CHAPTER 6

Design of Digital Control Systems Using State-Space Methods

6.1 INTRODUCTION

In Chapter 5, we discussed how to design digital controllers using transform techniques, methods now commonly designated as "classical design." The goal of this chapter is to solve the identical problem using the state-space formulation. The difference in the two approaches is entirely in the design method; the end result, a set of difference equations providing control, is identical. In fact, given the same set of specifications, the control equations should be very similar if not identical.

Advantages of the state-space formulation are especially apparent when designing controllers for Multi-Input, Multi-Output (MIMO) systems, that is, those with more than one control input and/or sensed output. However, state-space methods are also an aid in the design of controllers for Single-Input, Single-Output (SISO) systems because of the widespread use of CAD tools, which often rely heavily on this system representation. Chapters 2 and 3 have already demonstrated the advantages of the state-space formulation in using CAD packages for the computation of discrete equivalents. In this chapter, we will limit our state-space design efforts to SISO controllers, the same controllers found in Chapter 5 with classical methods. Techniques for MIMO design are discussed in Chapter 9.

In Chapter 5, two basic methods were described: *emulation* and *direct digital design.* The same two methods apply to the state-space formulation as well. Using emulation, one would design a continuous controller using state-space methods then, transform the controller to a discrete form by using one of the discrete equivalents from Chapter 4. The discussion of the method and its accuracy in Chapter 5 applies equally well here. Furthermore, the development in Chapter 4 used both classical and state space system descriptions in the computation of the equivalents. Therefore, no further discussion of emulation is required, and we will concentrate solely on the direct digital design method.

6.2 CONTROL-LAW DESIGN

In Chapter 2, we saw that the state-space description of a continuous system is given by (2.43),

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}u, \tag{6.1}$$

and (2.44),

$$y = \mathbf{H}\mathbf{x}. \tag{6.2}$$

We assume the control is applied from the computer by a ZOH as shown in Fig. 1.1. Therefore, (6.1) and (6.2) have an exact discrete representation as given by (2.57),

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}u(k),\\ y(k) &= \mathbf{H}\mathbf{x}(k),\end{aligned} \tag{6.3}$$

where

$$\mathbf{\Phi} = e^{\mathbf{F}T}, \tag{6.4a}$$

$$\mathbf{\Gamma} = \int_0^T e^{\mathbf{F}\eta} d\eta \mathbf{G}, \tag{6.4b}$$

Using CAD packages, one can easily transform between the classical transfer function of a continuous system, $G(s)$, to the state-space continuous description (see X-TF2SS in Table E.1), $\mathbf{F}, \mathbf{G}, \mathbf{H}$, and from there to the discrete (with ZOH) description (see X-C2D in Table E.1), $\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{H}$. For very simple systems, one could also compute by hand the transformations as shown in Chapter 2.

One of the attractive features of state-space design methods is that the procedure consists of two independent steps. The first step *assumes* that we have all the states at our disposal for feedback purposes. In general, of course, this would be a ridiculous assumption; a practical engineer would not, as a rule, find it necessary to purchase such a large number of sensors, especially because he or she knows that they would not be needed using classical design methods. The assumption that all states are available merely allows us to proceed with the first design step, namely, the control law. The remaining step is to design an "estimator" (or "observer"[1]), which estimates the entire state vector, given measurements of the portion of the state provided by (6.2). The final control algorithm will consist of a combination of the control law and the estimator with the control-law calculations based on the estimated states rather than on the actual states. In Section 6.4 we show that this substitution is reasonable and that the combined control law and estimator can give closed-loop dynamic characteristics that are unchanged from those assumed in designing the control law and estimator separately. The dynamic system we obtain from the combined control law and estimator is called the *controller*. The first step is to get a good control law.

A control law that has considerable convenience is simply the feedback of a linear combination of all the state elements, that is,

$$\boxed{u = -\mathbf{K}\mathbf{x} = -[K_1 K_2 \ldots] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix}.} \tag{6.5}$$

Note that this structure does not allow for a reference input to the system. The topology that we used all through Chapter 5 (Fig. 5.2) always included a reference input, r. The control law above (6.5) assumes that $r = 0$ and is, therefore, usually referred to as a *regulator*. Section 6.4 will discuss how one introduces reference inputs.

Substituting (6.5) in the difference equation (6.3), we have

$$\mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) - \mathbf{\Gamma}\mathbf{K}\mathbf{x}(k). \tag{6.6}$$

Therefore the z-transform of (6.6) is

$$(z\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{K})\mathbf{X}(z) = 0,$$

and the characteristic equation of the system with the hypothetical control law is

$$\boxed{\det| z\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma}\mathbf{K} | = 0.} \tag{6.7}$$

[1] The literature [Luenberger (1960)] commonly refers to these devices as "observers;" however, we feel that the term "estimator" is much more descriptive of their function because "observe" implies a direct measurement. In this book the term "estimator" is used but the reader can think of the terms interchangeably.

6.2.1 Pole Placement

The approach we wish to take at this point is *pole placement*; that is, having picked a control law with enough parameters to influence all the closed-loop roots, we will arbitrarily select the desired root locations of the closed-loop system and see if the approach will work. Although this approach can often lead to trouble in the design of complex systems (see Example 6.2, the discussion in Section 6.4, and Franklin, Powell, and Emami-Naeini, 1986), we use it here to illustrate the power of full state feedback. In Chapter 9, we will build on this idea to arrive at a more practical design methodology.

The control-law design, then, consists of finding the elements of $\mathbf{K}$ so that the roots of (6.7) are in the desired locations. Unlike classical design, where we iterated on parameters in the compensator (hoping) to find acceptable root locations, the full state feedback, pole-placement approach guarantees success and allows us to arbitrarily pick any root locations, providing that n roots are specified for an nth-order system.

Given desired root locations,[2] say,

$$z_i = \beta_1, \beta_2, \beta_3, \ldots,$$

the desired control-characteristic equation is

$$\boxed{\alpha_c(z) = (z - \beta_1)(z - \beta_2)(z - \beta_3) \cdots = 0.} \tag{6.8}$$

Equations (6.7) and (6.8) are both the characteristic equation of the controlled system; therefore, they must be identical, term by term. Thus we see that the required elements of $\mathbf{K}$ are obtained by matching the coefficients of each power of z in (6.7) and (6.8), and there will be n equations for an nth-order system.

Example 6.1: Suppose we want to design a control law for the satellite attitude-control system described by (2.45) with $\boldsymbol{x} = [x_1 \;\; x_2]$. Example 2.13 showed that the discrete model for this system is

$$\mathbf{\Phi} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{\Gamma} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}.$$

[2] Discussion of how one selects root locations will occur through the following examples and will be reviewed in Section 6.4.2. The results of the specification discussion in Chapter 5 can also be used to specify roots. Furthermore, a complete discussion of root selection is contained in Franklin, Powell, and Emami-Naeini (1986).

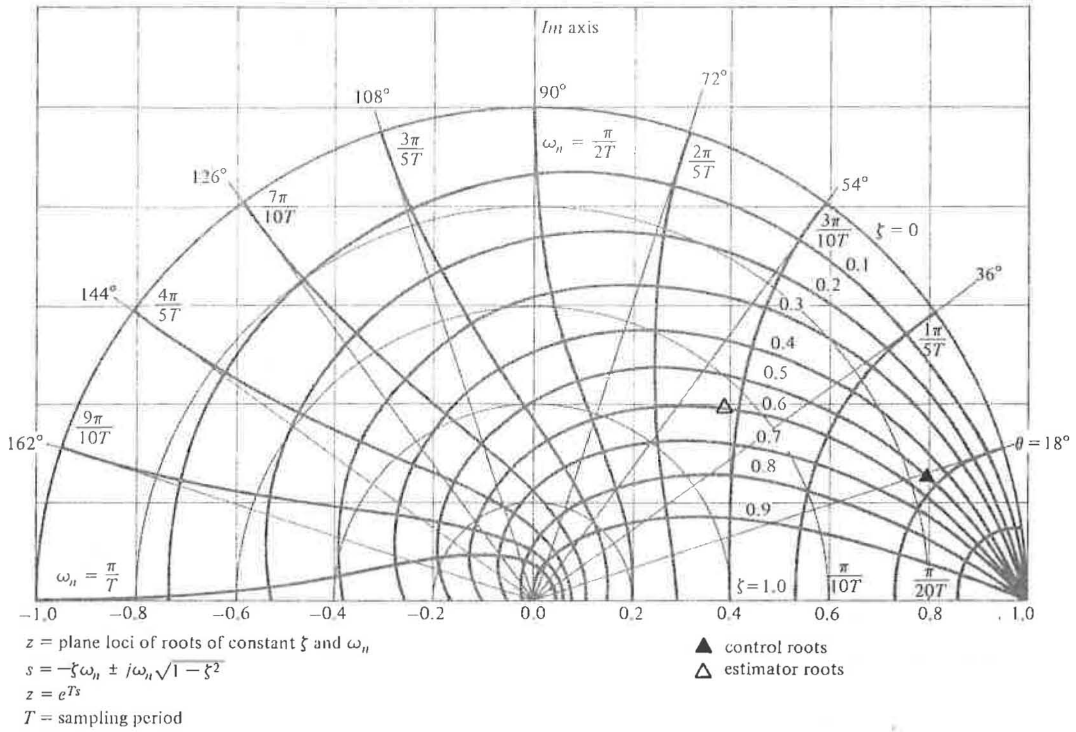

Figure 6.1 Desired root locations for satellite attitude-control system of Examples 6.1 and 6.4.

We want to pick z-plane roots of the closed-loop characteristic equation so that the equivalent s-plane roots have a damping ratio of $\zeta = 0.5$ and real part of $s = -1.8$ rad/sec (i.e., $s = -1.8 \pm j3.12$ rad/sec). Using $z = e^{sT}$ with a sample period of $T = 0.1$ sec, we find that $z = 0.8 \pm j0.25$, as shown in Fig. 6.1. The desired characteristic equation is then

$$z^2 - 1.6z + 0.70 = 0, \tag{6.9}$$

and the evaluation of (6.7) for any control law $\mathbf{K}$ leads to

$$\det \left| z \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} [K_1 \quad K_2] \right| = 0$$

or

$$z^2 + (TK_2 + (T^2/2)K_1 - 2)z + (T^2/2)K_1 - TK_2 + 1 = 0. \tag{6.10}$$

Equating coefficients in (6.9) and (6.10) with like powers of z, we obtain two simultaneous equations in' the two unknown elements of **K**:

$$TK_2 + (T^2/2)K_1 - 2 = -1.6,$$
$$(T^2/2)K_1 - TK_2 + 1 = 0.70,$$

which are easily solved for the coefficients and evaluated for $T = 0.1$ sec:

$$K_1 = \frac{0.10}{T^2} = 10, \qquad K_2 = \frac{0.35}{T} = 3.5.$$

The calculation of the gains using the method illustrated in the previous example becomes rather tedious when the order of the system (and therefore the order of the determinant to be evaluated) is greater than 2. A computer does not solve the tedium unless it is used to perform the algebraic manipulations necessary in expanding the determinant in (6.7) to obtain the characteristic equation. Therefore, other approaches have been developed to provided convenient computer-based solutions to this problem.

The algebra for finding the specific value of **K** is especially simple if the system matrices happen to be in the form associated with the block diagram of Fig. 2.8(c). This structure is called "control canonical form" because it is so useful in control law design. Referring to that figure and taking the states as the outputs of the delay elements, numbered from the left, we get assuming $b_0 = 0$ for this case)

$$\mathbf{\Phi}_c = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad \mathbf{\Gamma}_c = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \qquad \mathbf{H}_c = [b_1 \quad b_2 \quad b_3]. \quad (6.11)$$

Note that from (2.15), the characteristic polynomial of this system is $a(z) = z^3 - a_1 z^2 - a_2 z - a_3$. The key idea here is that the elements of the first row of $\mathbf{\Phi}_c$ are exactly the coefficients of the characteristic polynomial of the system. If we now form the closed-loop system matrix $\mathbf{\Phi}_c - \mathbf{\Gamma}_c\mathbf{K}$, we find

$$\mathbf{\Phi}_c - \mathbf{\Gamma}_c\mathbf{K} = \begin{bmatrix} -a_1 - K_1 & -a_2 - K_2 & -a_3 - K_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \qquad (6.12)$$

By inspection, we find that the characteristic equation of (6.12) is

$$z^3 + (a_1 + K_1)z^2 + (a_2 + K_2)z + (a_3 + K_3) = 0.$$

Thus, if the desired root locations result in the characteristic equation

$$z^3 + \alpha_1 z^2 + \alpha_2 z + \alpha_3 = 0,$$

then the necessary values for control gains are

$$K_1 = \alpha_1 - a_1, \qquad K_2 = \alpha_2 - a_2, \qquad K_3 = \alpha_3 - a_3. \tag{6.13}$$

Conceptually, then, we have the canonical-form design method: Given an arbitrary $(\mathbf{\Phi}, \mathbf{\Gamma})$ and a desired characteristic equation $\alpha(z) = 0$, we convert (by redefinition of the states) $(\mathbf{\Phi}, \mathbf{\Gamma})$ to control form $(\mathbf{\Phi}_c, \mathbf{\Gamma}_c)$ and solve for the gain by (6.13). Because this gain is for states in the control form, we must, finally, express the result back in terms of the original states. This method is sometimes used by CAD packages because of the numerical advantages; however, the transformation is transparent to the designer, who generally prefers to use a state definition that is related to the physical system's characteristics.

6.2.2 Controllability

The first question this process raises is existence: Is it always possible to find an equivalent $(\mathbf{\Phi}_c, \mathbf{\Gamma}_c)$ for arbitrary $(\mathbf{\Phi}, \mathbf{\Gamma})$? The answer is almost always "yes." The exception occurs in certain pathological systems, dubbed "uncontrollable," for which no control will give arbitrary root locations. These systems have certain modes or subsystems that are unaffected by the control. Uncontrollability is best exhibited by a realization (selection of states) where each state represents a natural mode of the system. If all the roots of the open-loop characteristic equation,

$$\det|\, z\mathbf{I} - \mathbf{\Phi} \,| = 0$$

are distinct, then (6.3) written in this way (normal mode or "Jordan canonical form") becomes

$$\mathbf{x}(k+1) = \begin{bmatrix} \lambda_1 & & \bigcirc \\ & \lambda_2 & \\ \bigcirc & & \ddots \\ & & & \lambda_n \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} \mathbf{\Gamma}_1 \\ \mathbf{\Gamma}_2 \\ \vdots \\ \mathbf{\Gamma}_n \end{bmatrix} u(k) \tag{6.14}$$

and explicitly exhibits the criterion for controllability: no element in $\mathbf{\Gamma}$ can be zero. If any $\mathbf{\Gamma}$ element were zero, no control would influence that normal mode directly, and the associated state would remain uncontrolled. A good physical understanding of the system being controlled usually prevents any attempt to design a controller for an uncontrollable system; however, there is a mathematical test for controllability applicable to any system description, which may be an additional aid in discovering this condition; a discussion of this test is contained in Section 6.8.

6.2.3 Ackermann's Formula

The second question, if the system is found to be controllable and a gain is known to exist, is that of computational complexity. The process described above of converting to $(\mathbf{\Phi}_c, \mathbf{\Gamma}_c)$ needs to be organized to make the design easy to use. A very convenient formula has been derived by Ackermann (1972), and the proof for it is repeated in the Appendix to Chapter 6. The relation is:[3]

$$\boxed{\mathbf{K} = [0 \ldots 0 \quad 1][\mathbf{\Gamma} \quad \mathbf{\Phi\Gamma} \quad \mathbf{\Phi}^2\mathbf{\Gamma} \ldots \mathbf{\Phi}^{n-1}\mathbf{\Gamma}]^{-1}\alpha_c(\mathbf{\Phi}),} \tag{6.15}$$

where $\mathcal{C} = [\mathbf{\Gamma} \quad \mathbf{\Phi\Gamma} \ldots]$ is called the controllability matrix, n is the order of the system or number of state elements, and we substitute $\mathbf{\Phi}$ for z in $\alpha_c(z)$ to form

$$\alpha_c(\mathbf{\Phi}) = \mathbf{\Phi}^n + \alpha_1\mathbf{\Phi}^{n-1} + \alpha_2\mathbf{\Phi}^{n-2} + \cdots + \alpha_n\mathbf{I}, \tag{6.16}$$

where the α_i's are the coefficients of the desired characteristic equation, that is,

$$\alpha_c(z) = |\, z\mathbf{I} - \mathbf{\Phi} + \mathbf{\Gamma K}\,| = z^n + \alpha_1 z^{n-1} + \cdots + \alpha_n. \tag{6.17}$$

Note that if the z were replaced with s in (6.17), $\mathbf{\Phi}$ with $\mathbf{F}$, and $\mathbf{\Gamma}$ with $\mathbf{G}$, the statement of the continuous pole-placement problem would result. Therefore, we see that Ackermann's formula can be used for either the discrete or the continuous case.

[3] We note that the matrix $\mathcal{C}$ in (6.15) may be poorly conditioned and should not be inverted, but rather the equations $b^T\mathcal{C} = e^T$ should be solved by a stable method such as Gaussian elimination with pivoting. Also we note that careful selection of the state variables and their amplitude scaling will help avoid trouble in computing $\mathbf{K}$. For MIMO systems a much better algorithm numerically is described in Kautsky, Nichols, and Van Dooren (1985).

Example 6.2: Applying Ackermann's formula to the satellite attitude-control system of Example 6.1, we find from (6.9) that

$$\alpha_1 = -1.6, \qquad \alpha_2 = +0.70,$$

and therefore

$$\alpha_c(\boldsymbol{\Phi}) = \begin{bmatrix} 1 & 2T \\ 0 & 1 \end{bmatrix} - 1.6 \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} + 0.70 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.4T \\ 0 & 0.1 \end{bmatrix}.$$

Furthermore, we find that

$$[\boldsymbol{\Gamma} \;\; \boldsymbol{\Phi\Gamma}] = \begin{bmatrix} T^2/2 & 3T^2/2 \\ T & T \end{bmatrix}$$

and

$$[\boldsymbol{\Gamma} \;\; \boldsymbol{\Phi\Gamma}]^{-1} = 1/T^2 \begin{bmatrix} -1 & +3T/2 \\ 1 & -T/2 \end{bmatrix},$$

and finally

$$\mathbf{K} = [K_1 \;\; K_2] = (1/T^2)[0 \;\; 1] \begin{bmatrix} -1 & 3T/2 \\ 1 & -T/2 \end{bmatrix} \begin{bmatrix} 0.1 & 0.4T \\ 0 & 0.1 \end{bmatrix};$$

therefore

$$\begin{aligned} [K_1 \;\; K_2] &= \frac{1}{T^2}[0.1 \;\; 0.35T] \\ &= [10 \quad 3.5], \end{aligned}$$

which is the same result as that obtained earlier.

A program logic for application of Ackermann's formula to compute the control law is given in Fig. 6.2. Most CAD packages will contain this algorithm or its equivalent (see PLACE in Table E.1).

Example 6.3: A more complex system will demonstrate some of the difficulties with the pole-placement concept. Appendix A.4 describes a double mass–spring system that, if d is the measurement, is generic of many systems where there is some flexibility between the measured output and control input. We will use this system where

```
1. Read in $\mathbf{\Phi}, \mathbf{\Gamma}, T$, and $N_s$, the number of states.
2. Comment: First we will read in the desired pole locations in the
   s-plane convert them to z-plane polynomial coefficients, and construct
   $\alpha(\mathbf{\Phi})$.
3. $\mathbf{I} \leftarrow$ identity matrix, $N_s \times N_s$
4. $\mathbf{ALPHA} \leftarrow \mathbf{I}$
5. $k \leftarrow 1$
6. If $k > N_s$, go to step 18.
7. Read in pole location $k$ as $a + jb$.
8. If $b = 0$, go to step 14.
9. $A_1 \leftarrow -2\ \exp(aT)\ \cos bT$
10. $A_2 \leftarrow \exp(2aT)$
11. $\mathbf{ALPHA} \leftarrow \mathbf{ALPHA} \times (\mathbf{\Phi} \times \mathbf{\Phi} + A_1\mathbf{\Phi} + A_2\mathbf{I})$
12. $k \leftarrow k + 2$
13. Go to step 6.
14. $A_1 \leftarrow \exp(aT)$
15. $\mathbf{ALPHA} \leftarrow \mathbf{ALPHA} \times (\mathbf{\Phi} - \mathbf{A}_1 \times \mathbf{I})$
16. $k \leftarrow k + 1$
17. Go to step 6.
18. Comment: Now we construct the controllability matrix.
19. $\mathbf{C} \leftarrow \mathbf{I}$
20. $\mathbf{E} \leftarrow \mathbf{\Gamma}$
21. $k \leftarrow 1$
22. If $k > N_s$, go to step 28.
23. Comment: Replace column $k$ of $\mathbf{C}$ by $\mathbf{E}$.
24. $\mathbf{C}[\ \ ;k] \leftarrow \mathbf{E}$
25. $k \leftarrow k + 1$
26. $\mathbf{E} \leftarrow \mathbf{\Phi} \times \mathbf{E}$
27. Go to step 22.
28. Comment: Now solve for the control law; first form $e_n^T$ as the last row
    of $\mathbf{I}$.
29. $\mathbf{E} \leftarrow \mathbf{I}[N_s;\ \ ]$
30. Solve $\mathbf{BC} = \mathbf{E}$ for $\mathbf{B}$.
31. $\mathbf{K} = \mathbf{B} \times \mathbf{ALPHA}$
32. END
```

Figure 6.2 Program logic for computing control law **K** via Ackermann's formula.

the resonant mode has a frequency $\omega_n = 1$ rad/sec and damping $\zeta = 0.02$. We also will select a 10:1 ratio of the two masses. The parameters that provide these characteristics are: $M = 1$ kg, $m = 0.1$ kg, $b = 0.0036$ N-sec/m, and $k = 0.091$ N/m. Thus, from (A.17) we can write the state-space description as

$$\mathbf{x} \triangleq [d \quad \dot{d} \quad y \quad \dot{y}]^T, \qquad \mathbf{H} = [1 \quad 0 \quad 0 \quad 0],$$

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -0.91 & -0.036 & 0.91 & 0.036 \\ 0 & 0 & 0 & 1 \\ 0.091 & 0.0036 & -0.091 & -0.0036 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

The use of a CAD package (see X-C2D in Table E.1.) allows us to convert this $\mathbf{F}, \mathbf{G}$ to the discrete ZOH model, $\mathbf{\Phi}, \mathbf{\Gamma}$, which can then be used in the Ackermann pole-placement algorithm (see PLACE in Table E.1.) to arrive at the required $\mathbf{K}$. But first we need to pick some roots and a sample period. Because the resonance is at $\omega_n = 1$ rad/sec, let us select the sample time to be about 15 times faster, that is, $\omega_s = 15$ rad/sec. This translates to $T = 0.4$ sec. Picking roots for this system is more difficult than for the system of Example 6.1 because there are more of them. One possibility is for *all* the desired root locations to be at $z = 0.9$. This results in the feedback gain

$$\mathbf{K} = [0.650 \quad -0.651 \quad -0.645 \quad 0.718], \tag{6.18}$$

which produces the response to an initial condition, $d = 1$ m, shown in Fig. 6.3(a). It exhibits a response that is *much* larger than that of the initial condition, but the time characteristics are consistent with the selected roots.

In our next root selection, we will ask for only a modest increase in the damping of the resonant mode. This approach is suggested by an optimal control-design method that is the subject of Chapter 9.[4] But even without the benefits of an optimal design, it makes sense to limit our enthusiasm for increasing the damping of the resonant mode because that large a change in the natural dynamics will clearly require a large part of the control effort and possibly be the reason for the poor control of the d state element. So let's try $z = 0.9 \pm$

[4] See the optimal pole placement in Example 9.5.

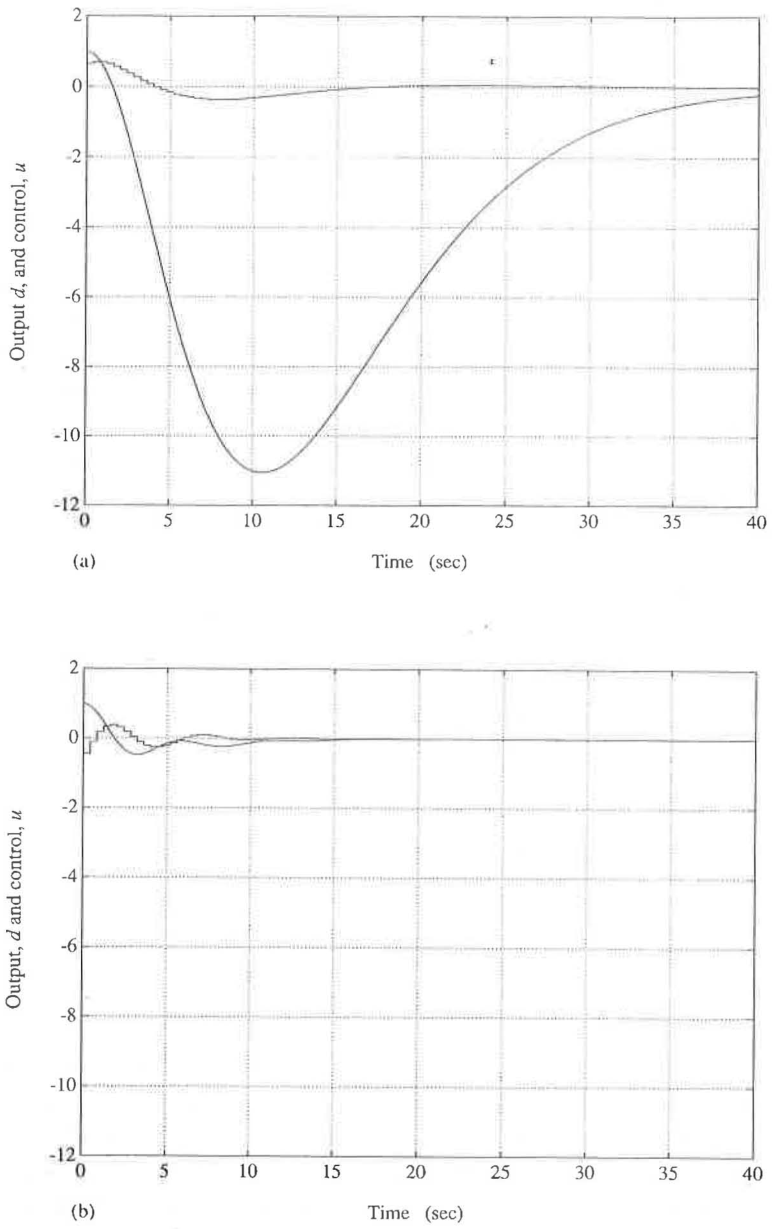

Figure 6.3 Initial condition response for Example 6.3; (a) desired roots all at $z = 0.9$ and **K** from (6.18), and (b) desired roots at $z = 0.9 \pm j0.05, 0.8 \pm j0.4$, and **K** from (6.19).

$j0.05, 0.8 \pm j0.4$. This results in the feedback gain

$$\mathbf{K} = [-0.458 \quad -0.249 \quad 0.568 \quad 0.968], \tag{6.19}$$

which produces the response to an initial condition, $d = 1$ m, shown in Fig. 6.3(b). It exhibits *much* less response of d with no increase in control effort, although the resonant mode oscillations did influence the response with a damping consistent with the roots selected. (The selected closed-loop roots had a $\zeta \cong 0.2$, a factor of 10 better than the open-loop roots but still visible on the output.) The control is clearly much more effective with the latter choice of roots.

So we see that the mechanics of computing the control law is easy, once the desired root locations are known. The trick is to pick a good set of roots! The designer would have to iterate between root selections and some other system evaluation to determine when the design is complete. System evaluation might consist of an initial-condition time response as shown in the example, a step response, steady-state errors, gain and phase margins, or the entire frequency-response shape. Pole placement by itself leaves something to be desired. But it is useful as a design tool to be used in conjunction with the other design methods discussed in Chapter 5 or as a part of an optimal design process that will be discussed in Chapter 9.

6.3 ESTIMATOR DESIGN

The control law designed in the last section assumed that all state elements were available for feedback. Because typically, not all elements are measured, the missing portion of the state needs to be reconstructed for use in the control law. We will first discuss methods to obtain an estimate of the entire state given a measurement of one of the state elements. This will provide the missing elements as well as providing a smoothed value of the measurement, which is often contaminated with random errors or "noise." There are two basic kinds of estimates of the state, $\mathbf{x}(k)$: We call it the *current* estimate, $\hat{\mathbf{x}}(k)$, if based on measurements $y(k)$ up to and including the kth instant; and we call it the *predictor* estimate, $\bar{\mathbf{x}}(k)$, if based on measurements up to $y(k-1)$. The idea eventually will be to let $u = -\mathbf{K}\hat{\mathbf{x}}$ or $u = -\mathbf{K}\bar{\mathbf{x}}$, replacing the true state used in (6.5) by its estimate.

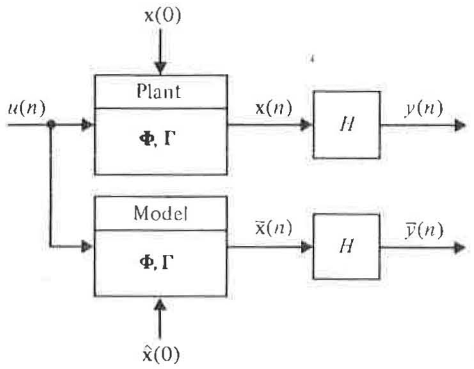

Figure 6.4 Open-loop estimator.

6.3.1 Prediction Estimators

One method of estimating the state which might come to mind is to construct a model of the plant dynamics,

$$\bar{\mathbf{x}}(k+1) = \mathbf{\Phi}\bar{\mathbf{x}}(k) + \mathbf{\Gamma}u(k). \tag{6.20}$$

We know $\mathbf{\Phi}, \mathbf{\Gamma}$, and $u(k)$, and hence this estimator should work if we can obtain the correct $\mathbf{x}(0)$ and set $\bar{\mathbf{x}}(0)$ equal to it. Figure 6.4 depicts this "open-loop" estimator. If we define the error in the estimate as

$$\tilde{\mathbf{x}} \overset{\triangle}{=} \bar{\mathbf{x}} - \mathbf{x}. \tag{6.21}$$

and substitute (6.3) and (6.20) into (6.21), we find that the dynamics of the resulting system are described by the estimator-error equation

$$\tilde{\mathbf{x}}(k+1) = \mathbf{\Phi}\tilde{\mathbf{x}}(k). \tag{6.22}$$

Thus, if the initial value of $\bar{\mathbf{x}}$ is off, the dynamics of the estimate error are those of the uncompensated plant, $\mathbf{\Phi}$. For a marginally stable or unstable plant, the error will never decrease from the initial value. For an asymptotically stable plant, an initial error will decrease only because the plant and estimate will both approach zero. Basically, the estimator is running open loop and not utilizing any continuing measurements of the system's behavior, and we would expect that it would diverge from the truth. However, if we feed back the difference between the measured output and the estimated output and constantly correct the model with this error signal, the divergence should be minimized. The idea is to construct a feedback system