

Due date of the homework is: Sunday, November 26th @ 11:59pm.

---

1. The following LTI system is given:

$$\dot{x}(t) = \begin{bmatrix} 0 & \alpha \\ -\alpha & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \quad y(t) = [0 \quad 1] x(t).$$

- Assume that  $\alpha = 2$ . Find a state feedback controller  $u(t) = -Kx(t)$  such that the closed loop eigenvalues are both at  $-1$ .
- Assume now that  $\alpha$  is not known. Find the range of values of  $\alpha$  that would make the state-feedback controlled system stable. That is, set  $u(t) = -Kx(t)$  and find the range of values of  $\alpha$  that would produce a stable closed loop system performance.
- For the question in part (b), what is the percentage of change in  $\alpha$  from  $\alpha = 2$  that we considered in part (a), that the system can tolerate before becoming unstable?
- Assume again that  $\alpha = 2$ , and that  $x(t)$  is not available in real-time. This requires building a state estimator or an observer. Design an observer gain  $L$  such that the estimation error eigenvalues are both at  $-1$ .
- After solving (d), combine the controller from (a) with the observer in (d) to arrive at an observer-based state-feedback controller.
- Implement the whole setup on MATLAB via the `ode45` solver. Show that the observer is succeeding in estimating the system states. Consider that the estimator initial conditions are zero (i.e.,  $\hat{x}(0) = 0$ ) and that the system's initial conditions are  $x(0) = [-2 \quad 2]^\top$ .
- Assume now that we connect the observer designed in part (d) with the system with unknown  $\alpha$  in part (b). What are the values of  $\alpha$  that would drive the observer-based controller system to stay stable?

*Hint:* For any fourth order polynomial

$$s^4 + a_3s^3 + a_2s^2 + a_1s + a_0 = 0$$

to have roots with negative real parts, the necessary and sufficient conditions are that:

$$\{a_3, a_2, a_1, a_0\} > 0, \quad a_1a_2a_3 - a_0a_3^2 - a_1^2 > 0.$$

*Answer:*

- First, notice that the system is controllable, hence it is possible to design a state feedback control. The closed-loop dynamics with the controller is

$$A_{cl} = A - BK = \begin{bmatrix} 0 & 2 \\ -2 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} [k_1 \quad k_2] = \begin{bmatrix} 0 & 2 \\ -2 - k_1 & -k_2 \end{bmatrix}.$$

The characteristic polynomial of  $A_{cl}$  is

$$\text{Det}(\lambda I - A_{cl}) = \lambda^2 + k_2\lambda + 2k_1 + 4 = 0.$$

Since the desired poles are at  $-1$ , then

$$\lambda^2 + k_2\lambda + 2k_1 + 4 = \lambda^2 + 2\lambda + 1 = 0.$$

From the above, we get  $k_1 = \frac{-3}{2}$  and  $k_2 = 2$ , hence  $K = [\frac{-3}{2} \quad 2]$ .

- (b) From the structure of controllability matrix, it is true that the system is controllable for any  $\alpha \in \mathbb{R}$ . The closed-loop system with state feedback controller is

$$A_{cl} = A - BK = \begin{bmatrix} 0 & \alpha \\ -\alpha & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} [k_1 \quad k_2] = \begin{bmatrix} 0 & \alpha \\ -\alpha - k_1 & -k_2 \end{bmatrix}.$$

The characteristic polynomial of  $A_{cl}$  is

$$\text{Det}(\lambda I - A_{cl}) = \lambda^2 + k_2\lambda + \alpha k_1 + \alpha^2 = 0.$$

By using the value of  $K$  obtained from the previous calculation, the above equation becomes

$$\lambda^2 + 2\lambda - \frac{3}{2}\alpha + \alpha^2 = 0.$$

The roots of the characteristic polynomial equation can be analytically determined as

$$\lambda_{1,2} = \frac{1}{2} \left( -2 \pm \sqrt{4 + 6\alpha - 4\alpha^2} \right).$$

For a stable system, then it is necessary to have

$$\text{Re}(\lambda_{1,2}) = \text{Re} \left( \frac{1}{2} \left( -2 \pm \sqrt{4 + 6\alpha - 4\alpha^2} \right) \right) < 0.$$

From the above, the roots are real if  $\alpha = \{-0.5, 2\}$  and imaginary elsewhere, because when  $\alpha = (-\infty, -0.5)$  or  $\alpha = (2, \infty)$ , then  $4 + 6\alpha - 4\alpha^2 < 0$ , hence the system is stable with  $\text{Re}(\lambda_{1,2}) = -1$ . Next, when  $\alpha = \{-0.5, 2\}$ , then for the system to be stable, we have

$$-2 + \sqrt{4 + 6\alpha - 4\alpha^2} < 0 \Leftrightarrow 4 + 6\alpha - 4\alpha^2 < 4 \Leftrightarrow \alpha(6 - 4\alpha) < 0.$$

Which implies  $\alpha = (-\infty, 0)$  or  $\alpha = (\frac{3}{2}, \infty)$ . Finally, it can be concluded that for the system to be stable with  $K = [\frac{-3}{2} \quad 2]$  is necessary to have  $\alpha \in \mathcal{A}$  where  $\mathcal{A} = \{x \in \mathbb{R} \mid x < 0 \vee x > \frac{3}{2}\}$ .

- (c) From the above result, then the system is still stable until  $\alpha > \frac{3}{2}$ , which makes the tolerance limit to be  $-50\%$  from 2.
- (d) Notice that the system is observable, hence it is possible to design an observer. The closed-loop error dynamics is

$$A_{cl} = A - LC = \begin{bmatrix} 0 & 2 \\ -2 & 0 \end{bmatrix} - \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} [0 \quad 1] = \begin{bmatrix} 0 & 2 - l_1 \\ -2 & -l_2 \end{bmatrix}.$$

The characteristic polynomial of  $A_{cl}$  is

$$\text{Det}(\lambda I - A_{cl}) = \lambda^2 + l_2\lambda - 2l_1 + 4 = 0.$$

Since the desired poles are at  $-1$ , then

$$\lambda^2 + k_2\lambda + 2k_1 + 4 = \lambda^2 + 2\lambda + 1 = 0.$$

From the above, we get  $l_1 = \frac{3}{2}$  and  $l_2 = 2$ , hence  $L = [\frac{3}{2} \quad 2]^\top$ .

- (e) The dynamics of observer-based state feedback control is

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{e}_1(t) \\ \dot{e}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 2 & 0 & 0 \\ -0.5 & -2 & -1.5 & 2 \\ 0 & 0 & 0 & 0.5 \\ 0 & 0 & -2 & -2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ e_1(t) \\ e_2(t) \end{bmatrix}.$$

where  $e = x - \hat{x}$ .

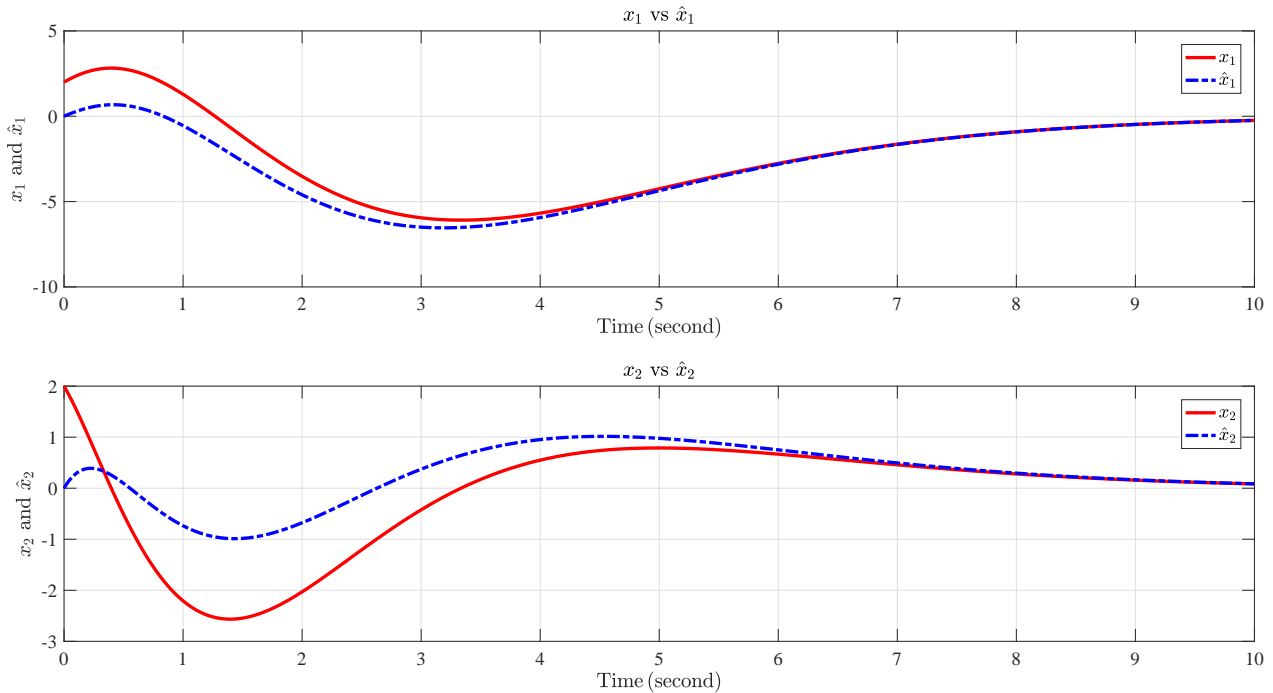


Figure 1: Trajectory of  $x$  and  $\hat{x}$ .

- (f) The simulation results for this observer-based state feedback control is given in Figure 1. It is obvious that the observer's states are converging to the system's states as  $t$  goes to infinity. The MATLAB code is given as follows

```
%Homework 7
%Problem 1
%Author: Sebastian A. Nugroho

%System description
sys.A = [0 2; -2 0];
sys.B = [0; 1];
sys.C = [0 1];

%Controller and observer matrices
sys.K = [-3/2 2];
sys.L = [3/2; 2];

%Interval
tff=10;
tspan = 0:0.01:tff;

%Initial condition
x0 = [2; 2];
xhat0 = [0; 0];

%Simulate system
[t, x] = ode45(@model, tspan, [x0; xhat0], [], sys);

%Plot figure
figure(1);
```

```

fs = 14;
set(gcf,'numbertitle','off','name','Estimation Error');
subplot(2,1,1);
plot(t,x(:,1),'r',t,x(:,3),'b-.','LineWidth',2.5);
grid on
title('$x_1$ vs $\hat{x}_1$', 'interpreter','latex','FontName','Times New Roman');
xlabel('$\text{Time}\,(\text{second})$', 'interpreter','latex','FontName','Times New Roman','FontSize',fs);
ylabel('$x_1$ and $\hat{x}_1$', 'interpreter','latex','FontName','Times New Roman','FontSize',fs);
set(gca,'FontName','Times New Roman','fontsize',fs);
h = legend('$x_1$', '$\hat{x}_1$', 'Location','northeast');
set(h,'Interpreter','latex');
set(h,'FontSize',fs);
subplot(2,1,2);
plot(t,x(:,2),'r',t,x(:,4),'b-.','LineWidth',2.5);
grid on
title('$x_2$ vs $\hat{x}_2$', 'interpreter','latex','FontName','Times New Roman');
xlabel('$\text{Time}\,(\text{second})$', 'interpreter','latex','FontName','Times New Roman','FontSize',fs);
ylabel('$x_2$ and $\hat{x}_2$', 'interpreter','latex','FontName','Times New Roman','FontSize',fs);
set(gca,'FontName','Times New Roman','fontsize',fs);
h = legend('$x_2$', '$\hat{x}_2$', 'Location','northeast');
set(h,'Interpreter','latex');
set(h,'FontSize',fs);
set(gcf,'color','w');

%System dynamics
function xtdot = model(t, xt, sys)

%Extract vector
x = xt(1:2);
xhat = xt(3:4);

% Control input
u = -sys.K*xhat;

% Plant
xdot = sys.A*x + sys.B*u;
y = sys.C*x;

%Observer
yhat = sys.C*xhat;
xhatdot = sys.A*xhat + sys.B*u + sys.L*(y-yhat);

%Combine vector
xtdot = [xdot; xhatdot];

end

```

(g) The dynamics of observer-based state feedback control with unknown  $\alpha$  is given as

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{e}_1(t) \\ \dot{e}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & \alpha & 0 & 0 \\ -\alpha - \frac{3}{2} & -2 & -\frac{3}{2} & 2 \\ 0 & 0 & 0 & \alpha - \frac{3}{2} \\ 0 & 0 & -\alpha & -2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ e_1(t) \\ e_2(t) \end{bmatrix}.$$

The characteristic polynomial of the above is

$$\lambda^4 + 4\lambda^3 + (4 + 2\alpha^2)\lambda^2 + 4\alpha^2\lambda + \alpha^4 - \frac{9}{4}\alpha^2 = 0.$$

In order to get a stable system, the following conditions must be satisfied

- $4 + 2a^2 > 0$
- $4a^2 > 0$
- $a^2(a^2 - \frac{9}{4}) > 0$
- $16a^2(4 + 2a^2) - 16a^2(a - \frac{9}{4}) - 16a^4 = \frac{25}{4}a^2 > 0.$

From the above conditions, it can be concluded that for the closed-loop system to be stable, then  $a \in \mathcal{A}$  where  $\mathcal{A} = \{x \in \mathbb{R} \mid x < -\frac{3}{2} \vee x > \frac{3}{2}\}.$

2. The following autonomous LTI system is given:

$$\dot{x}(t) = Ax(t) = \begin{bmatrix} -1 & 1 \\ 0 & 1 \end{bmatrix} x(t), \quad y(t) = Cx(t) = [0 \quad 1] x(t).$$

- (a) Is the above system observable? Prove that via the four tests of observability.
- (b) What is the unobservable subspace (if any)?
- (c) Is the above system detectable?
- (d) Design a Luenberger observer such that the closed loop estimation error dynamics have both eigenvalues placed at  $-1$ .
- (e) Show an overall diagram of the dynamic system and the observer.
- (f) Find  $\lim_{t \rightarrow \infty} e(t)$ ?
- (g) Assume now we slightly change the dynamics of the system. Specifically, we change the  $A$ -matrix to:

$$\bar{A} = \begin{bmatrix} -0.9 & 1 \\ 0 & 1 \end{bmatrix}.$$

Given this slight change in matrix  $A$ , will the observer designed in the previous part yield converging and asymptotically stable estimation error dynamics or no?

*Hint:* Derive the closed loop dynamics of  $\begin{bmatrix} \dot{x}(t) \\ \dot{e}(t) \end{bmatrix}$  and investigate the stability of the closed loop estimation error dynamics.

*Answer:*

- (a) First, by the observability matrix, we have

$$\mathcal{O} = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix},$$

which is rank deficient, hence it is unobservable.

Second, by the PBH test, the eigenvalues of  $A$  are  $-1$  and  $1$ . To proceed

- $\text{Rank} \left( \begin{bmatrix} -1I - A \\ C \end{bmatrix} \right) = \text{Rank} \left( \begin{bmatrix} 0 & -1 \\ 0 & -2 \\ 0 & 1 \end{bmatrix} \right) = 1$
- $\text{Rank} \left( \begin{bmatrix} 1I - A \\ C \end{bmatrix} \right) = \text{Rank} \left( \begin{bmatrix} 2 & -1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \right) = 2$

since not all are full rank, then according to the PBH test, the system is unobservable.

Third, by the right eigenvector test, we have

- $Cv_1 = [0 \quad 1] \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 0$
- $Cv_2 = [0 \quad 1] \begin{bmatrix} 0.4472 \\ 0.8944 \end{bmatrix} = 0.8944$

since one of them is zero, then according to the eigenvector test, the system is unobservable. Fourth, by the observability Gramian, we have

$$e^{A\tau} = Te^{D\tau}T^{-1} = \begin{bmatrix} e^{-\tau} & 0.5e^\tau - 0.5e^{-\tau} \\ 0 & e^\tau \end{bmatrix}$$

The observability Gramian is

$$\begin{aligned} W &= \int_0^t e^{A^\top \tau} C^\top C e^{A\tau} d\tau \\ &= \int_0^t \begin{bmatrix} e^{-\tau} & 0 \\ 0.5e^\tau - 0.5e^{-\tau} & e^\tau \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e^{-\tau} & 0.5e^\tau - 0.5e^{-\tau} \\ 0 & e^\tau \end{bmatrix} d\tau \\ &= \int_0^t \begin{bmatrix} 0 & 0 \\ 0 & e^\tau \end{bmatrix} \begin{bmatrix} e^{-\tau} & 0.5e^\tau - 0.5e^{-\tau} \\ 0 & e^\tau \end{bmatrix} d\tau \\ &= \int_0^t \begin{bmatrix} 0 & 0 \\ 0 & e^{2\tau} \end{bmatrix} d\tau, \end{aligned}$$

since the matrix is of rank 1, then the observability Gramian matrix is not invertible, hence the system is unobservable.

- (b) The unobservable subspace is the nullspace of the observability matrix, that is

$$\text{Null} \left( \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \right) = c \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

for  $c \in \mathbb{R}$ .

- (c) The system is indeed detectable, because for the unstable mode, the PBH test shows that

$$\text{Rank} \left( \begin{bmatrix} 1I - A \\ C \end{bmatrix} \right) = \text{Rank} \left( \begin{bmatrix} 2 & -1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \right) = 2.$$

- (d) The closed-loop error dynamics is

$$A_{cl} = A - LC = \begin{bmatrix} -1 & 1 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 1 - l_1 \\ 0 & 1 - l_2 \end{bmatrix}.$$

The characteristic polynomial of  $A_{cl}$  is

$$\text{Det}(\lambda I - A_{cl}) = \lambda^2 + l_2\lambda + l_2 - 1 = 0.$$

Since the desired poles are at  $-1$ , then

$$\lambda^2 + l_2\lambda + l_2 - 1 = \lambda^2 + 2\lambda + 1 = 0.$$

From the above, we get  $l_2 = 2$  whereas  $l_1$  can be any real number, hence we can choose to have  $L = [0 \ 2]^\top$ .

- (e) The diagram of the plant and observer is shown in Figure 2.

- (f) We have

$$\dot{e}(t) = (A - LC)e(t) = \left( \begin{bmatrix} -1 & 1 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 2 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} \right) e(t) = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix} e(t).$$

The time domain solution is

$$e(t) = e^{At}e(0) = \begin{bmatrix} e^{-t} & te^{-t} \\ 0 & e^{-t} \end{bmatrix} \begin{bmatrix} e_1(0) \\ e_2(0) \end{bmatrix} = \begin{bmatrix} e^{-t}e_1(0) + te^{-t}e_2(0) \\ e^{-t}e_2(0) \end{bmatrix},$$

hence

$$\lim_{t \rightarrow \infty} e(t) = \lim_{t \rightarrow \infty} \begin{bmatrix} e^{-t}e_1(0) + te^{-t}e_2(0) \\ e^{-t}e_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

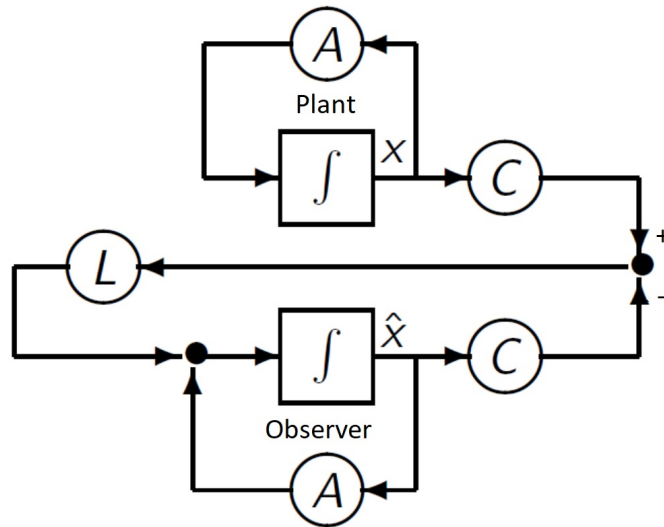


Figure 2: Plant-observer diagram.

(g) The new dynamics is

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{e}_1(t) \\ \dot{e}_2(t) \end{bmatrix} = \begin{bmatrix} -0.9 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -0.9 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ e_1(t) \\ e_2(t) \end{bmatrix}.$$

Since the eigenvalues of the error dynamics are  $-0.9$  and  $-1$ , then the observer's states are still converging to the system's states.

3. You are given a CT-LTI system defined by

$$A = \begin{bmatrix} 3 & -2 \\ 4 & -3 \end{bmatrix}, B = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, C = [1 \ 0].$$

- Design a dynamic observer such that the error dynamics have eigenvalues at  $-4$  and  $-2$ . First, you should derive the dynamics of the observer and explain how it works. Then, obtain a gain  $L$  that achieves the desired objective.
- Finally, plot the norm of the estimation error dynamics, i.e.,  $\|e(t)\|_2$  as a function of time given that the observer's initial conditions are  $\hat{x}(0) = [10 \ -10]^T$  and  $x(0) = [0 \ 0]^T$ . Consider that the input is  $u(t) = 2 \sin(10t)$ . Your plot should be a single plot versus time, since the norm function returns a scalar. Implement the whole setup on MATLAB via the `ode45` solver.
- How can you improve the convergence of the estimation error? Show that both analytically and via implementation.
- Most systems are noisy, in the sense that the real dynamics are more like:

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t)$$

where  $w(t)$  is the noise. Often, we assume that  $w(t)$  is a random variable with a well known distribution. Assume that  $w(t)$  is a white Gaussian noise. You can implement this via the `randn` or `rand` commands in MATLAB. For example, you can write in the ODE solver file `w=0.1*randn(n,1)` which defines a vector of random quantities.

Investigate whether the observer you designed in the above parts is robust enough to this noise signal in the state evolution. Of course, you should not add noise in the observer dynamics, but only in the state dynamics. You can change the scaling factor  $0.1$  to something smaller or bigger—depending on the robustness of this observer.

Can you draw any conclusions?

Answer:

- (a) The Luenberger observer is constructed as follows

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)) \\ &= A\hat{x}(t) + Bu(t) + L(Cx(t) - C\hat{x}(t)) \\ &= A\hat{x}(t) + Bu(t) + LC(x(t) - \hat{x}(t))\end{aligned}$$

the error dynamics then can be obtained as

$$\begin{aligned}\dot{e}(t) &= \dot{x}(t) - \dot{\hat{x}}(t) \\ &= Ax(t) + Bu(t) - (A\hat{x}(t) + Bu(t) + LC(x(t) - \hat{x}(t))) \\ &= A(x(t) - \hat{x}(t)) - LC(x(t) - \hat{x}(t)) \\ &= (A - LC)(x(t) - \hat{x}(t)) \\ &= (A - LC)e(t) \\ &= \left( \begin{bmatrix} 3 & -2 \\ 4 & -3 \end{bmatrix} - \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \right) e(t) \\ &= \begin{bmatrix} 3 - l_1 & -2 \\ 4 - l_2 & -3 \end{bmatrix} e(t) = A_{cl}e(t).\end{aligned}$$

Realize that the pair  $(A, C)$  is indeed observable. The characteristic polynomial of the above is

$$\lambda^2 + l_1\lambda + 3l_1 - 2l_2 - 1 = 0$$

Since the desired poles are at  $-4$  and  $-2$ , then

$$\lambda^2 + l_1\lambda + 3l_1 - 2l_2 - 1 = \lambda^2 + 6\lambda + 8 = 0.$$

From the above, we obtain  $L = [6 \quad 4.5]^\top$ .

- (b) The simulation result is given is Figure 3. The MATLAB code is given below

```
%Homework 7
%Problem 3a
%Author: Sebastian A. Nugroho

%System description
sys.A = [3 -2; 4 -3];
sys.B = [1; -1];
sys.C = [1 0];

%Controller and observer matrices
sys.L = [6; 4.5];

%Interval
tff=10;
tspan = 0:0.01:tff;

%Initial condition
x0 = [0; 0];
xhat0 = [10; -10];

%Simulate system
[t, x] = ode45(@model, tspan, [x0; xhat0], [], sys);
```



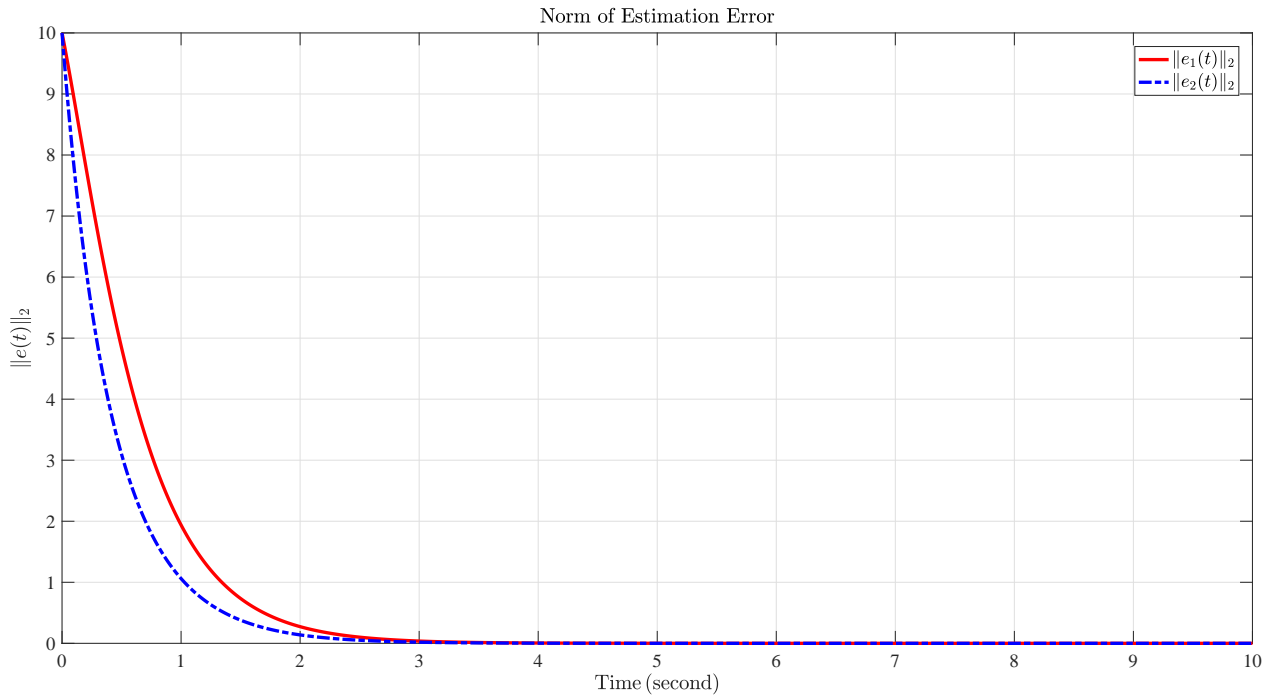


Figure 3: Norm of estimation error.

```

%Calculate the error norm
e = [];
for i = 1:length(t)
    e = [e; [norm(x(i,1)-x(i,3),2) norm(x(i,2)-x(i,4),2)]];
end

%Plot figure
figure(1);
fs = 14;
set(gcf,'numbertitle','off','name','Norm of Estimation Error');
plot(t,e(:,1),'r',t,e(:,2),'b-.','LineWidth',2.5);
grid on
title('Norm of Estimation Error','interpreter','latex','FontName','Times New Roman');
xlabel('\textrm{Time\,(second)}','interpreter','latex','FontName','Times New Roman','FontSize',fs);
ylabel('\Vert e(t)\Vert_2','interpreter','latex','FontName','Times New Roman','FontSize',fs);
set(gca,'FontName','Times New Roman','fontsize',fs);
h = legend('\Vert e_1(t)\Vert_2','\Vert e_2(t)\Vert_2','Location','northeast');
set(h,'Interpreter','latex');
set(h,'FontSize',fs);
set(gcf,'color','w');

%System dynamics
function xtdot = model(t, xt, sys)

%Extract vector
x = xt(1:2);
xhat = xt(3:4);

% Control input

```

```

u = 2*sin(10*t);

% Plant
xdot = sys.A*x + sys.B*u;
y = sys.C*x;

%Observer
yhat = sys.C*xhat;
xhatdot = sys.A*xhat + sys.B*u + sys.L*(y-yhat);

%Combine vector
xtdot = [xdot; xhatdot];

end

```

- (c) The convergence of the estimation can be increased by placing the eigenvalues of the error dynamics farther from the imaginary axis. For instance, if the desired poles are moved to  $-4$  and  $-6$ , then

$$\lambda^2 + l_1\lambda + 3l_1 - 2l_2 - 1 = \lambda^2 + 10\lambda + 24 = 0.$$

From the above, we obtain  $L = [10 \ 2.5]^\top$ . The simulation results are depicted in Figure 4. From the simulation, it is clear that the estimation states converge faster to the real states, hence making the error norm to be smaller. The MATLAB code are given below

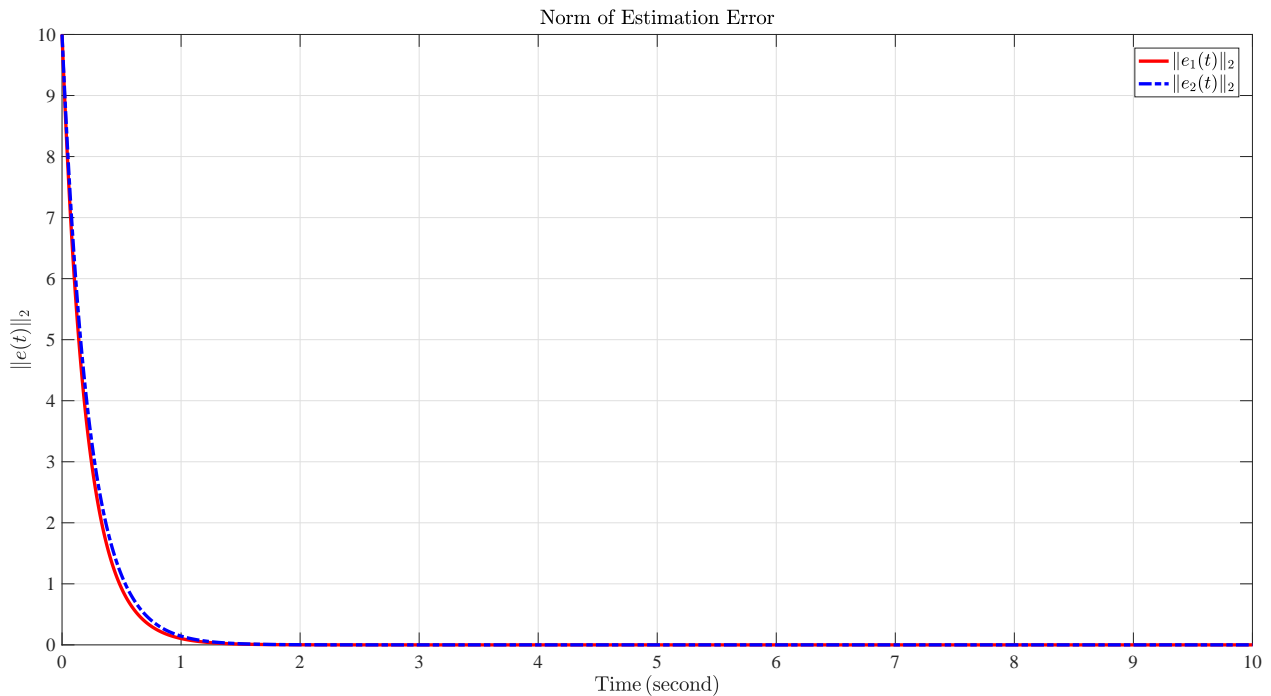


Figure 4: Norm of estimation error.

```

%Homework 7
%Problem 3b
%Author: Sebastian A. Nugroho

%System description
sys.A = [3 -2; 4 -3];

```

```

sys.B = [1; -1];
sys.C = [1 0];

%Controller and observer matrices
sys.L = [10; 2.5];

%Interval
tff=10;
tspan = 0:0.01:tff;

%Initial condition
x0 = [0; 0];
xhat0 = [10; -10];

%Simulate system
[t, x] = ode45(@model, tspan, [x0; xhat0], [], sys);

%Calculate the error norm
e = [];
for i = 1:length(t)
    e = [e; [norm(x(i,1)-x(i,3),2) norm(x(i,2)-x(i,4),2)]];
end

%Plot figure
figure(1);
fs = 14;
set(gcf,'numbertitle','off','name','Norm of Estimation Error');
plot(t,e(:,1),'r',t,e(:,2),'b-.','LineWidth',2.5);
grid on
title('Norm of Estimation Error','interpreter','latex','FontName','Times New Roman');
xlabel('$\text{Time}\,(second)$','interpreter','latex','FontName','Times New Roman','FontSize',fs);
ylabel('$\text{Vert } e(t)\text{Vert}_2$','interpreter','latex','FontName','Times New Roman','FontSize',fs);
set(gca,'FontName','Times New Roman','fontsize',fs);
h = legend('$\text{Vert } e_1(t)\text{Vert}_2$','$\text{Vert } e_2(t)\text{Vert}_2$','Location','northeast');
set(h,'Interpreter','latex');
set(h,'FontSize',fs);
set(gcf,'color','w');

%System dynamics
function xtdot = model(t, xt, sys)

%Extract vector
x = xt(1:2);
xhat = xt(3:4);

% Control input
u = 2*sin(10*t);

% Plant
xdot = sys.A*x + sys.B*u;
y = sys.C*x;

%Observer
yhat = sys.C*xhat;
xhatdot = sys.A*xhat + sys.B*u + sys.L*(y-yhat);

```

```

%Combine vector
xtdot = [xdot; xhatdot];

end

```

- (d) With disturbance, the simulation results are given in Figure 5. The disturbance fails the estimation with the most significant influence happens after  $t = 7$  seconds. This shows that the Luenberger observer only suitable for linear system without the presence of any disturbance. More advanced observer design is needed to overcome this disturbance. The MATLAB code is given below

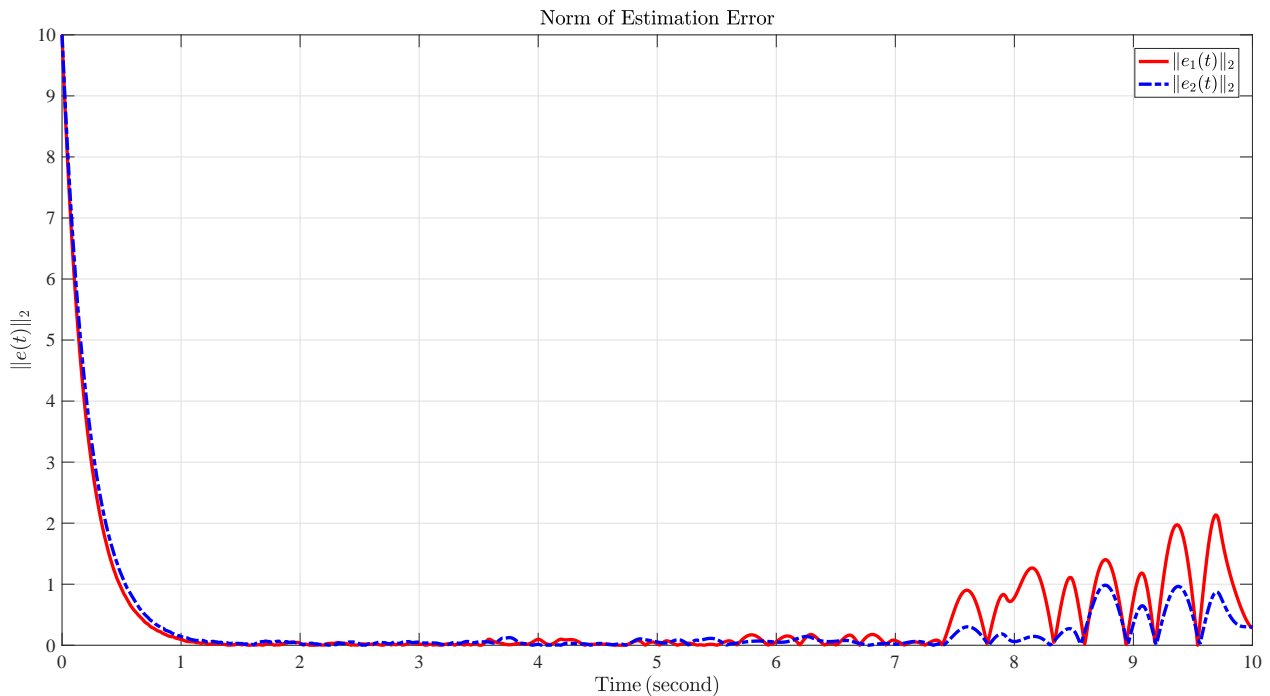


Figure 5: Norm of estimation error.

```

%Homework 7
%Problem 3c
%Author: Sebastian A. Nugroho

%System description
sys.A = [3 -2; 4 -3];
sys.B = [1; -1];
sys.C = [1 0];

%Controller and observer matrices
sys.L = [10; 2.5];

%Interval
tff=10;
tspan = 0:0.01:tff;

%Initial condition
x0 = [0; 0];
xhat0 = [10; -10];

```

```

%Simulate system
[t, x] = ode45(@model, tspan, [x0; xhat0], [], sys);

%Calculate the error norm
e = [];
for i = 1:length(t)
    e = [e; [norm(x(i,1)-x(i,3),2) norm(x(i,2)-x(i,4),2)]];
end

%Plot figure
figure(1);
fs = 14;
set(gcf,'numbertitle','off','name','Norm of Estimation Error');
plot(t,e(:,1),'r',t,e(:,2),'b-','LineWidth',2.5);
grid on
title('Norm of Estimation Error','interpreter','latex','FontName','Times New Roman');
xlabel('\textrm{Time\,(second)}','interpreter','latex','FontName','Times New Roman','FontSize',fs);
ylabel('\Vert e(t)\Vert_2','interpreter','latex','FontName','Times New Roman','FontSize',fs);
set(gca,'FontName','Times New Roman','fontsize',fs);
h = legend('\Vert e_1(t)\Vert_2','\Vert e_2(t)\Vert_2','Location','northeast');
set(h,'Interpreter','latex');
set(h,'FontSize',fs);
set(gcf,'color','w');

% %Plot figure
% figure(2);
% fs = 14;
% set(gcf,'numbertitle','off','name','Estimation Error');
% subplot(2,1,1);
% plot(t,x(:,1),'r',t,x(:,3),'b-','LineWidth',2.5);
% grid on
% title('$x_1$ vs $\hat{x}_1$','interpreter','latex','FontName','Times New Roman');
% xlabel('\textrm{Time\,(second)}','interpreter','latex','FontName','Times New Roman','FontSize',fs);
% ylabel('$x_1$ and $\hat{x}_1$','interpreter','latex','FontName','Times New Roman','FontSize',fs);
% set(gca,'FontName','Times New Roman','fontsize',fs);
% h = legend('$x_1$','$\hat{x}_1$','Location','northeast');
% set(h,'Interpreter','latex');
% set(h,'FontSize',fs);
% subplot(2,1,2);
% plot(t,x(:,2),'r',t,x(:,4),'b-','LineWidth',2.5);
% grid on
% title('$x_2$ vs $\hat{x}_2$','interpreter','latex','FontName','Times New Roman');
% xlabel('\textrm{Time\,(second)}','interpreter','latex','FontName','Times New Roman','FontSize',fs);
% ylabel('$x_2$ and $\hat{x}_2$','interpreter','latex','FontName','Times New Roman','FontSize',fs);
% set(gca,'FontName','Times New Roman','fontsize',fs);
% h = legend('$x_2$','$\hat{x}_2$','Location','northeast');
% set(h,'Interpreter','latex');
% set(h,'FontSize',fs);
% set(gcf,'color','w');

%System dynamics
function xtdot = model(t, xt, sys)

%Extract vector
x = xt(1:2);

```

```
xhat = xt(3:4);

% Control input
u = 2*sin(10*t);

%Disturbance
w = 0.5*randn(2,1);

% Plant
xdot = sys.A*x + sys.B*u + w;
y = sys.C*x;

%Observer
yhat = sys.C*xhat;
xhatdot = sys.A*xhat + sys.B*u + sys.L*(y-yhat);

%Combine vector
xtdot = [xdot; xhatdot];

end
```