# 5

# Control System Design in State-Space

## 5.1 Design: Classical vs. Modern

A fashion designer tailors the apparel to meet the tastes of fashionable people, keeping in mind the desired fitting, season and the occasion for which the clothes are to be worn. Similarly, a control system engineer *designs* a control system to meet the desired objectives, keeping in mind issues such as where and how the control system is to be implemented. We need a control system because we do not like the way a plant behaves, and by *designing* a control system we try to *modify* the behavior of the plant to suit our needs. *Design* refers to the process of changing a control system's *parameters* to meet the specified stability, performance, and robustness objectives. The design *parameters* can be the unknown constants in a *controller's* transfer function, or its state-space representation. In Chapter 1 we compared open- and closed-loop control systems, and saw how a closed-loop control system has a better chance of achieving the desired performance. In Chapter 2 we saw how the classical transfer function approach can be used to design a closed-loop control system, i.e. the use of graphical methods such as Bode, Nyquist, and root-locus plots. Generally, the classical design consists of varying the controller transfer function until a desired closed-loop performance is achieved. The classical indicators of the closed-loop performance are the closed-loop frequency response, or the locations of the closed-loop poles. For a large order system, by varying a limited number of constants in the controller transfer function, we can vary in a pre-specified manner the locations of only a few of the closed-loop poles, *but not all of them*. This is a major limitation of the classical design approach. The following example illustrates some of the limitations of the classical design method.

### *Example 5.1*

Let us try to design a closed-loop control system for the following plant transfer function in order to achieve a zero steady-state error when the desired output is the unit step function, $u_s(t)$:

$$G(s) = (s+1)/[(s-1)(s+2)(s+3)] \tag{5.1}$$

The single-input, single-output plant, $G(s)$, has poles located at $s = 1$, $s = -2$, and $s = -3$. Clearly, the plant is unstable due to a pole, $s = 1$, in the right-half s-plane. Also, the plant is of type 0. For achieving a zero steady-state error, we need to do *two things*: make the closed-loop system stable, and make type of the closed-loop system at least unity. Selecting a closed-loop arrangement of Figure 2.32, both of these requirements are apparently met by the following choice of the controller transfer function, $H(s)$:

$$H(s) = K(s-1)/s \tag{5.2}$$

Such a controller would apparently cancel the plant's unstable pole at $s = 1$ by a zero at the same location in the closed-loop transfer function, and make the system of type 1 by having a pole at $s = 0$ in the open-loop transfer function. The open-loop transfer function, $G(s)H(s)$, is then the following:

$$G(s)H(s) = K(s+1)/[s(s+2)(s+3)] \tag{5.3}$$

and the closed-loop transfer function is given by

$$\begin{aligned} Y(s)/Y_d(s) &= G(s)H(s)/[1 + G(s)H(s)] \\ &= K(s+1)/[s(s+2)(s+3) + K(s+1)] \end{aligned} \tag{5.4}$$

From Eq. (5.4), it is apparent that the closed-loop system can be made stable by selecting those value of the design parameter, $K$, such that all the closed-loop poles lie in the left-half s-plane. The root-locus of the closed-loop system is plotted in Figure 5.1 as $K$

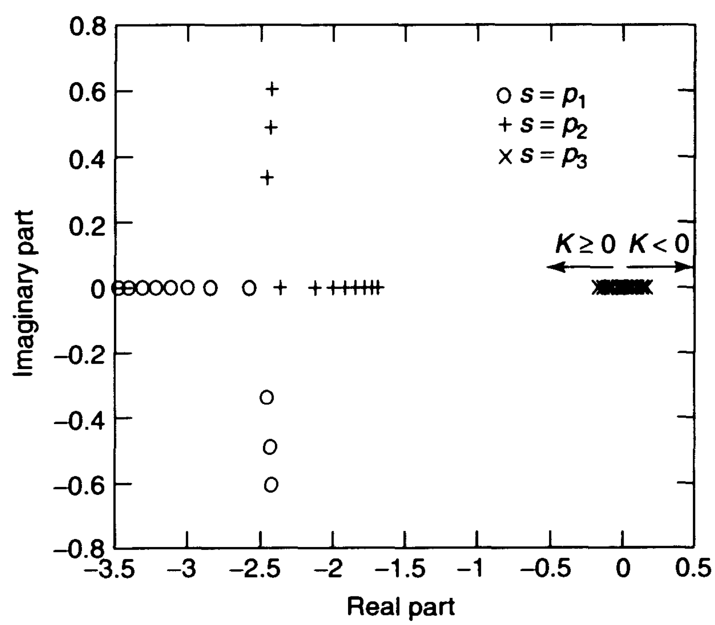


**Figure 5.1** Apparent loci of the closed-loop poles of Example 5.1 as the classical design parameter, $K$, is varied from −1 to 1

is varied from $-1$ to 1. Apparently, from Figure 5.1, the closed-loop system is stable for $K \geq 0$ and unstable for $K < 0$. In Figure 5.1, it appears that the pole $s = p_3$ determines the stability of the closed-loop system, since it is the pole which crosses into the right-half s-plane for $K < 0$. This pole is called the *dominant pole* of the system, because being closest to the imaginary axis it *dominates* the system's response (recall from Chapter 4 that the smaller the real part magnitude of a pole, the longer its contribution to the system's response persists). By choosing an appropriate value of $K$, we can place only the dominant pole, $s = p_3$, at a desired location in the left-half s-plane. The locations of the other two poles, $s = p_1$ and $s = p_2$, would then be governed by such a choice of $K$. In other words, by choosing the sole parameter $K$, the locations of all the three poles cannot be chosen *independently of each other*. Since all the poles contribute to the closed-loop performance, the classical design approach may fail to achieve the desired performance objectives when only a few poles are being directly affected in the design process.

Furthermore, the chosen design approach of Example 5.1 is misleading, because it *fails to even stabilize* the closed-loop system! Note that the closed-loop transfer function given by Eq. (5.4) is of *third order*, whereas we expect that it should be of *fourth order*, because the closed-loop system is obtained by combining a third order plant with a first order controller. This discrepancy in the closed-loop transfer function's order has happened due to our attempt to *cancel* a pole with a zero at the same location. Such an attempt is, however, doomed to fail as shown by a state-space analysis of the closed-loop system.

### *Example 5.2*

Let us find a state-space representation of the closed-loop system designed using the classical approach in Example 5.1. Since the closed-loop system is of the configuration shown in Figure 3.7(c), we can readily obtain its state-space representation using the methods of Chapter 3. The Jordan canonical form of the plant, $G(s)$, is given by the following state coefficient matrices:

$$\mathbf{A_p} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -3 \end{bmatrix}; \quad \mathbf{B_p} = \begin{bmatrix} 0 \\ 1/3 \\ -1/2 \end{bmatrix}$$

$$\mathbf{C_p} = [1 \quad 1 \quad 1]; \quad \mathbf{D_p} = 0 \tag{5.5}$$

A state-space representation of the controller, $H(s)$, is the following:

$$\mathbf{A_c} = 0; \quad \mathbf{B_c} = K; \quad \mathbf{C_c} = -1; \quad \mathbf{D_c} = K \tag{5.6}$$

Therefore, on substituting Eqs. (5.5) and (5.6) into Eqs. (3.146)–(3.148), we get the following state-space representation of the closed-loop system:

$$\mathbf{x}^{(1)}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{y_d}(t) \tag{5.7}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{y_d}(t) \tag{5.8}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -K/3 & (-2-K/3) & -K/3 & -1/3 \\ K/2 & K/2 & (-3+K/2) & 1/2 \\ -K & -K & -K & 0 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 0 \\ K/3 \\ -K/2 \\ K \end{bmatrix}$$

$$\mathbf{C} = [1 \quad 1 \quad 1 \quad 0]; \quad \mathbf{D} = 0 \tag{5.9}$$

The closed-loop system is of fourth order, as expected. The closed-loop poles are the eigenvalues of **A**, i.e. the solutions of the following characteristic equation:

$$|\lambda\mathbf{I} - \mathbf{A}| = \begin{vmatrix} (\lambda-1) & 0 & 0 & 0 \\ K/3 & (\lambda+2+K/3) & K/3 & 1/3 \\ -K/2 & -K/2 & (\lambda+3-K/2) & -1/2 \\ K & K & K & \lambda \end{vmatrix} = 0 \tag{5.10}$$

It is evident from Eq. (5.10) that, irrespective of the value of $K$, one of the eigenvalues of **A** is $\lambda = 1$, which corresponds to a closed-loop pole at $s = 1$. Hence, irrespective of the design parameter, $K$, we have an *unstable* closed-loop system, which means that the chosen design approach of cancelling an unstable pole with a zero *does not work*. More importantly, even though we have an *unconditionally* unstable closed-loop system, the closed-loop transfer function given by Eq. (5.4) *fools us into believing* that we can stabilize the closed-loop system by selecting an *appropriate value* for $K$. Such a system which remains unstable irrespective of the values of the control design parameters is called an *unstabilizable system*. The classical design approach of Example 5.1 gave us an unstabilizable closed-loop system, and we didn't even know it! Stabilizability of a system is a consequence of an important property known as *controllability*, which we will consider next. (Although we considered a closed-loop system in Example 5.2, the properties *controllability* and *stabilizability* are more appropriately defined for a plant.)

## 5.2 Controllability

When as children we sat in the back seat of a car, our collective effort to move the car by pushing on the front seat always ended in failure. This was because the input we provided to the car in this manner, *no matter how large*, did not affect the overall motion of the car. There was something known as the *third law of Newton*, which physically prevented us from achieving our goal. Hence, for us the car was *uncontrollable when we were sitting in the car*. The same car could be moved, however, by stepping out and giving a hefty push to it from the outside; then it became a *controllable* system for our purposes. *Controllability* can be defined as the property of a system when it is possible to take the system from *any initial state*, $\mathbf{x}(t_0)$, to *any final state*, $\mathbf{x}(t_f)$, in a *finite* time, $(t_f - t_0)$, by means of the input vector, $\mathbf{u}(t)$, $t_0 \leq t \leq t_f$. It is important to stress the words *any* and *finite*, because it may be possible to move an uncontrollable system from *some initial* *states* to *some final states*, or take an *infinite* amount of time in moving the uncontrollable system, using the input vector, $\mathbf{u}(t)$. Controllability of a system can be easily determined if we can *decouple* the state-equations of a system. Each decoupled scalar state-equation corresponds to a *sub-system.* If any of the decoupled state-equations of the system is *unaffected* by the input vector, then it is not possible to change the corresponding state variable using the input, and hence, the sub-system is *uncontrollable.* If any sub-system is uncontrollable, i.e. if any of the state variables is unaffected by the input vector, then it follows that the entire system is uncontrollable.

### *Example 5.3*

Re-consider the closed-loop system of Example 5.2. The state-equations of the closed-loop system (Eqs. (5.7)–(5.9)) can be expressed in scalar form as follows:

$$x_1^{(1)}(t) = x_1(t) \tag{5.11a}$$

$$x_2^{(1)}(t) = -Kx_1(t)/3 - (2+K/3)x_2(t) - Kx_3(t)/3 - x_4(t)/3 + Ky_d(t)/3 \tag{5.11b}$$

$$x_3^{(1)}(t) = Kx_1(t)/2 + Kx_2(t)/2 + (-3+K/2)x_3(t) + x_4(t)/2 - Ky_d(t)/2 \tag{5.11c}$$

$$x_4^{(1)}(t) = -Kx_1(t) - Kx_2(t) - Kx_3(t) + Ky_d(t) \tag{5.11d}$$

On examining Eq. (5.11a), we find that the equation is decoupled from the other state-equations, and does not contain the input to the closed-loop system, $y_d(t)$. Hence, the state variable, $x_1(t)$, is entirely unaffected by the input, $y_d(t)$, which implies that the system is *uncontrollable.* Since the uncontrollable sub-system described by Eq. (5.11a) is also unstable (it corresponds to the eigenvalue $\lambda = 1$), there is no way we can stabilize the closed-loop system by changing the controller design parameter, $K$. Hence, the system is *unstabilizable.* In fact, the plant of this system given by the state-space representation of Eq. (5.5) is itself unstabilizable, because of the zero in the matrix $\mathbf{B_p}$ corresponding to the sub-system having eigenvalue $\lambda = 1$. The unstabilizable plant leads to an unstabilizable closed-loop system.

Example 5.3 shows how a decoupled state-equation indicating an *uncontrollable* and *unstable* sub-system implies an *unstabilizable* system.

### *Example 5.4*

Let us analyze the controllability of the following system:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} \tag{5.12}$$

The system is *unstable*, with four zero eigenvalues. Since the state-equations of the system are coupled, we cannot directly deduce controllability. However, some of the state-equations can be decoupled by transforming the state-equations using the transformation $\mathbf{z}(t) = \mathbf{T}\mathbf{x}(t)$, where

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \tag{5.13}$$

The transformed state-equations can be written in the following scalar form:

$$z_1^{(1)}(t) = x_3'(t) \tag{5.14a}$$

$$z_2^{(1)}(t) = x_4'(t) \tag{5.14b}$$

$$z_3^{(1)}(t) = 0 \tag{5.14c}$$

$$z_4^{(1)}(t) = -2u(t) \tag{5.14d}$$

Note that the state-equation, Eq. (5.14c) denotes an uncontrollable sub-system in which the state variable, $z_3(t)$, is unaffected by the input, $u(t)$. Hence, the system is uncontrollable. However, since the only uncontrollable sub-system denoted by Eq. (5.14c) is *stable* (its eigenvalue is, $\lambda = 0$), we can safely *ignore* this sub-system and *stabilize* the remaining sub-systems denoted by Eqs. (5.14a),(5.14b), and (5.14d), using a feedback controller that modifies the control input, $u(t)$. An uncontrollable system all of whose uncontrollable sub-systems are stable is thus said to be *stabilizable*. The process of stabilizing a stabilizable system consists of ignoring all uncontrollable but stable sub-systems, and designing a controller based on the remaining (controllable) sub-systems. Such a control system will be successful, because each ignored sub-system will be stable.

In the previous two examples, we could determine controllability, only because certain state-equations were decoupled from the other state-equations. Since decoupling state-equations is a cumbersome process, and may not be always possible, we need another criterion for testing whether a system is controllable. The following *algebraic controllability test theorem* provides an easy way to check for controllability.

**Theorem**

*A linear, time-invariant system described by the matrix state-equation,* $\mathbf{x}^{(1)}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$ *is* controllable *if and only if the* controllability test matrix

$$\mathbf{P} = [\mathbf{B};\ \ \mathbf{AB};\ \ \mathbf{A}^2\mathbf{B};\ \ \mathbf{A}^3\mathbf{B};\ \ \ldots;\ \ \mathbf{A}^{n-1}\mathbf{B}]$$

*is of* rank $n$*, the order of the system.*

(The *rank* of a matrix, $\mathbf{P}$, is defined as the dimension of the *largest non-zero* determinant formed out of the matrix, $\mathbf{P}$ (see Appendix B). If $\mathbf{P}$ is a square matrix, the largest determinant formed out of **P** is $|\mathbf{P}|$. If **P** is not a square matrix, the largest determinant formed out of **P** is either the determinant formed by taking all the rows and equal number of columns, or all the columns and equal number of rows of **P**. See Appendix B for an illustration of the rank of a matrix. Note that for a system of order $n$ with $r$ inputs, the size of the controllability test matrix, **P**, is $(n \times nr)$. The largest non-zero determinant of **P** can be of dimension $n$. Hence, the rank of **P** can be *either less than or equal* to $n$.)

A rigourous proof of the algebraic controllability test theorem can be found in Friedland [2]. An analogous form of algebraic controllability test theorem can be obtained for linear, time-varying systems [2]. Alternatively, we can form a *time-varying* controllability test matrix as

$$\mathbf{P}(t) = [\mathbf{B}(t); \quad \mathbf{A}(t)\mathbf{B}(t); \quad \mathbf{A}^2(t)\mathbf{B}(t); \quad \mathbf{A}^3(t)\mathbf{B}(t); \quad \ldots; \quad \mathbf{A}^{n-1}(t)\mathbf{B}(t)] \tag{5.15}$$

and check the rank of $\mathbf{P}(t)$ for all times, $t \geq t_0$, for a linear, time-varying system. If at any instant, $t$, the rank of $\mathbf{P}(t)$ is less than $n$, the system is uncontrollable. However, we must use the time-varying controllability test matrix of Eq. (5.15) with great *caution*, when the state-coefficient matrices are rapidly changing with time, because the test can be practically applied at *discrete time step* – rather than at all possible times (see Chapter 4) – and there may be some time intervals (smaller than the time steps) in which the system may be uncontrollable.

### *Example 5.5*

Using the controllability test theorem, let us find whether the following system is controllable:

$$\mathbf{A} = \begin{bmatrix} -2 & 1 \\ -1 & -3 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{5.16}$$

The controllability test matrix is the following:

$$\mathbf{P} = [\mathbf{B}; \quad \mathbf{AB}] = \begin{bmatrix} 1 & -2 \\ 0 & -1 \end{bmatrix} \tag{5.17}$$

The largest determinant of **P** is $|\mathbf{P}| = -1 \neq 0$, Hence the rank of **P** is equal to 2, the order of the system. Thus, by the controllability test theorem, the system is *controllable*.

Applying the algebraic controllability test involves finding the rank of **P**, and checking whether it is equal to $n$. This involves forming all possible determinants of dimension $n$ out of the matrix **P**, by removing some of the columns (if $m > 1$), and checking whether all of those determinants are non-zero. By any account, such a process is cumbersome if performed by hand. However, MATLAB provides us the command *rank(P)* for finding the rank of a matrix, **P**. Moreover, MATLAB's Control System Toolbox (CST) lets you directly form the controllability test matrix, **P**, using the command *ctrb* as follows:

```
>>P = ctrb(A, B) <enter>
```

or

```
>>P = ctrb(sys) <enter>
```

where **A** and **B** are the state coefficient matrices of the system whose LTI object is *sys*.

### *Example 5.6*

Let us verify the uncontrollability of the system given in Example 5.4, using the controllability test. The controllability test matrix is constructed as follows:

```
>>A=[0 0 1 0; zeros(1,3)1; zeros(2,4)]; B=[0 0 -1 1]'; P=ctrb(A,B)
  <enter>

P =

    0  -1   0   0
    0   1   0   0
   -1   0   0   0
    1   0   0   0
```

Then the rank of **P** is found using the MATLAB command *rank:*

```
>>rank(P) <enter>

ans =

  2
```

Since the rank of **P** is *less than* 4, the order of the system, it follows from the controllability test theorem that the system is *uncontrollable*.

What are the causes of uncontrollability? As our childhood attempt of pushing a car while sitting inside it indicates, whenever we choose an input vector that does not affect *all* the state variables *physically*, we will have an uncontrollable system. An attempt to cancel a pole of the plant by a zero of the controller may also lead to an uncontrollable closed-loop system *even though the plant itself may be controllable*. Whenever you see a system in which pole-zero cancellations have occurred, the chances are high that such a system is uncontrollable.

### *Example 5.7*

Let us analyze the controllability of the closed-loop system of configuration shown in Figure 2.32, in which the controller, $H(s)$, and plant, $G(s)$, are as follows:

$$H(s) = K(s-2)/(s+1); \quad G(s) = 3/(s-2) \tag{5.18}$$

The closed-loop transfer function in which a pole-zero cancellation has occurred at $s = 2$ is the following:

$$Y(s)/Y_d(s) = G(s)H(s)/[1 + G(s)H(s)] = 3K/(s + 3K + 1) \quad (5.19)$$

The Jordan canonical form of the plant is the following:

$$\mathbf{A_p} = 2; \quad \mathbf{B_p} = 3; \quad \mathbf{C_p} = 1; \quad \mathbf{D_p} = 0 \quad (5.20)$$

Note that the plant is controllable (the controllability test matrix for the plant is just $\mathbf{P} = \mathbf{B_p}$, which is of rank 1). The Jordan canonical form of the controller is the following:

$$\mathbf{A_c} = -1; \quad \mathbf{B_c} = K; \quad \mathbf{C_c} = -3; \quad \mathbf{D_c} = K \quad (5.21)$$

The closed-loop state-space representation is obtained using Eqs. (3.146)–(3.148) as the following:

$$\mathbf{A} = \begin{bmatrix} (2-3K) & -9 \\ -K & -1 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 3K \\ K \end{bmatrix}$$
$$\mathbf{C} = [1 \quad 0]; \quad \mathbf{D} = 0 \quad (5.22)$$

The controllability test matrix for the closed-loop system is the following:

$$\mathbf{P} = [\mathbf{B} \quad \mathbf{AB}] = \begin{bmatrix} 3K & -(9K^2 + 3K) \\ K & -(3K^2 + K) \end{bmatrix} \quad (5.23)$$

To see whether $\mathbf{P}$ is of rank 2 (i.e. whether $\mathbf{P}$ is non-singular) let us find its determinant as follows:

$$|\mathbf{P}| = \begin{vmatrix} 3K & -(9K^2 + 3K) \\ K & -(3K^2 + K) \end{vmatrix} = -9K^3 - 3K^2 + 9K^3 + 3K^2 = 0 \quad (5.24)$$

Since $|\mathbf{P}| = 0$, $\mathbf{P}$ is singular, its rank is less than 2. Therefore, the closed-loop system is uncontrollable no matter what value of the controller design parameter, $K$, is chosen. Hence, a *controllable* plant has led to an *uncontrollable* closed-loop system in which a pole-zero cancellation has occurred.

Other causes of uncontrollability could be *mathematical*, such as using *superfluous* state variables (i.e. more state variables than the order of the system) when modeling a system; the superfluous state variables will be definitely unaffected by the inputs to the system, causing the state-space representation to be uncontrollable, even though the system may be physically controllable. A rare cause of uncontrollability is *too much symmetry* in the system's mathematical model. Electrical networks containing *perfectly balanced bridges* are examples of systems with too much symmetry. However, perfect symmetry almost never exists in the real world, or in its digital computer model.

Now that we know how to determine the controllability of a system, we can avoid the pitfalls of Examples 5.1 and 5.7, and are ready to design a control system using state-space methods.

## 5.3 Pole-Placement Design Using Full-State Feedback

In Section 5.1 we found that it may be required to change a plant's characteristics by using a closed-loop control system, in which a controller is designed to *place* the *closed-loop poles* at desired locations. Such a design technique is called the *pole-placement* approach. We also discussed in Section 5.1 that the classical design approach using a controller transfer function with a few design parameters is insufficient to place all the closed-loop poles at desired locations. The state-space approach using *full-state feedback* provides sufficient number of controller design parameters to move all the closed-loop poles independently of each other. *Full-state feedback* refers to a controller which generates the input vector, $\mathbf{u}(t)$, according to a *control-law* such as the following:

$$\mathbf{u}(t) = \mathbf{K}[\mathbf{x_d}(t) - \mathbf{x}(t)] - \mathbf{K_d}\mathbf{x_d}(t) - \mathbf{K_n}\mathbf{x_n}(t) \tag{5.25}$$

where $\mathbf{x}(t)$ is the state-vector of the plant, $\mathbf{x_d}(t)$ is the *desired* state-vector, $\mathbf{x_n}(t)$ is the *noise* state-vector and $\mathbf{K}$, $\mathbf{K_d}$ and $\mathbf{K_n}$ are the *controller gain matrices*. The desired state-vector, $\mathbf{x_d}(t)$, and the noise state-vector, $\mathbf{x_n}(t)$, are generated by external processes, and act as inputs to the control system. The task of the controller is to achieve the desired state-vector in the steady state, while counteracting the affect of the noise. The input vector, $\mathbf{u}(t)$, generated by Eq. (5.25) is applied to the plant described by the following state and output equations:

$$\mathbf{x}^{(1)}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{F}\mathbf{x_n}(t) \tag{5.26}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{E}\mathbf{x_n}(t) \tag{5.27}$$

where $\mathbf{F}$ and $\mathbf{E}$ are the noise coefficient matrices in the state and output equations, respectively. Designing a control system using full-state feedback requires that the plant described by Eq. (5.26) must be *controllable*, otherwise the control input generated using Eq. (5.25) will not affect all the state variables of the plant. Furthermore, Eq. (5.25) requires that the all the state variables of the system must be *measurable*, and capable of being fed back to the controller. The controller thus consists of physical *sensors*, which measure the state variables, and electrical or mechanical devices, called *actuators*, which provide inputs to the plant based on the desired outputs and the *control-law* of Eq. (5.25). Modern controllers invariably use digital electronic circuits to implement the control-law in a hardware. The controller gain matrices, $\mathbf{K}$, $\mathbf{K_d}$, and $\mathbf{K_n}$ are the *design parameters* of the control system described by Eqs. (5.25)–(5.27). Note that the order of the full-state feedback closed-loop system is the *same* as that of the plant. A schematic diagram of the general control system with full-state feedback is shown in Figure 5.2.

Let us first consider control systems having $\mathbf{x_d}(t) = \mathbf{0}$. A control system in which the desired state-vector is zero is called a *regulator*. Furthermore, for simplicity let us assume that all the measurements are *perfect*, and that there is *no error* committed in *modeling*

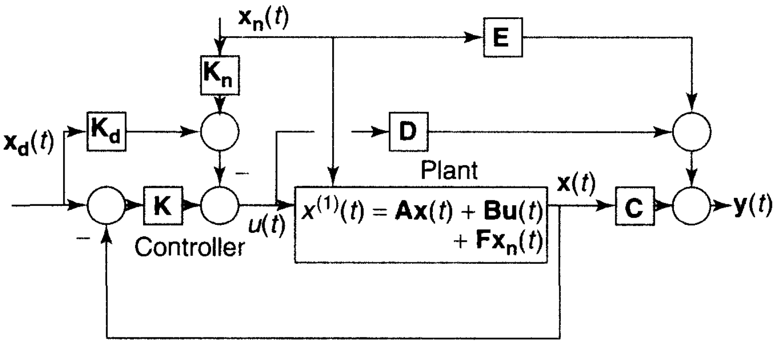


**Figure 5.2** Schematic diagram of a general full-state feedback control system with desired state, $\mathbf{x_d}(t)$, and noise, $\mathbf{x_n}(t)$

the plant by Eqs. (5.26) and (5.27). These two assumptions imply that all undesirable inputs to the system in the form of noise, are absent, i.e. $\mathbf{x}_n(t) = \mathbf{0}$. Consequently, the control-law of Eq. (5.25) reduces to

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t) \tag{5.28}$$

and the schematic diagram of a noiseless regulator is shown in Figure 5.3.

On substituting Eq. (5.28) into Eqs. (5.26) and (5.27), we get the closed-loop state and output equations of the regulator as follows:

$$\mathbf{x}^{(1)}(t) = (\mathbf{A} - \mathbf{BK})\mathbf{x}(t) \tag{5.29}$$

$$\mathbf{y}(t) = (\mathbf{C} - \mathbf{DK})\mathbf{x}(t) \tag{5.30}$$

Equations. (5.29) and (5.30) indicate that the regulator is a *homogeneous* system, described by the closed-loop state coefficient matrices $\mathbf{A_{CL}} = \mathbf{A} - \mathbf{BK}$, $\mathbf{B_{CL}} = \mathbf{0}$, $\mathbf{C_{CL}} = \mathbf{C} - \mathbf{DK}$, and $\mathbf{D_{CL}} = \mathbf{0}$. The closed-loop poles are the *eigenvalues* of $\mathbf{A_{CL}}$. Hence, by selecting the controller gain matrix, $\mathbf{K}$, we can place the closed-loop poles at desired locations. For a plant of order $n$ with $r$ inputs, the size of $\mathbf{K}$ is $(r \times n)$. Thus, we have a total of $r \cdot n$ scalar design parameters in our hand. For multi-input systems (i.e. $r > 1$), the number of design parameters are, therefore, *more than sufficient* for selecting the locations of $n$ poles.

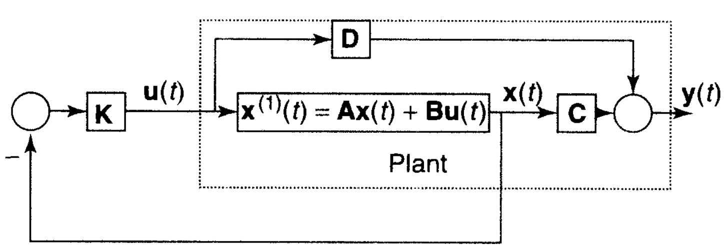


**Figure 5.3** Schematic diagram of a full-state feedback regulator (i.e. control system with a zero desired state-vector) without any noise

*Example 5.8*

Let us design a full-state feedback regulator for the following plant such that the closed-loop poles are $s = -0.5 \pm i$:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{5.31}$$

The plant, having poles at $s = 1$ and $s = -2$, is unstable. Also, the plant is controllable, because its decoupled state-space representation in Eq. (5.31) has no elements of $\mathbf{B}$ equal to zero. Hence, we can place closed-loop poles *at will* using the following full-state feedback gain matrix:

$$\mathbf{K} = [K_1; \quad K_2] \tag{5.32}$$

The closed-loop state-dynamics matrix, $\mathbf{A_{CL}} = \mathbf{A} - \mathbf{BK}$, is the following:

$$\mathbf{A_{CL}} = \mathbf{A} - \mathbf{BK} = \begin{bmatrix} (1 - K_1) & -K_2 \\ K_1 & (-2 + K_2) \end{bmatrix} \tag{5.33}$$

The closed-loop poles are the eigenvalues of $\mathbf{A_{CL}}$, which are calculated as follows:

$$|\lambda\mathbf{I} - \mathbf{A_{CL}}| = \begin{vmatrix} (\lambda - 1 + K_1) & K_2 \\ -K_1 & (\lambda + 2 - K_2) \end{vmatrix}$$
$$= (\lambda - 1 + K_1)(\lambda + 2 - K_2) + K_1 K_2 = 0 \tag{5.34}$$

The roots of the characteristic equation (Eq. (5.34)) are the closed-loop eigenvalues given by

$$\lambda_{1,2} = -0.5(K_1 - K_2 + 1) \pm 0.5(K_1^2 + K_2^2 - 2K_1K_2 - 6K_1 - 2K_2 + 9)^{1/2}$$
$$= -0.5 \pm i \tag{5.35}$$

Solving Eq. (5.35) for the unknown parameters, $K_1$ and $K_2$, we get

$$K_1 = K_2 = 13/12 \tag{5.36}$$

Thus, the full-state feedback regulator gain matrix which moves the poles from $s = 1$, $s = -2$ to $s = -0.5 \pm i$ is $\mathbf{K} = [13/12; \quad 13/12]$.

## 5.3.1 Pole-placement regulator design for single-input plants

Example 5.8 shows that even for a single-input, second order plant, the calculation for the required regulator gain matrix, **K**, by hand is rather involved, and is likely to get out of hand as the order of the plant increases beyond three. Luckily, if the plant is in the *controller companion form*, then such a calculation is greatly simplified for single-input plants. Consider a single-input plant of order $n$ whose controller companion form is the following (see Chapter 3):

$$\mathbf{A} = \begin{bmatrix} -a_{n-1} & -a_{n-2} & -a_{n-3} & \dots & -a_1 & -a_0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ . & . & . & \dots & . & . \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}; \quad \mathbf{B} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ . \\ . \\ 0 \\ 0 \end{bmatrix} \tag{5.37}$$

where $a_0, \dots, a_{n-1}$ are the coefficients of the plant's characteristic polynomial $|s\mathbf{I} - \mathbf{A}| = s^n + a_{n-1}s^{n-1} + \dots + a_1 s + a_0$. The full-state feedback regulator gain matrix is a row vector of $n$ unknown parameters given by

$$\mathbf{K} = [K_1; \quad K_2; \quad \dots; \quad K_n] \tag{5.38}$$

It is desired to place the closed-loop poles such that the closed-loop characteristic polynomial is the following:

$$|s\mathbf{I} - \mathbf{A}_{\mathbf{CL}}| = |s\mathbf{I} - \mathbf{A} + \mathbf{BK}| = s^n + \alpha_{n-1}s^{n-1} + \alpha_{n-2}s^{n-2} \dots + \alpha_1 s + \alpha_0 \tag{5.39}$$

where the closed-loop state dynamics matrix, $\mathbf{A}_{\mathbf{CL}} = \mathbf{A} - \mathbf{BK}$, is the following:

$$\mathbf{A}_{\mathbf{CL}} = \begin{bmatrix} (-a_{n-1} - K_1) & (-a_{n-2} - K_2) & (-a_{n-3} - K_3) & \dots & (-a_1 - K_{n-1}) & (-a_0 - K_n) \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ . & . & . & \dots & . & . \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \tag{5.40}$$

It is interesting to note that the closed-loop system is also in the controller companion form! Hence, from Eq. (5.40), the coefficients of the closed-loop characteristic polynomial must be the following:

$$\alpha_{n-1} = a_{n-1} + K_1; \quad \alpha_{n-2} = a_{n-2} + K_2; \quad \dots; \quad \alpha_1 = a_1 + K_{n-1}; \quad \alpha_0 = a_0 + K_n \tag{5.41}$$

or, the unknown regulator parameters are calculated simply as follows:

$$K_1 = \alpha_{n-1} - a_{n-1}; \quad K_2 = \alpha_{n-2} - a_{n-2}; \quad \dots; \quad K_{n-1} = \alpha_1 - a_1; \quad K_n = \alpha_0 - a_0 \tag{5.42}$$

In vector form, Eq. (5.42) can be expressed as

$$\mathbf{K} = \boldsymbol{\alpha} - \mathbf{a} \tag{5.43}$$

where $\boldsymbol{\alpha} = [\alpha_{n-1}; \alpha_{n-2}; \dots; \alpha_1; \alpha_0]$ and $\mathbf{a} = [a_{n-1}; a_{n-2}; \dots; a_1; a_0]$. If the state-space representation of the plant is *not* in the controller companion form, a state-transformation can be used to transform the plant to the controller companion form as follows:

$$\mathbf{x}'(t) = \mathbf{T}\mathbf{x}(t);\ \mathbf{A}' = \mathbf{T}\mathbf{A}\mathbf{T}^{-1};\ \mathbf{B}' = \mathbf{T}\mathbf{B} \tag{5.44}$$

where $\mathbf{x}'(t)$ is the state-vector of the plant in the controller companion form, $\mathbf{x}(t)$ is the original state-vector, and $\mathbf{T}$ is the state-transformation matrix. The single-input regulator's control-law (Eq. (5.28)) can thus be expressed as follows:

$$u(t) = -\mathbf{K}\mathbf{x}(t) = -\mathbf{K}\mathbf{T}^{-1}\mathbf{x}'(t) \tag{5.45}$$

Since $\mathbf{K}\mathbf{T}^{-1}$ is the regulator gain matrix when the plant is in the controller companion form, it must be given by Eq. (5.43) as follows:

$$\mathbf{K}\mathbf{T}^{-1} = \boldsymbol{\alpha} - \mathbf{a} \tag{5.46}$$

or

$$\mathbf{K} = (\boldsymbol{\alpha} - \mathbf{a})\mathbf{T} \tag{5.47}$$

Let us derive the state-transformation matrix, $\mathbf{T}$, which transforms a plant to its controller companion form. The controllability test matrix of the plant in its *original* state-space representation is given by

$$\mathbf{P} = [\mathbf{B};\quad \mathbf{AB};\quad \mathbf{A}^2\mathbf{B};\quad \ldots;\quad \mathbf{A}^{n-1}\mathbf{B}] \tag{5.48}$$

Substitution of inverse transformation, $\mathbf{B} = \mathbf{T}^{-1}\mathbf{B}'$, and $\mathbf{A} = \mathbf{T}^{-1}\mathbf{A}'\mathbf{T}$ into Eq. (5.48) yields

$$\begin{aligned}\mathbf{P} &= [\mathbf{T}^{-1}\mathbf{B}';\quad (\mathbf{T}^{-1}\mathbf{A}'\mathbf{T})\mathbf{T}^{-1}\mathbf{B}';\quad (\mathbf{T}^{-1}\mathbf{A}'\mathbf{T})^2\mathbf{T}^{-1}\mathbf{B}';\quad \ldots;\quad (\mathbf{T}^{-1}\mathbf{A}'\mathbf{T})^{n-1}\mathbf{T}^{-1}\mathbf{B}'] \\ &= \mathbf{T}^{-1}[\mathbf{B}';\quad \mathbf{A}'\mathbf{B}';\quad (\mathbf{A}')^2\mathbf{B}';\quad \ldots;\quad (\mathbf{A}')^{n-1}\mathbf{B}'] = \mathbf{T}^{-1}\mathbf{P}'\end{aligned} \tag{5.49}$$

where $\mathbf{P}'$ is the controllability test matrix of the plant in *controller companion form*. Pre-multiplying both sides of Eq. (5.49) with $\mathbf{T}$, and then post-multiplying both sides of the resulting equation with $\mathbf{P}^{-1}$ we get the following expression for $\mathbf{T}$:

$$\mathbf{T} = \mathbf{P}'\mathbf{P}^{-1} \tag{5.50}$$

You can easily show that $\mathbf{P}'$ is the following *upper triangular* matrix (thus called because all the elements *below* its main diagonal are zeros):

$$\mathbf{P}' = \begin{bmatrix} 1 & -a_{n-1} & -a_{n-2} & \ldots & -a_2 & -a_1 \\ 0 & 1 & -a_{n-1} & \ldots & -a_3 & -a_2 \\ 0 & 0 & 1 & \ldots & -a_4 & -a_3 \\ \cdot & \cdot & \cdot & \ldots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \ldots & \cdot & \cdot \\ 0 & 0 & 0 & \ldots & 1 & -a_{n-1} \\ 0 & 0 & 0 & \ldots & 0 & 1 \end{bmatrix} \tag{5.51}$$

Also note from Eq. (5.51) that the determinant of $\mathbf{P}'$ is unity, and that $(\mathbf{P}')^{-1}$ is obtained merely by replacing all the elements above the main diagonal of $\mathbf{P}'$ by their *negatives*. Substituting Eq. (5.50) into Eq. (5.47), the regulator gain matrix is thus given by

$$\mathbf{K} = (\alpha - \mathbf{a})\mathbf{P}'\mathbf{P}^{-1} \tag{5.52}$$

Equation (5.52) is called the *Ackermann's pole-placement formula*. For a single-input plant considered here, both $\mathbf{P}$ and $\mathbf{P}'$ are square matrices of size $(n \times n)$. Note that if the plant is uncontrollable, $\mathbf{P}$ is singular, thus $\mathbf{T} = \mathbf{P}'\mathbf{P}^{-1}$ *does not exist*. This confirms our earlier requirement that for pole-placement, a plant must be controllable.

### *Example 5.9*

Let us design a full-state feedback regulator for an inverted pendulum on a moving cart (Figure 2.59). A linear state-space representation of the plant is given by Eqs. (3.31) and (3.32), of which the state coefficient matrices are the following:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ (M+m)g/(ML) & 0 & 0 & 0 \\ -mg/M & 0 & 0 & 0 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ -1/(ML) \\ 1/M \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}; \quad \mathbf{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5.53}$$

The single-input, $u(t)$, is a force applied horizontally to the cart, and the two outputs are the angular position of the pendulum, $\theta(t)$, and the horizontal position of the cart, $x(t)$. The state-vector of this fourth order plant is $\mathbf{x}(t) = [\theta(t);\ x(t);\ \theta^{(1)}(t);\ x^{(1)}(t)]^T$. Let us assume the numerical values of the plant's parameters as follows: $M = 1$ kg, $m = 0.1$ kg, $L = 1$ m, and $g = 9.8$ m/s$^2$. Then the matrices $\mathbf{A}$ and $\mathbf{B}$ are the following:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 10.78 & 0 & 0 & 0 \\ -0.98 & 0 & 0 & 0 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} \tag{5.54}$$

Let us first determine whether the plant is controllable. This is done by finding the controllability test matrix, $\mathbf{P}$, using the MATLAB (CST) command *ctrb* as follows:

```
>>P = ctrb(A,B)) <enter>

P =

0          -1.0000     0          -10.7800
0          1.0000      0          0.9800
-1.0000    0           -10.7800   0
1.0000     0           0.9800     0
```

The determinant of the controllability test matrix is then computed as follows:

```
>>det(P) <enter>

ans =
-96.0400
```

Since $|\mathbf{P}| \neq 0$, it implies that the plant is controllable. However, the *magnitude* of $|\mathbf{P}|$ depends upon the *scaling* of matrix **P**, and is *not* a good indicator of how *far away* **P** is from being singular, and thus how *strongly* the plant is controllable. A better way of detecting the measure of controllability is the condition number, obtained using the MATLAB function *cond* as follows:

```
>>cond(p) <enter>

ans =
 12.0773
```

Since condition number of **P** is *small* in magnitude, the plant is *strongly* controllable. Thus, our pole-placement results are expected to be accurate. (Had the condition number of **P** been *large* in magnitude, it would have indicated a *weakly* controllable plant, and the inversion of **P** to get the feedback gain matrix would have been inaccurate.) The poles of the plant are calculated by finding the eigenvalues of the matrix **A** using the MATLAB command *damp* as follows:

```
>>damp(A) <enter>

Eigenvalue      Damping       Freq. (rad/sec)
 3.2833         -1.0000       3.2833
    0           -1.0000       0
    0           -1.0000       0
-3.2833          1.0000       3.2833
```

The plant is *unstable* due to a pole with positive real-part (and also due to a pair of poles at $s = 0$). Controlling this unstable plant is like balancing a vertical stick on your palm. The task of the regulator is to stabilize the plant. Let us make the closed-loop system stable, by selecting the closed-loop poles as $s = -1 \pm i$, and $s = -5 \pm 5i$. The coefficients of the plant's characteristic polynomial can be calculated using the MATLAB command *poly* as follows:

```
>>a=poly(A) <enter>

a =

        1.0000  0.0000  -10.7800  0  0
```

which implies that the characteristic polynomial of the plant is $s^4 - 10.78s^2 = 0$. Hence, the polynomial coefficient vector, **a**, is the following:

$$\mathbf{a} = [0; \quad -10.78; \quad 0; \quad 0] \tag{5.55}$$

The characteristic polynomial of the closed-loop system can also be calculated using the command *poly* as follows:

```
>>v = [-1+j; -1-j; -5+5*j; -5-5*j]; alpha = poly(v) <enter>

alpha =

  1  12  72  120  100
```

which implies that the closed-loop characteristic polynomial is $\alpha^4 + 12\alpha^3 + 72\alpha^2 + 120\alpha + 100$, and the vector $\boldsymbol{\alpha}$ is thus the following:

$$\boldsymbol{\alpha} = [12; \quad 72; \quad 120; \quad 100] \tag{5.56}$$

Note that the MATLAB function *poly* can be used to compute the characteristic polynomial either *directly* from a square matrix, or from the *roots* of the characteristic polynomial (i.e. the eigenvalues of a square matrix). It now remains to find the upper triangular matrix, $\mathbf{P}'$, by either Eq. (5.49) or Eq. (5.51). Since a controller companion form is generally ill-conditioned (see Chapter 3), we would like to avoid using Eq. (5.49) which involves higher powers of the ill-conditioned matrix, $\mathbf{A}'$. From Eq. (5.51), we get

$$\mathbf{P}' = \begin{bmatrix} 1 & 0 & 10.78 & 0 \\ 0 & 1 & 0 & 10.78 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.57}$$

Finally, the regulator gain matrix is obtained through Eq. (5.52) as follows:

```
>>Pdash=[1 0 10.78 0; 0 1 0 10.78; 0 0 1 0; 0 0 0 1]; a=[0 -10.78 0 0];

              alpha=[12 72 120 100]; K = (alpha-a)*Pdash*inv(P) <enter>

K =

-92.9841   -10.2041   -24.2449   -12.2449
```

The regulator gain matrix is thus the following:

$$\mathbf{K} = [-92.9841; \quad -10.2041; \quad -24.2449; \quad -12.2449] \tag{5.58}$$

Let us confirm that the eigenvalues of the closed-loop state-dynamics matrix, $\mathbf{A}_{\mathrm{CL}} = \mathbf{A} - \mathbf{BK}$, are indeed what we set out to achieve as follows:

```
>>ACL = A-B*K <enter>

ACL =
 0            0            1.0000       0
 0            0            0            1.0000
 -82.2041     -10.2041     -24.2449     -12.2449
 92.0041      10.2041      24.2449      12.2449
```

The closed-loop poles are then evaluated by the command *eig* as follows:

```
>>eig(ACL) <enter>

ans =
 -5.0000+5.0000i
 -5.0000-5.0000i
 -1.0000+1.0000i
 -1.0000-1.0000i
```

Hence, the desired locations of the closed-loop poles have been obtained.

The computational steps of Example 5.9 are programmed in the MATLAB (CST) function called *acker* for computing the regulator gain matrix for single-input plants using the Ackermann's formula (Eq. (5.52). The command *acker* is used as follows:

```
>>K = acker(A,B,V) <enter>
```

where $A$, $B$ are the state coefficient matrices of the plant, $V$ is a vector containing the desired closed-loop pole locations, and $K$ is the returned regulator gain matrix. Since Ackermann's formula is based on transforming the plant into the controller companion form, which becomes ill-conditioned for large order plants, the computed regulator gain matrix may be inaccurate when $n$ is greater than, say, 10. The command *acker* produces a warning, if the computed closed-loop poles are *more than* 10% *off* from their desired locations. A similar MATLAB (CST) function called *place* is also available for computing the pole-placement regulator gain for single-input plants. The function *place* also provides an output *ndigits*, which indicates the number of significant digits to which the closed-loop poles have been placed. The design of Example 5.9 is simply carried out by using the command *place* as follows:

```
>>V = [-1+j; -1-j; -5+5*j; -5-5*j]; K = place(A,B,V) <enter>

place: ndigits= 17

K =

-92.9841   -10.2041   -24.2449   -12.2449
```

The result is identical to that obtained in Example 5.9; $ndigits = 17$ indicates that the locations of the closed-loop poles match the desired values up to 17 significant digits.

The locations of closed-loop poles determine the performance of the regulator, such as the settling time, maximum overshoot, etc. (see Chapter 2 for performance parameters) when the system is disturbed by a non-zero initial condition. A design is usually specified in terms of such performance parameters, rather than the locations of the closed-loop poles themselves. It is the task of the designer to ensure that the desired performance is achieved by selecting an appropriate set of closed-loop poles. This is illustrated in the following example.

### *Example 5.10*

For the inverted-pendulum on a moving cart of Example 5.9, let us design a regulator which achieves a 5% maximum overshoot and a settling time less than 1 second for both the outputs, when the cart is initially displaced by 0.01 m. The state coefficient matrices, **A**, **B**, **C**, and **D**, of the plant are given in Eq. (5.53). The initial condition vector has the perturbation to the cart displacement, $x(t)$, as the only non-zero element; thus, $\mathbf{x}(0) = [0; 0.01; 0; 0]^T$. Let us begin by testing whether the regulator designed in Example 5.9 meets the performance specifications. This is done by using the MATLAB (CST) function *initial* to find the initial response as follows:

```
>>t = 0:0.1:10; sysCL=ss(A-B*K, zeros(4,1),C,D); [y,t,X] = initial
  (sysCL,[0 0.01 0 0]',t); <enter>
```

where **y**, **X**, and **t** denote the returned output, state, and time vectors and *sysCL* is the state-space LTI model of the closed-loop system. The resulting outputs $\mathbf{y}(t) = [\theta(t); x(t)]^T$ are plotted in Figure 5.4.

In Figure 5.4, both the responses are seen to have acceptably small maximum overshoots, but settling-times in excess of 5 s, which is unacceptable. In order to *speed-up* the closed-loop response, let us move all the poles *deeper* inside the left-half plane by decreasing their real parts such that the new desired closed-loop poles are $s = -7.5 \pm 7.5i$, and $s = -10 \pm 10i$. Then, the new regulator gain matrix, the closed-loop dynamics matrix, and the initial response are obtained as follows:

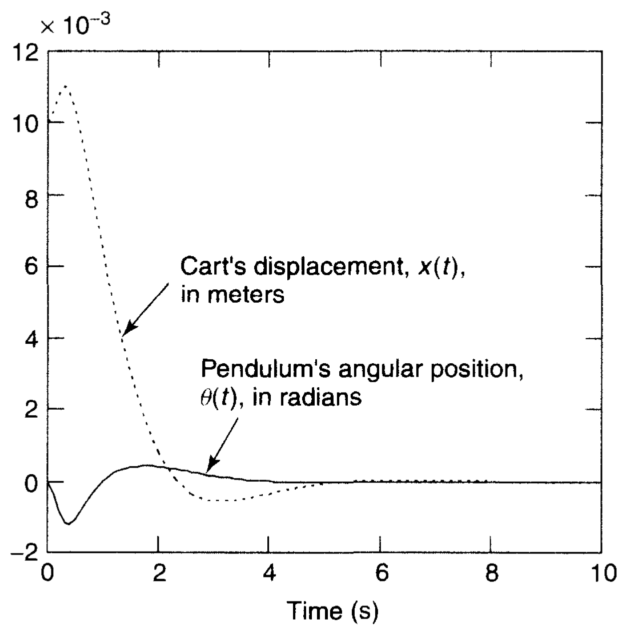


**Figure 5.4** Closed-loop initial response of the regulated inverted pendulum on a moving cart to perturbation on cart displacement for the regulator gain matrix, $\mathbf{K} = [-92.9841; -10.2041; -24.2449; -12.2449]$

```
>>V=[-7.5+7.5*j -7.5-7.5*j -10+10*j -10-10*j]'; K = place(A,B,V) <enter>

place: ndigits= 19

K =

-2.9192e+003   -2.2959e+003   -5.7071e+002   -5.3571e+002
>>t = 0:0.01:2; sysCL=ss(A-B*K, zeros(4,1),C,D); [y,t,X] = initial(sysCL,
  [0 0.01 0 0]',t); <enter>
```

The resulting outputs are plotted in Figure 5.5, which indicates a maximum overshoot of the steady-state values less than 4%, and a settling time of less than 1 s for both the responses.

How did we know that the new pole locations will meet our performance requirements? We didn't. We tried for several pole configurations, until we hit upon the one that met our requirements. This is the design approach in a nutshell. On comparing Figures 5.4 and 5.5, we find that by moving the closed-loop poles further inside the left-half plane, we *speeded-up* the initial response *at the cost* of increased maximum overshoot. The settling time and maximum overshoot are, thus, *conflicting requirements*. To decrease one, we have to accept an increase in the other. Such a compromise, called a *trade-off*, is a hallmark of control system design. Furthermore, there is another cost associated with moving the poles deeper inside the left-half plane – that of the control input. Note that the new regulator gain elements are *several times larger* than those calculated in Example 5.9, which implies that the regulator must now apply an input which is much

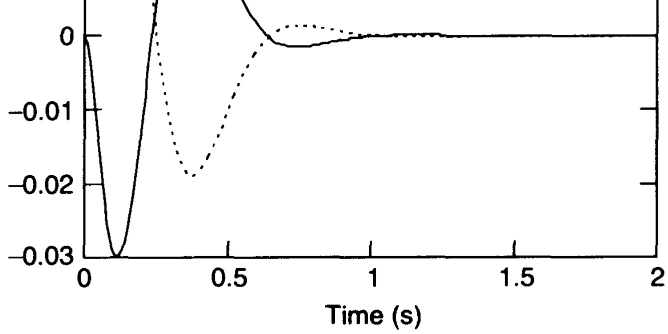


**Figure 5.5** Closed-loop initial response of the regulated inverted pendulum on a moving cart to perturbation on cart displacement for the regulator gain matrix, $\mathbf{K} = [-2919.2; -2295.9; -570.71; -535.71]$

larger in magnitude than that in Example 5.9. The input, $\mathbf{u}(t) = -\mathbf{Kx}(t)$, can be calculated from the previously calculated matrices, **K** and **x**, as follows:

```
>>u = -K*X'; <enter>
```

The control inputs for the two values of the regulator gain matrix are compared in Figure 5.6. The control input, $u(t)$, which is a force applied to the cart, is seen to be *more than 200 times* in magnitude for the design of Example 5.10 than that of Example 5.9. The *actuator*, which applies the input force to the cart, must be physically able to generate this force for the design to be successful. The cost of controlling a plant is a function of the largest control input magnitude expected in *actual operating conditions*. For example, if the largest *expected* initial disturbance in cart displacement were 0.1 m instead of 0.01 m, a *ten times larger* control input would be required than that in Figure 5.6. The larger the control input magnitude, the bigger would be the energy spent by the actuator in generating the control input, and the higher would be the cost of control. It is possible to *minimize* the control effort required in controlling a plant by imposing conditions – other than pole-placement – on the regulator gain matrix, which we will see in Chapter 6. However, a rough method of ensuring that the performance requirements are met with the minimum control effort is to ensure that all the closed-loop poles are about the *same distance* from the imaginary axis in the left-half plane. The poles in the left-half plane that are *farthest* away from the imaginary axis dictate the control input magnitude, while the *speed* of response (i.e. the settling time of the transients) is governed by the poles with the *smallest* real parts, called the *dominant poles*. If some closed-loop poles are close to, and some are very far from the imaginary axis, it implies that *too much control energy* is being spent for a given settling time, and thus the design is *inefficient*. The most efficient closed-loop configuration thus appears to be the one where all the poles are placed in the

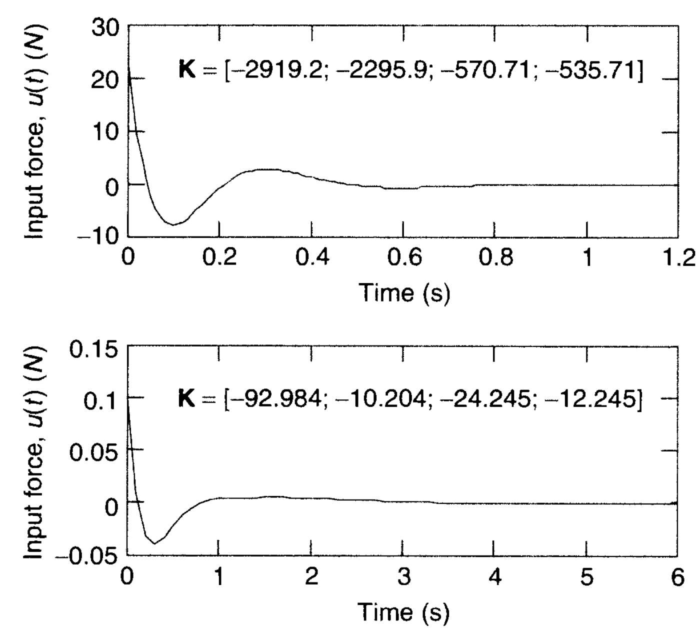


**Figure 5.6** Control inputs of the regulated inverted pendulum on a moving cart for two designs of the full-state feedback regulator

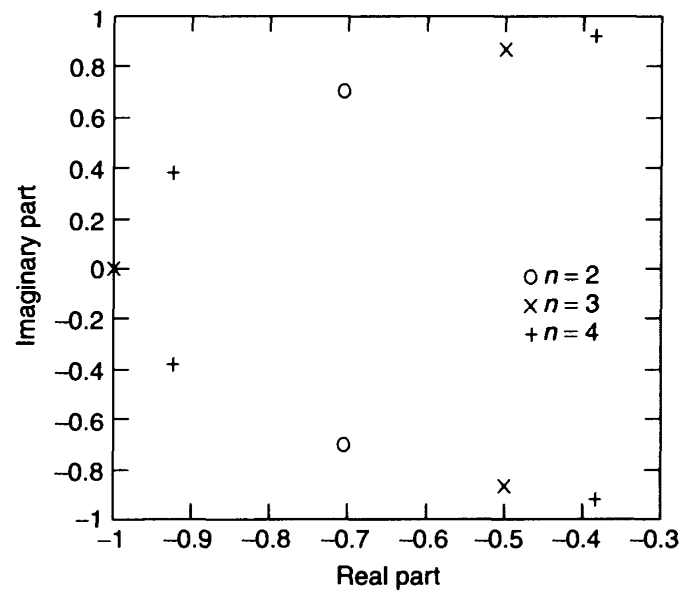


**Figure 5.7** Butterworth pattern of poles in the left-half plane for $n = 2$, 3, and 4

left half plane, roughly the same distance from the imaginary axis. To increase the speed of the closed-loop response, one has to just increase this distance. One commonly used closed-loop pole configuration is the *Butterworth pattern*, in which the poles are placed on a circle of radius $R$ *centered at the origin*, and are obtained from the solution of the following equation:

$$(s/R)^{2n} = (-1)^{n+1} \tag{5.59}$$

where $n$ is the number of poles in the left-half plane (usually, we want all the closed-loop poles in the left-half plane; then $n$ is the order of the system). For $n = 1$, the pole in the left-half plane satisfying Eq. (5.59) is $s = -R$. For $n = 2$, the poles in the left-half plane satisfying Eq. (5.59) are the solutions of $(s/R)^2 + (s/R)\sqrt{2} + 1 = 0$. The poles satisfying Eq. (5.59) in the left-half plane for $n = 3$ are the solutions of $(s/R)^3 + 2(s/R)^2 + 2(s/R) + 1 = 0$. For a given $n$, we can calculate the poles satisfying Eq. (5.59) by using the MATLAB function *roots*, and discard the poles having positive real parts. The Butterworth pattern for $n = 2$, 3, and 4 is shown in Figure 5.7. Note, however, that as *n increases*, the real part of the two Butterworth poles closest to the imaginary axis *decreases*. Thus for large $n$, it may be required to move these two poles further inside the left-half plane, in order to meet a given speed of response.

### *Example 5.11*

Let us compare the closed-loop initial response and the input for the inverted pendulum on a moving cart with those obtained in Example 5.10 when the closed-loop poles are in a Butterworth pattern. For $n = 4$, the poles satisfying Eq. (5.59) in the left-half plane are calculated as follows:

```
>>z = roots([1 0 0 0 0 0 0 0 1]) <enter>

z =

 -0.9239+0.3827i
 -0.9239-0.3827i
 -0.3827+0.9239i
 -0.3827-0.9239i
  0.3827+0.9239i
  0.3827-0.9239i
  0.9239+0.3827i
  0.9239-0.3827i
```

The first four elements of $z$ are the required poles in the left-half plane, i.e. $s/R = -0.9239 \pm 0.3827i$ and $s/R = -0.3827 \pm 0.9239i$. For obtaining a maximum overshoot less than 5% and settling-time less than 1 s for the initial response (the design requirements of Example 5.10), let us choose $R = 15$. Then the closed-loop characteristic polynomial are obtained as follows:

```
>>i = find(real(z) < 0); p = poly(15*z(i)) <enter>

p =
 Columns 1 through 3
 1.0000e+000        3.9197e+001-3.5527e-015i     7.6820e+002-5.6843e-014i

 Columns 4 through 5
 8.8193e+003-3.1832e-012i        5.0625e+004-2.1654e-011i
```

Neglecting the small imaginary parts of **p**, the closed-loop characteristic polynomial is $s^4 + 39.197s^3 + 768.2s^2 + 8819.3s + 50625$, with the vector $\alpha$ given by

```
>>alpha=real(p(2:5)) <enter>

alpha =
  3.9197e+001   7.6820e+002   8.8193e+003   5.0625e+004
```

$$\boldsymbol{\alpha} = [39.197; \quad 768.2; \quad 8819.3; \quad 50\,625] \tag{5.60}$$

Then using the values of **a, P**, and **P′** calculated in Example 5.9, the regulator gain matrix is calculated by Eq. (5.52) as follows:

```
>>K = (alpha-a)*Pdash*inv(P) <enter>

K =
-5.9448e+003   -5.1658e+003   -9.3913e+002   -8.9993e+002
```

and the closed-loop state-dynamics matrix is obtained as

```
>>ACL=A-B*K <enter>

ACL =

0               0               1.0000e+000    0
0               0               0              1.0000e+000
-5.9340e+003    -5.1658e+003    -9.3913e+002   -8.9993e+002
5.9438e+003     5.1658e+003     9.3913e+002    8.9993e+002
```

The closed-loop eigenvalues are calculated as follows:

```
>>eig(ACL) <enter>

ans =
-5.7403e+000+1.3858e+001i
-5.7403e+000-1.3858e+001i
-1.3858e+001+5.7403e+000i
-1.3858e+001-5.7403e+000i
```

which are the required closed-loop Butterworth poles for $R = 15$. The initial response of the closed-loop system is calculated as follows, and is plotted in Figure 5.8:

```
>>t = 0:1.0753e-2:1.2; sysCL=ss(ACL,zeros(4,1),C,D); [y,t,X]=initial
  (sysCL,[0 0.01 0 0]',t); <enter>
```

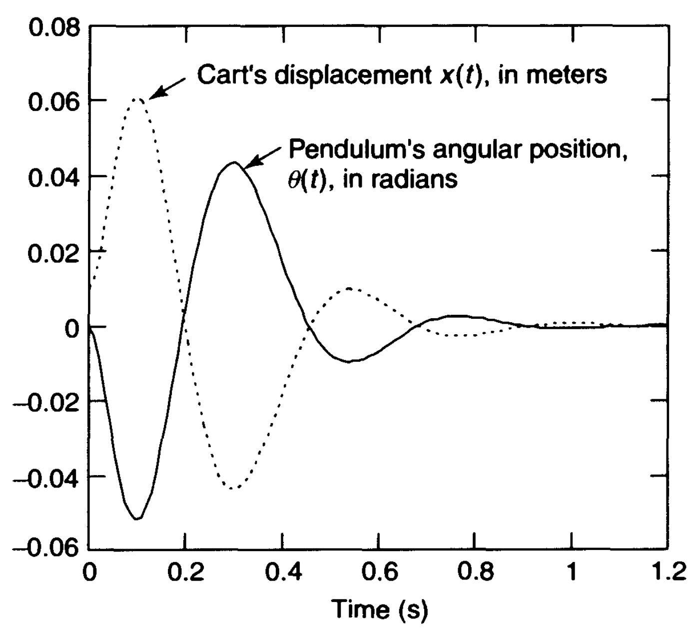


**Figure 5.8** Initial response of the regulated inverted pendulum on a moving cart, for the closed-loop poles in a Butterworth pattern of radius, $R = 15$

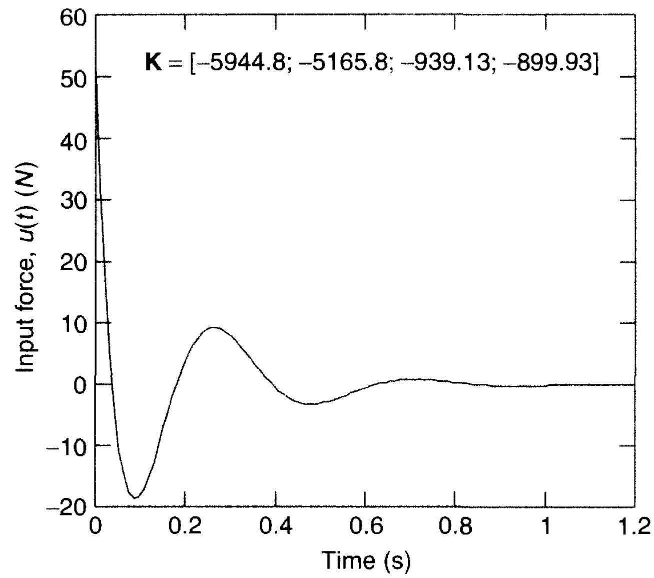


**Figure 5.9** Control input for the regulated inverted pendulum on a moving cart, for closed-loop poles in a Butterworth pattern of radius, $R = 15$

Note from Figure 5.8 that the maximum overshoot for cart displacement is about 6% for both the outputs, and the settling time is greater than 1 s. The design is thus unacceptable. The *slow* closed-loop response is caused by the pair of *dominant poles* with real part $-5.7403$. If we try to increase the real part magnitude of the dominant poles by increasing $R$, we will have to pay for the increased speed of response in terms of *increased* input magnitude, because the poles furthest from the imaginary axis ($s/R = -0.9239 \pm 0.3827i$) will move still further away. The control input, $u(t)$, is calculated and plotted in Figure 5.9 as follows:

```
>>u = -K*X'; plot(t,u) <enter>
```

Figure 5.9 shows that the control input magnitude is much larger than that of the design in Example 5.10. The present pole configuration is unacceptable, because it does not meet the design specifications, and requires a large control effort. To reduce the control effort, we will try a Butterworth pattern with $R = 8.5$. To increase the speed of the response, we will move the *dominant poles* further inside the left-half plane than dictated by the Butterworth pattern, such that *all* the closed-loop poles have the *same* real parts. The selected closed-loop pole configuration is $s = -7.853 \pm 3.2528i$, and $s = -7.853 \pm 7.853i$. The regulator gain matrix which achieves this pole placement is obtained using MATLAB as follows:

```
>>format long e <enter>

>>v=[-7.853-3.2528i -7.853+3.2528i -7.853-7.853i -7.853+7.853i]';K=place
  (A,B,v) <enter>

place: ndigits= 18
```

```
K =
Columns 1 through 3
-1.362364050360232e+003 -9.093160795202226e+002 -3.448741667548096e+002

Column 4
-3.134621667548089e+002
```

Note that we have printed out **K** using the *long format*, because we will need this matrix later. A short format would have introduced unacceptable truncation errors. The closed-loop initial response is calculated and plotted in Figure 5.10 as follows:

```
>>sysCL=ss(A-B*K,zeros(4,1),C,D); [y,t,X] = initial(sysCL,
  [0 0.01 0 0]', t); <enter>
```

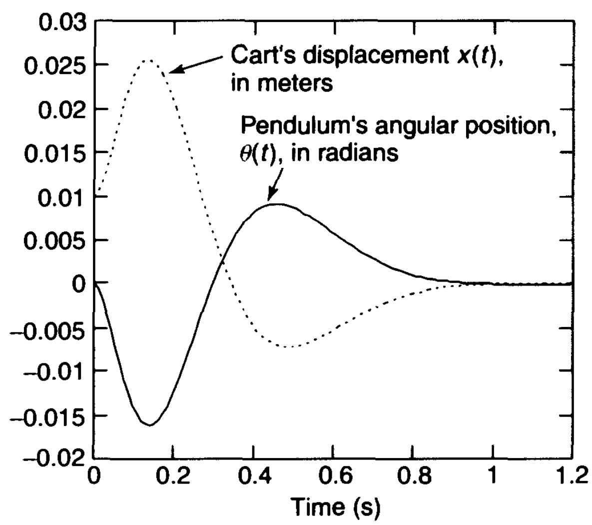


**Figure 5.10** Initial response of the regulated inverted pendulum on a moving cart for the design of Example 5.11 with the closed-loop poles at $s = -7.853 \pm 3.2528i$, and $s = -7.853 \pm 7.853i$

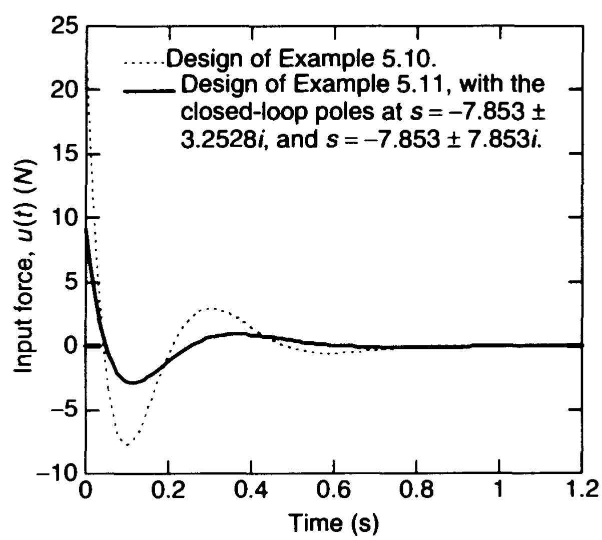


**Figure 5.11** Comparison of the control input for the design of Example 5.10 with that of Example 5.11 with closed-loop poles at $s = -7.853 \pm 3.2528i$, and $s = -7.853 \pm 7.853i$

Figure 5.10 shows that the closed-loop response has a maximum overshoot of about 2.5% and a settling time of 1 s, which is a *better performance* than the design of Example 5.10. The control input of the present design is compared with that of Example 5.10 in Figure 5.11, which shows that the former is less than half of the latter. Hence, the present design results in a *better performance*, while requiring a *much smaller control effort*, when compared to Example 5.10.

### 5.3.2 Pole-placement regulator design for multi-input plants

For a plant having more than one input, the full-state feedback regulator gain matrix of Eq. (5.28) has $(r \times n)$ elements, where $n$ is the order of the plant and $r$ is the number of inputs. Since the number of poles that need to be placed is $n$, we have *more design parameters* than the number of poles. This over-abundance of design parameters allows us to specify *additional design conditions*, apart from the location of $n$ poles. What can be these additional conditions? The answer depends upon the nature of the plant. For example, it is possible that a particular state variable is *not necessary* for generating the control input vector by Eq. (5.28); hence, the *column* corresponding to that state variable in $\mathbf{K}$ can be chosen as zero, and the pole-placement may yet be possible. Other conditions on $\mathbf{K}$ could be due to *physical relationships* between the inputs and the state variables; certain input variables could be *more closely related* to *some* state variables, requiring that the elements of $\mathbf{K}$ corresponding to the *other* state variables should be zeros. Since the structure of the regulator gain matrix for multi-input systems is *system specific*, we cannot derive a *general* expression for the regulator gain matrix, such as Eq. (5.52) for the single-input case. The following example illustrates the multi-input design process.

***Example 5.12***

Let us design a full-state feedback regulator for the following plant:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & -0.1 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \\ 0 & -2 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}; \quad \mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.61}$$

The plant is unstable due to a pole at $s = 0.01$. The rank of the controllability test matrix of the plant is obtained as follows:

```
>>rank(ctrb(A, B)) <enter>

ans =

    3
```

Hence, the plant is controllable, and the closed-loop poles can be placed at will. The general regulator gain matrix is as follows:

$$\mathbf{K} = \begin{bmatrix} K_1 & K_2 & K_3 \\ K_4 & K_5 & K_6 \end{bmatrix} \tag{5.62}$$

and the closed-loop state dynamics matrix is the following:

$$\mathbf{A_{CL}} = \mathbf{A} - \mathbf{BK} = \begin{bmatrix} -K_1 & -K_2 & -K_3 \\ K_4 & (0.01 + K_5) & K_6 \\ 2K_4 & 2K_5 & (-0.1 + 2K_6) \end{bmatrix} \tag{5.63}$$

which results in the following closed-loop characteristic equation:

$$|s\mathbf{I} - \mathbf{A_{CL}}| = \begin{vmatrix} (s + K_1) & K_2 & K_3 \\ -K_4 & (s - 0.01 - K_5) & -K_6 \\ -2K_4 & -2K_5 & (s + 0.1 - 2K_6) \end{vmatrix} = 0 \tag{5.64}$$

or

$$\begin{aligned}(s + K_1)[(s - 0.01 - K_5)(s + 0.1 - 2K_6) - 2K_5K_6] + K_4[K_2(s + 0.1 - 2K_6) \\ + 2K_3K_5] + 2K_4[K_2K_6 + K_3(s - 0.01 - K_5)] = 0\end{aligned} \tag{5.65}$$

or

$$\begin{aligned}s^3 + (0.09 - K_5 - 2K_6 + K_1)s^2 + (K_2K_4 + 2K_3K_4 + 0.09K_1 - 2K_1K_6 \\ - K_1K_5 - 0.001 + 0.02K_6 - 0.1K_5)s + 0.1K_2K_4 + 0.02K_1K_6 \\ - 0.001K_1 - 0.1K_1K_5 - 0.02K_3K_4 = 0\end{aligned} \tag{5.66}$$

Let us choose the closed-loop poles as $s = -1$, and $s = -0.045 \pm 0.5i$. Then the closed-loop characteristic equation must be $(s + 1)(s + 0.045 - 0.5i)(s + 0.045 + 0.5i) = s^3 + 1.09s^2 + 0.342s + 0.252 = 0$, and comparing with Eq. (5.66), it follows that

$$\begin{aligned}&K_1 - K_5 - 2K_6 = 1; \\ &K_2K_4 + 2K_3K_4 + 0.09K_1 - 2K_1K_6 - K_1K_5 + 0.02K_6 - 0.1K_5 = 0.343 \\ &0.1K_2K_4 + 0.02K_1K_6 - 0.001K_1 - 0.1K_1K_5 - 0.02K_3K_4 = 0.252\end{aligned} \tag{5.67}$$

which is a set of nonlinear algebraic equations to be solved for the regulator design parameters – apparently a hopeless task by hand. However, MATLAB (CST) again comes to our rescue by providing the function *place*, which allows placing the poles of multi-input plants. The function *place* employs an *eigenstructure assignment* algorithm [3], which specifies *additional conditions* to be satisfied by the regulator gain elements, provided the *multiplicity* of each pole to be placed *does not exceed* the number of inputs, and all complex closed-loop poles must appear in conjugate pairs. For the present example, the regulator gain matrix is determined using *place* as follows:

```
>>A=[0 0 0;0 0.01 0;0 0 -0.1];B=[1 0;0 -1;0 -2]; p=[-1 -0.045-0.5i
  -0.045+0.5I]; K=place(A,B,p) <enter>

place: ndigits= 16

K =

 0.9232 0.1570  -0.3052
 0.1780 -2.4595 1.1914

>>eig(A-B*K) <enter>

ans =

 -1.0000
 -0.0450+0.5000i
 -0.0450-0.5000i
```

You may verify that the computed values of the gain matrix satisfies Eq. (5.67).The *optimal control* methods of Chapter 6 offer an alternative design approach for regulators based on multi-input plants.

### 5.3.3 Pole-placement regulator design for plants with noise

In the previous two sections, we had ignored the presence of disturbances, or *noise*, in a plant when designing full-state feedback regulators. Designs that ignore noise in a plant are likely to fail when implemented in actual conditions where noise exists. Noise can be divided into two categories: *measurement noise*, or the noise caused by imperfections in the sensors that measure the output variables; and the *process noise*, or the noise which arises due to ignored dynamics when modeling a plant. Since neither the sensors nor a plant's mathematical model can be perfect, we should always expect some noise in a plant. The state-equation of a plant with noise vector, $\mathbf{x}_n(t)$, is the following:

$$\mathbf{x}^{(1)}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{F}\mathbf{x}_n(t) \tag{5.68}$$

where $\mathbf{F}$ is the noise coefficient matrix. To place the closed-loop poles at desired locations while counteracting the effect of the noise, a full-state feedback regulator is to be designed based on the following control-law:

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t) - \mathbf{K}_n\mathbf{x}_n(t) \tag{5.69}$$

Substituting Eq. (5.69) into Eq. (5.68) yields the following state-equation of the closed-loop system:

$$\mathbf{x}^{(1)}(t) = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}(t) + (\mathbf{F} - \mathbf{B}\mathbf{K}_n)\mathbf{x}_n(t) \tag{5.70}$$

Note that Eq. (5.70) implies that the noise vector, $\mathbf{x}_n(t)$, acts as an *input vector* for the closed-loop system, whose state-dynamics matrix is $\mathbf{A}_{CL} = (\mathbf{A} - \mathbf{B}\mathbf{K})$. A schematic diagram of the full-state feedback regulator with noise is shown in Figure 5.12.

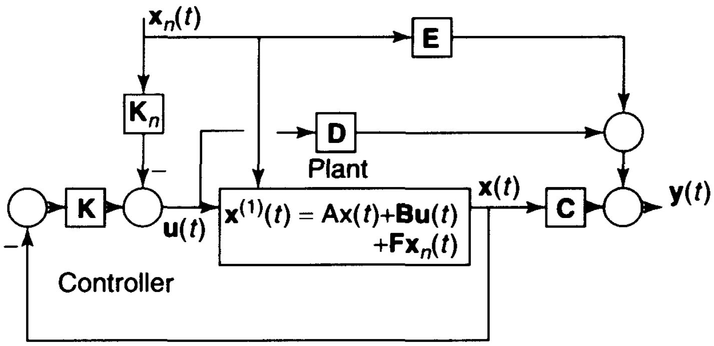


**Figure 5.12** Schematic diagram of a full-state feedback regulator with noise, $\mathbf{x}_n(t)$

The regulator feedback gain matrix, **K**, is selected, as before, to place the closed-loop poles (eigenvalues of $\mathbf{A}_{\mathrm{CL}}$) at desired locations. While we may not know the exact process by which the noise, $\mathbf{x}_n(t)$, is generated (because it is usually a *stochastic process*, as discussed in Chapter 1), we can develop an *approximation* of how the noise affects the plant by deriving the noise coefficient matrix, **F**, from experimental observations. Once **F** is known reasonably, the regulator noise gain matrix, $\mathbf{K}_n$, can be selected such that the effect of the noise vector, $\mathbf{x}_n(t)$, on the closed-loop system is minimized. It would, of course, be ideal if we can make $(\mathbf{F} - \mathbf{B}\mathbf{K}_n) = \mathbf{0}$, in which case there would be absolutely no influence of the noise on the closed-loop system. However, it may not be always possible to select the $(rq)$ unknown elements of $\mathbf{K}_n$ to satisfy the $(nq)$ scalar equations constituting $(\mathbf{F} - \mathbf{B}\mathbf{K}_n) = \mathbf{0}$, where $n$ is the order of the plant, $r$ is the number of inputs, and $q$ is the number of noise variables in the noise vector, $\mathbf{x}_n(t)$. When $r < n$ (as it is usually the case), the number of unknowns in $(\mathbf{F} - \mathbf{B}\mathbf{K}_n) = \mathbf{0}$ is *less than* the number of scalar equations, and hence all the equations cannot be satisfied. If $r = n$, and the matrix **B** is non-singular, then we can uniquely determine the regulator noise gain matrix by $\mathbf{K}_n = -\mathbf{B}^{-1}\mathbf{F}$. In the rare event of $r > n$, the number of unknowns *exceed* the number of equations, and all the equations, $(\mathbf{F} - \mathbf{B}\mathbf{K}_n) = \mathbf{0}$, can be satisfied by appropriately selecting the unknowns, though not uniquely.

### *Example 5.13*

Consider a fighter aircraft whose state-space description given by Eqs. (5.26) and (5.27) has the following coefficient matrices:

$$\mathbf{A} = \begin{bmatrix} -1.7 & 50 & 260 \\ 0.22 & -1.4 & -32 \\ 0 & 0 & -12 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} -272 \\ 0 \\ 14 \end{bmatrix}; \quad \mathbf{F} = \begin{bmatrix} 0.02 & 0.1 \\ -0.0035 & 0.004 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{C} = \mathbf{I}; \quad \mathbf{D} = \mathbf{0}; \quad \mathbf{E} = \mathbf{0} \tag{5.71}$$

The state variables of the aircraft model are normal acceleration in m/s$^2$, $x_1(t)$, pitch-rate in rad/s, $x_2(t)$, and elevator deflection in rad, $x_3(t)$, while the input, $u(t)$, is the desired elevator deflection in rad. (For a graphical description of the system's variables, see Figure 4.5.) The poles of the plant are calculated as follows:

```
>>A=[-1.7 50 260; 0.22 -1.4 -32; 0 0 -12]; damp(A) <enter>
Eigenvalue    Damping        Freq. (rad/sec)
  1.7700      -1.0000        1.7700
 -4.8700      1.0000         4.8700
-12.0000      1.0000 12.0000
```

The plant is unstable due to a pole at $s = 1.77$. To stabilize the closed-loop system, it is desired to place the closed-loop poles at $s = -1 \pm i$ and $s = -1$. The following controllability test reveals a controllable plant, implying that pole-placement is possible:

```
>>B=[-272 0 14]'; rank(ctrb(A,B)) <enter>

ans =

    3
```

The regulator feedback gain matrix is thus obtained as follows:

```
>>v = [-1-i -1+i -1]; K = place(A,B,v) <enter>

place: ndigits= 19

K =

0.0006   -0.0244   -0.8519
```

and the closed-loop state dynamics matrix is the following:

```
>>ACL=A-B*K <enter>
ACL =

-1.5267      43.3608      28.2818
0.2200       -1.4000      -32.0000
-0.0089      0.3417       -0.0733
```

To determine the remaining regulator matrix, $\mathbf{K}_n = [K_{n1} \quad K_{n2}]$, let us look at the matrix $(\mathbf{F} - \mathbf{B}\mathbf{K}_n)$:

$$\mathbf{F} - \mathbf{B}\mathbf{K}_n = \begin{bmatrix} 0.02 + 272K_{n1} & 0.1 + 272K_{n2} \\ -0.0035 & 0.004 \\ -14K_{n1} & -14K_{n2} \end{bmatrix} \tag{5.72}$$

Equation (5.72) tells us that it is *impossible* to make all the elements of $(\mathbf{F} - \mathbf{B}\mathbf{K}_n)$ zeros, by selecting the two unknown design parameters, $K_{n1}$ and $K_{n2}$. The next best thing to $(\mathbf{F} - \mathbf{B}\mathbf{K}_n) = 0$ is making the *largest elements* of $(\mathbf{F} - \mathbf{B}\mathbf{K}_n)$ zeros, and living with the other non-zero elements. This is done by selecting $K_{n1} = -0.02/272$ and $K_{n2} = -0.1/272$ which yields the following $(\mathbf{F} - \mathbf{B}\mathbf{K}_n)$:

$$\mathbf{F} - \mathbf{B}\mathbf{K}_n = \begin{bmatrix} 0 & 0 \\ -0.0035 & 0.004 \\ 0.00103 & 0.00515 \end{bmatrix} \tag{5.73}$$

With $(\mathbf{F} - \mathbf{BK}_n)$ given by Eq. (5.73), we are always going to have some effect of noise on the closed-loop system, which hopefully, will be small. The most satisfying thing about Eq. (5.73) is that the closed-loop system given by Eq. (5.70) is *uncontrollable with noise as the input* (you can verify this fact by checking the rank of *ctrb* $(\mathbf{A}_{\mathrm{CL}}, (\mathbf{F} - \mathbf{BK}_n))$. This means that the noise is not going to affect all the state variables of the closed-loop system. Let us see by what extent the noise affects our closed-loop design by calculating the system's response with a noise vector, $\mathbf{x}_n(t) = [1 \times 10^{-5}; -2 \times 10^{-6}]^T \sin(100t)$, which acts as an input to the closed-loop system given by Eq. (5.70), with zero initial conditions. Such a noise model is too simple; actual noise is *non-deterministic* (or *stochastic*), and consists of a combination of several frequencies, rather than only one frequency (100 rad/s) as assumed here. The closed-loop response to noise is calculated by using the MATLAB (CST) command *lsim* as follows:

```
>>t=0:0.01:5; xn=[1e-5 -2e-6]'*sin(100*t); Bn=[0 0;-3.5e-3 0.004;1.03e-3 5.15e-3];
  <enter>

>>sysCL=ss(ACL,Bn,eye(3),zeros(3,2)); [y,t,X]=lsim(sysCL,xn',t'); plot(t,X) <enter>
```

The resulting closed-loop state variables, $x_1(t)$, $x_2(t)$, and $x_3(t)$, are plotted in Figure 5.13, which shows oscillations with very small amplitudes. Since the amplitudes are very small, the effect of the noise on the closed-loop system can be said to be negligible. Let us see what may happen if we make the closed-loop system *excessively* stable. If the closed-loop poles are placed at $s = -100$, $s = -100 \pm 100i$, the resulting closed-loop response to the noise is shown in Figure 5.14. Note that the closed-loop response has increased by about 300 times in

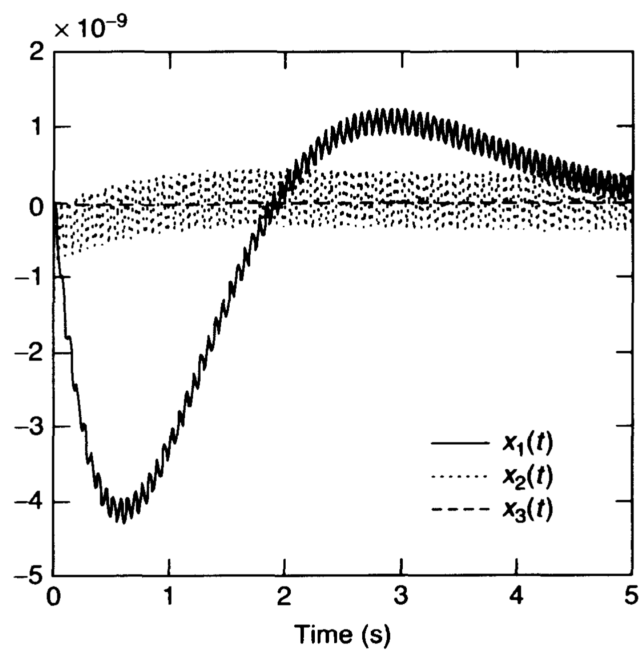


**Figure 5.13** Closed-loop response of the regulated fighter aircraft to noise, when the closed-loop poles are $s = -1$, $s = -1 \pm i$

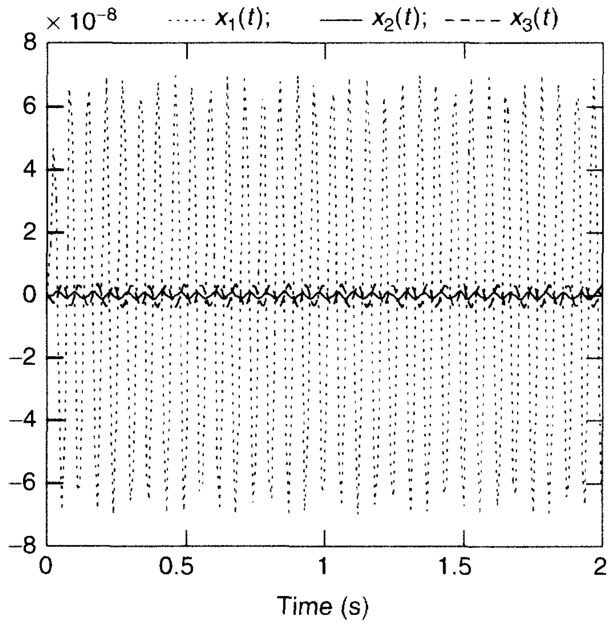


**Figure 5.14** Closed-loop response of the regulated fighter aircraft to noise, when the closed-loop poles are $s = -100$, $s = -100 \pm 100i$

magnitude, compared with that of Figure 5.13. Therefore, moving the poles too far into the left-half plane has the effect of increasing the response of the system due to noise, which is undesirable. This kind of amplified noise effect is due to the resulting *high gain feedback*. High gain feedback is to be avoided in the frequency range of expected noise. This issue is appropriately dealt with by *filters* and *compensators* (Chapter 7).

The conflicting requirements of increasing the speed of response, and decreasing the effect of noise are met by a pole configuration that is neither too deep inside the left-half plane, nor too close to the imaginary axis. The *optimum* pole locations are obtained by trial and error, if we follow the pole-placement approach. However, the *optimal control* methods of Chapters 6 and 7 provide a more effective procedure of meeting both speed and noise attenuation requirements than the pole-placement approach.

## 5.3.4 Pole-placement design of tracking systems

Now we are in a position to extend the pole-placement design to *tracking systems*, which are systems in which the desired state-vector, $\mathbf{x}_d(t)$, is *non-zero*. Schematic diagram of a tracking system with noise was shown in Figure 5.2, with the plant described by Eqs. (5.26) and (5.27), and the control-law given by Eq. (5.25). The objective of the tracking system is to make the error, $\mathbf{e}(t) = (\mathbf{x}_d(t) - \mathbf{x}(t))$, zero in the steady-state, while counteracting the effect of the noise, $\mathbf{x}_n(t)$. If the process by which the desired state-vector is generated is *linear* and *time-invariant*, it can be represented by the following state-equation:

$$\mathbf{x}_d^{(1)}(t) = \mathbf{A}_d\mathbf{x}_d(t) \tag{5.74}$$

Note that Eq. (5.74) represents a homogeneous system, because the desired state vector is unaffected by the input vector, $\mathbf{u}(t)$. Subtracting Eq. (5.26) from Eq. (5.74), we can write the following plant state-equation in terms of the error:

$$\mathbf{x}_d^{(1)}(t) - \mathbf{x}^{(1)}(t) = \mathbf{A}_d\mathbf{x}_d(t) - \mathbf{A}\mathbf{x}(t) - \mathbf{B}\mathbf{u}(t) - \mathbf{F}\mathbf{x}_n(t) \tag{5.75}$$

or

$$\mathbf{e}^{(1)}(t) = \mathbf{A}\mathbf{e}(t) + (\mathbf{A}_d - \mathbf{A})\mathbf{x}_d(t) - \mathbf{B}\mathbf{u}(t) - \mathbf{F}\mathbf{x}_n(t) \tag{5.76}$$

and the control-law (Eq. (5.25)) can be re-written as follows:

$$\mathbf{u}(t) = \mathbf{K}\mathbf{e}(t) - \mathbf{K}_d\mathbf{x}_d(t) - \mathbf{K}_n\mathbf{x}_n(t) \tag{5.77}$$

Referring to Figure 5.2, we see that while $\mathbf{K}$ is a *feedback* gain matrix (because it multiplies the error signal which is generated by the fed back state-vector), $\mathbf{K}_d$ and $\mathbf{K}_n$ are *feedforward* gain matrices, which multiply the desired state-vector and the noise vector, respectively, and hence *feed* these two vectors *forward* into the control system. Substituting Eq. (5.77) into Eq. (5.76) yields the following state-equation for the tracking system:

$$\mathbf{e}^{(1)}(t) = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{e}(t) + (\mathbf{A}_d - \mathbf{A} + \mathbf{B}\mathbf{K}_d)\mathbf{x}_d(t) + (\mathbf{B}\mathbf{K}_n - \mathbf{F})\mathbf{x}_n(t) \tag{5.78}$$

The design procedure for the tracking system consists of determining the full-state feedback gain matrix, $\mathbf{K}$, such that the poles of the closed-loop system (i.e. eigenvalues of $\mathbf{A}_{CL} = \mathbf{A} - \mathbf{B}\mathbf{K}$) are placed at desired locations, and choose the gain matrices, $\mathbf{K}_d$ and $\mathbf{K}_n$, such that the error, $\mathbf{e}(t)$, is either *reduced to zero*, or *made as small as possible* in the *steady-state*, in the presence of the noise, $\mathbf{x}_n(t)$. Of course, the closed-loop system described by Eq. (5.78) must be *asymptotically stable*, i.e. all the closed-loop poles must be in the left-half plane, otherwise the error will not reach a steady-state *even in the absence of noise*. Furthermore, as seen in Example 5.13, there may not be enough design parameters (i.e. elements in $\mathbf{K}_d$ and $\mathbf{K}_n$) to make the error zero in the steady-state, in the presence of noise. If all the closed-loop poles are placed in the left-half plane, the tracking system is asymptotically stable, and the steady-state condition for the error is reached (i.e. the error becomes constant in the limit $t \to \infty$). Then the steady state condition is described by $\mathbf{e}^{(1)}(t) = \mathbf{0}$, and Eq. (5.78) becomes the following in the steady state:

$$\mathbf{0} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{e}_{ss} + (\mathbf{A}_d - \mathbf{A} + \mathbf{B}\mathbf{K}_d)\mathbf{x}_{dss} + (\mathbf{B}\mathbf{K}_n - \mathbf{F})\mathbf{x}_{nss} \tag{5.79}$$

where $\mathbf{e}(t) \to \mathbf{e}_{ss}$ (the steady state error vector), $\mathbf{x}_d(t) \to \mathbf{x}_{dss}$, and $\mathbf{x}_n(t) \to \mathbf{x}_{nss}$ as $t \to \infty$. From Eq. (5.79), we can write the steady state error vector as follows:

$$\mathbf{e}_{ss} = (\mathbf{A} - \mathbf{B}\mathbf{K})^{-1}[(\mathbf{A} - \mathbf{B}\mathbf{K}_d - \mathbf{A}_d)\mathbf{x}_{dss} + (\mathbf{F} - \mathbf{B}\mathbf{K}_n)\mathbf{x}_{nss}] \tag{5.80}$$

Note that the closed-loop state-dynamics matrix, $\mathbf{A}_{\mathrm{CL}} = \mathbf{A} - \mathbf{BK}$, is non-singular, because all its eigenvalues are in the left-half plane. Hence, $(\mathbf{A} - \mathbf{BK})^{-1}$ exists. For $\mathbf{e}_{ss}$ to be zero, irrespective of the values of $\mathbf{x}_{dss}$ and $\mathbf{x}_{nss}$, we should have $(\mathbf{A} - \mathbf{BK}_d - \mathbf{A}_d) = \mathbf{0}$ and $(\mathbf{F} - \mathbf{BK}_n) = \mathbf{0}$, by selecting the appropriate gain matrices, $\mathbf{K}_d$ and $\mathbf{K}_n$. However, as seen in Example 5.13, this is seldom possible, owing to the number of inputs to the plant, $r$, being *usually smaller* than the order of the plant, $n$. Hence, as in Example 5.13, the best one can usually do is to make *some elements* of $\mathbf{e}_{ss}$ zeros, and living with the other non-zero elements, provided they are small. *In the rare case of the plant having as many inputs as the plant's order*, i.e. $n = r$, we can uniquely determine $\mathbf{K}_d$ and $\mathbf{K}_n$ as follows, to make $\mathbf{e}_{ss} = \mathbf{0}$:

$$\mathbf{K}_d = \mathbf{B}^{-1}(\mathbf{A} - \mathbf{A}_d); \quad \mathbf{K}_n = \mathbf{B}^{-1}\mathbf{F} \tag{5.81}$$

### *Example 5.14*

For the fighter aircraft of Example 5.13, let us design a controller which makes the aircraft track a target, whose state-dynamics matrix, $\mathbf{A}_d$, is the following:

$$\mathbf{A}_d = \begin{bmatrix} -2.1 & 35 & 150 \\ 0.1 & -1.1 & -21 \\ 0 & 0 & -8 \end{bmatrix} \tag{5.82}$$

The eigenvalues of $\mathbf{A}_d$ determine the poles of the target, which indicate how rapidly the desired state-vector, $\mathbf{x}_d(t)$, is changing, and are calculated as follows:

```
>>Ad = [-10.1 35 150; 0.1 -1.1 -21; 0 0 -8]; damp(Ad) <enter>

Eigenvalue     Damping     Freq. (rad/sec)
-0.7266        1.0000      0.7266
-8.0000        1.0000      8.0000
-10.4734       1.0000      10.4734
```

The target dynamics is asymptotically stable, with the pole closest to the imaginary axis being, $s = -0.7266$. This pole determines the settling time (or the speed) of the target's response. To track the target successfully, the closed-loop tracking system must be *fast enough*, i.e. the poles closest to the imaginary axis must have sufficiently small real parts, i.e. smaller than $-0.7266$. However, if the closed-loop dynamics is made *too fast* by increasing the negative real part magnitudes of the poles, there will be an *increased effect* of the noise on the system, as seen in Example 5.13. Also, recall that for an *efficient* design (i.e. smaller control effort), all the closed-loop poles must be about the same distance from the imaginary axis. Let us choose a closed-loop pole configuration as $s = -1$, $s = -1 \pm i$. The feedback gain matrix for this pole configuration was determined in Example 5.13 to be the following:

$$\mathbf{K} = [0.0006; \quad -0.0244; \quad -0.8519] \tag{5.83}$$

with the closed-loop state-dynamics matrix given by

$$\mathbf{A}_{\mathrm{CL}} = \mathbf{A} - \mathbf{BK} = \begin{bmatrix} -1.5267 & 43.3608 & 28.2818 \\ 0.2200 & -1.4000 & -32.0000 \\ -0.0089 & 0.3417 & -0.0733 \end{bmatrix} \tag{5.84}$$

The noise gain matrix, $\mathbf{K}_n$, was determined in Example 5.13 by making the largest elements of $(\mathbf{F} - \mathbf{BK}_n)$ vanish, to be the following:

$$\mathbf{K}_n = [-0.02/272; \quad -0.1/272] \tag{5.85}$$

It remains to find the feedforward gain matrix, $\mathbf{K}_\mathrm{d} = [K_{d1}; \quad K_{d2}; \quad K_{d3}]$, by considering the steady state error, $\mathbf{e}_{ss}$, given by Eq. (5.81). Note from Eq. (5.80) that, since the target is asymptotically stable, it follows that $\mathbf{x}_{dss} = \mathbf{0}$, hence $\mathbf{K}_\mathrm{d}$ will not affect the *steady state error*. However, the *transient error*, $\mathbf{e}(t)$, can be reduced by considering elements of the following matrix:

$$\mathbf{A} - \mathbf{A}_\mathrm{d} - \mathbf{BK}_\mathrm{d} = \begin{bmatrix} (8.4 + 272K_{d1}) & (15 + 272K_{d2}) & (110 + 272K_{d3}) \\ 0.12 & -0.3 & -11 \\ -14K_{d1} & -14K_{d2} & -4 - 14K_{d3} \end{bmatrix} \tag{5.86}$$

Since by changing $\mathbf{K}_\mathrm{d}$ we can only affect the first and the third rows of $(\mathbf{A} - \mathbf{A}_\mathrm{d} - \mathbf{BK}_\mathrm{d})$, let us select $\mathbf{K}_\mathrm{d}$ such that the largest elements of $(\mathbf{A} - \mathbf{A}_\mathrm{d} - \mathbf{BK}_\mathrm{d})$, which are in the first row, are minimized. By selecting $K_{d1} = -8.4/272$, $K_{d2} = -15/272$, and $K_{d3} = -110/272$, we can make the elements in the first row of $(\mathbf{A} - \mathbf{A}_\mathrm{d} - \mathbf{BK}_\mathrm{d})$ zeros, and the resulting matrix is the following:

$$\mathbf{A} - \mathbf{A}_\mathrm{d} - \mathbf{BK}_\mathrm{d} = \begin{bmatrix} 0 & 0 & 0 \\ 0.12 & -0.3 & -11 \\ 0.432 & 0.772 & 1.704 \end{bmatrix} \tag{5.87}$$

and the required feedforward gain matrix is given by

$$\mathbf{K}_\mathrm{d} = [-8.4/272; \quad -15/272; \quad -110/272] \tag{5.88}$$

The closed-loop error response to target initial condition, $\mathbf{x}_\mathrm{d}(0) = [3; 0; 0]^T$, and noise given by $\mathbf{x}_n(t) = [1 \times 10^{-5}; \quad -2 \times 10^{-6}]^T \sin(100t)$, can be obtained by solving Eq. (5.78) with $\mathbf{x}_\mathrm{d}(t)$ and $\mathbf{x}_n(t)$ as the known inputs. The noise vector, $\mathbf{x}_n(t)$, and the matrix $(\mathbf{BK_n} - \mathbf{F})$, are calculated for time upto 10 s as follows:

```
>>t = 0:0.01:10; Xn = [1e-5 -2e-6]'*sin(100*t); Bn = -[0 0; -3.5e-3
 0.004;1.03e-3 5.15e-3]; <enter>
```

The desired state-vector, $\mathbf{x}_\mathrm{d}(t)$, is obtained by solving Eq. (5.74) using the MATLAB (CST) command *initial* as follows:

```
>>sysd=ss(Ad,zeros(3,1),eye(3),zeros(3,1)); [yd,t,Xd,] = initial(sysd,
 [3 0 0]',t); <enter>
```

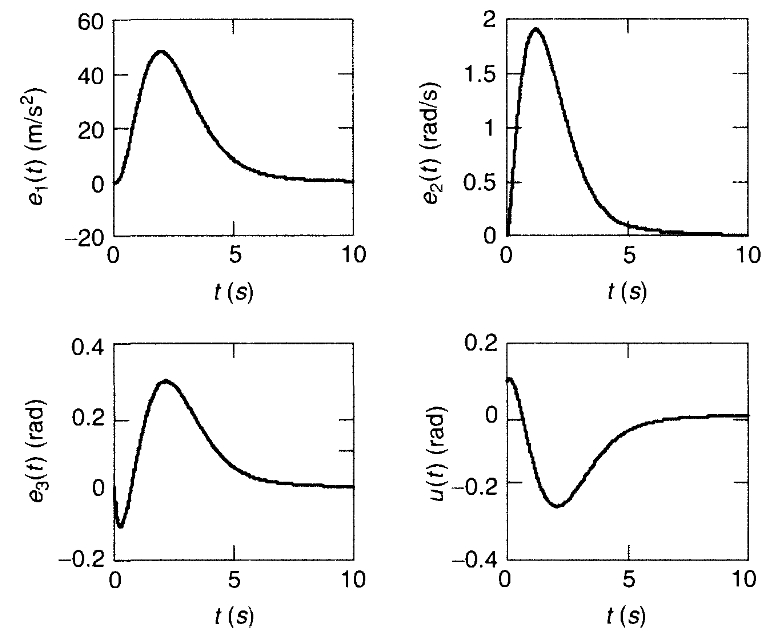


**Figure 5.15** Closed-loop error and control input response of the fighter aircraft tracking a target with initial condition $\mathbf{X}_d(0) = [3; 0; 0]^T$

The closed-loop error dynamics given by Eq. (5.78) can be written as follows:

$$\mathbf{e}^{(1)}(t) = \mathbf{A}_{CL}\mathbf{e}(t) + \mathbf{B}_{CL}\mathbf{f}(t) \tag{5.89}$$

where $\mathbf{A}_{CL} = \mathbf{A} - \mathbf{BK}$, $\mathbf{B}_{CL} = [\,(\mathbf{A}_d - \mathbf{A} + \mathbf{BK}_d);\quad (\mathbf{BK}_n - \mathbf{F})\,]$, and the input vector, $\mathbf{f}(t) = [\,\mathbf{x}_d(t)^T;\quad \mathbf{x}_n(t)^T\,]^T$, which are calculated as follows:

```
>>ACL = A-B*K; BCL = [Ad-A+B*Kd Bn]; f = [Xd Xn']; <enter>
```

Finally, using the MATLAB command *lsim*, the closed-loop error response, $\mathbf{e}(t)$, is calculated as follows:

```
>>sysCL=ss(ACL,BCL,eye(3),zeros(3,5)); e = lsim(sysCL,f,t'); <enter>
```

The error, $\mathbf{e}(t) = [\,e_1(t);\quad e_2(t);\quad e_3(t)\,]^T$, and control input, $u(t) = \mathbf{Ke}(t) - \mathbf{K}_d\mathbf{x}_d(t) - \mathbf{K}_n\mathbf{x}_n(t)$, are plotted in Figure 5.15. Note that all the error transients decay to zero in about 10 s, with a negligible influence of the noise. The settling time of error could be made smaller than 10 s, but with a larger control effort and increased vulnerability to noise.

The controller design with gain matrices given by Eqs. (5.83), (5.85), and (5.88) is the best we can do with pole-placement, because there are not enough design parameters (controller gain elements) to make the steady state error identically zero. Clearly, this is a major drawback of the pole-placement method. A better design approach with full-state feedback is the optimal control method, which will be discussed in Chapters 6 and 7.

## 5.4 Observers, Observability, and Compensators

When we designed control systems using full-state feedback in the previous section, it was assumed that we can measure and feedback all the state variables of the plant using sensors. However, it is rarely possible to measure all the state variables. Some state variables are not even physical quantities. Even in such cases where all the state variables are physical quantities, accurate sensors may not be available, or may be too expensive to construct for measuring all the state variables. Also, some state variable measurements can be so noisy that a control system based on such measurements would be unsuccessful. Hence, it is invariably required to *estimate* rather than measure the state-vector of a system. How can one estimate the state-vector, if it cannot be measured? The answer lies in *observing* the output of the system for a known input and for a finite time interval, and then reconstructing the state-vector from the record of the output. The mathematical model of the process by which a state-vector is estimated from the measured output and the known input is called an *observer* (or *state estimator*). An observer is an essential part of modern control systems. When an observer estimates the *entire* state-vector, it is called a *full-order observer*. However, the state variables that can be measured need not be estimated, and can be directly deduced from the output. An observer which estimates only the unmeasurable state variables is called the *reduced-order observer*. A reduced-order observer results in a smaller order control system, when compared to the full-order observer. However, when the measured state variables are noisy, it is preferable to use a full-order observer to reduce the effect of noise on the control system. A controller which generates the control input to the plant based on the estimated state-vector is called a *compensator*. We will consider the design of observers and compensators below.

Before we can design an observer for a plant, the plant must be *observable*. *Observability* is an important property of a system, and can be defined as the property that makes it possible to determine *any initial state*, $\mathbf{x}(t_0)$, of an *unforced* system (i.e. when the input vector, $\mathbf{u}(t)$, is *zero*) by using a *finite record* of the output, $\mathbf{y}(t)$. The term *finite record* implies that the output is recorded for only a *finite time interval* beginning at $t = t_0$. In other words, observability is a property which enables us to determine what the system was doing at some time, $t_0$, after measuring its output for a finite time interval beginning at that time. The term *any initial state* is significant in the definition of observability; it may be possible to determine *some initial states* by recording the output, and the system may yet be *unobservable*. Clearly, observability requires that *all* the state variables must contribute to the output of the system, otherwise we cannot reconstruct *all possible* combinations of state variables (i.e. *any* initial state-vector) by measuring the output. The relationship between observability and the output is thus the *dual* of that between controllability and the input. For a system to be controllable, all the state variables must *be affected* by the input; for a system to be observable, all the state variables must *affect* the output. If there are some state variables which *do not* contribute to the output, then the system is *unobservable*. One way of determining observability is by looking at the decoupled state-equations, and the corresponding output equation of a system.

*Example 5.15*

Consider a system with the following scalar state-equations:

$$\begin{aligned} x_1^{(1)}(t) &= 2x_1(t) + 3u(t) \\ x_2^{(1)}(t) &= -x_2(t) \\ x_3^{(1)}(t) &= 5x_3(t) - u(t) \end{aligned} \tag{5.90}$$

The scalar output equations of the system are the following:

$$\begin{aligned} y_1(t) &= x_1(t) \\ y_2(t) &= 2x_2(t) + x_1(t) + u(t) \end{aligned} \tag{5.91}$$

Equation (5.90) implies that the state variable, $x_3(t)$, is decoupled from the other two state variables, $x_1(t)$ and $x_2(t)$. Also, $x_3(t)$ does not affect either of the two output variables, $y_1(t)$ and $y_2(t)$. Since the state variable $x_3(t)$, does not contribute to the output vector, $\mathbf{y}(t) = [\,y_1(t);\quad y_2(t)\,]^T$, either *directly* or *indirectly* through $x_1(t)$ and $x_2(t)$, it follows that the system is *unobservable.*

As it is not always possible to decouple the state-equations, we need another way of testing for observability. Similar to the algebraic controllability test theorem, there is an *algebraic observability test theorem* for linear, time-invariant systems stated as follows.

**Theorem**
*The unforced system,* $\mathbf{x}^{(1)}(t) = \mathbf{A}\mathbf{x}(t)$, $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)$, *is observable if and only if the rank of the* observability test matrix, $\mathbf{N} = [\,\mathbf{C}^{\mathrm{T}};\quad \mathbf{A}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}};\quad (\mathbf{A}^{\mathrm{T}})^2\mathbf{C}^{\mathrm{T}};\quad \ldots;\quad (\mathbf{A}^{\mathrm{T}})^{n-1}\mathbf{C}^{\mathrm{T}}\,]$, *is equal to n, the order of the system.*

The proof of this theorem, given in Friedland [2], follows from the definition of observability, and recalling from Chapter 4 that the output of an unforced (homogeneous) linear, time-invariant system is given by $\mathbf{y}(t) = \mathbf{C}\exp\{\mathbf{A}(t - t_0)\}\mathbf{x}(t_0)$, where $\mathbf{x}(t_0)$ is the initial state-vector.

*Example 5.16*

Let us apply the observability test theorem to the system of Example 5.15. The state coefficient matrices, **A** and **C**, are the following:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 5 \end{bmatrix}; \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \end{bmatrix} \tag{5.92}$$

The observability test matrix, **N**, is constructed as follows:

$$\mathbf{A}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 5 \end{bmatrix}; \quad \mathbf{C}^{T} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}; \quad \mathbf{A}^{\mathrm{T}}\mathbf{C}^{\mathrm{T}} = \begin{bmatrix} 1 & 2 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}$$

$$(\mathbf{A}^T)^2\mathbf{C}^T = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (5.93)$$

or

$$\mathbf{N} = \begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 2 \\ 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.94)$$

The entire third row of **N** consists of zeros; hence it is impossible tò form a (3 × 3) sized, non-zero determinant out of the rows and columns of **N**. Thus rank (**N**) < 3 for this third order system, therefore the system is *unobservable*.

Rather than forming the observability test matrix, **N**, by hand as in Example 5.16, which could be a tedious process for large order systems, we can use the MATLAB (CST) command *ctrb*, noting that **N** is the controllability test matrix in which **A** is replaced by $\mathbf{A}^T$ and **B** is replaced by $\mathbf{C}^T$. Thus, the command

```
>>N = ctrb(A',C') <enter>
```

will give us the observability test matrix.

The reasons for unobservability of a system are pretty much the same as those for uncontrollability, namely the use of superfluous state variables in state-space model, pole-zero cancellation in the system's transfer matrix, too much symmetry, and physical unobservability (i.e. selection of an output vector which is physically unaffected by one or more state variables). If the sub-systems which cause unobservability are *stable*, we can safely ignore those state variables that do not contribute to the output, and design an observer based on the remaining state variables (which would constitute an observable *sub-system*). Thus a stable, unobservable system is said to be *detectable*. If an unobservable sub-system is *unstable*, then the entire system is said to be *undetectable*, because an observer cannot be designed by ignoring the unobservable (and unstable) sub-system. In Example 5.15, the unobservable sub-system corresponding to the decoupled state variable, $x_3(t)$, is unstable (it has a pole at $s = 5$). Hence, the system of Example 5.15 is *undetectable*.

### 5.4.1 Pole-placement design of full-order observers and compensators

A full-order observer estimates the entire state-vector of a plant, based on the measured output and a known input. If the plant for which the observer is required is linear, the observer's dynamics would also be described by linear state-equations. Consider a noise-free, linear, time-invariant plant described by the following state and output equations:

$$\mathbf{x}^{(1)}(t) = \mathbf{Ax}(t) + \mathbf{Bu}(t) \quad (5.95)$$

$$\mathbf{y}(t) = \mathbf{Cx}(t) + \mathbf{Du}(t) \quad (5.96)$$

The linear, time-invariant state-equation which describes the dynamics of a full-order observer can be expressed as follows:

$$\mathbf{x}_o^{(1)}(t) = \mathbf{A}_o\mathbf{x}_o(t) + \mathbf{B}_o\mathbf{u}(t) + \mathbf{L}\mathbf{y}(t) \tag{5.97}$$

where $\mathbf{x}_o(t)$ is the *estimated state-vector*, $\mathbf{u}(t)$ is the input vector, $\mathbf{y}(t)$ is the output vector, $\mathbf{A}_o$, $\mathbf{B}_o$ are the state-dynamics and control coefficient matrices of the observer, and $\mathbf{L}$ is the *observer gain matrix*. The matrices $\mathbf{A}_o$, $\mathbf{B}_o$, and $\mathbf{L}$ must be selected in a design process such that the *estimation error*, $\mathbf{e}_o(t) = \mathbf{x}(t) - \mathbf{x}_o(t)$, is brought to zero in the steady state. On subtracting Eq. (5.97) from Eq. (5.95), we get the following *error dynamics* state-equation:

$$\mathbf{e}_o^{(1)}(t) = \mathbf{A}_o\mathbf{e}_o(t) + (\mathbf{A} - \mathbf{A}_o)\mathbf{x}(t) + (\mathbf{B} - \mathbf{B}_o)\mathbf{u}(t) - \mathbf{L}\mathbf{y}(t) \tag{5.98}$$

Substitution of Eq. (5.96) into Eq. (5.98) yields

$$\mathbf{e}_o^{(1)}(t) = \mathbf{A}_o\mathbf{e}_o(t) + (\mathbf{A} - \mathbf{A}_o)\mathbf{x}(t) + (\mathbf{B} - \mathbf{B}_o)\mathbf{u}(t) - \mathbf{L}[\mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)] \tag{5.99}$$

or

$$\mathbf{e}_o^{(1)}(t) = \mathbf{A}_o\mathbf{e}_o(t) + (\mathbf{A} - \mathbf{A}_o - \mathbf{L}\mathbf{C})\mathbf{x}(t) + (\mathbf{B} - \mathbf{B}_o - \mathbf{L}\mathbf{D})\mathbf{u}(t) \tag{5.100}$$

From Eq. (5.100), it is clear that estimation error, $\mathbf{e}_o(t)$, will go to zero in the steady state irrespective of $\mathbf{x}(t)$ and $\mathbf{u}(t)$, if all the *eigenvalues* of $\mathbf{A}_o$ are in the *left-half plane*, and the coefficient matrices of $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are *zeros*, i.e. $(\mathbf{A} - \mathbf{A}_o - \mathbf{L}\mathbf{C}) = \mathbf{0}$, $(\mathbf{B} - \mathbf{B}_o - \mathbf{L}\mathbf{D}) = \mathbf{0}$. The latter requirement leads to the following expressions for $\mathbf{A}_o$ and $\mathbf{B}_o$:

$$\mathbf{A}_o = \mathbf{A} - \mathbf{L}\mathbf{C}; \quad \mathbf{B}_o = \mathbf{B} - \mathbf{L}\mathbf{D} \tag{5.101}$$

The error dynamics state-equation is thus the following:

$$\mathbf{e}_o^{(1)}(t) = (\mathbf{A} - \mathbf{L}\mathbf{C})\mathbf{e}_o(t) \tag{5.102}$$

The observer gain matrix, $\mathbf{L}$, must be selected to place all the eigenvalues of $\mathbf{A}_o$ (which are also the poles of the observer) at desired locations in the left-half plane, which implies that the estimation error dynamics given by Eq. (5.102) is *asymptotically stable* (i.e. $\mathbf{e}_o(t) \to \mathbf{0}$ as $t \to \infty$). On substituting Eq. (5.101) into Eq. (5.97), we can write the full-order observer's state-equation as follows:

$$\mathbf{x}_o^{(1)}(t) = (\mathbf{A} - \mathbf{L}\mathbf{C})\mathbf{x}_o(t) + (\mathbf{B} - \mathbf{L}\mathbf{D})\mathbf{u}(t) + \mathbf{L}\mathbf{y}(t) = \mathbf{A}\mathbf{x}_o(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{L}[\mathbf{y}(t) - \mathbf{C}\mathbf{x}_o(t) - \mathbf{D}\mathbf{u}(t)] \tag{5.103}$$

Note that Eq. (5.103) approaches Eq. (5.95) in the steady state if $\mathbf{x}_o(t) \to \mathbf{x}(t)$ as $t \to \infty$. Hence, the observer *mirrors* the plant dynamics if the error dynamics is asymptotically stable. The term $[\mathbf{y}(t) - \mathbf{C}\mathbf{x}_o(t) - \mathbf{D}\mathbf{u}(t)]$ in Eq. (5.103) is called the *residual*, and can be expressed as follows:

$$[\mathbf{y}(t) - \mathbf{C}\mathbf{x}_o(t) - \mathbf{D}\mathbf{u}(t)] = \mathbf{C}\mathbf{x}(t) - \mathbf{C}\mathbf{x}_o(t) = \mathbf{C}\mathbf{e}_o(t) \tag{5.104}$$

From Eq. (5.104), it is clear that the residual is also forced to zero in the steady-state if the error dynamics is asymptotically stable.

The observer design process merely consists of selecting **L** by pole-placement of the observer. For single-output plants, the pole-placement of the observer is carried out in a manner similar to the pole-placement of *regulators* for *single-input* plants (see Section 5.3.1). For a plant with the characteristic polynomial written as $|s\mathbf{I} - \mathbf{A}| = s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0$, it can be shown by steps similar to Section 5.3.1 that the observer gain matrix, **L**, which places the observer's poles such that the observer's characteristic polynomial is $|s\mathbf{I} - \mathbf{A}_o| = s^n + \beta_{n-1}s^{n-1} + \cdots + \beta_1 s + \beta_0$ is given by

$$\mathbf{L} = [(\boldsymbol{\beta} - \mathbf{a})\mathbf{N}'\mathbf{N}^{-1}]^T \qquad (5.105)$$

where $\boldsymbol{\beta} = [\,\beta_{n-1};\ \ \beta_{n-2};\ \ \ldots;\ \ \beta_1;\ \ \beta_0\,]$, $\mathbf{a} = [\,a_{n-1};\ \ a_{n-2};\ \ \ldots;\ \ a_1;\ \ a_0\,]$, **N** is the *observability test matrix* of the plant described by Eqs. (5.95) and (5.96), and **N′** is the observability test matrix of the plant when it is in the *observer companion form*. Since for single-input, single-output systems, the observer companion form can be obtained from the controller companion form merely by substituting **A** by $\mathbf{A}^\mathrm{T}$, **B** by $\mathbf{C}^\mathrm{T}$, and **C** by $\mathbf{B}^\mathrm{T}$ (see Chapter 3), you can easily show that $\mathbf{N}' = \mathbf{P}'$, where **P′** is the *controllability test matrix* of the plant when it is in the *controller companion form*. Thus, we can write

$$\mathbf{L} = [(\boldsymbol{\beta} - \mathbf{a})\mathbf{P}'\mathbf{N}^{-1}]^T \qquad (5.106)$$

Recall that **P′** is an upper triangular matrix, given by Eq. (5.51).

### *Example 5.17*

Let us try to design a full-order observer for the inverted pendulum on a moving cart (Example 5.9). A state-space representation of the plant is given by Eq. (5.53), with the numerical values of **A** and **B** given by Eq. (5.54). For this single-input, two-output plant, let us try to design an observer using *only one of the outputs*. If we select the single output to be $y(t) = \theta(t)$, the angular position of the inverted pendulum, the matrices **C** and **D** are the following:

$$\mathbf{C} = [\,1;\ \ 0;\ \ 0;\ \ 0\,];\quad \mathbf{D} = 0 \qquad (5.107)$$

The first thing to do is to check whether the plant is *observable* with this choice of the output. We do so by the following MATLAB command:

```
>>N = (ctrb(A',C'); rank(N) <enter>

ans =
         2
```

Since the rank of the observability test matrix, **N**, is 2, i.e. *less than 4*, the order of the plant, the plant is *unobservable* with the angular position of the pendulum as the only output. Hence, we cannot design an observer using $y(t) = \theta(t)$. If we choose $y(t) = x(t)$, the cart's displacement, then the output coefficient matrices are as follows:

$$\mathbf{C} = [\,0;\ \ 1;\ \ 0;\ \ 0\,];\quad \mathbf{D} = 0 \qquad (5.108)$$

On forming the observability test matrix, **N**, with this choice of output, and checking its rank we get

```
>>N = (ctrb(A',C')); rank(N) <enter>

ans =
       4
```

Since now rank (**N**) = 4, the order of the plant, the plant is observable with $y(t) = x(t)$, and an observer can be designed based on this choice of the output. Let us place the observer poles at $s = -10 \pm 10i$, and $s = -20 \pm 20i$. Then the observer's characteristic polynomial coefficients vector, $\boldsymbol{\beta}$, is calculated as follows:

```
>>v = [-10-10i -10+10i -20-20i -20+20i]'; p = poly(v); beta = p(2:5)
  <enter>

beta =
    60   1800   24000   160000
```

The plant's characteristic polynomial coefficient vector, **a**, is calculated as follows:

```
>>p = poly(A); a = p(2:5) <enter>

a =
                  0 -10.7800 0 0
```

and the matrix **P′** is evaluated using Eq. (5.51) as follows:

```
>>Pdash = [1 -a(1:3); 0 1 -a(1:2); 0 0 1 -a(1); 0 0 0 1] <enter>

Pdash =
        1.0000   0        10.7800   0
        0        1.000    0         10.7800
        0        0        1.0000    0
        0        0        0         1.0000
```

Finally, the observer gain matrix, **L**, is calculated using Eq. (5.106) as follows:

```
>>format long e; L = ((beta-a)*Pdash*inv(N))' <enter>

L =
 -2.514979591836735e+004
  6.000000000000000e+001
 -1.831838861224490e+005
  1.810780000000000e+003
```

Note that we have printed out **L** in the *long format*, since we need to store it for later calculations. Let us check whether the observer poles have been placed at desired locations, by calculating the eigenvalues of $\mathbf{A}_o = (\mathbf{A} - \mathbf{LC})$ as follows:

```
>>Ao = A-L*C; eig(Ao) <enter>

ans =
 -20.0000+20.0000i
 -20.0000-20.0000i
 -10.0000+10.0000i
 -10.0000-10.0000i
```

Hence, observer pole-placement has been accurately achieved.

Example 5.17 illustrates the ease by which single-output observers can be designed. However, it is impossible to design single-output observers for those plants which are *unobservable* with *any single* output. When *multi-output* observers are required, generally there are more design parameters (i.e. elements in the observer gain matrix, **L**) than the observer poles, hence all of these parameters cannot be determined by pole-placement alone. As in the design of regulators for multi-input plants (Section 5.3.2), additional conditions are required to be satisfied by multi-output observers, apart from pole-placement, to determine the observer gain matrix. These additional conditions are hard to come by, and thus pole-placement is not a good method of designing multi-output observers. A better design procedure in such cases is the *Kalman filter* approach of Chapter 7.

MATLAB's Control System Toolbox (CST) provides the command *estim* for constructing a state-space model, *syso*, of the observer with the observer gain matrix, **L**, and a state-space model, *sysp*, of the plant, with state coefficient matrices **A, B, C, D**, as follows:

```
>>sysp=ss[A,B,C,D]; sysp = estim(syso,L) <enter>
```

The input to the observer thus formed is the plant's output, $\mathbf{y}(t)$, while output vector of the observer is $[\{\mathbf{C}\mathbf{x}_o(t)\}^T; \quad \mathbf{x}_o(t)^T\,]^T$, where $\mathbf{x}_o(t)$ is the estimated state-vector.

Observers (also known as *estimators*) by themselves are very useful in estimating the plant dynamics from a limited number of outputs, and are employed in *parameter estimation, fault detection*, and other similar applications. The utility of an observer in a control system lies in feeding the estimated state-vector to a controller for generating input signals for the plant. The controllers which generate input signals for the plant based on the estimated state-vector (rather than the actual, fed back state-vector) are called *compensators*. However, design of compensators involves a dilemma. The estimated state-vector is obtained from an observer, which treats the plant's input vector as a *known quantity*, while the compensator is *yet to* generate the input vector based on the estimated state-vector. It is like the classic chicken and egg problem, since we do not know which came first: the control input on which the estimated state-vector is based, or the estimated state-vector on which the input is based! A practical way of breaking this vicious circle is the *separation principle*, which states that if we design an observer (assuming known input vector), and a compensator (assuming known estimated state-vector) *separately*, and then combine the two, we will end up with a control system *that works*. The separation principle thus allows us to design the observer and the controller *independently of each* *other*. The resulting control system can be a *regulator* or a *tracking system*, depending on the desired state-vector being *zero* or *non-zero*, respectively.

Let us consider a tracking system (i.e. a control system with a non-zero desired state-vector) based on a noise-free plant described by Eqs. (5.95) and (5.96), for which a full-order observer, given by Eq. (5.103) has been designed. Then a compensator can be designed to generate the input vector for the plant according to the following control-law:

$$\mathbf{u}(t) = \mathbf{K}[\mathbf{x}_d(t) - \mathbf{x}_o(t)] - \mathbf{K}_d\mathbf{x}_d(t) \tag{5.109}$$

where $\mathbf{x}_o(t)$ is the *estimated state-vector*, $\mathbf{x}_d(t)$ is the *desired state-vector*, $\mathbf{K}$ is the *feedback gain matrix*, and $\mathbf{K}_d$ is the *feedforward gain matrix*. On substituting Eq. (5.109) into Eq. (5.103), the observer state-equation becomes

$$\mathbf{x}_o^{(1)}(t) = (\mathbf{A} - \mathbf{LC} - \mathbf{BK} + \mathbf{LDK})\mathbf{x}_o(t) + (\mathbf{B} - \mathbf{LD})(\mathbf{K} - \mathbf{K}_d)\mathbf{x}_d(t) + \mathbf{Ly}(t) \tag{5.110}$$

On substituting the output equation, Eq. (5.96), into Eq. (5.110), and again substituting Eq. (5.109), we get the following state-equation for the compensator:

$$\mathbf{x}_o^{(1)}(t) = (\mathbf{A} - \mathbf{LC} - \mathbf{BK})\mathbf{x}_o(t) + \mathbf{B}(\mathbf{K} - \mathbf{K}_d)\mathbf{x}_d(t) + \mathbf{LCx}(t) \tag{5.111}$$

The plant's state-equation, Eq. (5.95), when the input is given by Eq. (5.109), becomes the following:

$$\mathbf{x}^{(1)}(t) = \mathbf{Ax}(t) - \mathbf{BKx}_o(t) + \mathbf{B}(\mathbf{K} - \mathbf{K}_d)\mathbf{x}_d(t) \tag{5.112}$$

Equations. (5.111) and (5.112) are the state-equations of the closed-loop system, and can be expressed as follows:

$$\begin{bmatrix} \mathbf{x}^{(1)}(t) \\ \mathbf{x}_o^{(1)}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{LC} & (\mathbf{A} - \mathbf{LC} - \mathbf{BK}) \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}_o(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B}(\mathbf{K} - \mathbf{K}_d) \\ \mathbf{B}(\mathbf{K} - \mathbf{K}_d) \end{bmatrix} \mathbf{x}_d(t) \tag{5.113}$$

The closed-loop tracking system is thus of order $2n$, where $n$ is the order of the plant. The input to the closed-loop system is the desired state-vector, $\mathbf{x}_d(t)$. A schematic diagram of the tracking system is shown in Figure 5.16. Note that this control system is essentially

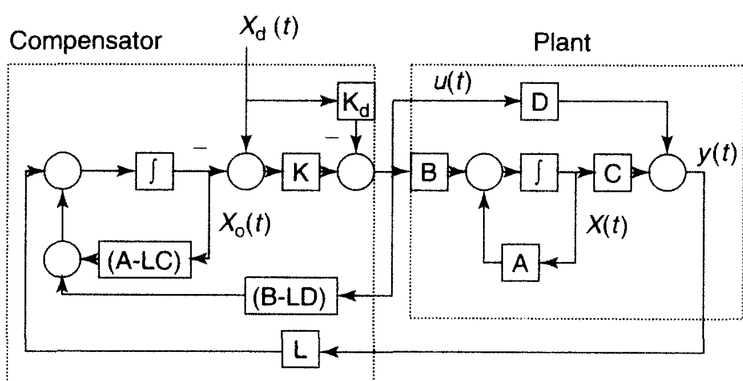


**Figure 5.16** Closed-loop tracking system with a full-order compensator

based on the feedback of the output vector, $\mathbf{y}(t)$, to the compensator, which generates the input vector, $\mathbf{u}(t)$, for the plant.

To obtain the state-equation for the estimation error, $\mathbf{e}_o(t) = \mathbf{x}(t) - \mathbf{x}_o(t)$, let us write Eq. (5.112) as follows:

$$\mathbf{x}^{(1)}(t) = (\mathbf{A} - \mathbf{BK})\mathbf{x}(t) + \mathbf{BKe}_o(t) + \mathbf{B}(\mathbf{K} - \mathbf{K}_d)\mathbf{x}_d(t) \tag{5.114}$$

On subtracting Eq. (5.111) from Eq. (5.114) we get

$$\mathbf{e}_o^{(1)}(t) = (\mathbf{A} - \mathbf{LC})\mathbf{e}_o(t) \tag{5.115}$$

which is the same as Eq. (5.102). The state-equation for the tracking error, $\mathbf{e}(t) = \mathbf{x}_d(t) - \mathbf{x}(t)$, is obtained by subtracting Eq. (5.114) from Eq. (5.74), which results in

$$\mathbf{e}^{(1)}(t) = (\mathbf{A} - \mathbf{BK})\mathbf{e}(t) + (\mathbf{A}_d - \mathbf{A} + \mathbf{BK}_d)\mathbf{x}_d(t) - \mathbf{BKe}_o(t) \tag{5.116}$$

The tracking system's error dynamics is thus represented by Eqs. (5.115) and (5.116), which can be expressed together as follows:

$$\begin{bmatrix} \mathbf{e}^{(1)}(t) \\ \mathbf{e}_o^{(1)}(t) \end{bmatrix} = \begin{bmatrix} (\mathbf{A} - \mathbf{BK}) & -\mathbf{BK} \\ \mathbf{0} & (\mathbf{A} - \mathbf{LC}) \end{bmatrix} \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{e}_o(t) \end{bmatrix} + \begin{bmatrix} (\mathbf{A}_d - \mathbf{A} + \mathbf{BK}_d) \\ \mathbf{0} \end{bmatrix} \mathbf{x}_d(t) \tag{5.117}$$

Note that Eq. (5.117) represents the closed-loop tracking system in a *decoupled* state-space form. The closed-loop poles must be the eigenvalues of the following closed-loop state-dynamics matrix, $\mathbf{A}_{CL}$:

$$\mathbf{A}_{CL} = \begin{bmatrix} (\mathbf{A} - \mathbf{BK}) & \mathbf{0} \\ \mathbf{0} & (\mathbf{A} - \mathbf{LC}) \end{bmatrix} \tag{5.118}$$

Equation (5.117) implies that the closed-loop poles are the eigenvalues of $\mathbf{A}_{CL}$, i.e. the roots of the characteristic equation $|s\mathbf{I} - \mathbf{A}_{CL}| = 0$, which can be written as $|[s\mathbf{I} - (\mathbf{A} - \mathbf{BK})][s\mathbf{I} - (\mathbf{A} - \mathbf{LC})]| = 0$, resulting in $|s\mathbf{I} - (\mathbf{A} - \mathbf{BK})| = 0$ and $|s\mathbf{I} - (\mathbf{A} - \mathbf{LC})| = 0$. Hence, the *closed-loop poles* are the *eigenvalues of* $(\mathbf{A} - \mathbf{BK})$ and *eigenvalues of* $(\mathbf{A} - \mathbf{LC})$, which are also the *poles of* the full-state feedback *regulator* and the *observer*, respectively. Note from Eq. (5.117) that for the estimation error, $\mathbf{e}_o(t)$, to go to zero in the steady state, all the eigenvalues of $(\mathbf{A} - \mathbf{LC})$ must be in the left-half plane. Also, for the tracking error, $\mathbf{e}(t)$, to go to zero in the steady state, irrespective of the desired state-vector, $\mathbf{x}_d(t)$, all the eigenvalues of $(\mathbf{A} - \mathbf{BK})$ must be in the left-half plane, and the coefficient matrix multiplying $\mathbf{x}_d(t)$ must be zero, $(\mathbf{A}_d - \mathbf{A} + \mathbf{BK}_d) = \mathbf{0}$. Recall from Section 5.3 that $(\mathbf{A} - \mathbf{BK})$ is the state-dynamics matrix of the *full-state feedback regulator*, and from Eq. (5.103) that $(\mathbf{A} - \mathbf{LC})$ is the state-dynamics matrix of the *full-order observer*. Hence, the compensator design process consists of *separately* deriving the feedback gain matrices $\mathbf{L}$ and $\mathbf{K}$, by pole-placement of the observer and the full-state feedback regulator, respectively, and selecting $\mathbf{K}_d$ to satisfy $(\mathbf{A}_d - \mathbf{A} + \mathbf{BK}_d) = \mathbf{0}$. Usually, it is impossible to satisfy $(\mathbf{A}_d - \mathbf{A} + \mathbf{BK}_d) = \mathbf{0}$ by selecting the feedforward gain matrix, $\mathbf{K}_d$. Alternatively, it may be possible to satisfy $(\mathbf{A}_d - \mathbf{A} + \mathbf{BK}_d)\mathbf{x}_d(t) = \mathbf{0}$ when some elements of $\mathbf{x}_d(t)$ are zeros. Hence, the steady state tracking error can generally be reduced to zero only for some values of the desired state-vector. In the above steps, we have assumed that the desired state-vector, $\mathbf{x}_d(t)$, is available for measurement. In many cases, it is possible to measure only a desired output, $\mathbf{y}_d(t) = \mathbf{C}_d\mathbf{x}_d(t)$, rather than $\mathbf{x}_d(t)$ itself. In such cases, an observer can be designed to estimate $\mathbf{x}_d(t)$ based on the measurement of the desired output. It is left to you as an exercise to derive the state-equations for the compensator when $\mathbf{x}_d(t)$ is not measurable.

### *Example 5.18*

Let us design a compensator for the inverted pendulum on a moving cart (Example 5.9), when it is desired to move the cart by 1 m, while not letting the pendulum fall. Such a tracking system is representative of a *robot*, which is bringing to you an inverted champagne bottle precariously balanced on a finger! The plant is clearly unstable (as seen in Example 5.9). The task of the compensator is to stabilize the inverted pendulum, while moving the cart by the desired displacement. The desired state-vector is thus a constant, consisting of the desired angular position of the inverted pendulum, $\theta_d(t) = 0$, desired cart displacement, $x_d(t) = 1$ m, desired angular velocity of the pendulum, $\theta_d^{(1)}(t) = 0$, and desired cart velocity, $x_d^{(1)}(t) = 0$. Hence, $\mathbf{x}_d(t) = [0; 1; 0; 0]^T$. Since $\mathbf{x}_d(t)$ is constant, it implies that $\mathbf{x}_d^{(1)}(t) = \mathbf{0}$, and from Eq. (5.74), $\mathbf{A}_d = \mathbf{0}$. By the separation principle, we can design a tracking system *assuming full-state feedback*, and then combine it with a full-order observer, which estimates the plant's state-vector. A full-state feedback regulator has already been designed for this plant in Example 5.11, which places the eigenvalues of the regulator state-dynamics matrix, $(\mathbf{A} - \mathbf{BK})$, at $s = -7.853 \pm 3.2528i$, and $s = -7.853 \pm 7.853i$ using the following feedback gain matrix:

$$\mathbf{K} = [\,-1362.364050360232;\quad -909.3160795202226;\quad -344.8741667548096;\quad -313.4621667548089\,] \tag{5.119}$$

We have also designed a full-order observer for this plant using the cart displacement, $x(t)$, as the output in Example 5.17. The observer poles, i.e. the eigenvalues of $(\mathbf{A} - \mathbf{LC})$, were selected to be at $s = -10 \pm 10i$, and $s = -20 \pm 20i$, and the observer gain matrix which achieved this observer pole configuration was obtained to be the following:

$$\mathbf{L} = [\,-25\,149.79591836735;\quad 60.0;\quad -183\,183.8861224490;\quad 1810.780\,]^T \tag{5.120}$$

The separation principle allows us to combine the separately designed observer and regulator into a compensator. However, it remains for us to determine the feedforward gain matrix, $\mathbf{K}_d$. The design requirement of zero tracking error in the steady state is satisfied if $(\mathbf{A}_d - \mathbf{A} + \mathbf{BK}_d)\mathbf{x}_d(t) = \mathbf{0}$ in Eq. (5.117). The elements of

$\mathbf{K}_d = [\, K_{d1};\quad K_{d2};\quad K_{d3};\quad K_{d4}\,]$ are thus determined as follows:

$$(\mathbf{A}_d - \mathbf{A} + \mathbf{B}\mathbf{K}_d)\mathbf{x}_d(t) = \begin{bmatrix} 0 \\ 0 \\ K_{d2} \\ K_{d2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.121}$$

Equation (5.121) is exactly satisfied by selecting $K_{d2} = 0$. What about the other elements of $\mathbf{K}_d$? There are no conditions placed on the other elements of $\mathbf{K}_d$, and thus we can *arbitrarily* take them to be zeros. Therefore, by choosing $\mathbf{K}_d = \mathbf{0}$, we are able to meet the zero tracking error requirement in the steady state. On substituting the designed values of the gain matrices, $\mathbf{K}$, $\mathbf{L}$, and $\mathbf{K}_d$ into Eq. (5.113), we can get the closed-loop state-equations for the tracking system in terms of the plant's state-vector, $\mathbf{x}(t)$, and the estimated state-vector, $\mathbf{x}_o(t)$, and then solve them to get the closed-loop response. This is done using MATLAB as follows:

```
>>K=[-1362.364050360232 -909.3160795202226 -344.8741667548096 -313.46216675
  48089]; <enter>

>>L=[-25149.79591836735 60.0 -183183.8861224490 1810.780]'; Kd=zeros(1,4);
  <enter>

>>ACL = [A -B*K; L*C (A-L*C-B*K)]; BCL = [B*(K-Kd); B*(K-Kd)];<enter>
```

Let us confirm that the eigenvalues of $\mathbf{A}_{CL}$ are the poles of the regulator designed in Example 5.11 and the observer designed in Example 5.17 as follows:

```
>>eig(ACL) <enter>

ans =
-20.0000+20.0000i
-20.0000-20.0000i
-10.0000+10.0000i
-10.0000-10.0000i
-7.8530+7.8530i
-7.8530-7.8530i
-7.8530+3.2528i
-7.8530-3.2528i
```

which indeed they are. Finally, the closed-loop response to the desired state-vector is calculated as follows:

```
>>t = 0:1.0753e-2:1.2; n=size(t,2); for i=1:n; Xd(i,:) = [0 1 0 0]; end
  <enter>

>>sysCL=ss(ACL, BCL,[C zeros(1,4)],zeros(1,4)); [y,t,X] = lsim(sysCL,Xd,t');
  <enter>
```

The closed-loop cart's displacement, $x(t)$, and pendulum's angular position, $\theta(t)$, are plotted in Figure 5.17, as follows:

```
>>plot(t,X(:,1:2)) <enter>
```

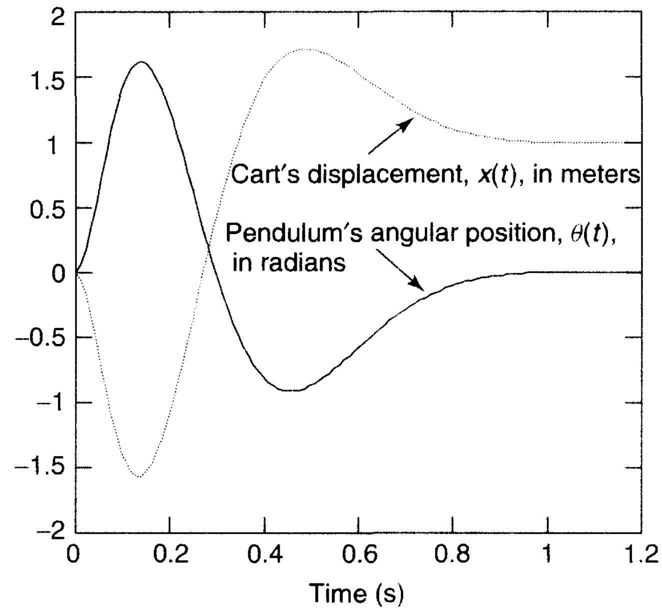


**Figure 5.17** Response of the compensator based tracking system for inverted- pendulum on a moving cart, with desired angular position, $\theta_d(t) = 0$, and desired cart's displacement, $x_d(t) = 1$ m, when the regulator poles are $s = -7.853 \pm 3.2528i$, and $s = -7.853 \pm 7.853i$

The closed-loop transient response for $x(t)$ and $\theta(t)$ is seen in Figure 5.17 to settle to their respective desired values in about 1 s, with maximum overshoots of 1.65 m and 1.57 rad., respectively. However, an overshoot of 1.57 rad. corresponds to 90°, which implies that the pendulum *has been allowed to fall* and then brought back up to the inverted position, $\theta(t) = 0°$. If the inverted pendulum represents a drink being brought to you by a robot (approximated by the moving cart), clearly this compensator design would be unacceptable, and it will be necessary to reduce the maximum overshoot to an angle less than 90° by suitably modifying the closed-loop poles. Recall from Example 3.3 that the linearized state-space model of the system given by Eq. (5.53) is *invalid* when the pendulum sways by a large angle, $\theta(t)$, and the results plotted in Figure 5.17 are thus *inaccurate*. Hence, the regulator design that was adequate for stabilizing the plant in the presence of a *small* initial disturbance in cart displacement, is *unsatisfactory* for moving the cart by a *large* displacement. Note that the location of the regulator poles, i.e. the eigenvalues of $(\mathbf{A} - \mathbf{BK})$, governs the closed-loop response of the plant's state-vector, $\mathbf{x}(t)$. By moving the regulator poles closer to the imaginary axis, it would be possible to reduce the maximum overshoot *at the cost of increased settling time*. Let us select the new regulator poles as $s = -0.7853 \pm 3.25328i$ and $s = -0.7853 \pm 0.7853i$. The new feedback gain matrix, $\mathbf{K}$, is calculated as follows:

```
>>v=[-0.7853+3.25328i -0.7853-3.25328i -0.7853+0.7853i -0.7853-0.7853i]';
  K=place(A,B,v)
place: ndigits= 16

K =
  -27.0904   -1.4097   -5.1339   -1.9927
```

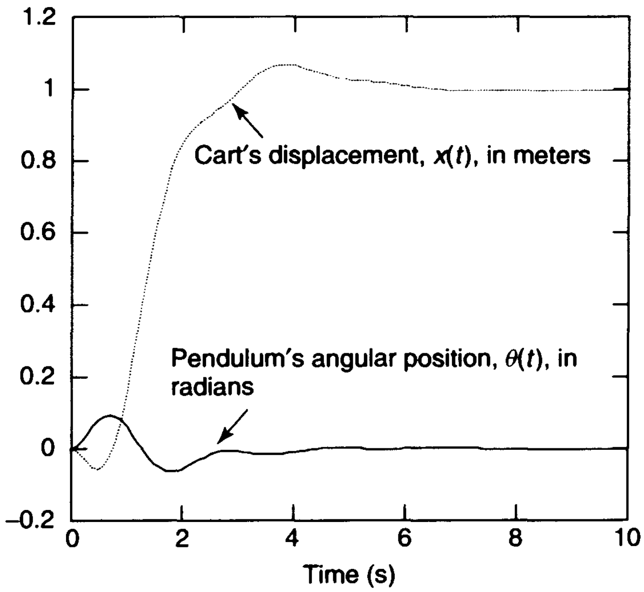


**Figure 5.18** Response of the compensator based tracking system for inverted- pendulum on a moving cart, with desired angular position, $\theta_d(t) = 0$, and desired cart's displacement, $x_d(t) = 1$ m, when regulator poles are $s = -0.7853 \pm 3.25328i$ and $s = -0.7853 \pm 0.7853i$

and the new closed-loop response is plotted in Figure 5.18, which shows that the maximum overshoots have been reduced to less than 1.1 m and 0.1 rad. (5.7°) for $x(t)$ and $\theta(t)$, respectively, but the settling time is increased to about 7 s. Since the pendulum now sways by small angles, the linearized model of Eq. (5.53) is valid, and the compensator design is acceptable. However, the robot now takes 7 seconds in bringing your drink placed 1 m away! You may further refine the design by experimenting with the regulator pole locations.

Let us see how well the compensator estimates the state-vector by looking at the estimation error vector, $\mathbf{e}_o(t) = \mathbf{x}(t) - \mathbf{x}_o(t)$. The elements of the estimation error vector, $e_{o1}(t) = \theta_d(t) - \theta(t)$, $e_{o2}(t) = x_d(t) - x(t)$, $e_{o3}(t) = \theta_d^{(1)}(t) - \theta^{(1)}(t)$, and $e_{o4}(t) = x_d^{(1)}(t) - x^{(1)}(t)$ are plotted in Figure 5.19 as follows:

```
>>plot(t,X(:,1)-X(:,5),t,X(:,2)-X(:,6),t,X(:,3)-X(:,7),t,X(:,4)-X(:,8))
  <enter>
```

Figure 5.19 shows that the largest estimation error magnitude is about $1.5 \times 10^{-9}$ rad/s for estimating the pendulum's angular velocity, $\theta^{(1)}(t)$, and about $5 \times 10^{-10}$ rad. for estimating the pendulum's angular position, $\theta(t)$. Since the observer is based on the measurement of the cart's displacement, $x(t)$, the estimation error magnitudes of $x(t)$ and $x^{(1)}(t)$ are seen to be negligible in comparison with those of $\theta(t)$ and $\theta^{(1)}(t)$. All the estimation errors decay to zero in about 7 s, which is the same time as the settling time of the closed-loop response for the state-vector, $\mathbf{x}(t)$. The observer poles are therefore at acceptable locations. Note that we can move the observer poles as much inside the left-half plane as we want, because there is no control input cost associated with the observer. However, if the measurements of the output are noisy, there will be an increased influence

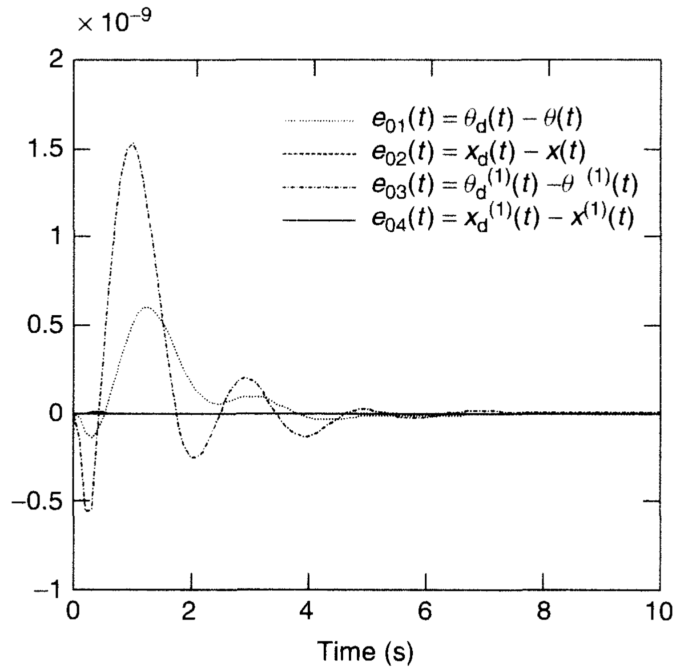


**Figure 5.19** Estimation errors for the compensator based tracking system for inverted- pendulum on a moving cart, with desired angular position, $\theta_d(t) = 0$, and desired cart's displacement, $x_d(t) = 1$ m, when regulator poles are $s = -0.7853 \pm 3.25328i$ and $s = -0.7853 \pm 0.7853i$

of noise on the closed-loop system if the observer poles are too far inside the left-half plane.

### 5.4.2 Pole-placement design of reduced-order observers and compensators

When some of the state variables of a plant can be measured, it is unnecessary to estimate those state variables. Hence, a *reduced-order observer* can be designed which estimates only those state variables that cannot be measured. Suppose the state-vector of a plant, $\mathbf{x}(t)$, can be partitioned into a vector containing measured state variables, $\mathbf{x}_1(t)$, and unmeasurable state variables, $\mathbf{x}_2(t)$, i.e. $\mathbf{x}(t) = [\mathbf{x}_1(t)^T;\ \mathbf{x}_2(t)^T]^T$. The measured output vector, $\mathbf{y}(t)$, may either be equal to the vector, $\mathbf{x}_1(t)$ – implying that all the state variables constituting $\mathbf{x}_1(t)$ can be *directly* measured – or it may be equal to a *linear combination* of the state variables constituting $\mathbf{x}_1(t)$. Hence, the output equation can be generally expressed as

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}_1(t) \tag{5.122}$$

where $\mathbf{C}$ is a constant, *square* matrix, indicating that there are as many outputs as the number of elements in $\mathbf{x}_1(t)$. When $\mathbf{x}_1(t)$ can be directly measured, $\mathbf{C} = \mathbf{I}$. The plant's state-equation (Eq. (5.95)) can be expressed in terms of the partitioned state-vector,

$\mathbf{x}(t) = [\,\mathbf{x}_1(t)^T; \quad \mathbf{x}_2(t)^T\,]^T$, as follows:

$$\mathbf{x}_1^{(1)}(t) = \mathbf{A}_{11}\mathbf{x}_1(t) + \mathbf{A}_{12}\mathbf{x}_2(t) + \mathbf{B}_1\mathbf{u}(t) \qquad (5.123)$$

$$\mathbf{x}_2^{(1)}(t) = \mathbf{A}_{12}\mathbf{x}_1(t) + \mathbf{A}_{22}\mathbf{x}_2(t) + \mathbf{B}_2\mathbf{u}(t) \qquad (5.124)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \qquad (5.125)$$

Let the order of the plant be $n$, and the number of measured state variables (i.e. the dimension of $\mathbf{x}_1(t)$) be $k$. Then a reduced-order observer is required to estimate the vector $\mathbf{x}_2(t)$, which is of dimension $(n - k)$. Hence, the estimated state-vector is simply given by

$$\mathbf{x}_o(t) = \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_{o2}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{C}^{-1}\mathbf{y}(t) \\ \mathbf{x}_{o2}(t) \end{bmatrix} \qquad (5.126)$$

where $\mathbf{x}_{o2}(t)$ is the *estimation* of the vector $\mathbf{x}_2(t)$. Note that Eq. (5.126) requires that $\mathbf{C}$ should be a *non-singular* matrix, which implies that the plant should be *observable* with the output given by Eq. (5.122). If the plant is *unobservable* with the output given by Eq. (5.122), $\mathbf{C}$ would be *singular*, and a reduced-order observer *cannot* be designed.

The observer state-equation should be such that the *estimation error*, $\mathbf{e}_{o2}(t) = \mathbf{x}_2(t) - \mathbf{x}_{o2}(t)$, is always brought to zero in the steady state. A possible observer state-equation would appear to be the extension of the full-order observer state-equation (Eq. (5.103)) for the reduced-order observer, written as follows:

$$\mathbf{x}_o^{(1)}(t) = \mathbf{A}\mathbf{x}_o(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{L}[\mathbf{y}(t) - \mathbf{C}\mathbf{x}_o(t)] \qquad (5.127)$$

where the observer gain matrix, $\mathbf{L}$, would determine the estimation error dynamics. On substituting Eq. (5.126) into Eq. (5.127), and subtracting the resulting state-equation for $\mathbf{x}_{o2}(t)$ from Eq. (5.124), we can write the estimation error state-equation as follows:

$$\mathbf{e}_{o2}^{(1)}(t) = \mathbf{A}_{22}\mathbf{e}_{o2}(t) \qquad (5.128)$$

However, Eq. (5.128) indicates that the estimation error is *unaffected* by the observer gain matrix, $\mathbf{L}$, and solely depends upon the plant's sub-matrix, $\mathbf{A}_{22}$. If $\mathbf{A}_{22}$ turns out to be a matrix having eigenvalues with *positive* real parts, we will be stuck with an estimation error that goes to *infinity* in the steady state! Clearly, the observer state-equation given by Eq. (5.127) is *unacceptable*. Let us try the following reduced-order observer dynamics:

$$\mathbf{x}_{o2}(t) = \mathbf{L}\mathbf{y}(t) + \mathbf{z}(t) \qquad (5.129)$$

where $\mathbf{z}(t)$ is the solution of the following state-equation:

$$\mathbf{z}^{(1)}(t) = \mathbf{F}\mathbf{z}(t) + \mathbf{H}\mathbf{u}(t) + \mathbf{G}\mathbf{y}(t) \tag{5.130}$$

Note that the reduced-order observer gain matrix, $\mathbf{L}$, defined by Eq. (5.129), is of size $[(n-k) \times k]$, whereas the full-order observer gain matrix would be of size $(n \times k)$. On differentiating Eq. (5.129) with respect to time, subtracting the result from Eq. (5.124), and substituting $\mathbf{z}(t) = \mathbf{x}_{o2}(t) - \mathbf{L}\mathbf{y}(t) = \mathbf{x}_2(t) - \mathbf{e}_{o2}(t) - \mathbf{L}\mathbf{C}\mathbf{x}_1(t)$, the state-equation for estimation error is written as follows:

$$\mathbf{e}_{o2}^{(1)}(t) = \mathbf{F}\mathbf{e}_{o2}(t) + (\mathbf{A}_{21} - \mathbf{L}\mathbf{C}\mathbf{A}_{11} + \mathbf{F}\mathbf{L}\mathbf{C})\mathbf{x}_1(t) + (\mathbf{A}_{22} - \mathbf{L}\mathbf{C}\mathbf{A}_{12} - \mathbf{F})\mathbf{x}_2(t) + (\mathbf{B}_2 - \mathbf{L}\mathbf{C}\mathbf{B}_1 - \mathbf{H})\mathbf{u}(t) \tag{5.131}$$

Equation (5.131) implies that for the estimation error, $\mathbf{e}_{o2}(t)$, to go to zero in the steady state, irrespective of $\mathbf{x}_1(t)$, $\mathbf{x}_2(t)$, and $\mathbf{u}(t)$, the coefficient matrices multiplying $\mathbf{x}_1(t)$, $\mathbf{x}_2(t)$, and $\mathbf{u}(t)$ *must vanish*, and $\mathbf{F}$ must have *all* eigenvalues in the left-half plane. Therefore, it follows that

$$\mathbf{F} = \mathbf{A}_{22} - \mathbf{L}\mathbf{C}\mathbf{A}_{12}; \quad \mathbf{H} = \mathbf{B}_2 - \mathbf{L}\mathbf{C}\mathbf{B}_1; \quad \mathbf{G} = \mathbf{F}\mathbf{L} + (\mathbf{A}_{21} - \mathbf{L}\mathbf{C}\mathbf{A}_{11})\mathbf{C}^{-1} \tag{5.132}$$

The reduced-order observer design consists of selecting the observer gain matrix, $\mathbf{L}$, such that all the eigenvalues of $\mathbf{F}$ are in the left-half plane.

### *Example 5.19*

Let us design a reduced-order observer for the inverted pendulum on a moving cart (Example 5.9), based on the measurement of the cart displacement, $x(t)$. The first step is to partition the state-vector into measurable and unmeasurable parts, i.e. $\mathbf{x}(t) = [\mathbf{x}_1(t)^T; \ \mathbf{x}_2(t)^T]^T$, where $\mathbf{x}_1(t) = x(t)$, and $\mathbf{x}_2(t) = [\theta(t); \ \theta^{(1)}(t); \ x^{(1)}(t)]^T$. However, in Example 5.9, the state-vector was expressed as $[\theta(t); \ x(t); \ \theta^{(1)}(t); \ x^{(1)}(t)]^T$. We must therefore rearrange the state coefficient matrices (Eq. (5.54)) such that the state-vector is $\mathbf{x}(t) = [x(t); \ \theta(t); \ \theta^{(1)}(t); \ x^{(1)}(t)]^T$ and partition them as follows:

$$\mathbf{A} = \left[\begin{array}{c|ccc} 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 10.78 & 0 & 0 \\ 0 & -0.98 & 0 & 0 \end{array}\right]; \qquad \mathbf{B} = \left[\begin{array}{c} 0 \\ \hline 0 \\ -1 \\ 1 \end{array}\right] \tag{5.133}$$

From Eq. (5.133) it is clear that

$$\mathbf{A}_{11} = 0; \quad \mathbf{A}_{12} = [0 \quad 0 \quad 1]; \quad \mathbf{B}_1 = 0$$

$$\mathbf{A}_{21} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \quad \mathbf{A}_{22} = \begin{bmatrix} 0 & 1 & 0 \\ 10.78 & 0 & 0 \\ -0.98 & 0 & 0 \end{bmatrix}; \quad \mathbf{B}_2 = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \tag{5.134}$$

Since the measured output is $\mathbf{x}_1(t) = x(t)$, the output equation is $\mathbf{y}(t) = \mathbf{C}\mathbf{x}_1(t)$, where $\mathbf{C} = 1$. We have to select an observer gain-matrix, **L**, such that the eigenvalues of $\mathbf{F} = (\mathbf{A}_{22} - \mathbf{L}\mathbf{C}\mathbf{A}_{12})$ are in the left-half plane. Let us select the observer poles, i.e. the eigenvalues of **F**, to be $s = -20$, $s = -20 \pm 20i$. Then **L** is calculated by pole-placement as follows:

```
>>A12 = [0 0 1]; A22 = [0 1 0; 10.78 0 0; -0.98 0 0]; C = 1; <enter>

>>v = [-20 -20+20i -20-20i]'; L = (place(A22',A12'*C',v))' <enter>

L =
-1.6437e+003
-1.6987e+004
6.0000e+001
```

Therefore, the observer dynamics matrix, **F**, is calculated as follows:

```
>>F = A22 - L*C*A12 <enter>

F =
  0             1.0000e+000  1.6437e+003
  1.0780e+001   0            1.6987e+004
 -9.8000e-001   0           -6.0000e+001
```

Let us verify that the eigenvalues of **F** are at desired locations:

```
>>eig(F) <enter>

ans =
-2.0000e+001+2.0000e+001i
-2.0000e+001-2.0000e+001i
-2.0000e+001
```

which indeed they are. The other observer coefficient matrices, **G** and **H**, are calculated as follows:

```
>>A11 = 0; A21 = [0 0 0]'; B1 = 0; B2 = [0 -1 1]';
   H = B2 - L*C*B1 <enter>

H =
      0
      -1
      1

>>G = F*L + (A21 - L*C*A11)*inv(C) <enter>

G =
      8.1633e+004
      1.0015e+006
      -1.9892e+003
```

A compensator based on the reduced-order observer can be designed by the *separation principle*, in a manner similar to the compensator based on the full-order observer. The control-law defining the reduced-order compensator for a tracking system can be expressed as follows, after substituting Eq. (5.126) into Eq. (5.109):

$$\mathbf{u}(t) = \mathbf{K}[\mathbf{x}_d(t) - \mathbf{x}_o(t)] - \mathbf{K}_d\mathbf{x}_d(t) = (\mathbf{K} - \mathbf{K}_d)\mathbf{x}_d(t) - \mathbf{K}_1\mathbf{x}_1(t) - \mathbf{K}_2\mathbf{x}_{o2}(t) \quad (5.135)$$

where $\mathbf{x}_d(t)$ is the desired state-vector, $\mathbf{K}_d$ is the feedforward gain matrix, and $\mathbf{K}$ is the feedback gain matrix, which can be partitioned into gain matrices that feedback $\mathbf{x}_1(t)$ and $\mathbf{x}_{o2}(t)$, respectively, as $\mathbf{K} = [\,\mathbf{K}_1; \quad \mathbf{K}_2\,]$. A schematic diagram of the reduced-order compensator is shown in Figure 5.20.

The estimation error dynamics of the reduced-order compensator is described by the following state-equation, obtained by substituting Eq. (5.132) into Eq. (5.131):

$$\mathbf{e}_{o2}{}^{(1)}(t) = \mathbf{F}\mathbf{e}_{o2}(t) \quad (5.136)$$

while the state-equation for the tracking error, $\mathbf{e}(t) = \mathbf{x}_d(t) - \mathbf{x}(t)$, is obtained by subtracting Eq. (5.74) from Eq. (5.95), and substituting Eq. (5.135) as follows:

$$\mathbf{e}^{(1)}(t) = \mathbf{A}\mathbf{e}(t) + (\mathbf{A}_d - \mathbf{A} + \mathbf{B}\mathbf{K}_d)\mathbf{x}_d(t) - \mathbf{B}\mathbf{K}[\mathbf{x}_d(t) - \mathbf{x}_o(t)] \quad (5.137)$$

On substituting for $\mathbf{x}_o(t)$ from Eq. (5.126), Eq. (5.137) can be written as follows:

$$\mathbf{e}^{(1)}(t) = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{e}(t) + (\mathbf{A}_d - \mathbf{A} + \mathbf{B}\mathbf{K}_d)\mathbf{x}_d(t) - \mathbf{B}\mathbf{K}_2\mathbf{e}_{o2}(t) \quad (5.138)$$

Hence, the dynamics of the tracking system can be described by Eqs. (5.136) and (5.138). To have the tracking error go to zero in the steady state, irrespective of $\mathbf{x}_d(t)$, we must select the feedforward gain matrix, $\mathbf{K}_d$, such that $(\mathbf{A}_d - \mathbf{A} + \mathbf{B}\mathbf{K}_d)\mathbf{x}_d(t) = \mathbf{0}$, and the feedback gain matrix, $\mathbf{K}$, such that the eigenvalues of $(\mathbf{A} - \mathbf{B}\mathbf{K})$ are in the left-half plane. Since the eigenvalues of $(\mathbf{A} - \mathbf{B}\mathbf{K})$ are the regulator poles, and eigenvalues of $\mathbf{F}$ are

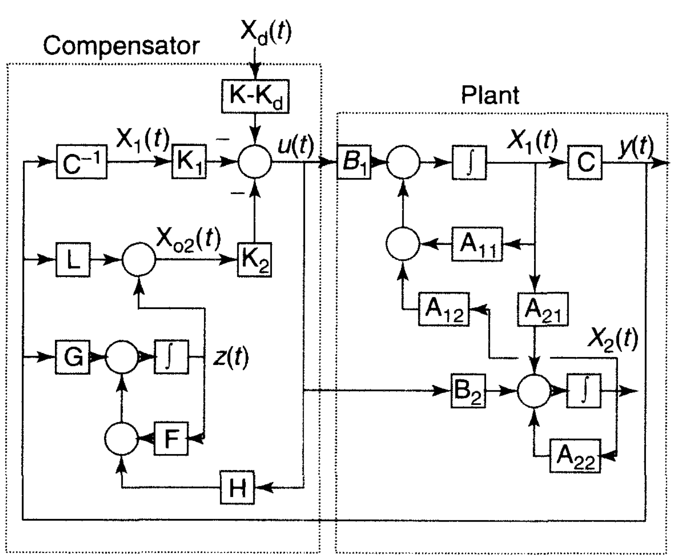


**Figure 5.20** Tracking system based on reduced-order compensator

the reduced-order observer poles, it follows from Eqs. (5.136) and (5.138) that the *poles of the tracking system* are of *observer and regulator poles*. (Prove this fact by finding the eigenvalues of the closed-loop system whose state-vector is $[\mathbf{e}(t)^T; \ \mathbf{e}_{o2}(t)^T]^T$.) According to the separation principle, the design of regulator and observer can be carried out separately by pole-placement. Note from Eqs. (5.136) and (5.138) that the order of the reduced-order tracking system is $(2n - k)$, where $k$ is the number of measurable state-variables. Recall form the previous sub-section that the order of the full-order tracking system was $2n$. Thus, the more state-variables we can measure, the smaller will be the order of the tracking system based on reduced-order observer.

### *Example 5.20*

Let us re-design the tracking system for the inverted pendulum on a moving cart (Example 5.18), using a reduced-order observer. Recall that it is desired to move the cart by 1 m, while not letting the pendulum fall. We have already designed a reduced-order observer for this plant in Example 5.19, using the measurement of the cart's displacement, $x(t)$, such that the observer poles are $s = -20$, $s = -20 \pm 20i$. In Example 5.18, we were able to make $(\mathbf{A}_d - \mathbf{A} + \mathbf{B}\mathbf{K}_d)\mathbf{x}_d(t) = \mathbf{0}$ with $\mathbf{K}_d = \mathbf{0}$. It remains to select the regulator gain matrix, $\mathbf{K}$, such that the eigenvalues of $(\mathbf{A} - \mathbf{B}\mathbf{K})$ are at desired locations in the left-half plane. As in Example 5.18, let us choose the regulator poles to be $s = -0.7853 \pm 3.25328i$ and $s = -0.7853 \pm 0.7853i$. Note that we cannot directly use the regulator gain matrix of Example 5.18, because the state-vector has been *re-defined* in Example 5.19 to be $\mathbf{x}(t) = [x(t); \ \ \theta(t); \ \ \theta^{(1)}(t); \ \ x^{(1)}(t)]^T$, as opposed to $\mathbf{x}(t) = [\theta(t); x(t); \theta^{(1)}(t); x^{(1)}(t)]^T$ of Example 5.18. The new regulator gain matrix would thus be obtained by *switching* the *first* and *second* elements of $\mathbf{K}$ calculated in Example 5.18, or by repeating pole-placement using the re-arranged state coefficient matrices as follows:

```
>>A = [A11 A12; A21 A22]; B = [B1; B2]; <enter>

>>v=[-0.7853+3.25328i -0.7853-3.25328i -0.7853+0.7853i
  -0.7853-0.7853i]'; K=place(A,B,v) <enter>

place: ndigits= 16

K =

-1.4097e+000   -2.7090e+001   -5.1339e+000   -1.9927e+000
```

The partitioning of $\mathbf{K}$ results in $\mathbf{K}_1 = -1.4097$ and $\mathbf{K}_2 = [-27.090; -5.1339; -1.9927]$. The closed-loop error dynamics matrix, $\mathbf{A}_{CL}$, is the state-dynamics matrix obtained by combining Eqs. (5.136) and (5.138) into a state-equation, with the state-vector, $[\mathbf{e}(t); \ \ \mathbf{e}_{o2}(t)]^T$, and is calculated as follows:

```
>>K2 = K(2:4); ACL = [A-B*K -B*K2; zeros(3,4) F]; <enter>
```

The eigenvalues of $\mathbf{A}_{CL}$ are calculated as follows:

```
>>eig(ACL) <enter>

ans =
-7.8530e-001+3.2533e+000i
-7.8530e-001-3.2533e+000i
-7.8530e-001+7.8530e-001i
-7.8530e-001-7.8530e-001i
-2.0000e+001+2.0000e+001i
-2.0000e+001-2.0000e+001i
-2.0000e+001
```

Note that the closed-loop eigenvalues consist of the regulator and observer poles, as expected. The *closed-loop error response* (i.e. the solution of Eqs. (5.136) and (5.138)) to $\mathbf{x}_d(t) = [1; 0; 0; 0]^T$ is nothing else but the initial response to $[\mathbf{e}(0)^T; \mathbf{e}_{o2}(0)^T]^T = [1; 0; 0; 0; 0; 0; 0]^T$, which is computed as follows:

```
>>sysCL=ss(ACL,zeros(7,1),eye(7),zeros(7,1));
  [y,t,e]= initial(sysCL,[1 zeros(1,6)]); <enter>
```

The estimation error vector, $\mathbf{e}_{o2}(t)$ is *identically zero* for this example, while the tracking errors, i.e. elements of $\mathbf{e}(t)$, are plotted in Figure 5.21. Note in Figure 5.21 that all the error transients decay to zero in about 7 s. The maximum value for the cart's velocity, $x^{(1)}(t)$, is seen to be about 0.9 m/s, while the angular velocity of the pendulum, $\theta^{(1)}(t)$, reaches a maximum value of 0.25 rad/s. The angular displacement of the pendulum, $\theta(t)$, is always less than 0.1 rad (5.73°) in magnitude, which is acceptably small for the validity of the linear plant model.

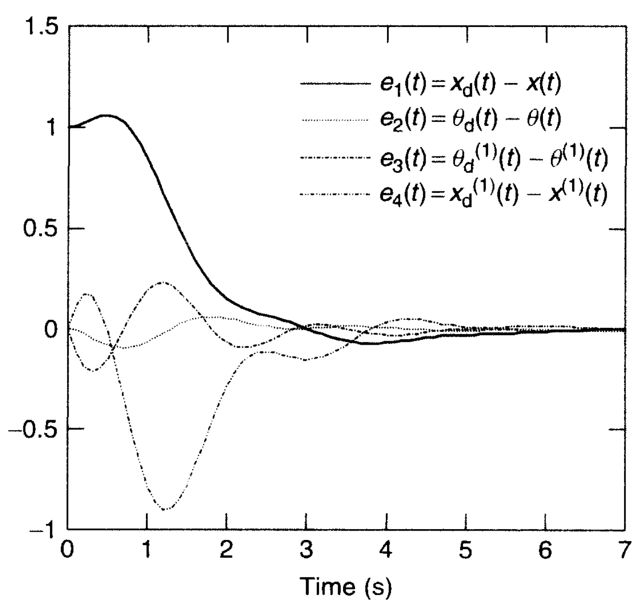


**Figure 5.21** Tracking error response of closed-loop system consisting of an inverted pendulum on a moving cart and a reduced-order compensator (Example 5.20)

### 5.4.3 Noise and robustness issues

If noise is present in the plant, the plant's state-space representation is given by Eqs. (5.26) and (5.27), and the feedback control-law, Eq. (5.109), is modified as follows:

$$\mathbf{u}(t) = \mathbf{K}[\mathbf{x}_d(t) - \mathbf{x}_o(t)] - \mathbf{K}_d\mathbf{x}_d(t) - \mathbf{K}_n\mathbf{x}_n(t) \tag{5.139}$$

where $\mathbf{x}_d(t)$ and $\mathbf{x}_n(t)$ are the desired state-vector and the noise vector, respectively, and $\mathbf{K}_d$ and $\mathbf{K}_n$ are the feedforward gain matrices. In Eq. (5.139) it is assumed that both $\mathbf{x}_d(t)$ and $\mathbf{x}_n(t)$ can be measured, and thus need not be estimated by the observer. In case $\mathbf{x}_d(t)$ and $\mathbf{x}_n(t)$ are unmeasurable, we have to know the state-space model of the processes by which they are generated, in order to obtain their estimates. While it may be possible to know the dynamics of the desired state-vector, $\mathbf{x}_d(t)$, the noise-vector, $\mathbf{x}_n(t)$, is usually generated by a non-deterministic process whose mathematical model is unknown. In Chapter 7, we will derive observers which include an approximate model for the stochastic processes that generate noise, and design compensators for such plants. In Chapter 7 we will also study the robustness of multivariable control systems with respect to random noise.

SIMULINK can be used to simulate the response of a control system to noise, parameter variations, and nonlinearities, thereby giving a direct information about a system's robustness.

***Example 5.21***

Let us simulate the inverted-pendulum on a moving cart with the control system designed in Example 5.18 with a full-order compensator, with the addition of *measurement noise* modeled as a *band limited white noise* source block of SIMULINK. *White noise* is a statistical model of a special random process that we will discuss in Chapter 7. The parameter *power* of the white noise block representing the intensity of the noise is selected as $10^{-8}$. A SIMULINK block-diagram of the plant with full-order compensator with regulator and observer gains designed in Example 5.18 is shown in Figure 5.22. Note the use of *matrix gain* blocks to synthesize the compensator, and a masked *subsystem* block for the state-space model of the plant. The matrix gain blocks are named B = B1 for the matrix **B**, C = C1 for the matrix **C**, L = L1 for the observer gain matrix **L**, and K = K1 = K2 for the regulator gain matrix, **K**. The *scope* outputs *thet, x, thdot* and *xdot* are the state-variables $\theta(t)$, $x(t)$, $\theta^{(1)}(t)$, and $x^{(1)}(t)$, respectively, which are *demux-ed* from the state vector of the plant, and are also saved as variables in the MATLAB workspace. The resulting simulation of $\theta(t)$ and $x(t)$ is also shown in Figure 5.22. Note the random fluctuations in both $\theta(t)$ and $x(t)$ about the desired steady-state values of $x_d = 1$ m and $\theta_d = 0$. The maximum magnitude of $\theta(t)$ is limited to 0.1 rad., which is within the range required for linearizing the equations of motion. However, if the intensity of the measurement noise is increased, the $\theta(t)$ oscillations quickly surpass the linear range. The simulation of Figure 5.22 also conforms to our usual experience in trying to balance a stick vertically on a finger.

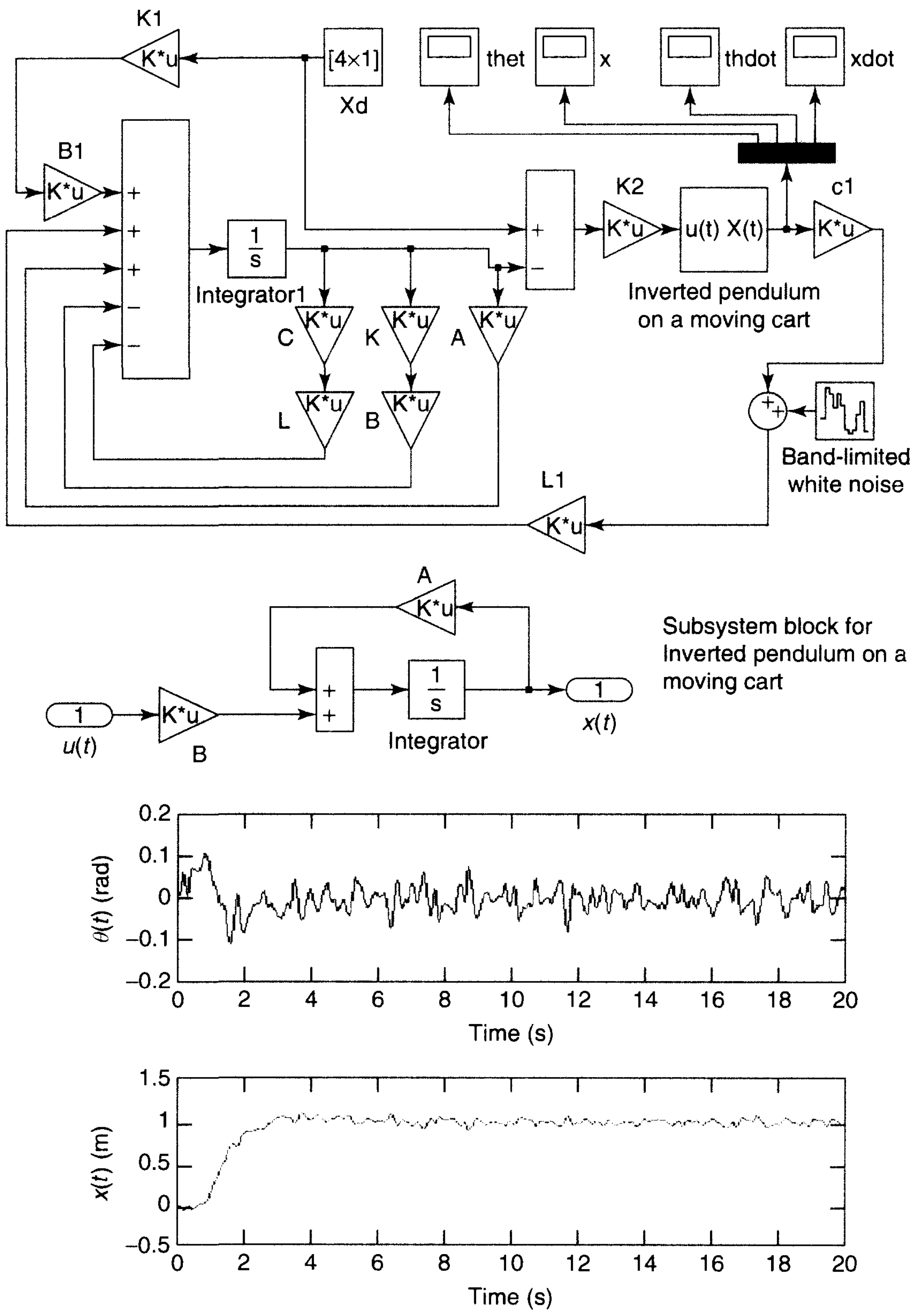


**Figure 5.22** Simulation of the inverted pendulum on a moving cart with a full-order compensator and measurement noise with SIMULINK block-diagram

# Exercises

5.1. Check the controllability of the plants with the following state-coefficient matrices:

(a)
$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ -1 & 1 \end{bmatrix} \tag{5.140}$$

(b)
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 25 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \\ 5 \end{bmatrix} \tag{5.141}$$

5.2. As discussed in Section 5.1, an unstable plant is said to be *stabilizable* if all the *uncontrollable* sub-systems have *stable* eigenvalues. Check whether the plants given in Exercise 5.1 are stabilizable.

5.3. If a plant is *stabilizable* (see Exercise 5.2), we can safely ignore the *uncontrollable* sub-systems by removing the rows and columns corresponding to the uncontrollable states from the state coefficient matrices, **A** and **B**. The resulting state-space representation would be controllable, and is called a *minimal realization.* Find the minimal realization of the state coefficient matrices, **A** and **B** for the plants in Exercise 5.1.

5.4. A *distillation column* in a chemical plant has the following state-coefficient matrices:

$$\mathbf{A} = \begin{bmatrix} -21 & 0 & 0 & 0 \\ 0.1 & -5 & 0 & 0 \\ 0 & -1.5 & 0 & 0 \\ 0 & -4 & 0 & 0 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 6000 & 0 \\ 0 & 0 \\ 0 & 2.3 \\ 0 & 0.1 \end{bmatrix} \tag{5.142}$$

(a) Is the plant controllable?

(b) Suppose we would like to control the plant using *only one* input at a time. Is the plant controllable with only the *first* input, i.e. with $\mathbf{B} = [6000; 0; 0; 0]^T$? Is the plant controllable with only the *second* input, i.e. with $\mathbf{B} = [0; 0; 2.3; 0.1]^T$?

5.5. For the aircraft with lateral dynamics given in Eq. (4.97) in Exercise 4.3:

(a) is the aircraft controllable using both the inputs?

(b) is the aircraft controllable using *only* the aileron input, $\delta_A(t)$?

(c) is the aircraft controllable using *only* the rudder input, $\delta_R(t)$?

5.6. Consider the longitudinal dynamics of a flexible bomber airplane of Example 4.7, with the state-space representation given by Eq. (4.71).

(a) Is the aircraft controllable using both the inputs, $u_1(t)$ and $u_2(t)$?

(b) Is the aircraft controllable using *only* the desired elevator deflection, $u_1(t)$?

(c) Is the aircraft controllable using *only* the desired canard deflection, $u_2(t)$?

5.7. For the aircraft in Exercise 5.5, can you design a full-state feedback regulator which places the closed-loop poles of the aircraft at $s_{1,2} = -1 \pm i$, $s_3 = -15$, $s_4 = -0.8$ using *only one of the inputs* ? If so, which one, and what is the appropriate gain matrix?

5.8. For the aircraft in Exercise 5.6, design a full-state feedback regulator using both the inputs and the MATLAB (CST) command *place*, such that the closed-loop poles are located at $s_{1,2} = -3 \pm 3i$, $s_{3,4} = -1 \pm 2i$, $s_5 = -100$, $s_6 = -75$. Find the maximum overshoots and settling time of the closed-loop initial response if the initial condition vector is $\mathbf{x}(0) = [0; 0.5; 0; 0; 0; 0]^T$.

5.9. For the distillation column of Exercise 5.4, design a full-state feedback regulator to place the closed-loop poles at $s_{1,2} = -0.5 \pm 0.5i$, $s_3 = -5$, $s_4 = -21$.

5.10. Repeat Exercise 5.9 for the closed-loop poles in a Butterworth pattern of radius, $R = 5$. Compare the initial response of the *first state-variable* (i.e. for $\mathbf{C} = [1; 0; 0; 0]$ and $\mathbf{D} = [0; 0]$) of the resulting closed-loop system with that of Exercise 5.9 for initial condition, $\mathbf{x}(0) = [1; 0; 0; 0]^T$. Which of the two (present and that of Exercise 5.9) regulators requires the larger control input magnitudes for this initial condition?

5.11. Consider the turbo-generator of Example 3.14, with the state-space representation given by Eq. (3.117).

(a) Is the plant controllable using both the inputs, $u_1(t)$ and $u_2(t)$?

(b) Is the plant controllable using *only* the input, $u_1(t)$?

(c) Is the plant controllable using *only* the input, $u_2(t)$?

(d) Design a full-state feedback regulator for the plant using only the input, $u_1(t)$, such that the closed-loop eigenvalues are at $s_{1,2} = -2.5 \pm 2.5i$, $s_{3,4} = -1 \pm i$, $s_5 = -10$, $s_6 = -15$.

(e) Repeat part (d) using only the input, $u_2(t)$.

(f) Repeat part (d) using both the inputs, $u_1(t)$ and $u_2(t)$, and the MATLAB (CST) command *place* for designing the multi-input regulator.

(g) Re-design the regulators in parts (d)–(f), such that the maximum overshoot and settling time for the output, $y_1(t)$, are less than 0.3 units and 6 seconds, respectively, if the initial condition vector is $\mathbf{x}(0) = [0.1; 0; 0; 0; 0; 0]^T$.

(h) Re-design the regulators in parts (d)–(f), such that the closed-loop poles are in a Butterworth pattern of radius, $R = 10$, and compare the closed-loop initial responses and input magnitudes with those of part (g).

5.12. Check the observability of the plants with the following state coefficient matrices:

$$\text{(a)}\ \mathbf{A} = \begin{bmatrix} -1 & 0 & 0 \\ 0.3 & -0.1 & 0.05 \\ 1 & 0 & 0 \end{bmatrix}; \quad \mathbf{C} = [0 \quad 0 \quad -2] \tag{5.143}$$

$$\text{(b)}\ \mathbf{A} = \begin{bmatrix} 0 & 0.1 & -100 & 4 \\ -250 & -7 & 3 & 50 \\ 0 & 0 & -3.3 & 0.06 \\ 2 & 0 & 0 & 0.25 \end{bmatrix}; \quad \mathbf{C} = [0 \quad 0 \quad 1 \quad 0] \tag{5.144}$$

$$\text{(c)}\ \mathbf{A} = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}; \quad \mathbf{C} = \begin{bmatrix} 1 & 0 \\ -2 & 0 \end{bmatrix} \tag{5.145}$$

5.13. An unstable plant is said to be *detectable* if all the *unobservable* sub-systems have *stable* eigenvalues. Check whether the plants given in Exercise 5.12 are detectable.

5.14. If a plant is *detectable* (see Exercise 5.13), we can safely ignore the *unobservable* sub-systems by removing the rows and columns corresponding to the unobservable states from the state coefficient matrices, **A** and **B**. The resulting state-space representation would be observable, and is called a *minimal realization.* Find the minimal realization of the state coefficient matrices, **A** and **B** for the plants in Exercise 5.12.

5.15. For the distillation column of Exercise 5.4, the matrices **C** and **D** are as follows:

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad \mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.146}$$

(a) Is the plant observable?

(b) Is the plant observable if *only* the *first output* was measured, i.e. $\mathbf{C} = [0; 0; 1; 0]$, $\mathbf{D} = 0$?

(c) Is the plant observable if *only* the *second output* was measured, i.e. $\mathbf{C} = [0; 0; 0; 1]$, $\mathbf{D} = 0$?

5.16. For the aircraft with lateral dynamics given in Eq. (4.97) in Exercise 4.3:

(a) is the aircraft observable with the output vector, $\mathbf{y}(t) = [p(t); r(t)]^T$?

(b) is the aircraft observable with the output vector, $\mathbf{y}(t) = [p(t); \phi(t)]^T$?

(c) is the aircraft observable with the bank-angle, $\phi(t)$, being the only measured output?

(d) is the aircraft observable with the sideslip-angle, $\beta(t)$, being the only measured output?

(e) design a full-order observer for the aircraft using *only one* of the state-variables as the output, such that the observer poles are placed at $s_{1,2} = -2 \pm 2i$, $s_3 = -16$, $s_4 = -2$.

(f) design a full-order compensator based on the regulator of Exercise 5.7, and the observer designed in part (e). Calculate and plot the initial response, $\phi(t)$, of the compensated system if the initial condition is $\mathbf{x}(0) = [0.5; 0; 0; 0]^T$.

5.17. Consider the longitudinal dynamics of a flexible bomber airplane of Example 4.7.

(a) Is the aircraft observable using both the outputs, $y_1(t)$ and $y_2(t)$?

(b) Is the aircraft observable with *only* the normal acceleration, $y_1(t)$, as the measured output?

(c) Is the aircraft observable with *only* the pitch-rate, $y_2(t)$, as the measured output?

(d) Design a full-order observer for the aircraft using *only* the normal acceleration output, $y_1(t)$, such that the observer poles are placed at $s_{1,2} = -4 \pm 4i$, $s_{3,4} = -3 \pm 3i$, $s_5 = -100$, $s_6 = -75$.

(e) Design a full-order compensator for the aircraft with the regulator of Exercise 5.8 and the observer of part (d). Compute the initial response of the compensated system with the initial condition given by $\mathbf{x}(0) = [0; 0.5; 0; 0; 0; 0]^T$, and compare with that obtained for the regulated system in Exercise 5.8. Also compare the required inputs for the regulated and compensated systems.

5.18. For the distillation column of Exercise 5.15, design a two-output, full-order observer using the MATLAB (CST) command *place* such that the observer poles are placed at $s_{1,2} = -2 \pm 2i, s_3 = -5, s_4 = -21$. With the resulting observer and the regulator designed in Exercise 5.9, design a full-order compensator and find the initial response of the compensated system for the initial condition $\mathbf{x}(0) = [1; 0; 0; 0]^T$. What are the control inputs required to produce the compensated initial response?

5.19. Consider the turbo-generator of Example 3.14, with the state-space representation given by Eq. (3.117).

(a) Is the plant observable with both the inputs, $y_1(t)$ and $y_2(t)$?

(b) Is the plant observable with *only* the output, $y_1(t)$?

(c) Is the plant observable with *only* the output, $y_2(t)$?

(d) Design a full-order observer for the plant using only the output, $y_1(t)$, such that the observer poles are placed at $s_{1,2} = -3.5 \pm 3.5i, s_{3,4} = -5 \pm 5i, s_5 = -10, s_6 = -15$.

(e) Repeat part (d) using only the output, $y_2(t)$.

(f) Repeat part (d) using both the outputs, $y_1(t)$ and $y_2(t)$, and the MATLAB (CST) command *place* for designing the two-output full-order observer.

(g) Re-design the observers in parts (d)–(f), and combine them with the corresponding regulators designed in Exercise 5.11(g) to form compensators, such that the maximum overshoot and settling time for the compensated initial response, $y_1(t)$, are less than 0.3 units and 6 seconds, respectively, if the initial condition vector is $\mathbf{x}(0) = [0.1; 0; 0; 0; 0; 0]^T$. How do the input magnitudes compare with the required inputs of the corresponding regulators in Exercise 5.11(g)?

5.20. Design a reduced-order observer for the aircraft of Exercise 5.16 using the bank-angle, $\phi(t)$, as the only output, such that the observer poles are in a Butterworth pattern of radius, $R = 16$, and combine it with the regulator of Exercise 5.7, to form a reduced-order compensator. Compare the initial response, $\phi(t)$, and the required inputs of the reduced-order compensated system to that of the full-order compensator in Exercise 5.16 (f) with the initial condition $\mathbf{x}(0) = [0.5; 0; 0; 0]^T$.

5.21. Design a reduced-order observer for the aircraft of Exercise 5.17 with normal acceleration, $y_1(t)$, as the only output, such that the observer poles are in a Butterworth pattern of radius, $R = 100$, and combine it with the regulator of Exercise 5.8, to form a reduced-order compensator. Compare the initial response and the required inputs of the reduced-order compensated system to that of the full-order compensator in Exercise 5.17(e) with the initial condition $\mathbf{x}(0) = [0; 0.5; 0; 0; 0; 0]^T$.

5.22. For the distillation column of Exercise 5.15, design a two-output, reduced-order observer using the MATLAB (CST) command *place* such that the observer poles are placed in Butterworth pattern of radius, $R = 21$. With the resulting observer and the regulator designed in Exercise 5.9, form a reduced-order compensator and compare the initial response and required inputs of the compensated system for the initial condition $\mathbf{x}(0) = [1; 0; 0; 0]^T$, with the corresponding values obtained in Exercise 5.18.

5.23. Using SIMULINK, simulate the tracking system for the inverted pendulum on a moving cart with the reduced-order compensator designed in Example 5.20, including a measurement white noise of intensity (i.e. *power* parameter in the *band-limited white noise* block) when the desired plant state is $\mathbf{x}_d = [0; 1; 0; 0]^T$.

5.24. Using SIMULINK, test the *robustness* of the full-order compensator designed (using linear plant model) in Example 5.18 in controlling the plant model described by the *nonlinear* state-equations of Eqs. (3.17) and (3.18) when the desired plant state is $\mathbf{x}_d = [0; 1; 0; 0]^T$. (Hint: replace the *subsystem* block in Figure 5.22 with a *function* M-file for the nonlinear plant dynamics.)

## References

1. Kreyszig, E. *Advanced Engineering Mathematics*. John Wiley & Sons, New York, 1972.
2. Friedland, B. *Control System Design – An Introduction to State-Space Methods*. McGraw-Hill International Edition, Singapore, 1987.
3. Kautsky, J. and Nichols, N.K. *Robust Eigenstructure Assignment in State Feedback Control*. Numerical Analysis Report NA/2/83, School of Mathematical Sciences, Flinders University, Australia, 1983.

# 6

# Linear Optimal Control

## 6.1 The Optimal Control Problem

After designing control systems by pole-placement in Chapter 5, we naturally ask why we should need to go any further. Recall that in Chapter 5 we were faced with an *over-abundance* of design parameters for multi-input, multi-output systems. For such systems, we did not quite know how to determine all the design parameters, because only a limited number of them could be found from the closed-loop pole locations. The MATLAB M-file *place.m* imposes *additional conditions* (apart from closed-loop pole locations) to determine the design parameters for multi-input regulators, or multi-output observers; thus the design obtained by *place.m* cannot be regarded as pole-placement alone. *Optimal control* provides an alternative design strategy by which *all* the control design parameters can be determined even for multi-input, multi-output systems. Also in Chapter 5, we did not know *a priori* which pole locations would produce the desired performance; hence, some trial and error with pole locations was required before a satisfactory performance could be achieved. *Optimal control* allows us to directly formulate the performance objectives of a control system (provided we know how to do so). More importantly – apart from the above advantages – *optimal control* produces the *best possible* control system for a given set of performance objectives. What do we mean by the adjective *optimal*? The answer lies in the fact that there are many ways of doing a particular thing, but only one way which requires the *least effort*, which implies the least expenditure of *energy* (or money). For example, we can hire the most expensive lawyer in town to deal with our inconsiderate neighbor, or we can directly talk to the neighbor to achieve the desired result. Similarly, a control system can be designed to meet the desired performance objectives with the *smallest control energy*, i.e. the energy associated with generating the control inputs. Such a control system which *minimizes the cost* associated with generating control inputs is called an *optimal control system*. In contrast to the pole-placement approach, where the desired performance is *indirectly* achieved through the location of closed-loop poles, the optimal control system *directly addresses* the desired performance objectives, while minimizing the control energy. This is done by formulating an *objective function* which must be *minimized* in the design process. However, one must know how the performance objectives can be precisely translated into the objective function, which usually requires some experience with a given system.

If we define a system's *transient energy* as the total energy of the system when it is undergoing the transient response, then a successful control system must have a transient energy which quickly decays to zero. The maximum value of the transient energy indicates the maximum overshoot, while the time taken by the transient energy to decay to zero indicates the settling time. By including the transient energy in the *objective function*, we can specify the values of the acceptable maximum overshoot and settling time. Similarly, the *control energy* must also be a part of the objective function that is to be minimized. It is clear that the total control energy and total transient energy can be found by integrating the control energy and transient energy, respectively, with respect to time. Therefore, the objective function for the optimal control problem must be a *time integral* of the sum of transient energy and control energy expressed as functions of time.

### 6.1.1 The general optimal control formulation for regulators

Consider a linear plant described by the following state-equation:

$$\mathbf{x}^{(1)}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \tag{6.1}$$

Note that we have deliberately chosen a *time-varying* plant in Eq. (6.1), because the optimal control problem is generally formulated for time-varying systems. For simplicity, suppose we would like to design a full-state feedback regulator for the plant described by Eq. (6.1) such that the control input vector is given by

$$\mathbf{u}(t) = -\mathbf{K}(t)\mathbf{x}(t) \tag{6.2}$$

The control law given by Eq. (6.2) is linear. Since the plant is also linear, the closed-loop control system would be linear. The *control energy* can be expressed as $\mathbf{u}^T(t)\mathbf{R}(t)\mathbf{u}(t)$, where $\mathbf{R}(t)$ is a *square, symmetric matrix* called the *control cost matrix*. Such an expression for control energy is called a *quadratic form*, because the *scalar* function, $\mathbf{u}^T(t)\mathbf{R}(t)\mathbf{u}(t)$, contains quadratic functions of the elements of $\mathbf{u}(t)$. Similarly, the *transient energy* can also be expressed in a quadratic form as $\mathbf{x}^T(t)\mathbf{Q}(t)\mathbf{x}(t)$, where $\mathbf{Q}(t)$ is a *square, symmetric matrix* called the *state weighting matrix*. The *objective function* can then be written as follows:

$$J(t, t_f) = \int_t^{t_f} [\mathbf{x}^T(\tau)\mathbf{Q}(\tau)\mathbf{x}(\tau) + \mathbf{u}^T(\tau)\mathbf{R}(\tau)\mathbf{u}(\tau)]\, d\tau \tag{6.3}$$

where $t$ and $t_f$ are the *initial* and *final* times, respectively, for the control to be exercised, i.e. the control *begins* at $\tau = t$ and *ends* at $\tau = t_f$, where $\tau$ is the variable of integration. The optimal control problem consists of solving for the feedback gain matrix, $\mathbf{K}(t)$, such that the scalar objective function, $J(t, t_f)$, given by Eq. (6.3) is *minimized*. However, the minimization must be carried out in such a manner that the state-vector, $\mathbf{x}(t)$, is the solution of the plant's state-equation (Eq. (6.1)). Equation (6.1) is called a *constraint* (because in its absence, $\mathbf{x}(t)$ would be free to assume *any* value), and the resulting minimization is said to be a *constrained minimization*. Hence, we are looking for a regulator gain matrix, $\mathbf{K}(t)$, which *minimizes* $J(t, t_f)$ *subject to the constraint* given by Eq. (6.1). Note that the transient term, $\mathbf{x}^T(\tau)\mathbf{Q}(\tau)\mathbf{x}(\tau)$, in the objective function implies that a *departure* of the system's state, $\mathbf{x}(\tau)$, from the *final desired state*, $\mathbf{x}(t_f) = \mathbf{0}$, is to be minimized. In other words, the design objective is to bring $\mathbf{x}(\tau)$ to a *constant value of zero* at final time, $\tau = t_f$. If the final desired state is *non-zero*, the objective function can be modified appropriately, as we will see later.

By substituting Eq. (6.2) into Eq. (6.1), the closed-loop state-equation can be written as follows:

$$\mathbf{x}^{(1)}(t) = [\mathbf{A}(t) - \mathbf{B}(t)\mathbf{K}(t)]\mathbf{x}(t) = \mathbf{A}_{\mathbf{CL}}(t)\mathbf{x}(t) \tag{6.4}$$

where $\mathbf{A}_{\mathbf{CL}}(t) = [\mathbf{A}(t) - \mathbf{B}(t)\mathbf{K}(t)]$, the closed-loop state-dynamics matrix. The solution to Eq. (6.4) can be written as follows:

$$\mathbf{x}(t) = \Phi_{\mathbf{CL}}(t, t_0)\mathbf{x}(t_0) \tag{6.5}$$

where $\Phi_{\mathbf{CL}}(t, t_0)$ is the *state-transition matrix* of the time-varying closed-loop system represented by Eq. (6.4). Since the system is time-varying, $\Phi_{\mathbf{CL}}(t, t_0)$, is *not* the matrix exponential of $\mathbf{A}_{\mathbf{CL}}(t - t_0)$, but is related in *some other way* (which we do not know) to $\mathbf{A}_{\mathbf{CL}}(t)$. Equation (6.5) indicates that the state at any time, $\mathbf{x}(t)$, can be obtained by post-multiplying the state at some initial time, $\mathbf{x}(t_0)$, with $\Phi_{\mathbf{CL}}(t, t_0)$. On substituting Eq. (6.5) into Eq. (6.3), we get the following expression for the objective function:

$$J(t, t_f) = \int_t^{t_f} \mathbf{x}^{\mathbf{T}}(t)\Phi_{\mathbf{CL}}^{\mathbf{T}}(\tau, t)[\mathbf{Q}(\tau) + \mathbf{K}^{\mathbf{T}}(\tau)\mathbf{R}(\tau)\mathbf{K}(\tau)]\Phi_{\mathbf{CL}}(\tau, t)\mathbf{x}(t)\, d\tau \tag{6.6}$$

or, taking the *initial state-vector*, $\mathbf{x}(t)$, outside the integral sign, we can write

$$J(t, t_f) = \mathbf{x}^{\mathbf{T}}(t)\mathbf{M}(t, t_f)\mathbf{x}(t) \tag{6.7}$$

where

$$\mathbf{M}(t, t_f) = \int_t^{t_f} \Phi_{\mathbf{CL}}^{\mathbf{T}}(\tau, t)[\mathbf{Q}(\tau) + \mathbf{K}^{\mathbf{T}}(\tau)\mathbf{R}(\tau)\mathbf{K}(\tau)]\Phi_{\mathbf{CL}}(\tau, t)\, d\tau \tag{6.8}$$

Equation (6.7) shows that the objective function is a *quadratic function* of the initial state, $\mathbf{x}(t)$. Hence, the linear optimal regulator problem posed by Eqs. (6.1)–(6.3) is also called the *linear, quadratic regulator* (LQR) problem. You can easily show from Eq. (6.8) that $\mathbf{M}(t, t_f)$ is a symmetric matrix, i.e. $\mathbf{M}^{\mathrm{T}}(t, t_f) = \mathbf{M}(t, t_f)$, because both $\mathbf{Q}(t)$ and $\mathbf{R}(t)$ are symmetric. On substituting Eq. (6.5) into Eq. (6.6), we can write the objective function as follows:

$$J(t, t_f) = \int_t^{t_f} \mathbf{x}^{\mathbf{T}}(\tau)[\mathbf{Q}(\tau) + \mathbf{K}^{\mathbf{T}}(\tau)\mathbf{R}(\tau)\mathbf{K}(\tau)]\mathbf{x}(\tau)\, d\tau \tag{6.9}$$

On differentiating Eq. (6.9) *partially* with respect to the lower limit of integration, $t$, according to the *Leibniz rule* (see a textbook on integral calculus, such as that by Kreyszig [1]), we get the following:

$$\partial J(t, t_f)/\partial t = -\mathbf{x}^{\mathbf{T}}(t)[\mathbf{Q}(t) + \mathbf{K}^{\mathbf{T}}(t)\mathbf{R}(t)\mathbf{K}(t)]\mathbf{x}(t) \tag{6.10}$$

where $\partial$ denotes *partial differentiation*. Also, partial differentiation of Eq. (6.7) with respect to $t$ results in the following:

$$\partial J(t, t_f)/\partial t = [\mathbf{x}^{(1)}(t)]^T\mathbf{M}(t, t_f)\mathbf{x}(t) + \mathbf{x}^{\mathrm{T}}(t)[\partial\mathbf{M}(t, t_f)/\partial t]\mathbf{x}(t) + \mathbf{x}^{\mathbf{T}}(t)\mathbf{M}(t, t_f)\mathbf{x}^{(1)}(t) \tag{6.11}$$

On substituting $\mathbf{x}^{(1)}(t) = \mathbf{A}_{\mathrm{CL}}(t)\mathbf{x}(t)$ from Eq. (6.4) into Eq. (6.11), we can write

$$\partial J(t, t_f)/\partial t = \mathbf{x}^{\mathrm{T}}(t)[\mathbf{A}_{\mathrm{CL}}^{\mathrm{T}}(t)\mathbf{M}(t, t_f) + \partial \mathbf{M}(t, t_f)/\partial t + \mathbf{M}(t, t_f)\mathbf{A}_{\mathrm{CL}}(t)]\mathbf{x}(t) \quad (6.12)$$

Equations (6.10) and (6.12) are *quadratic forms* for the *same* scalar function, $\partial J(t, t_f)/\partial t$ in terms of the *initial state*, $\mathbf{x}(t)$. Equating Eqs. (6.10) and (6.12), we get the following *matrix differential equation* to be satisfied by $\mathbf{M}(t, t_f)$:

$$-[\mathbf{Q}(t) + \mathbf{K}^{\mathrm{T}}(t)\mathbf{R}(t)\mathbf{K}(t)] = \mathbf{A}_{\mathrm{CL}}^{\mathrm{T}}(t)\mathbf{M}(t, t_f) + \partial \mathbf{M}(t, t_f)/\partial t + \mathbf{M}(t, t_f)\mathbf{A}_{\mathrm{CL}}(t) \quad (6.13)$$

or

$$-\partial \mathbf{M}(t, t_f)/\partial t = \mathbf{A}_{\mathrm{CL}}^{\mathrm{T}}(t)\mathbf{M}(t, t_f) + \mathbf{M}(t, t_f)\mathbf{A}_{\mathrm{CL}}(t) + [\mathbf{Q}(t) + \mathbf{K}^{\mathrm{T}}(t)\mathbf{R}(t)\mathbf{K}(t)] \quad (6.14)$$

Equation (6.14) is a first order, *matrix partial differential equation* in terms of the initial time, $t$, whose solution $\mathbf{M}(t, t_f)$ is given by Eq. (6.8). However, since we do not know the state transition matrix, $\Phi_{\mathrm{CL}}(\tau, t)$, of the general time-varying, closed-loop system, Eq. (6.8) is useless to us for determining $\mathbf{M}(t, t_f)$. Hence, the *only way* to find the unknown matrix $\mathbf{M}(t, t_f)$ is by solving the *matrix differential equation*, Eq. (6.14). We need *only one initial condition* to solve the *first order* matrix differential equation, Eq. (6.14). The simplest initial condition can be obtained by putting $t = t_f$ in Eq. (6.8), resulting in

$$\mathbf{M}(t_f, t_f) = \mathbf{0} \quad (6.15)$$

The linear optimal control problem is thus posed as finding the *optimal regulator gain matrix*, $\mathbf{K}(t)$, such that the solution, $\mathbf{M}(t, t_f)$, to Eq. (6.14) (and hence the objective function, $J(t, t_f)$) is *minimized*, subject to the *initial condition*, Eq. (6.15). The choice of the matrices $\mathbf{Q}(t)$ and $\mathbf{R}(t)$ is left to the designer. However, as we will see below, these two matrices specifying performance objectives and control effort, cannot be arbitrary, but must obey certain conditions.

### 6.1.2 Optimal regulator gain matrix and the Riccati equation

Let us denote the *optimal feedback gain matrix* that *minimizes* $\mathbf{M}(t, t_f)$ by $\mathbf{K}_o(t)$. The *minimum* value of $\mathbf{M}(t, t_f)$ which results from the optimal gain matrix, $\mathbf{K}_o(t)$, is denoted by $\mathbf{M}_o(t, t_f)$, and the minimum value of the objective function is denoted by $J_o(t, t_f)$. For simplicity of notation, let us drop the functional arguments for the time being, and denote $\mathbf{M}(t, t_f)$ by $\mathbf{M}$, $J(t, t_f)$ by $J$, etc. Then, according to Eq. (6.7), the minimum value of the objective function is the following:

$$J_o = \mathbf{x}^T(t)\mathbf{M}_o\mathbf{x}(t) \quad (6.16)$$

Since $J_o$ is the minimum value of $J$ for any initial state, $\mathbf{x}(t)$, we can write $J_o \leq J$, or

$$\mathbf{x}^T(t)\mathbf{M}_o\mathbf{x}(t) \leq \mathbf{x}^T(t)\mathbf{M}\mathbf{x}(t) \quad (6.17)$$

If we express **M** as follows:

$$\mathbf{M} = \mathbf{M}_o + \mathbf{m} \tag{6.18}$$

and substitute Eq. (6.18) into Eq. (6.17), the following condition must be satisfied:

$$\mathbf{x}^T(t)\mathbf{M}_o\mathbf{x}(t) \leq \mathbf{x}^T(t)\mathbf{M}_o\mathbf{x}(t) + \mathbf{x}^T(t)\mathbf{m}\mathbf{x}(t) \tag{6.19}$$

or

$$\mathbf{x}^T(t)\mathbf{m}\mathbf{x}(t) \geq 0 \tag{6.20}$$

A matrix, **m**, which satisfies Eq. (6.20), is called a *positive semi-definite* matrix. Since $\mathbf{x}(t)$ is an *arbitrary* initial state-vector, you can show that according to Eq. (6.20), *all eigenvalues of* **m** must be *greater than or equal to zero.*

It now remains to derive an expression for the optimum regulator gain matrix, $\mathbf{K}_o(t)$, such that **M** is minimized. If $\mathbf{M}_o$ is the minimum value of **M**, then $\mathbf{M}_o$ must satisfy Eq. (6.14) when $\mathbf{K}(t) = \mathbf{K}_o(t)$, i.e.

$$-\partial\mathbf{M}_o/\partial t = \mathbf{A}_{\mathbf{CL}}^{\mathbf{T}}(t)\mathbf{M}_o + \mathbf{M}_o\mathbf{A}_{\mathrm{CL}}(t) + [\mathbf{Q}(t) + \mathbf{K}_o^{\mathbf{T}}(t)\mathbf{R}(t)\mathbf{K}_o(t)] \tag{6.21}$$

Let us express the gain matrix, $\mathbf{K}(t)$, in terms of the optimal gain matrix, $\mathbf{K}_o(t)$, as follows:

$$\mathbf{K}(t) = \mathbf{K}_o(t) + \mathbf{k}(t) \tag{6.22}$$

On substituting Eqs. (6.18) and (6.21) into Eq. (6.14), we can write

$$\begin{aligned} -\partial(\mathbf{M}_o + \mathbf{m})/\partial t = {} & \mathbf{A}_{\mathbf{CL}}^{\mathbf{T}}(t)(\mathbf{M}_o + \mathbf{m}) + (\mathbf{M}_o + \mathbf{m})\mathbf{A}_{\mathrm{CL}}(t) \\ & + [\mathbf{Q}(t) + \{\mathbf{K}_o(t) + \mathbf{k}(t)\}^{\mathbf{T}}\mathbf{R}(t)\{\mathbf{K}_o(t) + \mathbf{k}(t)\}] \end{aligned} \tag{6.23}$$

On subtracting Eq. (6.21) from Eq. (6.23), we get

$$-\partial\mathbf{m}/\partial t = \mathbf{A}_{\mathbf{CL}}^{\mathbf{T}}(t)\mathbf{m} + \mathbf{m}\mathbf{A}_{\mathrm{CL}}(t) + \mathbf{S} \tag{6.24}$$

where

$$\mathbf{S} = [\mathbf{K}_o^{\mathbf{T}}(t)\mathbf{R}(t) - \mathbf{M}_o\mathbf{B}(t)]\mathbf{k}(t) + \mathbf{k}^{\mathbf{T}}(t)[\mathbf{R}(t)\mathbf{K}_o(t) - \mathbf{B}^{\mathbf{T}}(t)\mathbf{M}_o] + \mathbf{k}^{\mathbf{T}}(t)\mathbf{R}(t)\mathbf{k}(t) \tag{6.25}$$

Comparing Eq. (6.24) with Eq. (6.14), we find that the two equations are of the *same form*, with the term $[\mathbf{Q}(t) + \mathbf{K}^{\mathbf{T}}(t)\mathbf{R}(t)\mathbf{K}(t)]$ in Eq. (6.14) replaced by **S** in Eq. (6.24). Since the non-optimal matrix, **M**, in Eq. (6.14) satisfies Eq. (6.8), it must be true that **m** satisfies the following equation:

$$\mathbf{m}(t, t_f) = \int_t^{t_f} \Phi_{\mathbf{CL}}^{\mathbf{T}}(\tau, t)\mathbf{S}(\tau, t_f)\Phi_{\mathrm{CL}}(\tau, t)\, d\tau \tag{6.26}$$

Recall from Eq. (6.20) that **m** must be positive semi-definite. However, Eq. (6.26) requires that for **m** to be positive semi-definite, the matrix **S** given by Eq. (6.25) must be positive semi-definite. Looking at Eq. (6.25), we find that **S** can be positive semi-definite if and only if the *linear terms* in Eq. (6.25) are zeros, i.e. which implies

$$\mathbf{K}_o^T(t)\mathbf{R}(t) - \mathbf{M}_o\mathbf{B}(t) = \mathbf{0} \tag{6.27}$$

or the optimal feedback gain matrix is given by

$$\mathbf{K}_o(t) = \mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{M}_o \tag{6.28}$$

Substituting Eq. (6.28) into Eq. (6.21), we get the following differential equation to be satisfied by the optimal matrix, $\mathbf{M}_o$:

$$-\partial\mathbf{M}_o/\partial t = \mathbf{A}^T(t)\mathbf{M}_o + \mathbf{M}_o\mathbf{A}(t) - \mathbf{M}_o\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{M}_o + \mathbf{Q}(t) \tag{6.29}$$

Equation (6.29) has a special name: the *matrix Riccati equation*. The matrix Riccati equation is special because it's solution, $\mathbf{M}_o$, substituted into Eq. (6.28), gives us the optimal feedback gain matrix, $\mathbf{K}_o(t)$. Exact solutions to the Riccati equation are rare, and in most cases a numerical solution procedure is required. Note that Riccati equation is a first order, nonlinear differential equation, and can be solved by numerical methods similar to those discussed in Chapter 4 for solving the nonlinear state-equations, such as the *Runge–Kutta* method, or other more convenient methods (such as the one we will discuss in Section 6.5). However, in contrast to the state-equation, the solution is a matrix rather than a vector, and the solution procedure has to *march backwards in time*, since the *initial condition* for Riccati equation is specified (Eq. (6.15)) at the *final time*, $t = t_f$, as follows:

$$\mathbf{M}_o(t_f, t_f) = \mathbf{0} \tag{6.30}$$

For this reason, the condition given by Eq. (6.30) is called the *terminal condition* rather than initial condition. Note that the solution to Eq. (6.29) is $\mathbf{M}_o(t, t_f)$ where $t < t_f$. Let us defer the solution to the matrix Riccati equation until Section 6.5.

In summary, the optimal control procedure using full-state feedback consists of specifying an objective function by suitably selecting the performance and control cost weighting matrices, $\mathbf{Q}(t)$ and $\mathbf{R}(t)$, and solving the Riccati equation subject to the terminal condition, in order to determine the full-state feedback matrix, $\mathbf{K}_o(t)$. In most cases, rather than solving the general time-varying optimal control problem, certain simplifications can be made which result in an easier problem, as seen in the following sections.

## 6.2 Infinite-Time Linear Optimal Regulator Design

A large number of control problems are such that the control interval, $(t_f - t)$, is *infinite*. If we are interested in a specific *steady-state* behavior of the control system, we are interested in the response, $\mathbf{x}(t)$, when $t_f \to \infty$, and hence the control interval is *infinite*. The approximation of an infinite control interval results in a simplification in the optimal control problem, as we will see below. For infinite final time, the quadratic objective function can be expressed as follows:

$$J_\infty(t) = \int_t^\infty [\mathbf{x}^T(\tau)\mathbf{Q}(\tau)\mathbf{x}(\tau) + \mathbf{u}^T(\tau)\mathbf{R}(\tau)\mathbf{u}(\tau)]\,d\tau \qquad (6.31)$$

where $J_\infty(t)$ indicates the objective function of the infinite final time (or steady-state) optimal control problem. For the infinite final time, the *backward time integration* of the matrix Riccati equation (Eq. (6.29)), beginning from $\mathbf{M_o}(\infty, \infty) = \mathbf{0}$, would result in a solution, $\mathbf{M_o}(t, \infty)$, which is *either* a *constant*, or *does not converge* to any limit. If the numerical solution to the Riccati equation converges to a constant value, then $\partial\mathbf{M_o}/\partial t = \mathbf{0}$, and the Riccati equation becomes

$$\mathbf{0} = \mathbf{A}^T(t)\mathbf{M_o} + \mathbf{M_o}\mathbf{A}(t) - \mathbf{M_o}\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{M_o} + \mathbf{Q}(t) \qquad (6.32)$$

Note that Eq. (6.32) is no longer a differential equation, but an *algebraic equation*. Hence, Eq. (6.32) is called the *algebraic Riccati equation*. The feedback gain matrix is given by Eq. (6.28), in which $\mathbf{M_o}$ is the (constant) solution to the algebraic Riccati equation. It is (relatively) much easier to solve Eq. (6.32) rather that Eq. (6.29). However, a solution to the algebraic Riccati equation *may not* always exist.

What are the conditions for the existence of the positive semi-definite solution to the algebraic Riccati equation? This question is best answered in a textbook devoted to optimal control, such as that by Bryson and Ho [2], and involves precise mathematical conditions, such as *stabilizability*, *detectability*, etc., for the existence of solution. Here, it suffices to say that for all practical purposes, if *either* the plant is *asymptotically stable*, *or* the plant is *controllable* and *observable* with the output, $\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t)$, where $\mathbf{C}^T(t)\mathbf{C}(t) = \mathbf{Q}(t)$, and $\mathbf{R}(t)$ is a symmetric, *positive definite* matrix, then there is a *unique, positive definite* solution, $\mathbf{M_o}$, to the algebraic Riccati equation. Note that $\mathbf{C}^T(t)\mathbf{C}(t) = \mathbf{Q}(t)$ implies that $\mathbf{Q}(t)$ must be a *symmetric* and *positive semi-definite* matrix. Furthermore, the requirement that the control cost matrix, $\mathbf{R}(t)$, must be *symmetric* and *positive definite* (i.e. *all eigenvalues* of $\mathbf{R}(t)$ must be *positive* real numbers) for the solution, $\mathbf{M_o}$ to be *positive definite* is clear from Eq. (6.25), which implies that $\mathbf{S}$ (and hence $\mathbf{m}$) will be positive definite only if $\mathbf{R}(t)$ is positive definite. Note that these are *sufficient* (but not *necessary*) conditions for the existence of a unique solution to the algebraic Riccati equation, i.e. there may be plants that *do not* satisfy these conditions, and yet there *may exist* a unique, positive definite solution for such plants. A less restrictive set of sufficient conditions for the existence of a unique, positive definite solution to the algebraic Riccati equation is that the plant must be *stabilizable* and *detectable* with the output, $\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t)$, where $\mathbf{C}^T(t)\mathbf{C}(t) = \mathbf{Q}(t)$, and $\mathbf{R}(t)$ is a symmetric, *positive definite* matrix (see Bryson and Ho [2] for details).

While Eq. (6.32) has been derived for linear optimal control of time-varying plants, its usual application is to *time-invariant* plants, for which the algebraic Riccati equation is written as follows:

$$\mathbf{0} = \mathbf{A}^T\mathbf{M_o} + \mathbf{M_o}\mathbf{A} - \mathbf{M_o}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{M_o} + \mathbf{Q} \qquad (6.33)$$

In Eq. (6.33), all the matrices are *constant* matrices. MATLAB contains a solver for the algebraic Riccati equation for time-invariant plants in the M-file named *are.m*. The command *are* is used as follows:

```
>>x = are(a,b,c) <enter>
```

where $\mathbf{a} = \mathbf{A}$, $\mathbf{b} = \mathbf{BR}^{-1}\mathbf{B}^{T}$, $\mathbf{c} = \mathbf{Q}$, in Eq. (6.33), and the returned solution is $\mathbf{x} = \mathbf{M}_o$. For the existence of a unique, positive definite solution to Eq. (6.33), the sufficient conditions remains the same, i.e. the plant with coefficient matrices **A**, **B** must be controllable, **Q** must be symmetric and positive semi-definite, and **R** must be symmetric and positive definite. Another MATLAB function, *ric*, computes the error in solving the algebraic Riccati equation. Alternatively, MATLAB's Control System Toolbox (CST) provides the functions *lqr* and *lqr2* for the solution of the linear optimal control problem with a quadratic objective function, using two different numerical schemes. The command *lqr* (or *lqr2*) is used as follows:

```
>>[Ko,Mo,E]= lqr(A,B,Q,R) <enter>
```

where **A**, **B**, **Q**, **R** are the same as in Eq. (6.33), $\mathbf{Mo} = \mathbf{M}_o$, the returned solution of Eq. (6.33), $\mathbf{Ko} = \mathbf{K}_o = \mathbf{R}^{-1}\mathbf{B}^{T}\mathbf{M}_o$ the returned optimal regulator gain matrix, and **E** is the vector containing the closed-loop eigenvalues (i.e. the eigenvalues of $\mathbf{A}_{CL} = \mathbf{A} - \mathbf{BK}_o$). The command *lqr* (or *lqr2*) is more convenient to use, since it directly works with the plant's coefficient matrices and the weighting matrices. Let us consider a few examples of linear optimal control of time-invariant plants, based upon the solution of the algebraic Riccati equation. (For time-varying plants, the optimal feedback gain matrix can be determined by solving the algebraic Riccati equation at each instant of time, $t$, using either *lqr* or *lqr2* in a time-marching procedure.)

### *Example 6.1*

Consider the longitudinal motion of a flexible bomber aircraft of Example 4.7. The sixth order, two input system is described by the linear, time-invariant, state-space representation given by Eq. (4.71). The inputs are the *desired elevator deflection* (rad.), $u_1(t)$, and the *desired canard deflection* (rad.), $u_2(t)$, while the outputs are the *normal acceleration* (m/s$^2$), $y_1(t)$, and the *pitch-rate* (rad./s), $y_2(t)$. Let us design an optimal regulator which would produce a maximum overshoot of less than $\pm 2$ m/s$^2$ in the normal-acceleration and less than $\pm 0.03$ rad/s in pitch-rate, and a settling time less than 5 s, while requiring elevator and canard deflections *not exceeding* $\pm 0.1$ rad. (5.73°), if the initial condition is 0.1 rad/s perturbation in the pitch-rate, i.e. $\mathbf{x}(0) = [0; 0.1; 0; 0; 0; 0]^T$.

What **Q** and **R** matrices should we choose for this problem? Note that **Q** is a square matrix of size $(6 \times 6)$ and **R** is a square matrix of size $(2 \times 2)$. Examining the plant model given by Eq. (4.71), we find that while the normal acceleration, $y_1(t)$, depends upon all the six state variables, the pitch-rate, $y_2(t)$, is equal to the second state-variable. Since we have to enforce the maximum overshoot limits on $y_1(t)$ and $y_2(t)$, we must, therefore, impose certain limits on the maximum overshoots of all the state variables, which is done by selecting an appropriate state weighting matrix, **Q**. Similarly, the maximum overshoot limits on the two input variables, $u_1(t)$ and $u_2(t)$, must be specified through the control cost matrix, **R**. The settling time would be determined by both **Q** and **R**. *A priori*, we do not quite know what values of **Q** and **R** will produce the desired objectives. Hence, some trial and error is required in selecting the appropriate **Q** and **R**. Let us begin by selecting both **Q** and **R** as *identity matrices*. By doing so, we are specifying that all the six state variables and the two control inputs are *equally important* in the objective function, i.e. it is equally important to bring all the state variables and the control inputs to zero, while minimizing their overshoots. Note that the existence of a unique, positive definite solution to the algebraic Riccati equation will be guaranteed if **Q** and **R** are positive semi-definite and positive definite, respectively, and the plant is controllable. Let us test whether the plant is controllable as follows:

```
>>rank(ctrb(A,B)) <enter>

ans=
  6
```

Hence, the plant is controllable. By choosing $\mathbf{Q} = \mathbf{I}$, and $\mathbf{R} = \mathbf{I}$, we are ensuring that both are positive definite. Therefore, all the sufficient conditions for the existence of an optimal solution are satisfied. For solving the algebraic Riccati equation, let us use the MATLAB command *lqr* as follows:

```
>>[Ko,Mo,E]=lqr(A,B,eye(6),eye(2)) <enter>

Ko=
 3.3571e+000 -4.2509e-001 -6.2538e-001 -7.3441e-001 2.8190e+000  1.5765e+000
 3.8181e+000 1.0274e+000  -5.4727e-001 -6.8075e-001 2.1020e+000  1.8500e+000

Mo=
 1.7429e+000 2.8673e-001  1.1059e-002  -1.4159e-002 4.4761e-002  3.8181e-002
 2.8673e-001 4.1486e-001  1.0094e-002  -2.1528e-003 -5.6679e-003 1.0274e-002
 1.1059e-002 1.0094e-002  1.0053e+000  4.4217e-003  -8.3383e-003 -5.4727e-003
-1.4159e-002 -2.1528e-003 4.4217e-003  4.9047e-003  -9.7921e-003 -6.8075e-003
 4.4761e-002 -5.6679e-003 -8.3383e-003 -9.7921e-003 3.7586e-002  2.1020e-002
 3.8181e-002 1.0274e-002  -5.4727e-003 -6.8075e-003 2.1020e-002  1.8500e-002

E =
-2.2149e+002+2.0338e+002i
-2.2149e+002-2.0338e+002i
-1.2561e+002
-1.8483e+000+1.3383e+000i
-1.8483e+000-1.3383e+000i
-1.0011e+000
```

To see whether this design is acceptable, we calculate the initial response of the closed-loop system as follows:

```
>>sys1=ss(A-B*Ko,zeros(6,2),C,zeros(2,2));<enter>

>>[Y1,t1,X1]=initial(sys1,[0.1 zeros(1,5)]'); u1=-Ko*X1'; <enter>
```

Let us try another design with $\mathbf{Q} = 0.01\mathbf{I}$, and $\mathbf{R} = \mathbf{I}$. As compared with the previous design, we are now specifying that it is 100 times *more important* to minimize the total control energy than minimizing the total transient energy. The new regulator gain matrix is determined by re-solving the algebraic Riccati equation with $\mathbf{Q} = 0.01\mathbf{I}$ and $\mathbf{R} = \mathbf{I}$ as follows:

```
>>[Ko,Mo,E] = lqr(A,B,0.01*eye(6),eye(2)) <enter>

Ko=
 1.0780e+000 -1.6677e-001 -4.6948e-002 -7.5618e-002 5.9823e-001  3.5302e-001
 1.3785e+000 3.4502e-001  -1.3144e-002 -6.5260e-002 4.7069e-001  3.0941e-001

Mo=
 4.1913e-001 1.2057e-001  9.2728e-003  -2.2727e-003 1.4373e-002  1.3785e-002
 1.2057e-001 1.0336e-001  6.1906e-003  -3.9125e-004 -2.2236e-003 3.4502e-003
 9.2728e-003 6.1906e-003  1.0649e-002  9.7083e-005  -6.2597e-004 -1.3144e-004
-2.2727e-003 -3.9125e-004 9.7083e-005  1.7764e-004  -1.0082e-003 -6.5260e-004
 1.4373e-002 -2.2236e-003 -6.2597e-004 -1.0082e-003 7.9764e-003  4.7069e-003
 1.3785e-002 3.4502e-003  -1.3144e-004 -6.5260e-004 4.7069e-003  3.0941e-003

E =
-9.1803e+001
-7.8748e+001+5.0625e+001i
-7.8748e+001-5.0625e+001i
-1.1602e+000+1.7328e+000i
-1.1602e+000-1.7328e+000i
-1.0560e+000
```

The closed-loop state-space model, closed-loop initial response and the required inputs are calculated as follows:

```
>>sys2=ss(A-B*Ko,zeros(6,2),C,zeros(2,2)); <enter>

>>[Y2,t2,X2] = initial(sys2,[0.1 zeros(1,5)]'); u2=-Ko*X2'; <enter>
```

Note that the closed-loop eigenvalues (contained in the returned matrix $\mathbf{E}$) of the first design are *further inside* the left-half plane than those of the second design, which indicates that the first design would have a *smaller* settling time, and a *larger* input requirement when compared to the second design. The resulting outputs, $y_1(t)$ and $y_2(t)$, for the two regulator designs are compared with the plant's initial response to the same initial condition in Figure 6.1.

The plant's oscillating initial response is seen in Figure 6.1 to have maximum overshoots of $-20$ m/s$^2$ and $-0.06$ rad/s, for $y_1(t)$ and $y_2(t)$, respectively, and a settling time exceeding 5 s (actually about 10 s). Note in Figure 6.1 that while the first design ($\mathbf{Q} = \mathbf{I}$, $\mathbf{R} = \mathbf{I}$) produces the closed-loop initial response of $y_2(t)$, $u_1(t)$, and $u_2(t)$ within acceptable limits, the response of $y_1(t)$ displays a maximum overshoot of 10 m/s$^2$ (beginning at $-15$ m/s$^2$ at $t = 0$, and shooting to $-25$ m/s$^2$), which is unacceptable. The settling time of the first design is about 3 s, while that of the second design ($\mathbf{Q} = 0.01\mathbf{I}$, $\mathbf{R} = \mathbf{I}$) is slightly less than 5 s. The second design produces a maximum overshoot of $y_1(t)$ less than 2 m/s$^2$ and that of $y_2(t)$ about $-0.025$ rad/s, which is acceptable. The required control inputs, $u_1(t)$ and $u_2(t)$, for the two designs are plotted in Figure 6.2. While the first design requires a maximum

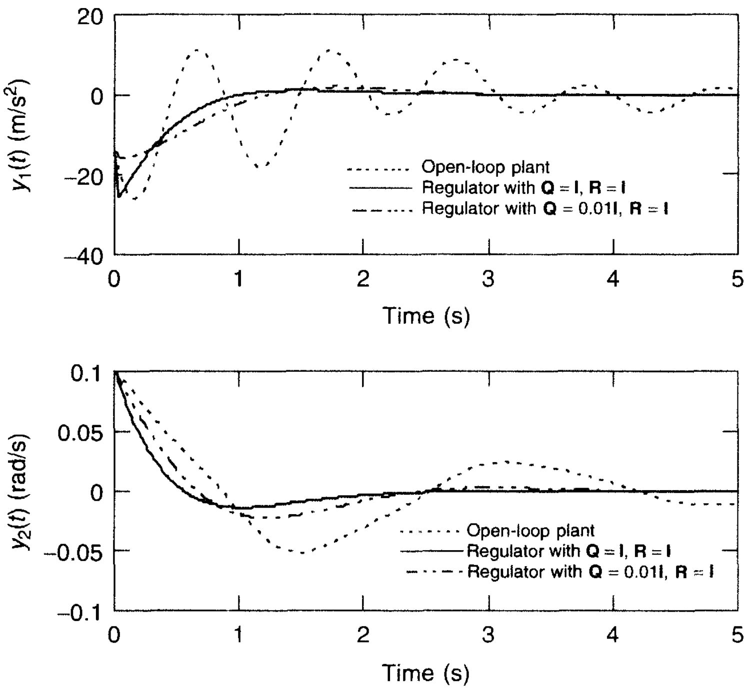


**Figure 6.1** Open and closed-loop initial response of the regulated flexible bomber aircraft, for two optimal regulator designs

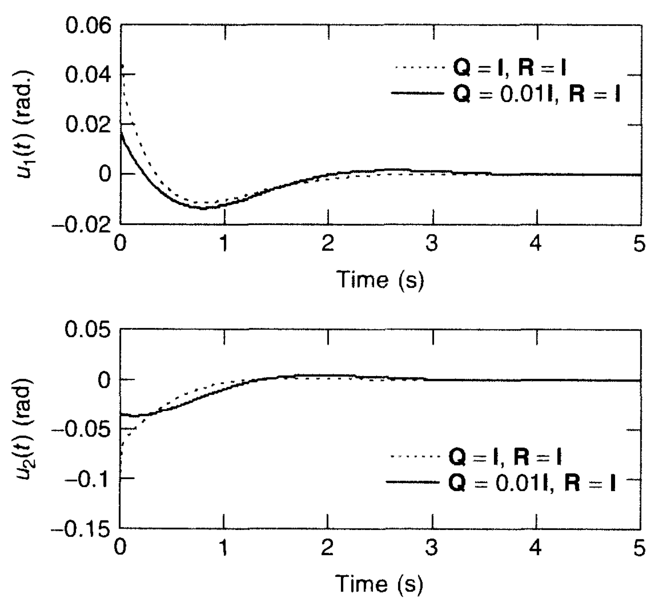


**Figure 6.2** Required initial response control inputs of the regulated flexible bomber aircraft, for two optimal regulator designs

value of elevator deflection, $u_1(t)$, about 0.045 rad., the second design is seen to require a maximum value of $u_1(t)$ less than 0.02 rad. Similarly, the canard deflection, $u_2(t)$, for the second design has a smaller maximum value (−0.04 rad) than that of the first design (−0.1 rad.). Hence, the second design fulfills all the design objectives.

In Example 6.1, we find that the total transient energy is *more sensitive* to the *settling time*, than the maximum overshoot. Recall from Chapter 5 that if we try to reduce the settling time, we have to accept an increase in the maximum overshoot. Conversely, to reduce the maximum overshoot of $y_1(t)$, which depends upon all the state variables, we must allow an increase in the settling time, which is achieved in the second design by reducing the importance of minimizing the transient energy by hundred-fold, as compared to the first design. Let us now see what effect a measurement noise will have on the closed-loop initial response. We take the second regulator design (i.e. $\mathbf{Q} = 0.01\mathbf{I}$, $\mathbf{R} = \mathbf{I}$) and simulate the initial response assuming a random error (i.e. measurement noise) in feeding back the pitch-rate (the second state-variable of the plant). The simulation is carried out using SIMULINK block-diagram shown in Figure 6.3, where the measurement noise is simulated by the *band-limited white noise* block with a *power* parameter of $10^{-4}$. Note the

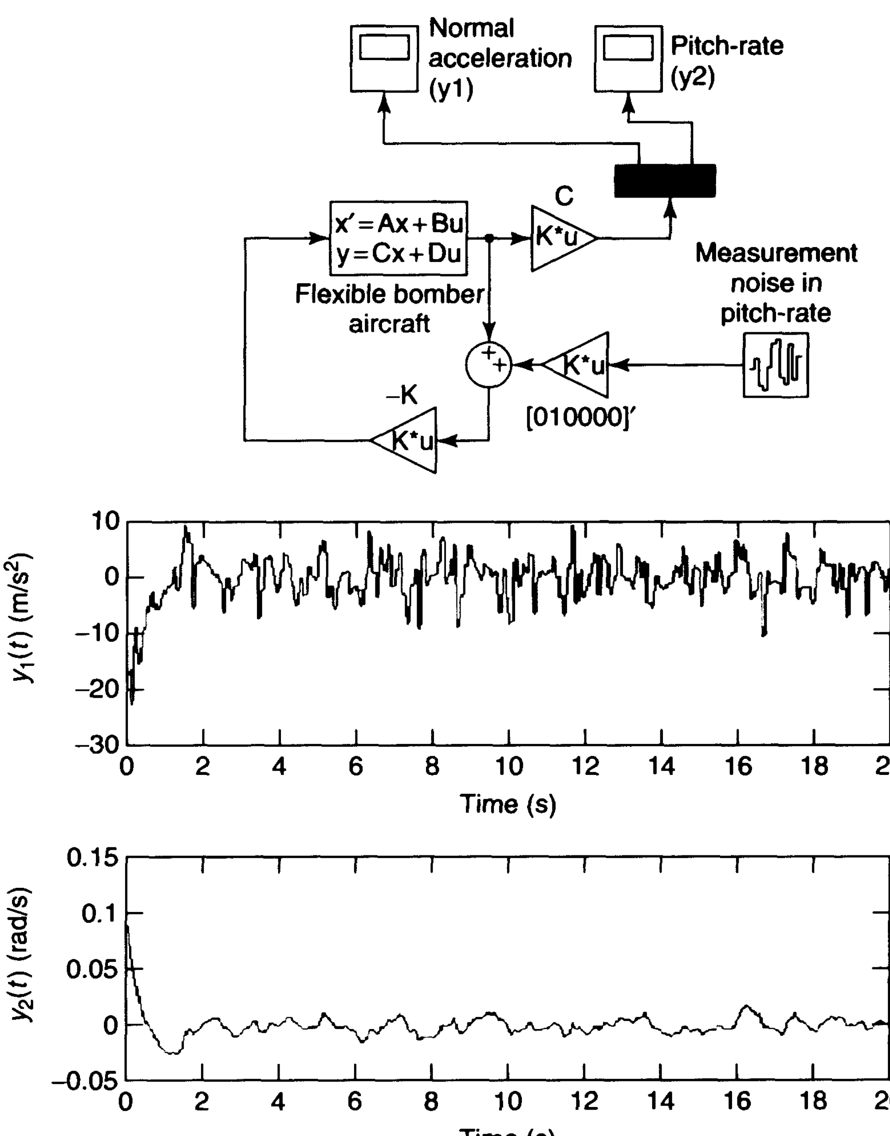


**Figure 6.3** Simulation of initial response of the flexible bomber with a full-state feedback regulator and measurement noise in the pitch-rate channel

manner in which the noise is added to the feedback loop through the *matrix gain* block. The simulated initial response is also shown in Figure 6.3. Note the random fluctuations in both normal acceleration, $y_1(t)$, and pitch-rate, $y_2(t)$. The aircraft crew are likely to have a rough ride due to large sustained fluctuations ($\pm 10$ m/s$^2$) in normal acceleration, $y_1(t)$, resulting from the small measurement noise! The feedback loop results in an amplification of the measurement noise. If the elements of the feedback gain matrix, **K**, corresponding to pitch-rate are reduced in magnitude then the noise due to pitch-rate feedback will be alleviated. Alternatively, pitch-rate (or any other state-variable that is noisy) can be removed from state feedback, with the use of an observer based compensator that feeds back only selected state-variables (see Chapter 5).

### *Example 6.2*

Let us design an optimal regulator for the flexible, rotating spacecraft shown in Figure 6.4. The spacecraft consists of a *rigid hub* and four *flexible appendages*, each having a *tip mass*, with three *torque* inputs, $u_1(t)$, $u_2(t)$, $u_3(t)$, and three *angular rotation* outputs in rad., $y_1(t)$, $y_2(t)$, $y_3(t)$. Due to the flexibility of the appendages, the spacecraft is a *distributed parameter* system (see Chapter 1). However, it is approximated by a *lumped parameter*, linear, time-invariant state-space representation using a *finite-element model* [3]. The order of the spacecraft can be reduced to 26 for accuracy in a desired frequency range [4]. The 26th order state-vector, $\mathbf{x}(t)$, of the spacecraft consists of the angular displacement, $y_1(t)$, and *angular velocity* of the rigid hub, combined with individual *transverse* (i.e. *perpendicular* to the appendage) *displacements* and *transverse velocities* of three points on each appendage. The state-coefficient matrices of the spacecraft are given as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{-1}\mathbf{K} & \mathbf{0} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1}\mathbf{d} \end{bmatrix} \quad \mathbf{C} = [\,\mathbf{d}^T; \quad \mathbf{0}\,]; \quad \mathbf{D} = \mathbf{0} \qquad (6.34)$$

where **M, K**, and **d** are the *mass*, *stiffness*, and *control influence* matrices, given in Appendix C.

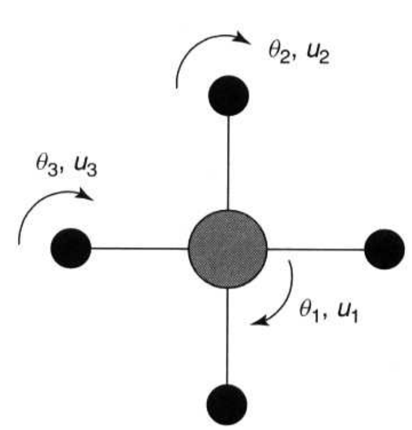


**Figure 6.4** A rotating, flexible spacecraft with three inputs, $(u_1, u_2, u_3)$, and three outputs $(\theta_1, \theta_2, \theta_3)$

The eigenvalues of the spacecraft are the following:

```
>>damp(A) <enter>

Eigenvalue                    Damping         Freq.(rad/sec)
9.4299e-013+7.6163e+003i     -6.1230e-017    7.6163e+003
9.4299e-013-7.6163e+003i     -6.1230e-017    7.6163e+003
8.1712e-013+4.7565e+004i     -6.1230e-017    4.7565e+004
8.1712e-013-4.7565e+004i     -6.1230e-017    4.7565e+004
8.0539e-013+2.5366e+003i     -2.8327e-016    2.5366e+003
8.0539e-013-2.5366e+003i     -2.8327e-016    2.5366e+003
7.4867e-013+4.7588e+004i     -6.1230e-017    4.7588e+004
7.4867e-013-4.7588e+004i     -6.1230e-017    4.7588e+004
7.1276e-013+2.5982e+004i     -6.1230e-017    2.5982e+004
7.1276e-013-2.5982e+004i     -6.1230e-017    2.5982e+004
5.2054e-013+1.4871e+004i     -6.1230e-017    1.4871e+004
5.2054e-013-1.4871e+004i     -6.1230e-017    1.4871e+004
4.8110e-013+2.0986e+002i     -2.2817e-015    2.0986e+002
4.8110e-013-2.0986e+002i     -2.2817e-015    2.0986e+002
4.4812e-013+2.6009e+004i     -6.1230e-017    2.6009e+004
4.4812e-013-2.6009e+004i     -6.1230e-017    2.6009e+004
3.1387e-013+7.5783e+003i     -6.1230e-017    7.5783e+003
3.1387e-013-7.5783e+003i     -6.1230e-017    7.5783e+003
2.4454e-013+3.7952e+002i     -7.2736e-016    3.7952e+002
2.4454e-013-3.7952e+002i     -7.2736e-016    3.7952e+002
     0                       -1.0000e+000    0
     0                       -1.0000e+000    0
-9.9504e-013+2.4715e+003i    3.8286e-016     2.4715e+003
-9.9504e-013-2.4715e+003i    3.8286e-016     2.4715e+003
-1.1766e-012+1.4892e+004i    1.6081e-016     1.4892e+004
-1.1766e-012-1.4892e+004i    1.6081e-016     1.4892e+004
```

Clearly, the spacecraft is *unstable* due to a pair of zero eigenvalues (we can ignore the negligible, positive real parts of some eigenvalues, and assume that those real parts are zeros). The natural frequencies of the spacecraft range from 0 to 47 588 rad/s. The nonzero natural frequencies denote structural vibration of the spacecraft. The control objective is to design a controller which stabilizes the spacecraft, and brings the transient response to zero within 5 s, with zero maximum overshoot, while requiring input torques not exceeding 0.1 N-m, when the spacecraft is initially perturbed by a hub rotation of 0.01 rad. due to the movement of astronauts. The initial condition corresponding to the initial perturbation caused by the astronauts' movement is $\mathbf{x}(0) = [0.01;\ \text{zeros}(1,25)]^T$. Let us see whether the spacecraft is controllable:

```
>>rank(ctrb(A,B)) <enter>

ans=

  6
```

Since the rank of the controllability test matrix is *less than* 26, the order of the plant, it follows that the spacecraft is *uncontrollable*. The *uncontrollable* modes are the structural vibration modes, while the *unstable* mode is the rigid-body rotation with zero natural frequency. Hence, the spacecraft is *stabilizable* and an optimal regulator can be designed for the spacecraft, since stabilizability of the plant is a sufficient condition for the existence of a unique, positive definite solution to the algebraic Riccati equation. Let us select $\mathbf{Q} = 200\mathbf{I}$, and $\mathbf{R} = \mathbf{I}$, noting that the size of $\mathbf{Q}$ is $(26 \times 26)$ while that of $\mathbf{R}$ is $(3 \times 3)$, and solve the Riccati equation using *lqr* as follows:

```
>>[Ko,Mo,E] = lqr(A,B,200*eye(26),eye(3)); <enter>
```

A positive definite solution to the algebraic Riccati equation *exists* for the present choice of **Q** and **R**, *even though the plant is uncontrollable*. Due to the size of the plant, we avoid printing the solution, **Mo**, and the optimal feedback gain matrix, **Ko**, here, but the closed-loop eigenvalues, **E**, are the following:

```
E =
-1.7321e+003+4.7553e+004i
-1.7321e+003-4.7553e+004i
-1.7502e+003+4.7529e+004i
-1.7502e+003-4.7529e+004i
-1.8970e+003+2.5943e+004i
-1.8970e+003-2.5943e+004i
-1.8991e+003+2.5916e+004i
-1.8991e+003-2.5916e+004i
-1.8081e+003+1.4569e+004i
-1.8081e+003-1.4569e+004i
-1.8147e+003+1.4550e+004i
-1.8147e+003-1.4550e+004i
-7.3743e+002+7.6536e+003i
-7.3743e+002-7.6536e+003i
-7.3328e+002+7.6142e+003i
-7.3328e+002-7.6142e+003i
-2.6794e+002+2.5348e+003i
-2.6794e+002-2.5348e+003i
-2.5808e+002+2.4698e+003i
-2.5808e+002-2.4698e+003i
-3.9190e+001+3.7744e+002i
-3.9190e+001-3.7744e+002i
-1.1482e+000+4.3165e-001i
-1.1482e+000-4.3165e-001i
-1.8066e+001+2.0911e+002i
-1.8066e+001-2.0911e+002i
```

All the closed-loop eigenvalues (contained in the vector **E**) have negative real-parts, indicating that the closed-loop system is asymptotically stable, which is a bonus! Let us check whether the performance objectives are met by this design by calculating the closed-loop initial response as follows:

```
>>sysCL=ss(A-B*Ko,zeros(26,3),C,D); <enter>

>>[y,t,X] = initial(sysCL, [0.01 zeros(1, 25)]'); u = -Ko*X'; <enter>
```

The closed-loop outputs, $y_1(t)$, $y_2(t)$, and $y_3(t)$, and the required torque inputs, $u_1(t)$, $u_2(t)$, and $u_3(t)$, are plotted in Figure 6.5. We see from Figure 6.5 that all the three outputs settle to zero in 5 s, with zero overshoot, and that the input torque magnitudes are smaller than 0.1 N-m, as desired. Therefore, our design is successful in meeting all the performance objectives and input effort limits.

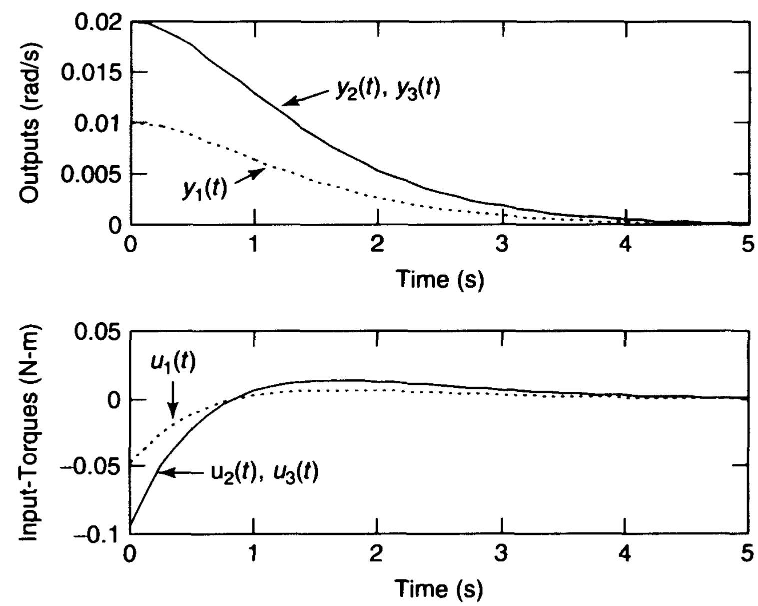


**Figure 6.5** Closed-loop initial response and required inputs of the regulated flexible spacecraft for optimal regulator designed with $\mathbf{Q} = 200\mathbf{I}$ and $\mathbf{R} = \mathbf{I}$

Examples 6.1 and 6.2 illustrate two practical problems, one with a controllable plant, and the other with an uncontrollable plant, which are successfully solved using linear optimal control. Note the ease with which the two multi-input problems are solved when compared with the pole-placement approach (which, as you will recall, would have resulted in far too many design parameters than can be fixed by pole-placement). We will now consider the application of optimal control to more complicated problems.

## 6.3 Optimal Control of Tracking Systems

Consider a linear, time-varying plant with state-equation given by Eq. (6.1). It is required to design a tracking system for this plant if the desired state-vector, $\mathbf{x_d}(t)$, is the solution of the following equation:

$$\mathbf{x_d}^{(1)}(t) = \mathbf{A_d}(t)\mathbf{x_d}(t) \tag{6.35}$$

Recall from Chapter 5 that the desired state dynamics is described by a homogeneous state-equation, because $\mathbf{x_d}(t)$ it is unaffected by the input, $\mathbf{u}(t)$. Subtracting Eq. (6.1) from Eq. (6.35), we get the following state-equation for the tracking-error, $\mathbf{e}(t) = \mathbf{x_d}(t) - \mathbf{x}(t)$:

$$\mathbf{e}^{(1)}(t) = \mathbf{A}(t)\mathbf{e}(t) + [\mathbf{A_d}(t) - \mathbf{A}(t)]\mathbf{x_d}(t) - \mathbf{B}(t)\mathbf{u}(t) \tag{6.36}$$

The control objective is to find the control input, $\mathbf{u}(t)$, such that the tracking-error, $\mathbf{e}(t)$, is brought to zero in the steady-state. To achieve this objective by optimal control, we have to first define the objective function to be minimized. Note that, as opposed to the regulator problem in which the input $\mathbf{u}(t) = -\mathbf{K}(t)\mathbf{x}(t)$, now the control input will also depend linearly on the desired state-vector, $\mathbf{x_d}(t)$. If we express Eqs. (6.1) and (6.35) by a *combined system* of which the state-vector is $\mathbf{x_c}(t) = [\mathbf{e}(t)^T; \mathbf{x_d}(t)^T]^T$, then the control input must be given by the following linear control-law:

$$\mathbf{u}(t) = -\mathbf{K_c}(t)\mathbf{x_c}(t) = -\mathbf{K_c}(t)[\mathbf{e}(t)^T; \mathbf{x_d}(t)^T]^T \tag{6.37}$$

where $\mathbf{K_c}(t)$ is the *combined feedback gain* matrix. Note that Eqs. (6.35) and (6.36) can be expressed as the following *combined state-equation*:

$$\mathbf{x_c}^{(1)}(t) = \mathbf{A_c}(t)\mathbf{x_c}(t) + \mathbf{B_c}(t)\mathbf{u}(t) \tag{6.38}$$

where

$$\mathbf{A_c}(t) = \begin{bmatrix} \mathbf{A}(t) & [\mathbf{A_d}(t) - \mathbf{A}(t)] \\ \mathbf{0} & \mathbf{A_d}(t) \end{bmatrix}; \quad \mathbf{B_c}(t) = \begin{bmatrix} -\mathbf{B}(t) \\ \mathbf{0} \end{bmatrix} \tag{6.39}$$

Since Eqs. (6.1) and (6.2) are now replaced by Eqs. (6.38) and (6.37), respectively, the objective function for the *combined system* can be expressed as an extension of Eq. (6.3) as follows:

$$J(t, t_f) = \int_t^{t_f} [\mathbf{x_c}^{\mathrm{T}}(\tau)\mathbf{Q_c}(\tau)\mathbf{x_c}(\tau) + \mathbf{u}^{\mathrm{T}}(\tau)\mathbf{R}(\tau)\mathbf{u}(\tau)]\, d\tau \tag{6.40}$$

Note that although we desire that the tracking error, $\mathbf{e}(t) = \mathbf{x_d}(t) - \mathbf{x}(t)$, be reduced to zero in the steady-state (i.e. when $t \to \infty$), we *cannot* pose the tracking system design as an optimal control problem with *infinite* control interval (i.e. $t_f = \infty$). The reason is that the desired state-vector, $\mathbf{x_d}(t)$ (hence $\mathbf{x_c}(t)$), *may not* go to zero in the steady-state, and thus a *non-zero* control input, $\mathbf{u}(t)$, may be required in the steady-state. Also, note that the combined system described by Eq. (6.38) is *uncontrollable*, because the desired state dynamics given by Eq. (6.35) is unaffected by the input, $\mathbf{u}(t)$. Therefore, the combined system's optimal control problem, represented by Eqs. (6.37)–(6.40) is *not guaranteed* to have a unique, positive definite solution. Hence, to have a guaranteed unique, positive definite solution to the optimal control problem, let us *exclude* the uncontrollable desired state-vector from the objective function, by choosing the *combined state-weighting matrix* as follows:

$$\mathbf{Q_c}(t) = \begin{bmatrix} \mathbf{Q}(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{6.41}$$

which results in the following objective function:

$$J(t, t_f) = \int_t^{t_f} [\mathbf{e}^T(\tau)\mathbf{Q}(\tau)\mathbf{e}(\tau) + \mathbf{u}^T(\tau)\mathbf{R}(\tau)\mathbf{u}(\tau)]\, d\tau \tag{6.42}$$

which is the same as Eq. (6.3), with the crucial difference that $\mathbf{u}(t)$ in Eq. (6.42) is given by Eq. (6.37), rather than Eq. (6.2). By choosing $\mathbf{Q}(t)$ and $\mathbf{R}(t)$ to be positive semi-definite and positive definite, respectively, we satisfy the remaining sufficient conditions for the existence of a unique, positive definite solution to the optimal control problem. Note that the *optimal feedback gain matrix*, $\mathbf{K}_{oc}(t)$, is given by the following extension of Eq. (6.28):

$$\mathbf{K}_{oc}(t) = \mathbf{R}^{-1}(t)\mathbf{B}_c^T(t)\mathbf{M}_{oc} \tag{6.43}$$

where $\mathbf{M}_{oc}$ is the solution to the following Riccati equation:

$$-\partial\mathbf{M}_{oc}/\partial t = \mathbf{A}_c^T(t)\mathbf{M}_{oc} + \mathbf{M}_{oc}\mathbf{A}_c(t) - \mathbf{M}_{oc}\mathbf{B}_c(t)\mathbf{R}^{-1}(t)\mathbf{B}_c^T(t)\mathbf{M}_{oc} + \mathbf{Q}_c(t) \tag{6.44}$$

subject to the terminal condition, $\mathbf{M}_{oc}(t_f, t_f) = \mathbf{0}$. Since $\mathbf{M}_{oc}$ is symmetric (see Section 6.1), it can be expressed as

$$\mathbf{M}_{oc} = \begin{bmatrix} \mathbf{M}_{o1} & \mathbf{M}_{o2} \\ \mathbf{M}_{o2}^T & \mathbf{M}_{o3} \end{bmatrix} \tag{6.45}$$

where $\mathbf{M}_{o1}$ and $\mathbf{M}_{o2}$ correspond to the plant and the desired state dynamics, respectively. Substituting Eqs. (6.45) and (6.39) into Eq. (6.43), we can express the optimal feedback gain matrix as follows:

$$\mathbf{K}_{oc}(t) = -[\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{M}_{o1}; \quad \mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{M}_{o2}] \tag{6.46}$$

and the optimal control input is thus obtained by substituting Eq. (6.46) into Eq. (6.37) as follows:

$$\mathbf{u}(t) = \mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{M}_{o1}\mathbf{e}(t) + \mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{M}_{o2}\mathbf{x}_d(t) \tag{6.47}$$

Note that Eq. (6.47) does not require the sub-matrix, $\mathbf{M}_{o3}$. The individual matrix differential equations to be solved for $\mathbf{M}_{o1}$ and $\mathbf{M}_{o2}$ can be obtained by substituting Eqs. (6.39) and (6.45) into Eq. (6.44) as follows:

$$-\partial\mathbf{M}_{o1}/\partial t = \mathbf{A}^T(t)\mathbf{M}_{o1} + \mathbf{M}_{o1}\mathbf{A}(t) - \mathbf{M}_{o1}\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{M}_{o1} + \mathbf{Q}(t) \tag{6.48}$$

$$\begin{aligned} -\partial\mathbf{M}_{o2}/\partial t &= \mathbf{M}_{o2}\mathbf{A}_d(t) + \mathbf{M}_{o1}[\mathbf{A}_d(t) - \mathbf{A}(t)] \\ &\quad + [\mathbf{A}^T(t) - \mathbf{M}_{o1}\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)]\mathbf{M}_{o2} \end{aligned} \tag{6.49}$$

Note that Eq. (6.48) is identical to the *matrix Riccati equation*, Eq. (6.29), which can be solved *independently* of Eq. (6.49), without taking into account the desired state dynamics. Once the optimal matrix, $\mathbf{M}_{o1}$, is obtained from the solution of Eq. (6.48), it can be substituted into Eq. (6.49), which can then be solved for $\mathbf{M}_{o2}$. Equation (6.49) is a *linear, matrix differential equation*, and can be written as follows:

$$-\partial\mathbf{M}_{o2}/\partial t = \mathbf{M}_{o2}\mathbf{A}_d(t) + \mathbf{M}_{o1}[\mathbf{A}_d(t) - \mathbf{A}(t)] + \mathbf{A}_{CL}^T(t)\mathbf{M}_{o2} \tag{6.50}$$

where $\mathbf{A}_{CL}(t) = \mathbf{A}(t) - \mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{M}_{o1}$, the closed-loop state-dynamics matrix. The solution of the optimal tracking system thus requires the solution of the linear differential equation, Eq. (6.50), in addition to the solution of the optimal regulator problem given by Eq. (6.48). The solutions of Eqs. (6.48) and (6.49) are subject to the terminal condition, $\mathbf{M}_{oc}(t_f, t_f) = \mathbf{0}$, which results in $\mathbf{M}_{o1}(t_f, t_f) = \mathbf{0}$, and $\mathbf{M}_{o2}(t_f, t_f) = \mathbf{0}$.

Often, it is required to track a *constant* desired state-vector, $\mathbf{x}_d(t) = \mathbf{x}_d^c$, which implies $\mathbf{A}_d(t) = \mathbf{0}$. Then the matrices $\mathbf{M}_{o1}$ and $\mathbf{M}_{o2}$ are both constants in the *steady-state* (i.e. $t_f \to \infty$), and are the solutions of the following *steady-state* equations (obtained by setting $\partial\mathbf{M}_{o1}/\partial t = \partial\mathbf{M}_{o2}/\partial t = \mathbf{0}$ and $\mathbf{A}_d(t) = \mathbf{0}$ in Eqs. (6.48) and (6.49)):

$$\mathbf{0} = \mathbf{A}^T(t)\mathbf{M}_{o1} + \mathbf{M}_{o1}\mathbf{A}(t) - \mathbf{M}_{o1}\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{M}_{o1} + \mathbf{Q}(t) \tag{6.51}$$

$$\mathbf{0} = -\mathbf{M}_{o1}\mathbf{A}(t) + \mathbf{A}_{CL}^T(t)\mathbf{M}_{o2} \tag{6.52}$$

We immediately recognize Eq. (6.51) as the *algebraic Riccati equation* (Eq. (6.32)) of the *steady-state, optimal regulator* problem. From Eq. (6.52), we must have

$$\mathbf{M}_{o2} = [\mathbf{A}_{CL}^T(t)]^{-1}\mathbf{M}_{o1}\mathbf{A}(t) \tag{6.53}$$

where $\mathbf{M}_{o1}$ is the solution to the algebraic Riccati equation, Eq. (6.51). Note, however, that even though $\mathbf{M}_{o1}$ and $\mathbf{M}_{o2}$ are finite constants in the steady-state, the matrix $\mathbf{M}_{oc}$ is *not* a finite constant in the steady-state, because as $\mathbf{x}(t_f)$ tends to a constant desired state in the limit $t_f \to \infty$, the objective function (Eq. (6.42)) becomes *infinite*, hence a steady-state solution to Eq. (6.44) does not exist. The only way $\mathbf{M}_{oc}$ can *not* be a finite constant (when both $\mathbf{M}_{o1}$ and $\mathbf{M}_{o2}$ are finite constants) is when $\mathbf{M}_{o3}$ (the discarded matrix in Eq. (6.45)) is *not* a finite constant in the steady-state.

Substituting Eq. (6.53) into Eq. (6.47), we get the following input for the *constant* desired state vector, $\mathbf{x}_d^c$:

$$\mathbf{u}(t) = \mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{M}_{o1}\mathbf{e}(t) + \mathbf{R}^{-1}(t)\mathbf{B}^T(t)[\mathbf{A}_{CL}^T(t)]^{-1}\mathbf{M}_{o1}\mathbf{A}(t)\mathbf{x}_d^c \tag{6.54}$$

Substituting Eq. (6.54) into Eq. (6.36), we get the following closed-loop tracking error state-equation with $\mathbf{A}_d(t) = \mathbf{0}$ and $\mathbf{x}_d(t) = \mathbf{x}_d^c$:

$$\mathbf{e}^{(1)}(t) = \mathbf{A}_{CL}(t)\mathbf{e}(t) - [\mathbf{A}(t) + \mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\{\mathbf{A}_{CL}^T(t)\}^{-1}\mathbf{M}_{o1}\mathbf{A}(t)]\mathbf{x}_d^c \tag{6.55}$$

From Eq. (6.55), it is clear that the tracking error can go to *zero* in the steady-state (i.e. as $t \to \infty$) for *any non-zero, constant* desired state, $\mathbf{x}_d^c$, if $\mathbf{A}_{CL}(t)$ is *asymptotically stable* and

$$\mathbf{A}(t) + \mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\{\mathbf{A}_{CL}^T(t)\}^{-1}\mathbf{M}_{o1}\mathbf{A}(t) = \mathbf{0} \tag{6.56}$$

or

$$\mathbf{M}_{o1}\mathbf{A}(t) = -\left[\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\{\mathbf{A}_{CL}^T(t)\}^{-1}\right]^{-1}\mathbf{A}(t) = -\mathbf{A}_{CL}^T(t)\{\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\}^{-1}\mathbf{A}(t) \tag{6.57}$$

Equation (6.57) can be expanded to give the following equation to be satisfied by $\mathbf{M}_{o1}$ for the steady-state tracking error to be zero:

$$\mathbf{M}_{o1}\mathbf{A}(t) = -\mathbf{A}^T(t)\{\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\}^{-1}\mathbf{A}(t) + \mathbf{M}_{o1}\mathbf{A}(t) \tag{6.58}$$

which implies that $\mathbf{A}^T(t)\{\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\}^{-1}\mathbf{A}(t) = \mathbf{0}$, or $\{\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\}^{-1} = \mathbf{0}$. Clearly, this is an *impossible* requirement, because it implies that $\mathbf{R}(t) = \mathbf{0}$. Hence, we *cannot* have an optimal tracking system in which the tracking error, $\mathbf{e}(t)$, goes to zero in the steady-state for *any* constant desired state-vector, $\mathbf{x}_d^c$. As in Chapter 5, the best we can do is to have $\mathbf{e}(t)$ going to zero for *some* values of $\mathbf{x}_d^c$. However, if we want this to happen *while satisfying* the optimality condition for $\mathbf{M}_{o2}$ given by Eq. (6.53), we will be left with the requirement that $[\mathbf{A}(t) + \mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\{\mathbf{A}_{CL}^T(t)\}^{-1}\mathbf{M}_{o1}\mathbf{A}(t)]\mathbf{x}_d^c = \mathbf{0}$ for some non-zero $\mathbf{x}_d^c$, resulting in $\{\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\}^{-1}\mathbf{A}(t)\mathbf{x}_d^c = \mathbf{0}$, which implies that $\{\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\}^{-1}$ must be a *singular matrix* – again, an impossible requirement. Therefore, the only possible way we can ensure that the tracking error goes to zero for some desired state is by *dropping the optimality condition* on $\mathbf{M}_{o2}$ given by Eq. (6.53). Then we can write the input vector as follows:

$$\mathbf{u}(t) = \mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{M}_{o1}\mathbf{e}(t) - \mathbf{K}_d(t)\mathbf{x}_d^c \tag{6.59}$$

where $\mathbf{K}_d(t)$ is the (non-optimal) feedforward gain matrix which would make $\mathbf{e}(t)$ zero in the steady-state for some values of $\mathbf{x}_d^c$. Substituting Eq. (6.59) into Eq. (6.36) we get the following state-equation for the tracking error:

$$\mathbf{e}^{(1)}(t) = \mathbf{A}_{CL}(t)\mathbf{e}(t) - [\mathbf{A}(t) - \mathbf{B}(t)\mathbf{K}_d(t)]\mathbf{x}_d^c \tag{6.60}$$

Equation (6.60) implies that for a zero steady-state tracking error, $\mathbf{K}_d(t)$ must be selected such that $[\mathbf{A}(t) - \mathbf{B}(t)\mathbf{K}_d(t)]\mathbf{x}_d^c = \mathbf{0}$. The closed-loop state-dynamics matrix, $\mathbf{A}_{CL}$, in Eq. (6.60) is an optimal matrix given by $\mathbf{A}_{CL}(t) = \mathbf{A}(t) - \mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{M}_{o1}$, where $\mathbf{M}_{o1}$ is the solution to the algebraic Riccati equation, Eq. (6.51). Hence, the design of a tracking system *does not* end with finding a unique, positive definite solution, $\mathbf{M}_{o1}$, to the algebraic Riccati equation (which would make $\mathbf{A}_{CL}(t)$ asymptotically stable); we should also find a (non-optimal) feedforward gain matrix, $\mathbf{K}_d(t)$, such that $[\mathbf{A}(t) - \mathbf{B}(t)\mathbf{K}_d(t)]\mathbf{x}_d^c = \mathbf{0}$ for some values of the constant desired state-vector, $\mathbf{x}_d^c$. Note that if the plant has *as many inputs* as there are state variables, then $\mathbf{B}(t)$ is a square matrix, and it would be possible to make $\mathbf{e}(t)$ zero in the steady-state for *any arbitrary* $\mathbf{x}_d^c$, by choosing $\mathbf{K}_d(t) = \mathbf{B}^{-1}(t)\mathbf{A}(t)$ (provided $\mathbf{B}(t)$ is non-singular, i.e. the plant is *controllable*.)

### *Example 6.3*

Consider the amplifier-motor of Example 3.7, with the numerical values given as $J = 1$ kg.m$^2$, $R = 1000$ ohms, $L = 100$ henry, $a = 0.3$ kg.m$^2$/s$^2$/Ampere, and $K_A = 10$. Recall from Example 3.7 that the state-vector of the amplifier-motor is $\mathbf{x}(t) = [\theta(t); \theta^{(1)}(t); i(t)]^T$, where $\theta(t)$ is the angular position of the load on the motor, and $i(t)$ is the current supplied to the motor. The input vector is $\mathbf{u}(t) = [v(t); T_L(t)]^T$, where $v(t)$ is the input voltage to the amplifier and $T_L(t)$ is the torque applied by the load on the motor. It is desired to design a tracking system such that the load on the motor moves from an initial angular position, $\theta(0) = 0$, to desired angular position $\theta_d(t) = 0.1$ rad. in about six seconds, and comes to rest at the desired position. The maximum angular velocity of the load, $\theta^{(1)}(t)$, should not exceed 0.05 rad/s. After the load comes to rest at the desired position, the current supplied to the motor should be zero. The desired state-vector is thus $\mathbf{x}_\mathbf{d}^\mathbf{c} = [0.1; 0; 0]^T$. The plant's state coefficient matrices are the following:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0.01 & 0.3 \\ 0 & -0.003 & -10 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & -1 \\ 0.1 & 0 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}; \quad \mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{6.61}$$

The eigenvalues of the linear, time-invariant plant are calculated as follows:

```
>>damp(A) <enter>

Eigenvalue   Damping   Freq. (rad/sec)
    0        -1.0000   0
-0.0101       1.0000   0.0101
-9.9999       1.0000   9.9999
```

The plant is stable, with an eigenvalue at the origin. Since the plant is time-invariant, the controller gain matrices must be constants. Let us first find the optimal feedback gain matrix, $\mathbf{K_{o1}} = -\mathbf{R}^{-1}\mathbf{B}^\mathbf{T}\mathbf{M_{o1}}$ by choosing $\mathbf{Q} = \mathbf{I}$ and $\mathbf{R} = \mathbf{I}$, and solving the algebraic Riccati equation as follows:

```
>>[Ko1,Mo1,E] = lqr(A,B,eye(3),eye(2)) <enter>

Ko1 =
  0.0025   0.0046   0.0051
 -1.0000  -1.7220  -0.0462

Mo1 =
 1.7321   1.0000   0.0254
 1.0000   1.7220   0.0462
 0.0254   0.0462   0.0513

E =
-10.0004
-0.8660 + 0.5000i
-0.8660 - 0.5000i
```

The closed-loop eigenvalues are all in the left-half plane, as desired for asymptotic stability of the tracking error dynamics. Next, we calculate the feedforward gain matrix, $\mathbf{K_d}$, which will make the steady-state tracking error zero for the specified constant desired state, $\mathbf{x}_\mathbf{d}^\mathbf{c}$. This is done by selecting $\mathbf{K_d}$ such that $\mathbf{A}\mathbf{x}_\mathbf{d}^\mathbf{c} = \mathbf{B}\mathbf{K_d}\mathbf{x}_\mathbf{d}^\mathbf{c}$ as follows:

$$\mathbf{K_d} = \begin{bmatrix} K_{d1} & K_{d2} & K_{d3} \\ K_{d4} & K_{d5} & K_{d6} \end{bmatrix} \tag{6.62}$$

$$\mathbf{BK_d x_d^c} = \begin{bmatrix} 0 \\ -0.1K_{d4} \\ 0.01K_{d1} \end{bmatrix} \tag{6.63}$$

Therefore, $\mathbf{Ax_d^c} = \mathbf{BK_d x_d^c}$ implies that $K_{d1} = 0$ and $K_{d4} = 0$. We can also choose the remaining elements of $\mathbf{K_d}$ as zeros, and still satisfy $\mathbf{Ax_d^c} = \mathbf{BK_d x_d^c}$. Hence, $\mathbf{K_d} = \mathbf{0}$, and the control input is given by $\mathbf{u}(t) = \mathbf{K_{o1} e}(t)$.

Let us obtain the tracking error response of the system to the initial tracking error, $\mathbf{e}(0) = \mathbf{x_d^c} - \mathbf{x}(0) = \mathbf{x_d^c}$ (since $\mathbf{x}(0) = \mathbf{0}$) as follows:

```
>>t=0:0.05:6; sysCL=ss(A-B*Ko1, zeros(3,2),C,D); [y,t,e]
 = initial(sysCL, [0.1 0 0]'); <enter>
```

Then, the state-vector, $\mathbf{x}(t)$, of the closed-loop system can be calculated using $\mathbf{x}(t) = \mathbf{x_d^c} - \mathbf{e}(t)$ as follows:

```
>>n=size(t, 1); for i=1:n; Xd(i,:)=[0.1 0 0]; end; X=Xd-e; <enter>
```

while the input vector, $\mathbf{u}(t)$, is calculated as

```
>>u = Ko1*e'; <enter>
```

The calculated state variables, $\theta(t)$, $\theta^{(1)}(t)$, $i(t)$, the input voltage, $v(t)$, and the loading torque, $T_L(t)$, are plotted in Figure 6.6. Note that all the state variables reach their desired values in about 6 s, with a maximum overshoot in angular

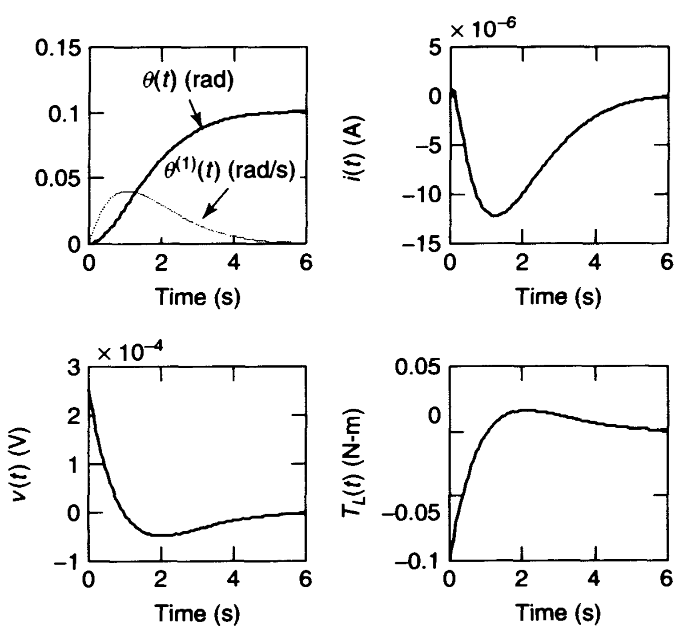


**Figure 6.6** Closed-loop initial response of the tracking system for amplifier-motor, with constant desired state, $\theta(t) = 0.1$ rad, $\theta^{(1)}(t) = 0$ rad, and $i(t) = 0$ amperes

velocity, $\theta^{(1)}(t)$, of about 0.04 rad/s, and the maximum overshoot in current, $i(t)$, of $-12.5 \times 10^{-6}$ A. The maximum input voltage, $v(t)$, is $2.5 \times 10^{-4}$ V and the maximum loading torque is $-0.1$ N-m.

Using SIMULINK, let us now investigate the effect of an uncertainty in the amplifier gain, $K_A$, on the tracking system. A SIMULINK block-diagram of the closed-loop tracking system is shown in Figure 6.7. The uncertainty in the amplifier gain, $\Delta K_A$, affects only the third state-variable, $i(t)$, and is incorporated into the

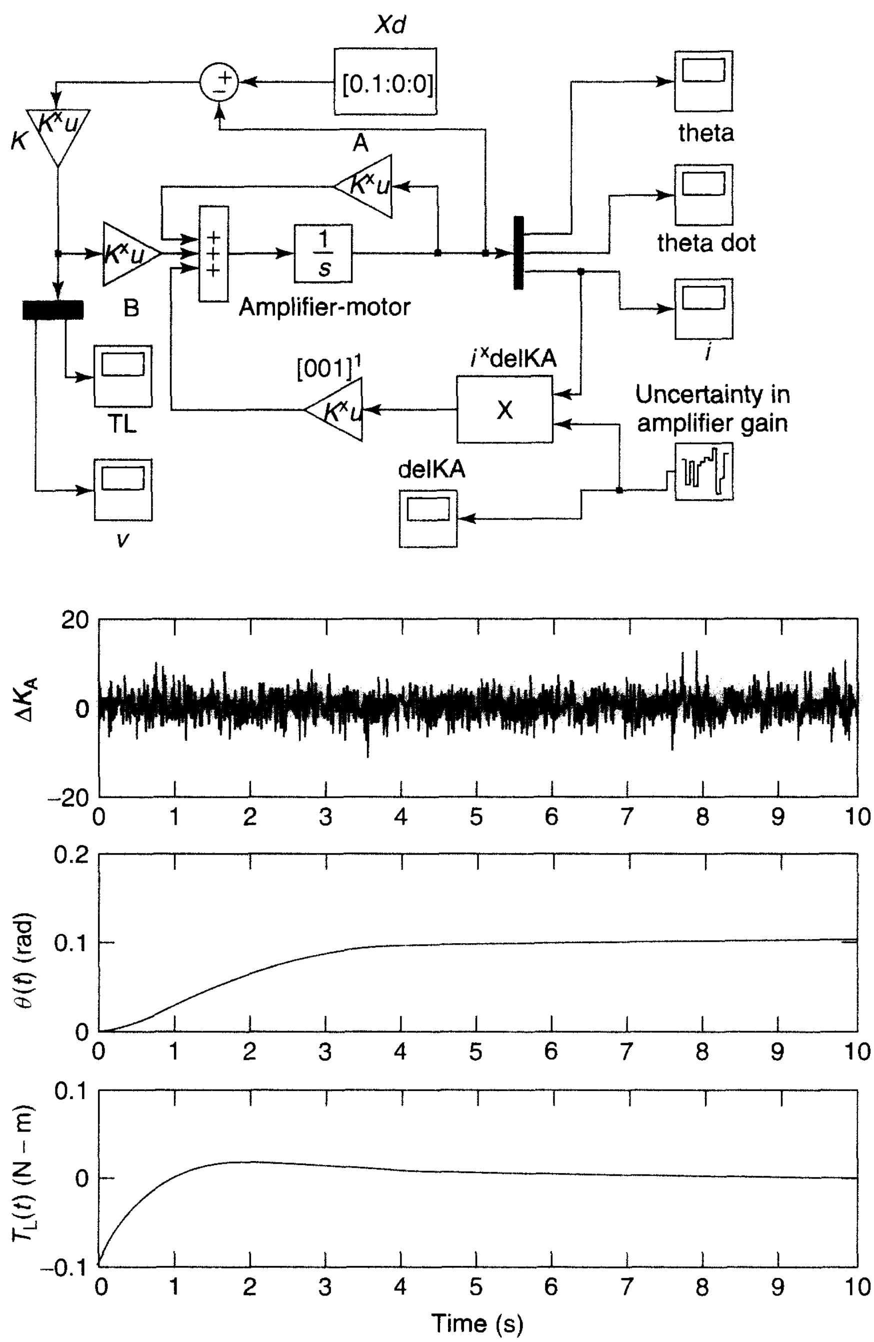


**Figure 6.7** Simulation of the tracking system for amplifier-motor, with uncertainty in the amplifier gain, $K_A$

plant dynamics model using the *band-limited white noise* block output, $\Delta K_A$, multiplied with the vector $[0; 0; i(t)]^T$, and added to the summing junction, which results in the plant dynamics being represented as $\mathbf{x}^{(1)}(t) = (\mathbf{A} + \Delta\mathbf{A})\mathbf{x}(t) + \mathbf{Bu}(t)$, where

$$\Delta\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Delta K_A \end{bmatrix} \quad (6.64)$$

The simulated values of $\Delta K_A$, $\theta(t)$, and $T_L(t)$ are shown in Figure 6.7. Note that despite a random variation in $K_A$ between $\pm 10$, the tracking system's performance is unaffected. This signifies a design which is quite robust to variations in $K_A$.

## *Example 6.4*

For a particular set of flight conditions, the lateral dynamics of an aircraft are described by a linear, time-invariant state-space representation with the following coefficient matrices:

$$\mathbf{A} = \begin{bmatrix} -9.75 & 0 & -9.75 & 0 \\ 0 & -0.8 & 8 & 0 \\ 0 & -1 & -0.8 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}; \quad \mathbf{B} = \begin{bmatrix} 20 & 2.77 \\ 0 & -3 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (6.65)$$

The state-vector consists of the *roll-rate*, $p(t)$, *yaw-rate*, $r(t)$, *side-slip angle*, $\beta(t)$, and *bank angle*, $\phi(t)$, and is written as $\mathbf{x}(t) = [p(t); r(t); \beta(t); \phi(t)]^T$. The input vector consists of the *aileron deflection angle*, $\delta_A(t)$, and *rudder deflection angle*, $\delta_R(t)$, i.e. $\mathbf{u}(t) = [\delta_A(t); \delta_R(t)]^T$. It is desired to execute a *steady turn* with a constant yaw-rate, $r_d(t) = 0.05$ rad/s, a constant bank angle, $\phi_d(t) = 0.02$ rad, and zero roll-rate and sideslip angle, $p_d(t) = \phi_d(t) = 0$. The desired state-vector is thus $\mathbf{x}_d^c = [0; 0.05; 0; 0.02]^T$. The desired state must be reached in about two seconds, with a maximum roll-rate, $p(t)$, less than 0.1 rad/s and the control inputs ($\delta_A(t)$ and $\delta_R(t)$) not exceeding 0.3 rad. Let us first select a feedforward gain matrix, $\mathbf{K_d}$, which satisfies $\mathbf{Ax}_d^c = \mathbf{BK_dx}_d^c$ as follows:

$$\mathbf{K_d} = \begin{bmatrix} K_{d1} & K_{d2} & K_{d3} & K_{d4} \\ K_{d5} & K_{d6} & K_{d7} & K_{d8} \end{bmatrix} \quad (6.66)$$

$$\mathbf{BK_dx}_d^c = \begin{bmatrix} 20(0.05K_{d2} + 0.02K_{d4}) + 2.77(0.05K_{d6} + 0.02K_{d8}) \\ -3(0.05K_{d6} + 0.1K_{d8}) \\ 0 \\ 0 \end{bmatrix}; \quad \mathbf{Ax}_d^c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6.67)$$

Equation (6.67) indicates that, to make $\mathbf{Ax}_d^c = \mathbf{BK_dx}_d^c$, we must have $0.05K_{d6} + 0.02K_{d8} = 0$ and $0.05K_{d2} + 0.02K_{d4} = 0$, which can be satisfied by selecting $\mathbf{K_d} = \mathbf{0}$. It now remains for us to calculate the optimal feedback gain matrix, $\mathbf{K_{o1}}$, by solving the algebraic Riccati equation. Note that the plant is stable with the following eigenvalues: