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:

$$\mathbf{u}(t) = \mathbf{K}[\mathbf{x}_\mathbf{d}(t) - \mathbf{x}(t)] - \mathbf{K}_\mathbf{d}\mathbf{x}_\mathbf{d}(t) - \mathbf{K}_\mathbf{n}\mathbf{x}_\mathbf{n}(t) \tag{5.25}$$

where $\mathbf{x}(t)$ is the state-vector of the plant, $\mathbf{x}_\mathbf{d}(t)$ is the *desired* state-vector, $\mathbf{x}_\mathbf{n}(t)$ is the *noise* state-vector and $\mathbf{K}$, $\mathbf{K}_\mathbf{d}$ and $\mathbf{K}_\mathbf{n}$ are the *controller gain matrices*. The desired state-vector, $\mathbf{x}_\mathbf{d}(t)$, and the noise state-vector, $\mathbf{x}_\mathbf{n}(t)$, 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:

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

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

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, $\mathbf{K}$, $\mathbf{K}_\mathbf{d}$, and $\mathbf{K}_\mathbf{n}$ 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 $\mathbf{x}_\mathbf{d}(t) = \mathbf{0}$. 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*

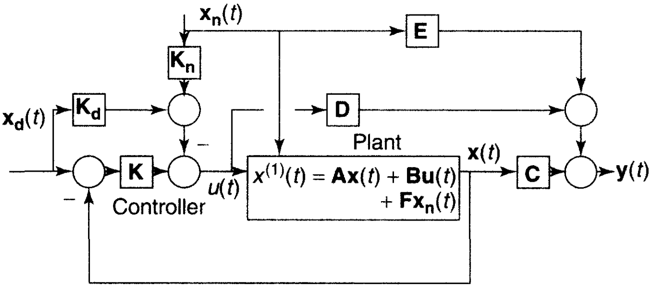

Figure 5.2 Schematic diagram of a general full-state feedback control system with desired state, $\mathbf{x_d}(t)$, and noise, $\mathbf{x_n}(t)$

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.

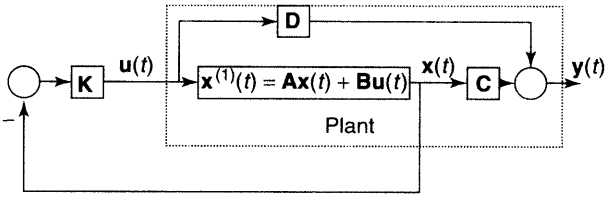

Figure 5.3 Schematic diagram of a full-state feedback regulator (i.e. control system with a zero desired state-vector) without any noise

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:

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

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

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

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

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

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:

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

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 $\mathbf{a} = [a_{n-1}; a_{\mathrm{n}-2}; \dots; a_1; a_0]$. 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

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

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:

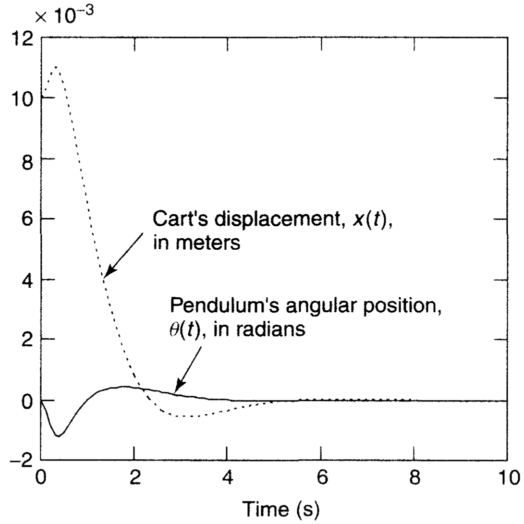

Figure 5.4 Closed-loop initial response of the regulated inverted pendulum on a moving cart to perturbation on cart displacement for the regulator gain matrix, $\mathbf{K} = [-92.9841; -10.2041; -24.2449; -12.2449]$

```
>>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

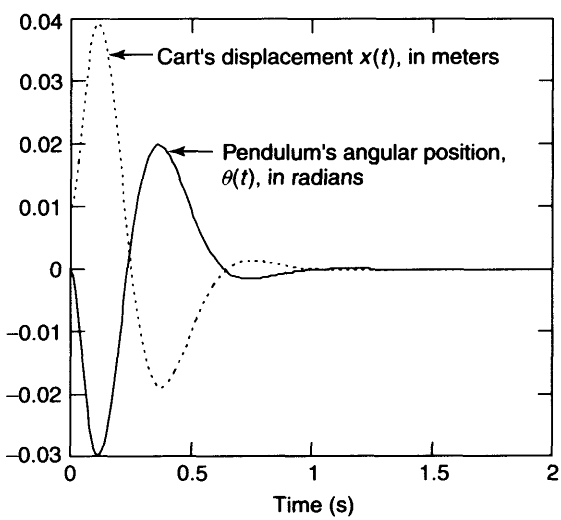

Figure 5.5 Closed-loop initial response of the regulated inverted pendulum on a moving cart to perturbation on cart displacement for the regulator gain matrix, **K** = [−2919.2; −2295.9; −570.71; −535.71]

larger in magnitude than that in Example 5.9. The input, $\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t)$, 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

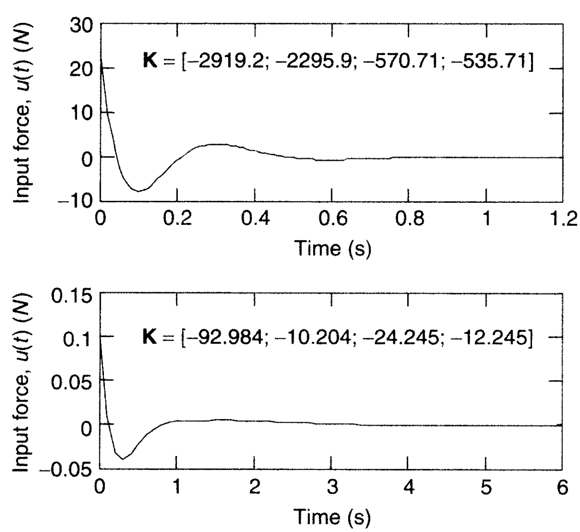

Figure 5.6 Control inputs of the regulated inverted pendulum on a moving cart for two designs of the full-state feedback regulator

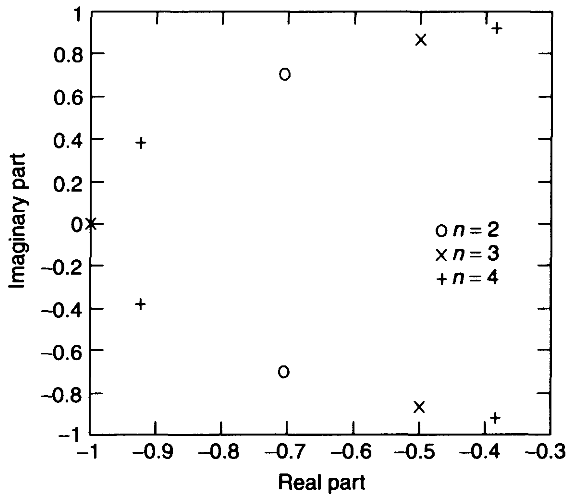

Figure 5.7 Butterworth pattern of poles in the left-half plane for $n = 2, 3$, and 4

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 α given by

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

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

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

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>
```

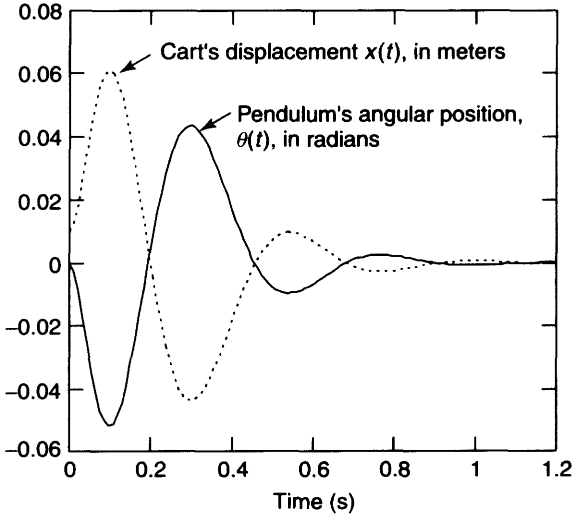

Figure 5.8 Initial response of the regulated inverted pendulum on a moving cart, for the closed-loop poles in a Butterworth pattern of radius, $R = 15$

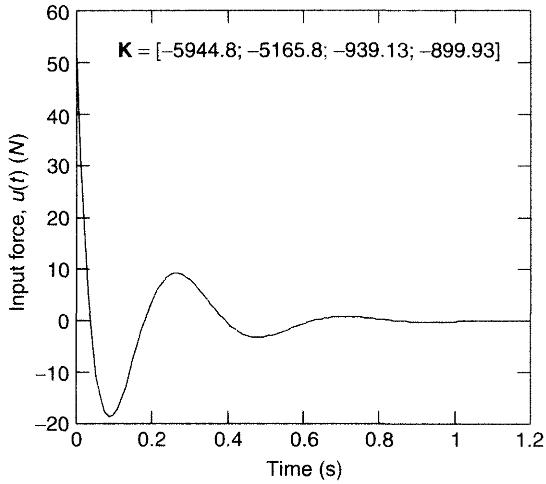

Figure 5.9 Control input for the regulated inverted pendulum on a moving cart, for closed-loop poles in a Butterworth pattern of radius, $R = 15$

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>
```

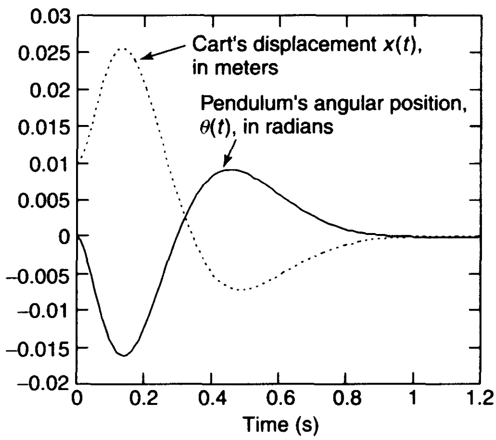

Figure 5.10 Initial response of the regulated inverted pendulum on a moving cart for the design of Example 5.11 with the closed-loop poles at $s = -7.853 \pm 3.2528i$, and $s = -7.853 \pm 7.853i$

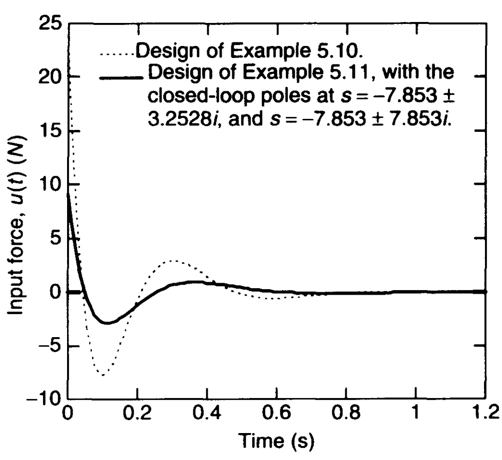

Figure 5.11 Comparison of the control input for the design of Example 5.10 with that of Example 5.11 with closed-loop poles at $s = -7.853 \pm 3.2528i$, and $s = -7.853 \pm 7.853i$

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:

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

which results in the following closed-loop characteristic equation:

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

or

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

or

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

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 $\mathbf{A}_{\mathbf{CL}} = (\mathbf{A} - \mathbf{BK})$. A schematic diagram of the full-state feedback regulator with noise is shown in Figure 5.12.

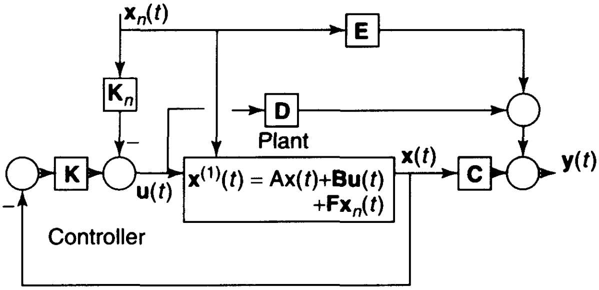

Figure 5.12 Schematic diagram of a full-state feedback regulator with noise, $\mathbf{x}_n(t)$

The regulator feedback gain matrix, $\mathbf{K}$, 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, $\mathbf{F}$, from experimental observations. Once $\mathbf{F}$ 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 $\mathbf{B}$ 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:

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

$$\mathbf{C} = \mathbf{I}; \quad \mathbf{D} = \mathbf{0}; \quad \mathbf{E} = \mathbf{0} \qquad (5.71)$$

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 $(\mathbf{F} - \mathbf{B}\mathbf{K}_n)$ 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* $(\mathbf{A}_{\mathrm{CL}}, (\mathbf{F} - \mathbf{B}\mathbf{K}_n))$). 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

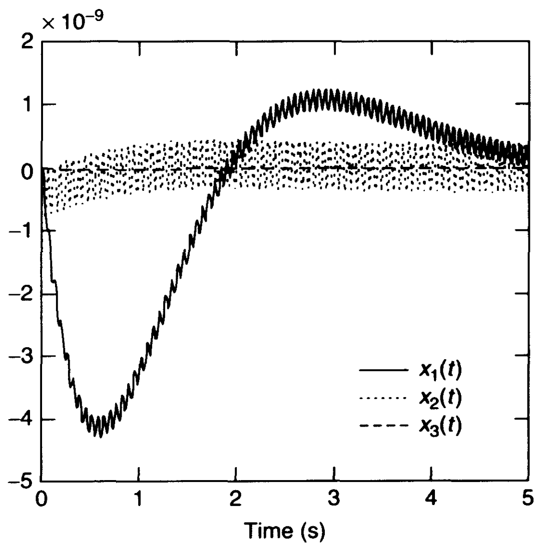

Figure 5.13 Closed-loop response of the regulated fighter aircraft to noise, when the closed-loop poles are $s = -1$, $s = -1 \pm i$

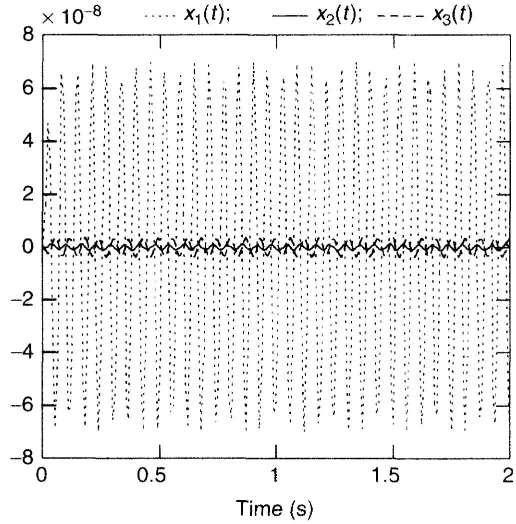

Figure 5.14 Closed-loop response of the regulated fighter aircraft to noise, when the closed-loop poles are $s = -100$, $s = -100 \pm 100i$

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 $\mathbf{e}_{\mathrm{ss}}$ to be zero, irrespective of the values of $\mathbf{x}_{\mathrm{dss}}$ and $\mathbf{x}_{n\mathrm{ss}}$, we should have $(\mathbf{A} - \mathbf{BK}_{\mathrm{d}} - \mathbf{A}_{\mathrm{d}}) = \mathbf{0}$ and $(\mathbf{F} - \mathbf{BK}_{n}) = \mathbf{0}$, by selecting the appropriate gain matrices, $\mathbf{K}_{\mathrm{d}}$ 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 $\mathbf{e}_{\mathrm{ss}}$ 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 $\mathbf{K}_{\mathrm{d}}$ and $\mathbf{K}_{n}$ as follows, to make $\mathbf{e}_{\mathrm{ss}} = \mathbf{0}$:

$$\mathbf{K}_{\mathrm{d}} = \mathbf{B}^{-1}(\mathbf{A} - \mathbf{A}_{\mathrm{d}}); \quad \mathbf{K}_{n} = \mathbf{B}^{-1}\mathbf{F} \tag{5.81}$$

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, $\mathbf{A}_{\mathrm{d}}$, is the following:

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

The eigenvalues of $\mathbf{A}_{\mathrm{d}}$ determine the poles of the target, which indicate how rapidly the desired state-vector, $\mathbf{x}_{\mathrm{d}}(t)$, 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

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

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:

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

Since by changing $\mathbf{K}_\mathrm{d}$ we can only affect the first and the third rows of $(\mathbf{A} - \mathbf{A}_\mathbf{d} - \mathbf{BK}_\mathrm{d})$, let us select $\mathbf{K}_\mathrm{d}$ such that the largest elements of $(\mathbf{A} - \mathbf{A}_\mathbf{d} - \mathbf{BK}_\mathrm{d})$, 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 $(\mathbf{A} - \mathbf{A}_\mathbf{d} - \mathbf{BK}_\mathrm{d})$ zeros, and the resulting matrix is the following:

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

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>
```

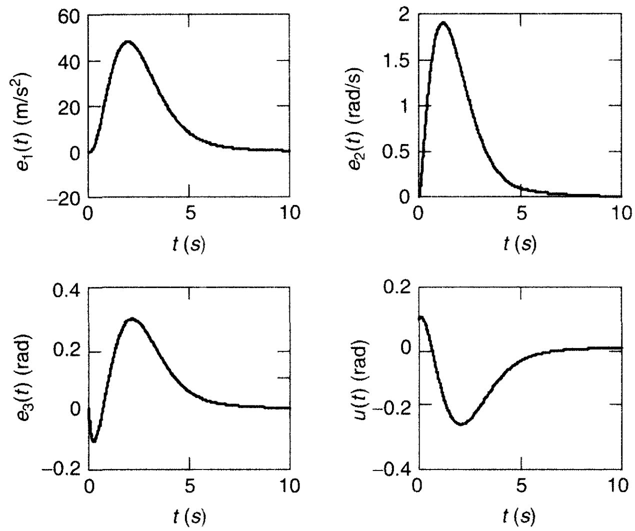

Figure 5.15 Closed-loop error and control input response of the fighter aircraft tracking a target with initial condition $\mathbf{X}_d(0) = [3;\ 0;\ 0]^T$

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

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

where $\mathbf{A}_{\mathrm{CL}} = \mathbf{A} - \mathbf{BK}$, $\mathbf{B}_{\mathrm{CL}} = [\,(\mathbf{A}_d - \mathbf{A} + \mathbf{BK}_d);\ \ (\mathbf{BK}_n - \mathbf{F})\,]$, and the input vector, $\mathbf{f}(t) = [\,\mathbf{x}_d(t)^T;\ \ \mathbf{x}_n(t)^T\,]^T$, 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, $\mathbf{e}(t) = [\,e_1(t);\ \ e_2(t);\ \ e_3(t)\,]^T$, 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.