ICC18 Lab

Asha Rani

SOLVING POLE-PLACEMENT PROBLEMS WITH MATLAB

Pole-placement problems can be solved easily with MATLAB. MATLAB has two commands—acker and place—for the computation of feedback-gain matrix **K**. The command acker is based on Ackermann's formula. This command applies to single-input systems only. The desired closed-loop poles can include multiple poles (poles located at the same place).

If the system involves multiple inputs, for a specified set of closed-loop poles the state-feedback gain matrix **K** is not unique and we have an additional freedom (or freedoms) to choose **K**. There are many approaches to constructively utilize this additional freedom (or freedoms) to determine **K**. One common use is to maximize the stability margin. The pole placement based on this approach is called the robust pole placement. The MATLAB command for the robust pole placement is place.

Although the command place can be used for both single-input and multiple-input systems, this command requires that the multiplicity of poles in the desired closed-loop poles be no greater than the rank of **B**. That is, if matrix **B** is an $n \times 1$ matrix, the command place requires that there be no multiple poles in the set of desired closed-loop poles.

For single-input systems, the commands acker and place yield the same **K**. (But for multiple-input systems, one must use the command place instead of acker.)

It is noted that when the single-input system is barely controllable, some computational problem may occur if the command acker is used. In such a case the use of the place command is preferred, provided that no multiple poles are involved in the desired set of closed-loop poles.

To use the command acker or place, we first enter the following matrices in the program:

A matrix, **B** matrix, **J** matrix

where **J** matrix is the matrix consisting of the desired closed-loop poles such that

$$\mathbf{J} = [\mu_1 \quad \mu_2 \quad \dots \quad \mu_n]$$

Then we enter

```
K = acker(A,B,J)
```

or

```
K = place(A,B,J)
```

It is noted that the command eig (A-B*K) may be used to verify that K thus obtained gives the desired eigenvalues.

)–2 Consider the same system as treated in Example 10–1. The system equation is

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -5 & -6 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

By using state feedback control $u = -\mathbf{K}\mathbf{x}$, it is desired to have the closed-loop poles at $s = \mu_i$ $(i = 1, 2, 3)$, where

$$\mu_1 = -2 + j4, \qquad \mu_2 = -2 - j4, \qquad \mu_3 = -10$$

Determine the state feedback-gain matrix **K** with MATLAB.

MATLAB Program 10–1

```
A = [0  1  0;0  0  1;-1  -5  -6];
B = [0;0;1];
J = [-2+j*4  -2-j*4  -10];
K = acker(A,B,J)

K =

   199    55    8
```

MATLAB Program 10–2

```
A = [0  1  0;0  0  1;-1  -5  -6];
B = [0;0;1];
J = [-2+j*4  -2-j*4  -10];
K = place(A,B,J)
  place: ndigits = 15

K =

  199.0000    55.0000    8.0000
```

Consider the same system as discussed in Example 10–1. It is desired that this regulator system have closed-loop poles at

$$s = -2 + j4, \qquad s = -2 - j4, \qquad s = -10$$

The necessary state feedback gain matrix $\mathbf{K}$ was obtained in Example 10–1 as follows:

$$\mathbf{K} = [199 \quad 55 \quad 8]$$

Using MATLAB, obtain the response of the system to the following initial condition:

$$\mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Response to Initial Condition: To obtain the response to the given initial condition $\mathbf{x}(0)$, we substitute $u = -\mathbf{K}\mathbf{x}$ into the plant equation to get

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}, \qquad \mathbf{x}(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

To plot the response curves (x_1 versus t, x_2 versus t, and x_3 versus t), we may use the command initial. We first define the state-space equations for the system as follows:

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} + \mathbf{I}\mathbf{u}$$

$$\mathbf{y} = \mathbf{I}\mathbf{x} + \mathbf{I}\mathbf{u}$$

where we included **u** (a three-dimensional input vector). This **u** vector is considered **0** in the computation of the response to the initial condition. Then we define

```
sys = ss(A - BK, eye(3), eye(3), eye(3))
```

and use the initial command as follows:

```
x = initial(sys, [1;0;0],t)
```

where t is the time duration we want to use, such as

```
t = 0:0.01:4;
```

Then obtain x1, x2, and x3 as follows:

```
x1 = [1  0  0]*x';
x2 = [0  1  0]*x';
x3 = [0  0  1]*x';
```

and use the plot command. This program is shown in MATLAB Program 10–3. The resulting response curves are shown in Figure 10–3.

MATLAB Program 10–3

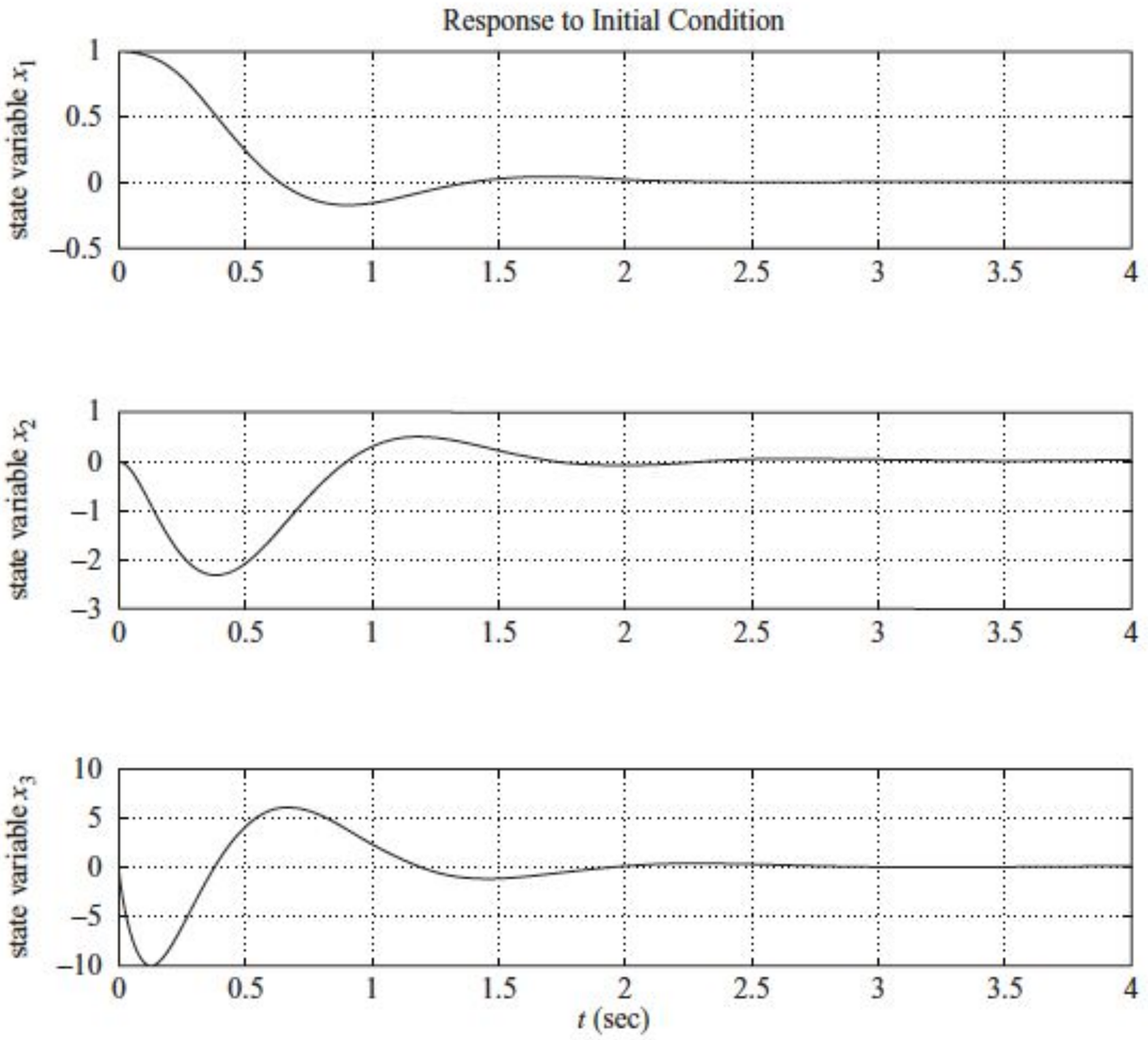
```
% Response to initial condition:

A = [0  1  0;0  0  1;-1  -5  -6];
B = [0;0;1];
K = [199  55  8];
sys = ss(A-B*K, eye(3), eye(3), eye(3));
t = 0:0.01:4;
x = initial(sys,[1;0;0],t);
x1 = [1  0  0]*x';
x2 = [0  1  0]*x';
x3 = [0  0  1]*x';

subplot(3,1,1); plot(t,x1), grid
title('Response to Initial Condition')
ylabel('state variable x1')

subplot(3,1,2); plot(t,x2),grid
ylabel('state variable x2')

subplot(3,1,3); plot(t,x3),grid
xlabel('t (sec)')
ylabel('state variable x3')
```


Response to Initial Condition
state variable x_1
1
0.5
0
−0.5
0
0.5
1
1.5
2
2.5
3
3.5
4
state variable x_2
1
0
−1
−2
−3
state variable x_3
10
5
0
−5
−10
t (sec)

Design a type 1 servo system when the plant transfer function has an integrator. Assume that the plant transfer function is given by

$$\frac{Y(s)}{U(s)} = \frac{1}{s(s+1)(s+2)}$$

The desired closed-loop poles are $s = -2 \pm j2\sqrt{3}$ and $s = -10$. Assume that the system configuration is the same as that shown in Figure 10–4 and the reference input r is a step function. Obtain the unit-step response of the designed system.

Define state variables x_1, x_2, and x_3 as follows:

$$x_1 = y$$
$$x_2 = \dot{x}_1$$
$$x_3 = \dot{x}_2$$

Then the state-space representation of the system becomes

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \tag{10–26}$$

$$y = \mathbf{C}\mathbf{x} \tag{10–27}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Referring to Figure 10–4 and noting that $n = 3$, the control signal u is given by

$$u = -(k_2x_2 + k_3x_3) + k_1(r - x_1) = -\mathbf{K}\mathbf{x} + k_1r \tag{10–28}$$

where

$$\mathbf{K} = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}$$

MATLAB Program 10–4

```
A = [0  1  0;0  0  1;0  -2  -3];
B = [0;0;1];
J = [-2+j*2*sqrt(3)  -2-j*2*sqrt(3)  -10];
K = acker(A,B,J)

K =

  160.0000    54.0000    11.0000
```

The state feedback gain matrix **K** is thus

$$\mathrm{K} = [160 \quad 54 \quad 11]$$

Unit-Step Response of the Designed System: The unit-step response of the designed system can be obtained as follows:

Since

$$\mathbf{A} - \mathbf{BK} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -3 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} [160 \quad 54 \quad 11] = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -160 & -56 & -14 \end{bmatrix}$$

from Equation (10–22) the state equation for the designed system is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -160 & -56 & -14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 160 \end{bmatrix} r \tag{10–29}$$

and the output equation is

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{10–30}$$

MATLAB Program 10–5

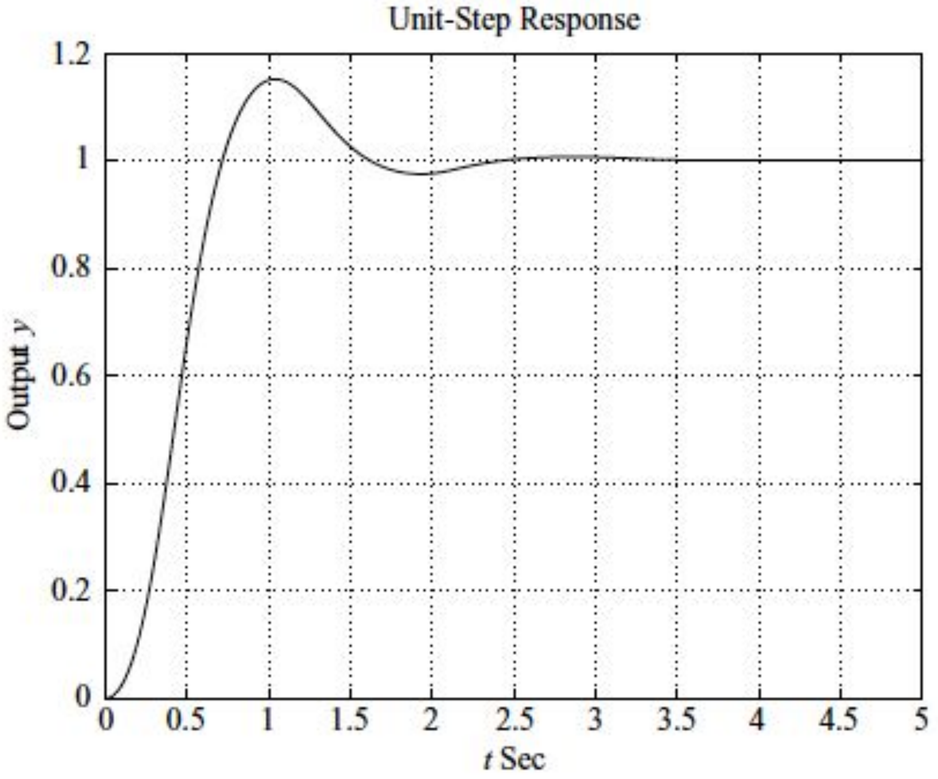
```
% ---------- Unit-step response ----------

% ***** Enter the state matrix, control matrix, output matrix,
% and direct transmission matrix of the designed system *****

AA = [0  1  0;0  0  1;-160  -56  -14];
BB = [0;0;160];
CC = [1  0  0];
DD = [0];

% ***** Enter step command and plot command *****

t = 0:0.01:5;
y = step(AA,BB,CC,DD,1,t);
plot(t,y)
grid
title('Unit-Step Response')
xlabel('t Sec')
ylabel('Output y')
```



Note that since

$$u(\infty) = -\mathbf{K}\mathbf{x}(\infty) + k_1 r(\infty) = -\mathbf{K}\mathbf{x}(\infty) + k_1 r$$

we have

$$u(\infty) = -[160 \quad 54 \quad 11]\begin{bmatrix} x_1(\infty) \\ x_2(\infty) \\ x_3(\infty) \end{bmatrix} + 160r$$

$$= -[160 \quad 54 \quad 11]\begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} + 160r = 0$$

MATLAB Program 10–7

```
%**** The following program is to obtain step response
% of the inverted-pendulum system just designed *****

A = [0  1  0  0;20.601  0  0  0;0  0  0  1;-0.4905  0  0  0];
B = [0;-1;0;0.5];
C = [0  0  1  0]
D = [0];
K = [-157.6336  -35.3733  -56.0652  -36.7466];
KI = -50.9684;
AA = [A - B*K  B*KI;-C  0];
BB = [0;0;0;0;1];
CC = [C  0];
DD = [0];

%***** To obtain response curves x1 versus t, x2 versus t,
% x3 versus t, x4 versus t, and x5 versus t, separately, enter
% the following command *****

t = 0:0.02:6;
[y,x,t] = step(AA,BB,CC,DD,1,t);

x1 = [1  0  0  0  0]*x';
x2 = [0  1  0  0  0]*x';
x3 = [0  0  1  0  0]*x';
x4 = [0  0  0  1  0]*x';
x5 = [0  0  0  0  1]*x';
```

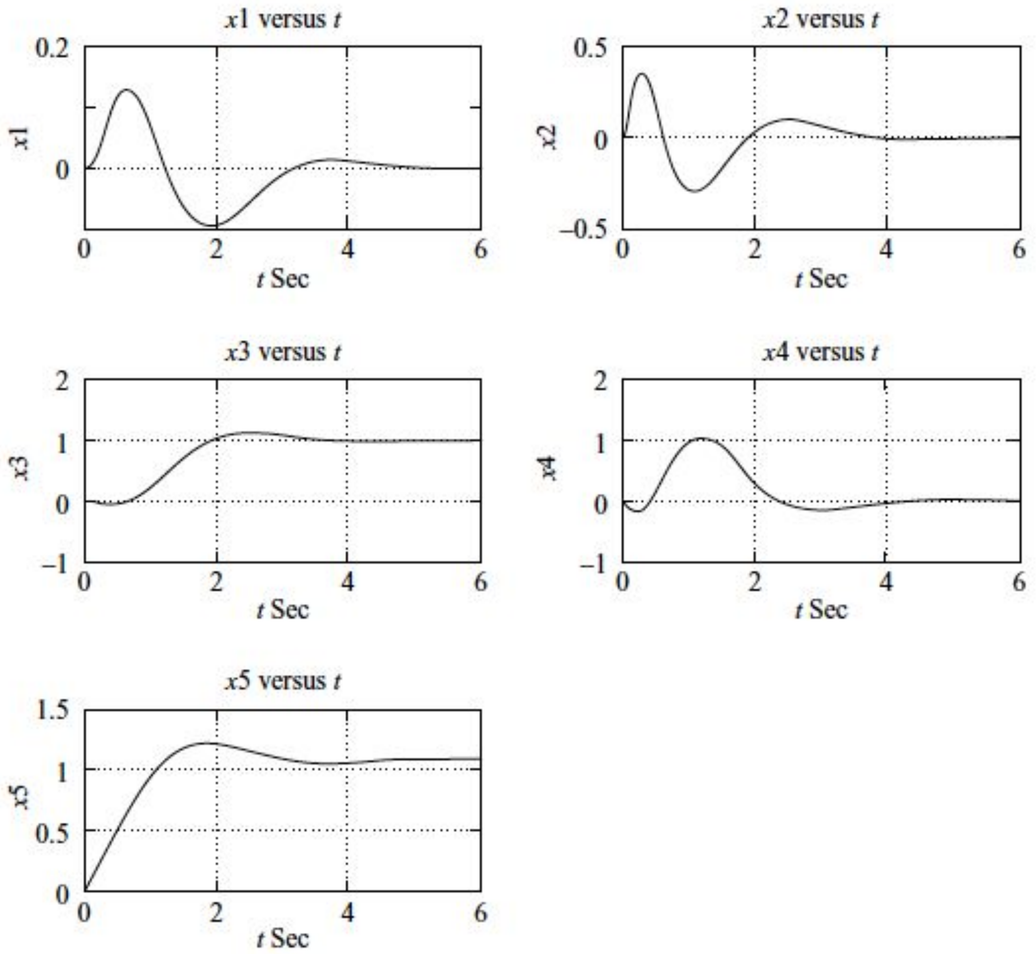
```
subplot(3,2,1); plot(t,x1); grid
title('x1 versus t')
xlabel('t Sec'); ylabel('x1')

subplot(3,2,2); plot(t,x2); grid
title('x2 versus t')
xlabel('t Sec'); ylabel('x2')

subplot(3,2,3); plot(t,x3); grid
title('x3 versus t')
xlabel('t Sec'); ylabel('x3')

subplot(3,2,4); plot(t,x4); grid
title('x4 versus t')
xlabel('t Sec'); ylabel('x4')

subplot(3,2,5); plot(t,x5); grid
title('x5 versus t')
xlabel('t Sec'); ylabel('x5')
```


x1 versus t
0.2
0
x1
0
2
4
6
t Sec
x2 versus t
0.5
0
−0.5
x2
0
2
4
6
t Sec
x3 versus t
2
1
0
−1
x3
0
2
4
6
t Sec
x4 versus t
2
1
0
−1
x4
0
2
4
6
t Sec
x5 versus t
1.5
1
0.5
0
x5
0
2
4
6
t Sec

MATLAB Program 10–8

```
% Obtaining transfer function of observer controller --- full-order observer

A = [0  1;20.6  0];
B = [0;1];
C = [1  0];
K = [29.6  3.6];
Ke = [16;84.6];
AA = A-Ke*C-B*K;
BB = Ke;
CC = K;
DD = 0;
[num,den] = ss2tf(AA,BB,CC,DD)

num =

   1.0e+003*

      0   0.7782   3.6907

den =

   1.0000  19.6000  151.2000
```

A MATLAB Program to obtain the response is shown in MATLAB Program 10–9. The resulting response curves are shown in Figure 10–15.

MATLAB Program 10–9

```
A = [0  1; 20.6  0];
B = [0;1];
C = [1  0];
K = [29.6  3.6];
Ke = [16; 84.6];
sys = ss([A-B*K  B*K; zeros(2,2)  A-Ke*C],eye(4),eye(4),eye(4));
t = 0:0.01:4;
z = initial(sys,[1;0;0.5;0],t);
x1 = [1  0  0  0]*z';
x2 = [0  1  0  0]*z';
e1 = [0  0  1  0]*z';
e2 = [0  0  0  1]*z';

subplot(2,2,1); plot(t,x1 ),grid
title('Response to Initial Condition')
ylabel('state variable x1')

subplot(2,2,2); plot(t,x2),grid
title('Response to Initial Condition')
ylabel('state variable x2')

subplot(2,2,3); plot(t,e1),grid
xlabel('t (sec)'), ylabel('error state variable e1')

subplot(2,2,4); plot(t,e2),grid
xlabel('t (sec)'), ylabel('error state variable e2')
```

Consider the system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

$$y = \mathbf{C}\mathbf{x}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad \mathbf{C} = [1 \quad 0 \quad 0]$$

Let us assume that we want to place the closed-loop poles at

$$s_1 = -2 + j2\sqrt{3}, \qquad s_2 = -2 - j2\sqrt{3}, \qquad s_3 = -6$$

Then the necessary state-feedback gain matrix $\mathbf{K}$ can be obtained as follows:

$$\mathbf{K} = [90 \quad 29 \quad 4]$$

(See MATLAB Program 10–10 for a MATLAB computation of this matrix $\mathbf{K}$.)

Next, let us assume that the output y can be measured accurately so that state variable x_1 (which is equal to y) need not be estimated. Let us design a minimum-order observer. (The minimum-order observer is of second order.) Assume that we choose the desired observer poles to be at

$$s = -10, \qquad s = -10$$

Referring to Equation (10–95), the characteristic equation for the minimum-order observer is

$$\begin{aligned} |s\mathbf{I} - \mathbf{A}_{bb} + \mathbf{K}_e\mathbf{A}_{ab}| &= (s - \mu_1)(s - \mu_2) \\ &= (s + 10)(s + 10) \\ &= s^2 + 20s + 100 = 0 \end{aligned}$$

In what follows, we shall use Ackermann's formula given by Equation (10–97).

$$\mathbf{K}_e = \phi(\mathbf{A}_{bb}) \begin{bmatrix} \mathbf{A}_{ab} \\ \hline \mathbf{A}_{ab}\mathbf{A}_{bb} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{10–99}$$

where

$$\phi(\mathbf{A}_{bb}) = \mathbf{A}_{bb}^2 + \hat{\alpha}_1 \mathbf{A}_{bb} + \hat{\alpha}_2 \mathbf{I} = \mathbf{A}_{bb}^2 + 20\mathbf{A}_{bb} + 100\mathbf{I}$$

Since

$$\tilde{\mathbf{x}} = \begin{bmatrix} x_a \\ \hline \tilde{\mathbf{x}}_b \end{bmatrix} = \begin{bmatrix} x_1 \\ \hline \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix}, \qquad \mathbf{A} = \left[\begin{array}{c|cc} 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ -6 & -11 & -6 \end{array}\right], \qquad \mathbf{B} = \begin{bmatrix} 0 \\ \hline 0 \\ 1 \end{bmatrix}$$

we have

$$A_{aa} = 0, \qquad \mathbf{A}_{ab} = [1 \quad 0], \qquad \mathbf{A}_{ba} = \begin{bmatrix} 0 \\ -6 \end{bmatrix}$$

$$\mathbf{A}_{bb} = \begin{bmatrix} 0 & 1 \\ -11 & -6 \end{bmatrix}, \qquad B_a = 0, \qquad \mathbf{B}_b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Equation (10–99) now becomes

$$\mathbf{K}_e = \left\{ \begin{bmatrix} 0 & 1 \\ -11 & -6 \end{bmatrix}^2 + 20 \begin{bmatrix} 0 & 1 \\ -11 & -6 \end{bmatrix} + 100 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right\} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 89 & 14 \\ -154 & 5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 14 \\ 5 \end{bmatrix}$$

Referring to Equations (10–88) and (10–89), the equation for the minimum-order observer can be given by

$$\dot{\tilde{\boldsymbol{\eta}}} = (\mathbf{A}_{bb} - \mathbf{K}_e\mathbf{A}_{ab})\tilde{\boldsymbol{\eta}} + [(\mathbf{A}_{bb} - \mathbf{K}_e\mathbf{A}_{ab})\mathbf{K}_e + \mathbf{A}_{ba} - \mathbf{K}_e A_{aa}]y + (\mathbf{B}_b - \mathbf{K}_e B_a)u \qquad (10\text{–}100)$$

where

$$\tilde{\boldsymbol{\eta}} = \tilde{\mathbf{x}}_b - \mathbf{K}_e y = \tilde{\mathbf{x}}_b - \mathbf{K}_e x_1$$

Noting that

$$\mathbf{A}_{bb} - \mathbf{K}_e\mathbf{A}_{ab} = \begin{bmatrix} 0 & 1 \\ -11 & -6 \end{bmatrix} - \begin{bmatrix} 14 \\ 5 \end{bmatrix}[1 \quad 0] = \begin{bmatrix} -14 & 1 \\ -16 & -6 \end{bmatrix}$$

the equation for the minimum-order observer, Equation (10–100), becomes

$$\begin{bmatrix} \dot{\tilde{\eta}}_2 \\ \dot{\tilde{\eta}}_3 \end{bmatrix} = \begin{bmatrix} -14 & 1 \\ -16 & -6 \end{bmatrix}\begin{bmatrix} \tilde{\eta}_2 \\ \tilde{\eta}_3 \end{bmatrix} + \left\{\begin{bmatrix} -14 & 1 \\ -16 & -6 \end{bmatrix}\begin{bmatrix} 14 \\ 5 \end{bmatrix} + \begin{bmatrix} 0 \\ -6 \end{bmatrix} - \begin{bmatrix} 14 \\ 5 \end{bmatrix}0\right\}y + \left\{\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 14 \\ 5 \end{bmatrix}0\right\}u$$

or

$$\begin{bmatrix} \dot{\tilde{\eta}}_2 \\ \dot{\tilde{\eta}}_3 \end{bmatrix} = \begin{bmatrix} -14 & 1 \\ -16 & -6 \end{bmatrix}\begin{bmatrix} \tilde{\eta}_2 \\ \tilde{\eta}_3 \end{bmatrix} + \begin{bmatrix} -191 \\ -260 \end{bmatrix}y + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u$$

where

$$\begin{bmatrix} \tilde{\eta}_2 \\ \tilde{\eta}_3 \end{bmatrix} = \begin{bmatrix} \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix} - \mathbf{K}_e y$$

or

$$\begin{bmatrix} \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix} = \begin{bmatrix} \tilde{\eta}_2 \\ \tilde{\eta}_3 \end{bmatrix} + \mathbf{K}_e x_1$$

If the observed-state feedback is used, then the control signal u becomes

$$u = -\mathbf{K}\tilde{\mathbf{x}} = -\mathbf{K}\begin{bmatrix} x_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix}$$

where $\mathbf{K}$ is the state feedback gain matrix. Figure 10–18 is a block diagram showing the configuration of the system with observed-state feedback, where the observer is the minimum-order observer.

MATLAB Program 10–10

```
A = [0  1  0;0  0  1;-6  -11  -6];
B = [0;0;1];
J = [-2+j*2*sqrt(3)  -2-j*2*sqrt(3)  -6];
K = acker(A,B,J)

K =

      90.0000    29.0000    4.0000

Abb = [0  1;-11  -6];
Aab = [1  0];
L = [-10  -10];
Ke = acker(Abb',Aab',L)'

Ke =

        14
         5
```