Roger Wienaah

ME 531 Problem Set 4

Roger Wienaah

ME 531 : Problem Set 4

Question 1:

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

$$A = \begin{bmatrix} 1 & 4 \\ -5 & 10 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & -4 \end{bmatrix}$$

- The system is observable if the observability matrix Θ is full rank

rank $(\Theta) = n = 2$

$$\Theta = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & -4 \\ 1+(-4)(-5) & (1)(4)+(-4)(10) \end{bmatrix} = \begin{bmatrix} 1 & -4 \\ 21 & -36 \end{bmatrix}$$

$$\det(\Theta) = \begin{vmatrix} 1 & -4 \\ 21 & -36 \end{vmatrix} = (1)(-36) - (-4)(21) = 48$$

since $\det(\Theta) \neq 0$

Since $\det(\Theta) = 48$, the system is observable

Observability check provided in MATLAB Code as well.

- Full state observer design:

Given the desired poles $[-1, -1]$, the observer gain matrix, L was calculated in MATLAB to be $\begin{bmatrix} -0.25 \\ -3.3125 \end{bmatrix}$

- The Luenberger Full state observer =

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

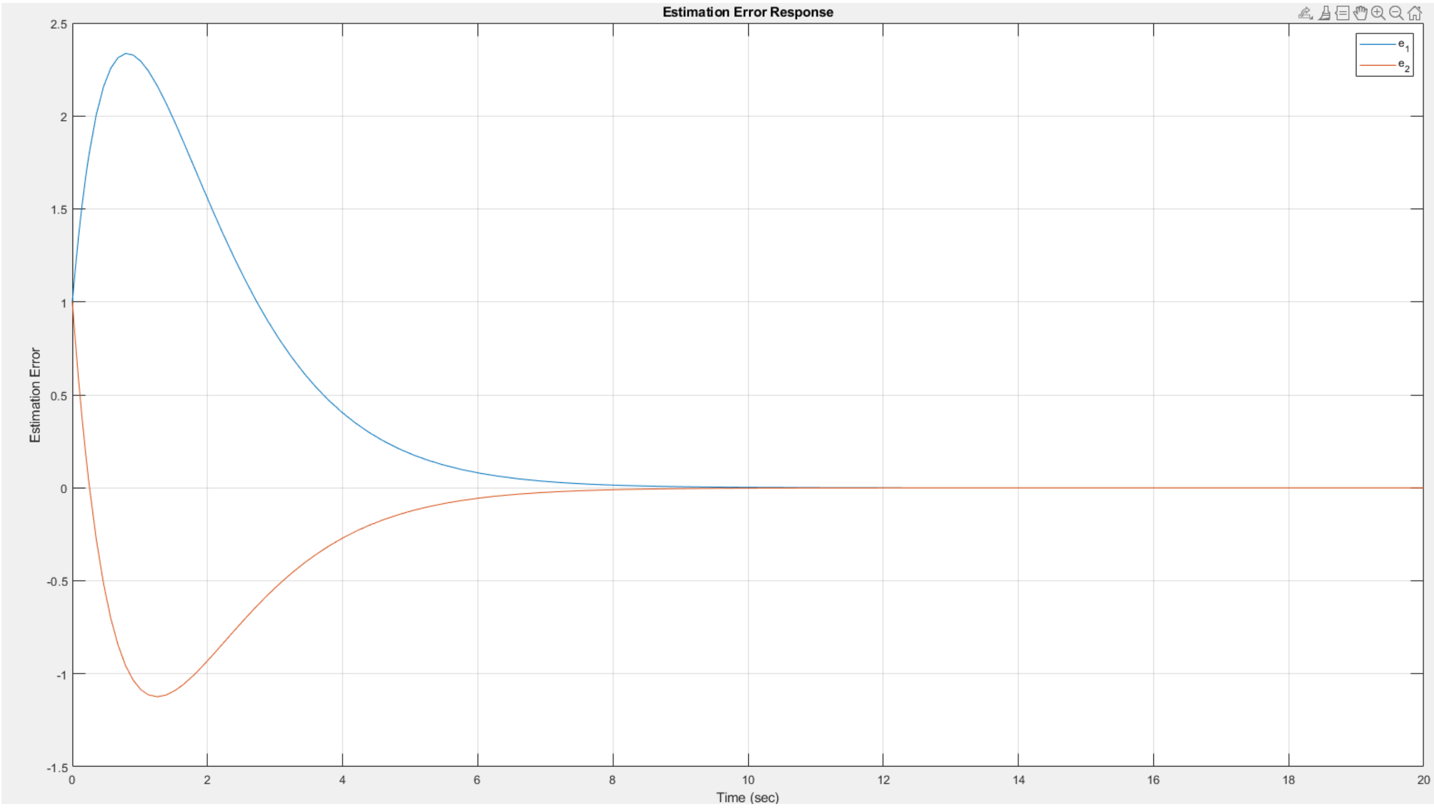
$$\dot{\hat{x}} = \begin{bmatrix} 1 & 4 \\ -5 & 10 \end{bmatrix}\hat{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u - \begin{bmatrix} -0.25 \\ -3.3125 \end{bmatrix}(y - [1 \ -4]\hat{x})$$

MATLAB Code for the system provided below.

MATLAB Code:

```
% ME 531 Pset 4 - Question 1
%Full state observer design

clear all;
A = [1, 4; -5, 10];
B = [0; 1];
C = [1 -4];

%check whether this system is observable
obsv_mat = obsv(A, C);
obsv_mat_rank = rank(obsv_mat);

if obsv_mat_rank == length(A)
    disp("System is fully observable")
else
    disp("System is not fully observable")
end

% Desired poles
op1 = -1;
op2 = -1.0000001; %MATLAB error when op2 is set to -1, so i made it a bit smaller

% the observer gain matrix .
L = place(A',C',[op1 op2]);

% initial estimation error
e0 = [1; 1];

% Error dynamics
sys = A-(L'*C);
f = @(t,e) [sys(1,1)*e(1)+sys(1,2)*e(2); sys(2,1)*e(1)+sys(2,2)*e(2)];

%Error calc.
[ts,ys] = ode45(f,[0,20],e0); %simulate response for 20 sec
plot(ts,ys(:,1),ts,ys(:,2))
xlabel('Time (sec)')
ylabel('Estimation Error')
title('Estimation Error Response')
legend('e_1','e_2')
grid on
```

MATLAB Plot:

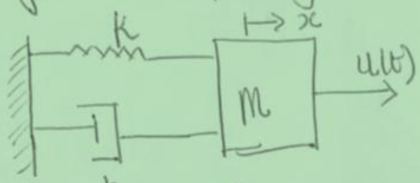


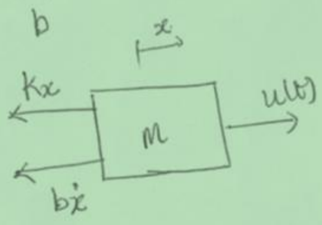
Estimation Error Response
e_1
e_2
Estimation Error
Time (sec)
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
0
2
4
6
8
10
12
14
16
18
20

Question 2:

Question 2:

Spring mass damper system





Equation of motion

$$m\ddot{x} = \sum F_x$$

$$m\ddot{x} = u(t) - kx - b\dot{x}$$

$$\ddot{x} = -\frac{k}{m}x - \frac{b}{m}\dot{x} + \frac{1}{m}u(t)$$

In state space form: $\dot{x} = Ax + Bu$, with position (x) and velocity ($\dot{x}$) as state variables.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u(t)$$

Output equation $y = Cx = [1 \ \ 0] \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$ since we are interested in the position (laser range finder tracks position)

$$A = \begin{bmatrix} 0 & 1 \\ -k/m & -b/m \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2/5 & -2/5 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1/m \end{bmatrix} = \begin{bmatrix} 0 \\ 1/5 \end{bmatrix}$$

$$C = [1 \ \ 0]$$

Initial conditions of the system = $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

% Overshoot of the full state feedback controller = 0.04 = $e^{\left(-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}\right)}$

$$\zeta = \sqrt{\frac{(\ln(0.04))^2}{\pi^2 + (\ln(0.04))^2}} = 0.7156$$

Damping ratio, $\zeta = 0.7156$

Setting time, $T_s = \frac{4}{\zeta\omega_n} = 2$

$$\omega_n = \frac{4}{2\zeta} = \frac{2}{0.7156} = 2.7947$$

Natural frequency, $\omega_n = 2.7947$

With the A and B matrices, the controllability of the system was checked in MATLAB (code provided) and the system was controllable.

Same was done to check the observability of the system using matrices A and C, and the system was also observable.

For the controller design:

The characteric equation = $\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2$

$$= \lambda^2 + 2(0.7156 \times 2.7949)\lambda + (2.7949)^2$$

$$= \lambda^2 + 4\lambda + 7.8115$$

With the roots function in MATLAB & the pole placement function (code provided) the state feedback gain matrix (K) was = $[37.0512, 18]$

∴ The controller, with $u = -Kx$

$$= \dot{x} = Ax - BKx$$

$$= \begin{bmatrix} 0 & 1 \\ -0.4 & -0.4 \end{bmatrix} x - \begin{bmatrix} 0 \\ 0.2 \end{bmatrix} \begin{bmatrix} 37.0512 & 18 \end{bmatrix} x$$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -7.8115 & -4 \end{bmatrix} x$$

For the observer:

The gain matrix, L was $= \begin{bmatrix} 3.6 \\ 5.9702 \end{bmatrix}$

The Luenberger full state observer =

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

$$\hat{x} = \begin{bmatrix} 0 & 1 \\ -0.4 & -0.4 \end{bmatrix} \hat{x} + \begin{bmatrix} 0 \\ 0.2 \end{bmatrix} u - \begin{bmatrix} 3.6 \\ 5.9702 \end{bmatrix} \left(y - \begin{bmatrix} 1 & 0 \end{bmatrix} \hat{x}\right)$$

The system simulation setup and plots are shown in the MATLAB Code below.

MATLAB Code:

```matlab
% ME 531 Pset 4 - Question 2
% Design of a Compensator (State Feedback + Observer)
% for a Spring-Mass-Damper System

%% system parameters
m = 5;  % Mass (kg)
k = 2;  % Spring constant (N/m)
b = 2;  % Damping coefficient (N-s/m)

% State-space form
A = [0 1; -k/m -b/m];
B = [0; 1/m];
C = [1 0]; % pos. measurement
D = 0;


%% Controller Design (State Feedback)

%check whether this system is controllable
contr_mat = ctrb(A, B);
contr_mat_rank = rank(contr_mat);

if contr_mat_rank == length(A)
    disp("System is fully controllable")
else
    disp("System is not fully controllable")
end

% Desired performance
OS_percent = 4; % perc. overshoot = 4%
Ts = 2;         % 2s Settling time

% Desired damping ratio (zeta) and natural frequency (omega_n)
OS = OS_percent / 100;
zeta = -log(OS) / sqrt(pi^2 + log(OS)^2);
omega_n = 4 / (zeta * Ts);

% Desired closed-loop poles for the controller
p_control = roots([1 2*zeta*omega_n omega_n^2]);

% state feedback gain matrix (K) using pole placement
K = place(A, B, p_control);
```

```matlab
%% Observer Design

%check whether this system is observable
obsv_mat = obsv(A, C);
obsv_mat_rank = rank(obsv_mat);

if obsv_mat_rank == length(A)
    disp("System is fully observable")
else
    disp("System is not fully observable")
end

% observer gain matrix (L) using pole placement
L = place(A', C', p_control)';
```

```matlab
%% System Simulation Setup
x0 = [1; 1];          % System initial state (1m position, 1 m/s velocity)
e0 = [0.5; -0.5];   % Initial estimation error
x_hat0 = x0 - e0;   % Initial observer state
x_combined0 = [x0; e0]; % Initial state for the combined system [x; e]

t = 0:0.01:10; % Simulate for 10 seconds

% 1. State Feedback System (for comparison)
%    x_dot = (A - B*K)*x
A_sf = A - B*K;
B_sf = zeros(2, 1);
C_sf = eye(2);
D_sf = zeros(2, 1);
sys_sf = ss(A_sf, B_sf, C_sf, D_sf);

% 2. Output Feedback System (Controller + Observer)
%    x_dot = A*x - B*K*x_hat
%    x_hat_dot = A*x_hat + B*u + L*(y - C*x_hat)
%    Using state vector [x; e], where e = x - x_hat
%    [x_dot] = [A-BK   BK ] [x]
%    [e_dot] = [ 0    A-LC] [e]
A_of = [A - B*K, B*K; zeros(size(A)), A - L*C];
B_of = zeros(4, 1);
C_of = eye(4);
D_of = zeros(4, 1);
sys_of = ss(A_of, B_of, C_of, D_of);
```

```matlab
%% Run Simulations
% Simulate state feedback system
[y_sf, t_sf, x_sf] = initial(sys_sf, x0, t);

% Simulate output feedback system
[y_of, t_of, x_of] = initial(sys_of, x_combined0, t);

% Extract states and error from output feedback simulation
x_obs = x_of(:, 1:2); % System states (x)
e_obs = x_of(:, 3:4); % Estimation error (e)
x_hat_obs = x_obs - e_obs; % Estimated states (x_hat)

% Calculate actuation expenditure (u = -K * x_hat)
u_obs = -K * x_hat_obs';


% Figure 1: System Response Comparison
figure('Name', 'System Response Comparison', 'NumberTitle', 'off');
plot(t_sf, x_sf(:, 1), 'b-', 'LineWidth', 1.5, 'DisplayName', 'x_1: pos (State Fbk)'); hold on;
plot(t_sf, x_sf(:, 2), 'b--', 'LineWidth', 1.5, 'DisplayName', 'x_2: vel (State Fbk)');
plot(t_of, x_obs(:, 1), 'r-', 'LineWidth', 1.5, 'DisplayName', 'x_1: pos (Output Fbk)');
plot(t_of, x_obs(:, 2), 'r--', 'LineWidth', 1.5, 'DisplayName', 'x_2: vel (Output Fbk)');
title('System Response: State Feedback vs. Output Feedback');
xlabel('Time (s)');
ylabel('State Values (Position m, Velocity m/s)');
legend('show', 'Location', 'northeast');
grid on;
set(gca, 'FontSize', 12);



% Figure 2: State Estimate Error Convergence
figure('Name', 'State Estimate Error', 'NumberTitle', 'off');
plot(t_of, e_obs(:, 1), 'm-', 'LineWidth', 1.5, 'DisplayName', 'pos Error e_1'); hold on;
plot(t_of, e_obs(:, 2), 'c-', 'LineWidth', 1.5, 'DisplayName', 'vel Error e_2');
title('State Estimate Error Convergence (e = x - \hat{x})');
xlabel('Time (s)');
ylabel('Estimation Error');
legend('show', 'Location', 'northeast');
grid on;
set(gca, 'FontSize', 12);



% Figure 3: Actuation Expenditure
figure('Name', 'Actuation Expenditure', 'NumberTitle', 'off');
plot(t_of, u_obs, 'b-', 'LineWidth', 1.5);
title('Actuation Expenditure using State Estimate (u = -K\hat{x})');
xlabel('Time (s)');
ylabel('Control Input (Force, N)');
grid on;
set(gca, 'FontSize', 12);
```

MATLAB plots:

Figure 1:

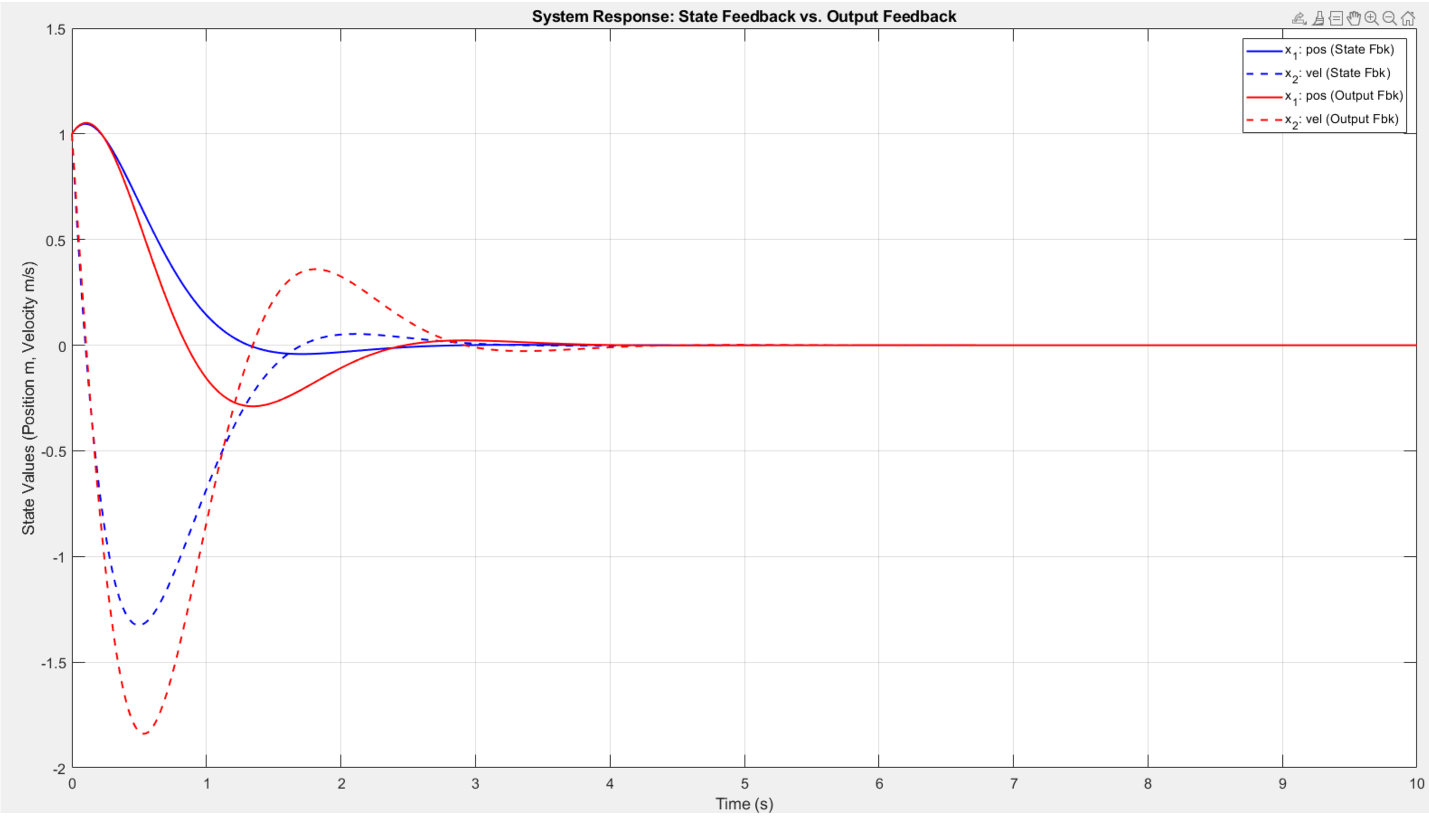


Figure 2:

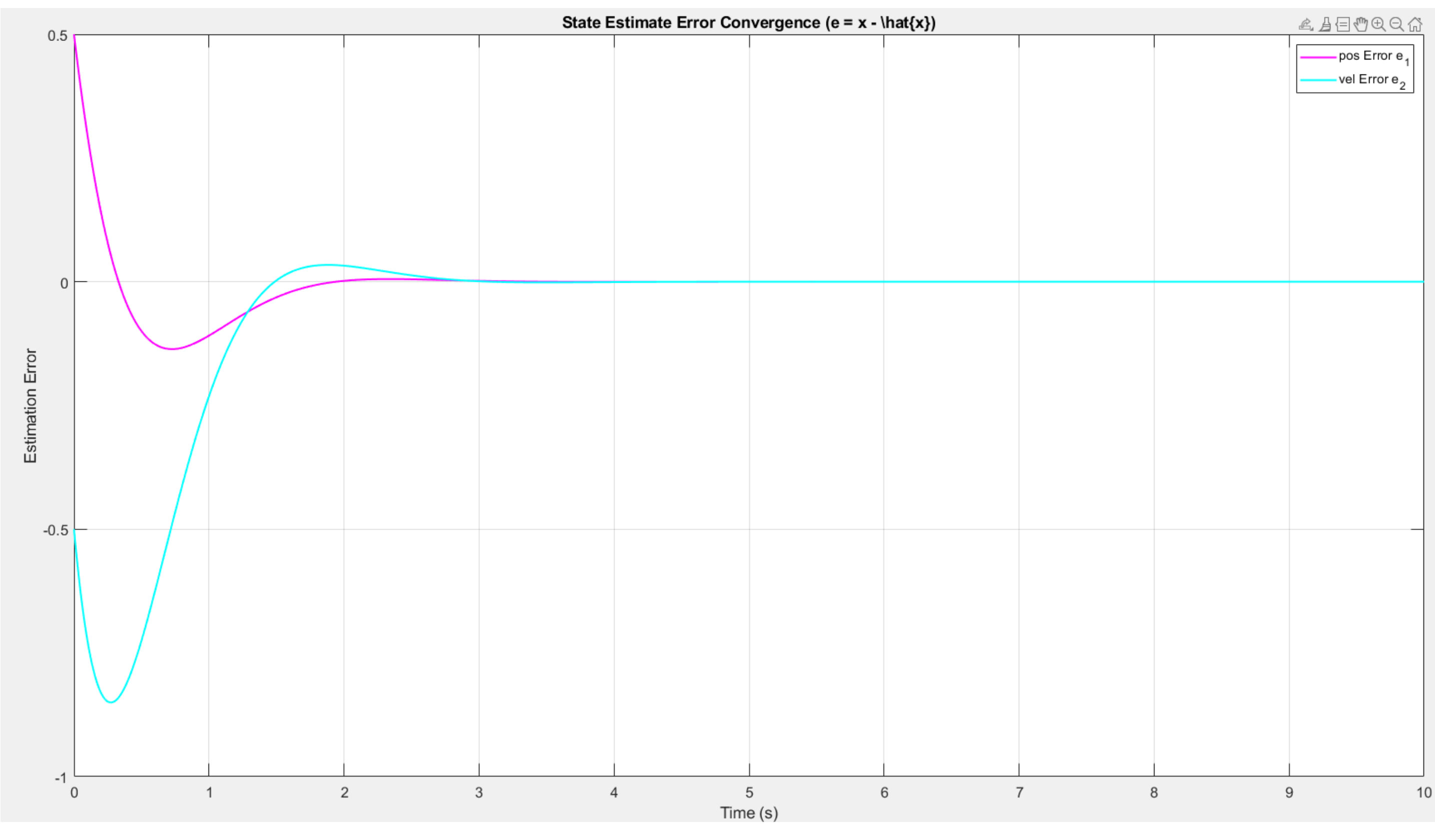
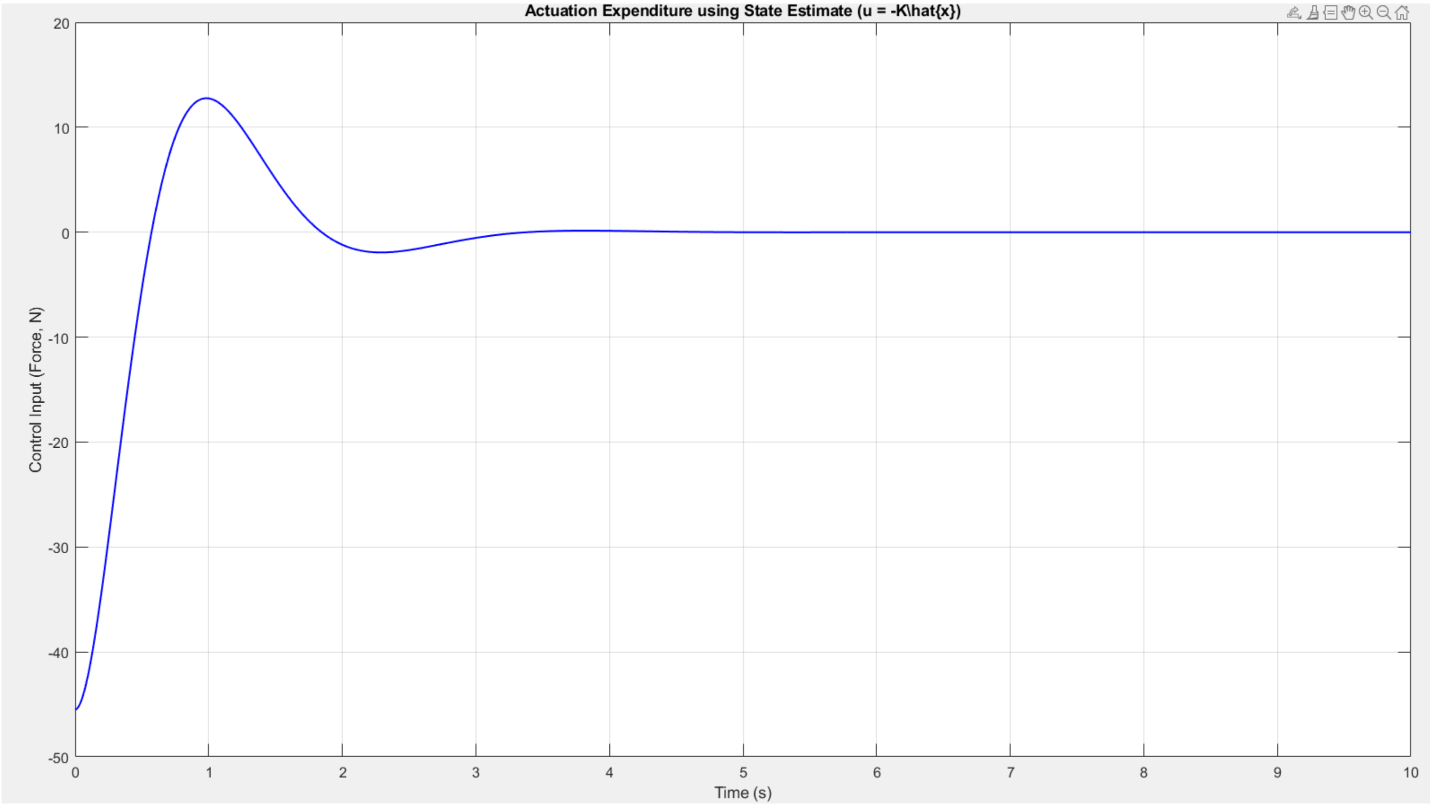


Figure 3:


Actuation Expenditure using State Estimate (u = -K\hat{x})
Control Input (Force, N)
Time (s)
20
10
0
-10
-20
-30
-40
-50
0
1
2
3
4
5
6
7
8
9
10