1) $u = -Kx$

a)

$$\dot{x} = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix} x + \begin{bmatrix} 2 \\ 0 \end{bmatrix} u$$

$$K = \begin{bmatrix} K_1 & K_2 \end{bmatrix}$$

$$\dot{x} = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix} x - \begin{bmatrix} 2 \\ 0 \end{bmatrix} Kx$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$K = \begin{bmatrix} K_P & K_D \end{bmatrix}$$

$$\dot{x} = (A - BK)x + BK_g r$$

$$y = (C - DK)x + DK_g r$$

desired closed loop poles for a second order system with a given natural frequency ω_n and damping ratio ξ.

$$s^2 + 2\xi\omega_n s + \omega_n^2 = 0$$

$$\omega_n = 1, \quad \xi = 0.707$$

$$s^2 + 1.414s + 1 = 0.$$

$$\det(sI - A_{cl}) = \det \begin{bmatrix} s & 0 \\ -(2-2K_1) & s+2K_2 \end{bmatrix} = 0$$

$$= s(s + 2K_2) \Rightarrow s^2 + 2K_2 s$$

$$s^2 + 1.414s + 1 : \quad 2K_1 = 1.414 \Rightarrow K_1 = 0.707$$

$$2 - 2K_2 = 0 \Rightarrow K_2 = 1$$

$$K = \begin{bmatrix} 0.707 & 1 \end{bmatrix}$$

b) $\omega_n = 10$ rad/s

$\zeta = 0.5$

Observer design: $\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$

$\dot{e} = \dot{x} - \dot{\hat{x}} = (A - LC)e$

e = estimated error.

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0.$$

$$\Rightarrow \quad s^2 + 10s + 100 = 0. \quad \text{——} \; (1)$$

$$L = [L_1 \; L_2]^T$$

$$A - LC = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix} - \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -l_1 \\ 2 & -l_2 \end{bmatrix}$$

$$\det(sI - (A - LC)) = \det \begin{bmatrix} s & l_1 \\ -2 & s + l_2 \end{bmatrix} = s(s + l_2) + 2l_1$$

$$= s^2 + l_2 s + 2l_1 = 0.$$

matching with eq (1) we get,

$$l_2 = 10$$

$$l_1 = 0.125$$

$$L = [0.125 \quad 10]^T$$

c)

$$K = \begin{bmatrix} 1 & 0.707 \end{bmatrix}$$

$$L = \begin{bmatrix} 50 \\ 10 \end{bmatrix}$$

$$\dot{x} = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix} x + \begin{bmatrix} 2 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} x$$

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$

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

$$u = -K\hat{x}$$

$$\dot{\hat{x}} = A\hat{x} + B(-K\hat{x}) + L(y - C\hat{x})$$

$$\dot{\hat{x}} = (A - BK)\hat{x} + L(Cx - C\hat{x})$$

$$\dot{\hat{x}} = (A - BK)\hat{x} + LC(x - \hat{x})$$

$$\dot{\hat{x}} = (A - BK + LC)\hat{x} - LCx$$

$$\dot{x} = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix} x + \begin{bmatrix} 2 \\ 0 \end{bmatrix} (-K\hat{x})$$ $\rightarrow$ True state Dynamics

$$\dot{x} = \begin{bmatrix} 0 & 0 \\ 2 & 0 \end{bmatrix} x - \begin{bmatrix} 2 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0.707 \end{bmatrix} \hat{x}$$

Observer state Dynamics:

$$\dot{\hat{x}} = \left(\begin{bmatrix}0 & 0\\ 2 & 0\end{bmatrix} - \begin{bmatrix}2\\ 0\end{bmatrix}\begin{bmatrix}1 & 0.707\end{bmatrix} + \begin{bmatrix}50\\ 10\end{bmatrix}\begin{bmatrix}0 & 1\end{bmatrix} \right)\hat{x} + \begin{bmatrix}50\\ 10\end{bmatrix}\begin{bmatrix}0 & 1\end{bmatrix} x$$

$$\dot{\hat{x}} = \begin{bmatrix}0 & -50\\ 2 & -2\end{bmatrix}\hat{x} + \begin{bmatrix}50\\ 10\end{bmatrix} y$$

d)

hw8.m × +

/MATLAB Drive/hw8.m

```
% Define system matrices
A = [0 0; 2 0];
B = [2; 0];
C = [0 1];
D = 0;

% Controller and Observer gains
K = [1 0.707];
L = [50; 10];

% Initial conditions
x0 = [-5; -4];
x_hat0 = [0; 0];

% Time span for the simulation
t = linspace(0, 10, 1000);

% Define the differential equations for the system
sys_ode = @(t, x) [A-B*K, B*K; L*C-A+L*C+B*K, -L*C]*x + [0; 0; 0; 0] + [0; 0; -L(1); -L(2)]*sin(t);

% Solve the ODEs
[T, X] = ode45(sys_ode, t, [x0; x_hat0]);

% Extract states and their estimates
x1 = X(:, 1);
x2 = X(:, 2);
x1_hat = X(:, 3);
x2_hat = X(:, 4);

% Plot x1 and its estimate
figure;
plot(T, x1, 'b', T, x1_hat, 'r--');
title('State $x_1$ and its estimate $\hat{x}_1$ over time', 'Interpreter', 'latex');
xlabel('Time (sec)', 'Interpreter', 'latex');
ylabel('State value', 'Interpreter', 'latex');
legend({'$x_1$', '$\hat{x}_1$'}, 'Interpreter', 'latex');

% Plot x2 and its estimate
figure;
plot(T, x2, 'b', T, x2_hat, 'r--');
title('State $x_2$ and its estimate $\hat{x}_2$ over time', 'Interpreter', 'latex');
xlabel('Time (sec)', 'Interpreter', 'latex');
ylabel('State value', 'Interpreter', 'latex');
legend({'$x_2$', '$\hat{x}_2$'}, 'Interpreter', 'latex');
```

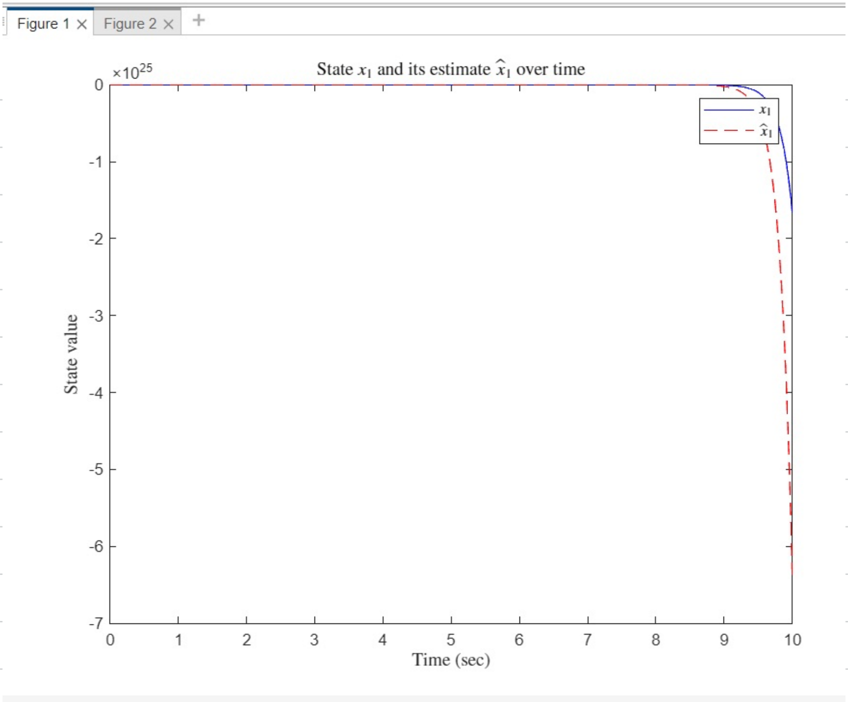


Figure 1
Figure 2
State x_1 and its estimate $\hat{x}_1$ over time
×10^25
State value
Time (sec)
x_1
$\hat{x}_1$

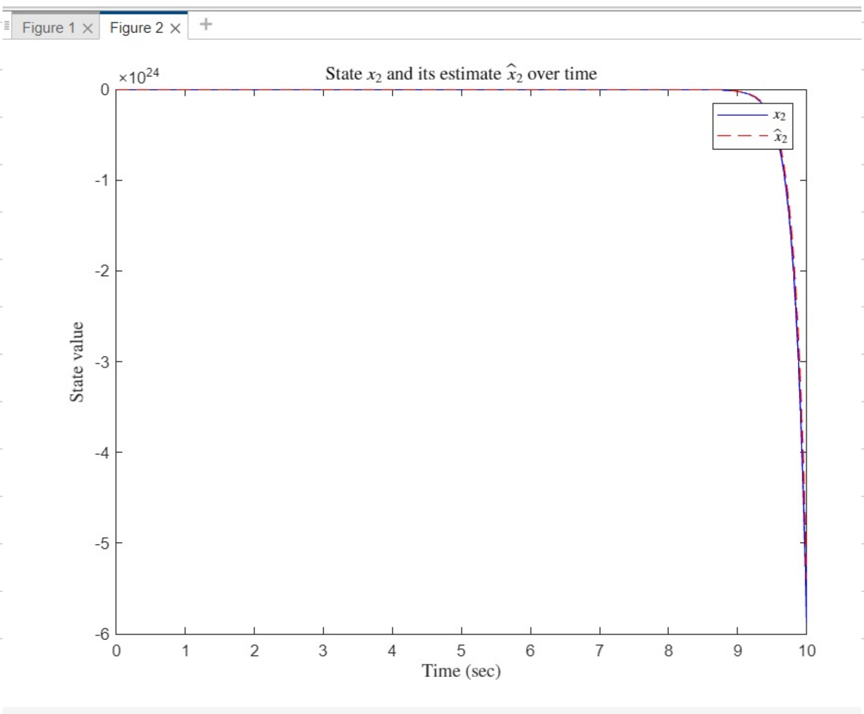


Figure 1
Figure 2
State x_2 and its estimate $\hat{x}_2$ over time
×10^24
State value
Time (sec)
x_2
$\hat{x}_2$

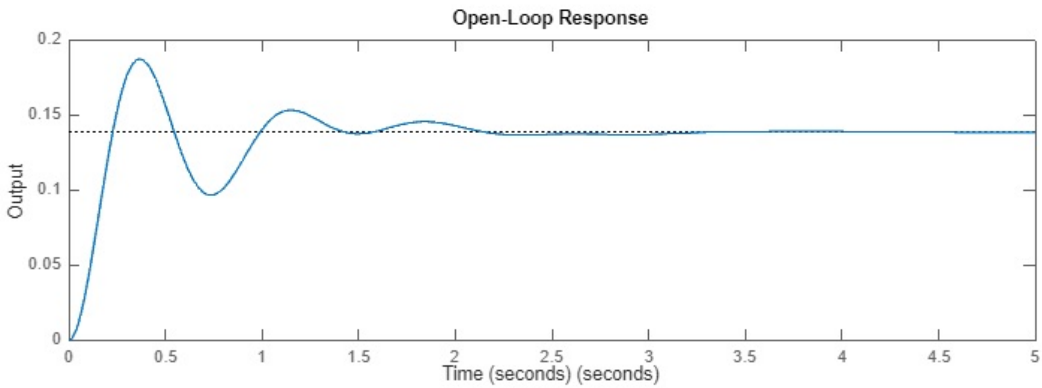
2)

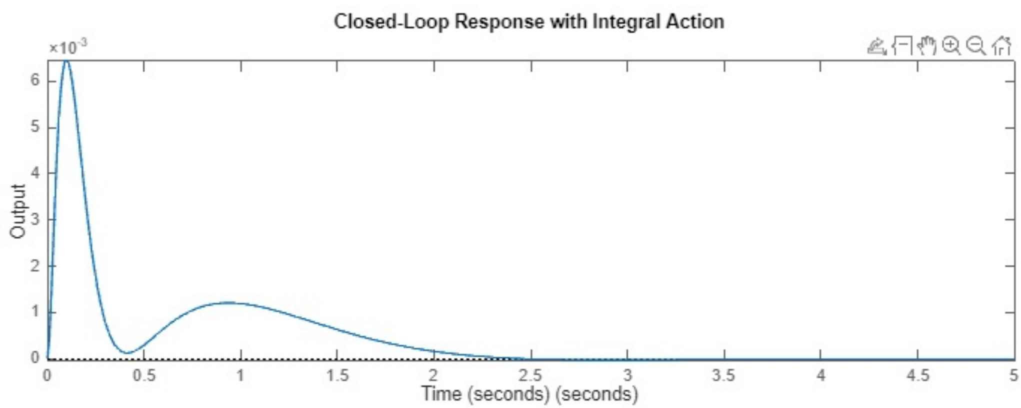
a)

hw8.m × hw8_1.m × hw8_2.m ×

/MATLAB Drive/hw8_1.m

```
% Define the original system matrices and initial state
A = [0 1 0 0;
     0 0 1 0;
     0 0 0 1;
     -650 -180 -90 -6];
B = [0; 0; 0; 1];
C = [90 15 10 0];
D = 0;
x0 = [2; 1; 3; 0];

% Extend the system to include integral action
Ae = [A, zeros(4, 1); -C, 0];
Be = [B; 0];
Ce = [C, 0];
De = 0;

% Initial conditions for extended system
xe0 = [x0; 0];

% Convert overshoot and settling time to damping ratio and natural frequency
zeta = -log(0.02)/sqrt(pi^2 + log(0.02)^2);
wn = 4/(zeta*2);  % Based on the settling time formula Ts ~ 4/(zeta*wn)

% Desired closed-loop poles
p1 = -zeta*wn + wn*sqrt(1-zeta^2)*1i;
p2 = -zeta*wn - wn*sqrt(1-zeta^2)*1i;
% Adding more poles far left on the real axis to ensure fast response
p3 = -10; p4 = -20; p5 = -30;

% Pole placement for controller design
K = place(Ae, Be, [p1, p2, p3, p4, p5]);

% System object for open-loop and closed-loop
sys_open = ss(A, B, C, D);
sys_closed = ss(Ae-Be*K, Be, Ce, De);

% Time span
t = 0:0.01:5;

% Step response for open-loop and closed-loop
figure;
subplot(2,1,1);
step(sys_open, t);
```


Open-Loop Response
Output
Time (seconds) (seconds)
0
0.05
0.1
0.15
0.2
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Closed-Loop Response with Integral Action
×10^-3
Output
Time (seconds) (seconds)
0
1
2
3
4
5
6
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

2 b)

hw8.m × hw8_1.m × hw8_2.m ×

/MATLAB Drive/hw8_1.m

```
% Define the original system matrices and initial state
A = [0 1 0 0;
     0 0 1 0;
     0 0 0 1;
     -650 -180 -90 -6];
B = [0; 0; 0; 1];
C = [90 15 10 0];
D = 0;
x0 = [2; 1; 3; 0];

% Extend the system to include integral action
Ae = [A, zeros(4, 1); -C, 0];
Be = [B; 0];
Ce = [C, 0];
De = 0;

% Initial conditions for extended system
xe0 = [x0; 0];

% Convert overshoot and settling time to damping ratio and natural frequency
zeta = -log(0.02)/sqrt(pi^2 + log(0.02)^2);
wn = 4/(zeta*2);  % Based on the settling time formula Ts ~ 4/(zeta*wn)

% Desired closed-loop poles
p1 = -zeta*wn + wn*sqrt(1-zeta^2)*1i;
p2 = -zeta*wn - wn*sqrt(1-zeta^2)*1i;
% Adding more poles far left on the real axis to ensure fast response
p3 = -10; p4 = -20; p5 = -30;

% Pole placement for controller design
K = place(Ae, Be, [p1, p2, p3, p4, p5]);

% System object for open-loop and closed-loop
sys_open = ss(A, B, C, D);
sys_closed = ss(Ae-Be*K, Be, Ce, De);

% Time span
t = 0:0.01:5;

% Step response for open-loop and closed-loop
figure;
subplot(2,1,1);
step(sys_open, t);
```