m$. This means that in (7.38), there are fewer equations $(n - m)$ than unknowns (m) that can be used to solve for each column of $\mathbf{K}^T$. In other words, the system of equations in (7.38) is *underdetermined*, and so it has an infinite number of solutions. We choose to use the unique "minimum norm" solution, because it results in a $\mathbf{L}$ matrix with the smallest possible entries, which is desirable from an implementation point of view. Since we assume that rank$(\mathbf{\Phi}_{12}) = n-m$ and $\mathbf{C}_1$ is nonsingular, the rank of the coefficient matrix in (7.38) is $n - m$ (i.e. full rank). Thus we can use the full rank formula for the minimum norm solution to a system of linear equations ((2.42) from Chapter 2). The minimum norm solution to (7.38) after taking transposes is

$$\mathbf{L} = (\mathbf{\Phi}_{22} - \mathbf{F})[(\mathbf{C}_1\mathbf{\Phi}_{12})^T(\mathbf{C}_1\mathbf{\Phi}_{12})]^{-1}(\mathbf{C}_1\mathbf{\Phi}_{12})^T. \tag{7.40}$$

Notice that the above equation reduces to (7.39) if $\mathbf{\Phi}_{12}$ is a square matrix (i.e. if $m = n/2$).

The above procedure can also be used when complex eigenvalues of $\mathbf{F}$ are desired. For each desired pair of complex-conjugate eigenvalues $\lambda_R \pm j\lambda_I$, we just need to define a 2×2 matrix $\mathbf{F}_i$ which has the desired complex eigenvalues. For instance, let $\mathbf{F}_i$ be the matrix

$$\begin{bmatrix} \lambda_R & \lambda_I \\ -\lambda_I & \lambda_R \end{bmatrix}.$$

The complete design procedure for Case 1 is shown in Table 7.4.

EXAMPLE 7.5

In this example we calculate reduced-order observer matrices for the plant given in Example 7.3. The ZOH equivalent of the plant at a sampling interval of $T = 0.1$ can be calculated to be

$$\mathbf{\Phi} = \begin{bmatrix} 1.0000 & 0.0952 & 0.0042 \\ 0 & 0.9048 & 0.0782 \\ 0 & 0 & 0.6703 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0001 \\ 0.0042 \\ 0.0824 \end{bmatrix}.$$

If we assume that the first two state variables can be measured $(m = 2)$ then the measurement matrix is

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

1. Specify the desired eigenvalues of $\mathbf{F}$ by choosing $f_1, f_2, \cdots$. For real-valued eigenvalues λ, f_i is equal to λ_i. For complex conjugate eigenvalues $\lambda_R \pm j\lambda_I$, f_i is a 2×2 matrix whose eigenvalues are $\lambda_R \pm j\lambda_I$.
2. Set $\mathbf{F} = \text{diag}(f_1, f_2, \cdots)$.
3. Solve for $\mathbf{L}$ from (D1). The minimum norm solution is

$$\mathbf{L} = (\mathbf{\Phi}_{22} - \mathbf{F})[(\mathbf{C}_1\mathbf{\Phi}_{12})^T(\mathbf{C}_1\mathbf{\Phi}_{12})]^{-1}(\mathbf{C}_1\mathbf{\Phi}_{12})^T.$$

   In the special case that $m = n/2$, the above equation reduces to

$$\mathbf{L} = (\mathbf{\Phi}_{22} - \mathbf{F})(\mathbf{C}_1\mathbf{\Phi}_{12})^{-1}.$$

4. Calculate $\mathbf{H}$ and $\mathbf{G}$ using (D2) and (D3) as follows:

$$\mathbf{H} = \mathbf{\Gamma}_2 - \mathbf{L}\mathbf{C}_1\mathbf{\Gamma}_1$$
$$\mathbf{G} = (\mathbf{\Phi}_{21} - \mathbf{L}\mathbf{C}_1\mathbf{\Phi}_{11})\mathbf{C}_1^{-1} + \mathbf{F}\mathbf{L}$$

Table 7.4 The design procedure for Case 1.

and $\mathbf{C}_1 = \mathbf{I}$. The ZOH of the plant is partitioned as follows

$$\mathbf{\Phi}_{11} = \begin{bmatrix} 1.0000 & 0.0952 \\ 0 & 0.9048 \end{bmatrix} \quad \mathbf{\Phi}_{12} = \begin{bmatrix} 0.0042 \\ 0.0782 \end{bmatrix}$$

$$\mathbf{\Phi}_{21} = \begin{bmatrix} 0 & 0 \end{bmatrix} \qquad \mathbf{\Phi}_{22} = 0.6703$$

$$\mathbf{\Gamma}_1 = \begin{bmatrix} 0.0001 \\ 0.0042 \end{bmatrix} \qquad \mathbf{\Gamma}_2 = 0.0824.$$

The state vector is partitioned as

$$\mathbf{x}_1[k] = \begin{bmatrix} x_1[k] \\ x_2[k] \end{bmatrix}, \quad \text{and } \mathbf{x}_2[k] = x_3[k].$$

Since $m = 2$, $n = 3$, and $\mathbf{\Phi}_{12}$ is non-zero (and thus has rank 1), this example fits under Case 1.

It is interesting to note that for this example, the eigenvalue of $\mathbf{\Phi}_{22}$ is 0.6703 which is within the unit circle. This means that the simple observer shown in Fig. 7.17 could be used and the estimation error would go to zero. The number of samples it would take for the error to go approximately to zero is (see (5.23) in Chapter 5)

$$N_T = -\frac{4.6}{\log(0.6703)} \approx 11.5 \text{ samples.}$$

Since the sampling interval is 0.1 seconds, this corresponds to an observer settling time of about 1.15 seconds.

Here we assume that the desired observer settling time is 0.5 seconds, and so the design equations will have to be used. We first calculate the value of the observer eigenvalue corresponding to a settling time of 0.5 seconds. The calculation consists of scaling the root of a first-order Bessel polynomial (see Table 6.2 on page 257) to

achieve a settling time of 0.5 seconds, and then mapping the result into the z-plane using the ZOH pole mapping formula with a sampling period of 0.1 seconds. The resulting eigenvalue is

$$f = e^{(-4.62/0.5)0.1} = 0.3969.$$

In general we would set the observer matrix $\mathbf{F}$ to be a diagonal matrix with the number 0.3969 on the diagonal. In this example, however, $\mathbf{F}$ is a 1×1 matrix and we simply have

$$\mathbf{F} = 0.3969.$$

Once $\mathbf{F}$ is chosen, we calculate $\mathbf{L}$ from Step 3 in the procedure shown in Table 7.4 with the result (recall $\mathbf{C}_1 = \mathbf{I}$ in this example)

$$\begin{aligned}\mathbf{L} &= (\mathbf{\Phi}_{22} - \mathbf{F})(\mathbf{\Phi}_{12}^T \mathbf{\Phi}_{12})^{-1} \mathbf{\Phi}_{12}^T \\ &= (0.6703 - 0.3969) \left(\begin{bmatrix} .0042 & .0782 \end{bmatrix} \begin{bmatrix} .0042 \\ .0782 \end{bmatrix} \right)^{-1} \begin{bmatrix} .0042 & .0782 \end{bmatrix} \\ &= \begin{bmatrix} 0.1895 & 3.4874 \end{bmatrix}.\end{aligned}$$

Finally, $\mathbf{H}$ and $\mathbf{G}$ are calculated from Step 4 to be

$$\begin{aligned}\mathbf{H} &= \mathbf{\Gamma}_2 - \mathbf{K}\mathbf{\Gamma}_1 \\ &= 0.0824 - \begin{bmatrix} 0.1895 & 3.4874 \end{bmatrix} \begin{bmatrix} .0001 \\ .0042 \end{bmatrix} \\ &= 0.0676\end{aligned}$$

and

$$\begin{aligned}\mathbf{G} &= \mathbf{\Phi}_{21} - \mathbf{K}\mathbf{\Phi}_{11} + \mathbf{F}\mathbf{L} \\ &= \begin{bmatrix} 0 & 0 \end{bmatrix} - \begin{bmatrix} 0.1895 & 3.4874 \end{bmatrix} \begin{bmatrix} 1.0000 & 0.0952 \\ 0.0000 & 0.9048 \end{bmatrix} + 0.3969 \begin{bmatrix} 0.1895 & 3.4874 \end{bmatrix} \\ &= \begin{bmatrix} -0.1143 & -1.7894 \end{bmatrix}.\end{aligned}$$

The reduced-order observer generates estimates of $x_3[k]$ (the only state variable that is not measured) from the measured state variables in $\mathbf{x}_1[k]$ and the input $u[k]$ according to the following equations

$$\begin{aligned}\mathbf{z}[k+1] &= \mathbf{F}\mathbf{z}[k] + \mathbf{G}\mathbf{x}_1[k] + \mathbf{H}u[k] \\ \hat{x}_3[k] &= \mathbf{z}[k] + \mathbf{L}\mathbf{x}_1[k].\end{aligned}$$

The selection of the initial state vector $\mathbf{z}[0]$ is discussed in the following section, and example is given to show the behavior of a regulator which uses $x_1[k]$, $x_2[k]$, and $\hat{x}_3[k]$ for feedback.

■

7.4.3 Case 2: $m = 1$

This is the case in which only a single state variable can be measured, or equivalently, the system has only one output that is measured. In this case, it is not possible to choose the $\mathbf{F}$ matrix to be diagonal in general. If we try to choose $\mathbf{F}$ to be diagonal and solve for $\mathbf{L}$ as before, the resulting equations will turn out to be overdetermined and inconsistent, and thus not have a solution. Nevertheless, we still want to satisfy design equation (D1) which is repeated below

$$\mathbf{F} = \underbrace{\mathbf{\Phi}_{22}}_{(n-1)\times(n-1)} - \underbrace{\mathbf{K}}_{(n-1)\times 1}\underbrace{\mathbf{C}_1\mathbf{\Phi}_{12}}_{1\times(n-1)}. \tag{7.41}$$

In order for the reduced-order observer to have good performance, we only need the *eigenvalues* of $\mathbf{F}$ to be within the unit circle. If $\mathbf{F}$ is also a diagonal matrix, then the estimation errors will be decoupled, but decoupling is not necessary for the errors to go to zero. Specifying the eigenvalues of $\mathbf{F}$ in (7.41) is a standard observer gains calculation for a system of order $n-1$. By standard calculation, we are referring to the design of a full-order observer as in (7.7).

Recall that for full-order observers, we specify the eigenvalues of $\mathbf{F}$, and then calculate the observer gains vector which will achieve those eigenvalues. The calculation uses the pole placement formulas derived in Chapter 6. The difference between the full-order observer discussed at the beginning of this chapter and a Case 2 reduced-order observer is this: the full-order observer is an n^{th}-order system, while a Case 2 reduced-order observer is an $(n-1)$st order system. To design a reduced-order observer for Case 2, the observer gains vector must be calculated by using the "pole placement" formulas for (7.41). The complete design procedure for Case 2 is shown in Table 7.5.

EXAMPLE 7.6

We repeat the previous example assuming that only the first state variable is measured ($m = 1$). In this case the measurement matrix is

$$\mathbf{C}_m = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \text{ and } \mathbf{C}_1 = 1.$$

The partitions of the plant ZOH are

$$\mathbf{\Phi}_{11} = 1 \qquad \mathbf{\Phi}_{12} = \begin{bmatrix} 0.0952 & 0.0042 \end{bmatrix}$$

$$\mathbf{\Phi}_{21} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \mathbf{\Phi}_{22} = \begin{bmatrix} 0.9048 & 0.0782 \\ 0 & 0.6703 \end{bmatrix}$$

$$\mathbf{\Gamma}_1 = 0.0001 \quad \mathbf{\Gamma}_2 = \begin{bmatrix} 0.0042 \\ 0.0824 \end{bmatrix}.$$

The eigenvalues of $\mathbf{\Phi}_{22}$ are 0.9048 and 0.6703 which are both within the unit circle, and so the simple reduced-order observer could be used. However, the eigenvalue at 0.9048 would result in a long settling time for the observer (about 4.6 seconds).

As in the previous example, we choose a settling time of 0.5 seconds. The observer poles will be specified as the roots of a 2nd-order Bessel polynomial scaled to have a settling time of 0.5 seconds and mapped into the z-plane. The result is

$$z_{1,2} = e^{[(-4.0530 \pm j2.3400)/0.5]0.1} = 0.3968 \pm j0.2006.$$

1. Choose desired observer poles as the roots of an $(n-1)$st degree polynomial $p(z) = z^{n-1} + p_1 z^{n-2} + \cdots + p_{n-1}$. Let the characteristic polynomial of $\mathbf{\Phi}_{22}$ be denoted by $z^{n-1} + a_1 z^{n-2} + \cdots + a_{n-1}$.
2. Solve for $\mathbf{L}$ by applying the pole placement formula to (D1). The solution is (compare with (7.9))

$$\mathbf{L} = \begin{bmatrix} \mathbf{C}_1\mathbf{\Phi}_{12} \\ \mathbf{C}_1\mathbf{\Phi}_{12}\mathbf{\Phi}_{22} \\ \vdots \\ \mathbf{C}_1\mathbf{\Phi}_{12}\mathbf{\Phi}_{22}^{n-2} \end{bmatrix}^{-1} \bar{\mathbf{W}}_c^T \begin{bmatrix} p_1 - a_1 \\ p_2 - a_2 \\ \vdots \\ p_{n-1} - a_{n-1} \end{bmatrix}$$

where

$$\bar{\mathbf{W}}_c = [\bar{\mathbf{\Gamma}}_2 \ \bar{\mathbf{\Phi}}_{22}\bar{\mathbf{\Gamma}}_2 \ \cdots \bar{\mathbf{\Phi}}_{22}^{n-2}\bar{\mathbf{\Gamma}}_2]$$

and

$$\bar{\mathbf{\Phi}}_{22} = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{n-1} \\ 1 & 0 & \cdots & 0 \\ 0 & \ddots & \vdots & 0 \\ 0 & \cdots & 1 & 0 \end{bmatrix}, \quad \bar{\mathbf{\Gamma}}_2 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Notice that $\mathbf{L}$ can be calculated only if the observability matrix for $(\mathbf{\Phi}_{22}, \mathbf{C}_1\mathbf{\Phi}_{12})$ is invertible. It can be shown that this matrix will be invertible if and only if the full ZOH model $(\mathbf{\Phi}, \mathbf{C})$ is observable. If the ZOH model is not observable, a reduced-order observer cannot be designed using a single measurement.

3. Calculate $\mathbf{F}$, $\mathbf{H}$, and $\mathbf{G}$ using (D1) – (D3) as follows:

$$\mathbf{F} = \mathbf{\Phi}_{22} - \mathbf{L}\mathbf{C}_1\mathbf{\Phi}_{12}$$
$$\mathbf{H} = \mathbf{\Gamma}_2 - \mathbf{L}\mathbf{C}_1\mathbf{\Gamma}_1$$
$$\mathbf{G} = (\mathbf{\Phi}_{21} - \mathbf{L}\mathbf{C}_1\mathbf{\Phi}_{11})\mathbf{C}_1^{-1} + \mathbf{F}\mathbf{L}$$

Table 7.5 The design procedure for Case 2.

These roots can be multiplied out to obtain the polynomial

$$\begin{aligned} p(z) &= (z - 0.3968 + j0.2006)(z - 0.3968 - j0.2006) \\ &= z^2 - 0.7936z + 0.1977. \end{aligned}$$

The characteristic polynomial of $\mathbf{\Phi}_{22}$ can be obtained by multiplying out the eigenvalues of $\mathbf{\Phi}_{22}$

$$\begin{aligned} a(z) &= (z - 0.9048)(z - 0.6703) \\ &= z^2 - 1.5752z + 0.6065. \end{aligned}$$

This completes Step 1 of the procedure for Case 2 shown in Table 7.5.

In Step 2 we form

$$\bar{\mathbf{\Phi}}_{22} = \begin{bmatrix} 1.5752 & -0.6065 \\ 1 & 0 \end{bmatrix}, \quad \bar{\mathbf{\Gamma}}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and calculate

$$\begin{aligned} \bar{\mathbf{W}}_c &= \begin{bmatrix} \bar{\mathbf{\Gamma}}_2 & \bar{\mathbf{\Phi}}_{22}\bar{\mathbf{\Gamma}}_2 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1.5752 \\ 0 & 1.0000 \end{bmatrix}, \end{aligned}$$

and

$$\begin{bmatrix} \mathbf{\Phi}_{12} \\ \mathbf{\Phi}_{12}\mathbf{\Phi}_{22} \end{bmatrix} = \begin{bmatrix} 0.0952 & 0.0042 \\ 0.0861 & 0.0103 \end{bmatrix}.$$

We use these matrices to calculate $\mathbf{L}$ as follows

$$\begin{aligned} \mathbf{L} &= \begin{bmatrix} 0.0952 & 0.0042 \\ 0.0861 & 0.0103 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1.5752 \\ 0 & 1.0000 \end{bmatrix}^T \begin{bmatrix} 0.7816 \\ -0.4089 \end{bmatrix} \\ &= \begin{bmatrix} 7.4162 \\ 17.8557 \end{bmatrix}. \end{aligned}$$

Using the formulas in Step 3 of the procedure in Table 7.5, we calculate the remaining matrices of the reduced-order observer with the following results

$$\mathbf{F} = \begin{bmatrix} 0.1991 & 0.0467 \\ -1.6992 & 0.5945 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 0.0032 \\ 0.0798 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} -5.1063 \\ -19.8425 \end{bmatrix}.$$

It can be verified that $\mathbf{F}$ has the desired eigenvalues.

■

1. Choose desired observer poles $\lambda_1, \cdots, \lambda_{n-m}$.
2. Use the multivariable pole-placement algorithm from Chapter 9 with $\mathbf{\Phi}_{22}^T$ and $(\mathbf{C}_1\mathbf{\Phi}_{12})^T$ in place of $\mathbf{\Phi}$ and $\mathbf{\Gamma}$. The resulting matrix will be the transpose of the observer gains matrix.
3. Calculate $\mathbf{H}$ and $\mathbf{G}$ using (D2) and (D3) as follows:

$$\mathbf{H} = \mathbf{\Gamma}_2 - \mathbf{L}\mathbf{C}_1\mathbf{\Gamma}_1$$
$$\mathbf{G} = (\mathbf{\Phi}_{21} - \mathbf{L}\mathbf{C}_1\mathbf{\Phi}_{11})\mathbf{C}_1^{-1} + \mathbf{F}\mathbf{L}$$

Table 7.6 The design procedure for Case 3.

7.4.4 Case 3: General Case

Case 3 is the most general since it applies to any situation in which an observer might be needed. However, it is also the most difficult. For that reason, if a problem fits under Cases 1–2, the solution procedures for those cases should be used. Case 3 should be used only when absolutely necessary! An examination of possible values of m, the number of measured signals, and n, the order of the plant, shows that we *never* have to use Case 3 if the order of the plant is less than or equal to four. For example, if the plant model is fourth order, we can have one measurement (Case 2), two or three measurements (Case 1), or four measurements which amounts to full-state feedback. The simplest case in which a Case 3 solution is required is a fifth-order system with two measurements.

The key design equation for Case 3 is still (D1), where we want to choose the $(n - m) \times m$ matrix $\mathbf{L}$ so that $\mathbf{F}$ has eigenvalues inside the unit circle. As with the design of a full-order observer and the Case 2 design, the algebraic problem of placing the eigenvalues of $\mathbf{F}$ at desired locations is a pole-placement problem. However, Case 3 is a *multivariable pole placement problem*. In other words, it is analogous to pole placement for a system with more than one input. The solution to the multivariable pole placement problem is given in Chapter 9, and the procedure given there can be used to design reduced-order observers under Case 3. The design procedure for Case 3 is given in Table 7.6.

We conclude our discussion of Case 3 by saying that in order for a solution to exist, the observability matrix for $(\mathbf{\Phi}_{22},\ \mathbf{C}_1\mathbf{\Phi}_{12})$ must be full rank. If this condition is not met, then a reduced-order observer cannot be designed.

7.5 Reduced-Order Observers for Regulation

To begin the discussion, assume that the plant we want to regulate is described by the ZOH model $(\mathbf{\Phi},\ \mathbf{\Gamma})$. Suppose that we have designed a feedback vector $\mathbf{K}$ using the techniques of Chapter 6 such that the closed-loop system with full state feedback has desired pole locations. In other words, the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}$ are the desired closed-loop pole locations.

Now suppose that we can only measure the first m state variables of the system; that is, the measurement vector is $\mathbf{y}_m[k] = \mathbf{C}_1\mathbf{x}_1[k]$, where $\mathbf{x}_1[k]$ consists of the first m state variables of the plant, and $\mathbf{C}_1$ is a nonsingular matrix. Since the state vector has been

partitioned, we also partition the feedback vector as follows

$$\mathbf{K} = [\underbrace{\mathbf{K}_1}_{m} \quad \underbrace{\mathbf{K}_2}_{(n-m)}]. \tag{7.42}$$

In order to use state feedback, we feed the measured state variables through the feedback vector $\mathbf{K}_1$, and the estimated state variables through the feedback vector $\mathbf{K}_2$, and add the results. This yields the closed loop system shown in Fig. 7.19. We now proceed to analyze

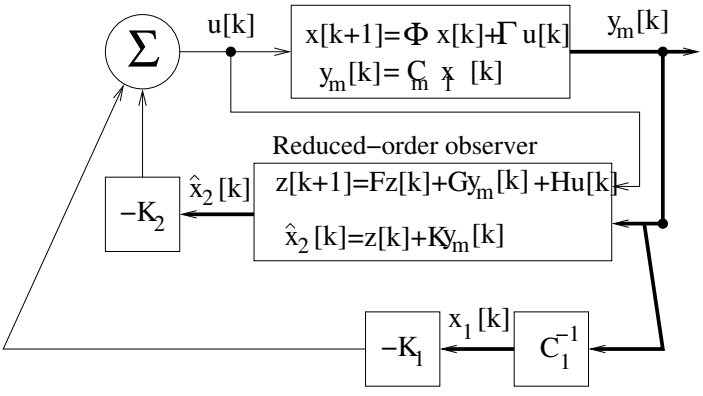


Figure 7.19 A closed-loop system with state feedback where some of the state variables are measured, and some are estimated by a reduced-order observer.

the behavior of this system.

The system in Fig. 7.19 is an interconnection of an n^{th}-order system (the plant) and an $(n-m)th$ order system (the reduced-order observer). We assume that the observer has been designed in accordance with design equations (D1)–(D3). The complete closed-loop system has dimension $2n-m$, and so we need to define $2n-m$ state variables in order to describe it. The "natural" choice of state variables to use for the interconnection of two systems is the concatenation of the state vectors of the individual systems (see Chapter 3). That is, we could define the state vector.

$$\begin{pmatrix} \mathbf{x}[k] \\ \mathbf{z}[k] \end{pmatrix}.$$

This choice of state vector can indeed be used to describe the system in Fig. 7.19. However, the analysis that follows turns out to be easier if the following state vector is chosen

$$\begin{pmatrix} \mathbf{x}[k] \\ \mathbf{e}_2[k] \end{pmatrix} \tag{7.43}$$

where $\mathbf{e}_2[k] = \mathbf{x}_2[k] - \hat{\mathbf{x}}_2[k]$. An update equation for this state vector can be obtained by writing the equations of the plant and observer, and using the equations which describe their interconnection. The plant is described by

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k] \tag{7.44}$$

and the feedback connection is described by

$$u[k] = -\mathbf{K}_1\mathbf{x}_1[k] - \mathbf{K}_2\hat{\mathbf{x}}_2[k]. \tag{7.45}$$

If we substitute (7.45) in (7.44) we obtain

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] - \mathbf{\Gamma}\mathbf{K}_1\mathbf{x}_1[k] - \mathbf{\Gamma}\mathbf{K}_2\hat{\mathbf{x}}_2[k] \tag{7.46}$$

and if we substitute $\hat{\mathbf{x}}_2[k] = \mathbf{x}_2[k] - \mathbf{e}_2[k]$ into the above equation we obtain

$$\begin{aligned}\mathbf{x}[k+1] &= \mathbf{\Phi}\mathbf{x}[k] \underbrace{-\mathbf{\Gamma}\mathbf{K}_1\mathbf{x}_1[k] - \mathbf{\Gamma}\mathbf{K}_2\hat{\mathbf{x}}_2[k]}_{-\mathbf{\Gamma}\mathbf{K}\mathbf{x}[k]} + \mathbf{\Gamma}\mathbf{K}_2\mathbf{e}_2[k] \\ &= (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})\mathbf{x}[k] + \mathbf{\Gamma}\mathbf{K}_2\mathbf{e}_2[k].\end{aligned} \tag{7.47}$$

The above equation is a state update equation for $\mathbf{x}[k]$ in terms of the chosen state vector (7.43). We now need an update equation for $\mathbf{e}_2[k]$. In fact, we already have this equation because we assumed that the reduced-order observer was calculated using the design equations (D1)–(D3). Those equations were formulated to yield the error equation

$$\mathbf{e}_2[k+1] = \mathbf{F}\mathbf{e}_2[k] \tag{7.48}$$

and this is the required update equation. Combining (7.47) and (7.48), we obtain the following description of the closed-loop system shown in Fig. 7.19 (notice that there is no external input since we are considering a regulator system)

$$\begin{bmatrix}\mathbf{x}[k+1] \\ \mathbf{e}_2[k+1]\end{bmatrix} = \underbrace{\begin{bmatrix}\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K} & \mathbf{\Gamma}\mathbf{K}_2 \\ 0 & \mathbf{F}\end{bmatrix}}_{\mathbf{\Phi_{CL}}} \begin{bmatrix}\mathbf{x}[k] \\ \mathbf{e}_2[k]\end{bmatrix}. \tag{7.49}$$

The poles of the closed-loop system are the eigenvalues of the matrix $\mathbf{\Phi_{CL}}$. Since this matrix is block upper-triangular, its eigenvalues are just the eigenvalues of the diagonal blocks. Thus we have that

$$\text{eig}(\mathbf{\Phi_{CL}}) = \text{eig}(\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}) \cup \text{eig}(\mathbf{F}) \tag{7.50}$$

where the symbol $\cup$ denotes union. But the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}$ are the desired closed-loop poles that were specified when the $\mathbf{K}$ vector was designed for full state feedback, and the eigenvalues of $\mathbf{F}$ are the observer poles which were chosen to be well within the unit circle. Equation (7.50) is the *the separation principle* for reduced-order observers. See (7.13) for a similar result using full-order observers.

In order to find the stability margins of a reduced-order observer-based compensator, we need to calculate the loop transfer function of the system shown in Fig. 7.19 with the loop broken at the input to the plant. This calculation is the topic of Problem 2 at the end of this chapter. A similar result for full-order observers was given in Section 7.3.2.

The poles of a reduced-order observer should be chosen using the guidelines that were developed in Section 7.3.3 for full-order observers. In particular, eigenvalues of $\mathbf{F}$ should be chosen to cancel "significant" zeros of the "plant," where the plant is the system $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C}_m)$. If $\mathbf{C}_m$ is a row vector (only one measured signal), then the plant is a single input, single output system whose zeros can be computed as the roots of the numerator polynomial of the transfer function. However, if $\mathbf{C}_m$ is a matrix (more than one measured signal), then the plant is a multivariable system with one input and more than one output. The zeros of a multivariable plant are called *transmission zeros*. A transmission zero is defined to be a number γ such that the matrix

$$\begin{bmatrix}\gamma\mathbf{I} - \mathbf{\Phi} & \mathbf{\Gamma} \\ \mathbf{C}_m & \mathbf{0}\end{bmatrix}$$

is rank deficient (see Problem 28 in Chapter 3). The transmission zeros of a multivariable plant can be computed using the Digital Control Toolbox. We remark that a system with one input and several outputs usually does not have *any* transmission zeros. Thus in the case of a single input plant with multiple measurements, the eigenvalues of the reduced-order observer matrix $\mathbf{F}$ can usually be chosen simply on the basis of settling time.

Initializing a Reduced-Order Observer It is important to initialize the state vector of the reduced-order observer in order to avoid large transients when the loop is closed. A simple rule for initialization is given here. Since we are dealing with the regulation problem, the desired value for all the state variables is zero. However, due to disturbances, the actual values of the state variables may be different from zero. The state variables in the observer are used to estimate the plant state variables $\mathbf{x}_2[k]$. How can we obtain an initial estimate $\hat{\mathbf{x}}_2(0)$ of the vector $\mathbf{x}_2[k]$? A reasonable choice is to set $\hat{\mathbf{x}}_2(0) = 0$. This is equivalent to initializing the state of the observer to

$$\mathbf{z}[0] = -\mathbf{L}\mathbf{y}_m[0]. \tag{7.51}$$

EXAMPLE 7.7

In this example we consider a regulator which uses a reduced-order observer to estimate some of the state variables of the plant. A block diagram of such an system is shown in Fig. 7.19. In that figure the ZOH of the plant is shown, instead of the continuous-time plant surrounded by D/A and A/D converters.

The plant for this example is specified by the following state-space matrices

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{C}_m = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

We assume that x_1 and x_2 are measured directly. This plant appears in several examples in Chapters 6 and 7. In particular, in Example 6.8 on page 258 a feedback vector $\mathbf{K}$ was calculated to place the poles of the closed-loop system (with full-state feedback) to achieve a settling time of 2 seconds. The feedback vector was

$$\mathbf{K} = [\,17.4134 \quad 11.4014 \quad 1.6358\,].$$

In Example 7.5 on page 321 a reduced-order observer which estimates x_3 was obtained, assuming x_1 and x_2 were measured. The observer matrices were

$$\mathbf{F} = 0.3969$$

$$\mathbf{G} = [\,-0.1143 \quad -1.7894\,]$$

$$\mathbf{H} = 0.0676$$

$$\mathbf{L} = [\,0.1895 \quad 3.4874\,].$$

In this example we combine the reduced-order observer with the partitioned state-feedback vector

$$\mathbf{K}_1 = [\,17.4134 \quad 11.4014\,], \quad \mathbf{K}_2 = [\,1.6358\,]$$

in a regulator structure like that shown in Fig. 7.19.

Suppose that the initial state vector of the plant is

$$\mathbf{x}[0] = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

and the purpose of the regulator is to drive all the state variables to zero. In this case the measurements at time zero will be

$$\begin{aligned} \mathbf{y}_m[0] &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned}$$

In order to implement a reduced-order observer, the initial state vector $\mathbf{z}[0]$ must be set to some value. Equation (7.51) suggests the initialization $\mathbf{z}[0] = -\mathbf{L}\mathbf{y}_m[0]$ which for this example is

$$\mathbf{z}[0] = \begin{bmatrix} 0.1895 & 3.4874 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = -3.4874.$$

We will compare this initialization with the trivial initialization $\mathbf{z}[0] = \mathbf{0}$.

In Fig. 7.20 the left-hand column of figures shows the transient response of the regulator with $\mathbf{z}[0] = -3.4874$, and the right-hand side figures show the performance with $\mathbf{z}[0] = \mathbf{0}$. For this example, there is not too much difference in the transient responses. However, the initializing $\mathbf{z}$ to $\mathbf{0}$ causes a larger initial value of the control input produced by the regulator. This could be a problem in a system subject to actuator saturation.

The stability margins for this regulator can be found by looking at a Nyquist plot of the loop transfer function with the loop broken at the input to the plant (see Problem 2). The stability margins for this regulator are

$$\text{GM} = 19.1 \text{ dB}$$

$$\text{PM} = 55.6^\circ.$$

In Example 6.8 on page 258 the stability margins for the full-state feedback regulator using the same $\mathbf{K}$ vector were GM = 21.7 dB, and PM = 60°. Using an observer to estimate x_3 has resulted in a slight decrease in the stability margins of the regulator. ■

EXAMPLE 7.8

We repeat the previous example but now assume that only x_1 can be measured. Thus the measurement matrix is

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

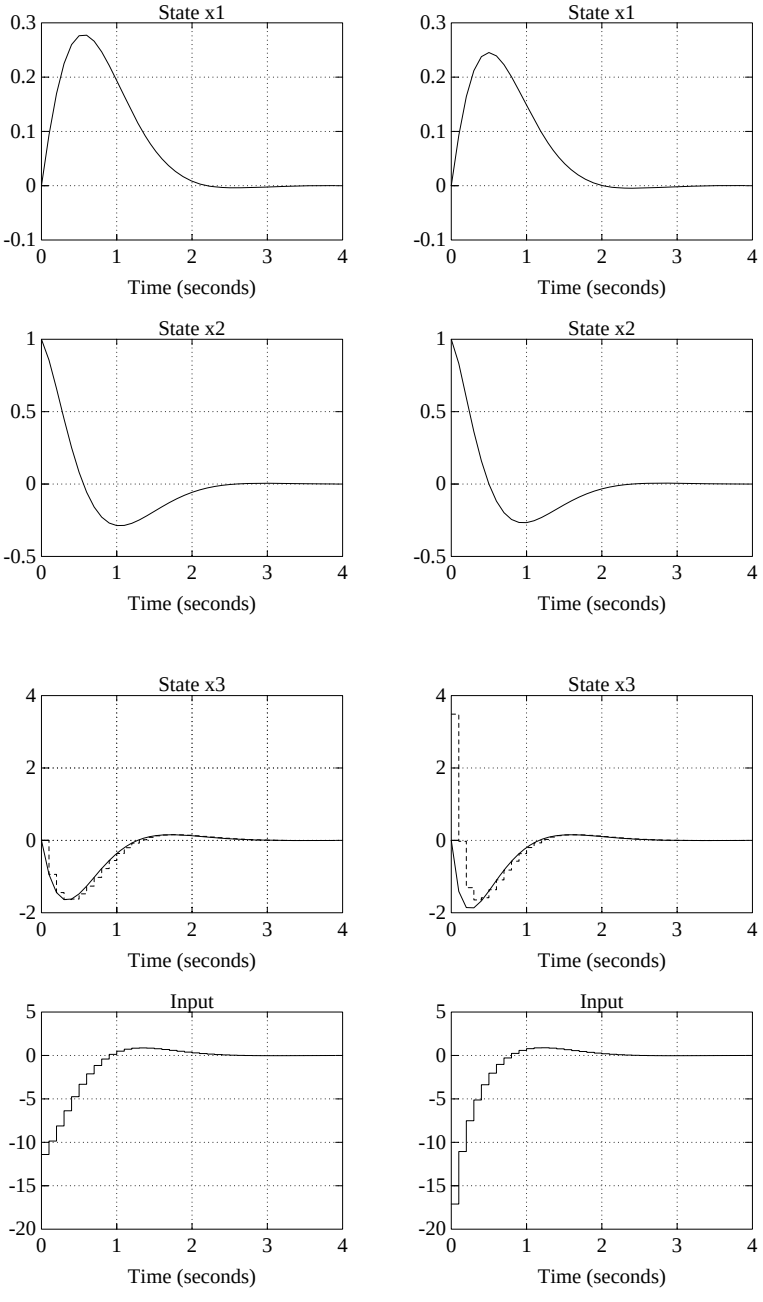


Figure 7.20 Example 7.7 – The graphs in the left-hand column are for the observer-based regulator with $\mathbf{z}[0] = -\mathbf{L}\mathbf{y}_m[0]$. The solid lines are the plant state variables and the dashed lines are the estimates produced by the observer. The graphs in the right-hand column are for the observer-based regulator with $\mathbf{z}[0] = \mathbf{0}$.

The reduced-order observer for this example will estimate the values of x_2 and x_3, and so will be a second-order system. In this example the system $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C}_m)$ is a single-input, single-output system, and we can calculate its zeros. The zeros are located at

$$z = -3.3039 \text{ and } -0.2357.$$

From the discussion in Section 7.3.3 on page 312, we know that observer poles should be chosen to cancel "significant" plant zeros in order for an observer-based regulator to have approximately the same stability margins as a full-state feedback regulator with the same $\mathbf{K}$ vector. The zero outside the unit circle in this example can be reflected inside the unit circle by taking its reciprocal. This yields a possible choice for the eigenvalues of $\mathbf{F}$ as

$$-\frac{1}{3.3039} = -0.3027 \text{ and } -0.2357.$$

Both of these eigenvalues are on the negative real axis, which usually can be considered "far" from the desired closed-loop poles. Using the same $\mathbf{K}$ vector as in the previous example, the eigenvalues of $(\mathbf{\Phi} - \mathbf{\Gamma K})$ are

$$0.7784,\ 0.8055 \pm j0.1543.$$

Nevertheless we can design a reduced-order observer with the eigenvalues of $\mathbf{F}$ at -0.3027 and -0.2357. The resulting observer matrices are

$$\mathbf{F} = \begin{bmatrix} -0.6275 & 0.0098 \\ -13.0209 & 0.0891 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} -24.8703 \\ -334.3020 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 0.0019 \\ 0.0622 \end{bmatrix}$$

$$\mathbf{L} = \begin{bmatrix} 16.1027 \\ 136.8275 \end{bmatrix}.$$

We will compare the performance a regulator which uses this observer with another regulator which uses the observer designed in Example 7.6 which chose the observer poles using a 2nd-order Bessel polynomial. Both observers are initialized with $\mathbf{z}[0] = -\mathbf{L}\mathbf{y}_m[0]$.

In Fig. 7.21 the left-hand column of figures shows the transient response of the regulator with observer poles on the negative real axis. The right-hand side figures show the performance of the regulator with observer poles corresponding to a Bessel polynomial. It can be seen that the observer poles on the negative real axis cause the state estimates to oscillate between positive and negative values. As a result, the input signal produced by this regulator has a large initial "spike." Thus the regulator with observer poles corresponding to a Bessel polynomial has the preferred transient behavior.

The two regulators can also be compared on the basis of stability margins. The margins for the system with observer poles on the negative real axis are

$$\text{GM} = 16.9 \text{ dB}$$

$$\text{PM} = 57.7^\circ.$$

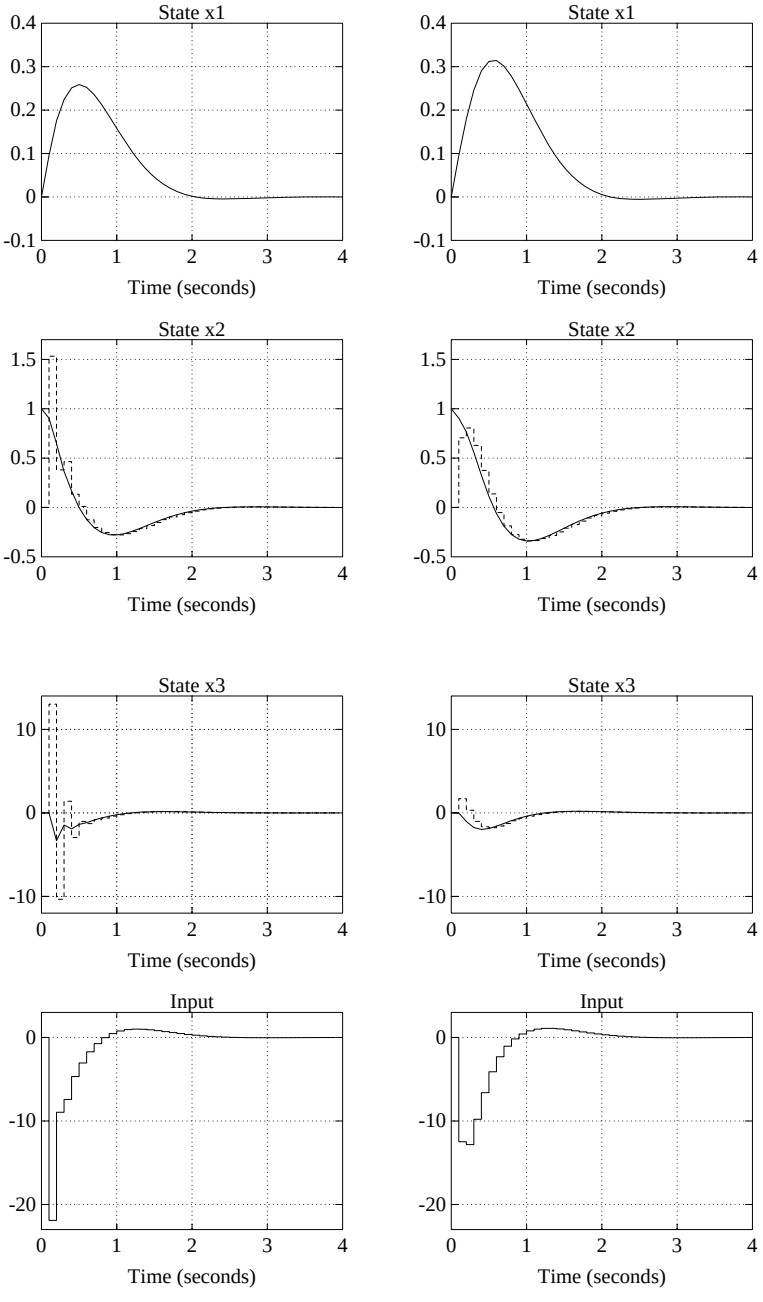


Figure 7.21 Example 7.8 – The graphs in the left-hand column are for the regulator with observer poles on the negative real axis. The solid lines are the plant state variables and the dashed lines are the estimates produced by the observer. The graphs in the right-hand column are for the regulator with observer poles corresponding to Bessel polynomial.

The stability margins for the system with Bessel observer poles are

$$\text{GM} = 12.6 \text{ dB}$$

$$\text{PM} = 57.6^\circ.$$

Thus the observer which cancels a zero of the plant does in fact yield better stability margins than an observer designed using Bessel polynomials. However, the difference in stability margins is not that great, and the transient response of the Bessel observer is quite good. The conclusion for this example is that it is better not to cancel the plant zeros on the negative real axis.

■

7.6 Reduced-Order Observers for Systems with an Arbitrary Output Matrix

In the previous sections we showed how to design reduced-order observers for the case in which the measured signals consisted of linear combinations of the first m state variables of the plant. For such systems the measurement equation is

$$\mathbf{y}_m[k] = \mathbf{C}_m\mathbf{x}[k] = [\,\mathbf{C}_1 \quad \mathbf{0}\,]\,\mathbf{x}[k] = \mathbf{C}_1\mathbf{x}_1[k]$$

where $\mathbf{C}_1$ is an $m \times m$ nonsingular matrix. In this section we show how to modify the design procedure developed previously to handle the general case where $\mathbf{C}_m$ is an arbitrary rank-m matrix. The derivation of the general case will proceed in two steps. In the first step we consider the case in which the measured signals consist of m state variables of the plant, but not the *first* m state variables. In the second step we show how to handle the general case.

7.6.1 Measure m State Variables

Suppose that the measured signals of the system consist of m state variables of the plant, but not the first m. To take a specific example, suppose that the system is fourth order and that the second and fourth state variables are measured; that is

$$\mathbf{y}_m[k] = \begin{bmatrix} x_2[k] \\ x_4[k] \end{bmatrix}.$$

Since $\mathbf{y}_m[k] = \mathbf{C}_m\mathbf{x}[k]$ the above equation implies that

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

The given system can be transformed into a new system whose measurements are the first two state variables. This transformation is accomplished by defining new state variables in terms of the original state variables. A useful definition of the new state variables for the example we are considering is

$$\begin{aligned} \bar{x}_1[k] &= x_2[k] \\ \bar{x}_2[k] &= x_4[k] \\ \bar{x}_3[k] &= x_1[k] \\ \bar{x}_4[k] &= x_3[k]. \end{aligned} \tag{7.52}$$

The important thing about this definition is that the measured variables ($x_2[k]$ and $x_4[k]$) appear as the first two state variables which is the case considered in previous sections. Note that the definition of new state variables in (7.52) is not unique. For example, $x_2[k]$ and $x_4[k]$ could be interchanged and $x_1[k]$ and $x_3[k]$ could be interchanged. The important thing about any of these definitions is simply that the measured variables appear only in the first m new state variables. For the purpose of this discussion, we will use the definition in (7.52).

Note that (7.52) can be written in matrix form as follows

$$\bar{\mathbf{x}}[k] = \mathbf{P}\mathbf{x}[k]$$

where

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{7.53}$$

In other words, the new state variables are just a linear transformation of the original state variables.

Let us now extend the discussion to consider an n^{th}-order system in which the measured signals consist of m state variables. We will also mention parallels between this case and the general case. The general case considered in the next subsection also uses a linear transformation to define a new set of state variables. It will be seen in the general case that there are two important properties that the linear transformation matrix $\mathbf{P}$ must possess:

P1 The first m rows of $\mathbf{P}$ equal the matrix $\mathbf{C}_m$.

P2 The matrix $\mathbf{P}$ must be nonsingular.

These two properties are satisfied by the matrix $\mathbf{P}$ in (7.53). In fact we can show that these properties will be true for *any* problem in which m of the state variables are measured. By definition, the first m state variables must equal the measured state variables, or

$$\begin{bmatrix} \bar{x}_1[k] \\ \vdots \\ \bar{x}_m[k] \end{bmatrix} = \mathbf{y}_m[k] = \mathbf{C}_m\mathbf{x}[k].$$

Since $\bar{\mathbf{x}}[k] = \mathbf{P}\mathbf{x}[k]$, the above equation shows that the first m rows of $\mathbf{P}$ must equal $\mathbf{C}_m$. If the state vector $\bar{\mathbf{x}}[k]$ is partitioned into its first m elements and last $n-m$ elements, the first m elements will equal $\mathbf{y}_m[k]$. The last $n-m$ elements will be denoted $\bar{\mathbf{x}}_2[k]$. Thus we have

$$\begin{bmatrix} \mathbf{y}_m[k] \\ \bar{\mathbf{x}}_2[k] \end{bmatrix} = \mathbf{P}\mathbf{x}. \tag{7.54}$$

We will return to this equation at the end of this section.

Since the new state variables are just a different ordering of the original state variables, the matrix $\mathbf{P}$ will be a *permutation matrix*. A permutation matrix consists of 0's and 1's with only a single 1 in each row and column. Each row of a permutation matrix has a single 1 as shown below

$$\text{row } i \; = [\,0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0\,].$$

The 1 appears in a different location in rows i and j (for $i \neq j$) since only a single 1 can appear in any column of $\mathbf{P}$. Thus for $i \neq j$

$$(\text{row } i)(\text{row } j)^T = 0.$$

This means that the rows of $\mathbf{P}$ are orthogonal to each other. From Chapter 2, we know that orthogonal vectors are linearly independent (Fact 2.25), and Property **P2** is satisfied.

We have just shown that when the measured signals consist of m state variables, a linear transformation matrix $\mathbf{P}$ can be found which satisfies properties **P1** and **P2** mentioned above. The matrix $\mathbf{P}$ can be used to transform the system $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C}_m)$ into $(\bar{\mathbf{\Phi}}, \bar{\mathbf{\Gamma}}, \bar{\mathbf{C}}_m)$ according to the formulas

$$\bar{\mathbf{\Phi}} = \mathbf{P}\mathbf{\Phi}\mathbf{P}^{-1}$$

$$\bar{\mathbf{\Gamma}} = \mathbf{P}\mathbf{\Gamma}$$

$$\bar{\mathbf{C}}_m = \mathbf{C}_m\mathbf{P}^{-1}.$$

The transformed system is *similar* to the original system and thus has the same input/output behavior as shown below

Original System	Transformed System
$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k]$	$\bar{\mathbf{x}}[k+1] = \bar{\mathbf{\Phi}}\bar{\mathbf{x}}[k] + \bar{\mathbf{\Gamma}}u[k]$
$\mathbf{y}_m[k] = \mathbf{C}_m\mathbf{x}[k]$	$\mathbf{y}_m[k] = \bar{\mathbf{C}}_m\bar{\mathbf{x}}[k].$

(7.55)

Recall that $\mathbf{P}$ was selected so that the first m state variables were the measured signals. This means that

$$\mathbf{y}_m[k] = \bar{\mathbf{C}}_m\bar{\mathbf{x}}[k] = \begin{bmatrix} \bar{x}_1[k] \\ \bar{x}_2[k] \\ \vdots \\ \bar{x}_m[k] \end{bmatrix}$$

which implies that

$$\bar{\mathbf{C}}_m = [\,\mathbf{I}_m \quad \mathbf{0}_{m,(n-m)}\,].$$

The form of $\bar{\mathbf{C}}_m$ is exactly that assumed in the development of the design equations for reduced-order observers in the previous section. Those design equations can be used to obtain a reduced-order observer for the transformed system. The resulting observer is shown in Fig. 7.22.

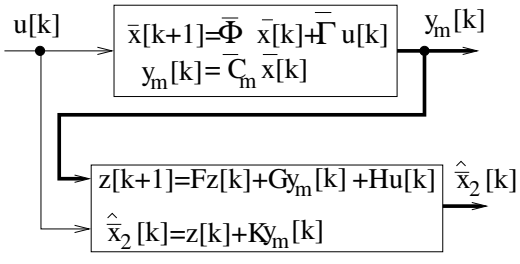


Figure 7.22 A reduced-order observer for the transformed system.

The observer in Fig. 7.22 takes $u[k]$ and $\mathbf{y}_m[k]$ as inputs. As we noted in (7.55), the original and transformed systems have the same input/output behavior. Thus the reduced-order observer which was designed for the *transformed* system can be used with the input

and measurements of the *original* system to produce the estimates $\hat{\bar{\mathbf{x}}}_2[k]$. The measurements $\mathbf{y}_m[k]$ and estimates $\hat{\bar{\mathbf{x}}}_2[k]$ can be transformed into estimates of the original state variables using the matrix $\mathbf{P}$ (see (7.54))

$$\hat{\mathbf{x}} = \mathbf{P}^{-1} \begin{bmatrix} \mathbf{y}_m[k] \\ \hat{\bar{\mathbf{x}}}_2[k] \end{bmatrix}.$$

The resulting reduced-order for the original system is shown in Fig. 7.23.

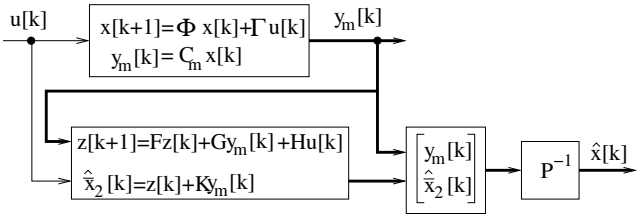


Figure 7.23 A reduced-order observer for the original system.

EXAMPLE 7.9

A state-space model for the cart/pendulum system introduced in Chapter 3 has the following state variables

$$x_1 = \text{pendulum position}$$

$$x_2 = \text{pendulum velocity}$$

$$x_3 = \text{motor position}$$

$$x_4 = \text{motor velocity}.$$

We have seen that this system can be regulated by state feedback with closed-loop poles at the roots of the 4th-order Bessel polynomial with $T_s = 1.2$. In this example, we assume that only x_1 and x_3 can be measured and we design a reduced-order observer to estimate x_2 and x_4.

The ZOH equivalent of the plant for sampling period $T = 0.005$ seconds is

$$\mathbf{\Phi} = \begin{bmatrix} 1.0003 & 0.0005 & 0 & 0 \\ 0.1154 & 1.0003 & 0 & -.0011 \\ 0 & 0 & 1 & .0044 \\ 0 & 0 & 0 & .7672 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} .0010 \\ .3875 \\ .2127 \\ 81.4779 \end{bmatrix}$$

and the measurement matrix is

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

We begin by defining a new set of state variables which permutes the given state variables such that the measured signals appear first

$$\begin{bmatrix} \bar{x}_1[k] \\ \bar{x}_2[k] \\ \bar{x}_3[k] \\ \bar{x}_4[k] \end{bmatrix} = \mathbf{P} \begin{bmatrix} x_1[k] \\ x_2[k] \\ x_3[k] \\ x_4[k] \end{bmatrix}$$

where

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

The transformed system is computed to be

$$\bar{\mathbf{\Phi}} = \begin{bmatrix} 1.0003 & 0 & .0050 & 0 \\ 0 & 1 & 0 & .0044 \\ .1154 & 0 & 1.0003 & -.0011 \\ 0 & 0 & 0 & .7672 \end{bmatrix}$$

$$\bar{\mathbf{\Gamma}} = [\, .0010 \quad .2127 \quad .3875 \quad 81.4779 \,]^T$$

$$\bar{\mathbf{C}}_m = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Note that $\bar{\mathbf{C}}_1 = \mathbf{I}$ as desired.

We now use Case 1 reduced-order observer procedure. Since the full-state feedback regulator was designed with $T_s = 1.2$, we choose the observer poles to have a settling time of $1.2/3.5 = 0.34$ seconds. In order to get decoupled error dynamics, we choose the observer poles to be real-valued each corresponding to a settling time of 0.34. In the s-plane a settling time of 0.34 corresponds to a pole location of $s = -4.6/0.34 = -13.5 \approx -13$ and mapping to the z-plane yields

$$z = e^{-13T} = .9371.$$

So we choose $f_1 = f_2 = .9371$ for the diagonal elements of the matrix $\mathbf{F}$ of the reduced-order observer. The design procedure for Case 1 then yields the following reduced-order observer for the transformed system (recall that $\bar{\mathbf{C}}_1 = \mathbf{I}$)

$$\mathbf{F} = \begin{bmatrix} .9371 & 0 \\ 0 & .9371 \end{bmatrix}$$

$$\mathbf{L} = \bar{\mathbf{\Phi}}_{12}^{-1}[\bar{\mathbf{\Phi}}_{22} - \mathbf{F}]$$

$$= \begin{bmatrix} 12.6365 & -.2437 \\ 0 & -38.6796 \end{bmatrix}$$

$$\mathbf{H} = \bar{\mathbf{\Gamma}}_2 - \mathbf{K}\bar{\mathbf{\Gamma}}_1$$

$$= \begin{bmatrix} .4266 \\ 89.7043 \end{bmatrix}$$

$$\mathbf{G} = [\bar{\mathbf{\Phi}}_{21} - \mathbf{K}\bar{\mathbf{\Phi}}_{11}] + \mathbf{FL}.$$

The equations which describe the reduced-order observer are

$$\mathbf{z}[k+1] = \mathbf{F}\mathbf{z}[k] + \mathbf{G}\mathbf{y}[k] + \mathbf{H}u[k]$$

$$\hat{\mathbf{x}}_2[k] = \mathbf{z}[k] + \mathbf{L}\mathbf{y}[k]$$

which is initialized with

$$\mathbf{z}[0] = -\mathbf{L}\mathbf{y}[0].$$

Consider the behavior of the closed-loop system when all state variables are initialized to 0 except pendulum position, which is initialized to 10°. Recall that the reduced-order observer is initialized so that its estimates at time 0 are all 0. In this case, the initial values of the unmeasured state variables (pendulum and motor velocity) are in fact zero, and so there is no erro between the true state variables and those estimated by the reduced-order observer. In the absence of any disturbances, the estimated state variables will be equal to the true state variables for all time, and the closed-loop regulator with the reduced-order observer will have identical performance to the regulator which uses full-state feedback.

Now consider the case when all state variables except pendulum velocity are initialized to zero. This case could occur in the following way. Suppose the loop was closed and the pendulum was balanced with the cart in the center of the track so that the values of all the state variables were zero. Then at time $t = 0$ someone tapped the pendulum. The tap would impart a nonzero velocity to the pendulum, but the values of all the other state variables would equal zero at $t = 0$. We have simulated this example with the initial value of pendulum velocity equal to 0.7 radians per second. Fig. 7.24(a), (b) show pendulum and cart position as a function of time – the solid curves are for the system with the observer, the dashed lines are for a system using full-state feedback. It can be seen that the performance of the system with the observer is slightly worse in terms of overshoot. The reason is that it takes the observer a short amount of time to correctly estimate the pendulum velocity. Fig. 7.24(c) shows the actual pendulum velocity (dashed line) and the estimated velocity (solid line). Notice that the estimated velocity starts at zero due to the initialization of the reduced-order observer. Also note that the observer output begins to "track" the true velocity at 0.3 seconds which is the settling time that the observer was designed for. In this example, the initial value of motor velocity was zero. Since the observer is initialized to make estimated motor velocity zero, there is no error at time zero. Fig. 7.24(d) shows the true and estimated motor velocities, and the two plots are identical. The plots are identical not only because the initial value was correct, but also because the $\mathbf{F}$ matrix of the observer was chosen to be diagonal which implies that the estimation errors are decoupled. Decoupled errors means that the error in estimated pendulum velocity does not cause an error in estimated motor velocity. If the reduced-order observer were designed using pole-placement techniques (Case 3) instead of using the Case 1 procedure, the $\mathbf{F}$ matrix would not have been diagonal, and there would have been some error in estimated motor velocity for $0 < t < 0.3$.

■

7.6.2 General Case

We now consider the general case in which the measurement matrix $\mathbf{C}_m$ has rank m but arbitrary structure. We begin by constructing a transformation matrix $\mathbf{P}$ which satisfies properties **P1** and **P2** given in the previous subsection; namely, that the first m rows of $\mathbf{P}$ equal $\mathbf{C}_m$ and that $\mathbf{P}$ is nonsingular. The matrix $\mathbf{P}$ will have the following form

$$\mathbf{P} = \begin{bmatrix} \mathbf{C}_m \\ \mathbf{P}_2 \end{bmatrix} \tag{7.56}$$

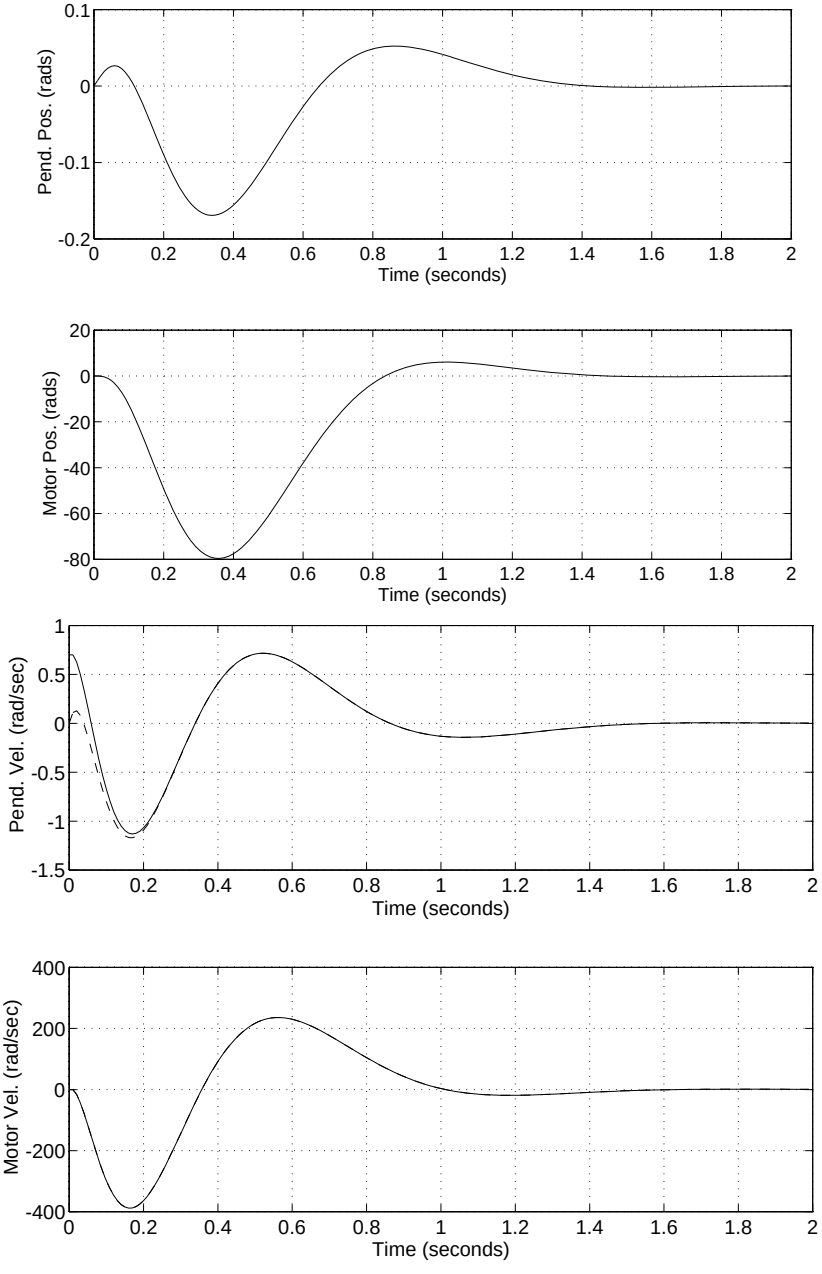


Figure 7.24 (a) Pendulum position for the observer-based regulator (solid line) and for a regulator using full-state feedback (dashed line); (b) Motor position for the observer-based regulator (solid line) and for a regulator using full-state feedback (dashed line); (c) Actual pendulum velocity (dashed line) and estimated velocity (solid line); (d) Actual and estimated motor velocities (plots are identical).

where $\mathbf{P}_2$ is an $(n-m) \times n$ matrix chosen to make $\mathbf{P}$ nonsingular. Before showing how to find such a matrix $\mathbf{P}_2$, we first show how the transformation matrix $\mathbf{P}$ is used. We begin by partitioning $\mathbf{P}^{-1}$ into the first m and remaining $n-m$ columns as follows

$$\mathbf{P}^{-1} = [\, \mathbf{Q}_1 \quad \mathbf{Q}_2 \,]. \tag{7.57}$$

We can then write the identity $\mathbf{P}\mathbf{P}^{-1} = \mathbf{I}$ in partitioned form using (7.56) and (7.57) as follows

$$\begin{bmatrix} \mathbf{C}_m \\ \mathbf{P}_2 \end{bmatrix} [\, \mathbf{Q}_1 \quad \mathbf{Q}_2 \,] = \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{m,n-m} \\ \mathbf{0}_{n-m,m} & \mathbf{I}_{n-m} \end{bmatrix}.$$

Multiplying out the left-hand side of this equation and equating the resulting partitions yields

$$\begin{aligned} \mathbf{C}_m\mathbf{Q}_1 &= \mathbf{I}_m \\ \mathbf{C}_m\mathbf{Q}_2 &= \mathbf{0}_{m,n-m} \\ \mathbf{P}_2\mathbf{Q}_1 &= \mathbf{0}_{n-m,m} \\ \mathbf{P}_2\mathbf{Q}_2 &= \mathbf{I}_{n-m}. \end{aligned} \tag{7.58}$$

The matrix $\mathbf{P}$ can be used to form a transformed system whose measurement matrix is

$$\bar{\mathbf{C}}_m = \mathbf{C}_m\mathbf{P}^{-1}$$

which can be simplified using (7.57) and (7.58) to yield

$$\begin{aligned} \bar{\mathbf{C}}_m &= \mathbf{C}_m\, [\, \mathbf{Q}_1 \quad \mathbf{Q}_2 \,] \\ &= [\, \mathbf{C}_m\mathbf{Q}_1 \quad \mathbf{C}_m\mathbf{Q}_2 \,] \\ &= [\, \mathbf{I}_m \quad \mathbf{0}_{m,n-m} \,]. \end{aligned}$$

That is, the measured signals consist of the first m transformed state variables and so a reduced-order observer can be designed for the transformed system using the design equations given in Section 7.4. An observer designed for the transformed system can be used with the original system because the original and transformed systems have the same input/output behavior.

The above development was based upon finding a matrix $\mathbf{P}_2$ such that

$$\mathbf{P} = \begin{bmatrix} \mathbf{C}_m \\ \mathbf{P}_2 \end{bmatrix}$$

is nonsingular. We now give a procedure to find such a matrix $\mathbf{P}_2$ using a singular value decomposition of $\mathbf{C}_m$. Let the SVD of $\mathbf{C}_m$ be (see Chapter 2)

$$\mathbf{C}_m = \mathbf{U}\, [\, \mathbf{\Sigma}_{m\times m} \quad \mathbf{0}_{m\times(n-m)} \,] \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix}$$

where $\mathbf{V}_1^T$ is $m \times n$ and $\mathbf{V}_2^T$ is $(n-m) \times n$. A property of the SVD is that the columns of $\mathbf{V}_2$ are an orthonormal basis for the null-space of $\mathbf{C}_m$. In other words,

$$\mathbf{C}_m\mathbf{V}_2 = \mathbf{0}\,, \quad \text{and } \mathbf{V}_2^T\mathbf{V}_2 = \mathbf{I}.$$

Given the ZOH of the plant $(\mathbf{\Phi}, \mathbf{\Gamma})$ and the measurement matrix $\mathbf{C}_m$

1. Calculate
$$\mathbf{P} = \begin{bmatrix} \mathbf{C}_m \\ \mathbf{P}_2 \end{bmatrix}.$$
If the measurements consist of m state variables, choose $\mathbf{P}_2$ to make $\mathbf{P}$ a permutation matrix. In general, take the SVD of $\mathbf{C}_m$
$$\mathbf{C}_m = \mathbf{U} \begin{bmatrix} \mathbf{\Sigma}_{m\times m} & \mathbf{0}_{m\times(n-m)} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix}$$
and let $\mathbf{P}_2 = \mathbf{V}_2^T$.
2. Compute the transformed system
$$\bar{\mathbf{\Phi}} = \mathbf{P}\mathbf{\Phi}\mathbf{P}^{-1}$$
$$\bar{\mathbf{\Gamma}} = \mathbf{P}\mathbf{\Gamma}$$
$$\bar{\mathbf{C}}_m = \mathbf{C}_m\mathbf{P}^{-1}.$$
3. Calculate a reduced-order observer $(\mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{L})$ using the procedure for Case 1, 2, or 3. The output of the reduced-order observer is $\hat{\mathbf{x}}_2[k]$.
4. Transform back to the original state variables
$$\hat{\mathbf{x}}[k] = \mathbf{P}^{-1} \begin{bmatrix} \mathbf{y}_m[k] \\ \hat{\mathbf{x}}_2[k] \end{bmatrix}.$$

Table 7.7 A general design procedure for reduced-order observers.

The first equation means that the rows of $\mathbf{V}_2^T$ are orthogonal to the rows of $\mathbf{C}_m$ and the second equation says that the rows of $\mathbf{V}_2$ are linearly independent. These two facts imply that the rows of

$$\mathbf{P} = \begin{bmatrix} \mathbf{C}_m \\ \mathbf{V}_2^T \end{bmatrix} \tag{7.59}$$

are linearly independent, and so $\mathbf{P}$ defined by (7.59) is nonsingular.

For the case in which m state variables are measured, $\mathbf{C}_m$ consists of 1's and 0's. In this case it is easy to choose $\mathbf{P}_2$ to make $\mathbf{P}$ nonsingular. A natural choice is to choose $\mathbf{P}_2$ to make $\mathbf{P}$ be a permutation matrix; the SVD is not needed in this case. However in the general case the matrix $\mathbf{P}_2$ cannot always be written down by inspection; it can be calculated using the SVD of $\mathbf{C}_m$. A general procedure to calculate reduced-order observer matrices is shown in Table 7.7.

EXAMPLE 7.10

Suppose the ZOH equivalent of a plant at sampling interval $T = 0.1$ is

$$\mathbf{\Phi} = \begin{bmatrix} 1.0000 & 0 & -0.2473 \\ 0 & 0.9048 & -0.1563 \\ 0 & 0 & 0.6703 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0868 \\ 0.0867 \\ 0.0824 \end{bmatrix}$$

and the measurement matrix is

$$\mathbf{C}_m = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}.$$

In order to design a reduced-order observer for this system, we must first find a transformation which causes the measurements to only depend on the first two state variables. Such a transformation $\mathbf{P}$ can be found from the SVD of $\mathbf{C}_m$

$$\begin{aligned} \mathbf{C}_m &= \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \\ &= \begin{bmatrix} -0.7071 & -0.7071 \\ 0.7071 & -0.7071 \end{bmatrix} \begin{bmatrix} 1.7321 & 0 & 0 \\ 0 & 1.0000 & 0 \end{bmatrix} \begin{bmatrix} -0.4082 & 0.8165 & -0.4082 \\ -0.7071 & 0.0000 & 0.7071 \\ 0.5774 & 0.5774 & 0.5774 \end{bmatrix}. \end{aligned}$$

The first two rows of $\mathbf{P}$ consist of the matrix $\mathbf{C}_m$, and the last row of $\mathbf{P}$ can be chosen to be the last row of the $\mathbf{V}^T$ matrix. In this example, another choice for the last row of $\mathbf{P}$ is $[\,0 \quad 0 \quad 1\,]$, because this choice also results in a nonsingular matrix $\mathbf{P}$. However, it is not always possible to pick the rows of $\mathbf{P}$ by inspection. Here we continue with the choice suggested in Table 7.7

$$\mathbf{P} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0.5774 & 0.5774 & 0.5774 \end{bmatrix}.$$

The transformed system is then computed to be

$$\begin{aligned} \boldsymbol{\Phi} = \mathbf{P}\boldsymbol{\Phi}\mathbf{P}^{-1} &= \begin{bmatrix} 1.0020 & 0.0937 & 0.0048 \\ 0.0364 & 0.8782 & 0.0892 \\ -0.1245 & 0.0239 & 0.6950 \end{bmatrix} \\ \boldsymbol{\Gamma} = \mathbf{P}\boldsymbol{\Gamma} &= \begin{bmatrix} 0.0001 \\ 0.0042 \\ 0.0734 \end{bmatrix} \\ \bar{\mathbf{C}}_m = \mathbf{C}_m\mathbf{P}^{-1} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \end{aligned}$$

Notice that $\bar{\mathbf{C}}_m$ is of the form assumed in the development of the reduced-order observer design equations.

Since there are two measurements of a third-order system, we can use the Case 1 design procedure. To get an observer settling time of 0.5 seconds, for example, we can

choose $\mathbf{F} = 0.3969$ (see Example 7.5). The remaining matrices are calculated to be

$$\begin{aligned}
\mathbf{L} &= (\mathbf{\Phi}_{22} - \mathbf{F})(\mathbf{\Phi}_{12}^T \mathbf{\Phi}_{12})^{-1} \mathbf{\Phi}_{12}^T \\
&= (0.6950 - 0.3969) \left(\begin{bmatrix} .0048 & .0892 \end{bmatrix} \begin{bmatrix} .0048 \\ .0892 \end{bmatrix} \right)^{-1} \begin{bmatrix} .0048 & .0892 \end{bmatrix} \\
&= \begin{bmatrix} 0.1810 & 3.3315 \end{bmatrix} \\
\mathbf{H} &= \mathbf{\Gamma}_2 - \mathbf{L}\mathbf{\Gamma}_1 \\
&= 0.0734 - \begin{bmatrix} 0.1810 & 3.3315 \end{bmatrix} \begin{bmatrix} .0001 \\ .0042 \end{bmatrix} \\
&= 0.0592 \\
\mathbf{G} &= \mathbf{\Phi}_{21} - \mathbf{L}\mathbf{\Phi}_{11} + \mathbf{F}\mathbf{L} \\
&= \begin{bmatrix} -0.1245 & 0.0239 \end{bmatrix} - \begin{bmatrix} 0.1810 & 3.3315 \end{bmatrix} \begin{bmatrix} 1.0020 & 0.0937 \\ 0.0364 & 0.8782 \end{bmatrix} + 0.3969 \begin{bmatrix} 0.1810 & 3.3315 \end{bmatrix} \\
&= \begin{bmatrix} -0.3554 & -1.5965 \end{bmatrix}.
\end{aligned}$$

If the output of the reduced-order observer is called $\hat{\mathbf{x}}_2[k]$, then estimates of the state variables of the original plant are

$$\hat{\mathbf{x}}[k] = \mathbf{P}^{-1} \begin{bmatrix} \mathbf{y}_m[k] \\ \hat{\mathbf{x}}_2[k] \end{bmatrix}.$$

■

7.7 Summary

In this chapter we showed how to design observers which estimate some or all of the plant state variables from input/output measurements. The design of a full-order observer is accomplished by calculating a vector of observer gains $\mathbf{L}$. This calculation is similar to the calculation of a state-feedback vector. The design of reduced-order observers is more complicated; design procedures were developed for three different cases which depend on the relationship between the number of measurements and the number of state variables.

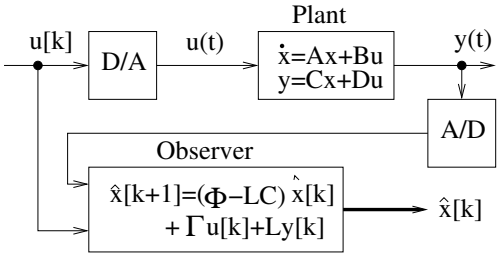
We showed how observers are used in regulation systems. The $2n$ closed-loop poles of an observer-based regulator can be divided into two groups: n poles are those that would be obtained with full-state feedback, and the other n poles are the observer poles. We usually choose the observer poles to be 3 to 5 times "faster" than the full-state feedback poles. However, if the plant has "significant" zeros, then we choose observer poles to cancel these zeros in order to preserve the stability margins of the observer-based regulator.

7.8 Problems

1. This chapter dealt with the design of observers for systems whose outputs were just a matrix times the state vector, i.e. for systems without a feedthrough term. Many

systems of interest are of this form, but for the sake of completeness we now examine what happens for plants which do have a feedthrough term. You will find that only a small change in the observer is needed.

Consider the following block diagram, which shows a full-order observer of the type discussed in this chapter connected to a plant with a feedthrough term



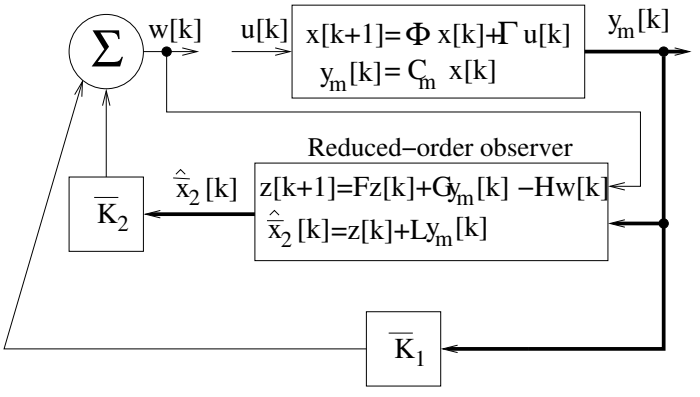
(a) Derive a difference equation for the error vector $\mathbf{e}[k] = \mathbf{x}[k] - \hat{\mathbf{x}}[k]$ for the above system. Show that the error equation is

$$\mathbf{e}[k+1] = (\mathbf{\Phi} - \mathbf{LC})\mathbf{e}[k] - \mathbf{L}du[k]$$

which is *not* a homogeneous equation. Thus the error will not go to zero in general unless the input is always zero.

(b) Let the $\mathbf{\Gamma}$ matrix which appears in the observer equations be replaced by a matrix $\bar{\mathbf{\Gamma}}$. Show that by choosing $\bar{\mathbf{\Gamma}}$ properly, the observer shown above will have the homogeneous error equation $\mathbf{e}[k+1] = (\mathbf{\Phi} - \mathbf{LC})\mathbf{e}[k]$. With this modification, the error will go to zero, regardless of the values of the inputs. (Note that the $\mathbf{\Gamma}$ matrix in the plant remains unchanged.)

2. In order to find the stability margins for a regulator which uses a reduced-order observer, the "loop transfer function" must be calculated. The following figure shows an observer-based regulator with the loop broken at the input to the plant.



Find a state-space description for the composite system with input $u[k]$ and output $w[k]$. Use the composite state vector

$$\begin{bmatrix} \mathbf{x}[k] \\ \mathbf{z}[k] \end{bmatrix}.$$

A state-space model of the loop transfer function for a regulator which uses a full-order observer is given in (7.16) on page 306. In this problem, you derive the corresponding result for a reduced-order observer. Recall that

$$\begin{bmatrix} \mathbf{y}_m \\ \bar{\mathbf{x}}_2 \end{bmatrix} = \mathbf{P}\mathbf{x} \text{ and } [\,\bar{\mathbf{K}}_1 \quad \bar{\mathbf{K}}_2\,] = \mathbf{K}\mathbf{P}^{-1}.$$

3. Find the stability margins for the observer-based regulator designed in Example 7.3. Then design another observer which has an eigenvalue equal to the plant zero located inside the unit circle. Find the stability margins and plot the transient response of the regulator which uses your observer together with the $\mathbf{K}$ vector given in Example 7.3.

4. A hydraulic positioning system was introduced in problem 9 on page 285. A full state feedback regulator was designed for that problem. The purpose of this problem is to use observers to estimate some of the state variables which are not measured.

   (a) Consider the case in which only the output position is measured, and a reduced-order observer is use to estimate the other three state variables. Choose the observer poles to be the roots of a 3rd-order Bessel polynomial with $T_S = 0.1$. Calculate the loop transfer function, and get a Nyquist plot of this observer-based regulator. Find its gain and phase margins.

   (b) Repeat the previous item but this time choose some of the observer poles to cancel stable zeros of the plant. Choose the remaining poles using a lower order Bessel polynomial. Find the stability margins of this observer-based regulator, and get step response plots.

   (c) Now consider the case in which x_1, x_2, and x_4 can all be measured. The measurement matrix is now

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

   The reduced-order observer only needs to estimate the value of x_3. Choose the observer pole as the root of a 1st-order Bessel polynomial. Find gain and phase margins, and get step response plots.

5. Problem 10 on page 286 considered the design of a full state feedback regulator to control the roll angle of a ship. Repeat that problem assuming that only x_1 and x_3 are measured.

6. Problem 12 on page 288 considered regulation of an antenna system using full state feedback. Repeat that problem assuming only x_1 can be measured.

7. Repeat the simulations in Example 6.10 on page 265 assuming only x_1 and x_3 can be measured.

8. Consider the following ZOH equivalent which corresponds to a sampling interval of $T = 0.1$ seconds.

$$\mathbf{\Phi} = \begin{bmatrix} 0.7242 & -0.3523 & -0.1716 & 0 \\ 0.0858 & 0.9816 & -0.0090 & 0 \\ 0.0045 & 0.0994 & 0.9997 & 0 \\ 0.0002 & 0.0050 & 0.1000 & 1.0000 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0858 \\ 0.0045 \\ 0.0002 \\ 0.0000 \end{bmatrix}.$$

The observers for this problem should have a settling time of 0.5 seconds.

(a) Suppose that the measurement matrix is

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

Calculate the reduced-order observer matrices with a diagonal $\mathbf{F}$ matrix using the Case 1 design procedure. Notice that the system will have to be transformed using a permutation matrix before the observer matrices can be calculated.

(b) Repeat this problem for the measurement matrix shown below.

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

9. Recall the Type-1 servo system model introduced in Chapter 3. The state-space matrices are

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

Suppose that this system will operate under digital control with a sampling interval of $T = 0.1$ seconds. We assume that the first state variable is measured, so that the measurement matrix $\mathbf{C}_m = \mathbf{C}$. The desired settling time for the observer is 0.4 seconds.

(a) Calculate the reduced-order observer matrices using the Case 1 procedure.

(b) Calculate the reduced-order observer matrices using the Case 2 procedure.

10. Consider the following ZOH equivalent which corresponds to a sampling interval of $T = 0.1$ seconds.

$$\mathbf{\Phi} = \begin{bmatrix} 0.9997 & 0.0994 & 0.0045 \\ -0.0090 & 0.9816 & 0.0858 \\ -0.1716 & -0.3523 & 0.7242 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0002 \\ 0.0045 \\ 0.0858 \end{bmatrix}.$$

The observers for this problem should have a settling time of 0.3 seconds.

(a) Suppose that the measurement matrix is

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

Calculate reduced-order observer matrices using the Case 2 design procedure.

(b) Repeat this problem for the following measurement matrix.

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

11. Consider the following ZOH equivalent which corresponds to a sampling interval of $T = 0.05$ seconds.

$$\mathbf{\Phi} = \begin{bmatrix} 1.0000 & 0.0500 & 0.0012 & 0.0000 & 0.0000 \\ 0 & 1.0000 & 0.0500 & 0.0012 & 0.0000 \\ 0 & -0.0003 & 0.9995 & 0.0496 & 0.0011 \\ 0 & -0.0181 & -0.0274 & 0.9792 & 0.0429 \\ 0 & -0.6859 & -1.0469 & -0.7991 & 0.7220 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0011 \\ 0.0429 \end{bmatrix}.$$

Suppose that the measurement matrix is

$$\mathbf{C}_m = \begin{bmatrix} 1 & 2 & 0 & 0 & 0 \\ 0 & 1 & -.5 & 0 & 0 \end{bmatrix}.$$

Design a reduced-order observer which achieves a settling time of 0.2 seconds.

12. The ZOH equivalent at sampling interval $T = 0.01$ for a cart/pendulum system is

$$\mathbf{\Phi} = \begin{bmatrix} 1.0012 & 0 & 0.0100 & -0.0000 \\ 0 & 1.0000 & 0 & 0.0088 \\ 0.2311 & 0 & 1.0012 & -0.0011 \\ 0 & 0 & 0 & 0.7788 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0006 \\ 0.1213 \\ 0.1108 \\ 23.2923 \end{bmatrix}.$$

The state variables for this model are as follows

$$x_1 = \text{pendulum position}$$

$$x_2 = \text{motor position}$$

$$x_3 = \text{pendulum velocity}$$

$$x_4 = \text{motor velocity}.$$

The measured signals are x_1 and x_2. Because of sensor noise considerations, we want to use an observer to estimate x_3 and x_4. The purpose of this problem is to design a reduced-order observer to estimate x_3 and x_4.

Suppose that we have already calculated a state feedback vector $\mathbf{K}$ to give the closed-loop system a settling time of 1 second. If we want the observer to be "3 times faster," then the settling time of the observer should be $T_S = 1/3$ second. Choose $\mathbf{F}$ to be a diagonal matrix (Case 1) which corresponds to the desired settling time. Note that the two diagonal elements of $\mathbf{F}$ can be selected to be the same number which can be obtained from a 1st-order Bessel polynomial. Calculate the reduced-order observer matrices corresponding to the diagonal $\mathbf{F}$ matrix chosen above.

13. Consider a single-input, single-output system described by the following equations:

$$\mathbf{x}[k+1] = \begin{bmatrix} 0.5921 & 0.0782 & 0.0042 \\ -0.3127 & 0.9830 & 0.0994 \\ 0 & 0 & 1.0000 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} 0.0782 \\ -0.0170 \\ 0 \end{bmatrix} u[k]$$

$$y[k] = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x}[k].$$

The first state variable is measured directly. We can collect the state variables x_2 and x_3 into a vector

$$\mathbf{x}_2 = \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}.$$

Design a reduced-order observer for $\mathbf{x}_2[k]$. Calculate an observer gains vector $\mathbf{L}$ so that both poles of the observer are located at $z = 0.9048$. Show all calculations. (*Hint:* This is a Case 2 reduced-order observer.)

CHAPTER 8

TRACKING SYSTEMS

We have seen in previous chapters that state feedback is a simple but powerful technique to solve the regulation problem. In particular, *all* of the state variables of a controllable n-th order system can be driven to zero using state feedback. If not all the state variables are measured, then observers can be used to estimate the variables which are needed for state feedback. The solution to the regulation problem means that the state variables of the system will remain close to their equilibrium value (zero) in spite of disturbances acting on the system. In addition, unstable systems can be stabilized using state feedback.

As important as regulation is, however, it is not the only control problem of interest. In this chapter, we introduce another important class of problems called *tracking problems*. In the simplest tracking problem, a *reference input* is defined and the output of the system is required to be equal (or close to) the value of the reference input. In other words, the output of the system should *track* the reference input. The first three sections of this chapter show how tracking and regulation problems are related. The design of digital tracking systems is presented in Section 8.4.

8.1 Tracking Zero-Input Trajectories using State Feedback

Let us begin by considering the simplest tracking problem in which the reference input is equal to zero. In this case, the tracking problem reduces to a regulation problem, because a regulator will drive the state variables (and hence any output) to zero. What is so special

about the zero state that enables a regulator to "track" it? The answer is that the zero state vector is an *equilibrium state for every linear system.*

An equilibrium state is a constant state vector which satisfies the state-update equation $\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k]$ of a discrete-time linear system when the input is equal to zero. In other words, and equilibrium state $\mathbf{x}_e$ satisfies

$$\mathbf{x}_e = \mathbf{\Phi}\mathbf{x}_e. \tag{8.1}$$

The above equation is always satisfied by $\mathbf{x}_e = \mathbf{0}$, and so $\mathbf{0}$ is an equilibrium state for any linear system. We will show shortly that a regulator can easily be modified to drive a system to any of its equilibrium states, not just the zero state. So we ask the question, when does a system have other equilibrium states besides $\mathbf{0}$? The answer is found by looking at (8.1) and recalling the definition of eigenvalues and eigenvectors. We see that (8.1) is satisfied by a *nonzero* vector if and only if $\mathbf{x}_e$ is an eigenvector of $\mathbf{\Phi}$ corresponding to the eigenvalue $z = 1$. If the matrix $\mathbf{\Phi}$ was obtained as the zero-order-hold equivalent of a given continuous-time plant, then the eigenvalue $z = 1$ comes from a pole at $s = 0$ of the continuous-time plant. (By the ZOH pole mapping formula, $z = e^{sT} = e^0 = 1$.) If an analog plant has a pole at $s = 0$ it is called a Type-1 system. We have just seen that the ZOH equivalent of a Type-1 system will have an eigenvalue at $z = 1$, and the corresponding eigenvector will be a nonzero equilibrium state. A nonzero equilibrium point is also called a *setpoint*. We have already seen that a regulator can be used to drive a system to its zero equilibrium state. We now show how a simple modification enables a regulator to drive a system to a nonzero equilibrium state, or setpoint.

8.1.1 Step Inputs for Type-1 Systems

We first consider Type-1 systems, and later show how the same results can be obtained even if the system is not Type-1. As a specific example, consider the Type-1 servo system introduced in Chapter 3. From Example 4.1, we know that the system matrix for the ZOH equivalent takes the following form

$$\mathbf{\Phi} = \begin{bmatrix} 1 & \frac{1-e^{-pT}}{p} \\ 0 & e^{-pT} \end{bmatrix}. \tag{8.2}$$

Clearly the vector

$$\mathbf{x}_e = \begin{bmatrix} 1 \\ 0 \end{bmatrix} r$$

satisfies

$$\mathbf{\Phi}\mathbf{x}_e = \mathbf{x}_e$$

for any value of the scalar r. Note that this equilibrium point does not depend on the value of the pole location p of the plant, but only on the fact that the plant also contains an integrator (a pole at $s = 0$).

Recall that the state variables for the Type-1 servo system are defined as follows: x_1 = position, and x_2 = velocity. So the equilibrium points (setpoints) are state vectors corresponding to a fixed position and zero velocity. Suppose that the system is at position zero with zero velocity, and we desire the system to move to a different setpoint corresponding to $x_1 = r$ and $x_2 = 0$. This is equivalent to a *step input of height r in position*, where r is a scalar reference signal.

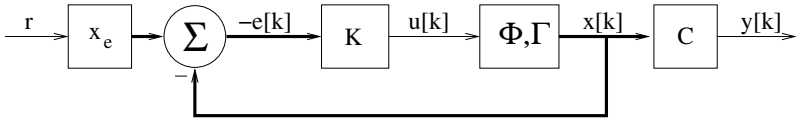


Figure 8.1 A control system for tracking step inputs with state feedback and a Type-1 plant.

We now show that the feedback gain vector $\mathbf{K}$ that is designed for a regulator (which will drive the state vector to zero) can also be used to drive the state vector to a desired setpoint. The control system is shown in Fig. 8.1. Notice that Fig. 8.1 reduces to a regulator if $r = 0$. To analyze the performance of this system , we first define an *error vector*

$$\mathbf{e}[k] = \mathbf{x}[k] - \mathbf{x}_e r,$$

and then write the following equations using Fig. 8.1.

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k] \\ u[k] &= \mathbf{K}(\mathbf{x}_e r - x[k]). \end{aligned}$$

If we substitute the second equation into the first, and subtract $\mathbf{x}_e r$ from both sides we obtain

$$\mathbf{x}[k+1] - \mathbf{x}_e r = (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})\mathbf{x}[k] + \mathbf{\Gamma}\mathbf{K}\mathbf{x}_e r - \mathbf{x}_e r. \tag{8.3}$$

This equation can be simplified using the definition of the error vector as follows, where we have added and subtracted $\mathbf{\Phi}\mathbf{x}_e r$ on the first line:

$$\begin{aligned} \mathbf{e}[k+1] &= (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})\mathbf{x}[k] + \mathbf{\Gamma}\mathbf{K}\mathbf{x}_e r - \mathbf{x}_e r + \mathbf{\Phi}\mathbf{x}_e r - \mathbf{\Phi}\mathbf{x}_e r \\ &= (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})\mathbf{x}[k] - (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})\mathbf{x}_e r + (\mathbf{\Phi} - I)\mathbf{x}_e r \\ &= (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})\mathbf{e}[k]. \end{aligned} \tag{8.4}$$

In the above derivation, notice that $(\mathbf{\Phi} - \mathbf{I})\mathbf{x}_e r = 0$ by definition of equilibrium point. Equation (8.4) shows that the system in Fig. 8.1 has the same homogeneous error equation that a regulator has, even when the input r is not equal to zero. This means that the error vector $\mathbf{e}[k]$ will go to zero at a rate determined by the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}$. These eigenvalues can be placed at desired values by proper choice of $\mathbf{K}$ as shown in the solution to the regulator problem. If the error vector goes to zero, this means that the state vector $\mathbf{x}[k]$ goes to the desired setpoint $\mathbf{x}_e r$, which means that tracking has been achieved with zero steady-state error.

From Fig. 8.1 we see that when $\mathbf{e}[k]$ goes to zero, the input $u[k]$ to the plant goes to zero. However, the state vector $\mathbf{x}[k]$ does not go to zero, but rather to $\mathbf{x}_e r$ as already noted. This is possible because $\mathbf{x}_e r$ is a (zero input) equilibrium state of the plant. We thus have made the following observation, which turns out to be true in general: *to get zero steady-state error tracking of a reference input state vector, that vector must be an equilibrium state of the plant.* Then we have seen that the regulator structure can be augmented as shown in Fig. 8.1 to obtain a tracking system. We will generalize this result in two ways. First, we will consider reference inputs that are not constant. Second, we will show what to do if the reference input is not an equilibrium state of the plant. In this case, we will show how to design a dynamic compensator (in state space form) with the property that the combined compensator/plant has the required equilibrium state. A state feedback regulator for the combined compensator/plant system can then be designed to obtain a tracking system with

zero steady-state error. Before moving ahead with the generalizations, we first examine the system for step inputs shown in Fig. 8.1 in a little more detail.

Notice that Fig. 8.1 is a single-input, single-output control system. The loop is closed using state feedback. From the development above, we know two things about this system: i) it is stable – the poles are specified by the designer when calculating the feedback gains $\mathbf{K}$, and ii) the system exhibits zero steady-state error to a step input. We now proceed to calculate the poles and zeros of the system. The state space description can be obtained from Fig. 8.1 as

$$\begin{aligned} \mathbf{x}[k+1] &= (\mathbf{\Phi} - \mathbf{\Gamma K})\mathbf{x}[k] + \mathbf{\Gamma K}\mathbf{x}_e r \\ y[k] &= \mathbf{C}\mathbf{x}[k]. \end{aligned} \tag{8.5}$$

The poles of this system are the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma K}$, and these poles are chosen when calculating the vector $\mathbf{K}$. The zeros of the system are values of z which make the following determinant equal to zero (see Problem 28 in Chapter 3)

$$\begin{vmatrix} zI - (\mathbf{\Phi} - \mathbf{\Gamma K}) & -\mathbf{\Gamma K}\mathbf{x}_e \\ \mathbf{C} & 0 \end{vmatrix}. \tag{8.6}$$

The inner product of $\mathbf{K}$ and $\mathbf{x}_e$ is some number, call it α. That is, $\alpha \stackrel{\text{def}}{=} \mathbf{K}\mathbf{x}_e$. Let the i-th element of the vector $\mathbf{K}$ be denoted K_i, and perform elementary column operations on the matrix shown in (8.6). The operations are as follows: replace the i-th column of the matrix by the sum of the i-th column and αK_i times the last column for $i = 1, \cdots, n$. These operations leave the determinant unchanged. Then divide the last column by $-\alpha$. This yields

$$\frac{-1}{\alpha} \begin{vmatrix} zI - \mathbf{\Phi} & -\mathbf{\Gamma} \\ \mathbf{C} & 0 \end{vmatrix}.$$

The above determinant is equal to zero at the zeros of the plant $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C})$. Thus the zeros of the system shown in Fig. 8.1 are identical to the zeros of the plant! We thus have the following observation: **state feedback can move the poles of the plant to arbitrary desired locations, but the location of the zeros is unchanged by state feedback.**

In the development of the tracking system shown in Fig. 8.1, we assumed that the scalar input r was a constant. However, suppose the input is not a constant, but is some sequence $r[k]$. The tracking system will produce an output $y[k]$ which is close to $r[k]$ (if $r[k]$ does not vary too quickly) and if $r[k]$ becomes constant, the output will converge to that constant with zero steady-state error. In fact, the bandwidth of the tracking system can be computed easily, because we know the poles and zeros computed above. After giving an example, we will generalize the problem by considering zero steady-state-error tracking systems whose inputs are not constant.

EXAMPLE 8.1

In this example we consider the step response of a Type-1 servo system having $p = 1$. Let the sampling period be $T = 0.1$ sec. From Example 4.1 on page 169, we can compute the ZOH of the plant to be

$$\mathbf{\Phi} = \begin{bmatrix} 1 & .0952 \\ 0 & .9048 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} .0048 \\ .0952 \end{bmatrix}.$$

The equilibrium state for this problem is

$$\mathbf{x}_e = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Suppose we want a settling time of $T_s = 2$ sec. The roots of a normalized second-order Bessel polynomial scaled by $1/T_s$ are

$$s_{1,2} = -2.0265 \pm j1.1700$$

and we map these into the z-plane using the ZOH pole-mapping formula

$$\begin{aligned} z_{1,2} &= e^{s_{1,2}T} \\ &= e^{(-2.0625 \pm j1.170)} \\ &= 0.8110 \pm j0.0953. \end{aligned}$$

Using the procedure described in Chapter 6, the vector of feedback gains which gives the desired closed-loop poles is

$$\mathbf{K} = \begin{bmatrix} 4.7092 & 2.7332 \end{bmatrix}.$$

The vectors $\mathbf{x}_e$ and $\mathbf{K}$ calculated in this example are used in the system shown in Fig. 8.2. We let $r = 1$ to get a unit step response. The input $u(t)$ and the output

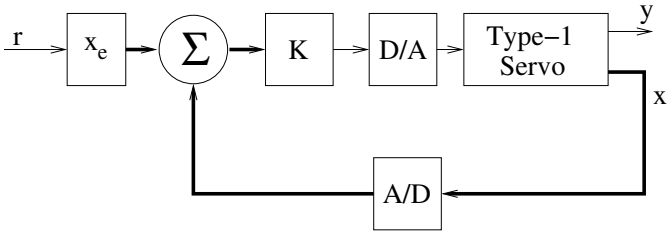


Figure 8.2 A tracking system for a type-1 servo system. A step input will be tracked with zero steady-state error.

$y(t)$ of the plant are shown in Fig. 8.3. The settling time is 2 seconds as desired. Other settling times could be achieved by appropriately scaling the Bessel roots and calculating the corresponding $\mathbf{K}$ vector.

To take this example one step further, we might ask how the system in Fig. 8.2 would behave for inputs other than a step input. We can answer this question by considering the equations of the closed-loop system. Using the numerical values from this example in (8.5), we get the following

$$\mathbf{x}[k+1] = \begin{bmatrix} .9772 & .0819 \\ -.4481 & .6447 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} .0228 \\ .4481 \end{bmatrix} r[k]$$

$$y[k] = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}[k].$$

The transfer function of the closed-loop system evaluated on the unit circle is

$$H(e^{j\omega T}) = \mathbf{c}(e^{j\omega T}\mathbf{I} - \mathbf{\Phi}_{CL})^{-1}\mathbf{\Gamma}_{CL}.$$

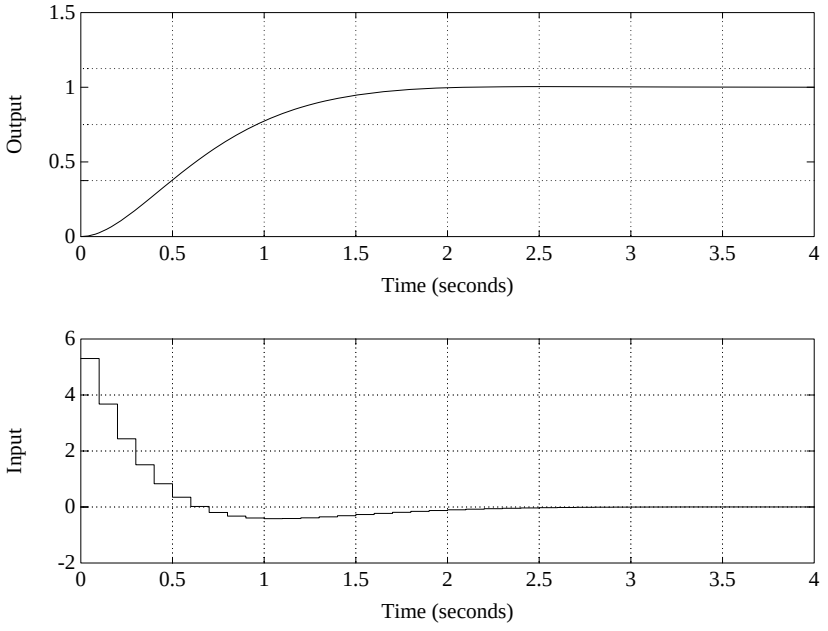


Figure 8.3 Step response of the system shown in Fig. 8.2.

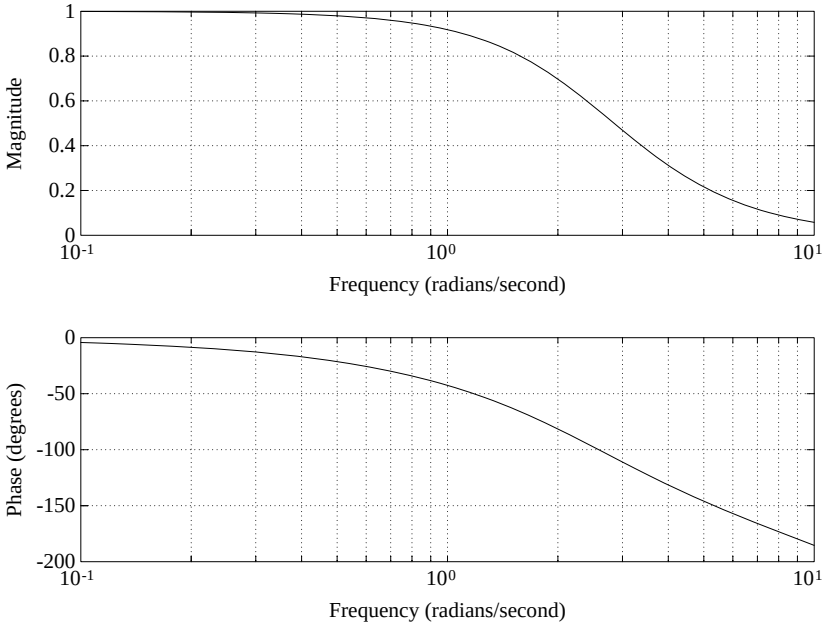


Figure 8.4 The Bode plot of the closed-loop system for example 8.1.

A plot of the magnitude and phase of $H(e^{j\omega T})$ (i.e. a Bode plot for the closed-loop system) is shown in Fig. 8.4.

If we consider a sinusoidal input at frequency $\omega_0 = 1$ rad/sec then from the Bode plot in Fig. 8.4 we would expect the output to be a 1 rad/sec sinusoid whose amplitude is about 0.9 and with a phase shift of about $-45^\circ = -.7884$ rad. For a sinusoid of frequency ω_0 rad/sec this phase shift is equivalent to a time-delay of

$$\begin{aligned} d &= \frac{\phi}{\omega_0} \\ &= \frac{-.7884}{1} \\ &= -.7884 \text{ sec.} \end{aligned}$$

Fig. 8.5 shows the plant input and output when the reference input is a 1 rad/sec sinusoid with unit amplitude. It can be seen that the output sinusoid has amplitude about 0.9 and is delayed by about 0.8 sec with respect to the input sinusoid.

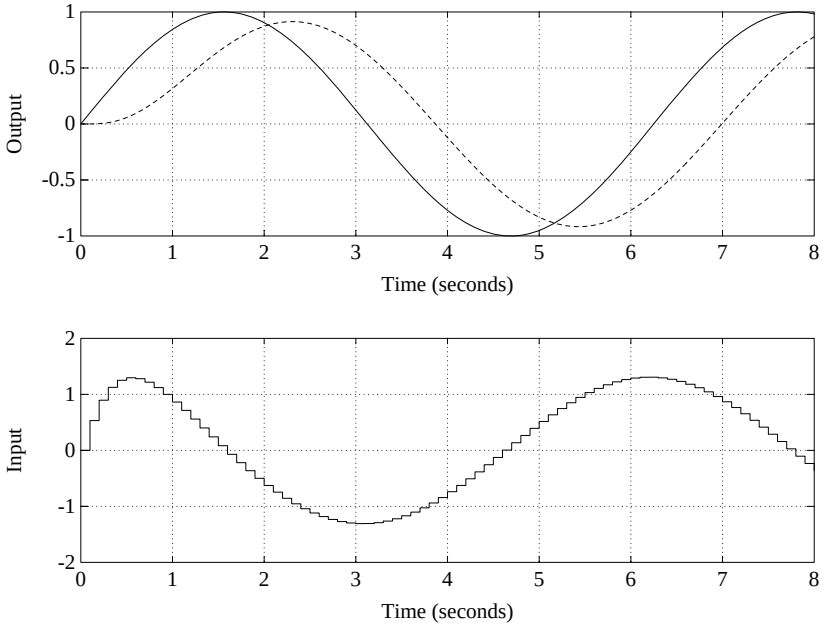


Figure 8.5 Response of the system in Fig. 8.2 to a 1 rad/sec sinusoid.

■

8.1.2 Ramp Inputs for Type-2 Systems

When the state vector of the plant is exactly tracking the reference state vector, the error vector $\mathbf{e}[k]$ is zero which means that the input to the plant is zero. This means that the state trajectory that is being tracked is a zero-input trajectory of the plant. We have seen that a plant with a single integrator has constant zero-input trajectories, and so exact tracking (zero steady-state error) for a step input is possible when the plant is a Type-1 system.

In this section, we show that zero input trajectories of a plant do not have to be constant. Regardless of the form of the zero-input trajectories of a plant, the regulator structure can be modified to track them with zero steady-state error. This will be demonstrated by considering ramp inputs for Type-2 systems. Later we will show how to design tracking systems for inputs which are *not* zero-input trajectories of the plant.

Let us begin with a specific example in which the plant is a double integrator with the following state-space description

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{8.7}$$

Recall from the development in Chapter 3 that the first state variable represents position and the second, velocity. From Example 4.3, we know that the ZOH equivalent for this plant is

$$\mathbf{\Phi} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}. \tag{8.8}$$

Recall that zero-input trajectories $\mathbf{x}_z[k]$ satisfy the following equation

$$\mathbf{x}_z[k+1] = \mathbf{\Phi}\mathbf{x}_z[k] \tag{8.9}$$

for some initial condition vector $\mathbf{x}_z(0) = \mathbf{x}_0$. We now show that it is possible to generate step and ramp signals as zero-input trajectories of this double integrator plant. Consider the following initial condition vector

$$\mathbf{x}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} r. \tag{8.10}$$

Substituting this vector into (8.9) and iterating (8.9), it is easy to verify that the following zero-input trajectory is produced for $k = 0, 1, 2, 3, \cdots$

$$\begin{bmatrix} r \\ 0 \end{bmatrix}, \quad \begin{bmatrix} r \\ 0 \end{bmatrix}, \quad \begin{bmatrix} r \\ 0 \end{bmatrix}, \quad \begin{bmatrix} r \\ 0 \end{bmatrix}, \cdots. \tag{8.11}$$

This zero-input trajectory corresponds to a step in position, with zero velocity.

Now consider the following initial condition vector

$$\mathbf{x}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} v. \tag{8.12}$$

Using (8.9), it is easy to verify that the following zero-input trajectory is produced for $k = 0, 1, 2, 3, \cdots$

$$\begin{bmatrix} 0 \\ v \end{bmatrix}, \quad \begin{bmatrix} Tv \\ v \end{bmatrix}, \quad \begin{bmatrix} 2Tv \\ v \end{bmatrix}, \quad \begin{bmatrix} 3Tv \\ v \end{bmatrix}, \cdots. \tag{8.13}$$

This zero-input trajectory corresponds to a constant velocity v, and a ramp in position. We now want to show that the regulator structure can be modified to track zero-input trajectories that are not constants, e.g. ramp inputs. Rather than show this for the specific case of a ramp input to a double integrator, we will show it in general, since the proof of the general result is no more difficult than the proof of the specific result. After proving the general result, we will return to the double integrator with a ramp input to make some additional observations.

Consider the control structure shown in Fig. 8.6. The "command generator" generates

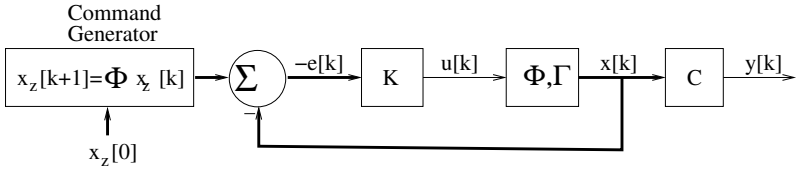


Figure 8.6 A general structure for tracking zero-input state trajectories of a system.

a zero-input trajectory of the plant corresponding to some initial condition vector $\mathbf{x}_z[0]$. The vector $\mathbf{L}$ shown in Fig. 8.6 has been calculated to place the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma K}$ in desired locations, as in the regulator problem. We now analyze the performance of the system shown in Fig. 8.6 and show that the state vector $\mathbf{x}[k]$ tracks the vector $\mathbf{x}_z[k]$ with zero steady-state error.

Define the tracking-error vector to be $\mathbf{e}[k] = \mathbf{x}[k] - \mathbf{x}_z[k]$. The following equations are obtained from Fig. 8.6

$$\mathbf{x}_z[k+1] = \mathbf{\Phi}\mathbf{x}_z[k], \tag{8.14}$$

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}u[k], \tag{8.15}$$

$$u[k] = \mathbf{K}[\mathbf{x}_z[k] - \mathbf{x}[k]]. \tag{8.16}$$

Subtracting (8.14) from (8.15) and substituting in (8.16) yields

$$\begin{aligned} \mathbf{e}[k+1] &= \mathbf{\Phi}\mathbf{e}[k] + \mathbf{\Gamma}u[k] \\ &= \mathbf{\Phi}\mathbf{e}[k] - \mathbf{\Gamma K}\mathbf{e}[k] \\ &= (\mathbf{\Phi} - \mathbf{\Gamma K})\mathbf{e}[k]. \end{aligned} \tag{8.17}$$

Since the error equation is homogeneous, and the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma K}$ are inside the unit circle, then $\mathbf{x}[k]$ will track $\mathbf{x}_z[k]$ with zero steady-state error.

EXAMPLE 8.2

In this example we consider step and ramp responses for a double integrator plant. With a sampling period of $T = 0.1$, (8.8) gives the following ZOH equivalent

$$\mathbf{\Phi} = \begin{bmatrix} 1 & .1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} .05 \\ .1 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{8.18}$$

Suppose we want a settling time of 2 seconds. To get the desired closed-loop poles in the Z-plane we should divide the roots of the normalized second-order Bessel polynomial by 2 and use the ZOH pole mapping formula. This was done in Example 8.1 to get

$$z_{1,2} = 0.8110 \pm j0.0953.$$

Using the procedure described in Chapter 6, the vector of feedback gains is calculated to be

$$\mathbf{K} = \begin{bmatrix} 4.4814 & 3.5563 \end{bmatrix}$$

The tracking system is shown in Fig. 8.6.

The matrix $\mathbf{\Phi}$ used in Fig. 8.6 is shown in (8.18). The initial condition vectors corresponding to step or ramp inputs are

$$\mathbf{x}_z[0] = \begin{bmatrix} r \\ 0 \end{bmatrix} \text{ for a step, or } \begin{bmatrix} 0 \\ v \end{bmatrix} \text{ for a ramp.}$$

The plant input and output for a step response are shown in Fig. 8.7, and for a ramp response in Fig. 8.8. Note that there is zero steady-state error for both responses, and

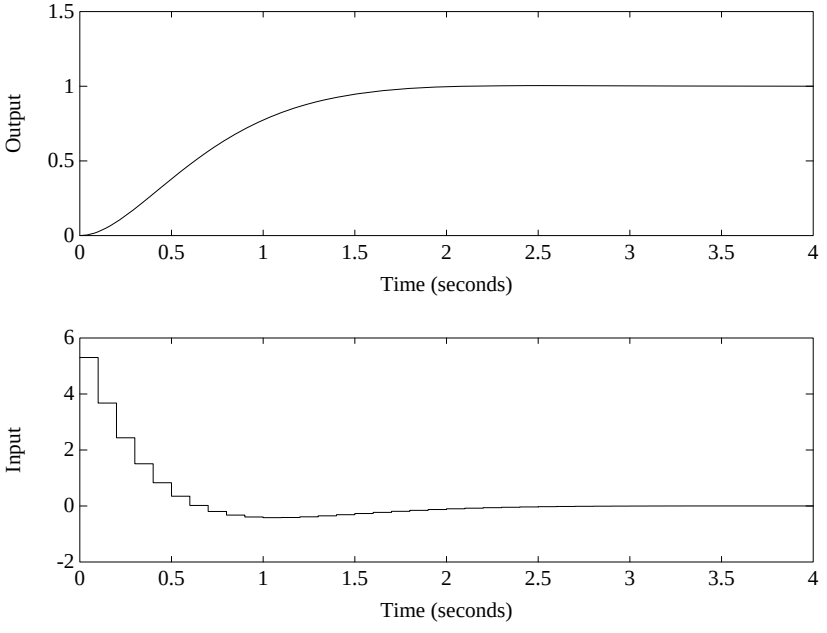


Figure 8.7 Step response for Example 8.2.

that the error between the plant output and the ramp is less than 2% by $t = 2$ seconds, as desired.

■

In Fig. 8.6, if the plant is a double integrator, then we have already seen that the command generator can generate step or ramp inputs if the proper initial condition vector is used. However, the command generator of Fig. 8.6 can *only* generate step or ramp inputs or a linear combination of a step and a ramp. A reader familiar with classical control theory might ask the following question: is it possible to construct a command generator which will convert the system in Fig. 8.6 to a single-input control system, with the input being desired position, for example? The answer to this question is yes, but we will wait until the next section to give the details of this answer. First, we will show how to track trajectories that are not zero-input state trajectories.

8.2 Adding Dynamics to Create Desired Zero-Input Trajectories

Everything we said regarding tracking systems with zero steady-state error assumed that the desired state trajectory was a zero-input trajectory of the plant. In this section, we show

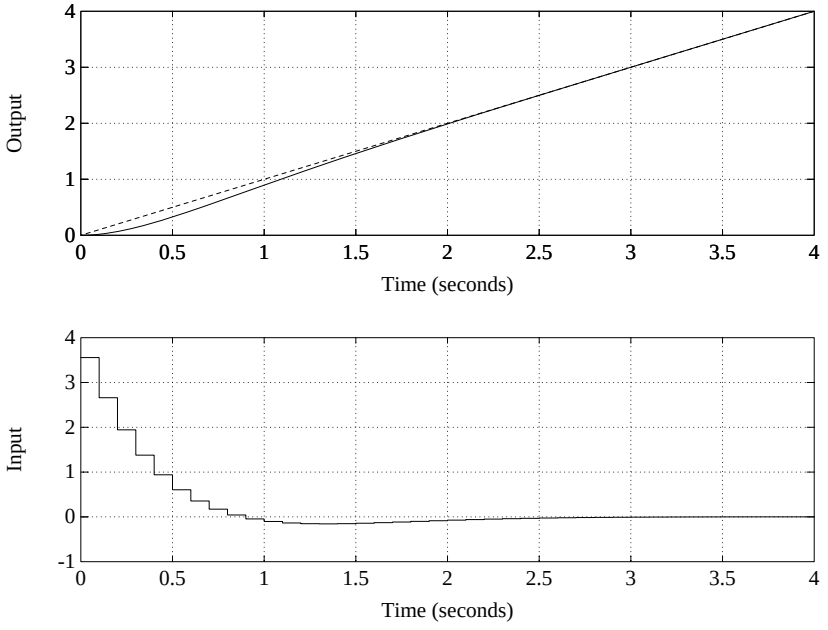


Figure 8.8 Ramp response for Example 8.2.

that it is possible to add dynamics to the plant so that the modified plant has a specified zero-input trajectory. The additional dynamics are implemented as part of the compensator. Before showing how to add dynamics, we first summarize the results obtained thus far.

Suppose that we want to track a reference signal $r[k]$. Let $R(z)$ be the z-transform of $r[k]$. We assume that the reference signals of interest have rational z-transforms so that $R(z) = n(z)/d(z)$. Signals with rational z-transforms include step, ramp, sinusoidal, and exponential signals. Then we have the following

Theorem 8.1 *Let $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C})$ be the ZOH equivalent of a plant, which is controllable and observable. Let $r[k]$ be a reference input with z-transform $R(z) = n(z)/d(z)$. Let the roots of $d(z)$ be z_i , $i = 1, \cdots, p$. If these roots are also eigenvalues of $\mathbf{\Phi}$ (counting multiplicities), then $r[k]$ can be tracked with zero steady-state error by a regulator structure with no additional dynamics. If at least one of the z_i is not an eigenvalue of $\mathbf{\Phi}$, then additional dynamics must be used to have a tracking system with zero steady-state error.*

PROOF. Let the transfer function of plant be $H(z) = b(z)/a(z)$, where $a(z) = \det(zI - \mathbf{\Phi})$. Recall that the roots of $a(z)$ are the eigenvalues of $\mathbf{\Phi}$. If the roots of $d(z)$ are also eigenvalues of $\mathbf{\Phi}$, then these roots must also be roots of $a(z)$, so we can write $a(z) = d(z)\bar{a}(z)$. The polynomial $\bar{a}(z)$ contains the roots of $a(z)$ that are not roots of $d(z)$.

We want to find an initial state vector $\mathbf{x}_0$ such that $r[k]$ is generated as follows

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{\Phi} x[k] \ , \ \ \mathbf{x}[0] = \mathbf{x}_0 \\ r[k] &= \mathbf{C}\mathbf{x}[k]. \end{aligned} \tag{8.19}$$

If we can show that (8.19) is satisfied, then $r[k]$ is associated with a zero-input state trajectory of the plant, and we showed in the previous section that such signals can be tracked with zero steady-state error using a regulator structure.

We start the derivation by taking z-transforms of (8.19) with non-zero initial conditions

$$zX(z) - \mathbf{x}[0] = \mathbf{\Phi}X(z) \tag{8.20}$$

or

$$X(z) = (z\mathbf{I} - \mathbf{\Phi})^{-1}\mathbf{x}_0. \tag{8.21}$$

Substituting the above equation into the z-transform of the output equation in (8.19) yields

$$\begin{aligned} R(z) &= \mathbf{C}(zI - \mathbf{\Phi})^{-1}\mathbf{x}_0 \\ &= \frac{\mathbf{c}\ \text{adj}(z\mathbf{I} - \mathbf{\Phi})\mathbf{x}_0}{d(z)\bar{a}(z)}. \end{aligned} \tag{8.22}$$

If $(\mathbf{\Phi}, \mathbf{C})$ is observable, we can choose $\mathbf{x}_0$ to make the numerator equal $\bar{a}(z)n(z)$. In other words, $\mathbf{x}_0$ can be chosen so that $R(z) = n(z)/d(z)$, and the output of (8.19) is the desired reference trajectory. (A method to calculate $\mathbf{x}_0$ is given in the paragraph following this proof.)

Notice from (8.22) that the denominator polynomial of $R(z)$ produced by a zero-input state trajectory can at most be $a(z)$ with possibly some factors cancelled. If one of the roots of $d(z)$ is not an eigenvalue of $\mathbf{\Phi}$, then such a reference trajectory cannot be obtained from a zero-input state trajectory. Additional dynamics must be added to the tracking system, as shown in the following section.

■

Once we know the conditions under which a reference input can be obtained from a zero-input state trajectory, it is easy to calculate the required initial state vector. For an n^{th}-order plant we have from (8.19)

$$\begin{bmatrix} r[0] \\ r[1] \\ \vdots \\ r[n-1] \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{C\Phi} \\ \vdots \\ \mathbf{C\Phi}^{n-1} \end{bmatrix} \mathbf{x}_0 \quad \text{or} \quad \mathbf{r} = \mathbf{W}_o\mathbf{x}_0. \tag{8.23}$$

We can solve for $\mathbf{x}_0$ as

$$\mathbf{x}_0 = \mathbf{W}_o^{-1}\mathbf{r}. \tag{8.24}$$

For example, recall the problem of tracking a ramp input with a double integrator system. Since $R(z) = 1/(z-1)^2$, we see that the roots of $d(z)$ are 1 and 1. The matrix $\mathbf{\Phi}$ has two eigenvalues at $z = 1$, and so the conditions of the theorem are met. The first two samples of the ramp input, and the observability matrix for the plant are

$$\mathbf{r} = \begin{bmatrix} 0 \\ vT \end{bmatrix}, \quad \mathbf{W}_o = \begin{bmatrix} 1 & 0 \\ 1 & T \end{bmatrix} \tag{8.25}$$

so

$$\mathbf{x}_0 = \frac{1}{T}\begin{bmatrix} T & 0 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ vT \end{bmatrix} = \begin{bmatrix} 0 \\ v \end{bmatrix}. \tag{8.26}$$

Notice that this is the same initial condition vector which was presented in (8.12).

We now show how to add dynamics to the tracking system when the conditions of the theorem are not satisfied. If $R(z)$ has poles $z_1, \cdots z_m$ that are not eigenvalues of $\mathbf{\Phi}$, form a polynomial from these roots as follows

$$\delta(z) = (z - z_1)(z - z_2) \cdots (z - z_m). \tag{8.27}$$

Using $\delta(z)$, we define an *additional dynamics* system $z^m/\delta(z)$ which is connected to the plant to form a *design plant* as shown in Fig. 8.9. By construction, the augmented design

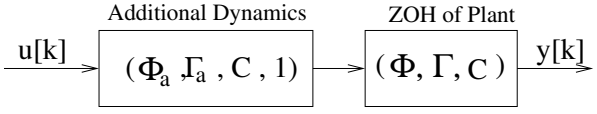


Figure 8.9 A design plant formed by cascading additional dynamics to a plant.

plant will satisfy the exact tracking property of the theorem, and so we can design a *regulator for the augmented plant that will achieve exact tracking*. The additional dynamics block will be implemented as part of the compensator.

A state-space description of the additional dynamics can always be written using observable canonical form as

$$\begin{aligned} \mathbf{x}_a[k+1] &= \mathbf{\Phi}_a \mathbf{x}_a[k] + \mathbf{\Gamma}_a u[k] \\ y_a[k] &= [\,1 \quad 0 \quad \cdots \quad 0\,]\,\mathbf{x}_a[k] + u[k] \end{aligned} \tag{8.28}$$

where

$$\mathbf{\Phi}_a = \begin{bmatrix} -\delta_1 & 1 & 0 & \cdots & 0 \\ -\delta_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\delta_{m-1} & 0 & \cdots & 0 & 1 \\ -\delta_m & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \mathbf{\Gamma}_a = \begin{bmatrix} -\delta_1 \\ -\delta_2 \\ \vdots \\ -\delta_m \end{bmatrix}. \tag{8.29}$$

The state-space description of the augmented plant (additional dynamics plus plant) can be obtained as the cascade combination of the description in (8.29) and the plant using the formula for the cascade connection of two state-space systems given in Chapter 3, (3.139). If we call the system matrix for the design system $\mathbf{\Phi}_d$ and the input vector for the design system $\mathbf{\Gamma}_d$ then the result is

$$\mathbf{\Phi}_d = \begin{bmatrix} \mathbf{\Phi}_a & 0 \\ \mathbf{\Gamma c}_a & \mathbf{\Phi} \end{bmatrix}, \quad \mathbf{\Gamma}_d = \begin{bmatrix} \mathbf{\Gamma}_a \\ \mathbf{0} \end{bmatrix}. \tag{8.30}$$

EXAMPLE 8.3

In this example, the plant is a Type-1 servo with $p = 1$ and the sampling interval is $T = 0.1$ sec. From Example 8.1, the ZOH equivalent of the plant is

$$\mathbf{\Phi} = \begin{bmatrix} 1 & .0952 \\ 0 & .9048 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} .0048 \\ .0952 \end{bmatrix}.$$

Suppose we want the output of the plant to track a unit triangle wave having a period of 2 sec. This reference input does not have a rational transform, and so we

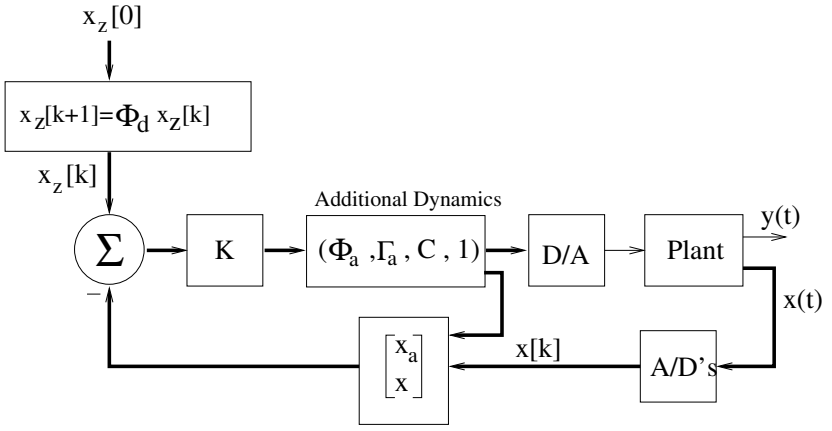


Figure 8.10 General structure for a tracking system with additional dynamics. The feedback vector L is designed for the design plant of Fig. 8.9.

will instead use just the first two terms in a Fourier series expansion of a triangle wave as the reference input. These terms can be calculated to be

$$r(t) = \frac{1}{1.125}[\sin(\pi t) - \frac{1}{8}\sin(3\pi t)]. \tag{8.31}$$

In Fig. 8.11 we show an actual triangle wave (dotted line) and the approximation $r(t)$ (solid line). The signal $r(t)$ is a smooth approximation to the triangle wave. The

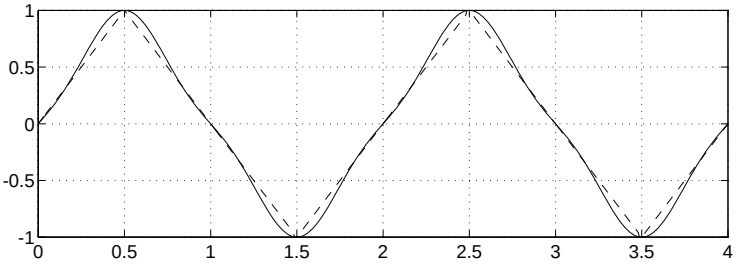


Figure 8.11 A triangle wave with a 2 second period (dashed line) and an approximation consisting of the first two harmonics in a Fourier series expansion of the triangle wave (solid line).

Laplace transform of $r(t)$ in (8.31) can be found using Table 3.2 on page 84. The result is

$$R(s) = \frac{1}{1.125}\left[\frac{\pi}{s^2+\pi^2} - \frac{1}{8}\frac{3\pi}{s^2+(3\pi^2)}\right].$$

The poles of $R(s)$ are at

$$s_{1,2} = \pm j\pi$$
$$s_{3,4} = \pm j3\pi.$$

These s-plane poles are mapped to the z-plane using the ZOH pole-mapping formula as follows

$$z_{1,2} = e^{s_{1,2}T} = e^{\pm j\pi(0.1)} = .9511 \pm .3090$$
$$z_{3,4} = e^{s_{3,4}T} = e^{\pm j3\pi(0.1)} = .5878 \pm .8090.$$

We must construct additional dynamics with these four pole locations. To do this, we first multiply out the poles to get $p(z)$

$$p(z) = \prod_{i=1}^{4}(z - z_i) = z^4 - 3.0777z^3 + 4.2361z^2 - 3.0777z + 1.$$

Using the coefficients of $p(z)$, the additional dynamics are obtained from (8.29) as

$$\mathbf{\Phi}_a = \begin{bmatrix} 3.0777 & 1 & 0 & 0 \\ -4.2361 & 0 & 1 & 0 \\ 3.0777 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{\Gamma}_a = \begin{bmatrix} 3.0777 \\ -4.2361 \\ 3.0777 \\ -1 \end{bmatrix}$$

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

The design model is the cascade combination of the additional dynamics and the plant, and is obtained from (8.30) as

$$\mathbf{\Phi}_d = \begin{bmatrix} \mathbf{\Phi}_a & \mathbf{0} \\ \mathbf{\Gamma C}_a & \mathbf{\Phi} \end{bmatrix}, \quad \mathbf{\Gamma}_d = \begin{bmatrix} \mathbf{\Gamma}_a \\ \mathbf{0} \end{bmatrix}$$

$$\mathbf{C}_d = \begin{bmatrix} \mathbf{0} & \mathbf{C} \end{bmatrix}.$$

The state vector for the design model consists of the state vector for the additional dynamics stacked on top of the state vector for the plant

$$\mathbf{x}_d[k] = \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{x}_a[k] \end{bmatrix}.$$

The design model is a 6th-order system. The first four state variables are for the additional dynamics, and the last two state variables are the state variables of the plant.

We choose a settling time of $T_s = 1$ second and map the roots of the 6th-order normalized Bessel polynomial into the z-plane using the ZOH pole mapping formula to obtain

$$z_{1,2} = e^{-4.2169 \pm j7.5300} = .4786 \pm j.4486$$
$$z_{3,4} = e^{-6.2613 \pm j4.4018} = .4837 \pm j.2278$$
$$z_{5,6} = e^{-7.1205 \pm j1.4540} = .4855 \pm j.0711$$

The feedback vector which places the poles of the closed-loop design model at these locations is calculated to be

$$\mathbf{K} = \begin{bmatrix} .9673 & .1459 & -.1700 & -.0732 & 4.2715 & 1.6551 \end{bmatrix}.$$

With these calculations, we are almost ready to implement the control system shown in Fig. 8.10. The only thing left to do is to calculate the initial condition vector $\mathbf{x}_c[0]$ corresponding to a triangle wave for plant position. To use (8.23), we need the first 6 samples of $r[k]$ which we get from (8.31) as follows

$$r[k] = r(t)\,|_{t=kT} = \frac{1}{1.125}[\sin(\pi k(.1)) - \frac{1}{8}\sin(3\pi k(.1))]$$

which gives

$$\mathbf{r} = \begin{bmatrix} r[0] \\ r[1] \\ r[2] \\ r[3] \\ r[4] \\ r[5] \end{bmatrix} = \begin{bmatrix} 0 \\ 0.2079 \\ 0.4689 \\ 0.7704 \\ 1.0245 \\ 1.1250 \end{bmatrix}.$$

To use (8.23) we also need the observability matrix for the design model

$$\mathbf{W}_o = \begin{bmatrix} \mathbf{c}_a \\ \mathbf{c}_a\mathbf{\Phi}_a \\ \vdots \\ \mathbf{c}_a\mathbf{\Phi}_a^5 \end{bmatrix}.$$

Then the initial condition vector is obtained as

$$\mathbf{x}_c[0] = \mathbf{W}_o^{-1}\mathbf{r} = \begin{bmatrix} 5.0821 \\ -7.1598 \\ -.7769 \\ 1.4961 \\ 0 \\ 1.6835 \end{bmatrix}.$$

With this initial condition vector, the tracking system in Fig. 8.10 will cause the plant output to track $r(t)$. The plant input and output are shown in Fig. 8.12. Notice that the difference between the plant output (solid line) and $r(t)$ (dashed line) is less than 2% by $t = 1$ second, as desired.

■

8.3 Command Generators Using Observers

In the previous sections, we considered tracking systems which were driven by zero-input state trajectories of the plant, or of the plant augmented with additional dynamics. For an n^{th}-order single-input design plant, we need to generate n reference trajectories, one for each state variable. Readers familiar with classical control theory will wonder if a tracking system can be designed which only needs a single reference trajectory which the output of the plant should follow. In this section, we show that it is possible to design a single-input tracking system by using an observer to estimate the state-command vector $\mathbf{x}_c[k]$ from a scalar reference trajectory. The results of this section should be thought of as a preliminary development to those of the following section. Section 8.4 describes a better way to design tracking systems; however, Section 8.4 builds upon ideas developed in this section.

Recall that a command generator generates zero-input trajectories $\mathbf{x}_z[k]$ of the plant. These trajectories satisfy

$$\mathbf{x}_z[k+1] = \mathbf{\Phi}\mathbf{x}_z[k]. \tag{8.32}$$

In general, the output of the plant is some linear combination of state variables, say $\mathbf{c}\mathbf{x}[k]$. Since the goal of a tracking system is to have $y[k]$ track a reference input $r[k]$, it is natural to express $r[k]$ as the same linear combination of state variables as $y[k]$, that is

$$r[k] = \mathbf{C}\mathbf{x}_z[k]. \tag{8.33}$$

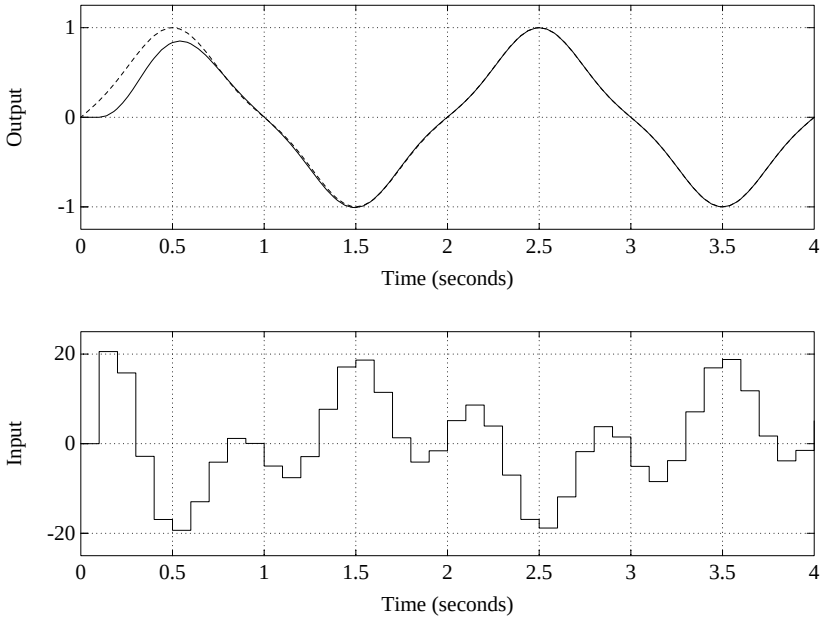


Figure 8.12 The plant input and output for the tracking system shown in Fig. 8.10. The additional dynamics and initial condition vector are calculated to make the plant output track $r(t)$ shown in (8.31). In the top figure, the dashed line is $r(t)$ and the solid line is the plant output.

An on-line command generator takes $r[k]$ as input and produces an estimate of the entire state vector $\hat{\mathbf{x}}_z[k] = \mathbf{x}_c[k]$ which is the state-vector command sequence for the tracking system. Equation (8.32) is a state-update equation with the input equal zero, or equivalently, with the input vector $\mathbf{\Gamma} = \mathbf{0}$. If we interpret (8.33) as an output equation, then an on-line command generator is just an observer for the system described by (8.32) and (8.33). In other words, the operation of an on-line command generator can be described as follows. Given an observation of a linear combination of the state vector (8.33), estimate the entire state vector using (8.32). If the output of the plant is just one of the state variables, then the specification of the on-line command generator is identical to the system shown in Fig. 7.18 on page 318 with $u[k] = 0$, or equivalently, $\mathbf{\Gamma}$ and H both zero. Thus the design equations in Table 7.3 on page 319 can be used to design on-line command generators. The design equations are used with $\mathbf{\Gamma}_1 = \mathbf{\Gamma}_2 = \mathbf{0}$ which implies that H will be zero.

For simplicity, assume that the output of the system is the first state variable so that $\mathbf{C} = [\,1 \quad 0 \quad \cdots \quad 0\,]$. Fig. 8.13 shows how $r[k]$ is modeled as the output of linear system, and the reduced-order observer generates estimates of the remaining state variables. The appropriate design procedure for Case 1, 2, or 3 from Chapter 6 can be used to design the reduced-order observer. The model in Fig. 8.13 assumes that $r[k]$ is generated by the dynamics of the plant. Actually, $r[k]$ is an arbitrary reference input. An on-line command generator that uses a reduced-order observer works best when the dynamics of the reference input signal are similar to the dynamics of the plant. This is illustrated in the following example.

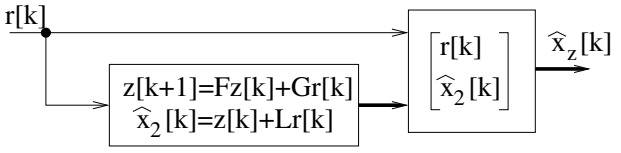


Figure 8.13 A system which models $r[k]$ as the first state variable of a zero-input system. A reduced-order observer estimates the remaining state variables.

EXAMPLE 8.4

In this example we consider step and ramp responses when the plant is a double integrator. No additional dynamics are needed in this case. In Example 8.2, step and ramp responses were obtained using a command generator which produced the required zero-input state trajectory. Here we show how similar results can be obtained by specifying only desired position, $x_1[k]$, and using a reduced-order observer to estimate desired velocity, $x_2[k]$. This is a special case of the system shown in Fig. 8.13. As in Example 8.2, we choose the sampling period to be $T = 0.1$ sec, and feedback gain vector $\mathbf{L}$ corresponding to Bessel roots with a settling time of 2 seconds.

A reduced-order observer for the double integrator plant was designed in Chapter 7 and has the following form

$$\begin{aligned} z[k+1] &= Fz[k] - \tfrac{(F-1)^2}{T} r[k]\ , \quad z[0] = \tfrac{F-1}{T} r[0] \\ \hat{x}_2[k] &= z[k] + \tfrac{(1-F)}{T} r[k] \end{aligned} \tag{8.34}$$

where F is the observer pole location which determines the speed of the observer.

As shown in Fig. 8.13, the reference input and the output of the reduced-order observer are combined to form an estimated zero-input state trajectory

$$\hat{\mathbf{x}}_z[k] = \begin{bmatrix} r[k] \\ \hat{x}_2[k] \end{bmatrix}. \tag{8.35}$$

The vector signal $\hat{\mathbf{x}}_z[k]$ can be used in place of $\mathbf{x}_z[k]$ shown in Fig. 8.6. When $r[k]$ is a unit step, the estimated velocity $\hat{x}_2[k]$ is always zero (for any value of F). The reason is that the observer is initialized to give $\hat{x}_2[0] = 0$ (the true initial velocity), and the actual velocity doesn't change from zero. Thus the step response will be identical to that shown in Example 8.2.

When the reference input is a ramp, the observer must estimate the velocity which is now non-zero. Different results are obtained for different values of F, the observer pole. In Fig. 8.14 we show the results for $F = 0.1$, and in Fig. 8.15 the results for $F = 0.8$ As expected, the faster observer with $F = 0.1$ causes $\hat{x}_2[k]$ to approach 1 (the true velocity) more quickly, and allows the difference between the plant output and $r[k]$ to go to zero more quickly.

■

The above approach to the design of tracking systems works, but it has the following two disadvantages:

1. It is necessary to design a reduced-order observer, in addition to designing the regulator.

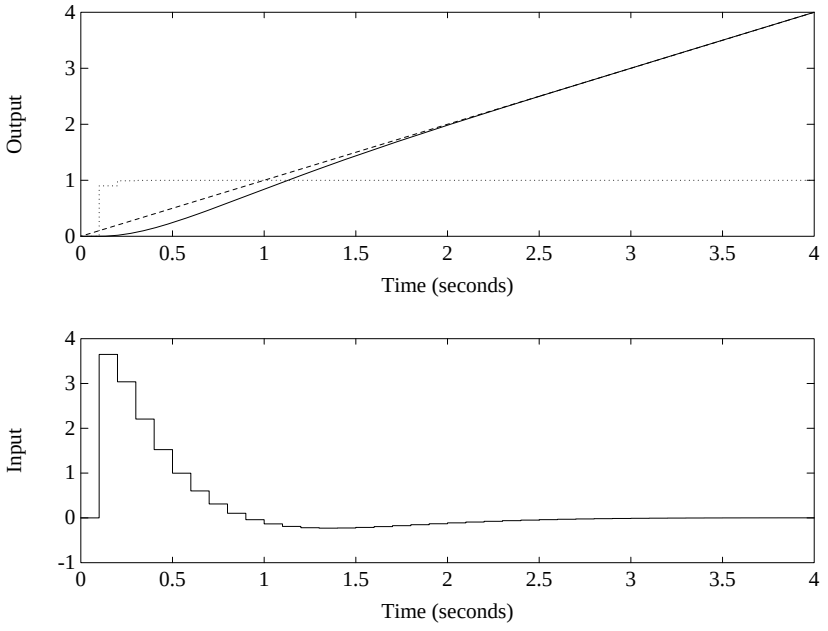


Figure 8.14 Ramp response for the system in Example 8.4. The observer pole is $F = 0.1$. In the top graph, the solid line is the plant output, the dashed line is the desired ramp, and the dotted line is the estimated velocity from the reduced-order observer.

2. The reduced-order observer may contain many of the same terms as the additional dynamics. This is an unnecessary duplication, as will be shown below.

Thus it will be to our advantage to re-arrange the structure of the tracking system derived above. The rearrangement does not have the duplication mentioned above, and the new structure also will be shown to have a desirable robustness property. The price for these advantages is that the additional dynamics for the new structure will be slightly more complicated than those for the structure derived above. However, the additional dynamics for the new structure are calculated in a manner similar to the one discussed above.

8.4 Tracking System Design

The tracking systems discussed in previous sections used state-feedback vectors calculated for a design plant consisting of the ZOH the system to be controlled possibly preceded by additional dynamics (see Fig. 8.10). In this section we also consider a design plant consisting of the ZOH of the plant and additional dynamics, but we reverse the order of the blocks, as shown in Fig. 8.16. A state-update equation for the design model shown in Fig. 8.16 can be obtained as before by defining a composite state vector

$$\mathbf{x}_d[k] = \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{x}_a[k] \end{bmatrix}. \tag{8.36}$$

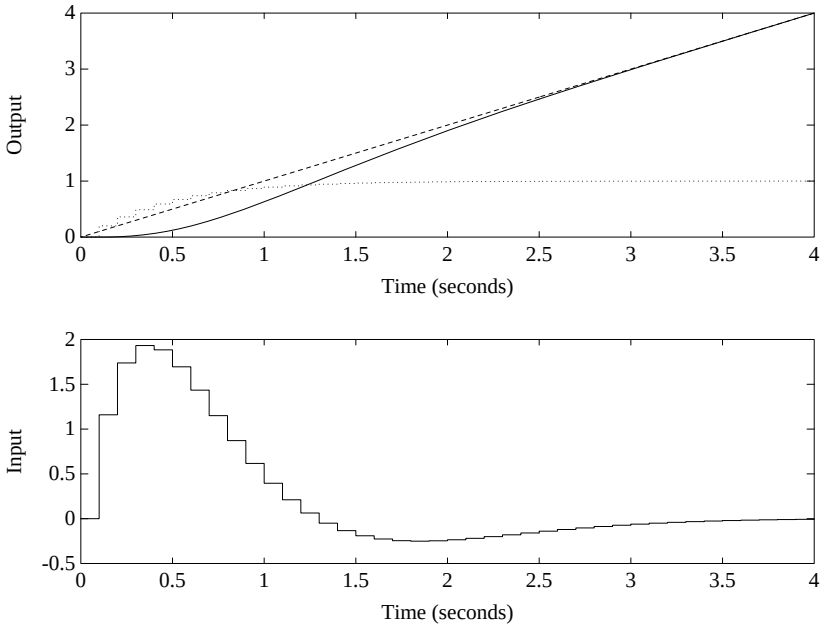


Figure 8.15 Ramp response for the system in Example 8.4. The observer pole is $F = 0.8$. In the top graph, the solid line is the plant output, the dashed line is the desired ramp, and the dotted line is the estimated velocity from the reduced-order observer.

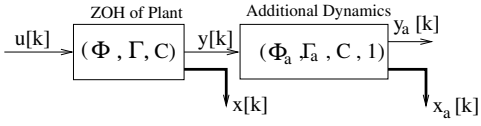


Figure 8.16 A design plant consisting of the ZOH of the system to be controlled followed by additional dynamics.

Then using the formula for the cascade connection of two state-space systems given in Chapter 3, (3.139), the state-space description of the design plant is

$$\mathbf{\Phi}_d = \begin{bmatrix} \mathbf{\Phi} & \mathbf{0} \\ \mathbf{\Gamma}_a\mathbf{C} & \mathbf{\Phi}_a \end{bmatrix}, \quad \mathbf{\Gamma}_d = \begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{0} \end{bmatrix}. \tag{8.37}$$

We can design a regulator for $(\mathbf{\Phi}_d, \mathbf{\Gamma}_d)$, and partition the feedback gains as

$$\mathbf{K} = [\,\mathbf{K}_1 \quad \mathbf{K}_2\,], \tag{8.38}$$

where $\mathbf{K}_1$ consists of the first n elements of $\mathbf{K}$. The design plant regulated with the feedback vector $\mathbf{K}$ is shown in Fig. 8.17. Now suppose we introduce a reference input $r[k]$

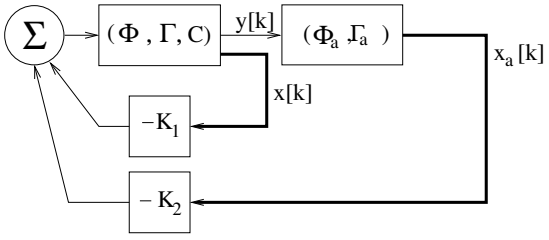


Figure 8.17 The design plant regulated by the feedback vector $\mathbf{K}$

into the regulated system of Fig. 8.17. A logical place to introduce the reference input is to compare it with the output of the plant, as shown in Fig. 8.18.

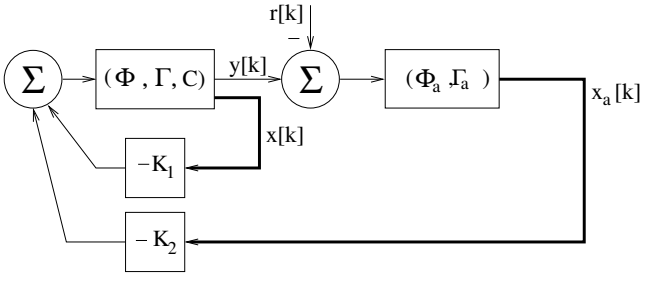


Figure 8.18 The system in Fig. 8.17 with the addition of a reference input.

The system in Fig. 8.18 can be re-arranged to get the system shown in Fig. 8.19. Note that the minus sign at the summing junction has been absorbed into $\mathbf{K}_2$. The system in

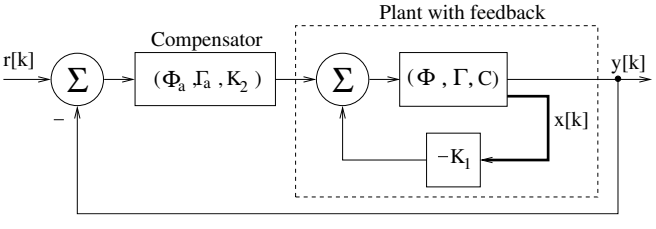


Figure 8.19 A re-arrangement of the system in Fig. 8.18.

Fig. 8.19 is a negative unity feedback system of the type studied in classical control theory.

It is known that such systems will track the reference input with zero steady-state error if the poles of $R(z)$ appear in the forward path of the control system (see Section 5.8 in Chapter 5). From Fig. 8.19 we see that the poles of the plant are modified by the feedback vector $\mathbf{K}_1$. Thus the poles of $R(z)$ must *all* appear as eigenvalues of the additional dynamics. Note the difference here with the results of the previous section, where the additional dynamics only had to provide poles of $R(z)$ that did not appear in the plant. Although the additional dynamics will be more complicated for the system in Fig. 8.19 as compared with the structure derived in the previous section, an advantage of the current structure is that no observer needs to be designed. The reference inputs are introduced in a natural way into the regulator structure.

Another advantage that the system in Fig. 8.19 has over the earlier structure is that the system in Fig. 8.19 provides robustness to disturbances and model inaccuracies. Both of these facts are known from classical control theory. If a disturbance $d[k]$ is introduced at the input to the plant in Fig. 8.19, then the effect of this disturbance at the output of the plant will go to zero if the poles of $D(z)$ are also eigenvalues of the compensator. Thus the system in Fig. 8.19 can reject disturbances.

The statement about robustness to model inaccuracies is explained as follows. Suppose that the feedback vector $\mathbf{L}$ is calculated from the design model, but that the design model does not contain an accurate model for the plant. The behavior of the actual closed-loop system will then be different from that predicted by the system in Fig. 8.19. However, if the actual closed-loop system is stable, *then the actual system will track the reference input with zero steady-state error*. The reason is that zero steady-state error is achieved whenever the poles of $R(z)$ appear as poles of the compensator. This property does not depend on an accurate model of the plant.

The properties of robustness to disturbances and model inaccuracies which the system in Fig. 8.19 possesses are not shared by the system derived in the previous section. This will be illustrated by an example. First, however, we give a complete design procedure for designing tracking systems of the type shown in Fig. 8.19. The above discussion indicates that the additional dynamics must include the poles of the reference trajectory, as well as poles of disturbances which are to be rejected. The following design procedure is a discrete-time version of one given by Davison [18], and formal proofs can be found in his paper. The design procedure is based on state-space models for the plant, the disturbances, the reference input, and the measurements. These models are shown below.

Plant

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + \mathbf{B}u(t) + \mathbf{E}\omega(t) \\ y(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{F}\omega(t) \end{aligned} \tag{8.39}$$

where $\omega(t)$ is a disturbance signal. Notice that the disturbance can enter the state or output equation (or both) depending on the values of $\mathbf{E}$ and $\mathbf{F}$. Examples of different disturbance scenarios will be given shortly.

Disturbance This design method assumes that the disturbance $\omega(t)$ is the output of a homogeneous (zero-input) state-space equation

$$\begin{aligned} \dot{\mathbf{z}}_d(t) &= A_d\mathbf{z}_d(t) \\ \omega(t) &= \bar{\mathbf{C}}_d\mathbf{z}_d(t). \end{aligned} \tag{8.40}$$

Reference Input The reference input is also assumed to be the output of a homogeneous state-space model

$$\begin{aligned} \dot{\mathbf{z}}_r(t) &= A_r \mathbf{z}_r(t) \\ r(t) &= \bar{\mathbf{C}}_r \mathbf{z}_r(t). \end{aligned} \tag{8.41}$$

Measurements The signals that are measured and available for the control system to use are assumed to be a combination of plant state variables possibly corrupted by the disturbance

$$\mathbf{y}_m(t) = \mathbf{C}_m \mathbf{x}(t) + \mathbf{F}_m \omega(t). \tag{8.42}$$

We assume that the disturbance $\omega(t)$ cannot be directly measured (then we could "subtract out" its effects). However, if $\mathbf{F}_m \neq \mathbf{0}$, the disturbance affects the measurements as shown in (8.42).

We make the following assumption regarding $\mathbf{C}_m$ and $\mathbf{F}_m$ from (8.42) and $\mathbf{C}$ and f from (8.39):

Assumption There exists a row vector $\mathbf{C}_o$ which satisfies

$$\mathbf{C}_o \mathbf{C}_m = \mathbf{C} \text{ and } \mathbf{C}_o \mathbf{F}_m = f. \tag{8.43}$$

The meaning of this assumption is given in the second remark below.

Remarks

1. The tracking system derived in this section can be used with reference inputs that do *not* satisfy the assumed model (8.41), and with disturbances that do not satisfy the model in (8.40). In this case, however, the tracking may not be perfect (output equals reference input), even asymptotically.

2. The actual output that should track the reference trajectory is the output y of the plant in (8.39). We assume that the measurement vector $\mathbf{y}_m$ contains the actual output in the sense that there exists a row-vector $\mathbf{c}_0$ such that

$$y = \mathbf{C}_0 \mathbf{y}_m. \tag{8.44}$$

Thus the output we want to control can be obtained from the measured signals. If this condition is *not* satisfied then we have no way of insuring that y is tracking the reference input. Using (8.39) and (8.42), we can re-write (8.44) as follows

$$\mathbf{C}\mathbf{x}(t) + f\omega(t) = \mathbf{C}_o[\mathbf{C}_m \mathbf{x}(t) + \mathbf{F}_m \omega(t)]. \tag{8.45}$$

Since this equation must be true for all values of $\mathbf{x}(t)$ and $\omega(t)$, we can equate the coefficients of $\mathbf{x}(t)$ and $\omega(t)$ separately to obtain the two equations given in (8.43).

The relationships between $\mathbf{C}$, $\mathbf{C}_m$, and $\mathbf{C}_o$ are given below for three common cases – when all the state variables are measured, when the measurements only depend on p state variables, and when the measurement consists only of the plant output. In the first case, suppose the output equation for the plant is

$$y(t) = \mathbf{C}\mathbf{x}(t)$$

and the measurement vector is $\mathbf{y}_m = \mathbf{x}$ (i.e. $\mathbf{C}_m = \mathbf{I}$). These measurement contain the output since (8.44) is satisfied with $\mathbf{C}_0 = \mathbf{C}$.

In the second case the output of the plant depends only on a subset of state variables, say the first p, and we measure only these state variables. In this case, the output vector for the plant is of the form $\mathbf{C} = [\mathbf{C}_1\ \mathbf{0}]$ and the measurements are described by

$$\mathbf{y}_m = \begin{bmatrix} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x}.$$

In this case, (8.44) is satisfied with $\mathbf{c}_0 = \mathbf{c}$.

A third possibility is that we only measure the output, $\mathbf{y}_m = y$. In this case, (8.44) is satisfied with $\mathbf{C}_0 = 1$.

3. The disturbance can affect the plant and/or the measurements. If $\mathbf{C}_m = \mathbf{I}$ and $\mathbf{F}_m = \mathbf{0}$, then the measurement vector $\mathbf{y}_m = \mathbf{x}$. In other words, we are measuring all the state variables. In this case, if $\mathbf{F}_m \neq \mathbf{0}$, we are measuring the state variables subject to some disturbance, but this disturbance is not necessarily affecting the plant. The disturbance only affects the plant if $\mathbf{E}$ or $\mathbf{F}$ is nonzero.

With the above models defined, we are ready to describe the design procedure, and to state the properties of the resulting control system. The design procedure consists of four steps which are described below. After explaining each step, the properties of the resulting control system will be given.

8.4.1 Algorithm to Design Tracking Systems

Step 1: Choosing the Sampling Interval for a Tracking System The gains for a state-feedback tracking system are obtained by calculating state-feedback regulator gains for the design model. Thus, the guidelines for choosing the sampling interval of a state-feedback regulator are relevant for a tracking system. The only difference is that the design model that is being regulated is of order $n + q$, where n is the order of the plant and q is the order of the additional dynamics. Thus, the guideline for choosing the sampling interval of a digital state-feedback tracking system with settling time T_s seconds is:

$$T = \min\left(\frac{\pi}{5\beta_{max}}, \frac{T_s}{20(n+q)}\right), \tag{8.46}$$

where β_{max} is the largest imaginary part of the plant poles, n is the order of the plant, and q is the order of the additional dynamics.

Step 2: Calculate the necessary additional dynamics The additional dynamics must contain the poles of the reference input and the disturbance. The poles are the eigenvalues of A_r and A_d which come from the above models for the reference input and disturbance. Let Λ_d and Λ_r be sets describing the eigenvalues (with multiplicities) of A_d and A_r, respectively. That is, let

$$\Lambda_d = \{(\lambda_{d_1}, m_{d_1}), (\lambda_{d_2}, m_{d_2}), \cdots\}$$

$$\Lambda_r = \{(\lambda_{r_1}, m_{r_1}), (\lambda_{r_2}, m_{r_2}), \cdots\}$$

where λ_{d_i} is an eigenvalue of A_d with multiplicity m_{d_i}, and the set Λ_r is similarly defined.

In the first part of Step 2, we define the set Λ to be the union of the sets Λ_d and Λ_r subject to the following restriction. If a common eigenvalue appears in Λ_d and Λ_r, we

include only one of these in Λ, the one with the higher multiplicity. Step 1 is illustrated by two examples

1. Track a step input, reject a step disturbance

$$\Lambda_r = \{(0,1)\}\ ,\ \ \Lambda_d = \{(0,1)\}\Lambda = \{(0,1)\}.$$

2. Track a step input, reject a sinusoidal disturbance with radian frequency ω_0

$$\Lambda_r = \{(0,1)\}\ ,\ \ \Lambda_d = \{(j\omega_0,1),(-j\omega_0,1)\}\Lambda = \{(0,1),(j\omega_0,1),(-j\omega_0,1)\}.$$

We make the following remarks about this part of Step 2. First, note that when we track an input of a certain type (e.g. a step or a ramp), we automatically reject disturbances of the same type. For instance, in example 1 above if we let Λ_d equal the null set (don't ask for any disturbance rejection), the result would still be $\Lambda = \{(0,1)\}$, and the control system would still reject step disturbances.

The second remark is that the control system will still work if the roles of the reference input and disturbance are reversed. For instance, if we want to track a sinusoidal input and reject a step disturbance, we would use the same set Λ given in example 2 above.

The next part of Step 2 is to map the eigenvalues in Λ using the ZOH pole-mapping formula and multiply them out to form the denominator polynomial for the additional dynamics. If Λ contains eigenvalues λ_i with multiplicity m_i, and if the total number of eigenvalues is $s \stackrel{\text{def}}{=} \sum m_i$, then the denominator polynomial of the additional dynamics is calculated as

$$\delta(z) = \prod_i (z - e^{\lambda_i T})^{m_i} \stackrel{\text{def}}{=} z^s + \delta_1^{s-1} + \cdots + \delta_s. \tag{8.47}$$

The denominator polynomials for the two examples given above can be calculated as follows

1. $\delta(z) = (z - e^{0T}) = z - 1$ so $\delta_1 = -1$.
2.

$$\begin{aligned}\delta(z) &= (z - e^{0T})(z - e^{j\omega_0 T})(z - e^{-j\omega_0 T}) \\ &= (z-1)(z^2 - 2\cos(\omega_0 T)z + 1) \\ &= z^3 - (1 + 2\cos(\omega_0 T)z^2 + (1 + 2\cos(\omega_0 T))z - 1\end{aligned}$$

so

$$\begin{aligned}\delta_1 &= 1 + 2\cos(\omega_0 T) \\ \delta_2 &= 1 + 2\cos(\omega_0 T) \\ \delta_3 &= -1.\end{aligned}$$

The last part of Step 2 is to use the denominator polynomial $\delta(z)$ to define the state-space matrices for the additional dynamics. These matrices are shown below

$$\mathbf{\Phi}_a = \begin{bmatrix} -\delta_1 & 1 & 0 & \cdots & 0 \\ -\delta_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\delta_{s-1} & 0 & 0 & \cdots & 1 \\ -\delta_s & 0 & 0 & \cdots & 0 \end{bmatrix}\ ,\ \ \mathbf{\Gamma}_a = \begin{bmatrix} -\delta_1 \\ -\delta_2 \\ \vdots \\ -\delta_{s-1} \\ -\delta_s \end{bmatrix}. \tag{8.48}$$

A summary of Step 2, as well as the remaining steps of the design procedure can be found in Table 8.1.

Step 3: Calculate a feedback vector for the design model The design model consists of a cascade combination of the ZOH of the plant and the additional dynamics. The state-space matrices for the design model are

$$\mathbf{\Phi}_d = \begin{bmatrix} \mathbf{\Phi} & 0 \\ \mathbf{\Gamma}_a\mathbf{C} & \mathbf{\Phi}_a \end{bmatrix}, \quad \mathbf{\Gamma}_d = \begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{x}_d[k] = \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{x}_a[k] \end{bmatrix}. \tag{8.49}$$

The design model has dimension $n + s$, where n is the dimension of the plant, and s is the dimension of the additional dynamics. A feedback vector $\mathbf{K}$ for the design model will have $n + s$ elements, and it can be partitioned into the first n and last s elements as follows

$$\mathbf{K} = [\,\mathbf{K}_1 \quad \mathbf{K}_2\,].$$

Step 4: Introduce the reference input and implement the additional dynamics as part of the compensator All of the calculations have been performed in the previous steps and this step merely consists of putting the pieces together. The digital control system is shown in Fig. 8.20. This system shows the measurement vector $\mathbf{y}_m = \mathbf{x}$, i.e. full state

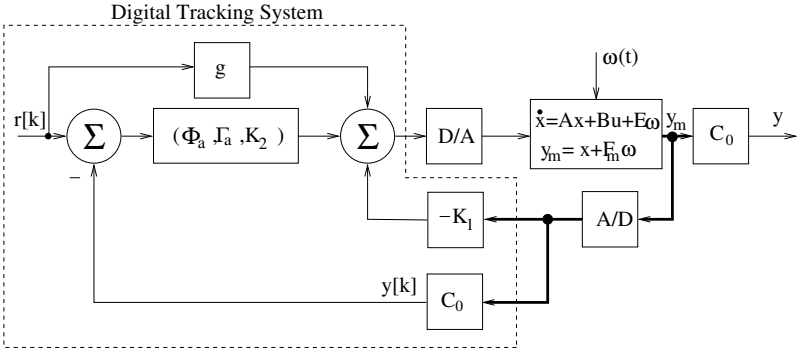


Figure 8.20 A digital tracking system which uses full state feedback.

feedback. If the full state vector cannot be measured, an observer can be used. This will be demonstrated in some of the following examples.

Remarks

- The control system shown in Fig. 8.20 has a feedforward gain g from the reference input to the plant input. This gain cannot affect the stability of the control system because it does not alter the closed-loop poles. However, this gain may improve the transient response of the system. The value of g is chosen on the basis of simulations, as demonstrated in the examples that follow.

- The tracking system shown in Fig. 8.20 has the property that the plant output $y[k]$ approaches the reference input $r[k]$ as k gets large. Tracking will be achieved in spite of disturbances and in spite of an inaccurate model of the plant, as long as the closed-loop system with the actual plant remains stable. We assume here that the model of the plant is accurate enough so that the feedback vector calculated for the design model will stabilize the actual plant.

1. The guideline for choosing the sampling interval of a digital state-feedback tracking system with settling time T_s seconds is:

$$T = \min\left(\frac{pi}{5\beta_{max}}, \frac{T_s}{20(n+q)}\right), \tag{8.50}$$

where β_{max} is the largest imaginary part of the plant poles, n is the order of the plant, and q is the order of the additional dynamics.

2. Calculate the necessary additional dynamics. The additional dynamics must contain the poles of the reference input and the disturbance. Let Λ_d and Λ_r be sets of eigenvalues (with multiplicities) of A_d and A_r, respectively. Define the set Λ to be the union of the sets Λ_d and Λ_r subject to the following restriction. If a common eigenvalue appears in Λ_d and Λ_r, we include only one of these in Λ, the one with the higher multiplicity. Map the eigenvalues in Λ using the ZOH pole-mapping formula and multiply them out to form the denominator polynomial for the additional dynamics. If Λ contains eigenvalues λ_i with multiplicity m_i, and if the total number of eigenvalues is $s \stackrel{\text{def}}{=} \sum m_i$, then the denominator polynomial of the additional dynamics is calculated as

$$\delta(z) = \prod_i (z - e^{\lambda_i T})^{m_i} \stackrel{\text{def}}{=} z^s + \delta_1 z^{s-1} + \cdots + \delta_s.$$

Use the denominator polynomial $\delta(z)$ to define the state-space matrices for the additional dynamics. These matrices are shown below

$$\mathbf{\Phi}_a = \begin{bmatrix} -\delta_1 & 1 & 0 & \cdots & 0 \\ -\delta_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\delta_{s-1} & 0 & 0 & \cdots & 1 \\ -\delta_s & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \mathbf{\Gamma}_a = \begin{bmatrix} -\delta_1 \\ -\delta_2 \\ \vdots \\ -\delta_{s-1} \\ -\delta_s \end{bmatrix}.$$

3. Calculate a feedback vector for the design model. The design model consists of a cascade combination of the ZOH of the plant and the additional dynamics. The state-space matrices for the design model are

$$\mathbf{\Phi}_d = \begin{bmatrix} \mathbf{\Phi} & 0 \\ \mathbf{\Gamma}_a\mathbf{C} & \mathbf{\Phi}_a \end{bmatrix}, \quad \mathbf{\Gamma}_d = \begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{x}_d[k] = \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{x}_a[k] \end{bmatrix}.$$

A feedback vector $\mathbf{L}$ for the design model will have $n + s$ elements, and it can be partitioned into the first n and last s elements as follows

$$\mathbf{K} = [\,\mathbf{K}_1 \quad \mathbf{K}_2\,].$$

4. Introduce the reference input and implement the additional dynamics as part of the compensator.

Table 8.1 Procedure to design a digital tracking system for a plant whose ZOH equivalent is $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C})$.

- It is important to assess the robustness of the tracking system shown in Fig. 8.20. Stability margins for tracking systems are calculated in the same way as for regulators (see Section 6.3.3 on page 248). That is, starting with the closed-loop system of Fig. 8.20, the loop is broken at the input to the plant. The resulting loop transfer function is calculated and a Nyquist plot is used to find the gain and phase margins. The loop transfer function for the tracking system in Fig. 8.20 is calculated in Problem 4.

The rules given below specify pole locations for continuous-time systems. For a digital control (discrete-time systems), these numbers, s_i, must be mapped using the ZOH pole-mapping formula, $e^{s_i T}$, where T is the sampling interval in seconds. Let `Ts` be the desired setting time for the regulator or tracking system. For what follows, we note that given a continuous-time state-space model `(A,B,C,D)`. for a plant (SISO or MIMO), the zeros of the plant are calculated by the command: `tzero(A,B,C,D)`. The rules below refer to a variable `s1`, which is the first-order normalized Bessel pole with 1-second settling time. Its numerical value is `s1=-4.62`.

Regulator Poles

1. Use normalized Bessel poles scaled (divided) by the desired settling time (see Table II).
2. Use sufficiently damped plant poles; that is, plant poles whose real parts lie to the left of `s1/Ts`.
3. If the plant has complex poles that are not sufficiently damped, choose closed-loop poles by replacing the real parts of these poles with `s1/Ts` and keeping the imaginary parts the same. Changing the real parts adds damping to those plant poles, and so these are called "added damping" poles.
4. If the plant has unstable pole locations (real part greater than zero) consider the reflection of these poles about the imaginary axis, i.e. change the sign of the (real-part) of the pole. If the reflected pole is to the left of `s1/Ts` use it as a closed-loop pole.

Tracking System Poles

1. Use all of the rules for regulator poles
2. If the plant has any zeros with negative real parts, consider using these as closed-loop poles for the tracking system. If the zero is located to the right of `s1/Ts`, this will result in choosing a closed-loop pole that is slower than the desired settling time of T_s seconds. However, this slow closed-loop pole will not affect the settling time of the tracking system because it is canceled by the corresponding plant zero.

Table 8.2 Rules for Selecting Tracking System Pole Locations.

EXAMPLE 8.5

The plant in this example is a double integrator which is subject to a constant disturbance at its input. A state-space model for the plant is

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

$$y(t) = \mathbf{C}\mathbf{x}(t)$$

where

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

and $\omega(t)$ is a step function representing the disturbance. A control system is desired which can track a reference step input in spite of constant disturbances. We will use full state feedback as shown in the system in Fig. 8.20. A settling time of $T_s = 2$ seconds is desired, and we choose a sampling interval of $T = 0.1$ seconds. The ZOH equivalent of the plant is then given by

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.005 \\ 0.1 \end{bmatrix}.$$

We follow the design procedure shown in Table 8.1. The model specification for the design algorithm is as follows

$$\begin{aligned} \mathbf{E} &= \mathbf{B} \\ \mathbf{F} &= 0 \\ \mathbf{F}_m &= \mathbf{0} \\ \mathbf{C}_m &= \mathbf{I}. \end{aligned} \tag{8.51}$$

Step 1 Since the reference input and disturbance are both step functions, we have $\mathbf{A}_r = \mathbf{A}_d = 0$ in (8.40) and (8.41). Thus the set $\Lambda = \{0\}$ and $s = 1$. We map this eigenvalue into the z-plane and for the polynomial $\delta(z)$ as follows

$$\delta(z) = z - e^{0T} = z - 1$$

which means that the coefficient $\delta_1 = -1$. Thus the additional dynamics are (using (8.48))

$$\mathbf{\Phi}_a = 1, \quad \mathbf{\Gamma}_a = 1.$$

Step 2 The design model, which consists of the cascade of the plant and the additional dynamics is shown below (see (8.49))

$$\mathbf{\Phi}_d = \begin{bmatrix} 1 & .1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad \mathbf{\Gamma}_d = \begin{bmatrix} .005 \\ .1 \\ 0 \end{bmatrix}.$$

To achieve the desired settling time of $T_s = 2$ seconds, we use the roots of the normalized third-order Bessel polynomial shown in Table 6.2 on page 257. The root locations are divided by 2 to get $T_s = 2$ with the result

$$s_1 = -2.5047$$

$$s_{2,3} = -1.9834 \pm j1.8922.$$

The s-plane closed-loop pole locations are mapped into the z-plane using the ZOH pole mapping formula, and the resulting poles are multiplied out to obtain the desired denominator polynomial for the discrete-time closed-loop system

$$z_1 = e^{s_1 T} = .7784$$

$$z_{2,3} = e^{s_{2,3} T} = .8055 \pm j.1543$$

so

$$p(z) = \prod_{i=1}^{3}(z - z_i) = z^3 - 2.3893z^2 + 1.92652z - .5235.$$

The feedback vector $\mathbf{K}$ that places the eigenvalues of $(\mathbf{\Phi}_d - \mathbf{\Gamma}_d\mathbf{K})$ at the roots of $p(z)$ is calculated to be

$$\mathbf{K} = [\,14.1023 \quad 5.4015 \quad 1.3658\,].$$

This vector is partitioned into the first two ($n = 2$) elements and the last element ($s = 1$) as follows

$$\mathbf{K}_1 = [\,14.1023 \quad 5.4015\,] \;,\; \mathbf{K}_2 = [\,1.3658\,].$$

We first simulate the step response when $\omega(t) = 0$ and vary the value of the feedforward gain g. Fig. 8.21 shows the step responses for $g = 0, 5$, and 10. We select $g = 5$ as the best value for the feedforward gain, and use this value in what follows.

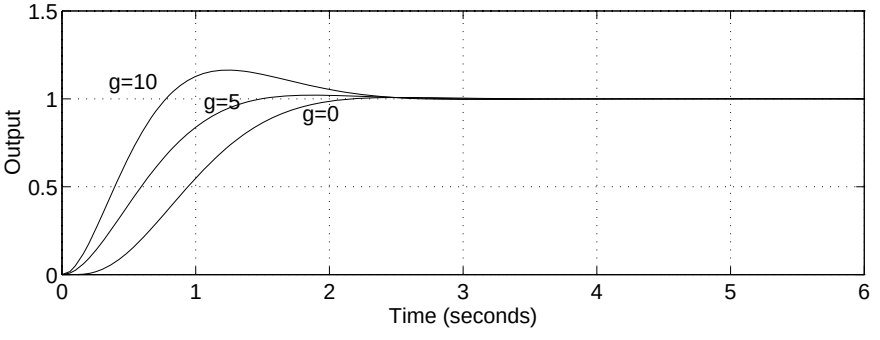


Figure 8.21 Step responses of the system without disturbance for different values of the feedforward gain g.

Now suppose that $\omega(t)$ is a step of height 5 that occurs at $t = 2.5$ seconds. Fig. 8.22 shows the input and output of the plant. Note that the plant output returns to the reference value of 1 after the onset of the disturbance at $t = 2.5$. This is accomplished when the output of the tracking system (shown with a dashed line) takes large negative values to cancel the influence of the step disturbance. After about 4.5 seconds, the tracking system output equals -5 which exactly cancels the disturbance $\omega(t) = 5$, producing an input of zero to the plant.

The loop transfer function for this tracking system is calculated using the results of Problem 4. A portion of the Nyquist plot is shown in Fig. 8.23 and the stability

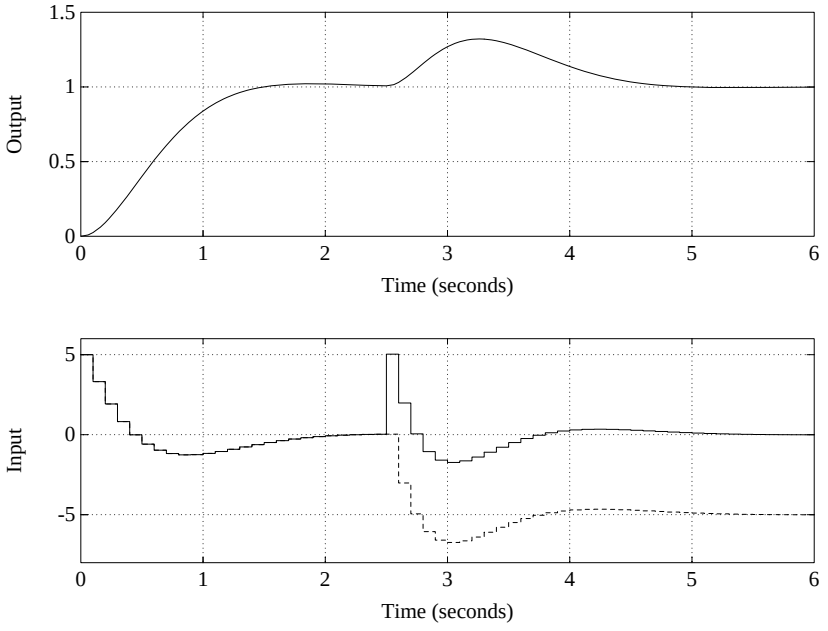


Figure 8.22 Input and output of the plant for Example 8.5 with a disturbance $\omega(t) = 5$ occurring at the input to the plant at $t = 2.5$ seconds. The dashed line shows the output of the tracking system.

margins are:

$$\text{GM} = 11.4 \text{ dB}$$

$$\text{PM} = 49^\circ$$

■

EXAMPLE 8.6

In this example we consider a head-positioning system for a computer floppy disk drive. The data tracks on a floppy disk are circular, but they are not exactly centered on the axis of rotation of the disk. In order for the read head to remain centered over the data track, the control system must track a sinusoid whose frequency is equal to the disk rotation frequency.

An experimental disk drive system is described in [88]. The read head is positioned using a 2-phase linear stepper motor. The following transfer function was experimentally derived to model the relationship between demanded and actual position

$$G(s) = \frac{118,438}{s^2 + 816.82s + 118,438}.$$

The paper [88] describes a digital control system with a sampling frequency of 160 Hz, and the microprocessor used to implement the digital control requires 0.5 msec to perform the calculations. The disk rotation frequency is 300 rpm, or 5 Hz. In this

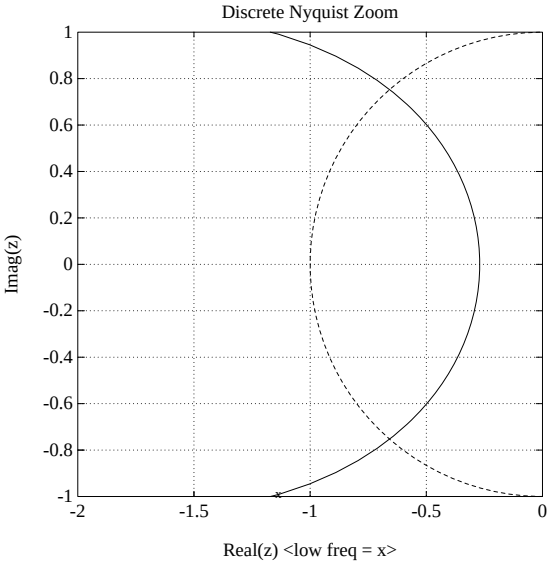


Figure 8.23 Nyquist plot for the tracking system of Example 8.5. The dashed line is the unit circle.

example, we design a digital tracking system which can follow a 5 Hz sinusoid. We will use the sampling interval $T = 1/160 = 0.00625$.

The first step is to calculate the ZOH equivalent of the plant including the computational time delay. Using results from Chapter 4 we get

$$\mathbf{\Phi} = \begin{bmatrix} -0.1038 & 0.0007 & 0.0404 \\ -77.5961 & 0.4314 & 26.7237 \\ 0 & 0 & 0 \end{bmatrix}, \ \mathbf{\Gamma} = \begin{bmatrix} 0.5283 \\ 515.3535 \\ 1.0000 \end{bmatrix}, \ \mathbf{C} = [1 \quad 0 \quad 0]$$

where one additional state variable is used to model the time delay of 0.0005 seconds. In this example we assume that all the state variables are measured. In subsection 8.4.2 following this example we show how observers can be used in tracking systems when only the output can be measured. It turns out for disk drive systems that the output y is not measured, but only the tracking error $r - y$. In subsection 8.4.3 we show how tracking systems can be modified when only the tracking error is measured. The refinements in the next two subsections are all based upon the design in this example.

Step 1

In order to generate a 5 Hz sinusoid, the matrix $\mathbf{A}_r$ must have eigenvalues at $\pm j2\pi5$. We map these poles to the z-plane and multiply out to obtain the polynomial $\delta(z)$ as follows

$$\begin{aligned} \delta(z) &= (z - e^{j10\pi(.00625)})(z - e^{j10\pi(.00625)}) \\ &= z^2 - 1.9616z + 1. \end{aligned}$$

Thus the additional dynamics are (see (8.48))

$$\mathbf{\Phi}_a = \begin{bmatrix} 1.9616 & 1 \\ -1 & 0 \end{bmatrix}, \ \mathbf{\Gamma}_a = \begin{bmatrix} 1.9616 \\ -1 \end{bmatrix}.$$

Step 2

The design model, which consists of the cascade of the plant and the additional dynamics is shown below (see (8.49))

$$\mathbf{\Phi}_d = \begin{bmatrix} -0.1038 & 0.0007 & 0.0404 & 0 & 0 \\ -77.5961 & 0.4314 & 26.7237 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1.9616 & 0 & 0 & 1.9616 & 1.0000 \\ -1.0000 & 0 & 0 & -1.0000 & 0 \end{bmatrix}, \quad \mathbf{\Gamma}_d = \begin{bmatrix} 0.5283 \\ 515.3535 \\ 1.0000 \\ 0 \\ 0 \end{bmatrix}.$$

The design model has 5 poles which must be placed using state feedback. Three closed-loop poles will be chosen as the roots of a Bessel polynomial, and the remaining two roots will simply add damping to the poles due to the sinusoid. The damping can be obtained from the root of the first-order Bessel polynomial. The desired settling time is 20 msec, or $T_S = 0.020$ seconds. Thus the closed-loop pole locations in the s-plane are

$$\begin{aligned} s_{1,2,3} &= -5.0093/T_S,\ (-3.9668 \pm j3.7845)/T_S \\ &= -250.47,\ -198.34 \pm -189.22 \\ s_{4,5} &= -4.62/T_S \pm j10\pi \\ &= -231 \pm j31.42 \end{aligned}$$

These poles are mapped into the z-plane using the ZOH pole mapping formula to obtain

$$z_{1,2,3} = 0.2090,\ 0.1096 \pm 0.2680$$

$$z_{4,5} = 0.2315 \pm 0.046.$$

The feedback vector $\mathbf{K}$ that places the eigenvalues of $(\mathbf{\Phi}_d - \mathbf{\Gamma}_d\mathbf{K})$ at the locations given above is calculated to be

$$\mathbf{K} = \begin{bmatrix} 1.3249 & 0.0015 & -0.0655 & 1.8050 & 1.2715 \end{bmatrix}.$$

This vector is partitioned into the first three ($n = 3$) and last two ($s = 2$) elements as follows

$$\mathbf{K}_1 = \begin{bmatrix} 1.3249 & 0.0015 & -0.0655 \end{bmatrix}, \quad \mathbf{K}_2 = \begin{bmatrix} 1.8050 & 1.2715 \end{bmatrix}.$$

The response of the system show in Fig. 8.20 (with g=0) to a 5 Hz reference sinusoid is shown in Fig. 8.24.

The Nyquist plot of the loop transfer function is shown in Fig. 8.25 and the stability margins are:

$$\text{GM} = 5.2 \text{ dB}$$

$$\text{PM} = 34^\circ.$$

■

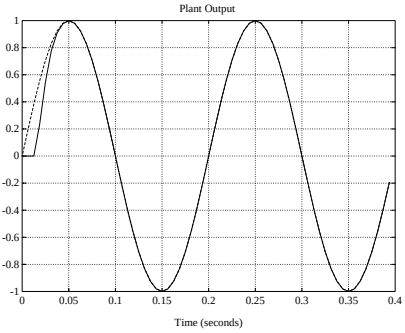


Figure 8.24 Response of the disk-drive tracking system of Example 8.6 to a 5 Hz reference sinusoid. The dashed line is the reference sinusoid.

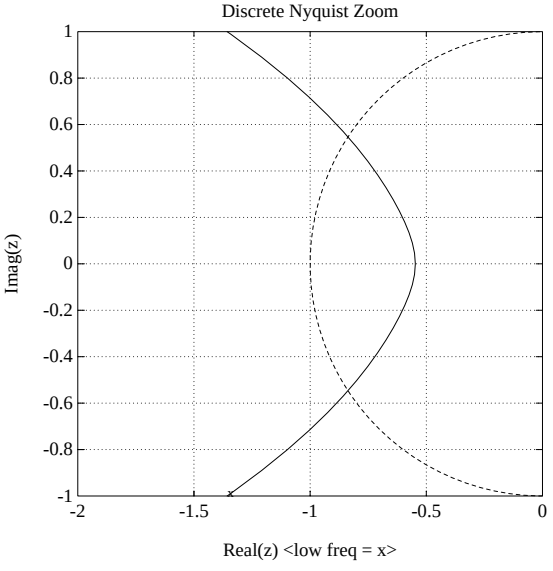


Figure 8.25 Nyquist plot for the tracking system of Example 8.6. The dashed line is the unit circle.

8.4.2 Using Observers in Tracking Systems

The tracking systems designed in the previous subsection assumed that all of the state variables could be measured, possibly subject to a disturbance. In this subsection we show how reduced-order observers can be used to estimate unmeasured state variables in a tracking system.

In Fig. 8.20 there is an "inner" feedback loop in which the measurement vector $\mathbf{y}_m$ is fed back through the gain vector $\mathbf{K}_1$. In that figure, $\mathbf{y}_m = \mathbf{x} + \mathbf{F}\omega$; in other words all state variables are measured possibly subject to disturbances. The plant is also possibly affected by disturbances as seen by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) + \mathbf{E}\omega(t).$$

It was shown in the previous subsection that additional dynamics could be constructed to eliminate the effect of the disturbance on steady-state tracking error. In calculating the feedback gains, the ZOH equivalent of the *undisturbed* plant was used

$$(\mathbf{\Phi}, \mathbf{\Gamma}) = \text{ ZOH equivalent of } (\mathbf{A}, \mathbf{B}).$$

Also the vector $\mathbf{y}_m$ was fed back *as if it were the undisturbed state vector* $\mathbf{x}$. The disturbance was "ignored" in calculating the ZOH of the plant and in the feedback through $-\mathbf{K}_1$ because its effects were taken into account by the additional dynamics. In a similar way we can "ignore" the disturbance when calculating a reduced-order observer. The state estimates produced by the reduced-order observer will have errors caused by the disturbance, but the steady-state effect of these errors will be eliminated by the additional dynamics.

In this subsection we assume that the measurement matrix $\mathbf{C}_m$ introduced in (8.42) is not equal to the identity matrix. We design a reduced-order observer for $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C}_m)$. As shown in Chapter 7, the design of this observer requires a transformation matrix $\mathbf{T}$, in general. With the appropriate choice of $\mathbf{T}$, a reduced-order observer $(\mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{L})$ can be designed using the procedure in Table 7.7. The output of the reduced-order observer, denoted $\hat{\hat{\mathbf{x}}}_2[k]$ must be combined with the measurements $\mathbf{y}_m[k]$ and transformed by $\mathbf{T}^{-1}$ to produce the estimated state vector, as shown in Fig. 8.26 (compare Fig. 7.23). The key

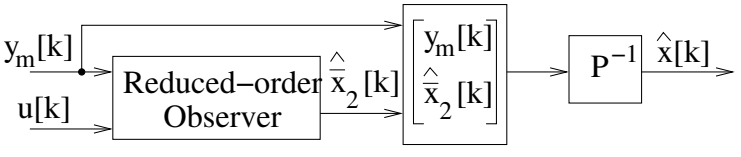


Figure 8.26 The measurements and reduced-order observer output are combined and transformed into an estimate of the state vector of the plant.

relationship in Fig. 8.26 is

$$\hat{\mathbf{x}}[k] = \mathbf{T}^{-1} \begin{bmatrix} \mathbf{y}_m[k] \\ \hat{\hat{\mathbf{x}}}_2[k] \end{bmatrix}. \tag{8.52}$$

If the reduced-order observer is to be used in a tracking system similar to that shown in Fig. 8.20, the estimated state vector will be fed back through the gain vector $-\mathbf{K}_1$. Suppose we call the resulting signal $\alpha[k]$; that is

$$\alpha[k] = -\mathbf{K}_1\hat{\mathbf{x}}[k].$$

Using (8.52) we have

$$\alpha[k] = -\mathbf{K}_1\mathbf{T}^{-1}\begin{bmatrix}\mathbf{y}_m[k]\\ \hat{\hat{\mathbf{x}}}_2[k]\end{bmatrix}. \tag{8.53}$$

If we define

$$\bar{\mathbf{K}}_1 = \mathbf{K}_1\mathbf{T}^{-1}$$

and partition $\bar{\mathbf{K}}_1$ into its first p and last $n-p$ elements,

$$\bar{\mathbf{K}}_1 = [\,\bar{\mathbf{K}}_{11} \quad \bar{\mathbf{K}}_{12}\,],$$

where p is the number of measured signals in $\mathbf{y}_m$, then (8.53) can be rewritten as

$$\alpha[k] = -\bar{\mathbf{K}}_{11}\mathbf{y}_m[k] - \bar{\mathbf{K}}_{12}\hat{\hat{\mathbf{x}}}_2[k]. \tag{8.54}$$

The resulting system is shown in Fig. 8.27. A procedure to design tracking systems which use an observer to estimate unmeasured state variables is given in Table 8.3.

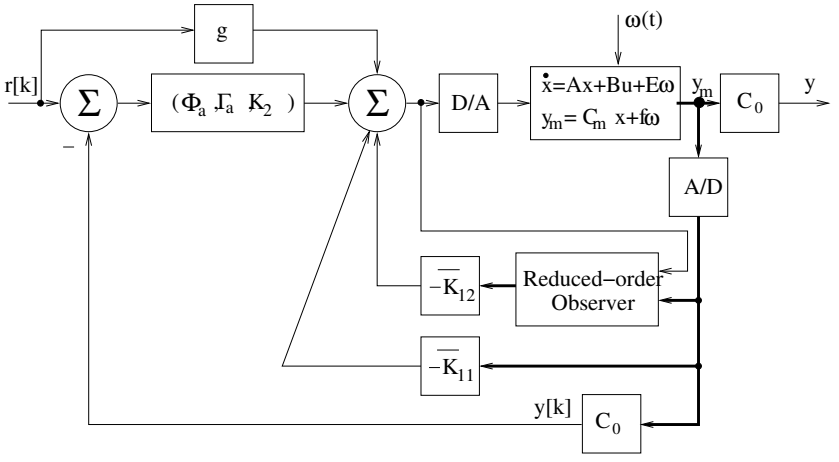


Figure 8.27 A tracking system that uses a reduced-order observer to estimate unmeasured state variables.

The robustness of the tracking system shown in Fig. 8.27 can be assessed by looking at the Nyquist plot of the loop transfer function obtained by breaking the loop at the input to the plant. This loop transfer function is calculated in Problem 8.

EXAMPLE 8.7

We continue the floppy disk drive example begun in Example 8.6 on page 382. Here we assume that only the output position $y = x_1$ is measured and design a reduced-order observer to estimate the values of x_2 and x_3. In an actual disk drive system the output y is not measured; only the tracking error $r - y$ is measured. In the next subsection we show that an "error driven" tracking system in which only $r - y$ is measured can be obtained directly from a tracking system in which y is measured.

In this example we follow the procedure given in Table 8.3 on page 388. Step 1 was done in Example 8.6. In Step 2 a reduced-order observer is designed for $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C}_m)$

1. Given an n^{th}-order plant model along with disturbance and measurement models, design a tracking system assuming all state variables are measured using the procedure given in Table 8.1. The ZOH equivalent of plant is $(\mathbf{\Phi}, \mathbf{\Gamma})$ and the measurement matrix is $\mathbf{C}_m$. The first n elements of the vector of feedback vector are denoted $\mathbf{K}_1$.
2. Design a reduced-order observer $(\mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{L})$ for $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C}_m)$ using the procedure given in Table 7.7 on page 343. The transformation matrix used in this design is denoted $\mathbf{T}$.
3. Let $\bar{\mathbf{K}}_1 = \mathbf{K}_1\mathbf{T}^{-1}$ and partition $\bar{\mathbf{K}}_1$ as
$$\bar{\mathbf{K}}_1 = [\,\bar{\mathbf{K}}_{11} \quad \bar{\mathbf{K}}_{12}\,].$$
4. The reduced-order observer and transformed feedback gains are used in the tracking system shown in Fig. 8.27.

Table 8.3 A procedure to design a tracking system which uses an observer to estimate unmeasured state variables.

where

$$\mathbf{\Phi} = \begin{bmatrix} -0.1038 & 0.0007 & 0.0404 \\ -77.5961 & 0.4314 & 26.7237 \\ 0 & 0 & 0 \end{bmatrix}, \ \mathbf{\Gamma} = \begin{bmatrix} 0.5283 \\ 515.3535 \\ 1.0000 \end{bmatrix}, \ \mathbf{C}_m = [\,1 \quad 0 \quad 0\,].$$

We use the procedure in Table 7.5 on page 325 to design the observer. In order to choose the observer pole locations we first compute the zeros of $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C}_m)$. These zeros can be computed by converting the state-space description to a transfer function and finding the roots of the numerator polynomial. The zeros are computed to be

$$-0.0007, \ -0.2834.$$

These zeros are inside the unit circle and so we could choose the observer poles to cancel them (see Section 7.3.3 on page 312 and the discussion preceding that Section). If we choose the observer poles to be -0.0007 and -0.2834 the reader can verify that perfect loop transfer function recovery is achieved, i.e. the loop transfer function of the observer-based tracking system is identical to that of the full state feedback tracking system designed in Example 8.6. However the zeros of $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C}_m)$ are on the negative real axis and so are "far" from the desired closed-loop pole locations. Thus we choose to design a "3-times faster" observer using Bessel polynomials. The full-state feedback tracking system was designed in Example 8.6 to have a settling time of 0.020 seconds. Hence the observer poles are chosen to have a settling time of $0.020/3 = 0.0067$ and mapped into the z-plane as follows

$$z_{1,2} = e^{s_{1,2}T/(.0067)}$$

where $s_{1,2}$ are the normalized Bessel roots $-4.0530 \pm j2.3400$ and $T = 0.00625$ is the sampling interval. The result is $z_{1,2} = -0.0131 \pm j0.0182$ and the polynomial for observer design is

$$p(z) = (z - z_1)(z - z_2) = z^2 + 0.0261z + 0.0005.$$

Using the procedure in Table 7.5 the reduced-order observer matrices are calculated to be

$$\mathbf{F} = \begin{bmatrix} 0.1698 & 10.6171 \\ -0.0032 & -0.1959 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 83.1863 \\ -1.7175 \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 304.4740 \\ -1.5655 \end{bmatrix}, \mathbf{L} = \begin{bmatrix} 399.1696 \\ 4.8561 \end{bmatrix}.$$

Since the measured state variable appears first (i.e. x_1) there is no need for a transformation matrix, or equivalently, the matrix $\mathbf{T}$ mentioned in Step 2 of Table 8.3 is just an identity matrix. The feedback gains $\bar{\mathbf{K}}_1$ equal the gains $\mathbf{K}_1$ calculated in Example 8.6. This gain vector is partitioned as follows

$$\bar{\mathbf{K}}_1 = [\,1.3249 \quad | \quad 0.0015 \quad -0.0655\,] = [\,\bar{\mathbf{K}}_{11} \quad | \quad \bar{\mathbf{K}}_{12}\,].$$

The observer matrices and feedback gains are used in the system shown in Fig. 8.27 to obtain an observer-based tracking system. The additional dynamics $(\mathbf{\Phi}_a, \mathbf{\Gamma}_a, \mathbf{K}_2)$ and the feedforward gain g are the same as in the full-state feedback example (8.6).

The reduced-order observer in this example is initialized with the state vector (see (7.51) on page 330)

$$\mathbf{z}[0] = -\mathbf{L}y[0]$$

which results in

$$\mathbf{z}[0] = \mathbf{0}, \text{ and } \hat{\mathbf{x}}_2[k+1] = \mathbf{0}.$$

Since the state variables of the plant are initially zero, the observer will estimate $\mathbf{x}_2[k]$ of the nominal plant without error, and the response of the observer-based tracking system to a 5 Hz sinusoid is identical to the response of the full-state feedback system shown in Fig. 8.24.

In order to find the stability margins for the observer-based tracking system, a portion of the Nyquist plot of the loop transfer function is shown in Fig. 8.28 (compare with the Nyquist plot of the full-state feedback system shown in Fig. 8.25). The stability margins of the observer-based tracking system are

$$\text{GM} = 7.3 \text{ dB}$$

$$\text{PM} = 34^\circ.$$

■

8.4.3 Error-driven Tracking Systems

In some applications the output y cannot be measured; only the tracking error $r - y$ is measured. This is the case for a disk-drive system, for example. The sensor in a disk-drive system does not measure the absolute position of the read/write head, but only the position relative to the center of a data track. Another example of an error-driven tracking systems include temperature control systems in which the thermostat only measures the difference between the desired and actual temperatures. Yet another example is a radar tracking system in which only the pointing error is measured (and not the absolute direction of the radar antenna).

The tracking system in Fig. 8.27 which uses a reduced-order observer can be modified to only depend on the tracking error. As shown in Fig. 8.27, an observer input is $\mathbf{y}_m$. Assume first that the measurement vector $\mathbf{y}_m$ is just the plant output (i.e. $\mathbf{C}_m = \mathbf{C}, \mathbf{C}_0 = 1$). In this case the tracking system uses measurements of y as an input to the observer. If y cannot be measured, consider what happens if we instead use $y - r$, the negative of the tracking error, as input to the observer. Then $-r$ acts like an *additive disturbance* on the input to the observer. This additive disturbance will be cancelled in the steady state because of the additional dynamics in the tracking system. The disturbance may degrade

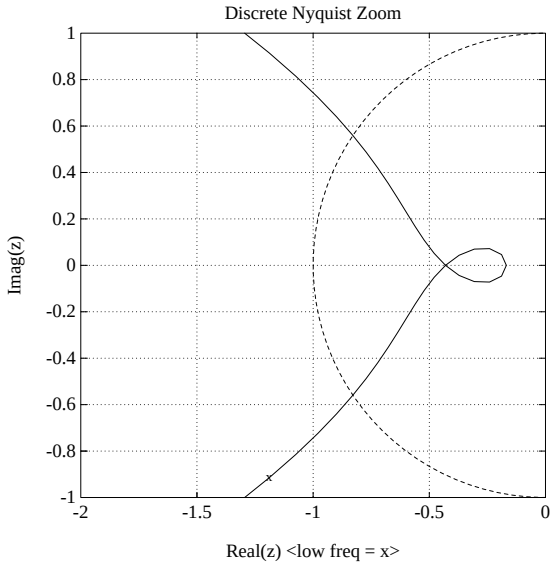


Figure 8.28 Nyquist plot for the tracking system of Example 8.7. The dashed line is the unit circle.

the transient behavior of the tracking system, but this is the price to pay for measuring only the tracking error.

Thus to obtain an error-driven tracking system we simply connect the observer to the negative of the tracking error, as shown in Fig. 8.29. It is easy to see that the loop transfer

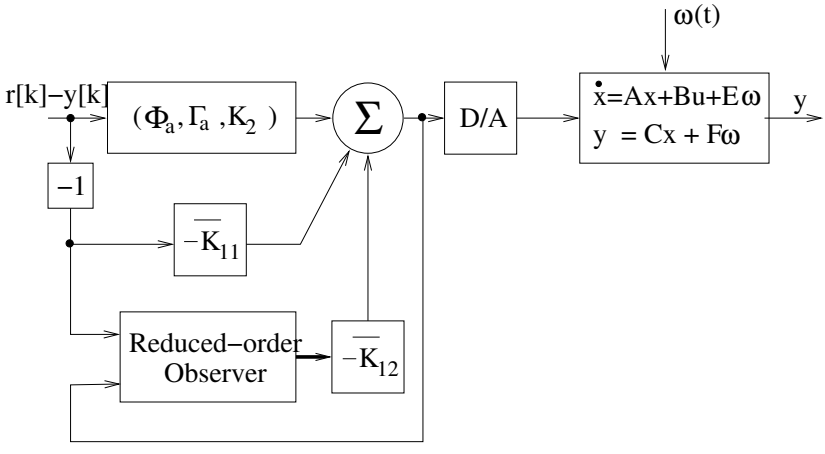


Figure 8.29 An error-driven tracking system.

function (with r set equal to zero) for the error-driven tracking system is identical to that of the observer-based tracking system considered in the previous subsection. This means that both tracking systems have the same stability margins.

EXAMPLE 8.8

We mentioned at the beginning of this section that the sensor in a disk-drive system does not measure the absolute position of the read/write head, but only the position relative to the center of a data track. Thus an error-driven tracking system must be used. This system can be obtained by taking the observer designed in Example 8.7 and driving it with the negative of the tracking error as shown in Fig. 8.29. The resulting error-driven tracking system will have the same stability margins as the observer-based tracking system. The response of the error-driven tracking system to a 5 Hz sinusoid is shown in Fig. 8.30. It can be seen that the transient response of this error-driven

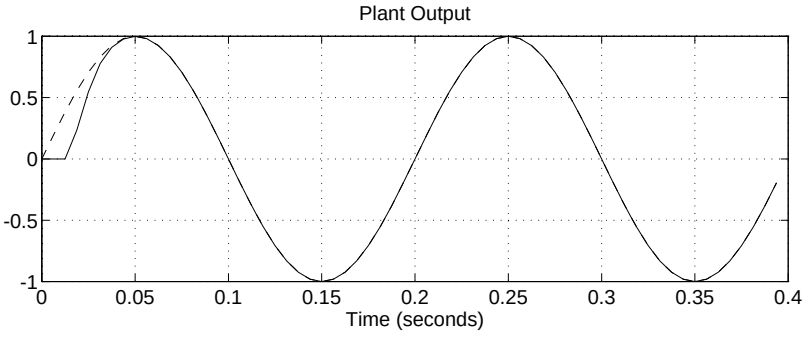


Figure 8.30 Response of the disk-drive tracking system of Example 8.8 to a 5 Hz reference sinusoid. The dashed line is the reference sinusoid.

tracking system is about the same as that of the tracking system of Example 8.6. ■

8.4.4 Robust Regulators

The goal of a regulator is to drive all of the state variables of a system to zero. The regulators designed in Chapters 6 and 7 used state feedback with or without observers to perform regulation. The regulators designed in those chapters were effective in dealing with *transient disturbances*. Such disturbances were modeled with a non-zero initial condition vector. A non-zero initial condition vector models a disturbance which occurs at $t = 0$ only, and a disturbance which occurs only at a single point in time is called a *transient disturbance*. An example of a transient disturbance for the cart/pendulum system was given in Example 7.9 on page 338 where it was argued that a tap on the pendulum could be modeled as a non-zero initial angular velocity of the pendulum. Note that there is nothing special about $t = 0$ in this context. If the pendulum is tapped at $t = 5$, for example, this is still a transient disturbance, and the state-feedback regulator will drive all the state variables to zero. The important point is that a state-feedback regulator with or without an observer is effective against transient disturbances.

Suppose however, that in addition to transient disturbances, there is a *persistent disturbance* (a disturbance that does not go to zero) acting on the system. What will happen to the performance of a state-feedback regulator? The answer is that a persistent disturbance will cause a non-zero steady state error (some of the state variables will not equal zero). An example of this fact will be given shortly.

One way to obtain zero steady-state error in spite of a persistent disturbance is to regard the regulation problem as a tracking problem in which the reference input is zero. Then a tracking system which rejects persistent disturbances can be designed. This tracking

system can be thought of as a *robust regulator* because the reference input is zero. For a single-input system, a robust regulator can drive a single variable (the output) to zero in spite of a persistent disturbance.

EXAMPLE 8.9

We consider again the cart/pendulum system. In Example 6.9 state feedback gains were calculated for this system, and in Example 7.9, an observer-based regulator was designed which was effective against transient disturbances. In this example we will see that a full-state feedback regulator is *not* effective against a constant disturbance, but that a robust regulator can be designed to handle this case.

The constant disturbance is introduced by tilting the track at an angle α as shown in Fig. 8.31. We assume that the angle α is unknown when the regulator is designed, and

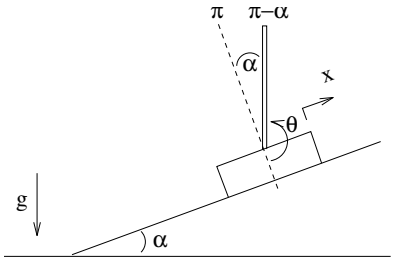


Figure 8.31 The pendulum/cart system with the track tilted at an angle α

that the variable we want to robustly regulate is motor position (which is proportional to cart position). Note that the incline of the track does not affect the cart dynamics since the velocity of the motor (track) is regulated by an analog control loop – see the discussion in Chapter 3.

When $\alpha = 0$ and the dynamics of the inverted pendulum are linearized (see page 143), the first state variable is defined to be $x_1 = \theta - \pi$, where θ is the angle between the gravity vector and the pendulum (see Fig. 3.25). When the track is tilted by an angle α as shown in Fig. 8.31, the first state variable is still the angle between the gravity vector and the pendulum, minus π, or

$$x_1 = \theta + \alpha - \pi.$$

However, the sensor which measures the angular position of the pendulum is mounted on the cart. When the track is tilted, this sensor measures $\theta - \pi$. Thus the first element of the measurement vector $\mathbf{y}_m$ is

$$\mathbf{y}_{m1} = \theta - \pi.$$

The previous two equations show that

$$\mathbf{y}_{m1} = x_1 - \alpha.$$

In other words, the measured angle has a constant disturbance equal to $-\alpha$. The regulators designed in this example will use full-state feedback with a disturbance of $-\alpha$ on the first state variable. The second, third, and fourth elements of the measurement vector are pendulum velocity, motor position, and motor velocity, respectively.

In the design of the robust regulator using full-state feedback, the measurement, disturbance, and output matrices from equations 8.39–8.42 are

$$\mathbf{C}_m = \mathbf{I}$$

$$\mathbf{F}_m = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{F} = 0$$

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

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

and the disturbance is $\omega(t) = -\alpha$. We will design a robust regulator which is effective against this constant disturbance.

Before designing a robust regulator we first verify that tilting the track will cause a steady-state error in motor position when a non-robust regulator is used. With the track at angle α and the pendulum balanced (i.e. aligned with the gravity vector) the steady-state measured angular position of the pendulum will be

$$\bar{\mathbf{y}}_{m1} = -\alpha.$$

In this example, a bar over a variable indicates a steady-state value. The steady-state values of the motor and pendulum velocities will be zero:

$$\bar{\mathbf{y}}_{m2} = 0\,, \quad \bar{\mathbf{y}}_{m4} = 0.$$

The cart will be motionless only if the input to the motor is zero. For full-state feedback, the input to the motor is set to zero as follows

$$-\mathbf{K}\bar{\mathbf{y}}_m = 0$$

and combining the previous three equations yields

$$-K_1(\alpha) - K_3(\bar{\mathbf{y}}_{m3}) = 0$$

which can be solved for steady-state motor position as follows

$$\bar{\mathbf{y}}_{m3} = \frac{K_1\alpha}{K_3}.$$

For the full-state feedback regulator designed in Example 6.9 the ratio of $K_1/K_3 \approx -810$, and so we expect a steady-state motor position of -810α when the track is at an angle α.

The above analysis is verified in the following simulation. All state variables as well as the angle α are initialized to zero. The value of α is increased linearly from 0 to 0.5236 radians (30°) over a 2 second interval, and remains constant for $t \geq 2$. Fig. 8.32 shows x_1 and x_3 as a function of time. It can be seen that the pendulum remains balanced, and that the steady-state value of motor position is indeed and $-810(0.5236) \approx -425$ radians as predicted by the analysis above.

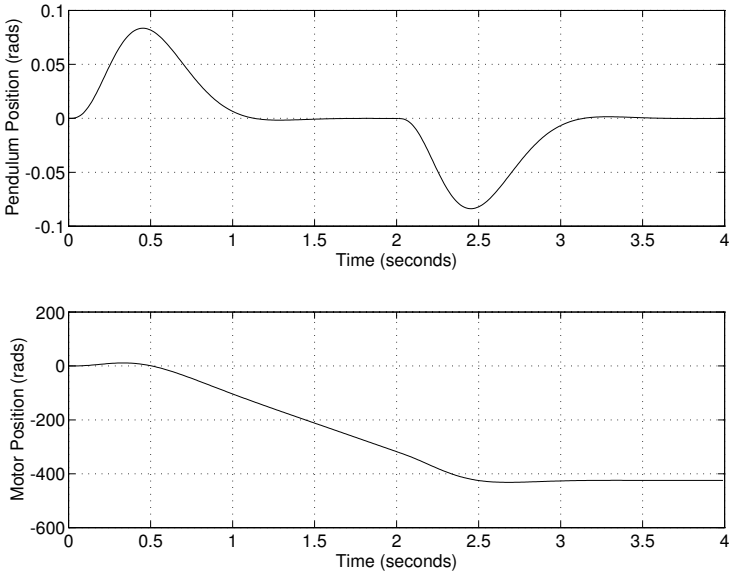


Figure 8.32 Pendulum position (x_1) and motor position (x_3)for the full-state feedback, nonrobust regulator of Example 8.9. The tilt of the track is increased linearly for the first two seconds.

We now design a robust regulator which employs full-state feedback using the procedure given in Table 8.1 on page 377. The feedback gains are computed for a design model which includes the additional dynamics required to reject constant disturbances. The additional dynamics are simply

$$\mathbf{\Phi}_a = 1 \ , \ \mathbf{\Gamma}_a = 1$$

and the design model is

$$\mathbf{\Phi}_d = \begin{bmatrix} \mathbf{\Phi} & \mathbf{0} \\ \mathbf{\Gamma}_a\mathbf{C} & \mathbf{\Phi}_a \end{bmatrix} \ , \ \mathbf{\Gamma}_d = \begin{bmatrix} \mathbf{\Gamma} \\ 0 \end{bmatrix} .$$

The closed-loop poles are chosen to be the roots of a 5-th order Bessel polynomial scaled to have a settling time of 0.95 seconds and mapped into the z-plane for a sampling interval of 0.01 seconds. The resulting vector of feedback gains is

$$\mathbf{K} = \begin{bmatrix} 70.6877 & 14.8661 & -0.1867 & -0.0690 & -0.0034 \end{bmatrix}$$

where the first 4 gains are used for $\mathbf{K}_1$ and the last gain is $\mathbf{K}_2$. A robust regulator is implemented as shown in Fig. 8.20 with feedforward gain $g = 0$ and reference input $r[k] = 0$.

As in the previous simulation, all state variables as well as the angle α are initialized to zero. The value of α is increased linearly from 0 to 0.5236 radians (30°) over a 2 second interval, and remains constant for $t \geq 2$. Since the robust regulator was designed for constant disturbances, there will be some error in motor position while the track is being tilted. However once the track reaches the constant angle of 0.5236 radians, the robust regulator will bring the motor position back to zero. Fig. 8.33 shows x_1 and x_3 as a function of time. It can be seen that the pendulum remains

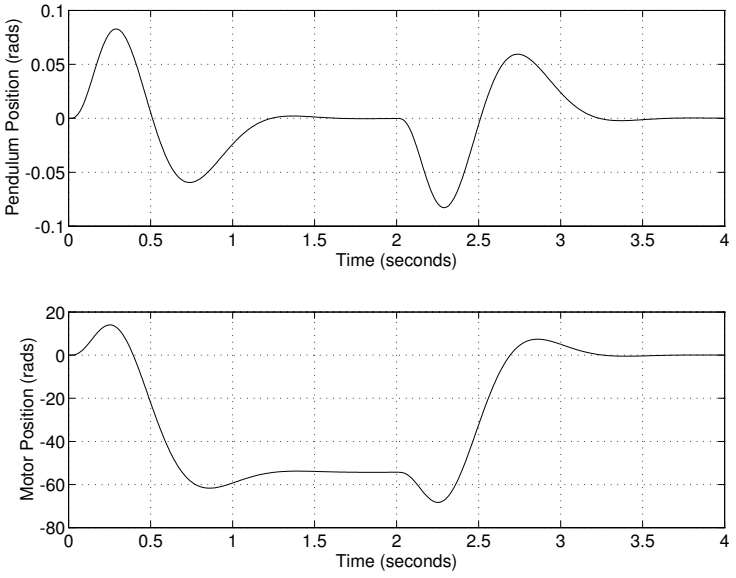


Figure 8.33 Pendulum position (x_1) and motor position (x_3)for the full-state feedback, robust regulator of Example 8.9. The tilt of the track is increased linearly for the first two seconds.

balanced, and that the motor position returns to zero shortly after the tilt of the track becomes constant.

The stability margins for the robust regulator can be calculated using the results of Problem 4 at the end of this chapter. This regulator has upper and lower gain margins

$$-4.38 \leq \text{ GM } \leq 33.1 \text{ dB}$$

and a phase margin of 28°. These margins are comparable to those of the non-robust regulator reported in Example 6.9.

■

8.5 Saturation Nonlinearities and Integrator Windup

In this section we show a simple way to deal with *saturation nonlinearities* in tracking systems which integrate the error signal. A saturation nonlinearity exists whenever the magnitude of signal is bounded. For instance, there is a maximum voltage that can be applied to a dc motor to produce a linear output. In a hydraulic system, flow has a maximum value when a valve is fully opened.

Digital control systems always have a saturation nonlinearity at the input to the plant due to the finite range of the D/A converter. For example, a D/A converter may have an output range of -5 to 5 volts. If the D/A converter is followed by an (analog) amplifier with a gain of 2, the input to the plant can now range from -10 to 10, but it is still bounded.

The presence of a saturation nonlinearity can have a deleterious effect on any control system, but the effect is particularly bad on a tracking system which contains an integrator. The reason is that the integrator accumulates (adds up) the tracking errors at each sampling instant in order to produce an input to the plant which will cause a decrease in the tracking

error. However, if the input to the plant is already saturated, it does no good for the tracking system to command a larger input signal. In fact, harm is done by allowing the integrator to keep accumulating the tracking error when the plant input is saturated. The problem is that if the saturation occurs for several sampling instants, then the *integrated* tracking error will become large even when the *current* tracking error becomes small. Since the input to the plant is determined in part by the integrated tracking error (i.e. the output of the integrator in Fig. 8.34), large input signals will continue to be sent to the pant resulting in excessive overshoot.

The solution to the problem described above is simple: *turn off the integrator whenever the plant input is saturated*. By "turn off" the integrator we mean stop accumulating the tracking error. An integrator which accumulates tracking error when the plant input is not saturated and does not accumulate error otherwise will be referred to as a "smart integrator." It is fairly easy to implement smart integrators in a digital tracking system as we now show.

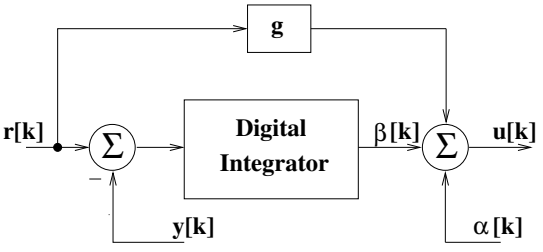


Figure 8.34 A block diagram of the signals associated with a "smart" digital integrator.

Suppose the saturation nonlinearity is described by the constraint $|u| < S$. The operation of the smart integrator is defined as follows. At time k compute a tentative value γ for $u[k]$ based on values of $r[k]$, $\beta[k]$, and $\alpha[k]$ (see Fig. 8.34), i.e.

$$\gamma = gr[k] + \beta[k] + \alpha[k].$$

If $|\gamma| < S$, set $u[k] = \gamma$ and also let the integrator accumulate the tracking error at time k. However, if $|\gamma| > S$, set $u[k] = S \cdot \text{sgn}(\gamma)$ and do not accumulate the tracking error at time k. A complete description of the operation of a smart integrator is shown in Table 8.4.

If an observer is used to estimate some of the state variables, the observer should use the signal $u[k]$ generated by the smart integrator. This signal satisfies the constraint $|u[k]| < S$. Since the observer and the plant receive the same $u[k]$, the observer will continue to generate accurate state estimates in the presence of a saturation nonlinearity.

EXAMPLE 8.10

In Example 8.5 on page 380 we designed a tracking system for a double-integrator plant. A block diagram of the tracking system is given in Fig. 8.20. In this example we assume that the D/A converter output is limited to 5 volts in magnitude. The integrator that is required for the additional dynamics in Example 8.5 will be implemented as a smart integrator in this example, and compared with the results of not using a smart integrator. The values of $(\mathbf{\Phi}_a, \mathbf{\Gamma}_a, \mathbf{K}_2)$, and $\mathbf{K}_1$ are all taken from Example 8.5, and we use the feedforward gain of $g = 5$. The response to a step input $r[k] = 3$ is simulated and the input and output of the plant are shown in Fig. 8.35. The system

1. $\beta[k] = \mathbf{K}_2 \mathbf{x}_a[k]$.
2. Compute $\gamma = gr[k] + \beta[k] + \alpha[k]$.
   If $|\gamma| < S$ then

$$u[k] = \gamma$$

$$\mathbf{x}_a[k+1] = \mathbf{x}_a[k] + r[k] - y[k]$$

3. else

$$u[k] = \gamma$$

$$\mathbf{x}_a[k+1] = \mathbf{x}_a[k]$$

   end

Table 8.4 Description of a smart integrator.

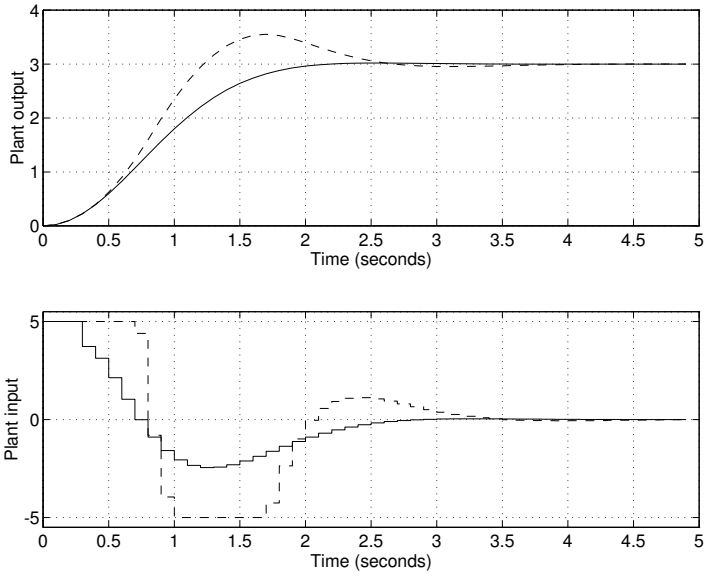


Figure 8.35 Step response and plant input for Example 8.10. The solid lines are for the system with the smart integrator, and the dashed lines are for the system without the smart integrator

with the smart integrator has essentially zero overshoot, while the system without the smart integrator has substantial overshoot. The plant input for the system with the smart integrator is saturated at $u[k] = 5$ only for the first three samples. Thus the smart integrator does not accumulate the error for the first three samples. This can be seen from Fig. 8.36 which shows the state variable $x_a[k]$ of the additional dynamics. For the system with the smart integrator, $x_a[k]$ is zero for the first three samples.

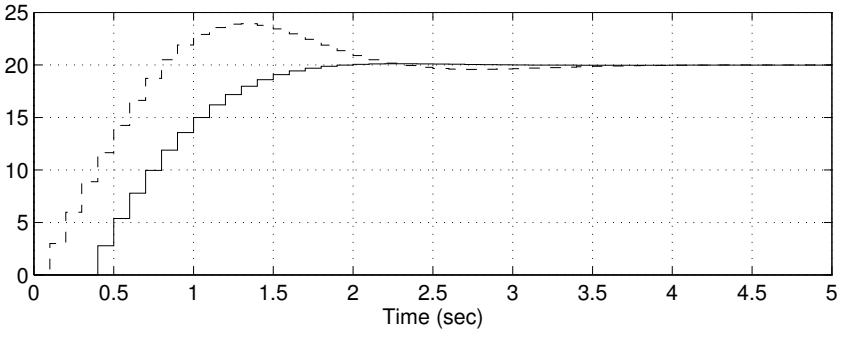


Figure 8.36 State variable for the additional dynamics of the tracking system in Example 8.10 with (solid line) and without (dashed line) the smart integrator

8.6 Chapter Summary

In this chapter we showed how to design digital tracking systems for single-input, single-output plants. We showed that additional dynamics must be used to achieve zero steady-state tracking error. The additional dynamics are implemented as part of the compensator.

A digital tracking system uses a feedback vector $\mathbf{K}$ which is calculated for the cascade combination of the ZOH of the plant and the additional dynamics. Tracking systems can use full-state feedback, or an observer can be used to estimate state variables that are not measured. In either case, the stability margins of a tracking system are found from the appropriate loop transfer function.

In some applications the plant output is not measured directly, but only the tracking error $r - y$ is measured. We showed that an observer-based tracking system can be used in this case to obtain an error-driven tracking system.

In addition to their use in tracking systems, Additional dynamics are needed for regulating plants which are subject to persistent disturbances. A regulator with additional dynamics can drive a single variable to zero in spite of a persistent disturbance.

8.7 Problems

1. Consider the hydraulic positioning system introduced in Chapter 6, problem 9. Observers for this system were considered in Chapter 7, problem 4.

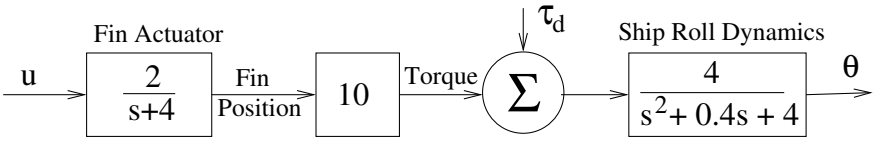
   (a) Design a tracking system which will have zero steady-state error for a step input. The desired settling time is $T_S = 0.35$ seconds. Assume full state feedback.

(b) Repeat part (a) assuming only the position of the piston can be measured.

(c) Repeat part (a) assuming x_1, x_2, and x_4 can be measured.

(d) Repeat (a), (b), and (c) when the reference input is a ramp.

2. Problem 10 in Chapter 6 considered the design of a full state feedback regulator to control the roll angle of a ship. However in that problem, disturbances due to waves were not considered. A block diagram which includes the disturbance is shown below.



A state-space model for this system is

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ -4 & -.4 & 40 \\ 0 & 0 & -4 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} u + \begin{bmatrix} 0 \\ 4 \\ 0 \end{bmatrix} \tau_d$$

where the state variables are defined as $x_1 = \theta$, $x_2 = \dot{\theta}$, and $x_3 =$ fin position. We assume that the disturbance torque τ_d caused by the waves can be described mathematically as follows

$$\tau_d = A_1 \sin(\omega_1 t) + A_2 \sin(\omega_2 t) + A_3 \sin(\omega_3 t)$$

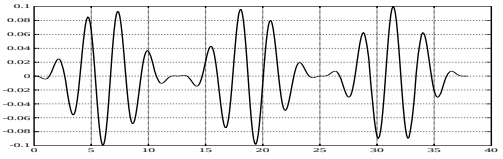
where the amplitudes A_i are functions of time, and the frequencies ω_i are in the range

$$1.85 \leq \omega_i \leq 2.85 \text{ rad/sec.}$$

For the purpose of simulation, we choose the amplitudes to be $A_1 = .025$, $A_2 = -.05$, $A_3 = .025$, and the frequencies

$$\omega_1 = 1.85\ ,\ \ \omega_2 = 2.35\ ,\ \ \omega_3 = 2.85.$$

A plot of the disturbance torque corresponding to this model is shown in the following figure.



(a) Temporarily ignore the disturbance and design a state feedback regulator to achieve a settling time of $T_S = 4.5$ seconds (this was done in problem 10). Set the initial state vector to zero, and perform a simulation to see how this regulator performs when the disturbance is present.

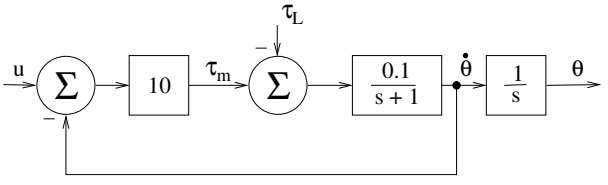
(b) Design a robust regulator to mitigate the effects of the persistent disturbance. The regulator should be effective against disturbance torques in the frequency range $1.85 \leq \omega \leq 2.85$. One way to do this is to define the pole locations for the additional dynamics of the robust regulator to be

$$-.2 \pm j2.35 \text{ in the } s\text{-plane.}$$

Another possibility is to use two sets of complex-conjugate pole locations in the frequency range of interest. For example,

$$-.2 \pm j2.18\,, \quad -.2 \pm j2.52 \text{ in the } s\text{-plane.}$$

3. Consider the large turntable system described in Problem 11 on page 287. A block diagram of the system is shown below

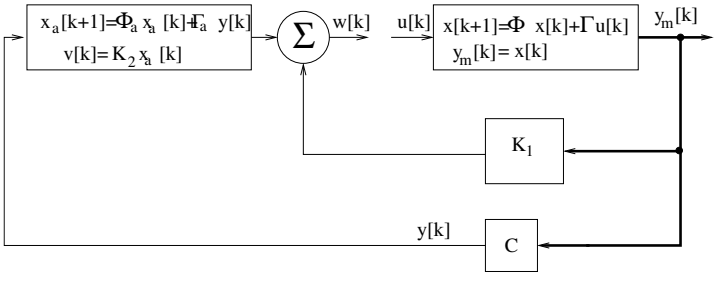


The state variables are $x_1 = \theta$ and $x_2 = \dot{\theta}$. τ_m represents the torque produced by the motor, and τ_L represents the torque required to drive an external load.

(a) Design a tracking system for a step input which achieves a settling time of 1 second. Obtain a step response for an input of 0.34 radians when $\tau_L = 0$. Also plot the control input $u(t)$.

(b) Repeat the step response and input plots when $\tau_L = 1$.

4. In order to find the stability margins for a tracking system which uses full-state feedback, the loop transfer function must be calculated. The following figure shows a full-state feedback tracking system with the loop broken at the input to the plant. (The loop would be closed by setting $u[k] = -w[k]$.)

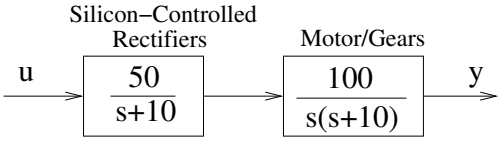


Find a state-space description for the composite system with input $u[k]$ and output $w[k]$. Use the composite state vector

$$\begin{bmatrix} \mathbf{x}[k] \\ \mathbf{x}_a[k] \end{bmatrix}.$$

5. *Path-Controlled Turret Lathe* [Dorf]

A block diagram of a numerical path-controlled turret lathe is shown below



The state variables and output are defined as follows

$x_1 =$ angular position of motor shaft (radians)

$x_2 =$ angular velocity of motor shaft (radians/second)

$x_3 =$ output voltage of SCR (volts)

$y =$ position of tool slide (millimeters)

The corresponding state-space description is

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -10 & 1000 \\ 0 & 0 & -10 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 50 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0.1 & 0 & 0 \end{bmatrix} \mathbf{x}.$$

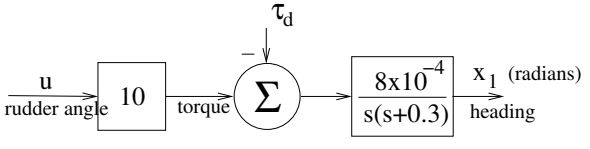
The purpose of this problem is to design tracking systems with a settling time of 0.1 seconds which can track a ramp input with zero steady-state error. Plot the step response of each system, and also plot $x_2(t)$ in response to a ramp input.

(a) Design a tracking system which uses full state feedback.

(b) Design a tracking system assuming only x_1 and x_2 can be measured.

(c) Design a tracking system assuming only x_1 can be measured.

6. A simplified model for a ship steering system is shown below



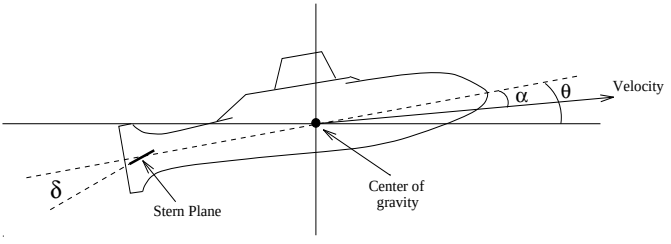
where τ_d represents a disturbance torque caused by ocean currents. We have the following state-space model

$$\dot{x}_1 = -0.3x_1 + x_2$$

$$\dot{x}_2 = 8 \times 10^{-3}u - 8 \times 10^{-}4\tau_d.$$

Design a control system which can track step commands for the ship's heading as well as reject constant disturbances. Assume that only x_1 can be measured. The ship dynamics correspond to a single pole with settling time $4.5/0.3 = 15$ seconds. Design the closed-loop system to have the same settling time.

7. The dynamics of a submarine in the vertical plane are described using the variables shown in the following picture



The angle θ is the *pitch angle*, α is the *angle of attack*, and δ is the *deflection of the stern plane*. If θ is small and the velocity of the submarine is constant, the motion of the submarine can be described by a linear state-space model. The state variables are

$$x_1 = \theta$$

$$x_2 = \dot{\theta}$$

$$x_3 = \alpha$$

$$x_4 = \delta$$

where all angles are in radians. A state-space model which includes the effects of the stern-plane actuator is

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -.0191 & -.364 & .364 & -.1250 \\ 0 & .3342 & -1.092 & .2255 \\ 0 & 0 & 0 & -2.5 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2.5 \end{bmatrix} u,$$

$$y = [\,1 \quad 0 \quad 0 \quad 0\,]\, x$$

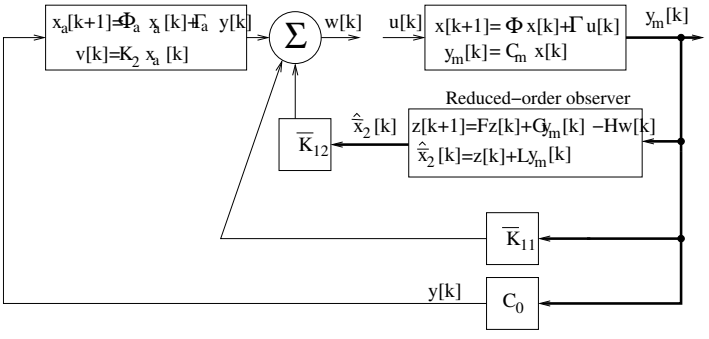
where the input is the commanded stern-plane angle.

Design a tracking system which will give zero steady-state error to a step input in pitch. The desired settling time is 5 seconds. Obtain a step response for a 5° step input.

(a) Assume x_1 and x_2 can be measured.

(b) Assume only x_1 can be measured.

8. In order to find the stability margins for a tracking system which uses full-state feedback, the loop transfer function must be calculated. The following figure shows a full-state feedback tracking system with the loop broken at the input to the plant. (The loop would be closed by setting $u[k] = -w[k]$.)



Find a state-space description for the composite system with input $u[k]$ and output $w[k]$. Use the composite state vector

$$\begin{bmatrix} \mathbf{x}[k] \\ \mathbf{z}[k] \\ \mathbf{x}_a[k] \end{bmatrix}.$$

9. Problem 12 on page 288 considered the regulation of an antenna system subject to a transient wind disturbance at time zero.

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & -1.5 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0.2 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0.2 \end{bmatrix} T_d$$

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

(a) Design a tracking system which will reject a constant wind disturbance and track a step input.

(b) Suppose that the input saturates at a value of 2. Use "smart integrators" which take this saturation into account. Plot step responses for steps of 0.17 radians and 1 radian.

(c) Design a tracking system which will have zero steady-state error for a ramp input.

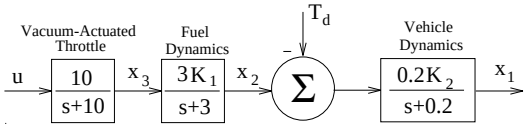
10. A state-space model for a heat exchanger is given below

$$\dot{\mathbf{x}} = \begin{bmatrix} -0.04 & 0.04 \\ 0 & -0.1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u(t-5)$$

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

Design a tracking system for a step input to achieve a settling time of 60 seconds. Let the sampling interval be $T_S = 2$ seconds. Assume that only x_1 is measured.

11. A block diagram of an automotive cruise-control system is shown below



The disturbance torque T_d is encountered when the car goes up a hill. Assume that the units of the variables are normalized so that all the conversion factors equal unity (i.e. $K_1 = K_2 = 1$). Then a state-space model is

$$\dot{\mathbf{x}} = \begin{bmatrix} -.2 & .2 & 0 \\ 0 & -3 & 3 \\ 0 & 0 & -10 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix} u - \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix} T_d$$

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

where

$x_1 =$ vehicle velocity

$x_2 =$ fuel flow

$x_3 =$ throttle position.

Design a tracking system which can follow a step command in desired velocity and reject a constant disturbance torque. The desired settling time is 2 seconds. Perform a simulation in which the reference input is a unit step in vehicle velocity, and the disturbance torque is a unit step starting at $t = 2$ seconds.

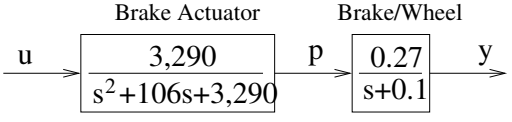
(a) Assume only vehicle velocity and fuel flow are measured.

(b) Assume only vehicle velocity and throttle position are measured.

(c) Assume only vehicle velocity is measured.

12. *Automobile Brake-Force Distribution Controller*

In this problem we consider an automobile braking system. When the brakes are applied, the total braking force is the sum of the forces developed by the front and rear brakes. The ideal distribution of force between front and rear brakes is not constant, but depends on such factors as vehicle loading and road inclination. One way of achieving ideal brake force distribution is to control the rear-brake pressure in such a way as to cause the rear-wheel speed to track the front-wheel speed [79]. A block diagram of an automobile braking system obtained from [79] is shown below:



where u is the commanded rear-brakeline pressure, p is the actual pressure, and y is the change in rear-wheel speed from its nominal value of 48 km/h. Note that the second block in the above diagram is obtained by linearizing the non-linear dynamics of the brake/wheel subsystem. We choose the state variables to be $x_1 = p$, $x_2 = \dot{p}$, and $x_3 = y$, and obtain the following state-space model for the system shown in the block diagram above:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ -3290 & -106 & 0 \\ 0.27 & 0 & -.1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 3290 \\ 0 \end{bmatrix} u$$

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

(a) Design a tracking system for a ramp input. The desired settling time is 0.2 seconds, and the ramp input is $r(t) = -24t$, $0 \leq t \leq 2$. The actual speed starts at the nominal value of 48 km/h, and decreases to 0 at $t = 2$ seconds. Assume all state variables can be measured.

(b) Find the stability margins for the tracking system designed in (a).

(c) Assume that the linearized model is inaccurate and that the transfer function from p to y is actually given by

$$\frac{0.4}{s + 0.2}.$$

Use the tracking system designed in part (a), but simulate the system with a new state-space model for the plant which incorporates the new transfer function from p to y.

(d) Assume that only y is measured and incorporate a reduced-order observer into the tracking system designed in (a).

13. *Piezoelectric Micro-positioning System*

A piezoelectric crystal is a material which, when subjected to mechanical strain (pressure) develops a voltage across the face of the crystal. The reverse effect also occurs in piezoelectric materials: an applied voltage results in mechanical strain which causes the face of the crystal to move. The magnitude of this motion is on the order of microns, and so piezoelectric crytals can be used as position actuators for ultra-precision machining [71]. A typical application of these devices is to produce metal mirrors for high-power lasers. A state-space description for a certain piezoelectric crystal is: [71]

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -5,970^2 & -441.7 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 35.64 \end{bmatrix} u$$

where x_1 is the displacement of the actuator in meters. The poles of this system are at $-220.8 \pm 5,966$, which are very lightly damped.

(a) Design two tracking systems for step inputs. Let the first tracking system use Bessel poles, and the second use "added damping" poles (see Example 6.10 in Chapter 6). The desired settling time for both tracking systems is 2 msec. Assume both state variables can be measured.

(b) Compare the stability margins of the two tracking systems from (a).

(c) Simulate the responses of the tracking systems to a step input of $1\ \mu m$. Let the simulation time be $4\ ms$.

(d) Assume only x_1 can be measured and design an observer to use with the tracking systems designed in (a). Compare the stability margins of the observer-based tracking systems with those obtained in (b). Simulate the responses of the observer-based tracking systems to a step input of $1\ \mu m$.

14. *Error-Driven Satellite Attitude Control*

The angular position (attitude) of a satellite can be described by a double integrator with the following state-space model (see Section 3.6.2)

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

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

where x_1 is the angular position in radians of the satellite, x_2 is the angular velocity, and u is the torque due to the thrusters. We wish to control the angular position of the satellite, but the only available measurement is the error $r - y$ between the desired position r and the actual position y. Design an error-driven tracking system which will have zero steady-state error for step inputs, and a settling time of 2 seconds. Find the stability margins of the tracking system, and simulate the response to a step input of 15°.

CHAPTER 9

MULTIVARIABLE SYSTEMS

The design of regulation and tracking systems presented in previous chapters has assumed that the plant had a single input and a single output (SISO plants). Even in the context of SISO plants, the use of multiple-input, multiple-output (MIMO) systems arose naturally in the design of observer systems. MIMO systems are also called *multivariable systems*. In this chapter, we extend the results developed in previous chapters for SISO plants to the case in which the plant is a multivariable system.

In previous chapters we saw that the design of regulation and tracking systems was based on calculating the unique feedback vector $\mathbf{K}$ that placed the closed-loop poles in desired locations. In this chapter we see that the feedback gains for a multivariable system form a matrix $\mathbf{K}$, and that there are an infinite number of different feedback matrices which yield a given set of closed-loop pole locations. We derive an algorithm to find the feedback matrix which optimizes a robustness criterion. This algorithm is used in the design of multivariable regulation and tracking systems.

9.1 Multivariable System Models

A continuous-time system with one input $u_1(t)$ and one output $y_1(t)$ has the usual state-space model

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}_1 u_1(t) \\ y_1(t) &= \mathbf{C}_1\mathbf{x}(t) + D_{11}u_1(t). \end{aligned} \tag{9.1}$$

Note that the number D_{11} is the feedthrough coefficient from the first input to the first output. If a second input $u_2(t)$ were added to the system described by the above equations, it would enter the state equation with its own input vector $\mathbf{B}_2$, in addition to the first input. The resulting state equation is

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}_1 u_1(t) + \mathbf{B}_2 u_2(t). \tag{9.2}$$

The second input could affect the first output with a feedthrough coefficient D_{12} giving the output equation

$$y_1(t) = \mathbf{C}_1\mathbf{x}(t) + D_{11}u_1(t) + D_{12}u_2(t). \tag{9.3}$$

If a second output $y_2(t)$ were defined for the system, it would have its own output vector $\mathbf{C}_2$. The feedthrough coefficients for the first and second inputs would be D_{21} and D_{22}, respectively

$$y_2(t) = \mathbf{C}_2\mathbf{x}(t) + D_{21}u_1(t) + D_{22}u_2(t). \tag{9.4}$$

For a system with p inputs and q outputs, the results described above generalize as follows. The state equation has a sum of input terms in which each input signal enters with its own input vector. The outputs are defined with their own output vectors and a sum of feedthrough coefficients from all the inputs. The result is

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \textstyle\sum_{j=1}^{p} \mathbf{B}_j u_j(t) \\ y_i(t) &= \mathbf{C}_i\mathbf{x}(t) + \textstyle\sum_{j=1}^{p} D_{ij}u_j(t)\ , \quad i = 1, \cdots, q. \end{aligned} \tag{9.5}$$

The above equation can be written in a more compact form by defining a vector of inputs

$$\mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_p(t) \end{bmatrix}$$

and a vector of outputs

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ \cdots \\ y_q(t) \end{bmatrix}.$$

With these definitions, the summations in (9.5) can be written in matrix form as follows

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{9.6}$$

where

$$\mathbf{B} = [\,\mathbf{B}_1 \quad \mathbf{B}_2 \quad \cdots \mathbf{B}_p\,], \tag{9.7}$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \vdots \\ \mathbf{C}_q \end{bmatrix}, \tag{9.8}$$

and

$$\mathbf{D} = \begin{bmatrix} D_{11} & D_{12} & \cdots & D_{1p} \\ D_{21} & D_{22} & \cdots & D_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ D_{q1} & D_{q2} & \cdots & D_{qp} \end{bmatrix}. \tag{9.9}$$

In light of the definitions of $\mathbf{u}(t)$ and $\mathbf{y}(t)$ and assuming the system is n^{th}-order , we must have the following dimensions for the matrices in (9.6): $\mathbf{A}_{n\times n}$, $\mathbf{B}_{n\times p}$, $\mathbf{C}_{q\times n}$, $\mathbf{D}_{q\times p}$. The jth column of the input matrix $\mathbf{B}$ is the input vector for the jth input of the system and the ith row of the output matrix $\mathbf{C}$ is an output vector for the ith output of the system. Finally, the ijth element of the feedthrough matrix $\mathbf{D}$ is the feedthrough coefficient from the jth input to the ith output.

9.1.1 Transfer Function Matrices

If a system has several inputs and several outputs, it is of interest to compute the transfer function from each input to each output. In particular, we can compute the transfer function from the jth input to the ith output by setting all inputs except $u_j(t)$ equal to zero. The state-space equations which describe the behavior of the system from the j-th input to the i-th output (with all other inputs set to zero) are

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x} + \mathbf{B}_j u_j(t)\\ y_i(t) &= \mathbf{C}_i\mathbf{x}(t) + D_{ij}u_j(t).\end{aligned} \tag{9.10}$$

The transfer function for this SISO model is obtained using the formula from Chapter 3

$$H_{ij}(s) = \mathbf{C}_i(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}_j + d_{ij}. \tag{9.11}$$

$H_{ij}(s)$ is the transfer function from the j-th input to the i-th output with all other inputs set to zero.

The transfer functions from every input to every output can be arranged into the following *transfer function matrix*

$$\begin{aligned}\mathbf{H}(s) &= \begin{bmatrix} H_{11}(s) & H_{12}(s) & \cdots & H_{1p}(s)\\ H_{21}(s) & H_{22}(s) & \cdots & H_{2p}(s)\\ \vdots & \vdots & \cdots & \vdots\\ H_{q1}(s) & H_{q2}(s) & \cdots H_{qp}(s) & \end{bmatrix}\\ &= \begin{bmatrix}\mathbf{C}_1\\ \vdots\\ \mathbf{C}_q\end{bmatrix}(s\mathbf{I}-\mathbf{A})^{-1}\begin{bmatrix}\mathbf{B}_1 & \cdots & \mathbf{B}_p\end{bmatrix} + \begin{bmatrix} D_{11} & \cdots & D_{1p}\\ \vdots & \vdots & \vdots\\ D_{q1} & \cdots & D_{qp}\end{bmatrix}\\ &= \mathbf{C}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B} + \mathbf{D}.\end{aligned} \tag{9.12}$$

From the above development, the ijth element of $\mathbf{H}(s)$ is the transfer function from the jth input to the ith output.

EXAMPLE 9.1

Consider the following multivariable state-space model for a 2-input 2-output system

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \begin{bmatrix}-1 & -2 & 0\\ -3 & -4 & 0\\ 0 & 0 & -5\end{bmatrix}\mathbf{x}(t) + \begin{bmatrix}0 & 1\\ 1 & 0\\ 0 & 0\end{bmatrix}\mathbf{u}(t)\\ \mathbf{y}(t) &= \begin{bmatrix}1 & 0 & 1\\ 0 & 0 & 1\end{bmatrix}\mathbf{x}(t) + \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\mathbf{u}(t).\end{aligned} \tag{9.13}$$

The transfer function matrix is computed as follows. Using the Fadeev algorithm, we compute

$$(s\mathbf{I}-\mathbf{A})^{-1}=\frac{S_1s^2+S_2s+S_3}{a(s)}$$

where

$$\mathbf{S}_1=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},\ \mathbf{S}_2=\begin{bmatrix}9&-2&0\\-3&6&0\\0&0&5\end{bmatrix},\ \mathbf{S}_3=\begin{bmatrix}20&-10&0\\-15&5&0\\0&0&-2\end{bmatrix}$$

and

$$a(s)=s^3+10s^2+232-10.$$

Using these results we can compute

$$\begin{aligned}\mathbf{C}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}&=\tfrac{1}{a(s)}\left\{\mathbf{C}\mathbf{S}_1\mathbf{B}s^2+\mathbf{C}\mathbf{S}_2\mathbf{B}s+\mathbf{C}\mathbf{S}_3\mathbf{B}\right\}\\&=\tfrac{1}{a(s)}\left\{\begin{bmatrix}0&1\\0&0\end{bmatrix}s^2+\begin{bmatrix}-2&9\\0&0\end{bmatrix}s+\begin{bmatrix}-10&20\\0&0\end{bmatrix}\right\}\\&=\tfrac{1}{a(s)}\begin{bmatrix}-2s-10&s^2+9s+20\\0&1\end{bmatrix}.\end{aligned}$$

Then

$$\begin{aligned}\mathbf{H}(s)&=\mathbf{C}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}+\mathbf{D}\\&=\begin{bmatrix}\frac{-2s-10}{a(s)}+1&\frac{s^2+9s+20}{a(s)}\\0&\frac{1}{a(s)+1}\end{bmatrix}.\end{aligned}$$

■

9.1.2 State Space Models From Differential Equations

A multivariable state-space model is a collection of coupled first-order differential equations. If a given multivariable plant is described by a set of coupled differential equations whose order is greater than one, it is necessary to convert these equations into state-space form in order to use state-space design techniques. This section gives one method of obtaining a state-space model from a given set of differential equations.

Obtaining state-space models for multivariable systems is more complicated than for SISO systems in the sense that a state-space model cannot be written by inspection of the coefficients of the differential equation. Rather a sequence of calculations is performed which results in a state-space model. One such algorithm for computing a multivariable state-space model is shown in Table 9.1. This algorithm is illustrated in the following examples.

EXAMPLE 9.2

Consider a 2-input, 2-output system described by the following differential equations

$$\ddot{y}_1(t)+2y_1(t)+\dot{y}_2(t)=u_1(t)$$

$$\dot{y}_2(t)+3y_1(t)=u_2(t)$$

1. Solve each differential equation for its highest derivative of an output variable. The terms on the right-hand side of each equation should begin with the highest derivatives of inputs and outputs, and proceed to lower-order derivatives and undifferentiated terms.
2. Formally integrate each differential equation as many times as required to achieve an (undifferentiated) output variable on the left-hand side of each equation.
3. Group terms on the right-hand side of each equation into a nested sequence of integrations. Use up integrations to remove all derivatives. Then define each integral to be a state variable, and let each integrand be the derivative of the corresponding state variable.
4. Using the definitions from the previous step, write equations for the derivatives of each state variable and also for each output. These equations will be in terms of state variables, inputs, and outputs.
5. Substitute the output equations as necessary into the state equations to obtain derivatives of state variables in terms of state variables and inputs only.
6. Write the final state-space model in matrix form.

Table 9.1 An algorithm for computing a state-space model for a multivariable system described by differential equations.

We apply each step of the algorithm in Table 9.1 with the following results. Note that the time dependence of all functions is suppressed for convenience.

Step 1

$$\ddot{y}_1 = -\dot{y}_2 - 2y_1 + u_1$$

$$\dot{y}_2 = -3y_1 + u_2$$

Step 2

$$y_1 = \int [-y_2 + \int (-2y_1 + u_1)]$$

$$y_2 = \int (-3y_1 + u_2)$$

Step 3

$$y_1 = \overbrace{\int \overbrace{[-y_2 + \underbrace{\int \underbrace{(-2y_1 + u_1)}_{\dot{x}_1}}_{x_1}]}^{\dot{x}_2}}^{x_2}$$

$$y_2 = \underbrace{\int \underbrace{(-3y_1 + u_2)}_{\dot{x}_3}}_{x_3}$$

Step 4

$$\begin{aligned}
\dot{x}_1 &= -2y_1 + u_1 \quad (a)\\
\dot{x}_2 &= -y_2 + x_1 \quad (b)\\
y_1 &= x_2 \quad (c)\\
\dot{x}_3 &= -3y_1 + u_2 \quad (c)\\
y_2 &= x_3
\end{aligned}$$

Step 5

$$\begin{aligned}
(c) \rightarrow (a) \quad \dot{x}_1 &= -2x_2 + u_1\\
(e) \rightarrow (b) \quad \dot{x}_2 &= -x_3 + x_1\\
(c) \rightarrow (d) \quad \dot{x}_3 &= -3x_2 + u_2\\
(c) \quad y_1 &= x_2\\
(e) \quad y_2 &= x_3
\end{aligned}$$

Step 6

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & -2 & 0 \\ 1 & 0 & -1 \\ 0 & -3 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}(t)$$

$$\mathbf{y}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}(t)$$

■

The next example illustrates what happens when the differential equations contain derivatives of the inputs.

EXAMPLE 9.3

Consider a 2-input, 2-output system described by the following differential equations

$$\ddot{y}_1(t) + 3\dot{y}_1(t) + 2y_2(t) = \ddot{u}_1(t) + 2u_2(t) + \dot{u}_2(t))$$

$$\dot{y}_2(t) + 3y_1(t) = u_1(t)$$

Step 1

$$\ddot{y}_1 = \ddot{u}_1 - 3\dot{y}_2 + \dot{u}_2 - 2y_2 + 2u_2$$

$$\dot{y}_2 = -3y_1 + u_1$$

Step 2

$$y_1 = u_1 \int [-3y_1 - u_2 \; + \int (-2y_2 + 2u_2)]$$

$$y_2 = \int (-3y_1 + u_1)$$

Step 3

$$y_1 = u_1 + \int \overbrace{[-3y_1 - u_2 \; + \; \underbrace{\int \underbrace{(-2y_2 + u_2)}_{\dot{x}_1}}_{x_1}]}^{\dot{x}_2} \overbrace{\qquad\qquad}^{x_2}$$

$$y_2 = \underbrace{\int \underbrace{(-3y_1 + u_1)}_{\dot{x}_3}}_{x_3}$$

Step 4

$$\begin{aligned}
\dot{x}_1 &= -2y_2 + 2u_2 && (a)\\
\dot{x}_2 &= -3y_1 - u_2 + x_1 && (b)\\
y_1 &= u_1 + x_2 && (c)\\
\dot{x}_3 &= -3y_1 + u_1 && (c)\\
y_2 &= x_3
\end{aligned}$$

Step 5

$$\begin{aligned}
(c) \rightarrow (a) \quad \dot{x_1} &= -2x_3 + 2u_2\\
(e) \rightarrow (b) \quad \dot{x}_2 &= -3(u_1 + x_2) - u_2 + x_1\\
(c) \rightarrow (d) \quad \dot{x}_3 &= -3(u_1 + x_2) + u_1\\
(c) \qquad y_1 &= u_1 + x_2\\
(e) \qquad y_2 &= x_3
\end{aligned}$$

Step 6

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 0 & -2 \\ 1 & -3 & 0 \\ 0 & -3 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 & 2 \\ -3 & -1 \\ -2 & 0 \end{bmatrix} \mathbf{u}(t)$$

$$\mathbf{y}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{u}(t)$$

■

9.1.3 ZOH Equivalent of a MIMO System

In order to design a digital control system for a multivariable plant, it is necessary to find the zero-order hold equivalent model for the plant. Such models were derived for SISO plants in Chapter 4. The same derivation holds for MIMO plants with minor changes in notation. The results are briefly described below.

The solution to the state equation for a multivariable system is

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^{t} e^{\mathbf{A}(t-\lambda)}\mathbf{B}\mathbf{u}(\lambda)d\lambda. \tag{9.14}$$

If we let $t_0 = kT$ and $t = kT + T$ then the above equation gives an update formula for the value of the state vector at sampling instants. If the vector of inputs $\mathbf{u}(t)$ is provided by a digital control system, then each input signal will have a constant value in the interval $kT \leq t < kT + T$. Thus the vector $\mathbf{u}(\lambda)$ can be taken outside the integral in (9.14). The resulting equation is a discrete-time state update equation

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}\mathbf{u}[k] \tag{9.15}$$

where

$$\mathbf{\Phi} = e^{\mathbf{A}T}\ , \ \ \mathbf{\Gamma} = \int_0^T e^{\mathbf{A}\lambda}\mathbf{B}d\lambda. \tag{9.16}$$

Thus the formula for the ZOH equivalent of a multi-input system is of the same form as that for a single-input system. The only difference is that the input matrix $\mathbf{B}$ is used instead of a single input vector $\mathbf{b}$. The output matrices $\mathbf{C}$ and $\mathbf{D}$ are identical for the continuous- and discrete-time systems, just as for SISO systems.

9.2 Tests for Controllability

In Chapter 6 the definition of controllability of SISO systems was given. The same definition is used for MIMO systems, namely that a system is controllable if and only if a (vector) input sequence can be found which takes the system from a given initial state to a given final state in finite time. Following the same derivation given in Chapter 6, the ability to find such an input sequence is equivalent to solving the following system of linear equations for $\mathbf{x}$

$$\mathbf{W}_c\mathbf{x} = \mathbf{z} \tag{9.17}$$

where $\mathbf{W}_c$ is the controllability matrix

$$\mathbf{W}_c = [\,\mathbf{\Gamma} \quad \mathbf{\Phi\Gamma} \quad \cdots \quad \mathbf{\Phi}^{n-1}\mathbf{\Gamma}\,] \tag{9.18}$$

and $\mathbf{z}$ is an arbitrary vector. Note that for a multivariable system, the controllability matrix (9.18) is not square, but has dimensions $n \times n \cdot p$. From the theory of linear equations

developed in Chapter 2, we know that (9.17) will have a solution if and only if the rank of $\mathbf{W}_c$ is equal to n. Thus we have

Fact 9.1 *The multivariable system $(\mathbf{\Phi}_{n\times n}, \mathbf{\Gamma}_{n\times p})$ is controllable if and only if the controllability matrix $\mathbf{W}_c$ in (9.18) has rank n.*

We mention another test for controllability which will be useful in deriving multivariable pole-placement algorithms.

Fact 9.2 *The multivariable system $(\mathbf{A}_{n\times n}, \mathbf{B}_{n\times m})$ is controllable if and only if the rank of the matrix*

$$[\,(\lambda\mathbf{I} - \mathbf{A}) \quad \mathbf{B}\,]$$

is n for all complex numbers λ.

This test for controllability is called the Popov-Belevitch-Hautus (PBH) rank test. We shall use this result in the next section to derive an algorithm for computing a matrix of feedback gains which will place the poles of a multivariable system at desired locations.

9.3 Pole Placement via State Feedback

The regulation problem was solved for SISO systems in Chapter 6 by computing state-feedback vectors which placed the poles of the closed-loop system at desired locations. In this section we show how multivariable regulators can be designed using pole placement.

A multivariable digital regulator is shown in Fig. 9.1. If $(\mathbf{\Phi}, \mathbf{\Gamma})$ is the ZOH equivalent

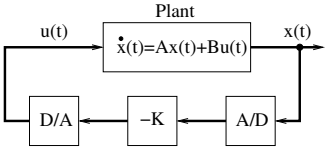


Figure 9.1 A multivariable digital regulation system.

of the plant, then the equations for the closed-loop system are

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}\mathbf{u}[k] \\ \mathbf{u}(k) &= -\mathbf{K}\mathbf{x}(k) \\ \mathbf{x}[k+1] &= (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})\mathbf{x}[k]. \end{aligned} \tag{9.19}$$

These equations are identical in form to the equations for a SISO regulator. The only difference in the multivariable case is that $\mathbf{\Gamma}$ is an $n \times p$ matrix and $\mathbf{K}$ is an $q \times n$ matrix. The solution to the last line of (9.19) is

$$\mathbf{x}[k] = (\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})^k \mathbf{x}[0]. \tag{9.20}$$

The regulation property is achieved if $\mathbf{x}[k] \to \mathbf{0}$ for any value of $\mathbf{x}[0]$. From (9.20), this will happen if the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma K}$ are within the unit circle. Thus the multivariable pole placement problem can be stated as

> Given $\mathbf{\Phi}_{n\times n}, \mathbf{\Gamma}_{n\times p}$ and desired closed-loop pole locations $\lambda_1, \cdots, \lambda_n$, find a feedback gain matrix $\mathbf{L}_{p\times n}$ such that the eigenvalues of the matrix $\mathbf{\Phi} - \mathbf{\Gamma K}$ are $\lambda_1, \cdots, \lambda_n$.

Notice that there are $p \cdot n$ unknown gains in the matrix $\mathbf{K}$, but only n desired pole locations are specified. If the plant has more than one input, $p > 1$, and so the number of unknowns in $\mathbf{K}$ $(p \cdot n)$ is greater than the number of desired pole locations (n). Since there are more unknowns than specified poles, the multivariable pole-placement problem is underdetermined, and does not have a unique solution. This lack of unique solution means that we cannot derive a formula for $\mathbf{K}$ as was done in Chapter 6 for SISO systems. However, we can develop algorithms for calculating $\mathbf{K}$. Two such algorithms will be developed. The first is convenient because it is noniterative. The second algorithm is iterative, but it uses the nonuniqueness in the solution to advantage. It finds a particular $\mathbf{K}$ matrix which optimizes a robustness criterion. We will use the noniterative algorithm to initialize the iterative algorithm. For both algorithms, we assume that the plant satisfies the controllability property.

9.4 Noniterative Algorithm

If λ_0 is to be a pole of the closed-loop system, then by definition it must be an eigenvalue of the closed-loop system matrix $\mathbf{\Phi} - \mathbf{\Gamma K}$. Let the vector $\boldsymbol{\psi}_0$ be an eigenvector of $\mathbf{\Phi} - \mathbf{\Gamma L}$ corresponding to eigenvalue λ_0; that is, $\boldsymbol{\psi}_0$ and λ_0 satisfy

$$(\mathbf{\Phi} - \mathbf{\Gamma K})\boldsymbol{\psi}_0 = \lambda_0 \boldsymbol{\psi}_0. \tag{9.21}$$

This equation can be rewritten as

$$(\lambda_0 \mathbf{I} - \mathbf{\Phi})\boldsymbol{\psi}_0 + \mathbf{\Gamma L}\boldsymbol{\psi}_0 = \mathbf{0}$$

or

$$\begin{bmatrix} (\lambda_0 \mathbf{I} - \mathbf{\Phi}) & \mathbf{\Gamma} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}_0 \\ \mathbf{K}\boldsymbol{\psi}_0 \end{bmatrix} = \mathbf{0}. \tag{9.22}$$

The coefficient matrix in the above equation will be denoted $\mathbf{P}(\lambda_0)$

$$\mathbf{P}(\lambda_0) \stackrel{\text{def}}{=} \begin{bmatrix} (\lambda_0 \mathbf{I} - \mathbf{\Phi}) & \mathbf{\Gamma} \end{bmatrix}. \tag{9.23}$$

Note that $\mathbf{P}(\lambda_0)$ is the matrix used in the PBH controllability test (Fact 9.2). Since we are assuming the system is controllable, $\mathbf{P}(\lambda_0)$ has rank n. We also note that the dimension of $\mathbf{P}(\lambda_0)$ is $n \times (n + p)$. Thus the dimension of the null-space of $\mathbf{P}(\lambda_0)$ is p. This means that we can always pick up to p linearly independent vectors from the null-space of $\mathbf{P}(\lambda_0)$. This fact will be used shortly.

We can rewrite (9.22) using the definition of $\mathbf{P}(\lambda_0)$ as

$$\mathbf{P}(\lambda_0) \begin{bmatrix} \boldsymbol{\psi} \\ \mathbf{K}\boldsymbol{\psi} \end{bmatrix} = \mathbf{0}. \tag{9.24}$$

This equation shows a null-space vector of $\mathbf{P}(\lambda_0)$ partitioned into an n-vector $\boldsymbol{\psi}$ and an p-vector $\mathbf{L}\boldsymbol{\psi}$. Equation (9.24) provides a relationship between eigenvectors of the closed-loop system (9.21) and null-space vectors of $\mathbf{P}(\lambda_0)$. The relationship provided by (9.24)

is simply this: if a null-space vector (which has $n + p$ elements) is partitioned into two vectors, the first consisting of n elements and the second consisting of p elements, then the second vector equals $\mathbf{L}$ times the first vector. This relationship will be used to develop a procedure for calculating the feedback gain matrix $\mathbf{K}$ from null-space vectors of $\mathbf{P}(\lambda_0)$.

To place closed-loop pole(s) at λ_0 with multiplicity $\mu_0 \leq p$, we begin by choosing μ linearly independent vectors from the null-space of $\mathbf{P}(\lambda_0)$. Each vector is partitioned into a "top part" vector $\mathbf{t}_i$ consisting of n elements, and a "bottom part" vector $\mathbf{b}_i$ consisting of p elements, $i = 1, \cdots, \mu_0$. Then, in accordance with (9.24), we set the bottom part of each null-space vector equal to $\mathbf{K}$ times the top part to get the equations

$$\mathbf{b}_i = \mathbf{K}\mathbf{t}_i \ , \quad i = 1, \cdots \mu_{.0}$$

If we can find a matrix $\mathbf{K}$ which satisfies these equations, we know that the resulting closed-loop system will have an pole of multiplicity μ_0 at the desired value λ_0. We now repeat this procedure for *all* the desired closed-loop pole locations. That is, we form another matrix $\mathbf{P}(\lambda_1)$ corresponding to another desired closed-loop pole location and extract μ_1 null-space vectors, where μ_1 is the desired multiplicity of the pole at λ_1. We continue this procedure until all the desired closed-loop poles have been included. Since we are placing a total of n poles, this procedure will yield a total of n null-space vectors. Although these vectors may correspond to different values of λ, they all satisfy the relationship shown in (9.24). Thus if we group the top parts of the null-space vectors into an $n \times n$ matrix $\mathbf{T}$, and the corresponding bottom parts into an $p \times n$ matrix $\mathbf{B}$, we get

$$\mathbf{B} = \mathbf{K}\mathbf{T} \tag{9.25}$$

from which we can solve for $\mathbf{L}$ uniquely as

$$\mathbf{K} = \mathbf{B}\mathbf{T}^{-1}. \tag{9.26}$$

It is important to note that although the solution for $\mathbf{K}$ in (9.26) is unique, the construction of the matrices $\mathbf{B}$ and $\mathbf{T}$ is *not* unique. The reason is that the selection of vectors from the null spaces of each of the $\mathbf{P}(\lambda)$ matrices is arbitrary. Different choices for these null-space vectors will result in different $\mathbf{K}$ matrices. All such $\mathbf{K}$ matrices will give the desired closed-loop pole locations, but it may be advantageous to find a particular $\mathbf{K}$ matrix which satisfies some additional properties beyond desired closed-loop poles. The next section introduces some additional properties for the $\mathbf{K}$ matrix and shows how to achieve them with an iterative algorithm.

Remarks

1. The matrix $\mathbf{T}$ in (9.26) is formed from the top parts of null-space vectors of $\mathbf{P}(\lambda_i)$ for different closed-loop poles λ_i. Since each null-space vector is chosen arbitrarily from a p-dimensional null space, it is possible that T is singular. In this case a different choice of null-space vectors can be made so that the resulting $\mathbf{T}$ is nonsingular.

2. In the above development, if a desired closed-loop pole location λ_0 is real-valued, then the corresponding null-space vectors will be real valued. However, if λ_0 is complex, then the null-space vectors will be complex, as will the corresponding top and bottom parts. Thus it is possible that (9.26) will yield a feedback gain matrix with complex elements. A complex $\mathbf{L}$ matrix is a correct mathematical solution to the pole-placement problem in the sense that the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}$ will bee at the

specified locations. But a complex $\mathbf{K}$ matrix is undesirable from a practical point of view since the input to the plant must be real-valued, and the input is obtained by the state feedback $-\mathbf{Kx}$. We will show that it is always possible to find a real-valued $\mathbf{K}$ matrix even when some of the desired closed-loop poles are complex.

We conclude this section with three examples. The first two examples illustrate the calculation of feedback matrices for the case of desired closed-loop poles which are real valued with multiplicity greater than one, and real valued, distinct poles. After showing how to handle complex closed-loop poles, we give an example of this case.

EXAMPLE 9.4

Consider the following ZOH equivalent of a 2-input second-order system:

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 1 \\ 0 & .9 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Suppose that the desired closed-loop poles are located at $z = 0.9$ in the z-plane with multiplicity two. The matrix $\mathbf{P}(0.9)$ defined in (9.23) with $\lambda = 0.9$ is

$$\mathbf{P}(0.9) = [\,(0.9\mathbf{I} - \mathbf{\Phi}) \quad \mathbf{\Gamma}\,] = \begin{bmatrix} -0.1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

According to the development in the previous section, we need to compute the null-space of $\mathbf{P}(0.9)$ and partition the null-space vectors into top and bottom parts. One way to compute the null-space of a matrix is to compute its singular value decomposition (SVD), $\mathbf{P}(0.9) = \mathbf{U}\Sigma\mathbf{V}^T$. The matrix $\mathbf{V}$ in the SVD of $\mathbf{P}(0.9)$ is computed to be

$$\mathbf{V} = \begin{bmatrix} .0705 & 0 & -.9975 & 0 \\ .7053 & 0 & .0499 & -.7071 \\ -.7053 & 0 & -.4099 & -.7071 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

and an orthonormal basis for the null-space of $\mathbf{P}(0.9)$ is given by the last two columns of $\mathbf{V}$. Since the multiplicity of the desired pole at 0.9 is two, we need to take two vectors from the null-space; a natural choice is the last two columns of $\mathbf{V}$. These vectors are partitioned into the first n rows to form the matrix $\mathbf{T}$, and the last m rows to form the matrix $\mathbf{B}$. Since $n = m = 2$, we have

$$\mathbf{T} = \begin{bmatrix} -.9975 & 0 \\ .0499 & -.7071 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -.0499 & -.7071 \\ 0 & 0 \end{bmatrix}.$$

From (9.26), we compute the feedback matrix $\mathbf{K}$ as

$$\mathbf{K} = \mathbf{B}\mathbf{T}^{-1} = \begin{bmatrix} .1 & 1 \\ 0 & 0 \end{bmatrix}.$$

It can be verified that the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma K}$ are both at 0.9, as desired. Since we had to take both null-space vectors of $\mathbf{P}(0.9)$, there was no flexibility in choosing null-space vectors, and so the matrix $\mathbf{L}$ is unique in this case.

■

EXAMPLE 9.5

Suppose we have the same system as in the previous example, but the desired closed-loop pole locations are now $z = 0.9$ and $z = 0.8$. In this case, we need to compute the matrices $P(0.9)$ and $\mathbf{P}(0.8)$ as well as their null-spaces. These calculations were done for $\mathbf{P}(0.9)$ in the previous example; the results for $\mathbf{P}(0.8)$ are

$$\mathbf{P}(0.8) = [\,(0.8\mathbf{I} - \mathbf{\Phi}) \quad \mathbf{\Gamma}\,] = \begin{bmatrix} -0.2 & -1 & 1 & 0 \\ 0 & -.1 & 0 & 1 \end{bmatrix}.$$

The last two columns of the matrix $\mathbf{V}$ from the SVD of $\mathbf{P}(0.8)$ are calculated to be

$$\begin{bmatrix} -.9901 & 0 \\ .0985 & .7053 \\ -.0995 & .7053 \\ .0099 & .0705 \end{bmatrix}.$$

Since the desired poles have multiplicity one, we only choose one vector each from the null spaces of $\mathbf{P}(0.9)$ and $\mathbf{P}(0.8)$. Since the choice of null-space vectors is not unique in this case, there will be many $\mathbf{L}$ matrices which give the desired closed-loop poles. We will calculate two of them here. If we choose the first vector from the null-space of $\mathbf{P}(0.9)$ given in the previous example, and the first vector from the null-space of $\mathbf{P}(0.8)$ given above, we obtain the following top and bottom matrices

$$\mathbf{T}_1 = \begin{bmatrix} -.9975 & -.9901 \\ .0499 & .0985 \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} -.0499 & -.0995 \\ 0 & .0099 \end{bmatrix}.$$

For this choice of null-space vectors, the feedback matrix is calculated to be

$$\mathbf{L}_1 = \mathbf{B}_1\mathbf{T}_1^{-1} = \begin{bmatrix} -.0010 & -1.0201 \\ 0.0101 & 0.2010 \end{bmatrix}.$$

It can be verified that the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{L}_1$ are at 0.9 and 0.8 as desired.

Suppose we now choose different null-space vectors: the first null-space vector from $\mathbf{P}(0.9)$ and the second null-space vector from $\mathbf{P}(0.8)$. This choice results in the following $\mathbf{T}$ and $\mathbf{B}$ matrices

$$\mathbf{T}_2 = \begin{bmatrix} -.9975 & 0 \\ 0.0499 & .7053 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} -.0499 & .7053 \\ 0 & .0705 \end{bmatrix}$$

resulting in the feedback matrix

$$\mathbf{K}_2 = \mathbf{B}_2\mathbf{T}_2^{-1} = \begin{bmatrix} 0.1000 & 1.000 \\ 0.0050 & 0.1000 \end{bmatrix}.$$

It can be verified that the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}_2$ are also at 0.9 and 0.8 for the matrix $\mathbf{L}_2$, even though $\mathbf{L}_2 \neq \mathbf{L}_1$. Since both $\mathbf{K}_1$ and $\mathbf{K}_2$ give the same closed-loop eigenvalues, is there any reason to prefer one $\mathbf{K}$ matrix over another? The answer is yes – in the next section, a robustness criterion will be introduced and an algorithm to find the $\mathbf{K}$ matrix that optimizes this criterion will be developed.

■

9.4.1 Complex Closed-Loop Poles

If λ_0 is a complex desired closed-loop pole, then its complex conjugate λ_0^* must also be a desired closed-loop pole. Suppose that $\mathbf{n}_0$ is a complex null-space vector of $\mathbf{P}(\lambda_0)$; that is

$$\mathbf{P}(\lambda_0)\mathbf{n}_0 = \mathbf{0}. \tag{9.27}$$

It is easy to see that

$$[\mathbf{P}(\lambda_0)]^* = \mathbf{P}(\lambda_0^*).$$

Using this result and taking the complex conjugate of (9.27) yields

$$\mathbf{P}(\lambda_0^*)\mathbf{n}_0^* = \mathbf{0}.$$

Thus $\mathbf{n}_0^*$ is in the null-space of $\mathbf{P}(\lambda_0^*)$. This means that we do not have to calculate null-space vectors for both $\mathbf{P}(\lambda_0)$ and $\mathbf{P}(\lambda_0^*)$. Once a null-space vector has been chosen for $\mathbf{P}(\lambda_0)$, we simply conjugate this vector to obtain a null-space vector for $\mathbf{P}(\lambda_0^*)$.

The null-space vector $\mathbf{n}_0$ for $\mathbf{P}(\lambda_0)$ is partitioned into its top and bottom parts, $\mathbf{t}$ and $\mathbf{b}$. The null-space vector $\mathbf{n}_0^*$ for $\mathbf{P}(\lambda_0^*)$ is partitioned into its top and bottom parts, $\mathbf{t}^*$ and $\mathbf{b}^*$.

Remark

It is always true that if $\mathbf{n}_0$ is in the null space of $\mathbf{P}(\lambda_0)$ then $\mathbf{n}^*$ is in the null space of $\mathbf{P}(\lambda_0^*)$. However, in order for (9.26) to have a solution, we require the top parts of $\mathbf{n}$ and $\mathbf{n}^*$, called $\mathbf{t}$ and $\mathbf{t}^*$ respectively, to be linearly independent. It is possible for certain choices of $\mathbf{n}$ that $\mathbf{t}$ and $\mathbf{t}^*$ will be linearly dependent. If this situation occurs, a different null space vector of $\mathbf{P}(\lambda_0)$ must be chosen so that the resulting $\mathbf{t}$ and $\mathbf{t}^*$ are linearly independent.

The matrices $\mathbf{T}$ and $\mathbf{B}$ are used in (9.25) to calculate the feedback gain matrix $\mathbf{K}$. It is shown in [10, page 149] that when complex columns of $\mathbf{B}$ and $\mathbf{T}$ occur in conjugate pairs, the solution of (9.25) will be real-valued. Moreover, the solution can be computed using only real numbers by replacing the columns $\mathbf{t}$ and $\mathbf{t}^*$ in the matrix $\mathbf{T}$ with the real part of $\mathbf{t}$ and the imaginary part of $\mathbf{t}$, respectively. The columns $\mathbf{b}$ and $\mathbf{b}^*$ in the matrix $\mathbf{B}$ must also be replaced with the real and imaginary parts of $\mathbf{b}$. An example of the procedure to calculate $\mathbf{K}$ to obtain complex closed-loop poles is given next.

EXAMPLE 9.6

This example uses the same plant as the previous two examples (see example 9.4, page 418). Suppose now that the desired pole locations are a complex-conjugate pair: $0.9 \pm j0.02$. For complex-conjugate eigenvalues, we only need to compute the $\mathbf{P}(\lambda)$ matrix for one of them, so we compute

$$\mathbf{P}(0.9 + j0.02) = \begin{bmatrix} -0.1 + j0.02 & -1.000 & 1.000 & 0 \\ 0 & j0.02 & 0 & 1.000 \end{bmatrix}.$$

An orthonormal basis for the null-space of this matrix obtained from an SVD is

$$\begin{bmatrix} -0.9974 & 0 \\ 0.0499 - j0.0100 & 0 + j0.7070 \\ -0.0499 + j0.0100 & 0 + j0.7070 \\ -0.0002 - j0.0010 & 0.0141 \end{bmatrix}.$$

We can take the real and imaginary parts of the first column of this matrix to form the $\mathbf{B}$ and $\mathbf{T}$ matrices as follows

$$\mathbf{T}_1 = \begin{bmatrix} -0.9974 & 0 \\ 0.0499 & -0.0100 \end{bmatrix}, \quad \mathbf{B}_1 = \begin{bmatrix} -0.0499 & 0.0100 \\ -0.0002 & -0.0010 \end{bmatrix}.$$

The resulting feedback gain matrix is then

$$\mathbf{K}_1 = \mathbf{B}_1\mathbf{T}_1^{-1} = \begin{bmatrix} 0 & -1.0004 \\ .0052 & .1000 \end{bmatrix}.$$

Notice that the real and imaginary parts of the second null-space vector cannot be used, because this choice would result in a singular $\mathbf{T}$ matrix. However, the sum of the two null-space vectors is also in the null space. The real and imaginary parts of this vector can be used to form the following $\mathbf{B}$ and $\mathbf{T}$ matrices

$$\mathbf{T}_2 = \begin{bmatrix} -.9974 & 0 \\ .0499 & .6971 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} -0.0499 & 0.7170 \\ 0.0139 & -0.0010 \end{bmatrix}$$

with the resulting $\mathbf{L}$ matrix as

$$\mathbf{K}_2 = \mathbf{B}_2\mathbf{T}_2^{-1} = \begin{bmatrix} .1014 & 1.0286 \\ -.0140 & -.0014 \end{bmatrix}.$$

It can be verified that both of the $\mathbf{K}_1$ and $\mathbf{K}_2$ give the desired closed-loop eigenvalues; that is, the eigenvalues of $(\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}_i)$, $i = 1, 2$, are 0.9 ± 0.02.

■

9.5 Iterative Algorithm

The algorithm presented in this section is due to Kautsky, Nichols, Van Dooren [52], and it uses the extra degrees of freedom in the multivariable pole-placement problem to optimize a robustness criterion. The robust solution has a number of useful properties which will be given shortly.

Recall that the pole-placement problem is to find a matrix $\mathbf{K}$ such that the matrix $(\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K})$ has specified eigenvalues. The matrices $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ are calculated as the ZOH equivalent of the plant. We assume that the mathematical model of the plant is good, but not perfect. Suppose that the actual plant ZOH is $(\mathbf{\Phi} + \Delta\mathbf{\Phi},\ \mathbf{\Gamma} + \Delta\mathbf{\Gamma})$, where $\Delta\mathbf{\Phi}$ and $\Delta\mathbf{\Gamma}$ represent modeling errors. Then the actual closed-loop poles will be the eigenvalues of $[\mathbf{\Phi} + \Delta\mathbf{\Phi} - (\mathbf{\Gamma} + \Delta\mathbf{\Gamma})\mathbf{K}]$, and these will be different from the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}$. It is desirable for the closed-loop pole locations to be as *insensitive* as possible to the perturbations $\Delta\mathbf{\Phi}$ and $\Delta\mathbf{\Gamma}$. In other words, out of all the matrices $\mathbf{K}$ which give the desired closed-loop pole locations for the nominal ZOH model, we would like to find that particular $\mathbf{K}$ for which the eigenvalues of $[\mathbf{\Phi} + \Delta\mathbf{\Phi} - (\mathbf{\Gamma} + \Delta\mathbf{\Gamma})\mathbf{K}]$ are as close as possible to the eigenvalues of $(\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}$ in spite of arbitrary perturbations $\Delta\mathbf{\Phi}$ and $\Delta\mathbf{\Gamma}$. Such a matrix $\mathbf{K}$ will be said to be a *robust* solution to the pole-placement problem.

It is known that the eigenvalues of a matrix are insensitive to perturbations in the matrix elements when the eigenvectors of that matrix are orthogonal to each other. If the eigenvectors cannot be orthogonal to each other, then they should be "as orthogonal as possible." Thus the robust pole-placement problem can be stated as follows:

Given $\mathbf{\Phi}_{n\times n}, \mathbf{\Gamma}_{n\times p}$ and desired closed-loop pole locations $\lambda_1, \cdots, \lambda_n$, find the feedback gain matrix $\mathbf{K}_{p\times n}$ such that the eigenvalues of the matrix $\mathbf{\Phi} - \mathbf{\Gamma K}$ are $\lambda_1, \cdots, \lambda_n$ and the eigenvectors of $\mathbf{\Phi} - \mathbf{\Gamma K}$ are as orthogonal to each other as possible.

The idea of the iterative pole-placement algorithm is to search over feedback matrices $\mathbf{K}$ which yield the desired closed-loop pole locations and find that particular $\mathbf{L}$ which causes the eigenvectors of $\mathbf{\Phi} - \mathbf{\Gamma K}$ to be as orthogonal as possible. It is shown in [52] that the resulting $\mathbf{K}$ will have small norm and it will minimize an upper bound on the magnitude of the transient response of the closed-loop system. These properties will be illustrated by example after the algorithm is derived.

Let us start with the simplest case in which the desired closed-loop poles $\lambda_1, \cdots, \lambda_n$ are distinct. Suppose we have a corresponding set of closed loop eigenvectors $\boldsymbol{\psi}_1, \cdots, \boldsymbol{\psi}_n$ obtained, for example, from the non-iterative algorithm of the previous section. Define $\mathcal{S}_i$ to be the space spanned by the top parts (first n elements) of the null-space vectors of $\mathbf{P}(\lambda_i)$ (see (9.23)). Then we know that $\boldsymbol{\psi}_i \in \mathcal{S}_i$. Recall that the null space of $\mathbf{P}(\lambda_i)$ is p-dimensional. This implies that $\mathcal{S}_i$ is p-dimensional, and so there is some freedom in choosing $\boldsymbol{\psi}_i$. The vector $\boldsymbol{\psi}_i$ can be chosen to be *any* vector from the p-dimensional space $\mathcal{S}_i$. Out of all the vectors in $\mathcal{S}_i$, we would like to find that vector $\boldsymbol{\psi^{opt}}_i$ which is as orthogonal as possible to all the other closed-loop eigenvectors. That is, we would like to find the vector $\boldsymbol{\psi^{opt}}_i$ which satisfies the following two properties:

1. $\boldsymbol{\psi^{opt}}_i \in \mathcal{S}_i$.
2. $\boldsymbol{\psi^{opt}}_i$ is as orthogonal as possible to $\boldsymbol{\psi}_j^{opt},\ j \neq i$.

We can find $\boldsymbol{\psi^{opt}}_i$ as follows. First, we calculate a vector $\mathbf{x}$ which is orthogonal to $\boldsymbol{\psi}_j^{opt},\ j \neq i$. Notice that $\mathbf{x}$ is unique up to a (complex) scale factor because it is orthogonal to $n-1$ vectors in $\mathbf{R}^n$. One way to calculate the vector $\mathbf{x}$ is to compute a singular value decomposition of the $n \times (n-1)$ the matrix $\mathbf{M}$ whose columns are given by $\boldsymbol{\psi}_j,\ j \neq i$. The left singular vectors will be contained in an $n \times n$ matrix $\mathbf{U}$. The first $n-1$ columns of $\mathbf{U}$ will be an orthonormal basis for the column space of the matrix $\mathbf{M}$, and the n-th column of $\mathbf{U}$ will be a basis for the (one dimensional) orthogonal complement of the column space of $\mathbf{M}$. Thus $\mathbf{x}$ can be chosen as the last column of the matrix $\mathbf{U}$ obtained from an SVD of $\mathbf{M}$, as shown in the following equations

$$\begin{aligned} \mathbf{M} &= [\boldsymbol{\psi}_j,\ j \neq i] \\ \mathbf{M} &= \mathbf{U}\Sigma\mathbf{V}^H \ \text{ singular value decomposition of } \mathbf{M} \\ \mathbf{x} &= n\text{-th column of the matrix } \mathbf{U}. \end{aligned} \tag{9.28}$$

If we could choose $\boldsymbol{\psi}_i^{opt} = \mathbf{x}$, then $\boldsymbol{\psi}_i^{opt}$ would be orthogonal to $\boldsymbol{\psi}_j\ j \neq i$, but in general, we cannot make this choice. The reason is that $\mathbf{x} \notin \mathcal{S}_i$, in general. To obtain the $\boldsymbol{\psi^{opt}}_i$ "most orthogonal" to $\boldsymbol{\psi}_j,\ j \neq i$, we want to perturb $\mathbf{x}$ as little as possible in such a way that the perturbed $\mathbf{x}$ is in the subspace $\mathcal{S}_i$. The way to do this is to *project* $\mathbf{x}$ onto $\mathcal{S}_i$ using an orthogonal projection. In other words,

$$\boldsymbol{\psi^{opt}}_i = \text{ projection of } \mathbf{x} \text{ onto } \mathcal{S}_i = \mathbf{U}_i\mathbf{U}_i^H\mathbf{x} \tag{9.29}$$

where the columns of $\mathbf{U}_i$ are an orthonormal basis for $\mathcal{S}_i$ and the superscript 'H' stands for complex-conjugate transpose. $\mathbf{U}_i$ can be chosen to be the first p left singular vectors of the matrix $\mathbf{T}_i$. As a final step, we normalize $\boldsymbol{\psi^{opt}}_i$ in (9.29) to have unit length.

EXAMPLE 9.7

This example demonstrates the calculation of the "most orthogonal" closed-loop eigenvector from a given set of eigenvectors as discussed above. Recall that closed-loop eigenvectors are selected from the top parts (first n rows) of null-space vectors of $\mathbf{P}(\lambda)$ (see (9.24)). Thus the matrix $\mathbf{T}$ computed in the non-iterative algorithm of the previous section contains closed-loop eigenvectors. For instance, in Example 9.5 the closed-loop eigenvectors are the columns of $\mathbf{T}$

$$\boldsymbol{\psi}_1 = \begin{bmatrix} -.9975 \\ .0499 \end{bmatrix}, \quad \boldsymbol{\psi}_2 = \begin{bmatrix} -.9901 \\ .0985 \end{bmatrix}.$$

It is easy to see that these two eigenvectors are quite far from being orthogonal (in fact they are almost parallel; see Fig. 9.2). Suppose that we want to compute $\boldsymbol{\psi}_1^{\boldsymbol{opt}}$, the

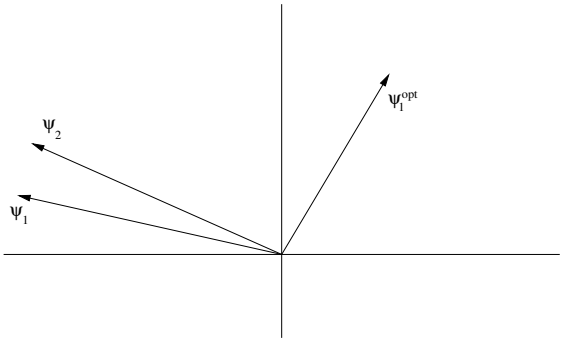


Figure 9.2 A given set of eigenvectors ψ_1 and ψ_2 and the vector ψ_1^{opt} that is "most orthogonal" to ψ_2. In this case, ψ_1^{opt} is actually orthogonal to ψ_2.

vector which is contained in $\mathcal{S}_1$ (the top part of the null-space of $\mathbf{P}(0.9)$) and is most orthogonal to $\boldsymbol{\psi}_j,\ j \neq i$. In this case, $\boldsymbol{\psi}_j,\ j \neq i$ is simply the vector $\boldsymbol{\psi}_2$. The first step in calculating $\boldsymbol{\psi}_1^{\boldsymbol{opt}}$ is to calculate $\mathbf{x}$ using (9.28). Since $\mathbf{M} = \boldsymbol{\psi}_2$, we need to compute the SVD of $\boldsymbol{\psi}_2$ to get

$$\begin{bmatrix} -.9901 \\ .0985 \end{bmatrix} = \begin{bmatrix} -0.9951 & 0.0990 \\ 0.0990 & 0.9951 \end{bmatrix} \begin{bmatrix} .9950 & 0 \\ 0 & 0 \end{bmatrix} [\,1\,]$$

$$= \mathbf{U}\Sigma\mathbf{V}^T.$$

The vector $\mathbf{x}$ is chosen to be the last column of the matrix $\mathbf{U}$

$$\mathbf{x} = \begin{bmatrix} .0990 \\ .9951 \end{bmatrix}.$$

We now need to project $\mathbf{x}$ onto the subspace $\mathcal{S}_1$ which is the span of the columns of $\mathbf{T}_1$. From Example 9.5, we know that $\mathbf{T}_1$ is given by

$$\mathbf{T}_1 = \begin{bmatrix} -.9975 & 0 \\ .0499 & -.7071 \end{bmatrix}$$

and so the projection is accomplished by

$$\boldsymbol{\psi}_{\mathbf{1}}^{\boldsymbol{opt}} = \mathbf{T}_1[\mathbf{T}_1^T\mathbf{T}_1]^{-1}\mathbf{T}_1\mathbf{x} = \mathbf{I}\mathbf{x} = \mathbf{x}.$$

In this case, the subspace $\mathcal{S}_1$ is all of $\mathbf{R}^2$, and so we can actually choose $\boldsymbol{\psi}^{\boldsymbol{opt}}$ to equal $\mathbf{x}$ (of course, we should normalize $\boldsymbol{\psi}^{\boldsymbol{opt}}$ to have unit length). The vector $\boldsymbol{\psi}_{\mathbf{1}}^{\boldsymbol{opt}}$ is shown in Fig. 9.2, and it is indeed orthogonal to $\boldsymbol{\psi}_2$.

■

An algorithm for computing a complete set of eigenvectors which are most orthogonal to each other is obtained by repeating the above procedure for calculating $\boldsymbol{\psi}_i^{opt}$ for all the eigenvectors $i = 1, \cdots, n$.

Remarks

1. The algorithm for computing the optimal eigenvectors assumes that an initial set of eigenvectors $\boldsymbol{\psi}_1^0, \cdots, \boldsymbol{\psi}_n^0$ has been obtained. The superscript 0 refers to the fact that these are the initial eigenvectors. Such a set of eigenvectors can be obtained, for instance, by choosing the top parts of the first null-space vector for each $\mathbf{P}(\lambda)$ matrix. A complete "sweep" through the eigenvectors is then performed as follows. The superscript refers to the sweep number.

   (a) The first eigenvector $\boldsymbol{\psi}_1^0$ is replaced by the eigenvector $\boldsymbol{\psi}_1^1$ most orthogonal to the other eigenvectors $\boldsymbol{\psi}_2^0, \cdots, \boldsymbol{\psi}_n^0$.

   (b) The second eigenvector $\boldsymbol{\psi}_2^0$ is replaced by $\boldsymbol{\psi}_2^1$, the eigenvector most orthogonal to the eigenvectors $\boldsymbol{\psi}_1^1,\ \boldsymbol{\psi}_3^0, \cdots \boldsymbol{\psi}_n^0$.

   (c) Step (b) is repeated for $\boldsymbol{\psi}_i^0, i = 3$ to n where at the last step ($i = n$), the eigenvector $\boldsymbol{\psi}_n^0$ is replaced by the eigenvector $\boldsymbol{\psi}_n^1$ most orthogonal to the vectors $\boldsymbol{\psi}_1^1,\ \boldsymbol{\psi}_2^1, \cdots, \boldsymbol{\psi}_{n-1}^1$.

   The steps (a) – (c) define the first sweep through the eigenvectors. It is possible to take a second sweep by starting with the vectors $\boldsymbol{\psi}_1^1, \cdots \boldsymbol{\psi}_n^1$ and replacing them one-by-one with the corresponding most orthogonal vectors. The vectors obtained from the second sweep could be used as input to a third sweep. Ideally, this iteration would stop when two successive sweeps gave the same set of eigenvectors. However, this algorithm is not guaranteed to converge to a fixed set of eigenvectors! In fact, it is not always true that sweep $i + 1$ gives a better (more orthogonal) set of eigenvectors than sweep i! Fortunately, the algorithm is known to give good results with a small number of sweeps. The algorithm given later in this section uses the following strategy: take the best solution obtained in five sweeps.

2. Calculating the optimal eigenvectors is only "half" the problem in the sense that the eigenvectors determine the matrix $\mathbf{T}$ used in (9.26), and the matrix $\mathbf{B}$ must also be specified. It will be shown that once the matrix $\mathbf{T}$ is specified using the optimal eigenvectors, the corresponding columns of the matrix $\mathbf{B}$ can be calculated in a simple way.

Before giving the complete algorithm, however, we first mention how to handle repeated and/or complex pole locations.

Suppose we have desired closed-loop poles at λ_0 with multiplicity $\mu_0 \leq p$. Then there will be μ_0 vectors that we choose from the null-space of $\mathbf{P}(\lambda_0)$, and the top parts of these

vectors will be eigenvectors of $\mathbf{\Phi} - \mathbf{\Gamma K}$. We still choose a total of n eigenvectors. The only difference between the cases of distinct and repeated poles if this: in the case of distinct poles only one null-space vector of each $\mathbf{P}(\lambda)$ matrix is used, while in the case of repeated poles, more than one null-space vector of a given $\mathbf{P}(\lambda)$ matrix is used. In either case, a total of n null-space vectors are chosen, and the top parts of these vectors are n closed-loop eigenvectors. These eigenvectors are made as orthogonal as possible by the procedure described above.

We now consider the case when the desired pole locations include a complex-conjugate pair λ_0 and λ_0^*. From the discussion on the non-iterative algorithm in the previous section we know that if a certain vector $\mathbf{v}$ is selected from the null-space of $\mathbf{P}(\lambda_0)$, then the vector $\mathbf{v}^*$ must be selected from the null-space of $\mathbf{P}(\lambda_0^*)$ in order for the resulting $\mathbf{L}$ matrix to be real-valued. The same thing is true for the iterative algorithm.

The inputs to the algorithm are $\mathbf{\Phi}$, $\mathbf{\Gamma}$, $\mathbf{p}$, and $\mathbf{q}$. The vector $\mathbf{p}$ contains the desired closed-loop pole locations. The vector $\mathbf{q}$ contains the desired multiplicities of the pole locations in $\mathbf{p}$. That is, the i-th element of $\mathbf{q}$ is the desired multiplicity for the pole location given by the i-th element of $\mathbf{p}$. Each element of $\mathbf{q}$ must be less than or equal to p, the number of inputs to the plant.

An algorithm to compute the optimal closed-loop eigenvectors is shown in Table 9.4 in Appendix B at the end of this chapter.

9.5.1 Calculating B Corresponding to the Optimal T

The discussion of the algorithm thus far has shown how to calculate an optimal set of closed-loop eigenvectors. These eigenvectors are in the spaces spanned by the top parts of the null-space vectors of the $\mathbf{P}(\lambda)$ matrices. To compute the feedback matrix $\mathbf{K}$, however, it is necessary to pick n complete null-space vectors, not just their top parts. We now show how to get the bottom parts of the null-space vectors which correspond to the top parts (closed-loop eigenvectors) that have been calculated.

Consider the matrix $\mathbf{P}(\lambda_i)$ corresponding to a desired closed-loop pole location λ_i. The null space of $\mathbf{P}(\lambda_i)$ can be partitioned into its top part (first n rows) and its bottom part (last p rows) as follows

$$\text{null}[\mathbf{P}(\lambda_i)] = \begin{bmatrix} \mathbf{T}(\lambda_i) \\ \mathbf{B}(\lambda_i) \end{bmatrix}.$$

Note that $\mathbf{T}(\lambda_i)$ is an $n \times p$ matrix and $\mathbf{B}(\lambda_i)$ is an $p \times p$ matrix. Suppose $\boldsymbol{\psi}_i$ is an optimal closed-loop eigenvector corresponding to eigenvalue λ_i. We know that $\boldsymbol{\psi}_i$ is the top part of some vector in the null-space of $\mathbf{P}(\lambda_i)$. An arbitrary vector in the null-space of $\mathbf{P}(\lambda_i)$ can be written as

$$\begin{bmatrix} \mathbf{t} \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{T}(\lambda_i)\boldsymbol{\alpha} \\ \mathbf{B}(\lambda_i)\boldsymbol{\alpha} \end{bmatrix} \tag{9.30}$$

for some vector $\boldsymbol{\alpha}$ of coefficients. In particular, there must be some coefficient vector $\boldsymbol{\alpha}_i$ for which the top part of $\mathbf{P}(\lambda_i)\boldsymbol{\alpha}_i$ equals $\boldsymbol{\psi}_i$. In other words, there must be a vector $\boldsymbol{\alpha}_i$ which satisfies

$$\mathbf{t}_i = \mathbf{T}(\lambda_i)\boldsymbol{\alpha}_i = \boldsymbol{\psi}_i. \tag{9.31}$$

Since $\boldsymbol{\psi}_i$ and $\mathbf{T}(\lambda_i)$ are known, we can solve for $\boldsymbol{\alpha}_i$ from (9.31).

If there are more state variables than inputs ($n > p$), (9.31) will be an overdetermined set of linear equations, but these equations will be consistent (i.e. an $\boldsymbol{\alpha}_i$ will exist which will give equality in (9.31)). In this case we can solve for $\boldsymbol{\alpha}_i$ as

$$\boldsymbol{\alpha}_i = [\mathbf{T}^H(\lambda_i)\mathbf{T}(\lambda_i)]^{-1}\mathbf{T}^H(\lambda_i)\boldsymbol{\psi}_i \quad (n > p). \tag{9.32}$$

If there are fewer state variables than inputs ($n < p$), then (9.31) does not have a unique solution. However, we can choose the min-norm solution given by

$$\boldsymbol{\alpha}_i = [\mathbf{T}(\lambda_i)\mathbf{T}^H(\lambda_i)]^{-1}\mathbf{T}(\lambda_i)\boldsymbol{\psi}_i \quad (n < p). \tag{9.33}$$

If the number of state variables equals the number of inputs, we have

$$\boldsymbol{\alpha}_i = [\mathbf{T}(\lambda_i)]^{-1}\boldsymbol{\psi}_i \quad (n = p). \tag{9.34}$$

We can now use the bottom p rows of (9.30) to compute

$$\mathbf{b}_i = \mathbf{B}(\lambda_i)\boldsymbol{\alpha}_i. \tag{9.35}$$

From (9.31) and (9.35), we have vectors $\mathbf{t}_i$ and $\mathbf{b}_i$ which must be related by

$$\mathbf{b}_i = \mathbf{K}\mathbf{t}_i.$$

The procedure described above is repeated n times ($i = 1, \cdots, n$), once for each of the optimal eigenvectors computed earlier. The $\mathbf{b}_i$ and $\mathbf{t}_i$ vectors are placed as columns into $\mathbf{B}$ and $\mathbf{T}$ matrices, and we can then solve for $\mathbf{L}$ as

$$\mathbf{K} = \mathbf{B}\mathbf{T}^{-1}.$$

An algorithm that takes a set of optimal eigenvectors and finds the corresponding feedback matrix $\mathbf{L}$ is shown in Table 9.6 in Appendix B at the end of this chapter.

We now give an example which shows that the performance of a multivariable regulator which uses the state feedback matrix calculated by the iterative algorithm can be vastly superior to that of a regulator which uses the results of the non-iterative algorithm.

EXAMPLE 9.8

Consider the following two-input, fourth-order model of a chemical reactor described by Munro []

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

where

$$\mathbf{A} = \begin{bmatrix} 1.3800 & -0.2077 & 6.7150 & -5.6760 \\ -0.5814 & -4.2900 & 0 & 0.6750 \\ 1.0670 & 4.2730 & -6.6540 & 5.8930 \\ 0.0480 & 4.2730 & 1.3430 & -2.1040 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 5.6790 & 0 \\ 1.1360 & -3.1460 \\ 1.1360 & 0 \end{bmatrix}. \tag{9.36}$$

The eigenvalues of the $\mathbf{A}$ matrix are

$$-8.6659\,, \quad -5.0566\,, \quad 0.0635\,, \quad 1.9910.$$

Although the plant is unstable, two of its eigenvalues are far into the left half plane. There is no reason to move these eigenvalues with feedback. The desired s-plane pole locations are

$$-8.6659\,, \quad -5.0566\,, \quad -0.5000\,, \quad -0.2000$$

which correspond to a settling time of about 20 seconds.

Suppose we want to design a digital multivariable regulator for this plant with a sampling rate of $T = 0.1$ seconds. The ZOH equivalent is calculated to be

$$\mathbf{\Phi} = \begin{bmatrix} 1.1782 & 0.0015 & 0.5116 & -0.4033 \\ -0.0515 & 0.6619 & -0.0110 & 0.0613 \\ 0.0762 & 0.3351 & 0.5606 & 0.3824 \\ -0.0006 & 0.3353 & 0.0893 & 0.8494 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0045 & -0.0876 \\ 0.4672 & 0.0012 \\ 0.2132 & -0.2353 \\ 0.2131 & -0.0161 \end{bmatrix}.$$

This example The iterative algorithm is initialized by choosing the first null-space vector computed for each of the $\mathbf{P}(\lambda)$ matrices. The feedback gain matrix corresponding to this choice of null-space vectors is denoted $\mathbf{L}_0$ and is calculated to be

$$\mathbf{K}_0 = \begin{bmatrix} 1.1965 & 0.2763 & 0.1598 & 0.7716 \\ 0.3798 & -0.0109 & -0.0233 & 0.6128 \end{bmatrix}.$$

The condition number of the matrix of closed-loop eigenvectors corresponding to $\mathbf{L}_0$ is 157.2.

The iterative algorithm finds a set of closed-loop eigenvectors which are as orthogonal as possible. The condition number of the matrix of optimal closed-loop eigenvectors is 3.34, and the corresponding feedback gain matrix is

$$\mathbf{K}^{opt} = \begin{bmatrix} -0.1746 & 0.0669 & -0.1611 & 0.1672 \\ -1.0794 & 0.0568 & -0.7374 & 0.2864 \end{bmatrix}.$$

The superiority of $\mathbf{K}^{opt}$ over $\mathbf{K}_0$ is now demonstrated in two ways. First, we will show that the optimal solution is indeed more robust to perturbations in the plant model than the initial solution. Second, we will show that the regulator with $\mathbf{K}^{opt}$ has a much better transient response than the regulator with $\mathbf{K}_0$. To illustrate the first point, we need to perturb the $\mathbf{A}$ and $\mathbf{B}$ matrices given in (9.36). Suppose we want to perturb the matrix elements by a random amount between -5% and +5%. These perturbations might represent uncertainty in the mathematical model of the chemical reactor. A simple way to generate these perturbations is to compute two matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ whose elements are random variables which are uniformly distributed in the interval [-0.95,+0.95]. Two such matrices are shown below (their elements have been rounded to two decimal places)

$$\mathbf{P}_1 = \begin{bmatrix} 1.03 & 1.02 & 0.99 & 1.03 \\ 0.95 & 0.95 & 1.02 & 1.00 \\ 0.96 & 0.99 & 1.01 & 0.96 \\ 1.00 & 0.96 & 1.04 & 1.02 \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} 0.99 & 0.98 \\ 1.02 & 0.95 \\ 1.04 & 1.02 \\ 1.03 & 0.98 \end{bmatrix}.$$

Using $\mathbf{P}_1$ and $\mathbf{P}_2$, we can generate perturbed $\mathbf{A}$ and $\mathbf{B}$ matrices as follows

$$\tilde{\mathbf{A}} = \mathbf{A}. * \mathbf{P}_1 \,, \quad \tilde{\mathbf{B}} = \mathbf{B}. * \mathbf{P}_2$$

where the symbol $.*$ refers to element-by-element multiplication of two matrices (i.e. the ij-th element of $\mathbf{A}. * \mathbf{P}_1$ is the ij-th element of $\mathbf{A}$ times the ij-th element of $\mathbf{P}_1$). The ZOH equivalent of $(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})$ is denoted as $(\tilde{\mathbf{\Phi}}, \tilde{\mathbf{\Gamma}})$. The feedback matrices $\mathbf{L}_0$ and $\mathbf{L}^{opt}$ were calculated to place the closed-loop poles at desired locations. We know that the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{L}_0$ and the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{L}^{opt}$ are both equal to the desired closed-loop pole locations. We can now show that the closed-loop poles of the

Desired Poles	Eigenvalues of $(\tilde{\mathbf{\Phi}} - \tilde{\mathbf{\Gamma}}\mathbf{K}^{opt})$	Eigenvalues of $(\tilde{\mathbf{\Phi}} - \tilde{\mathbf{\Gamma}}\mathbf{K}_0)$
.9802	.9635+j.0116	1.1338
.9512	.9635-j.0116	.7172+j.1338
.6031	.6160	.7172-j*.1338
.4204	.4157	.3920

Table 9.2 The desired closed-loop pole locations compared with the closed-loop pole locations of the perturbed system with $\mathbf{K}^{opt}$ and $\mathbf{K}_0$

perturbed system with $\mathbf{L}^{opt}$ are much closer to the desired values than the closed-loop poles of the perturbed system with $\mathbf{L}_0$. This is shown in table 9.2.

From the table, we see that the perturbed system with $\mathbf{K}_0$ is unstable! This means that if the non-iterative algorithm is used to calculate the feedback gains, the coefficients of the state-space system model have to be identified with less than 5% error or the actual closed-loop system could be unstable. The robust feedback matrix, however, will still place the closed-loop poles close to the desired locations even in the face of modeling errors.

There is another way to demonstrate the superiority of $\mathbf{K}^{opt}$ to $\mathbf{K}_0$. That is by comparing the transient responses of the regulators which use these feedback matrices. We will make this comparison using the nominal ZOH equivalent $\mathbf{\Phi}, \mathbf{\Gamma}$. Based on the closed-loop poles that have been selected, we know that the settling time of the regulator to an initial non-zero state vector will be about 20 – 25 seconds (there is a closed-loop pole at -0.2 in the s-plane). Suppose the plant is in the initial state

$$\mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The trajectories of the four state variables are shown in Fig. 9.3. From the figure, we see that the magnitude of the transient response of the regulator with $\mathbf{K}_0$ is about six times as large as that for the regulator with $\mathbf{K}^{opt}$. Thus even if the plant were modeled perfectly so that robustness to modeling uncertainties was not important, it would still be advantageous to use the iterative algorithm to compute the optimal feedback gains. ■

9.6 Multivariable Gain and Phase Margins

Example 9.8 in the previous section showed that two multivariable regulators may have vastly different performance when the plant is perturbed, even though both regulators have the same nominal closed-loop pole locations. One way to measure the sensitivity of a regulator to perturbations in the plant model is to compute the gain and phase margins.

Chapter 5 showed how to compute stability margins for single-input systems using Nyquist plots. Here we define individual gain and phase margins for each input of a multivariable system. These margins are computed by a numerical searching algorithm; a program is included in the Digital Control Toolbox.

A digital multivariable regulator is represented by the state-space model $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{K})$. Each column of $\mathbf{\Gamma}$ is the input vector corresponding to one of the plant inputs. A gain

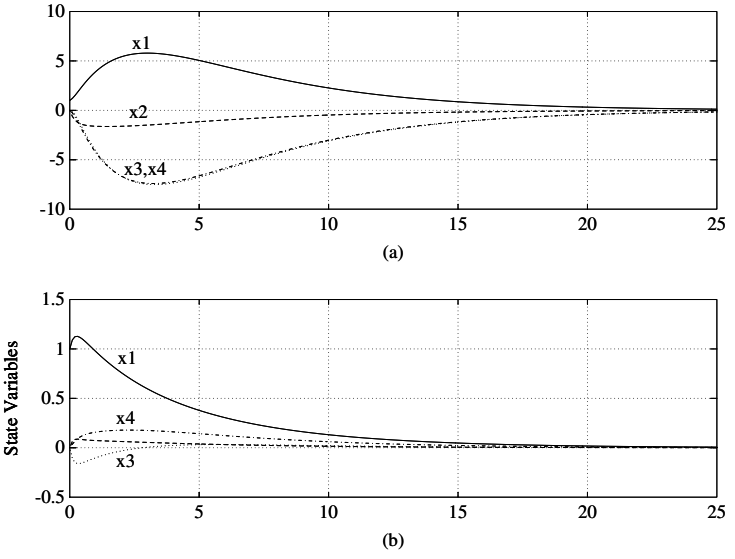


Figure 9.3 (a) Transient responses pf the regulator with $\mathbf{K}_0$. (b) Transient response of the regulator with $\mathbf{K}^{opt}$.

margin is the maximum amount of amplification or attenuation that an input to the plant can tolerate before the closed-loop system becomes unstable. We can change the gain on the i-th input by a factor α, while keeping all the other inputs at their nominal gain, using the following perturbed input matrix:

$$\tilde{\mathbf{\Gamma}}(\alpha, i) \stackrel{\text{def}}{=} \mathbf{\Gamma} \cdot \text{diag}(1, \cdots, \underbrace{\alpha}_{i-\text{th position}}, \cdots, 1).$$

We also define the eigenvalues of the perturbed closed-loop system to be

$$\text{eig}(\mathbf{\Phi} - \tilde{\mathbf{\Gamma}}(\alpha, i)\mathbf{L}) \stackrel{\text{def}}{=} \tilde{\lambda}_1, \cdots, \tilde{\lambda}_n$$

and the maximum magnitude (absolute value) of these eigenvalues to be

$$\gamma = \max_j(\tilde{\lambda}_j) \stackrel{\text{def}}{=} \max(\text{abs}(\text{eig}(\mathbf{\Phi} - \tilde{\mathbf{\Gamma}}(\alpha, i)\mathbf{K}))).$$

When $\alpha = 1$ we know that $\gamma < 1$ because the nominal closed-loop system is stable. However if α is increased from 1, the perturbed closed-loop system could be unstable (upper gain margin). Likewise if $\alpha < 1$ the perturbed closed-loop system could be unstable (lower gain margin). Thus we define the individual upper gain margin for input i of a multivariable system as follows:

$$\text{UGM}_i \stackrel{\text{def}}{=} \min(\alpha > 1 \text{ such that } \max(\text{abs}(\text{eig}(\mathbf{\Phi} - \tilde{\mathbf{\Gamma}}(\alpha, i)\mathbf{K}))) \geq 1)\ ,\ \ i = 1, \cdots p.$$

We stress that the individual gain margin for input i is calculated with all other columns of the input matrix except the i-th held at their nominal values. The algorithm which

computes the upper gain margins starts with $\alpha = 1$ and increases α until the perturbed closed-loop system becomes unstable. The individual lower gain margin for input i of a multivariable system is defined as follows:

$$\text{LGM}_i \stackrel{\text{def}}{=} \max(\alpha < 1 \text{ such that } \max(\text{abs}(\text{eig}(\mathbf{\Phi} - \tilde{\mathbf{\Gamma}}(\alpha, i)\mathbf{K}))) \geq 1) \ , \ \ i = 1, \cdots p.$$

The gain margins can be converted to decibels by $20 \log(\alpha)$.

We define the individual phase margins for each input of a multivariable system in a manner similar to gain margins. A phase margin is the maximum phase lag that an input to the plant can tolerate before the closed-loop system becomes unstable. We can change the phase on the i-th input by θ radians using the following perturbed input matrix:

$$\tilde{\mathbf{\Gamma}}(\theta, i) \stackrel{\text{def}}{=} \mathbf{\Gamma} \cdot \text{diag}(1, \cdots, \underbrace{e^{-j\theta}}_{i-\text{th position}}, \cdots, 1).$$

The the individual phase margin for input i of a multivariable system is defined as follows:

$$\text{PM}_i \stackrel{\text{def}}{=} \min(\theta > 0 \text{ such that } \max(\text{abs}(\text{eig}(\mathbf{\Phi} - \tilde{\mathbf{\Gamma}}(\theta, i)\mathbf{K}))) \geq 1) \ , \ \ i = 1, \cdots p.$$

It is customary to convert the phase margins to degrees. The algorithm which computes the phase margins starts with $\theta = 0$ and increases θ until the perturbed closed-loop system becomes unstable.

EXAMPLE 9.9

Example 9.8 in the previous section calculated two different feedback matrices, $\mathbf{K}_0$ and $\mathbf{K}^{opt}$ to regulate a chemical plant. It was shown that the regulator which used $\mathbf{K}^{opt}$ was less sensitive to plant perturbations than the regulator which used $\mathbf{K}_0$. This observation is confirmed by computing the stability margins for the two regulators. The stability margins for the $\mathbf{K}_0$ regulator are:

Input #1: UGM=0.88 dB, LGM=-1.40 dB, PM=7°

Input #2: UGM=1.54 dB, LGM=-0.44 dB, PM=2°.

These margins are too small for a practical regulation system. The stability margins for the $\mathbf{K}^{opt}$ regulator are much better:

Input #1: UGM=9.84 dB, LGM<-30.1 dB, PM > 128°

Input #2: UGM=18.7 dB, LGM=-0.63 dB, PM=19°.

The program in the Digital Control Toolbox which computes multivariable stability margins only searches for gain margins of up to 30 dB and phase margins of up to 128°. This explains the inequalities shown for the stability margins of the $\mathbf{K}^{opt}$ regulator.

■

9.7 Multivariable Tracking Systems

A procedure to design tracking systems for single-input, single-output plants was presented in Chapter 8. That procedure is extended here to handle the case of multiple-input, multiple-output plants. A summary of the design procedure from Chapter 8 is given below:

1. Calculate the additional dynamics needed for tracking the reference input and rejecting disturbances.
2. Calculate a vector of feedback gains which will regulate the augmented design system consisting of the plant cascaded with the additional dynamics.
3. Introduce the reference input at the appropriate place in the loop.

These same steps are followed in the design of multivariable tracking systems with a few modifications. The modifications are summarized as follows. In Step 1, the additional dynamics are now calculated for a vector of inputs. Thus, the additional dynamics must be replicated q times, once for each output of the plant. Each component of the tracking error vector must be connected to a copy of the additional dynamics. In Step 2, the feedback gains will now be a matrix which can be calculated using the algorithm described in the previous section. Finally in Step 3, q reference inputs are introduced into the loop, each with a feedforward gain. The complete design procedure is given below. We begin by showing state-space models for the plant, the disturbances, the reference input, and the measurements.

Plant

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{E}\boldsymbol{\omega}(t)\\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{F}\boldsymbol{\omega}(t)\end{aligned} \tag{9.37}$$

where $\boldsymbol{\omega}(t)$ is a vector of disturbance signals. Notice that the disturbances can enter the state or output equation (or both) depending on the values of $\mathbf{E}$ and $\mathbf{F}$.

Disturbances This design method assumes that the disturbance vector $\boldsymbol{\omega}(t)$ is the output of a homogeneous (zero-input) state-space equation

$$\begin{aligned}\dot{\mathbf{z}}_d(t) &= \mathbf{A}_d\mathbf{z}_d(t)\\ \boldsymbol{\omega}(t) &= \mathbf{C}_d\mathbf{z}_d(t).\end{aligned} \tag{9.38}$$

Reference Inputs The reference input vector is also assumed to be the output of a homogeneous state-space model

$$\begin{aligned}\dot{\mathbf{z}}_r(t) &= \mathbf{A}_r\mathbf{z}_r(t)\\ \mathbf{r}(t) &= \mathbf{C}_r\mathbf{z}_r(t).\end{aligned} \tag{9.39}$$

Measurements The signals that are measured and available for the control system to use are assumed to be a combination of plant state variables possibly corrupted by the disturbance

$$\mathbf{y}_m(t) = \mathbf{C}_m\mathbf{x}(t) + \mathbf{F}_m\boldsymbol{\omega}(t). \tag{9.40}$$

We make the following assumption regarding $\mathbf{C}_m$ and $\mathbf{F}_m$ from (9.40) and $\mathbf{C}$ and $\mathbf{F}$ from (9.37):

Assumption There exists a matrix $\mathbf{C}_0$ which satisfies

$$\mathbf{C}_0\mathbf{C}_m = \mathbf{C} \text{ and } \mathbf{C}_0\mathbf{F}_m = \mathbf{F}. \tag{9.41}$$

The meaning of this assumption is given in the second remark below.

Remarks

1. The tracking system derived in this section can be used with reference inputs that do *not* satisfy the assumed model (9.39), and with disturbances that do not satisfy the model in (9.38). In this case, however, the tracking may not be perfect (output equals reference input), even asymptotically.

2. The actual output that should track the reference trajectory is the output $\mathbf{y}$ of the plant in (9.37). We assume that the measurement vector $\mathbf{y}_m$ contains the actual output in the sense that there exists a matrix $\mathbf{C}_0$ such that

   $$\mathbf{y} = \mathbf{C}_0\mathbf{y}_m. \tag{9.42}$$

   Thus the output we want to control can be obtained from the measured signals. If this condition is *not* satisfied then we have no way of insuring that $\mathbf{y}$ is tracking the reference input. Using (9.37) and (9.40), we can re-write (9.42) as follows

   $$\mathbf{C}\mathbf{x}(t) + \mathbf{F}\omega(t) = \mathbf{C}_0[\mathbf{C}_m\mathbf{x}(t) + \mathbf{F}_m\omega(t)]. \tag{9.43}$$

   Since this equation must be true for all values of $\mathbf{x}(t)$ and $\boldsymbol{\omega}(t)$, we can equate the coefficients of $\mathbf{x}(t)$ and $\boldsymbol{\omega}(t)$ separately to obtain the two equations given in (9.41).

   The relationships between $\mathbf{C}$, $\mathbf{C}_m$, and $\mathbf{C}_0$ are given below for three common cases – when all the state variables are measured, when the measurements only depend on a r state variables, and when the measurement consists only of the plant output. In the first case, suppose the output equation for the plant is

   $$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)$$

   and the measurement vector is $\mathbf{y}_m = \mathbf{x}$ (i.e. $\mathbf{C}_m = \mathbf{I}$). These measurement contain the output since (8.44) is satisfied with $\mathbf{C}_0 = \mathbf{C}$.

   In the second case the output of the plant depends only on a subset of state variables, say the first r, and we measure only these state variables. In this case, the output matrix for the plant is of the form $\mathbf{C} = [\mathbf{C}_1\ \mathbf{0}]$ and the measurements are described by

   $$\mathbf{y}_m = \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x}.$$

   In this case, (8.44) is satisfied with $\mathbf{C}_0 = \mathbf{C}$.

   A third possibility is that we only measure the output, $\mathbf{y}_m = \mathbf{y}$. In this case, (8.44) is satisfied with $\mathbf{C}_0 = \mathbf{I}$.

3. The disturbance can affect the plant and/or the measurements. If $\mathbf{C}_m = \mathbf{I}$ and $\mathbf{F}_m = \mathbf{0}$, then the measurement vector $\mathbf{y}_m = \mathbf{x}$. In other words, we are measuring all the state variables. In this case, if $\mathbf{F}_m \neq \mathbf{0}$, we are measuring the state variables subject to some disturbance, but this disturbance is not necessarily affecting the plant. The disturbance only affects the plant if $\mathbf{E}$ or $\mathbf{F}$ is nonzero.

With the above models defined, we are ready to describe the design procedure, and to state the properties of the resulting control system. The design procedure consists of four steps which are described below. After explaining each step, the properties of the resulting control system will be given.

9.7.1 Algorithm to Design Multivariable Tracking Systems

Step 1: Calculate the necessary additional dynamics The additional dynamics must contain the poles each of the reference inputs and each of the disturbances. These poles are the eigenvalues of A_r and A_d which come from the above models for the reference input and disturbance. Let Λ_d and Λ_r be sets describing the eigenvalues (with multiplicities) of A_d and A_r, respectively. That is, let

$$\Lambda_d = \{(\lambda_{d_1}, m_{d_1}), (\lambda_{d_2}, m_{d_2}), \cdots\}$$

$$\Lambda_r = \{(\lambda_{r_1}, m_{r_1}), (\lambda_{r_2}, m_{r_2}), \cdots\}$$

where λ_{d_i} is an eigenvalue of A_d with multiplicity m_{d_i}, and the set Λ_r is similarly defined.

In the first part of step 1, we define the set Λ to be the union of the sets Λ_d and Λ_r subject to the following restriction. If a common eigenvalue appears in Λ_d and Λ_r, we include only one of these in Λ, the one with the higher multiplicity. The next part of step 1 is to map the eigenvalues in Λ using the ZOH pole-mapping formula and multiply them out to form the denominator polynomial for the additional dynamics. If Λ contains eigenvalues λ_i with multiplicity m_i, and if the total number of eigenvalues is $s \stackrel{\text{def}}{=} \sum m_i$, then the denominator polynomial of the additional dynamics is calculated as

$$\delta(z) = \prod_i (z - e^{\lambda_i T})^{m_i} \stackrel{\text{def}}{=} z^s + \delta_1^{s-1} + \cdots + \delta_s. \tag{9.44}$$

The last part of step 1 is to use the denominator polynomial $\delta(z)$ to define the state-space matrices for the additional dynamics. These matrices are shown below

$$\mathbf{\Phi}_a = \begin{bmatrix} -\delta_1 & 1 & 0 & \cdots & 0 \\ -\delta_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\delta_{s-1} & 0 & 0 & \cdots & 1 \\ -\delta_s & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \mathbf{\Gamma}_a = \begin{bmatrix} -\delta_1 \\ -\delta_2 \\ \vdots \\ -\delta_{s-1} \\ -\delta_s \end{bmatrix}. \tag{9.45}$$

A summary of step 1, as well as the remaining steps of the design procedure can be found in Table 9.3.

Step 2: Replicate the additional dynamics The additional dynamics system from Step 1 must be replicated into q parallel systems. The state-space matrices which describe the

additional dynamics are $(\bar{\mathbf{\Phi}}, \bar{\mathbf{\Gamma}})$ which are defined as follows:

$$\begin{aligned}
\bar{\mathbf{\Phi}} &\stackrel{\text{def}}{=} \text{block diagonal}(\underbrace{\mathbf{\Phi}_a, \cdots, \mathbf{\Phi}_a}_{q \text{ matrices}}) \\
&= \begin{bmatrix} \mathbf{\Phi}_a & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi}_a & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{\Phi}_a \end{bmatrix} \\
\bar{\mathbf{\Gamma}} &\stackrel{\text{def}}{=} \text{block diagonal}(\underbrace{\mathbf{\Gamma}_a, \cdots, \mathbf{\Gamma}_a}_{q \text{ matrices}}) \\
&= \begin{bmatrix} \mathbf{\Gamma}_a & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{\Gamma}_a & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{\Gamma}_a \end{bmatrix}.
\end{aligned}$$

Step 3: Calculate a feedback matrix for the design model The design model consists of a cascade combination of the ZOH of the plant and the additional dynamics. The state-space matrices for the design model are

$$\mathbf{\Phi}_d = \begin{bmatrix} \mathbf{\Phi} & \mathbf{0} \\ \bar{\mathbf{\Gamma}}\mathbf{C} & \bar{\mathbf{\Phi}} \end{bmatrix}, \quad \mathbf{\Gamma}_d = \begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{x}_d[k] = \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{x}_a[k] \end{bmatrix}. \tag{9.46}$$

The design model has dimension $n + q \cdot s$, where n is the dimension of the plant, s is the dimension of the additional dynamics from Step 1, and q is the number of outputs. A feedback matrix $\mathbf{L}$ for the design model will have $n+q\cdot s$ columns, and it can be partitioned into the first n and last $q \cdot s$ columns as follows

$$\mathbf{K} = [\, \mathbf{K}_1 \quad \mathbf{K}_2 \,].$$

Step 3: Introduce the reference input and implement the additional dynamics as part of the compensator All of the calculations have been performed in the previous steps and this step merely consists of putting the pieces together. The digital control system is shown in Fig. 9.4. This system shows the measurement vector $\mathbf{y}_m = \mathbf{x}$, i.e. full state feedback. If the full state vector cannot be measured, an observer can be used. This will be demonstrated in some of the following examples.

Remarks

- The control system shown in Fig. 9.4 has a feedforward gain matrix $\mathbf{G}$ from the reference input to the plant input. This matrix cannot affect the stability of the control system because it does not alter the closed-loop poles. However, this matrix may improve the transient response of the system. $\mathbf{G}$ can be a diagonal matrix of gains whose values are chosen on the basis of simulations.

- The tracking system shown in Fig. 9.4 has the property that the plant output vector $\mathbf{y}[k]$ approaches the reference input $\mathbf{r}[k]$ as k gets large. Tracking will be achieved in spite of disturbances and in spite of an inaccurate model of the plant, as long as

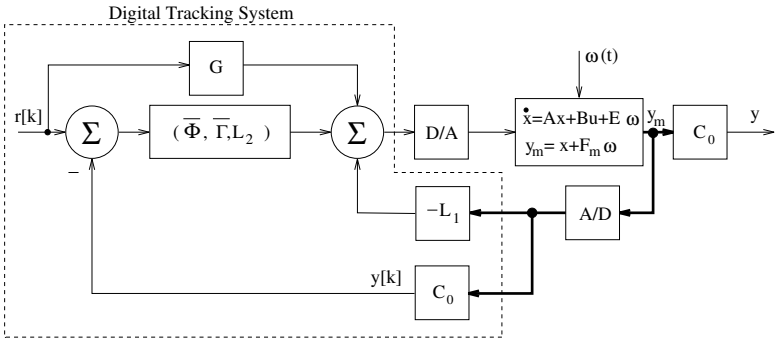


Figure 9.4 A multivariable digital tracking system that uses digital state feedback.

1. Calculate the necessary additional dynamics. The additional dynamics must contain the poles of the reference input and the disturbance vectors. The poles are the eigenvalues of A_r and A_d which come from the above models for the reference input and disturbance. Map the eigenvalues using the ZOH pole-mapping formula and multiply them out to form the denominator polynomial for the additional dynamics. The denominator polynomial of the additional dynamics is calculated as

$$\delta(z) = \prod_i (z - e^{\lambda_i T})^{m_i} \stackrel{\text{def}}{=} z^s + \delta_1 z^{s-1} + \cdots + \delta_s.$$

   Use the denominator polynomial $\delta(z)$ to define the state-space matrices for the additional dynamics. These matrices are shown below

$$\mathbf{\Phi}_a = \begin{bmatrix} -\delta_1 & 1 & 0 & \cdots & 0 \\ -\delta_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\delta_{s-1} & 0 & 0 & \cdots & 1 \\ -\delta_s & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \mathbf{\Gamma}_a = \begin{bmatrix} -\delta_1 \\ -\delta_2 \\ \vdots \\ -\delta_{s-1} \\ -\delta_s \end{bmatrix}.$$

2. Replicate the additional dynamics into q parallel systems described by the state-space matrices $(\bar{\mathbf{\Phi}}, \bar{\mathbf{\Gamma}})$.

3. Calculate a feedback vector for the design model. The design model consists of a cascade combination of the ZOH of the plant and the replicated additional dynamics. The state-space matrices for the design model are

$$\mathbf{\Phi}_d = \begin{bmatrix} \mathbf{\Phi} & 0 \\ \bar{\mathbf{\Gamma}}\mathbf{C} & \bar{\mathbf{\Phi}} \end{bmatrix}, \quad \mathbf{\Gamma}_d = \begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{x}_d[k] = \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{x}_a[k] \end{bmatrix}.$$

   A feedback vector $\mathbf{L}$ for the design model will have $n + s \cdot q$ columns, and it can be partitioned into the first n and last q columns as follows

$$\mathbf{K} = [\, \mathbf{K}_1 \quad \mathbf{K}_2 \,].$$

4. Introduce the reference input and implement the additional dynamics as part of the compensator.

Table 9.3 Procedure to design a multivariable digital tracking system.

the closed-loop system with the actual plant remains stable. We assume here that the model of the plant is accurate enough so that the feedback vector calculated for the design model will stabilize the actual plant.

- It is important to assess the robustness of the tracking system shown in Fig. 9.4. Stability margins for multivariable tracking systems are calculated in the same way as for multivariable regulators. That is, a state-space model for the loop transfer function for the system of Fig. 9.4 is obtained with the loop broken at the input to the plant. The columns of the input matrix are perturbed using gain or phase perturbations to obtain individual gain and phase margins. (See Section 9.6.)

EXAMPLE 9.10

In this example we design a multivariable tracking system for an experimental aircraft described in [21, 78]. The plant model represents the longitudinal dynamics as perturbed from steady-state straight and level trim conditions, with a nominal velocity of 881 ft/s. A state-space model is shown below:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ -32.1 & -0.0822 & 0.0472 & -19.7 \\ -0.705 & -0.0558 & -1.68 & 898 \\ 0 & -0.00317 & 0.0303 & -0.253 \end{bmatrix} \mathbf{x}(t)$$

$$+ \begin{bmatrix} 0 & 0 & 0 \\ 20.57 & -2.7459 & 0.125 \\ -36.33 & -115 & -4.263 \times 10^{-3} \\ 30.195 & -3.724 & -2.773 \times 10^{-4} \end{bmatrix} \mathbf{u}(t)$$

$$\mathbf{y}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 2.45 \times 10^{-5} & -1.12 \times 10^{-3} & 0 \\ 0 & 1 & 0.0219 & 0 \end{bmatrix} \mathbf{x}(t).$$

In what follows, all angles are measured in degrees. The state variables are

$$x_1 = \theta = \text{roll angle}$$

$$x_2 = u = \text{perturbation velocity in } x \text{ direction}$$

$$x_3 = w = \text{velocity in } z \text{ direction}$$

$$x_4 = q = \text{roll rate.}$$

The inputs are

$$u_1 = \delta_c = \text{canard angle}$$

$$u_2 = \delta_f = \text{flaperon angle}$$

$$u_3 = \delta_t = \text{thrust (pounds).}$$

The outputs to be controlled are

$$y_1 = \theta = \text{pitch angle}$$

$$y_2 = \gamma = \text{flight-path angle}$$

$$y_3 = v = \text{aircraft velocity (ft/s)}.$$

The desired settling time for a step change in any output is $T_S = 2$ seconds. For this example we choose a sampling interval of $T = T_S/100 = 0.02$ seconds. The ZOH equivalent for this system is

$$\mathbf{\Phi} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0200 \\ -0.6415 & 0.9984 & 0.0008 & -0.3914 \\ -0.0134 & -0.0017 & 0.9723 & 17.6489 \\ 0.0000 & -0.0001 & 0.0006 & 1.0003 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0060 & -0.0007 & -0.0000 \\ 0.2923 & -0.0413 & 0.0025 \\ 4.6425 & -2.9267 & -0.0001 \\ 0.6032 & -0.0751 & 0.0000 \end{bmatrix}.$$

A multivariable tracking system is designed using the steps shown in Table 9.3.

Step 1 In order to track step inputs the matrix $\mathbf{A}_r = 0$. The pole at $s = 0$ maps into the z-plane as a pole at $z = 1$. The polynomial $\delta(z) = z - 1$ and the additional dynamics are

$$\mathbf{\Phi}_a = 1\,, \quad \mathbf{\Gamma}_a = 1.$$

Step 2 Because the plant has three outputs, we replicate the additional dynamics into three parallel systems described by the following model

$$\bar{\mathbf{\Phi}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \bar{\mathbf{\Gamma}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Step 3 The design model is the cascade combination of the ZOH of the plant and the replicated additional dynamics. The state-space matrices for the design model are

$$\mathbf{\Phi}_d = \begin{bmatrix} \mathbf{\Phi} & 0 \\ \bar{\mathbf{\Gamma}}\mathbf{C} & \bar{\mathbf{\Phi}} \end{bmatrix}, \quad \mathbf{\Gamma}_d = \begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{0} \end{bmatrix}.$$

In order to choose the closed-loop poles, we first compute the eigenvalues of the $\mathbf{A}$ matrix of the plant. These eigenvalues are

$$-6.2364\,, \quad 4.3068\,, \quad -0.0428 \pm j0.0806.$$

The pole at $s = -6.2364$ is well damped, and we will keep it as a closed-loop pole. We add damping to the complex conjugate poles to achieve a settling time of 2 seconds. These desired closed-loop poles are at

$$-4.6/2 \pm 0.0806.$$

Since the design model is 7th order, there are 4 more closed-loop poles to choose. We will put them on the real axis at the following locations (corresponding to a settling time of about 2 seconds):

$$-2.3\,, \quad -2.4\,, \quad -2.5\,, \quad -2.6.$$

Thus the complete set of s-plane closed-loop poles is

$$-6.2364\,,\quad -2.3 \pm j0.0806\,,\quad -2.3\,,\quad -2.4\,,\quad -2.5\,,\quad -2.6.$$

These poles are mapped into the z-plane using the ZOH pole-mapping formula to obtain the following z-plane desired closed-loop poles:

$$0.8827\,,\quad 0.9550 \pm j0.0015\,,\quad 0.9550\,,\quad 0.9531\,,\quad 0.9512\,,\quad 0.9493.$$

The iterative pole-placement algorithm described in the previous section computes the following feedback gain matrix given $(\mathbf{\Phi}_d, \mathbf{\Gamma}_d)$ and the desired closed-loop poles given above:

$$\mathbf{K} = \begin{bmatrix} 1.3463 & -0.0014 & -0.0020 & -0.5137 & -0.0635 & 0.0901 & -0.0001 \\ 2.1641 & -0.0199 & -0.0256 & -6.8348 & -0.7513 & 0.7937 & -0.0009 \\ -381.1719 & 36.5171 & 0.1429 & -209.7011 & -21.5404 & 19.0491 & 0.8637 \end{bmatrix}.$$

This matrix is partitioned into two matrices. $\mathbf{K}_1$ consists of the first 3 columns of $\mathbf{K}$, and $\mathbf{K}_2$ consists of the last 4 columns of $\mathbf{K}$.

The stability margins for this tracking system are calculated to be:

Input #1: UGM=3.82 dB, LGM=-19.9 dB, PM=27°

Input #2: UGM=12.2 dB, LGM=-3.35 dB, PM=38°.

Input #3: UGM=26.8 dB, LGM<-30.1 dB, PM=74°.

These are excellent stability margins, and we now check the transient response for three different maneuvers of the aircraft. The first maneuver is called pitch pointing. The aircraft is flying straight and level at a given velocity, and a step input of 2° in the pitch angle y_1 is given while the other two outputs are commanded to remain unchanged. Thus the reference input vector is

$$\mathbf{r}[k] = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}\,,\ k = 0, 1, \cdots.$$

The responses and inputs are shown in Fig. 9.5. The pitch angle response has no overshoot. Also the system is nicely decoupled: the maximum flight path angle perturbation (y_2) is 0.13°, and the maximum velocity perturbation (y_3) 0.12 ft/s from a nominal velocity of 881 ft/s.

The second maneuver is called vertical translation. The pitch angle and velocity should remain constant, while the aircraft rises vertically (flight path angle increases 2°). Thus the reference input vector is

$$\mathbf{r}[k] = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}\,,\ k = 0, 1, \cdots.$$

The responses and inputs are shown in Fig. 9.6.

The flight path angle response has no overshoot, and there is very little coupling to the other two outputs.

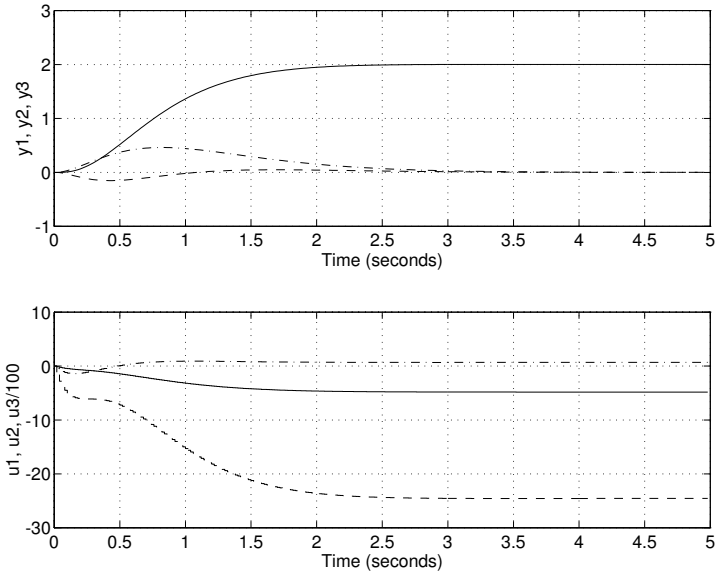


Figure 9.5 Outputs and inputs for Example 9.5 for the pitch pointing maneuver. y_1 and u_1 are shown by solid lines, y_2 and u_2 by dashed lines, and y_3 and u_3 by dash-dot lines.

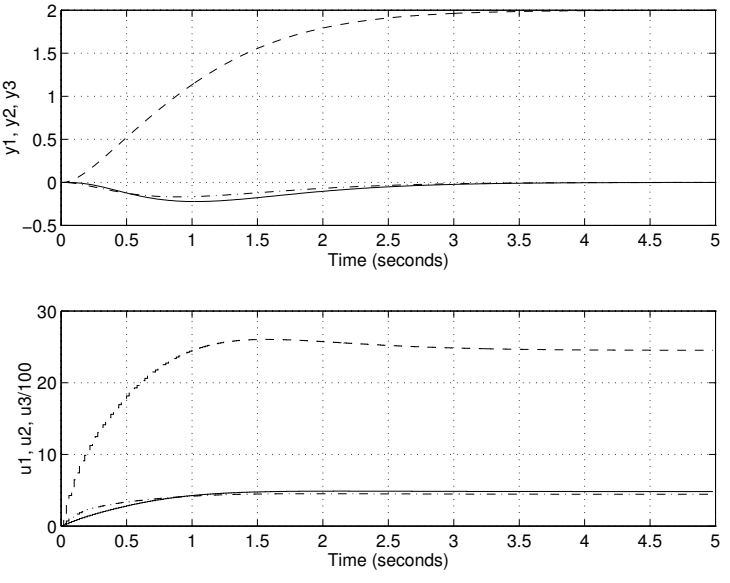


Figure 9.6 Outputs and inputs for Example 9.6 for the vertical translation maneuver. y_1 and u_1 are shown by solid lines, y_2 and u_2 by dashed lines, and y_3 and u_3 by dash-dot lines.

The final maneuver is called straight climb. The pitch and flight path angles should simultaneously increase 2° while holding velocity constant. The reference input vector is

$$\mathbf{r}[k] = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}, \; k = 0, 1, \cdots.$$

The responses and inputs are shown in Fig. 9.7. The pitch and flight path angle responses are identical, and the velocity remains essentially constant.

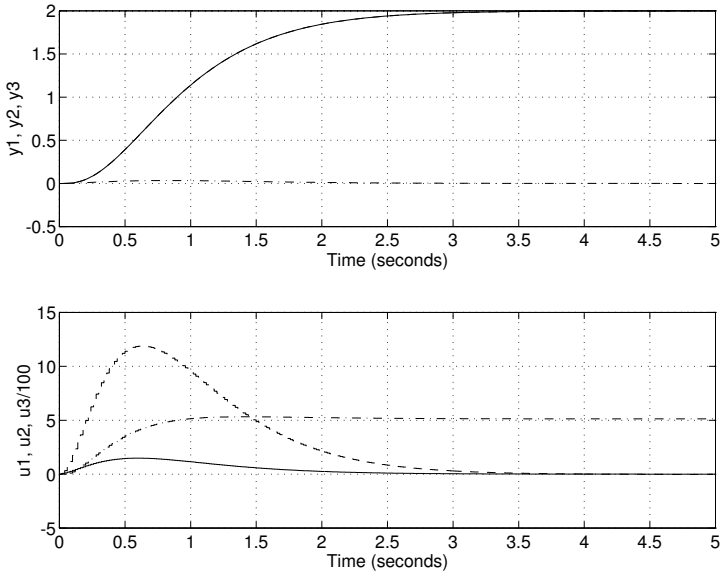


Figure 9.7 Outputs and inputs for Example 9.5 for the straight climb maneuver. y_1 and u_1 are shown by solid lines, y_2 and u_2 by dashed lines, and y_3 and u_3 by dash-dot lines.

9.8 Chapter Summary

In this chapter we presented state-space methods for multivariable systems. We gave an algorithm to compute a state-space model for a multivariable system described by differential equations. We derived an algorithm for multivariable pole placement. The feedback matrix $\mathbf{L}$ is not uniquely determined by the choice of closed-loop poles. Thus the pole-placement algorithm iterates to find the matrix $\mathbf{K}$ which makes the eigenvectors of $\mathbf{\Phi} - \mathbf{\Gamma K}$ as orthogonal as possible. This orthogonality makes the closed-loop system insensitive to perturbations of the plant model.

Stability margins can be defined for multivariable regulators by considering perturbations on individual plant inputs. The largest perturbation (gain or phase) that the system can tolerate before the closed-loop system becomes unstable yields the individual stability margins for that input.

We presented an algorithm to design multivariable tracking systems. Additional dynamics must be used to achieve zero steady-state tracking error. The additional dynamics

are implemented as part of the compensator, and these dynamics must be replicated into q parallel systems, one for each reference input. A multivariable tracking system uses a feedback matrix $\mathbf{K}$ which is calculated for the cascade combination of the ZOH of the plant and the additional dynamics. Tracking systems can use full-state feedback, or an observer can be used to estimate state variables that are not measured. In either case, the stability margins of a tracking system are found from the appropriate loop transfer function.

9.9 Appendix A: Derivation of PBH Controllability Test*

A fact from linear algebra is that a matrix and its transpose have the same rank. Thus in what follows, we remember that the rank of $\mathbf{W}_c^T$, is the same as the rank of $\mathbf{W}_c$. To establish the PBH controllability test (Fact 9.2), consider the following equation

$$\mathbf{W}_c^T \mathbf{s} = \mathbf{0} \tag{9.47}$$

where $\mathbf{s}$ is an n-vector. Since $\mathbf{W}_c^T$ has n columns, and a fact from linear algebra is that the rank plus the nullity of a matrix equals the number of columns, we have the following: the rank of $\mathbf{W}_c^T = n$ if and only if the nullity of $\mathbf{W}_c^T = 0$. In other words, the rank of $\mathbf{W}_c^T = n$ if and only if $\mathbf{s} = \mathbf{0}$ is the only solution to (9.47). If (9.47) has a nonzero solution, this means that the rank of $\mathbf{W}_c$ is less than n and so the system is not controllable.

Now consider another equation

$$\begin{bmatrix} \mathbf{B}^T \\ \lambda \mathbf{I} - \mathbf{A}^T \end{bmatrix} \mathbf{s} = \mathbf{0} \tag{9.48}$$

where λ is an arbitrary complex number. We claim that (9.48) has a nonzero solution if and only if (9.47) has a nonzero solution. We will first prove this claim, and then show how it gives a new test for controllability. To begin with, assume that $\mathbf{s}_1 \neq \mathbf{0}$ is a solution to (9.48). Thus

$$\mathbf{B}^T \mathbf{s}_1 = \mathbf{0} \text{ and } \mathbf{A}^T \mathbf{s}_1 = \lambda \mathbf{s}_1. \tag{9.49}$$

Using the definition of the controllability matrix, (9.47) is equivalent to

$$\mathbf{B}^T (\mathbf{A}^T)^i \mathbf{s} = \mathbf{0}, \;\; i = 0, 1, \cdots, n-1. \tag{9.50}$$

If we evaluate (9.50) at $\mathbf{s} = \mathbf{s}_1$ and use (9.49) we get

$$\begin{aligned} \mathbf{B}^T ((\mathbf{A}^T)^i \mathbf{s}_1) &= \mathbf{B}^T (\lambda^i \mathbf{s}_1) \\ &= \lambda^i B^T \mathbf{s}_1 \\ &= \mathbf{0}. \end{aligned}$$

Thus we have shown that if $\mathbf{s}_1$ is a nonzero solution to (9.48), then it also is a nonzero solution to (9.47).

We now want to show the reverse implication, namely that the existence of a nonzero solution to (9.47) implies the existence of a nonzero solution to (9.48). This implication is slightly more difficult to prove because it turns out that all nonzero solutions to (9.47) are not necessarily solutions to (9.48). We proceed as follows. Assume that (9.47) has a nonzero solution. This implies that $\mathbf{W}_c^T$ has rank r less than n and so has a nontrivial null

space of dimensions $n - r$. Let the r columns of the matrix $\mathbf{U}_1$ be an orthonormal basis for the row space of $\mathbf{W}_c^T$, and let the $n - r$ columns of the matrix $\mathbf{U}_2$ be an orthonormal basis for the null space of $\mathbf{W}_c^T$. Recall that the null space of a matrix is the same as the orthogonal complement of the row space. Thus the columns of $\mathbf{U}_2$ are orthogonal to the columns on $\mathbf{U}_1$. Because the columns of both $\mathbf{U}_1$ and $\mathbf{U}_2$ are orthonormal, we have the following relationships

$$\begin{aligned} \mathbf{U}_1^T\mathbf{U}_2 &= 0 \\ \mathbf{U}_1^T\mathbf{U}_1 &= \mathbf{I}_r \\ \mathbf{U}_2^T\mathbf{U}_2 &= \mathbf{I}_{n-r}. \end{aligned} \tag{9.51}$$

Then if we let

$$\mathbf{s} = \mathbf{U}_2\mathbf{z} \tag{9.52}$$

for an arbitrary vector $\mathbf{z} \neq \mathbf{0}$, the vector $\mathbf{s}$ will be a nonzero solution to (9.47), since $\mathbf{W}_c^T\mathbf{U}_2 = 0$ by the definition of $\mathbf{U}_2$. We will see that for certain choices of the vector $\mathbf{z}$, the vector $\mathbf{s}$ defined by (9.52) will also be a solution to (9.48).

Assume that $\mathbf{s}_2$ is a nonzero solution to (9.47) corresponding to the vector $\mathbf{z}_2$; that is

$$\mathbf{s}_2 = \mathbf{U}_s\mathbf{z}_2. \tag{9.53}$$

Then we know that $\mathbf{B}^T\mathbf{s}_2 = \mathbf{0}$, and so in order for $\mathbf{s}_2$ to be a solution to (9.48), we just need to show that $\mathbf{A}^T\mathbf{s}_2 = \lambda\mathbf{s}_2$; in other words, we need to show that $\mathbf{s}_2$ is an eigenvector of $\mathbf{A}^T$. We want to derive conditions on $\mathbf{z}_2$ such that $\mathbf{s}_2 = \mathbf{U}_2\mathbf{z}_2$ will be an eigenvector of $\mathbf{A}^T$. We claim that the required condition is that $\mathbf{z}_2$ must be an eigenvector of $\mathbf{U}_2^T\mathbf{A}^T\mathbf{U}_2$, or

$$\mathbf{U}_2^T\mathbf{A}^T\mathbf{U}_2\mathbf{z}_2 = \lambda\mathbf{z}_2 \tag{9.54}$$

for some eigenvalue λ. We now show that this choice for $\mathbf{z}_2$ works.

If (9.54) is satisfied, then we have the following result

$$\begin{aligned} \begin{bmatrix} \mathbf{U}_1^T \\ \mathbf{U}_2^T \end{bmatrix} \mathbf{A}^T \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{z}_2 \end{bmatrix} &= \begin{bmatrix} \mathbf{U}_1^T \\ \mathbf{U}_2^T \end{bmatrix} \mathbf{A}^T\mathbf{U}_2\mathbf{z}_2 \\ &= \begin{bmatrix} \mathbf{U}_1^T\mathbf{A}^T\mathbf{U}_2\mathbf{z}_2 \\ \mathbf{U}_2^T\mathbf{A}^T\mathbf{U}_2\mathbf{z}_2 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} \\ \lambda\mathbf{z}_2 \end{bmatrix}. \end{aligned} \tag{9.55}$$

In the last equality shown above, the fact that the top part of the vector equals zero needs to be shown, and we will do so shortly; the fact that the bottom part equals $\lambda\mathbf{z}_2$ is a result of (9.54). From (9.51) we know that

$$\begin{bmatrix} \mathbf{U}_1^T \\ \mathbf{U}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} = \mathbf{I} \text{ or } \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{U}_1^T \\ \mathbf{U}_2^T \end{bmatrix}^{-1}$$

which means that

$$\begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{U}_1^T \\ \mathbf{U}_2^T \end{bmatrix} = \mathbf{I}. \tag{9.56}$$

Thus if we multiply (9.55) on the left by $[\mathbf{U}_1\ \mathbf{U}_2]$ and use (9.56) we get

$$\mathbf{A}^T [\,\mathbf{U}_1 \quad \mathbf{U}_2\,] \begin{bmatrix} \mathbf{0} \\ \mathbf{z}_2 \end{bmatrix} = [\,\mathbf{U}_1 \quad \mathbf{U}_2\,] \begin{bmatrix} \mathbf{0} \\ \lambda \mathbf{z}_2 \end{bmatrix}$$

or

$$\mathbf{A}^T \mathbf{U}_2 \mathbf{z}_2 = \lambda \mathbf{U}_2 \mathbf{z}_2.$$

Using (9.53), the above equation becomes

$$\mathbf{A}^T \mathbf{s}_2 = \lambda \mathbf{s}_2$$

which is what we wanted to show.

To finish the proof, we need to verify the top part of the last equality in (9.55). From the definition of $\mathbf{W}_c$, we see that the rows of the matrix $\mathbf{W}_c^T \mathbf{A}^T$ are also rows of $\mathbf{W}_c^T$, except for the last block of rows in $\mathbf{B}^T (\mathbf{A}^T)^n$. However by the Cayley-Hamilton theorem the matrix $\mathbf{A}^n$ can be expressed as a linear combination of lower powers of $\mathbf{A}$. Thus the rows of $\mathbf{B}^T (\mathbf{A}^T)^n$ can be expressed as a linear combination of the rows of $\mathbf{W}_c$. The result is that the row space of $\mathbf{W}_c^T \mathbf{A}^T$ is contained in the row space of $\mathbf{W}_c^T$. Since the columns of $\mathbf{U}_1$ are a basis for the row space of $\mathbf{W}_c^T$ this means that the row space of $\mathbf{U}_1^T \mathbf{A}^T$ is contained in the row space of $\mathbf{W}_c^T$. Thus if we project the rows of $\mathbf{U}_1^T \mathbf{A}^T$ onto the row space of $\mathbf{W}_c^T$, they are unchanged. The projection of the rows of $\mathbf{U}_1^T \mathbf{A}^T$ is accomplished by multiplying on the right by $\mathbf{U}_1 \mathbf{U}_1^T$.

$$\mathbf{U}_1^T \mathbf{A}^T \mathbf{U}_1 \mathbf{U}_1^T = \mathbf{U}_1^T \mathbf{A}^T.$$

If we substitute this equation into (9.55) and use (9.51) we obtain the desired result

$$\mathbf{U}_1^T \mathbf{A}^T \mathbf{U}_2 \mathbf{z}_2 = \mathbf{U}_1^T \mathbf{A}^T \mathbf{U}_1 \mathbf{U}_1^T \mathbf{U}_2 \mathbf{z}_2 = \mathbf{0}$$

because $\mathbf{U}_1^T \mathbf{U}_2 = 0$.

Thus we have proved that (9.47) has a nonzero solution if and only if (9.48) has a nonzero solution. Recall that the existence of a nonzero solution to (9.47) is a necessary and sufficient condition for a system to be uncontrollable because W_c must have rank n for the system to be controllable. From the equivalence just established between (9.47) and (9.48), we see that the existence of a nonzero solution to (9.48) is also a necessary and sufficient condition for a system to be uncontrollable. This means that if (9.48) does *not* have a nonzero solution, the system must be controllable. Thus we have established the PBH controllability test.

9.10 Appendix B: Multivariable Pole Placement Algorithm*

Input $\mathbf{\Phi}$, $\mathbf{\Gamma}$, $\mathbf{p}$ and $\mathbf{q}$;

Let np = length($\mathbf{p}$);

Let **cv** be a vector with a 1 in the i-th position if $\mathbf{p}(i)$ is real,
and a 0 if $\mathbf{p}(i)$ is complex;

Let AT, ATT, and X be a null matrices;

For $i = 1$ to np

Compute T = an orthogonal basis for the null-space of
$[\,(\mathbf{p}(i)I - \mathbf{\Phi}) \quad \mathbf{\Gamma}\,]$;

Compute TT = an orthogonal basis for the columns in the top
part of T (i.e. the first n rows of T);

Append the matrix AT with the columns of T;

Append the matrix ATT with the columns of TT;

Append the matrix X with the first $\mathbf{q}(i)$ columns of TT;

End

Let $CD = 10^{15}$;

Let Xt = X;

For k = 1 to 5

Let $X1$ = a null matrix;

Let kk = 0;

For i = 1 to np

Let $\mathbf{U}$ = columns $(i-1) * m + 1$ to $i * m$ of AT;

Let Pr = $\mathbf{U} * \mathbf{U}^H$; % form projection matrix $\mathcal{S}_i$ (9.29)

For j = 1 to $\mathbf{q}$(i)

Let $kk = kk + 1$;

Let S = all columns of Xt except the i-th;

· · · continue on next page · · ·

Table 9.4 An algorithm to compute the optimal closed-loop eigenvectors. Text to the right of the symbol % are comments. Numbers in parenthesis refer to equation numbers from the preceding development.

```
        Append S with the complex conjugate of S;
        If cv(kk) = 0   % if p(kk) is complex, also orthogonalize
                          against the conjugate of the kk-th eigenvector
            Append S with the complex conjugate of the kk-th
                column of Xt;
        End
        if kk = 1 and np = 1   % handle the case of a 2nd order
                                 system with complex poles
            Let S = complex conjugate of Xt;
        End
        Take the singular value decomposition of S. Let u be
        the n-th left singular vector;   % calculate the vector u which
                          is orthogonal to all other eigenvectors (9.28)
        Replace the kk-th column of Xt with Pr * u; (9.29)
        Normalize the kk-th column of Xt to have unit length;
        If the kk-th column of Xt has complex elements
            Append X1 with the complex conjugate of the
            kk-th column of Xt;
        End
    End
    End
    Let C = the condition number of [ Xt   X1 ];
    If C is less than CD   % test the condition number
        Let Xtf = Xt;   % keep the best set of eigenvectors
        Let CD = C;
    End
End
```

Table 9.5 An algorithm to compute the optimal closed-loop eigenvectors (continued from the previous page).

Let kk=0; Let Xb and $X2$ be null matrices;

For i = 1 to np;

For j = 1 to $\mathbf{q}(i)$

$kk = kk + 1$;

if $\mathbf{cv}(kk)$ =1 % if $\mathbf{p}(kk)$ is real, choose the real or imaginary part of the eigenvector, whichever has larger norm;

Let $\mathbf{x}$ equal the real part of the kk-th column of Xtf;

Let $\mathbf{y}$ equal the imag. part of the kk-th column of Xtf;

if $\|\mathbf{x}\| > \|\mathbf{y}\|$

Let the kk-th column of Xtf equal $\mathbf{x}/\|\mathbf{x}\|$;

else

Let the kk-th column of Xtf equal $\mathbf{y}/\|\mathbf{y}\|$;

End

End

$\mathbf{a}$ = [rows 1 to n, columns $(i-1)*m+1$ to $i*m$ of $AT]^{-1}$ $*kk$-th column of Xtf; % See (9.32)–(9.34)

Let $\mathbf{t}$ = [rows $n+1$ to $n+m$, columns $(i-1)*m+1$ to $i*m$ of $AT] * \mathbf{a}$;

Let $\mathbf{x}$ equal the imaginary part of $\mathbf{t}$;

Append Xb with the real part of $\mathbf{t}$;

If $\mathbf{cv}(kk)$ = 0

Append $X2$ with $\mathbf{x}$;

Append $X1$ with the imaginary part of the kk-th column of Xtf;

Replace the kk-th column of Xtf with the real part of the kk-th column of Xtf;

End

End

End

K=$[\,Xb \quad X2\,][\,Xtf \quad X1\,]^{-1}$;

Table 9.6 An algorithm to compute the feedback matrix $\mathbf{L}$ from a set of optimal closed-loop eigenvectors (follows the algorithm given in Table 9.4). Text to the right of the symbol % are comments. Numbers in parenthesis refer to equation numbers from the preceding development.

9.11 Problems

1. Consider the following multivariable systems described by differential equations relating their inputs and outputs. Obtain a state-space model for each system.

   (a)

$$\ddot{y}_1 + \dot{y}_1 + y_2 = u_1$$

$$\ddot{y}_2 + \dot{y}_1 + 2y_2 = u_2$$

   (b)

$$\ddot{y}_1 + 2\ddot{y}_2 + 2y_1 + y_2 = 3u_1 + \dot{u}_2$$

$$\dot{y}_2 + y_1 = u_2$$

   (c)

$$\dot{y}_1 + 2y_2 = u_1 + u_2$$

$$\ddot{y}_2 + 2\dot{y}_2 + \dot{y}_1 = \dot{u}_1 - u_2$$

   (d)

$$\dot{y}_1 + 2y_1 + y_2 = u_1$$

$$\dot{y}_2 + 2\dot{y}_1 = u_2$$

2. For each pair of matrices $(\mathbf{\Phi}, \mathbf{\Gamma})$ shown below, use Fact 9.1 to test if the system is controllable. If the system is not controllable, do the following two things: (1) find which eigenvalue of $\mathbf{\Phi}$ causes the matrix in Fact 9.2 to be rank deficient, and (2) pick an arbitrary output matrix $\mathbf{C}$ so that the system $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{C})$ has an equal number of inputs and outputs, and show that the eigenvalue found in (1) is also a transmission zero of the system.

   (a)

$$\mathbf{\Phi} = \begin{bmatrix} 0 & 0 & -2 \\ 1 & -3 & 0 \\ 0 & -3 & 0 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0 & 2 \\ -3 & -1 \\ -2 & 0 \end{bmatrix}$$

   (b)

$$\mathbf{\Phi} = \begin{bmatrix} -1 & -2 & 0 \\ -3 & -4 & 0 \\ 0 & 0 & -5 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}$$

   (c)

$$\mathbf{\Phi} = \begin{bmatrix} 5 & 8 \\ -3 & -5 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

(d)

$$\mathbf{\Phi} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 2 & -2 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0 & 1 \\ 1 & -1 \\ -1 & 0 \end{bmatrix}$$

3. The non-iterative algorithm presented in Section 9.4 can be used to calculate feedback gains for single-input systems. Since a given set of desired closed-loop pole locations results in a unique set of feedback gains for single-input systems, there is no need for an iterative algorithm. In fact the non-iterative algorithm gives the same result as the formula for feedback gains given in Chapter 6. For each system described below, calculate the feedback gain matrix using the non-iterative algorithm.

(a)

$$\mathbf{\Phi} = \begin{bmatrix} 1 & .0952 \\ 0 & .9048 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} .0048 \\ .0952 \end{bmatrix},$$

desired closed-loop poles: $0.8110 \pm j0.0953$

$$\mathbf{\Phi} = \begin{bmatrix} 1.0000 & 0.0952 & 0.0042 \\ 0 & 0.9048 & 0.0782 \\ 0 & 0 & 0.6703 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.0001 \\ 0.0042 \\ 0.0824 \end{bmatrix},$$

desired closed-loop poles: $0.7784,\ 0.8055 \pm j0.1543$

4. Calculate a feedback gain matrix for each system shown below using the non-iterative algorithm. Find the condition number of the closed-loop eigenvectors for each system.

(a)

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 1 \\ 0 & 0.9 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

desired closed-loop poles are: $0.9,\ 0.9$

(b) Same as (a) with desired closed-loop poles: $0.6486 \pm j0.1546$

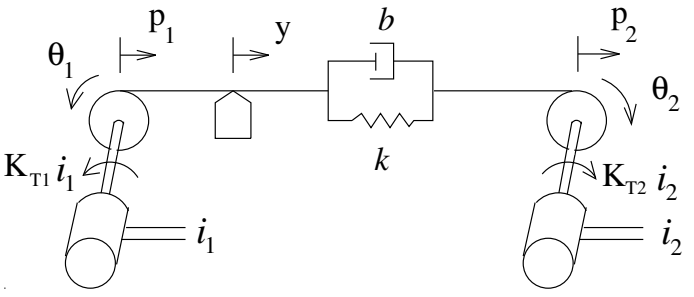
(c)

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0.03 & -0.2 \\ 0.9 & 0.75 & -0.01 \\ -0.01 & -0.25 & 1 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} 0.02 & 0.2 \\ -0.25 & -0.08 \\ -0.16 & -0.01 \end{bmatrix},$$

desired closed-loop poles are: $0.6060,\ 0.6250 \pm j0.2485$

5. *Magnetic Tape Drive (Franklin, Powell, Workman)*

Consider the magnetic tape drive system shown in the following figure



The system has two drive motors which can be controlled independently. The tape is modeled as a linear spring with a small amount of viscous damping. The goal is to control the position of the tape over the read head, and also control the tension of the tape.

The system is described by the following differential equations

$$
\begin{aligned}
J\ddot{\theta}_1 &= -T_e r + K_m i_1, \\
J\ddot{\theta}_2 &= -T_e r + K_m i_2, \\
T_e &= k(p_2 - p_1) + b(\dot{p}_2 - \dot{p}_1), \\
y &= (p_1 + p_2)/2,
\end{aligned}
$$

where

$i_1,\ i_2$ = current into motors 1 and 2, respectively (A)

T_e = tension in tape (N)

$\theta_1,\ \theta_2$ = angular position of motor/capstan assembly (radians)

$p_1,\ p_2$ = position of tape at capstan (mm)

y = position of tape over read head (mm)

J = 0.006375 kg-m^2, motor and capstan inertia

r = 0.1 m, capstan radius

K_m = 0.544 N-m/A, motor torque constant

k = 2113 N/m, tape spring constant

b = 3.75 N-sec/m, tape damping coefficient

If we define the following state variables

$$x_1 = p_1$$

$$x_2 = p_2$$

$$\mathbf{x}_3 = \dot{\theta}_1$$

$$\mathbf{x}_4 = \dot{\theta}_2$$

then the following state-space model is obtained

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -100 & 0 \\ 0 & 0 & 0 & 100 \\ 33.15 & -33.15 & -5.882 & -5.882 \\ 33.15 & -33.15 & -5.882 & -5.882 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 85.33 & 0 \\ 0 & 85.33 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix}.$$

The outputs are defined as follows

$$y_1 = y$$

$$y_2 = T_e.$$

The equation for the outputs is

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ -2.113 & 2.1130.3750.375 & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}.$$

(a) Design a tracking system for a step command in y_1 and y_2. It is desired to control the tape tension to 2N. Thus the reference input is the constant vector

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Assume full state feedback in your design.

(b) Repeat (a) assuming only y_1 and y_2 are measured.

6. The lateral axis model of an L-1011 aircraft at cruise flight conditions is given in []. The model includes actuator dynamics and a washout (high-pass) filter on the yaw rate. The state variables are defined as follows:

$$x_1 = \text{rudder deflection (rad)}$$

$$x_2 = \text{aileron deflection (rad)}$$

$$x_3 = \text{bank angle (rad)}$$

$$x_4 = \text{yaw rate (rad/s)}$$

$$x_5 = \text{roll rate (rad/s)}$$

$$x_6 = \text{sideslip angle (rad)}$$

$$x_7 = \text{washout filter state.}$$

The systems inputs are

$$u_1 = \text{rudder command (rad)}$$

$$u_2 = \text{aileron command (rad).}$$

The system outputs are

$$y_1 = \text{washed out yaw rate (rad/s)}$$

$$y_2 = \text{roll rate (rad/s)}$$

$$y_3 = \text{sideslip angle (rad)}$$

$$y_4 = \text{bank angle (rad).}$$

The matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ of the state-space model are given below

$$\mathbf{A} = \begin{bmatrix} -20 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -0.744 & -0.032 & 0-0.154 & -0.0042 & 1.54 & 0 & \\ 0.337 & -1.12 & 0 & 0.249 & -1 & -5.2 & 0 \\ 0.02 & 0 & 0.0386 & -0.996 & -0.000295 & -0.117 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.5 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 20 & 0 \\ 0 & 25 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

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

The open-loop poles (eigenvalues of the $\mathbf{A}$ matrix) have the following values and descriptions []

$\lambda_1 = -20$	rudder mode
$\lambda_2 = -25$	aileron mode
$\lambda_{3,4} = -0.0884 \pm j1.272$	dutch roll mode
$\lambda_5 = -1.085$	roll subsidence mode
$\lambda_6 = -0.00911$	spiral mode
$\lambda_7 = -0.5$	washout filter mode.

In order to provide lateral stability augmentation as well as to close the roll attitude loop, the following closed-loop poles are chosen [] (the superscript 'd' refers to de-

sired pole locations)

$$\lambda_{3,4}^d = -1.5 \pm j1.5 \quad \text{dutch roll}$$

$$\lambda_{5,6}^d = -2.0 \pm j1.5 \quad \text{roll mode.}$$

The other three closed-loop poles can be chosen as the open-loop poles λ_1, λ_2, and λ_7, or they can be chosen as the roots of a Bessel polynomial with settling time $T_s = -4.5/2 = -2.25$.

Calculate feedback gain matrices corresponding to the two different sets of closed-loop pole locations mentioned above. Choose the one which yields the best condition number for the closed-loop eigenvectors. Plot the response of the closed-loop regulator in response to the following initial conditions: initial sideslip angle of 1° (all other variables zero), and initial bank angle of 1° (all other variables zero).

7. A linearized model of the vertical-plane dynamics of an aircraft is given below. The model has three inputs, three outputs, and five state variables, which are defined below. The inputs are

$$u_1 = \text{spoiler angle (tenths of a degree)}$$

$$u_2 = \text{forward acceleration } (m\ s^{-2})$$

$$u_2 = \text{elevator angle (degrees).}$$

The state variables are

$$x_1 = \text{altitude } (m)$$

$$x_2 = \text{forward speed } (m/s)$$

$$x_3 = \text{pitch angle (degrees)}$$

$$x_4 = \text{pitch rate (degrees/sec)}$$

$$x_5 = \text{vertical speed } (m/s).$$

The three outputs are just the first three state variables. The matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ of the state-space model are

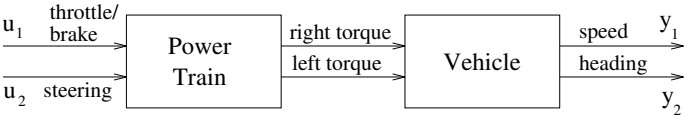
$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1.1320 & 0 & -1.000 \\ 0 & -0.0538 & -0.1712 & 0 & 0.0705 \\ 0 & 0 & 0 & 1.000 & 0 \\ 0 & 0.0485 & 0 & -0.8556 & -1.013 \\ 0 & -0.2909 & 0 & 1.0532 & -0.6859 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ -0.120 & 1.000 & 0 \\ 0 & 0 & 0 \\ 4.4190 & 0 & -1.665 \\ 1.5750 & 0 & -0.0732 \end{bmatrix},$$

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

Design a tracking system which will have zero steady-state error to step commands for each of the three outputs. Let the settling time be 0.4 seconds.

8. This problem considers the design of a control system for a tracked vehicle. In a tracked vehicle, the right and left tracks (treads) are operated at different speeds in order to turn the vehicle in a desired direction. A block diagram of the system is shown below.



A state-space model for a certain tracked vehicle is

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & -7.7144 & 0 & 0 \\ 1 & -11.5143 & 0 & 0 \\ 0 & 0 & -4.106 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 45.4572 & 12.8431 \\ 4.209 & 17.98 \\ 0 & 1.223 \\ 0 & 0 \end{bmatrix} \mathbf{u}$$

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

The inputs are

$$u_1 = \text{throttle/brake}$$

$$u_2 = \text{steering (rad)}$$

and the outputs are

$$y_1 = \text{longitudinal speed (m/sec)}$$

$$y_2 = \text{heading (rad)}.$$

Design a tracking system that has zero steady-state error to step commands for each of the output variables. The desired settling time is 1 second. Assume that only the output signals are measured. Verify your design by obtaining step responses of the closed-loop system.

CHAPTER 10

OPTIMAL CONTROL

Previous chapters have developed tools for designing digital control systems. The tools were developed using a combination of state-space system theory and linear algebra. The major results included pole placement by digital state feedback, and the design of observer systems to estimate unmeasured state variables.

The pole placement problem was solved by computing state feedback gains to achieve desired closed-loop pole locations. For single-input systems, desired pole locations uniquely specify a vector of feedback gains. For multi-input systems, the desired pole locations do not uniquely specify the feedback gains, and other objectives such as robustness or decoupling can be achieved. The calculation of observer gains is a dual problem to the calculation of feedback gains, and observer design also begins with the specification of desired (observer) pole locations.

While there are guidelines for selecting desired closed-loop pole locations and observer pole locations, there is usually some iteration over a number of different sets of pole locations until a final design is achieved. The pole placement formulas provide a "high level language" in which to iterate the design. In other word, instead of varying individual gains directly (this procedure would not even guarantee stability of the closed-loop system), the desired pole locations are altered in a way that achieves the desired effects. In summary, the input to the pole placement design procedure is a set of desired pole locations, and the design formulas convert these into a control system.

In this chapter we present a different set of design tools. The tools are developed using state-space system theory, linear algebra, and also calculus. The idea is to construct a function which measures the performance of a control system. For instance, the functions

considered here measure the error between the actual behavior of the system and the desired behavior. This error will generally be a function of the state variables and the inputs to the plant. Once the performance index has been specified, it is an exercise in calculus to find the minimum of this function with respect to the inputs to the plant. The inputs which minimize the performance index are thus optimal. Of course, if the performance index is changed, then the optimal inputs also change, in general. Optimal control design is also usually an iterative process in which the performance index is varied until the behavior of the control system is acceptable.

In this chapter, we will consider the design of regulators which drive the state variables of the plant to zero. The function which measures the performance of the regulator will be the following quadratic function of the states and inputs

$$J = \frac{1}{2}\sum_{k=0}^{\infty}\left(\mathbf{x}[k]^T\mathbf{Q}\mathbf{x}[k] + \mathbf{u}[k]\mathbf{R}\mathbf{u}[k]\right) \tag{10.1}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are symmetric, positive-definite matrices[1] of dimension $n \times n$ and $p \times p$, respectively. An optimal regulator is obtained by minimizing J. The first term in the summation of (10.1) adds to the value of J for each $\mathbf{x}[k]$ that is not equal to zero. Thus to minimize J, the optimal regulator drives $\mathbf{x}[k]$ to zero as desired.

The second term in the summation of (10.1) penalizes the use of large control signals, since these will cause J to be large. If the matrix $\mathbf{R}$ is large with respect to the matrix $\mathbf{Q}$, the optimal regulator will drive the state vector to zero without using much control effort. The settling time of such a regulator will be large.

If, on the other hand, $\mathbf{R}$ is small with respect to $\mathbf{Q}$, then the optimal regulator will use large control effort to drive the state vector to zero quickly, giving a small settling time. Thus by varying the matrices $\mathbf{Q}$ and $\mathbf{R}$, optimal regulators which trade off speed of response and control effort can be designed.

The careful reader may have realized that (10.1) by itself is not sufficient to design optimal regulators. The reason is that J can be minimized simply by choosing $\mathbf{u}[k]$ and $\mathbf{x}[k]$ to be zero for all k! This is not the result we are looking for. The problem is that the state-vector sequence $\mathbf{x}[k]$ and the input sequence $\mathbf{u}[k]$ are not independent variables which can be arbitrarily chosen to minimize J. Rather, these sequences are related by the state update equation of the ZOH of the plant:

$$\mathbf{x}[k+1] = \mathbf{\Phi}\mathbf{x}[k] + \mathbf{\Gamma}\mathbf{u}[k]\ ,\ \ \mathbf{x}[0] = \mathbf{x}_0. \tag{10.2}$$

If the initial state vector is nonzero, then the state vector sequence will be nonzero even if the input sequence is chosen to be zero. This means that the choosing $\mathbf{x}[k]$ and $\mathbf{u}[k]$ to be zero for all k is not physically meaningful.

In light of the above discussion, the design of optimal regulators requires both (10.1) and (10.2). We need to minimize J with respect to the sequences $\mathbf{x}[k]$ and $\mathbf{u}[k]$ *subject to the constraint* that these sequences are related by (10.2). The solution to this constrained optimization problem will be developed in the next two sections in a series of steps. First, the unconstrained optimization of functions of a single variable will be reviewed. This result will then be generalized to functions of vector-valued variables, and it will be shown how to incorporate constraints into the problem. All of these results will be developed first

[1] A symmetric matrix $\mathbf{Q}$ is said to be **positive definite** if $\mathbf{x}^T\mathbf{Q}\mathbf{x} > 0$ for any vector $\mathbf{x}$. All the eigenvalues of such a matrix are positive.

1. A *necessary* condition for x^* to be a local minimum of $f(x)$ is that the derivative of $f(x)$ evaluated at x^* should equal zero; that is

$$\frac{d}{dx} f(x) \mid_{x=x^*} = 0.$$

This implies that a line tangent to $f(x)$ at x^* has zero slope as shown in Fig. 10.1.

2. A *sufficient* condition for x^* to be local minimum is that the second derivative of $f(x)$ at x^* be greater that zero; that is

$$\frac{d^2}{dx^2} \mid_{x=x^*} > 0.$$

Thus implies that the function is "concave up" at x^*.

Table 10.1 Necessary and sufficient conditions for x^* to minimize the function $f(x)$.

for static problems in which the variables do not depend on time. Finally, the solution to the dynamic optimization problem represented by (10.1) and (10.2) will be derived.

10.1 Static Optimization

10.1.1 Functions of a Single Variable

Let us start with the simplest optimization problem from calculus. The solution to this problem will be a model for the problem of calculating optimal controls. Given a scalar-valued function $f(x)$ where x is a scalar variable, find the value of x, call it x^* which minimizes $f(x)$. Suppose that $f(x)$ is a smooth function which is at least twice differentiable, such as the function shown in Fig. 10.1. The conditions for a point x^* to minimize

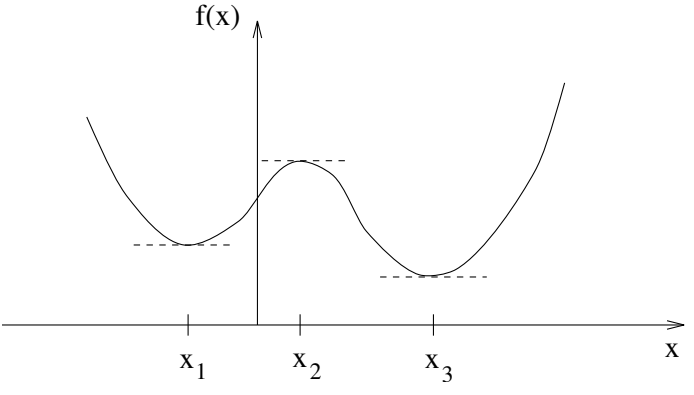


Figure 10.1 A smooth function $f(x)$ with local minima at x_1 and x_3, and a local maximum at x_2. The dashed lines indicate that the derivative of $f(x)$ has zero slope at the local extrema of $f(x)$.

$f(x)$ are shown in Table 10.1.1. These conditions can be derived by considering a Taylor

series expansion of the function $f(x)$ about some point x_0

$$f(x_0 + \Delta x) = f(x_0) + \frac{d}{dx} f(x) \mid_{x=x_0} \Delta x + \frac{d^2}{dx^2} \mid_{x=x_0} (\Delta)^2 + R \tag{10.3}$$

where R is a remainder term which is of the order of $(\Delta x)^3$. For Δx small enough, $f(x + \Delta x)$ can be approximated by the first 3 terms on the right-hand side of (10.3). For a fixed value of x_0, these terms are a quadratic function of Δx, i.e. $f(x_0 + \Delta x)$ is a parabola in Δx. If the coefficient of $(\Delta x)^2$ is positive, this parabola opens upward (condition 2 above). If, in addition, the coefficient of Δx is zero, the minimum occurs at $\Delta x = 0$ (condition 1 above). Thus conditions 1 and 2 above are necessary and sufficient for $f(x)$ to have a local minimum at $x = x_0$. We are only guaranteed of finding a local minimum because only points near x_0 were considered since we assumed Δx was small. Recall that Δx had to be small enough that the remainder term in (10.3) could be neglected. Note that if $f(x)$ is a *quadratic function of* x, then the remainder term in (10.3) is always zero. So in the simple case of quadratic functions, conditions 1 and 2 above are necessary and sufficient for x^* to be the *global minimum of* $f(x)$.

EXAMPLE 10.1

Consider the function

$$f(x) = \frac{1}{2} a x^2 + bx + c.$$

The first derivative of this function is

$$f'(x) = ax + b,$$

and the second derivative is

$$f''(x) = a.$$

The first derivative is equal to zero at

$$x^* = \frac{b}{a}$$

and if $a > 0$, the condition in Table 10.1.1 are satisfied and so x^* minimizes $f(x)$. Since $f(x)$ is just the parabola shown in Fig. 10.2, it is clear that x^* is the unique global minimum of $f(x)$.

■

10.1.2 Scalar Functions of Vector-Valued Variables

In order to develop formulas for optimal control, it is necessary to generalize the optimization theory discussed up to this point in several ways. The first generalization is to consider functions of several variables. In other words, how should conditions 1 and 2 be stated when $\mathbf{x}$ is a vector-valued variable?

We begin to answer this question by rewriting (10.3) in a sightly different form. First, we subtract $f(x_0)$ from both sides of the equation, and define $\Delta f \stackrel{\text{def}}{=} f(x_0 + \Delta x) - f(x_0)$ to be the increment in the value of the function $f(x)$ at x_0. Next, since we will only

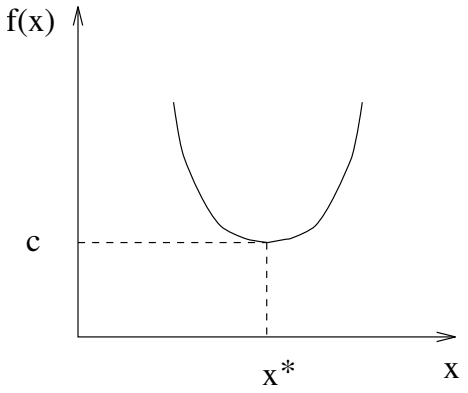


Figure 10.2 A parabola $f(x) = \frac{1}{2}ax^2 + bx + c$, with $a > 0$. The unique global minimum occurs at $x^* = -b/a$.

consider small (differential) changes in x and $f(x)$, we will use "d" notation instead of "Δ" notation. Then we can rewrite (10.3) as follows

$$df = \frac{d}{dx}f(x) \mid_{x=x_0} dx + \frac{1}{2}\frac{d^2}{dx^2} \mid_{x=x_0} (dx)^2 + R \tag{10.4}$$

If $\mathbf{x} = (x_1 \cdots x_n)^T$ is a vector-valued variable, then (10.4) is written using a Taylor series expansion about some nominal vector $\mathbf{x}_0$. The Taylor series expansion amounts to summing up the contributions of the first- and second-order perturbations in the variables $x_1, \cdots, x_n$. The result is

$$df = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \mid_{\mathbf{x}=\mathbf{x}_0} dx_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 f}{\partial x_i \partial \mathbf{x}_j} dx_i dx_j + R. \tag{10.5}$$

It is convenient to express the above equation using matrix-vector notation. In particular, if we define the following vectors

$$d\mathbf{x} \stackrel{\text{def}}{=} \begin{bmatrix} dx_1 \\ \vdots \\ dx_n \end{bmatrix}, \text{ and } f_{\mathbf{x}} \stackrel{\text{def}}{=} \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

and the following matrix

$$f_{\mathbf{xx}} \stackrel{\text{def}}{=} \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial \mathbf{x}_1} & \frac{\partial^2 f}{\partial x_1 \partial \mathbf{x}_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial \mathbf{x}_n} \\ \frac{\partial^2 f}{\partial x_2 \partial \mathbf{x}_1} & \frac{\partial^2 f}{\partial x_2 \partial \mathbf{x}_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial \mathbf{x}_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial \mathbf{x}_1} & \frac{\partial^2 f}{\partial x_n \partial \mathbf{x}_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial \mathbf{x}_n} \end{bmatrix}$$

then (10.5) can be written as

$$df = f_{\mathbf{x}}^T d\mathbf{x} + \frac{1}{2} d\mathbf{x}^T f_{\mathbf{xx}} d\mathbf{x} + R. \tag{10.6}$$

The vector $f_{\mathbf{x}}$ is called the *gradient vector* of f with respect to $\mathbf{x}$, and the matrix $f_{\mathbf{xx}}$ is called the Hessian matrix. Some treatments of multivariate Taylor series define the gradient

If L is a scalar function of a vector-valued variable $\mathbf{u}$ then

1. A *necessary* condition for $\mathbf{u}^*$ to minimize L is that the increment dL equal zero to first order for all increments $d\mathbf{u}$. Since to first order, $dL = L_{\mathbf{u}}^T d\mathbf{u}$, this condition is satisfied at $\mathbf{u} = \mathbf{u}^*$ if and only if

$$L_{\mathbf{u}}|_{\mathbf{u}=\mathbf{u}^*} = 0.$$

2. If the first condition is satisfied, then a sufficient condition for $\mathbf{u}^*$ to minimize L is that the Hessian matrix $L_{\mathbf{uu}}$ be positive definite.

Table 10.2 Necessary and sufficient conditions for $\mathbf{u}^*$ to minimize L.

as a row vector instead of as a column vector. The notation used here is consistent with that used in [12] and [56].

Up to this point, we have discussed conditions which characterize the minimum of a function $f(x)$. We now change to notation which is more descriptive of the optimal control problem. The independent variables in a control problem are the inputs to the plant $\mathbf{u}[k]$. In this section on static optimization, we don't consider the time variation of the inputs; thus the independent variable will just be a vector $\mathbf{u}$. Suppose that a scalar performance index is defined which only depends on $\mathbf{u}$, call it L. Then (10.6) can be rewritten in this new notation as

$$dL = L_{\mathbf{u}}^T d\mathbf{u} + \frac{1}{2} d\mathbf{u}^T L_{\mathbf{uu}} d\mathbf{u} + R. \tag{10.7}$$

We now consider conditions under which $L(\mathbf{u})$ has a minimum at $\mathbf{u} = \mathbf{u}^*$. A necessary condition is that the gradient vector should be zero at $\mathbf{u} = \mathbf{u}^*$, and, in addition, a sufficient condition is that the Hessian matrix should be positive definite at $\mathbf{u} = \mathbf{u}^*$. These conditions are the generalizations of the conditions given in Table 10.1.1 to functions of a vector-valued variable, and are shown in Table 10.2.

EXAMPLE 10.2

Consider a quadratic function of the following form

$$L(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T \mathbf{R}\mathbf{u} + \mathbf{s}^T \mathbf{u} + c.$$

The gradient of this function is given by

$$L_{\mathbf{u}} = \mathbf{R}\mathbf{u} + \mathbf{s}$$

and the Hessian matrix is

$$L_{\mathbf{uu}} = \mathbf{R}.$$

The gradient is equal to zero at

$$\mathbf{u}^* = -\mathbf{R}^{-1}\mathbf{s}, \tag{10.8}$$

and if $\mathbf{R}$ is a positive-definite matrix, this is the unique global minimum of $L(\mathbf{u})$. This result is the generalization of Example 10.1 to functions of vector-valued variables.

■

10.1.3 Optimization with Constraints

In this subsection, we restrict attention to quadratic functions and show how to incorporate constraints. We are still considering *static* problems in which the variables do not depend on time. The specific problem we will consider is to find $\mathbf{x}$ and $\mathbf{u}$ which minimize

$$L = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \frac{1}{2}\mathbf{u}^T\mathbf{R}\mathbf{u} \tag{10.9}$$

subject to the constraint that

$$\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \mathbf{c}. \tag{10.10}$$

In the previous subsection, all the variables were contained in a vector called $\mathbf{u}$. In this subsection we divide the variables up into two vectors $\mathbf{x}$ and $\mathbf{u}$ which are related by the constraint equation (10.10). By doing this, equations (10.9) and (10.10) are static versions of the problem we really want to solve given by equations (10.1) and (10.2).

10.1.3.1 Direct Substitution of Constraints The static problem can easily be solved by substituting the constraint equation (10.10) into L. To perform this substitution, we assume that the matrix $\mathbf{A}$ is nonsingular (a similar assumption will be needed for the optimal control problem) and solve for $\mathbf{x}$ from (10.10) to get

$$\mathbf{x} = \mathbf{A}^{-1}[\mathbf{c} - \mathbf{B}\mathbf{u}].$$

The above expression for $\mathbf{x}$ is then substituted into (10.9) to obtain a function of $\mathbf{u}$ only

$$L = \frac{1}{2}[\mathbf{c}^T - \mathbf{u}^T\mathbf{B}^T]\mathbf{A}^{-T}\mathbf{Q}\mathbf{A}^{-1}[\mathbf{c} - \mathbf{B}\mathbf{u}] + \frac{1}{2}\mathbf{u}\mathbf{R}\mathbf{u}. \tag{10.11}$$

We can now perform an *unconstrained* minimization of L in the above equation because the constraint has been satisfied by the substitution. In order to minimize L in (10.11), we first define

$$\bar{\mathbf{Q}} \stackrel{\text{def}}{=} \mathbf{A}^{-T}\mathbf{Q}\mathbf{A}^{-1}. \tag{10.12}$$

Next we expand (10.11) to obtain

$$\begin{aligned} L &= \tfrac{1}{2}\mathbf{c}^T\bar{\mathbf{Q}}\mathbf{c} - \mathbf{u}^T\mathbf{B}^T\bar{\mathbf{Q}}\mathbf{c} + \tfrac{1}{2}\mathbf{u}^T\mathbf{B}^T\bar{\mathbf{Q}}\mathbf{u} + \tfrac{1}{2}\mathbf{u}^T\mathbf{R}\mathbf{u} \\ &= \tfrac{1}{2}\mathbf{u}^T\bar{\mathbf{R}}\mathbf{u} - \mathbf{c}^T\bar{\mathbf{Q}}B\mathbf{u} + \tfrac{1}{2}\mathbf{c}^T\bar{\mathbf{Q}}\mathbf{c} \end{aligned} \tag{10.13}$$

where we have used the fact that $\bar{\mathbf{Q}}$ is a symmetric matrix, and we have introduced the matrix

$$\bar{\mathbf{R}} \stackrel{\text{def}}{=} \mathbf{R} + \mathbf{B}^T\bar{\mathbf{Q}}\mathbf{B} = \mathbf{R} + \mathbf{B}^T\mathbf{A}^{-T}\mathbf{Q}\mathbf{A}^{-1}\mathbf{B}.$$

Equation (10.13) is now in the form shown in Example 10.2 with $\mathbf{R}$ replaced by $\bar{\mathbf{R}}$ and $\mathbf{s}$ replaced by $\mathbf{B}^T\bar{\mathbf{Q}}\mathbf{c}$. With these replacements, the solution from Example 10.2 is

$$\begin{aligned} \mathbf{u}^* &= \bar{\mathbf{R}}^{-1}\mathbf{B}^T\bar{\mathbf{Q}}\mathbf{c} \\ &= (\mathbf{R} + \mathbf{B}^T\mathbf{A}^{-T}\mathbf{Q}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}^{-1}\mathbf{Q}\mathbf{A}^{-1}\mathbf{c}. \end{aligned} \tag{10.14}$$

We have just seen that when the constraint equation can be solved for $\mathbf{x}$ in terms of $\mathbf{u}$, it is possible to convert a constrained optimization problem to an unconstrained problem. In addition, since the constrained problem just considered (equations (10.9) and (10.10))

is the static version of the problem we really want to solve (equations (10.1) and (10.2)), we might hope that the optimal control problem could be solved simply by substituting the constraints into the performance index. Unfortunately, the solution to the dynamic optimization problem is not as simple as in the static case. The reason is as follows. In the static problem, the vector $\mathbf{c}$ in the constraint equation (10.10) was a fixed vector – it was *not* varied to minimize the performance index. However, the vector corresponding to $\mathbf{c}$ in the dynamic constraint equation (10.2) is $\mathbf{x}[k+1]$, and this vector is not fixed, but must be chosen in such a way as to minimize the performance index.

Since direct substitution of the constraint equation is not appropriate for the dynamic optimization problem, we will develop a more sophisticated solution procedure for constrained optimization problems. The new procedure will first be developed for the static case, and it will yield the same optimal solution (10.14) that was obtained by substitution of the constraint equation. The advantage of the new approach is that it can be generalized to the case of dynamic optimization in a straightforward way.

10.1.3.2 Lagrange Multipliers

We begin by repeating the static constrained optimization problem. We want to find $\mathbf{x}$ and $\mathbf{u}$ which minimize

$$L = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \frac{1}{2}\mathbf{u}^T\mathbf{R}\mathbf{u} \tag{10.15}$$

subject to the constraint that

$$\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \mathbf{c}. \tag{10.16}$$

If we temporarily ignore the constraint, the necessary condition to minimize L is given in Table 10.2. A small notational difference is that in Table 10.2, all of the variables are collected in a single vector $\mathbf{u}$, and the variables in the problem we are now considering are in two vectors, $\mathbf{x}$ and $\mathbf{u}$. Nevertheless, the necessary condition from Table 10.2 is that $dL = 0$ to first order. From (10.15) we see that $L_{\mathbf{x}} = \mathbf{Q}\mathbf{x}$, $L_{\mathbf{u}} = \mathbf{R}\mathbf{u}$, and so

$$dL = \mathbf{x}^T\mathbf{Q}d\mathbf{x} + \mathbf{u}^T\mathbf{R}d\mathbf{u} = 0. \tag{10.17}$$

This condition says that arbitrary increments in $\mathbf{x}$ and $\mathbf{u}$ do not change the value of L, and so L cannot be made any smaller. Thus this condition is indeed necessary for a minimum. However, in a constrained optimization problem, the increments $d\mathbf{x}$ and $d\mathbf{u}$ are not arbitrary, but are related by the constraint equation. We can derive this relationship as follows. Suppose $\mathbf{x}_0$ and $\mathbf{u}_0$ satisfy (10.16) and we consider nearby vectors $\tilde{\mathbf{x}} = \mathbf{x}_0 + d\mathbf{x}$ and $\tilde{\mathbf{u}} = \mathbf{u}_0 + d\mathbf{u}$. In order for $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{u}}$ to satisfy the constraint equation (10.16) it must be true that

$$\begin{aligned}
\mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\tilde{\mathbf{u}} &= \mathbf{c} \\
\mathbf{A}(\mathbf{x}_0 + d\mathbf{x}) + \mathbf{B}(\mathbf{u}_0 + d\mathbf{u}) &= \mathbf{c} \\
\mathbf{A}\mathbf{x}_0 + B\mathbf{u}_0 + \mathbf{A}d\mathbf{x} + \mathbf{B}d\mathbf{u} &= \mathbf{c} \\
\mathbf{c} + \mathbf{A}d\mathbf{x} + \mathbf{B}d\mathbf{u} &= \mathbf{c} \\
\mathbf{A}d\mathbf{x} + \mathbf{B}d\mathbf{u} &= \mathbf{0}.
\end{aligned} \tag{10.18}$$

Equation (10.17) gives necessary conditions in terms of $d\mathbf{x}$ and $d\mathbf{u}$ to be at a stationary point (minimum) of L. The last equation in (10.18) gives a condition in terms of $d\mathbf{x}$ and $d\mathbf{u}$ which insures that the constraints will be satisfied. To solve the constrained optimization problem, then, (10.17) and the last equation in (10.18) must both be satisfied simultaneously. These two equations can be written together in matrix form as follows

$$\begin{bmatrix} \mathbf{x}^T\mathbf{Q} & \mathbf{u}^T\mathbf{R} \\ \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} d\mathbf{x} \\ d\mathbf{u} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{0} \end{bmatrix}. \tag{10.19}$$

This equation says that $[d\mathbf{x}^T \;\; d\mathbf{u}^T]^T$ is a vector in the null space of the coefficient matrix. If we assume that $\mathbf{u}$ can take arbitrary values (it can in a control problem), then $d\mathbf{u}$ is arbitrary. Of course, for a given $d\mathbf{u}$, $d\mathbf{x}$ must satisfy (10.18). Now consider a set of increments in $\mathbf{u}$ obtained by perturbing only one element of $\mathbf{u}$ at a time by a small number ϵ

$$d\mathbf{u}_1 = \begin{bmatrix} \epsilon \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad d\mathbf{u}_2 = \begin{bmatrix} 0 \\ \epsilon \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \cdots, d\mathbf{u}_p = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ \epsilon \end{bmatrix}. \tag{10.20}$$

It is clear that an arbitrary increment $d\mathbf{u}$ can be written as a linear combination of $d\mathbf{u}_1, \cdots, d\mathbf{u}_p$. The linear combination can be written for a given $d\mathbf{u}$ as follows

$$d\mathbf{u} = \begin{bmatrix} du_1 \\ \vdots \\ du_p \end{bmatrix} = \sum_{i=1}^{p} \alpha_i d\mathbf{u}_i$$

where

$$\alpha_i = du_i/\epsilon\,, \;\; i = 1, \cdots, p.$$

It is also clear that the vectors $d\mathbf{u}_1, \cdots, d\mathbf{u}_p$ are linearly independent, since they are just the standard basis for $\mathbf{R}^p$ scaled by ϵ. For each $d\mathbf{u}_i$ in (10.20), there is a corresponding $d\mathbf{x}_i$ which is found using (10.18) as

$$d\mathbf{x}_i = -A^{-1}B d\mathbf{u}_i\,, \;\; i = 1, \cdots, p. \tag{10.21}$$

This means that any increment $[d\mathbf{x}^T \;\; d\mathbf{u}^T]^T$ can be written as a linear combination of the following p linearly independent vectors

$$\begin{bmatrix} d\mathbf{x}_1 \\ d\mathbf{u}_1 \end{bmatrix}, \quad \cdots, \quad \begin{bmatrix} d\mathbf{x}_p \\ d\mathbf{u}_p \end{bmatrix} \tag{10.22}$$

where $d\mathbf{u}_i$ and $d\mathbf{x}_i$ are defined in (10.20) and (10.21), respectively. In other words, the p vectors in (10.22) are a basis for the space of possible increments $[d\mathbf{x}^T \;\; d\mathbf{u}^T]^T$, and so this space has dimension p.

Returning to (10.19), we see that the coefficient matrix has a null-space of dimension p. Recall that the rank of a matrix plus the dimension of its null space must equal the number of columns of the matrix. The dimensions of the coefficient matrix in (10.19) are $(n+1) \times (n+p)$, and so its rank must equal $(n+p) - p = n$. Since the matrix has $n+1$ rows and only n of them are linearly independent (rank equals the number of linearly independent rows), then one of the rows can be written as a linear combination of the others. Since the $n \times n$ matrix A is assumed to be nonsingular, its rows are linearly independent. Thus the rows of the matrix $[\mathbf{A} \;\; \mathbf{B}]$ must be linearly independent. This means that the first row of the matrix in (10.19) is linearly dependent on the remaining n rows. One way to express this linear dependence is to say that the first row plus some linear combination of the remaining rows must equal zero, as in the following equation

$$[\,1 \quad \boldsymbol{\lambda}^T\,] \begin{bmatrix} \mathbf{x}^T\mathbf{Q} & \mathbf{u}^T\mathbf{R} \\ \mathbf{A} & \mathbf{B} \end{bmatrix} = \mathbf{0}^T \tag{10.23}$$

where

$$\boldsymbol{\lambda}^T = [\,\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_p\,].$$

If we multiply out (10.21) and transpose the results we obtain the following equations

$$\begin{aligned} \mathbf{Q}\mathbf{x} + \mathbf{A}^T\boldsymbol{\lambda} &= \mathbf{0} \\ \mathbf{R}\mathbf{u} + \mathbf{B}^T\boldsymbol{\lambda} &= \mathbf{0}. \end{aligned} \tag{10.24}$$

The above equations are a necessary condition for a constrained problem to have a minimum. It can be shown that for the constrained *quadratic* optimization problem (10.15), (10.16), a sufficient condition for a minimum is that $\mathbf{Q}$ and $\mathbf{R}$ be positive definite.

Using the first equation in (10.24) we obtain

$$\boldsymbol{\lambda}^* = -\mathbf{A}^{-T}\mathbf{Q}\mathbf{x}^*$$

and using the second equation in (10.24) we obtain

$$\mathbf{u}^* = -\mathbf{R}^{-1}\mathbf{B}^T\boldsymbol{\lambda}^* = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{A}^{-T}\mathbf{Q}\mathbf{x}^*. \tag{10.25}$$

Finally, we know from the constraint equation (10.16) that

$$\mathbf{x}^* = \mathbf{A}^{-1}(\mathbf{c} - \mathbf{B}\mathbf{u}^*) \tag{10.26}$$

and combining (10.25) and (10.26) yields

$$\mathbf{u}^* = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{A}^{-T}\mathbf{Q}\mathbf{A}^{-1}(\mathbf{c} - \mathbf{B}\mathbf{u}^*).$$

We can collect the terms which involve $\mathbf{u}^*$ and simplify to get

$$\begin{aligned} (\mathbf{I} + \mathbf{R}^{-1}\mathbf{B}^T\mathbf{A}^{-T}\mathbf{Q}\mathbf{A}^{-1}\mathbf{B})\mathbf{u}^* &= \mathbf{R}^{-1}\mathbf{B}^T\mathbf{A}^{-T}\mathbf{Q}\mathbf{A}^{-1}\mathbf{c} \\ \mathbf{u}^* &= (\mathbf{I} + \mathbf{R}^{-1}\mathbf{B}^T\mathbf{A}^{-T}\mathbf{Q}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{A}^{-T}\mathbf{Q}\mathbf{A}^{-1}\mathbf{c} \\ \mathbf{u}^* &= (\mathbf{R} + \mathbf{B}^T\mathbf{A}^{-T}\mathbf{Q}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}^{-1}\mathbf{Q}\mathbf{A}^{-1}\mathbf{c} \end{aligned} \tag{10.27}$$

where the last equation is derived using the matrix identity $\mathbf{M}_1^{-1}\mathbf{M}_2^{-1} = (\mathbf{M}_2\mathbf{M}_1)^{-1}$. Notice that the optimal solution in the last line of (10.27) is identical to the solution (10.14) derived earlier by direct substitution of the constraints!

In addition to the vectors $\mathbf{x}$ and $\mathbf{u}$ which are used in the statement of the problem, the solution procedure just developed also used a vector $\boldsymbol{\lambda}$ of auxiliary variables. The variables λ_i are called *Lagrange multipliers*, and they are useful for solving a variety of constrained optimization problems. In the next section, we show how Lagrange multipliers can be used to solve the optimal control problem (10.1) and (10.2), which is a dynamic, constrained optimization problem.

10.2 The Optimal Linear Quadratic Regulator

We begin by stating the constrained optimization problem over N time steps, and later let N go to infinity. Thus we consider the problem of minimizing

$$J_N = \frac{1}{2}\sum_{k=0}^{N-1}\left(\mathbf{x}[k]^T\mathbf{Q}\mathbf{x}[k] + \mathbf{u}[k]^T\mathbf{R}\mathbf{u}[k]\right) \tag{10.28}$$

subject to the constraint that

$$\mathbf{x}[k+1] = \mathbf{\Phi}x[k] + \mathbf{\Gamma}\mathbf{u}[k]\ ,\ \ k = 0, \cdots, N-1\ ,\ \ \mathbf{x}[0] = \mathbf{x}_0. \tag{10.29}$$

As in the previous section, there are two necessary conditions for an optimal solution. The first is that $dJ_N = 0$ to first order so that (compare with (10.17))

$$dJ_N = \sum_{k=0}^{N-1} \left(x[k]^T \mathbf{Q} d\mathbf{x}_k + \mathbf{u}[k]^T \mathbf{R} d\mathbf{u}[k] \right) = 0. \tag{10.30}$$

Note that since $\mathbf{x}[0]$ is a given, constant vector (see (10.29)), we must have $d\mathbf{x}[0] = 0$ in the above equation. The second necessary condition is that the constraints are satisfied so that (compare with the last equation in (10.18))

$$d\mathbf{x}[k+1] = \mathbf{\Phi} d\mathbf{x}[k] + \mathbf{\Gamma} d\mathbf{u}[k]\ ,\ \ k = 0, \cdots, N-1\ ,\ \ d\mathbf{x}[0] = 0. \tag{10.31}$$

Equations (10.30)and (10.31) are conditions involving $d\mathbf{u}[k]$, $d\mathbf{x}[k]$, $k = 0, \cdots, N-1$ which must be simultaneously satisfied. We can write these equations in a single matrix equation as follows

$$\begin{bmatrix} \mathbf{u}[0]^T\mathbf{R} & \mathbf{x}[1]^T\mathbf{Q} & \mathbf{u}[1]^T R & \mathbf{x}[2]^T\mathbf{Q} & \cdots \mathbf{x}[N-1]^T\mathbf{Q} & \mathbf{u}[N-1]^T\mathbf{R} & \mathbf{0} \\ \mathbf{\Gamma} & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi} & \mathbf{\Gamma} & -\mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Phi} & \mathbf{\Gamma} & -\mathbf{I} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{\Phi} & \mathbf{\Gamma} & -\mathbf{I} \end{bmatrix} \times \tag{10.32}$$

$$\begin{bmatrix} d\mathbf{u}[0] \\ d\mathbf{x}[1] \\ d\mathbf{u}[1] \\ d\mathbf{x}[2] \\ \vdots \\ d\mathbf{x}[N-1] \\ d\mathbf{u}[N-1] \\ d\mathbf{x}[N] \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

The coefficient matrix in the above equation has $nN + 1$ rows and $pN + nN$ columns. There are N independent control increments $d\mathbf{u}[i]\ ,\ \ i = 0, \cdots i, N$, each with p elements (the plant has p inputs). Thus there are a total of pN degrees of freedom in the null-space vector, and so the coefficient matrix has a null-space whose dimension is pN. The rank of the matrix is the number of columns minus the dimension of its null space, or $(pN+nN)-pN = nN$. This means that only nN rows are linearly independent, although the matrix has $nN + 1$ rows, and so one row can be written as a linear combination of the others. Because of the identity matrices, all rows below the first are linearly independent, and so the first row can be written as a linear combination of the remaining rows. We can express this linear combination as follows

$$[\,1 \quad \boldsymbol{\lambda}[1]^T \quad \boldsymbol{\lambda}[2]^T \quad \boldsymbol{\lambda}[3]^T \quad \boldsymbol{\lambda}[4]^T \quad \cdots \boldsymbol{\lambda}[N-1]^T \quad \boldsymbol{\lambda}[N]^T\,] \times \tag{10.33}$$

$$\begin{bmatrix} \mathbf{u}[0]^T\mathbf{R} & \mathbf{x}[1]^T\mathbf{Q} & \mathbf{u}[1]^T\mathbf{R} & \mathbf{x}[2]^T\mathbf{Q} & \cdots & \mathbf{u}[N-1]^T\mathbf{R} & \mathbf{0} \\ \boldsymbol{\Gamma} & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi} & \boldsymbol{\Gamma} & -\mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Phi} & \boldsymbol{\Gamma} & -\mathbf{I} & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots &]dots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\Phi} & \boldsymbol{\Gamma} & -\mathbf{I} \end{bmatrix} = \mathbf{0}^T$$

where we have partitioned the Lagrange multiplier vector conformably with the coefficient matrix, so that each $\boldsymbol{\lambda}[i]$ contains p Lagrange multipliers. Also note that the product of the Lagrange multiplier vector and the last block column of the coefficient matrix yields

$$\boldsymbol{\lambda}[N]^T = \mathbf{0}^T. \tag{10.34}$$

Equation (10.33) can be written as two separate equations. The product of the Lagrange multiplier vector and the odd-numbered block columns of the coefficient matrix yields, after transposing,

$$\mathbf{R}\mathbf{u}[k] + \boldsymbol{\Gamma}^T\boldsymbol{\lambda}[k+1] = \mathbf{0}\ ,\ \ k = 0, \cdots, N-1\ ,\ \ \boldsymbol{\lambda}[N] = \mathbf{0}. \tag{10.35}$$

In a similar way, (10.33) evaluated at the even-numbered columns yields, after transposing,

$$\boldsymbol{\lambda}[k] = \mathbf{Q}\mathbf{x}(k) + \boldsymbol{\Phi}^T\boldsymbol{\lambda}[k+1]\ ,\ \ k = 0, \cdots, N-1\ ,\ \ \boldsymbol{\lambda}[N] = \mathbf{0}. \tag{10.36}$$

In addition to the above two equations we also have the state-update equation for the plant

$$\mathbf{x}[k+1] = \boldsymbol{\Phi}\mathbf{x}[k] + \boldsymbol{\Gamma}\mathbf{u}[k]\ ,\ \ k = 0, \cdots, N-1\ ,\ \ \mathbf{x}[0] = \mathbf{x}_0. \tag{10.37}$$

The three equations (10.35)–(10.37) are used to calculate the optimal control. Although they were derived using necessary conditions to minimize J_N, it can be shown that the sufficient conditions are also satisfied because of the quadratic nature of J_N and the assumptions on $\mathbf{Q}$ and $\mathbf{R}$.

Before solving the optimal control problem, we first note an interesting feature of (10.35)–(10.37). Equation (10.35) can be solved for the optimal input to yield

$$\mathbf{u}^* = -\mathbf{R}^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\lambda}[k+1] k = 0, \cdots, N-1\ ,\ \ \boldsymbol{\lambda}[N] = \mathbf{0}. \tag{10.38}$$

Equation (10.36) looks like a state-update equation for the Lagrange multiplier vectors $\boldsymbol{\lambda}[k]$! The vectors $\boldsymbol{\lambda}[k]$ are called *costate* vectors. Notice that (10.36) updates the costate vectors backwards in time from $\boldsymbol{\lambda}[k+1]$ to $\boldsymbol{\lambda}[k]$ starting from the final condition $\boldsymbol{\lambda}[N] = \mathbf{0}$. The costate vector $\boldsymbol{\lambda}[k+1]$ is used to create the optimal input according to (10.38). Equations (10.36)–(10.38) can be combined into the block diagram shown in Fig. 10.3.

This system generates the optimal control, but it is a noncausal system. Nevertheless, Fig. 10.3 shows that to generate the optimal control input, we need to generate the sequence of costate vectors. We now show that the costate vectors can be obtained directly from $\mathbf{x}[k]$ without using the auxiliary system.

We begin by writing (10.36) as a forward recursion

$$\boldsymbol{\lambda}[k+1] = \mathbf{A}^{-T}\boldsymbol{\lambda}[k] - \mathbf{A}^{-T}\mathbf{Q}\mathbf{x}[k]. \tag{10.39}$$

Next we substitute (10.38) and (10.39) into (10.37) to obtain

$$\begin{aligned} \mathbf{x}[k+1] &= \boldsymbol{\Phi}\mathbf{x}[k] - \boldsymbol{\Gamma}\mathbf{R}^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\lambda}[k+1] \\ &= \boldsymbol{\Phi}\mathbf{x}[k] - \boldsymbol{\Gamma}\mathbf{R}^{-1}\boldsymbol{\Gamma}^T[\boldsymbol{\Phi}^{-T}\boldsymbol{\lambda}[k] - \boldsymbol{\Phi}^{-T}\mathbf{Q}\mathbf{x}[k]] \\ &= [\boldsymbol{\Phi} + \boldsymbol{\Gamma}\mathbf{R}^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\Phi}^{-T}\mathbf{Q}]\mathbf{x}[k] - \boldsymbol{\Gamma}\mathbf{R}^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\lambda}[k]. \end{aligned} \tag{10.40}$$

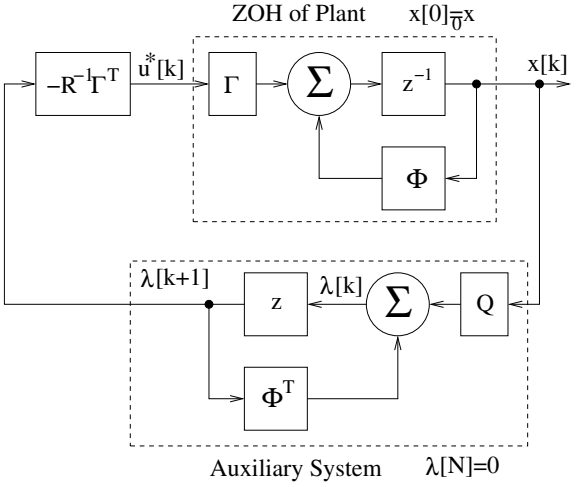


Figure 10.3 A noncausal optimal control system. The auxiliary system operates backwards in time to create the costate vectors which produce the optimal input.

Equations (10.39) and (10.40) can be combined into a single equation that describes the closed-loop system shown in Fig. 10.3

$$\begin{bmatrix} \mathbf{x}[k+1] \\ \boldsymbol{\lambda}[k+1] \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi} + \boldsymbol{\Gamma}\mathbf{R}^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\Phi}^{-T}\mathbf{Q} & -\boldsymbol{\Gamma}\mathbf{R}^{-1}\boldsymbol{\Gamma}^T\boldsymbol{\Phi}^{-T} \\ -\boldsymbol{\Phi}^{-T}\mathbf{Q} & \boldsymbol{\Phi}^{-T} \end{bmatrix} \begin{bmatrix} \mathbf{x}[k] \\ \boldsymbol{\lambda}[k] \end{bmatrix} \tag{10.41}$$

$$\stackrel{\text{def}}{=} \mathbf{H} \begin{bmatrix} \mathbf{x}[k] \\ \boldsymbol{\lambda}[k] \end{bmatrix} \quad k = 0, \cdots, N-1, \ \mathbf{x}[0] = \mathbf{x}_0 \ , \ \boldsymbol{\lambda}[N] = \mathbf{0}.$$

This equation is complicated by the fact that boundary conditions are given at $k = 0$ and $k = N$ – it is known as a two-point boundary value problem. We shall see that a simplification occurs when we consider the steady-state case as N goes to infinity.

In order to proceed we make use of a special property of the coefficient matrix $\mathbf{H}$ in (10.41), namely that it is a *symplectic matrix*. A symplectic matrix, by definition, satisfies

$$\mathbf{H}^T\mathbf{J}\mathbf{H} = \mathbf{J} \tag{10.42}$$

where

$$\mathbf{J} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix}.$$

It can be verified that the coefficient matrix $\mathbf{H}$ in (10.41) satisfies (10.42). Symplectic matrices have a useful eigenvalue symmetry property which we now derive. Using (10.42), we see that the inverse of a symplectic matrix is easy to find as follows

$$\mathbf{J} = \mathbf{H}^T\mathbf{J}\mathbf{H}$$
$$\mathbf{J}\mathbf{H}^{-1} = \mathbf{H}^T\mathbf{J}$$

or

$$\mathbf{H}^{-1} = \mathbf{J}^{-1}\mathbf{H}^T\mathbf{J}. \tag{10.43}$$

Equation (10.43) says that $\mathbf{H}^{-1}$ and $\mathbf{H}^T$ are related by a similarity transformation, and so they must have identical eigenvalues. But $\mathbf{H}^T$ and $\mathbf{H}$ also have identical eigenvalues. In addition, the eigenvalues of a matrix and its inverse must be reciprocals of each other. These observations lead to the following conclusion: the eigenvalues of $\mathbf{H}$ must occur in reciprocal pairs. That is, if μ is an eigenvalue of $\mathbf{H}$, then $1/\mu$ is also an eigenvalue of $\mathbf{H}$.

We now collect all of the eigenvalues of $\mathbf{H}$ which are inside the unit circle into a diagonal matrix $\mathbf{E}$, and partition the corresponding eigenvectors as follows

$$\begin{bmatrix} \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{R}^{-1}\mathbf{\Gamma}^T\mathbf{\Phi}^{-T}\mathbf{Q} & -\mathbf{\Gamma}\mathbf{R}^{-1}\mathbf{\Gamma}^T \\ -\mathbf{\Phi}^{-T}\mathbf{Q} & \mathbf{\Phi}^{-T} \end{bmatrix} \begin{bmatrix} \mathbf{X}_I \\ \mathbf{Y}_I \end{bmatrix} = \begin{bmatrix} \mathbf{X}_I \\ \mathbf{Y}_I \end{bmatrix} \mathbf{E}.$$

The matrix $\mathbf{E}^{-1}$ contains the reciprocals of the eigenvalues in $\mathbf{E}$, and these numbers are the eigenvalues of $\mathbf{H}$ which are outside the unit circle. If we partition the corresponding eigenvectors we get

$$\begin{bmatrix} \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{R}^{-1}\mathbf{\Gamma}^T\mathbf{\Phi}^{-T}\mathbf{Q} & -\mathbf{\Gamma}\mathbf{R}^{-1}\mathbf{\Gamma}^T \\ -\mathbf{\Phi}^{-T}\mathbf{Q} & \mathbf{\Phi}^{-T} \end{bmatrix} \begin{bmatrix} \mathbf{X}_O \\ \mathbf{Y}_O \end{bmatrix} = \begin{bmatrix} \mathbf{X}_O \\ \mathbf{Y}_O \end{bmatrix} \mathbf{E}^{-1}.$$

Thus a complete eigen decomposition of $\mathbf{H}$ is given by

$$\mathbf{H} \begin{bmatrix} \mathbf{X}_I & \mathbf{X}_O \\ \mathbf{Y}_I & \mathbf{Y}_O \end{bmatrix} = \begin{bmatrix} \mathbf{X}_I & \mathbf{X}_O \\ \mathbf{Y}_I & \mathbf{Y}_O \end{bmatrix} \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^{-1} \end{bmatrix}$$

or

$$\mathbf{H} = \begin{bmatrix} \mathbf{X}_I & \mathbf{X}_O \\ \mathbf{Y}_I & \mathbf{Y}_O \end{bmatrix} \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{X}_I & \mathbf{X}_O \\ \mathbf{Y}_I & \mathbf{Y}_O \end{bmatrix}^{-1}. \tag{10.44}$$

We now substitute (10.44) into (10.41) to obtain

$$\begin{bmatrix} \mathbf{x}[k+1] \\ \boldsymbol{\lambda}[k+1] \end{bmatrix} = \begin{bmatrix} \mathbf{X}_I & \mathbf{X}_O \\ \mathbf{Y}_I & \mathbf{Y}_O \end{bmatrix} \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{X}_I & \mathbf{X}_O \\ \mathbf{Y}_I & \mathbf{Y}_O \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathbf{x}[k] \\ \boldsymbol{\lambda}[k] \end{bmatrix}. \tag{10.45}$$

If we define a new state/costate vector

$$\begin{bmatrix} \bar{\mathbf{x}}[k] \\ \bar{\boldsymbol{\lambda}}[k] \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{X}_I & \mathbf{X}_O \\ \mathbf{Y}_I & \mathbf{Y}_O \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{x}[k] \\ \boldsymbol{\lambda}[k] \end{bmatrix} \tag{10.46}$$

then (10.45) becomes

$$\begin{bmatrix} \bar{\mathbf{x}}[k+1] \\ \bar{\boldsymbol{\lambda}}[k+1] \end{bmatrix} = \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^{-1} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}}[k] \\ \bar{\boldsymbol{\lambda}}[k] \end{bmatrix}.$$

This is a decoupled system of equations which can be solved to yield

$$\begin{aligned} \bar{\mathbf{x}}[k] &= \mathbf{E}^k \bar{\mathbf{x}}[0] \\ \bar{\boldsymbol{\lambda}}[k] &= \mathbf{E}^{-k} \bar{\boldsymbol{\lambda}}[0]. \end{aligned} \tag{10.47}$$

Recall that the eigenvalues in $\mathbf{E}^{-1}$ lie outside the unit circle. The second equation in (10.47) tells us that if $\bar{\boldsymbol{\lambda}}[0] \neq \mathbf{0}$, then $\bar{\boldsymbol{\lambda}}[k]$ goes to infinity as k goes to infinity. It is easy to see using (10.45) and (10.46) that if $\bar{\boldsymbol{\lambda}}[k]$ goes to infinity, this means that $\mathbf{x}[k]$ goes to infinity. This is clearly not the optimal solution to the regulation problem whose goal is to drive $\mathbf{x}[k]$ to zero! Here is where the simplification occurs: *in the steady-state optimal*

control problem (i.e. $N \longrightarrow \infty$) it is necessary that $\bar{\boldsymbol{\lambda}}[0] = 0$. From (10.47) this means that $\bar{\boldsymbol{\lambda}}[k] = 0$ for all k.

With this condition, we can rewrite (10.46) as follows

$$\begin{bmatrix} \mathbf{x}[k] \\ \boldsymbol{\lambda}[k] \end{bmatrix} = \begin{bmatrix} \mathbf{X}_I & \mathbf{X}_O \\ \mathbf{Y}_I & \mathbf{Y}_O \end{bmatrix} \begin{bmatrix} \bar{\mathbf{x}}[k] \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_I \\ \mathbf{Y}_I \end{bmatrix} \bar{\mathbf{x}}[k]$$

and substituting in the first equation from (10.47) we obtain

$$\begin{aligned} \mathbf{x}[k] &= \mathbf{X}_I \mathbf{E}^k \bar{\mathbf{x}}[0] \\ \boldsymbol{\lambda}[k] &= \mathbf{Y}_I \mathbf{E}^k \bar{\mathbf{x}}[0]. \end{aligned}$$

From these equations we see that

$$\begin{aligned} \boldsymbol{\lambda}[k+1] &= \mathbf{Y}_I \mathbf{E}^{k+1} \bar{\mathbf{x}}[0] \\ &= \mathbf{Y}_I \mathbf{E} \mathbf{X}_I^{-1} \mathbf{X}_I \mathbf{E}^k \bar{\mathbf{x}}[0] \\ &= \mathbf{Y}_I \mathbf{E} \mathbf{X}_I^{-1} \mathbf{x}[k]. \end{aligned} \tag{10.48}$$

In other words, $\boldsymbol{\lambda}[k+1]$ can be obtained directly from $\mathbf{x}[k]$, without the use of the auxiliary system shown in Fig. 10.3.

Note that since the optimal regulator drives the state vector to zero, then from (10.48), it must be true that $\boldsymbol{\lambda}[k]$ goes to zero so that the boundary condition $\boldsymbol{\lambda}[N]$ is satisfied when $N \longrightarrow \infty$. We can now redraw Fig. 10.3 using (10.48) to create $\boldsymbol{\lambda}[k+1]$. The result is shown in Fig. 10.4.

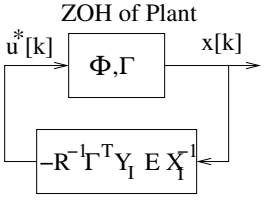


Figure 10.4 The optimal regulator that minimizes (10.1).

Notice that the optimal control is obtained by state feedback

$$\mathbf{u}^*[k] = -\mathbf{K}\mathbf{x}[k]$$

where the feedback gain matrix $\mathbf{L}$ is given by

$$\mathbf{K} = \mathbf{R}^{-1} \boldsymbol{\Gamma}^T \mathbf{Y}_I \mathbf{E} \mathbf{X}_I^{-1}. \tag{10.49}$$

The optimal control problem yields a unique feedback gain matrix for multivariable systems, in contrast to the pole placement technique. The matrix $\mathbf{K}$ computed in (10.49) is theoretically real valued, even though the matrices $\mathbf{Y}_I$, $\mathbf{X}_I$, and $\mathbf{E}$ could have complex elements. In order to prevent numerical round-off error from introducing a small imaginary part into the calculated $\mathbf{L}$ matrix, we separate the real and imaginary parts of any complex eigenvalues and eigenvectors. The details of how to do this, as well as a summary of the procedure for calculating optimal feedback gains is given in Table 10.3.

1. Given the ZOH equivalent $(\mathbf{\Phi}, \mathbf{\Gamma})$ of an n^{th}-order plant and symmetric matrices $\mathbf{Q}$ and $\mathbf{R}$ defining the quadratic cost function, form the Hamiltonian matrix shown below:

$$\mathbf{H} = \begin{bmatrix} \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{R}^{-1}\mathbf{\Gamma}^T\mathbf{\Phi}^{-T}\mathbf{Q} & -\mathbf{\Gamma}\mathbf{R}^{-1}\mathbf{\Gamma}^T\mathbf{\Phi}^{-T} \\ -\mathbf{\Phi}^{-T}\mathbf{Q} & \mathbf{\Phi}^{-T} \end{bmatrix}.$$

2. Calculate the eigenvalues and eigenvectors of $\mathbf{H}$. Put the eigenvalues which have magnitude less than one into a diagonal matrix $\mathbf{E}$ and put the corresponding eigenvectors into a matrix. Partition the eigenvector matrix into the first n and the last n rows as shown below:

$$\begin{bmatrix} \mathbf{X}_I \\ \mathbf{Y}_I \end{bmatrix}.$$

3. Replace every pair of complex-conjugate columns of $\mathbf{X}_I$ by the real part and the imaginary part of one of the columns.
4. Form the matrix $\mathbf{W} = \mathbf{Y}_I\mathbf{E}$. Replace every pair of complex-conjugate columns of $\mathbf{W}$ by the real part and the imaginary part of one of the columns.
5. Calculate the optimal feedback gain matrix

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{\Gamma}^T\mathbf{W}\mathbf{X}_I^{-1}.$$

Table 10.3 Procedure to calculate optimal feedback gains.

EXAMPLE 10.3

We wish to design a regulator for the inverted pendulum on a cart system by optimizing a quadratic cost function. In this example we pick several different cost functions and evaluate the corresponding feedback gains.

The state equations for the plant are given in Example 6.9 on page 262 where a regulator was designed using pole placement. As in that example, we use a sampling interval of 0.1 seconds to obtain the ZOH equivalent. In Example 6.9 the vector of feedback gains which placed the closed-loop poles at the roots of a 4-th order Bessel polynomial scaled to achieve a settling time of 0.95 seconds was calculated to be

$$\mathbf{K}_p = [\,23.3255 \quad 4.7691 \quad -0.0288 \quad -0.0240\,].$$

The subscript on this gain vector refers to the fact that it was calculated using the pole-placement formula. Several other gain vectors will be calculated using optimal control theory, and these will be subscripted with different numbers.

To use optimal control theory, we must pick $\mathbf{Q}$ and $\mathbf{R}$ which define the cost function. To begin the design process we pick

$$\mathbf{Q} = \mathbf{I}\,, \quad \mathbf{R} = 1.$$

Using the procedure in Table 10.3 we obtain the following vector of feedback gains

$$\mathbf{K}_1 = [\,104.1924 \quad 21.6786 \quad -0.0406 \quad -0.0676\,].$$

Note that the first two gains of $\mathbf{K}_1$ are much larger than the corresponding gains in $\mathbf{K}_p$. In fact, if the system is initialized as follows

$$\mathbf{x}[0] = \begin{bmatrix} 0.17 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

which corresponds to an initial pendulum angle of $10°$, then the initial value of the plant input signal would be

$$u[0] = -\mathbf{K}_1\mathbf{x}[0] = -17.7 \text{ volts.}$$

For the same initial conditions, the feedback gain vector $\mathbf{K}_p$ results in an initial plant input of less than 4 volts in magnitude.

We can penalize large input signals by increasing the value of $\mathbf{R}$ in the cost function. We choose

$$\mathbf{Q} = \mathbf{I}\,, \quad \mathbf{R} = 10,000,000$$

and use the procedure in Table 10.3 to calculate the following vector of feedback gains

$$\mathbf{K}_2 = [\,22.5313 \quad 4.6879 \quad -0.0003 \quad -0.0187\,].$$

This gain vector looks similar to $\mathbf{K}_p$ with the exception of the third gain, which corresponds to motor position. The motor position gain of $\mathbf{K}_2$ is about 100 times smaller that the motor position gain of $\mathbf{K}_p$. A simulation of the regulator with $\mathbf{K}_2$ shows that the motor position has a very large settling time – about 150 seconds!

In order to make the motor position return to zero more quickly we must increase the weight for motor position in the cost function. We do this by increasing the value of the 3,3 element of $\mathbf{Q}$. We choose

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1,000 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\,, \quad \mathbf{R} = 10,000,000$$

and calculate the following vector of feedback gains

$$\mathbf{K}_3 = [\,27.1263 \quad 5.6440 \quad -0.0095 \quad -0.0229\,].$$

We simulate the response of the regulator with $\mathbf{K}_3$ to the initial conditions shown above. The results are shown in Fig. 10.5. Note that the motor variables have a larger settling time than the pendulum variables, but the response is smooth and the plant input is not excessive. The gain margins for this regulator $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{K}_3)$ are found to be

$$-6.67 \leq GM \leq 25.7 \text{ dB}$$

and the phase margin is $57°$. These stability margins are much better than those reported in Example 6.9 for $(\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{K}_p)$.

Although the feedback gain vector $\mathbf{K}_3$ was calculated using the LQ procedure, we could have obtained $\mathbf{K}_3$ from the pole-placement formula with the appropriate choice of closed-loop poles. The z-plane closed-loop poles corresponding to $\mathbf{L}_3$ are the eigenvalues of $\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}_3$. We can map these poles into the s-plane using the

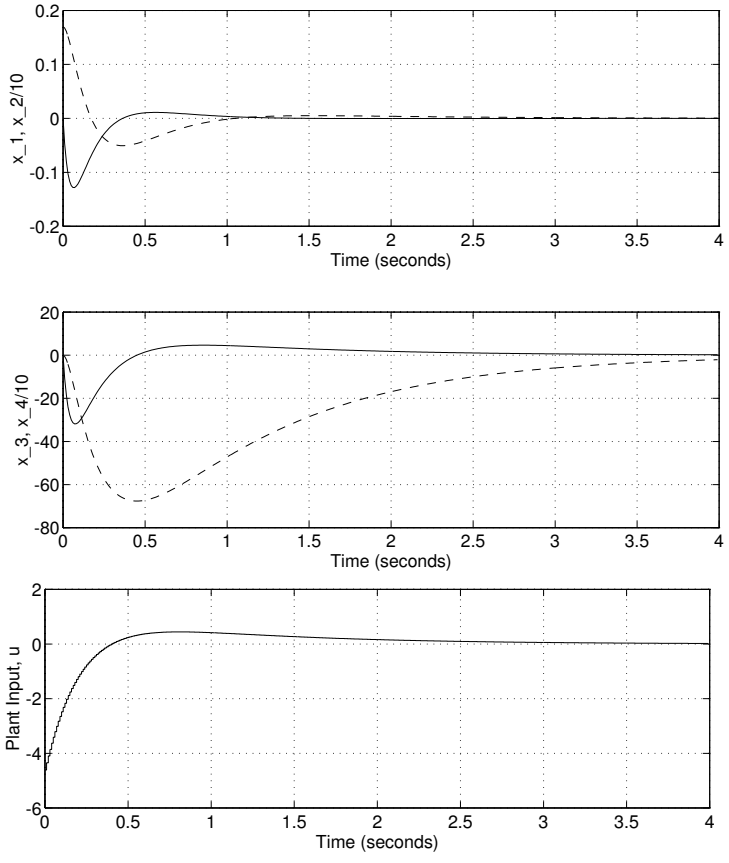


Figure 10.5 State variables and plant input for Example 10.3. Top graph shows x_1 (dashed line) and $x_2/10$ (solid line). Middle graph shows x_3 (dashed line) and $x_4/10$ (solid line).

inverse of the ZOH pole mapping formula (see (4.61) on page 181). The resulting s-plane roots are

$$\frac{1}{T}\ln(\text{eig}(\mathbf{\Phi}-\mathbf{\Gamma K}_3)) = -24.9915\ ,\quad -1.0563\ ,\quad -4.8062 \pm j0.0004.$$

■

The development of optimal control theory in this chapter has been in the context of the regulator problem. However the LQ formula for regulator feedback gains can be used in the design of tracking systems. The reason is that the procedure to design tracking systems presented in Chapters 8 and 9 requires the calculation of feedback gains to regulate the "design model," which consists of the cascade combination of the plant and the additional dynamics. In the next example we use the LQ formula in the design of a multivariable tracking system for an aircraft.

EXAMPLE 10.4

A state-space model for a certain experimental aircraft was given in Example 9.10 on page 436. In that example a tracking system for this aircraft was designed using pole placement, while in the this example we use optimal control theory. The state variables of the plant are The state variables are

$$x_1 = \text{roll angle}$$

$$x_2 = \text{perturbation velocity in } x \text{ direction}$$

$$x_3 = \text{velocity in } z \text{ direction}$$

$$x_4 = \text{roll rate}.$$

The inputs are

$$u_1 = \text{canard angle}$$

$$u_2 = \text{flaperon angle}$$

$$u_3 = \text{thrust (pounds)}.$$

The outputs to be controlled are

$$y_1 = \text{pitch angle}$$

$$y_2 = \text{flight-path angle}$$

$$y_3 = \text{aircraft velocity (ft/s)}.$$

All angles are measured in degrees.

In order for the outputs to track step commands with zero steady-state error, additional dynamics consisting of three parallel integrators are cascaded with the plant. The design model is thus 7-th order; the first four state variabless of the design model are the four plant state variables listed above. The last three state variables of the design model are the integrator state variables. We need to specify a 7×7 matrix $\mathbf{Q}$ and

a 3×3 matrix $\mathbf{R}$ in order to calculate a feedback gain matrix using the procedure given in Table 10.3. Guidelines for choosing $\mathbf{Q}$ and $\mathbf{R}$ are beyond the scope of this book, but some may be found in [...]. Here we restrict $\mathbf{Q}$ and $\mathbf{R}$ to be diagonal matrices and use a trial-and-error procedure as in the previous example. We end up with the following matrices:

$$\mathbf{Q} = \text{ diag}(.0025, .0025, 0, 0, 10, 10, 10),$$

$$\mathbf{R} = \text{ diag}(.3, 30, .3).$$

Using the procedure in Table 10.3 we calculate the followng matrix of feedback gains

$$\mathbf{K} = \begin{bmatrix} 37.8260 & -0.6619 & -0.0219 & 2.9068 & 1.3605 & 0.4643 & 1.0709 \\ 19.9951 & -1.5895 & -0.0463 & 0.9566 & 0.0274 & 0.4870 & -0.2474 \\ -181.6417 & 80.5708 & 1.5020 & -50.1487 & -9.9483 & 8.1667 & 10.4490 \end{bmatrix}$$

This matrix is partitioned into two matrices for use in the digital tracking system shown in Fig. 9.4. $\mathbf{K}_1$ consists of the first 3 columns of $\mathbf{K}$, and $\mathbf{K}_2$ consists of the last 4 columns of $\mathbf{K}$.

The stability margins for this tracking system are calculated to be:

Input #1: UGM=3.84 dB, LGM=-26.4 dB, PM=30°

Input #2: UGM=24.8 dB, LGM<-30.1 dB, PM=106°.

Input #3: UGM=20.2 dB, LGM<-30.1 dB, PM=72°.

These margins are even better than those reported in Example 9.10. We can compute the closed-loop poles of the system, and map them back into the s-plane as follows:

$$\frac{1}{T}\ln(\text{eig}(\mathbf{\Phi} - \mathbf{\Gamma}\mathbf{K}_3)) = -9.3078,\ -56.8157 \pm j47.7067,\ -4.3369 \pm j1.4205,$$

$$-1.6574 \pm j0.8553.$$

These numbers can be compared with the closed-loop poles chosen in Example 9.10 to use with the pole placement algorithm.

Finally, we check the transient response for the three different maneuvers of the aircraft that were simulated in Example 9.10. The first maneuver is called pitch pointing. The aircraft is flying straight and level at a given velocity, and a step input of 2° in the pitch angle y_1 is given while the other two outputs are commanded to remain unchanged. Thus the reference input vector is

$$\mathbf{r}[k] = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix},\ k = 0, 1, \cdots.$$

The responses and inputs are shown in Fig. 10.6. The pitch angle response has no overshoot. Also the system is nicely decoupled: the maximum flight path angle perturbation (y_2) is 0.2°, and the maximum velocity perturbation (y_3) is about 1 ft/s from a nominal velocity of 881 ft/s.

The second maneuver is called vertical translation. The pitch angle and velocity should remain constant, while the aircraft rises vertically (flight path angle increases 2°). Thus the reference input vector is

$$\mathbf{r}[k] = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix},\ k = 0, 1, \cdots.$$

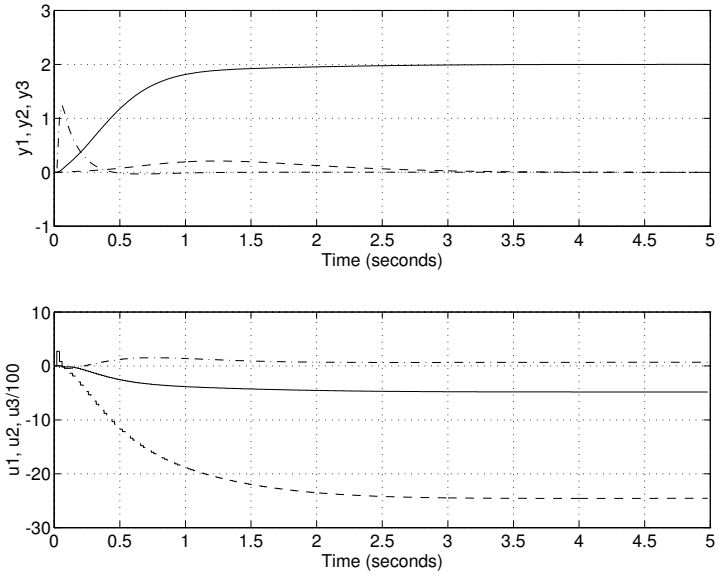


Figure 10.6 Outputs and inputs for Example 10.4 for the pitch pointing maneuver. y_1 and u_1 are shown by solid lines, y_2 and u_2 by dashed lines, and y_3 and u_3 by dash-dot lines.

The responses and inputs are shown in Fig. 10.7.

The flight path angle response has no overshoot, and there is very little coupling to the other two outputs.

The final maneuver is called straight climb. The pitch and flight path angles should simultaneously increase 2° while holding velocity constant. The reference input vector is

$$\mathbf{r}[k] = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}, \; k = 0, 1, \cdots.$$

The responses and inputs are shown in Fig. 10.8.

The pitch and flight path angle responses are no longer identical, as in Example 9.10, but they both have a settling time of about 2 seconds. The velocity remains essentially constant.

■

10.3 Chapter Summary

In this chapter we have derived an algorithm for computing regulator feedback gains. The algorithm finds the gains which minimize a quadratic function of the states and inputs of the plant. The resulting regulator is called a linear quadratic (LQ) regulator. The LQ design algorithm does not allow the user to specify desired closed-loop pole locations. Rather, the user specifies two matrices $\mathbf{Q}$ and $\mathbf{R}$ which define the quadratic function to be minimized. When the function is defined by an infinite summation (over all time), the optimal regulator is obtained by state feedback with constant feedback gains. The matrix $\mathbf{K}$ of feedback gains can be computed using the eigenvectors of a certain symplectic matrix. The formula

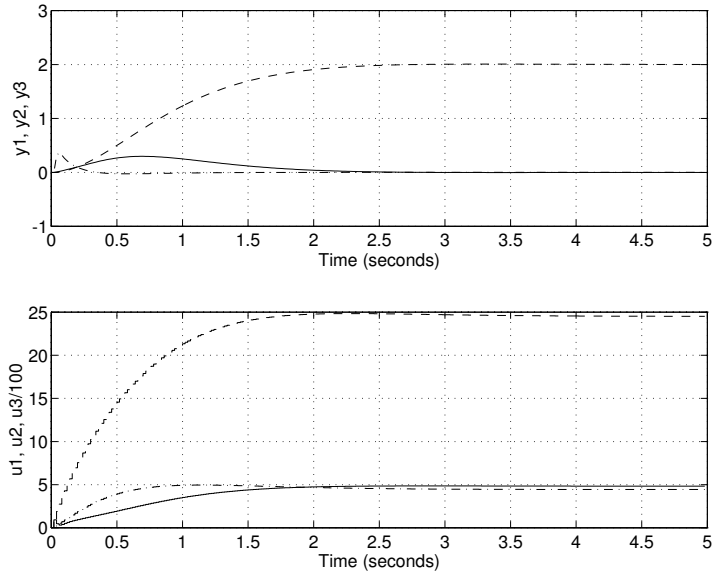


Figure 10.7 Outputs and inputs for Example 10.4 for the vertical translation maneuver. y_1 and u_1 are shown by solid lines, y_2 and u_2 by dashed lines, and y_3 and u_3 by dash-dot lines.

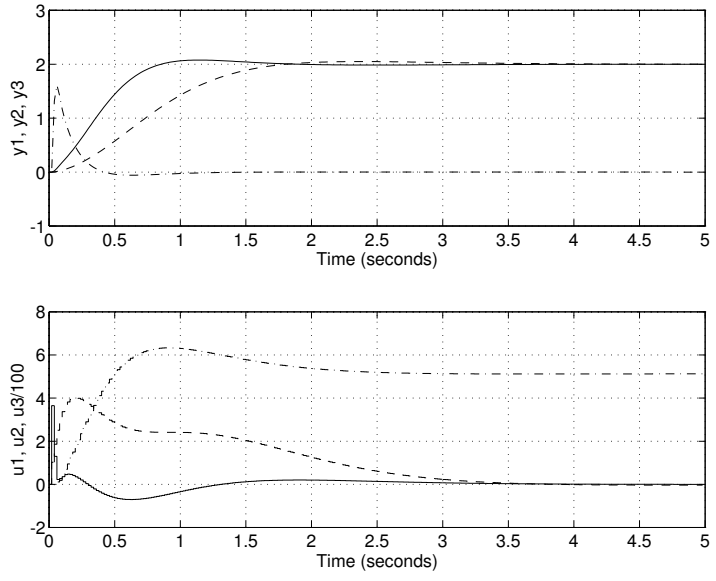


Figure 10.8 Outputs and inputs for Example 10.4 for the straight climb maneuver. y_1 and u_1 are shown by solid lines, y_2 and u_2 by dashed lines, and y_3 and u_3 by dash-dot lines.

for $\mathbf{K}$ is the same for single-input or multi-input systems, and results in a unique solution for $\mathbf{K}$.

10.4 Problems

The following problems are all related to examples or problems in previous chapters. For each system listed below, design a regulator or tracking system using the LQ formula to calculate the feedback gains. Begin with $\mathbf{Q} = \mathbf{I}$ and $\mathbf{R} = \mathbf{I}$ and then vary the diagonal elements of the weighting matrices to obtain different designs. Compare the transient performance and stability margins of your results with those obtained in the previous chapters.

1. Design a regulator for the system with structural flexibilty given in Example 6.10 on page 265. Use a sampling interval of $T = 0.1$ seconds. The desired settling time is 3 seconds.

2. Design a regulator for the hydraulic positioning system given in Chapter 6, Problem 9 on page 285. Use a sampling interval of $T = 0.01$ seconds. The desired settling time is 0.35 seconds.

3. Design a regulator for the ship stabilization system given in Chapter 6, Problem 10 on page 286. Use a sampling interval of $T = 0.1$ seconds. The desired settling time is 4.5 seconds.

4. Repeat Problems 4 – 9 of Chapter 9, using the LQ formula to compute the feedback gains.

REFERENCES

1. B.D.O. Anderson and J.B. Moore. *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, Englewood Cliffs, NJ, 1990.
2. A.N. Andry, Jr., E.Y. Shapiro, and J.C. Chung. Eigenstructure assignment for linear systems. *IEEE Transactions on Aerospace and Electronic Systems*, 19:711–729, 1983.
3. M. Araki and T. Hagiwara. Pole assignment by multirate sampled-data output feedback. *International Journal of Control*, 44:1661–1673, 1986.
4. K.J. Astrom, P. Hagender, and J. Sternby. Zeros of sampled systems. *Automatica*, 20:31–38, 1984.
5. Y. Bar-Ness and G. Langholz. Preservation of controllability under sampling. *International Journal of Control*, 22:39–47, 1975.
6. M.C. Berg, N. Amit, and J.D. Powell. Multirate digital control system design. *IEEE Transactions on Automatic Control*, 33:1139–1150, 1988.
7. S.P. Bhattacharyya and J.B. Pearson. On error systems and the servomechanism problem. *International Journal of Control*, 15:1041–1062, 1972.
8. J.J. Bongiorno, Jr. On the design of observers for insensitivity to plant parameter variations. *International Journal of Control*, 18:597–605, 1973.
9. J.J. Bongiorno, Jr. A recursive algorithm for computing the partial fraction expansion of rational functions having multiple poles. *IEEE Trans. Automat. Control*, AC-29:650–652, 1984.
10. W. L. Brogan. *Modern Control Theory*. Prentice-Hall, Englewood Cliffs, NJ, third edition, 1991.
11. J.R. Broussard, D.R. Downing, and W.H. Bryant. Design and flight testing of a digital optimal control general aviation autopilot. *Automatica*, 21:23–34, 1985.
12. A.E. Bryson and Y.C. Ho. *Applied Optimal Control*. Hemisphere, New York, 1975.

13. J. A. Cadzow. *Discrete-Time Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1973.

14. K.C. Cheok, N.K. Loh, and J.B. Ho. Continuous-time optimal robust servo-controller design with internal model principle. *International Journal of Control*, 48:1993–2010, 1988.

15. C.M. Close and D.K. Frederick. *Modeling and Analysis of Dynamic Systems*. Houghton-Miflin, Boston, MA, 1986.

16. E.J. Davison. On pole assignement in linear systems with incomplete state feedback. *IEEE Transactions on Automatic Control*, AC-15:348–351, 1970.

17. E.J. Davison. The output control of linear time-invariant multivariable systems with unmeasurable arbitrary disturbances. *IEEE Transactions on Automatic Control*, AC-17:621–630, 1972.

18. E.J. Davison and A. Goldenberg. Robust control of a general servomechanism problem: The servo compensator. *Automatica*, 11:461–471, 1975.

19. E.J. Davison and P. Patel. Application of the robust servomechanism controller to systems with periodic tracking/disturbance signals. *International Journal of Control*, 47:111–127, 1988.

20. E.J. Davison and H.W. Smith. A note on the design of industrial regulators: Integral feedback and feedforward controllers. *Automatica*, 10:329–332, 1974.

21. J.J. D'Azzo and C.H. Houpis. *Linear Control System Analysis and Design*. McGraw-Hill, New York, NY, third edition, 1988.

22. B.W. Dickinson. *Systems: Analysis, Design, and Computation*. Prentice-Hall, Englewood Cliffs, NJ, 1991.

23. Richard C. Dorf. *Modern Control Systems*. Addison-Wesley, Reading, MA, sixth edition, 1992.

24. D.R. Downing and W.H. Bryant. Flight test of a digital controller used in a helicopter autoland system. *Automatica*, 23:295–300, 1987.

25. J.C. Doyle, B.A. Francis, and A.R. Tannenbaum. *Feedback Control Theory*. Macmillan, New York, NY, 1992.

26. J.C. Doyle and G. Stein. Robustness with observers. *IEEE Transactions on Automatic Control*, AC-24:607–611, 1979.

27. B.A. Francis, O.A. Sebakhy, and W.M. Wonham. Synthesis of multivariable regulators: the internal model principle. *Journal of Applied Mathematics and Optimization*, 1:64–86, 1975.

28. B.A. Francis and W.M. Wonham. The internal model principle for linear multivariable regulators. *Journal of Applied Mathematics and Optimization*, 2:170–194, 1975.

29. B.A. Francis and W.M. Wonham. On the role of transmission zeros in linear multivariable regulators. *International Journal of Control*, 22:657–681, 1975.

30. B.A. Francis and W.M. Wonham. The internal model principle of control theory. *Automatica*, 12:457–465, 1976.

31. Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, Reading, MA, 1991.

32. Gene F. Franklin, J. David Powell, and Michael L. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, Reading, MA, second edition, 1990.

33. Bernard Friedland. *Control System Design*. McGraw-Hill, New York, NY, 1986.

34. K. Furuta, H. Kajiwara, and K. Kosuge. Digital control of a double inverted pendulum on an inclined rail. *International Journal of Control*, 32:907–924, 1980.

35. K. Furuta, T. Ochiai, and N. Ono. Attitude control of a triple inverted pendulum. *International Journal of Control*, 39:1351–1365, 1984.

36. J.A. Gibson and T.T. Ha. Further to the preservation of controllability under sampling. *International Journal of Control*, 31:1013–1026, 1980.

37. Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, second edition, 1989.

38. H. Hanselmann. Implementation of digital controllers–a survey. *Automatica*, 23:7–32, 1987.

39. S. Hara, H. Katori, and R. Kondo. The relationship between real poles and real zeros in siso sampled data systems. *IEEE Transactions on Automatic Control*, 34:632–635, 1989.

40. C.A. Harvey and G. Stein. Quadratic weights for asymptotic regulator properties. *IEEE Transactions on Automatic Control*, AC-23:378–387, 1978.

41. P. Hippe and C. Wurmthaler. Reduced-order observers for linear multivariable systems with disturbance and tracking signal accomodation. *International Journal of Control*, 38:831–842, 1983.

42. K.M. Hoffman and R.A. Kunze. *Linear Algebra*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

43. G.H. Hostetter. *Digital Control System Design*. Holt, Rinehart, and Winston, New York, 1988.

44. C.H. Houpis and G.B. Lamont. *Digital Control Systems, Theory, Hardware, Software*. McGraw-Hill, New York, NY, first edition, 1985.

45. C.H. Houpis and G.B. Lamont. *Digital Control Systems, Theory, Hardware, Software*. McGraw-Hill, New York, NY, second edition, 1992.

46. L.B. Jackson. *Signals, Systems, and Transforms*. Addison-Wesley, Reading, MA, 1991.

47. C.D. Johnson. Optimal control of the linear regulator with constant disturbances. *IEEE Transactions on Automatic Control*, AC-13:416–421, 1968.

48. C.D. Johnson. Disturbance-utilizing controllers for noisy measurements and disturbances ii: The discrete-time case. *International Journal of Control*, 39:869–877, 1984.

49. C.D. Johnson. Optimal initial conditions for full-order observers. *International Journal of Control*, 48:857–864, 1988.

50. T. Kailath. *Linear Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1980.

51. P. Katz. *Digital Control Using Microprocessors*. Prentice-Hall, Englewood Cliffs, NJ, 1981.

52. J. Kautsky, N.K. Nichols, and P. Van Dooren. Robust pole assignment in linear state feedback. *International Journal of Control*, 41:1129–1155, 1985.

53. Z. Kowalczuk. Finite register length issue in the digital implemenatation of discrete pid algorithms. *Automatica*, 25:393–405, 1989.

54. N.J. Krikelis and S.K. Barkas. Design of tracking systems subject to actuator saturation and integrator wind-up. *International Journal of Control*, 39:667–682, 1984.

55. B.C. Kuo. *Automatic Control Systems*. Prentice-Hall, Englewood Cliffs, NJ, sisxth edition, 1991.

56. F.L. Lewis. *Optimal Control*. John Wiley, New York, 1986.

57. L. Ljung. *System Identification – Theory for the User*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

58. J. Luckel and R. Kasper. Optimization of the disturbance and reference characteristics of linear time-invariant systems by stationary compensation of unstable excitation models. *International Journal of Control*, 41:259–269, 1985.

59. D.G. Luenberger. An introduction to observers. *IEEE Transactions on Automatic Control*, AC-16:596–602, 1971.

60. R.H. Middleton and G.C. Goodwin. Improved finite word length characteristics in digital control using delta operators. *IEEE Transactions on Automatic Control*, AC-31:1015–1021, 1986.

61. R.K. Miller, M.S. Mousa, and A.N. Michel. Quantization and overflow effects in digital implementations of linear dynamic controllers. *IEEE Transactions on Automatic Control*, 33:698–704, 1988.

62. G.S. Miminis and C.C. Paige. An algorithm for the pole assignment of time invariant linear systems. *International Journal of Control*, 35:341–354, 1982.

63. C.B. Moler and C.F. Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 20:801–836, 1978.

64. M. Moonen, B. De Moor, L. Vandenberghe, and J. Vandewalle. On- and off-line identification of linear state-space models. *International Journal of Control*, 49:219–232, 1989.

65. M. Moonen and J. Vandewalle. Qsvd approach to on- and off-line state-space identification. *International Journal of Control*, 5:1133–1146, 1990.

66. M. Morari and E. Zafiriou. *Robust Process Control*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

67. S. Mori, H. Nishihara, and K. Furuta. Control of unstable mechanical system: Control of pendulum. *International Journal of Control*, 23:673–692, 1976.

68. P. Moroney. *Issues in the Implementation of Digital Feedback Compensators*. MIT Press, Cambridge, MA, 1983.

69. N.K. Nichols. On computational algorithms for pole assigment. *IEEE Transactions on Autotmatic Control*, AC-31:643–645, 1986.

70. K. Ogata. *Modern Control Engineering*. Prentice-Hall, Englewood Cliffs, NJ, second edition, 1990.

71. Y. Okazaki. A micro-positioning tool post using a piezoelectric actuator for diamond turning machines. *Precision Engineering*, 12:151–156, 1990.

72. Alan V. Oppenheim, Allan A. Willsky, and Ian T. Young. *Signals and Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1983.

73. Charles L. Phillip and Jr. H. T. Nagle. *Digital Control System Analysis and Design*. Prentice-Hall, Englewood Cliffs, NJ, 1984.

74. Charles L. Phillip and R.D. Harbor. *Feedback Control Systems*. Prentice-Hall, Englewood Cliffs, NJ, second edition, 1991.

75. B. Porter and A. Bradshaw. Design of error-actuated controllers for multivariable plants with unknown dynamics and unmeasurable outputs. *International Journal of Control*, 37:1–16, 1983.

76. H.-P. Preuss. Perfect steady-state tracking and distubance rejection by constant state feedback. *International Journal of Control*, 35:75–94, 1982.

77. G.V. Rao. *Complex Digital Control Systems*. Van Nostrand Reinhold Co., New York, 1979.

78. D.B. Ridgely, S.S. Banda, and J.J. D'Azzo. Decoupling of multivariable high gain tracking systems. *AIAA J. Guidance, Control, Dynamics*, 8:44–49, 1985.

79. M.A. Salmon. A robust servo-electric controller for brake force distribution. *Trans. of the ASME: Journal of Dynamic Systems, Measurement, and Control*, 112:442–447, 1990.

80. A.M. Schneider, J.T. Kaneshige, and F.D. Groutage. Higher order s-to-z mapping functions and their application in digitizing continuous-time filters. *Proceedings of the IEEE*, 79:1661–1674, 1991.

81. E.D. Sontag. *Mathematical Control Theory*. Springer-Verlag, New York, 1990.

82. R.F. Stengel, J.R.Broussard, and P.W. Berry. Digital flight control design for a tandem-rotor helicopter. *Automatica*, 14:301–312, 1978.

83. G.W. Stewart. *Matrix Computations*. Academic Press, New York, 1973.

84. G. Strang. *Linear Algebra and Its Applications*. Academic Press, New York, second edition, 1980.

85. P.M. Thompson. Gain and phase margins of multi-rate sampled-data feedback systems. *International Journal of Control*, 44:833–846, 1986.

86. E. Tse. Observer-estimators for discrete-time systems. *IEEE Transactions on Automatic Control*, AC-18:10–16, 1973.

87. C.-C. Tsui. New approach to robust observer design. *International Journal of Control*, 47:745–751, 1988.

88. T. Tsujisawa, H. Murayama, and H. Inada. Modern control theory application in high track density fdd. In *Proceedings of IEEE IECON'86*, pages 247–252, 1986.

89. C. F. Van Loan. Computing integrals involving the matrix exponential. *IEEE Trans. Automat. Control*, AC-23:395–404, 1978.

90. R.B. Walters. *Hydraulic and Electro-Hydraulic Control Systems*. Elsevier Applied Science, London and New York, 1991.

91. B. Wittenmark. Sampling of a system with a time delay. *IEEE Transactions on Automatic Control*, AC-30:507–510, 1985.

92. C.A. Wolfe and J.S. Meditch. A modified configuration for linear multivariable servomechanisms. *International Journal of Control*, 27:57–64, 1978.

93. W.A. Wolovich and H. Elliott. Discrete models for linear multivariable systems. *International Journal of Control*, 38:337–357, 1983.

94. P.C. Young and J.C. Willems. An approach to the linear multivariable servomechanism problem. *International Journal of Control*, 15:961–979, 1972.