ENAE488O Multi-Robot Swarms HW1 | Romeo Perlstein

Tuesday, March 25, 2025 11:09 AM

Q1

1) Given That:

$$\Delta(G) = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A(G) = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$L = \Delta(G) - A(G)$$

$$L = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

Thus:

$$\dot{\vec{x}}(t) = -L\vec{x}$$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \\ \dot{x}_5(t) \end{bmatrix} = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \end{bmatrix}$$

2) <u>Agreement value</u>

<u>using</u>

$$c = \sum_{i=1}^{N} p_i x_i(0)$$, <u>Left eigenvector:</u> $\lambda_{1_L} = [p_1, p_2, \dots p_N]$

$$\boxed{C = 5.8138}$$

<u>From propogation</u>

$$\boxed{2.600}$$

<u>time to agreement:</u>

$$\boxed{t_{agree} \cong 6.21}$$

(rounding state to 3 decimals)

3) see matlab

4) <u>given that</u>:

$x_i \in R^2$

$$\dot{\vec{x}} = \begin{bmatrix} \dot{x}_1 & y_1 \\ \dot{x}_2 & \dot{y}_2 \\ x_3 & \dot{y}_3 \\ \dot{x}_4 & \dot{y}_4 \\ x_5 & \dot{y}_5 \end{bmatrix}$$

For $x_i \in R^N$, $N > 1$, the agreement dynamics become:

$$\dot{\vec{x}} = -(L \otimes I)\vec{x}$$

where $\otimes$ is the kronecker product.

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

<u>then</u>:

$$L \otimes I =$$

$$
\begin{bmatrix}
2 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 2 & 0 & -1 & & & & & -1 \\
-1 & 0 & 2 & 0 & -1 & & & & 0 \\
0 & -1 & 0 & 2 & 0 & -1 & & & 0 \\
0 & & -1 & 0 & 2 & 0 & -1 & & 0 \\
0 & & & -1 & 0 & 2 & 0 & -1 & 0 \\
0 & & & & -1 & 0 & 2 & 0 & -1 \\
-1 & & & & & -1 & 0 & 2 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 2
\end{bmatrix}
$$

Then finally $\dot{x}$

$$
\dot{\vec{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \\ \dot{y}_5 \end{bmatrix} =
\begin{bmatrix}
2 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 2 & 0 & -1 & & & & & & -1 \\
-1 & 0 & 2 & 0 & -1 & & & & & 0 \\
0 & -1 & 0 & 2 & 0 & -1 & & & & 0 \\
0 & & -1 & 0 & 2 & 0 & -1 & & & 0 \\
0 & & & -1 & 0 & 2 & 0 & -1 & & 0 \\
0 & & & & -1 & 0 & 2 & 0 & -1 & 0 \\
0 & & & & & -1 & 0 & 2 & 0 & -1 \\
-1 & & & & & & -1 & 0 & 2 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2
\end{bmatrix}
\underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}}_{\vec{x}}
$$

5) from MATLAB integration.

for node 2, node 4

$$\boxed{\text{Agreement value } [0, 0.8]}$$

Agreement value [0, 0.8]

Agreement time · 7.1750

For node 1, node 3, node 5

Agreement value : [0.8, 0]

Agreement time . 7 35

6) see MATLAB for plots

Interestingly, two of the kilobots rendezvous at one location, while the other three kilobots rendezvous at a different location, but these locations have their x and y values flipped! I'm not sure why this is the case but it's interesting that they rendezvous to these locations separately yet the locations they go to are still connected.

Seems like the kilobots that are the farthest north and south converge together, while the kilobots that are closet to the x and y axis converge together.

Q2)

1) In the sync lab, the robots attempt to synchronize their LED flashing period with the other kilobots in the network, so that each kilobot is flashing in sync (hence the name). The kilobots do this by converting their clock value to be a number bounded between 0-31 and storing it in a separate variable called modulo_clock and sending it out to the network.

When a kilobot receives a neighbors modulo_clock, it checks if its 0, and if it's not it then checks if the offset between the local modulo_clock and the neighbors modulo_clock is less than 16 (half the period). If the offset is less than 16, the offset is stored locally , however if it's greater than 0, the offset is adjusted by taking the max modulo_clock value (32) and subtracting the neighbors modulo clock, so that a kilobot will sync with its neighbor while ensuring its neighbor doesn't try to sync with the kilobot. Finally, the LED only blinks when the local modulo_clock is 0, and at the same time find the average offset from its neighbors and adjusts its own local offset!

2) Blue!

3) We know that the kilobot has received all of the code becaues it goes from slowly flashing blue, to quickly flashing blue and green, and when all of the code has been transferred the kilobot quickly blinks green

Contents


```
%   ENAE4880        %
% Romeo Perlstein   %
%      HW1          %
% I thought the only lonely place, was on the moon!
clear; clc; close all
```

Q1

```
fprintf("Q1)\n\n")
fprintf("1)\n")
```

1

Given the graph in fig 1., we can find the degree matrix as the following:

```
num_nodes = 5;
degree_mat = [
    2 0 0 0 0;
    0 2 0 0 0;
    0 0 2 0 0;
    0 0 0 2 0;
    0 0 0 0 2];
fprintf("Degree Matrix:\n")
disp(degree_mat);

adjacency_mat = [
    0 1 0 0 1;
    1 0 1 0 0;
    0 1 0 1 0;
    0 0 1 0 1;
    1 0 0 1 0];
fprintf("Adjacency Matrix:\n")
disp(adjacency_mat);

lapacian_mat = degree_mat - adjacency_mat;
fprintf("Laplacian Matrix\n")
disp(lapacian_mat);

L_rank = rank(lapacian_mat);

fprintf("Network is ")
if(L_rank == num_nodes - 1)
    fprintf("CONNECTED\n")
```

```
else
    fprintf("NOT CONNECTED\n")
end


[r_eig_vec, eig_vals_mat, l_eig_vec] = eig(lapacian_mat);
for i=1:num_nodes
    eig_vals(i) = eig_vals_mat(i,i);
end
eig_vals = eig_vals';
fprintf("Eigenvalues of Laplacian Matrix:\n")
disp(eig_vals);
fprintf("Right Eigenvectors:\n")
disp(r_eig_vec)
fprintf("Left Eigenvectors:\n")
disp(l_eig_vec)

x0 = [-2; 4; -1; 7; 5];
fprintf("Initial Starting States:\n")
disp(x0);
```

2

```
fprintf("2)\n")
c = 0;
for i=1:num_nodes
    c = c + (l_eig_vec(i,1)*x0(i));
end
fprintf("Agreement Value From Lecture:\n")
disp(c)

tau(:) = 1/eig_vals(2);
fprintf("Speed of Convergence From Lecture:\n")
disp(tau);

% ODE SETUP
tall_er_ant = (10^-13); % Tolerance
step_size = 0.01; % step size
max_time = 10; % max time (0->max_time)
t = [0:step_size:max_time]; % timestep

% ODE options
ODE_options = odeset("RelTol", tall_er_ant, "AbsTol", tall_er_ant);

[T,X] = ode45(@myodefun, t, x0, ODE_options, lapacian_mat);

rounded_X = round(X, 3);
for i=1:length(t)
    if((rounded_X(i,1) == rounded_X(i,2)) && (rounded_X(i,2) == rounded_X(i,3)) && (rounded_X(i,3)
== rounded_X(i,4)) && (rounded_X(i,4) == rounded_X(i,5)))
        agreement_val = round(X(i,1), 3);
        agreement_time = T(i);
        break
    end
end
fprintf("Agreement Value (From Integration of Agreement Protocol - rounded to 3 decimal places):\n")
disp(agreement_val)
fprintf("Agreement Time (From Integration of Agreement Protocol):\n")
disp(agreement_time)
```

3

```
fprintf("3)\n")
plot(T,X)
title("Q1-3) Plot of Each Node's State Over Time")
xlabel("Time (seconds)")
ylabel("Position")
legend(["x_1", "x_2", "x_3", "x_4", "x_5"]);
grid on

for i=1:num_nodes
    figure
    plot(T,X(:,i))
    title(["Q1-3) Plot of x_" + int2str(i) + " State Over Time"])
    xlabel("Time (seconds)")
    ylabel("Position")
    grid on
end
```

4

```
fprintf("4)\n")
Id = eye(2);
LxId = kron(lapacian_mat, Id);
fprintf("Kronecker Product of L and $I_D:\n")
disp(LxId)

% ODE SETUP
tall_er_ant = (10^-13); % Tolerance
step_size = 0.025; % step size
max_time = 10; % max time (0->max_time)
t = [0:step_size:max_time]; % timestep

% ODE options
ODE_options = odeset("RelTol", tall_er_ant, "AbsTol", tall_er_ant);
```

5)

```
fprintf("5)\n")
x0 = [-4;
      -2;
       1;
       7;
       5;

      -5;
       2;
       6;
       0;
      -6];
[T,X] = ode45(@myodefunR2, t, x0, ODE_options, LxId);

% Find time to consensus
agreement_val = [0,0];
agreement_time = 0;
for i=1:length(t)
```

```
    if(round(abs(X(i,2) - X(i,4)), 4) <= 1e-4)
        agreement_val = [X(i,2), X(i,5+2)];
        agreement_time = T(i);
        break
    end
end
fprintf("Agreement Value for x_2 and x_4 (From Integration of Agreement Protocol - rounded to 3 decimal places):\n")
disp(agreement_val)
fprintf("Agreement Time for x_2 and x_4 (From Integration of Agreement Protocol):\n")
disp(agreement_time)
agreement_val = [0,0];
agreement_time = 0;
for i=1:length(t)
    if(round(abs(X(i,3) - X(i,5)), 4) <= 1e-4)
        agreement_val = [X(i,3), X(i,5+3)];
        agreement_time = T(i);
        break
    end
end
fprintf("Agreement Value for x_1, x_3, and x_5 (From Integration of Agreement Protocol - rounded to 3 decimal places):\n")
disp(agreement_val)
fprintf("Agreement Time for x_1, x_3, and x_5 (From Integration of Agreement Protocol):\n")
disp(agreement_time)
```

6

```
fprintf("6)\n")
figure
plot(T,X)
title("Q1-6) Plot of Each Node's State Over Time")
xlabel("Time (seconds)")
ylabel("Position")
legend(["x_1","x_2","x_3","x_4","x_5","y_1","y_2","y_3","y_4","y_5"]);
grid on

figure
plot(T,X(:,1:5))
title("Q1-6) Plot of Each Node's X State Over Time")
xlabel("Time (seconds)")
ylabel("Position")
legend(["x_1","x_2","x_3","x_4","x_5"]);
grid on

figure
plot(T,X(:,6:10))
title("Q1-6) Plot of Each Node's Y State Over Time")
xlabel("Time (seconds)")
ylabel("Position")
legend(["y_1","y_2","y_3","y_4","y_5"]);
grid on

for i=1:num_nodes
    figure
    hold on
    plot(T,X(:,i))
    plot(T,X(:,5+i))
    title(["Q1-6) Plot of x_" + int2str(i) + " State Over Time"])
```

```
    xlabel("Time (seconds)")
    ylabel("Position")
    grid on
end

figure
% for i = 1:length(t)
%     hold off
%     scatter(X(i,1), X(i,6), "b")
%     hold on
%     scatter(X(i,2), X(i,7), "r")
%     scatter(X(i,3), X(i,8), "g")
%     scatter(X(i,4), X(i,9), "magenta")
%     scatter(X(i,5), X(i,10), "black")
%
%     plot(X(1:i,1), X(1:i,6), "b")
%     plot(X(1:i,2), X(1:i,7), "r")
%     plot(X(1:i,3), X(1:i,8), "g")
%     plot(X(1:i,4), X(1:i,9), "magenta")
%     plot(X(1:i,5), X(1:i,10), "black")
%     yline(0, "--black")
%     xline(0, "--black")
%     axis equal
%     grid on
%     title("Plot of Each Node's Position")
%     xlabel("X coordinate")
%     ylabel("Y coordinate")
%     legend(["x_1","x_2","x_3","x_4","x_5"]);
%
%     drawnow
% end
hold off
scatter(X(end,1), X(end,6), "b")
hold on
scatter(X(end,2), X(end,7), "r")
scatter(X(end,3), X(end,8), "g")
scatter(X(end,4), X(end,9), "magenta")
scatter(X(end,5), X(end,10), "black")

plot(X(1:end,1), X(1:end,6), "b")
plot(X(1:end,2), X(1:end,7), "r")
plot(X(1:end,3), X(1:end,8), "g")
plot(X(1:end,4), X(1:end,9), "magenta")
plot(X(1:end,5), X(1:end,10), "black")
yline(0, "--black")
xline(0, "--black")
axis equal
grid on
title("Q1-6) Plot of Each Node's Position")
xlabel("X coordinate")
ylabel("Y coordinate")
legend(["x_1","x_2","x_3","x_4","x_5"]);


function x_dot = myodefun(t, x, L)
    x_dot = -L*x;
end

function x_dot = myodefunR2(t, x, LxId)
    x_dot = -LxId*x;
end
```

Q1)

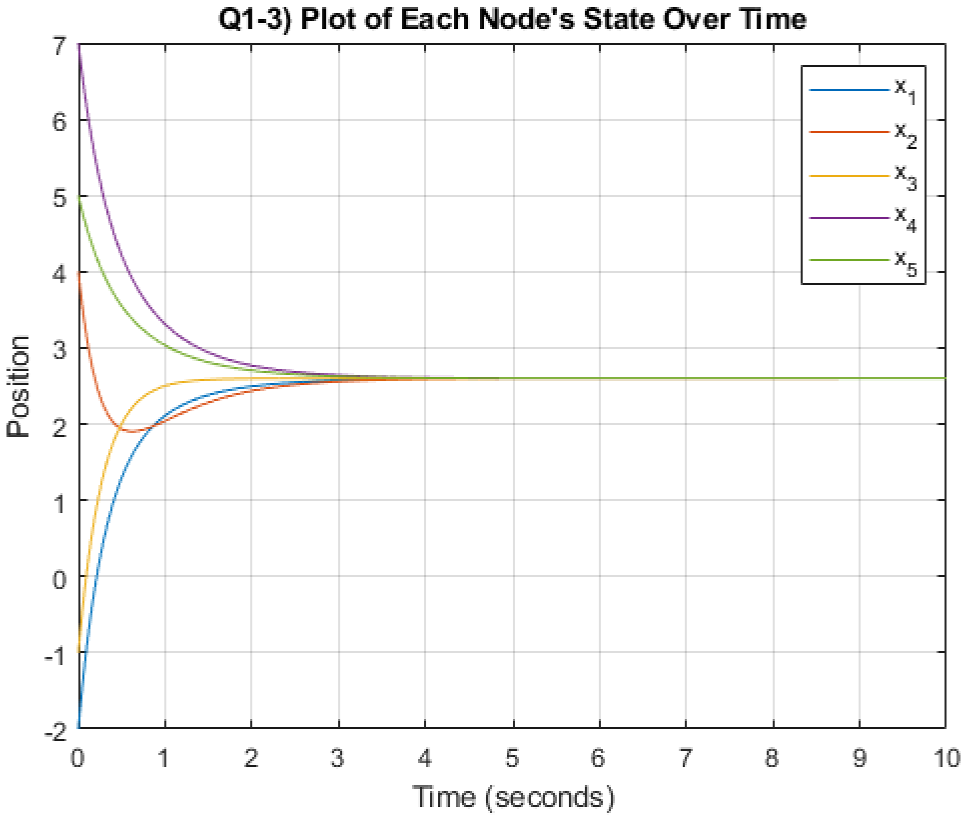
1)

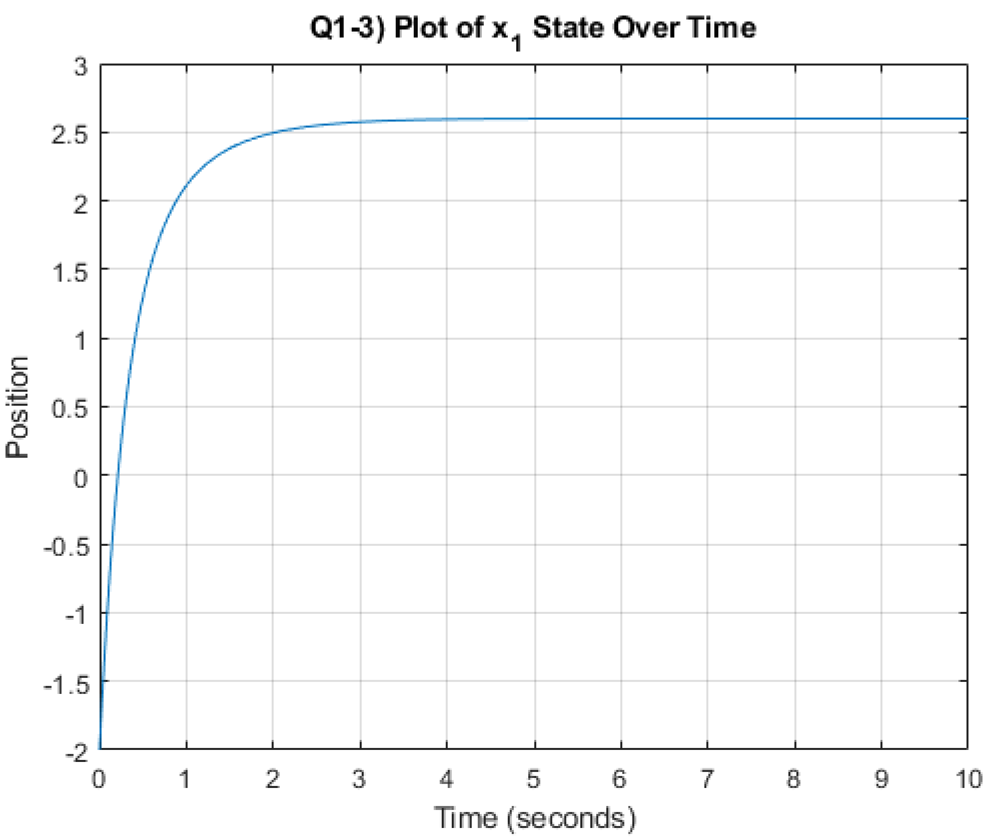
```
Degree Matrix:
     2     0     0     0     0
     0     2     0     0     0
     0     0     2     0     0
     0     0     0     2     0
     0     0     0     0     2

Adjacency Matrix:
     0     1     0     0     1
     1     0     1     0     0
     0     1     0     1     0
     0     0     1     0     1
     1     0     0     1     0

Laplacian Matrix
     2    -1     0     0    -1
    -1     2    -1     0     0
     0    -1     2    -1     0
     0     0    -1     2    -1
    -1     0     0    -1     2

Network is CONNECTED
Eigenvalues of Laplacian Matrix:
   -0.0000
    1.3820
    1.3820
    3.6180
    3.6180

Right Eigenvectors:
    0.4472    0.6286   -0.0697    0.2031   -0.5990
    0.4472    0.2605    0.5763   -0.5164    0.3652
    0.4472   -0.4676    0.4259    0.6324    0.0081
    0.4472   -0.5495   -0.3131   -0.5069   -0.3782
    0.4472    0.1280   -0.6194    0.1878    0.6039

Left Eigenvectors:
    0.4472    0.6286   -0.0697    0.2031   -0.5990
    0.4472    0.2605    0.5763   -0.5164    0.3652
    0.4472   -0.4676    0.4259    0.6324    0.0081
    0.4472   -0.5495   -0.3131   -0.5069   -0.3782
    0.4472    0.1280   -0.6194    0.1878    0.6039

Initial Starting States:
    -2
     4
    -1
     7
     5
```

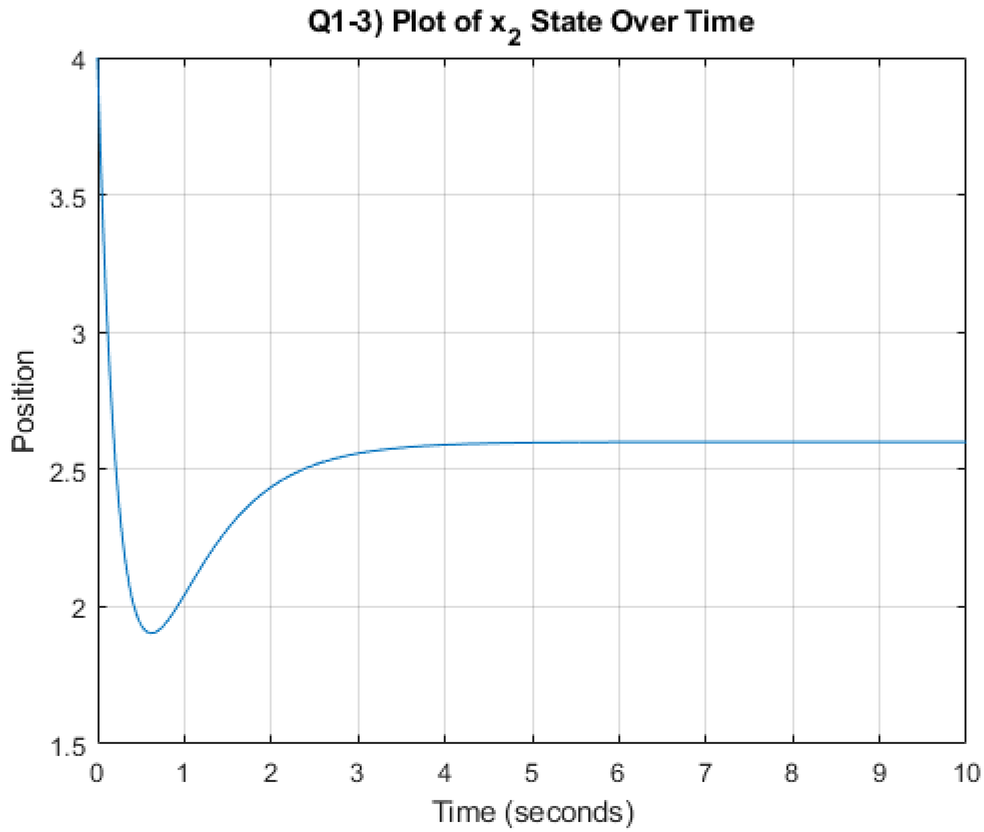
2)

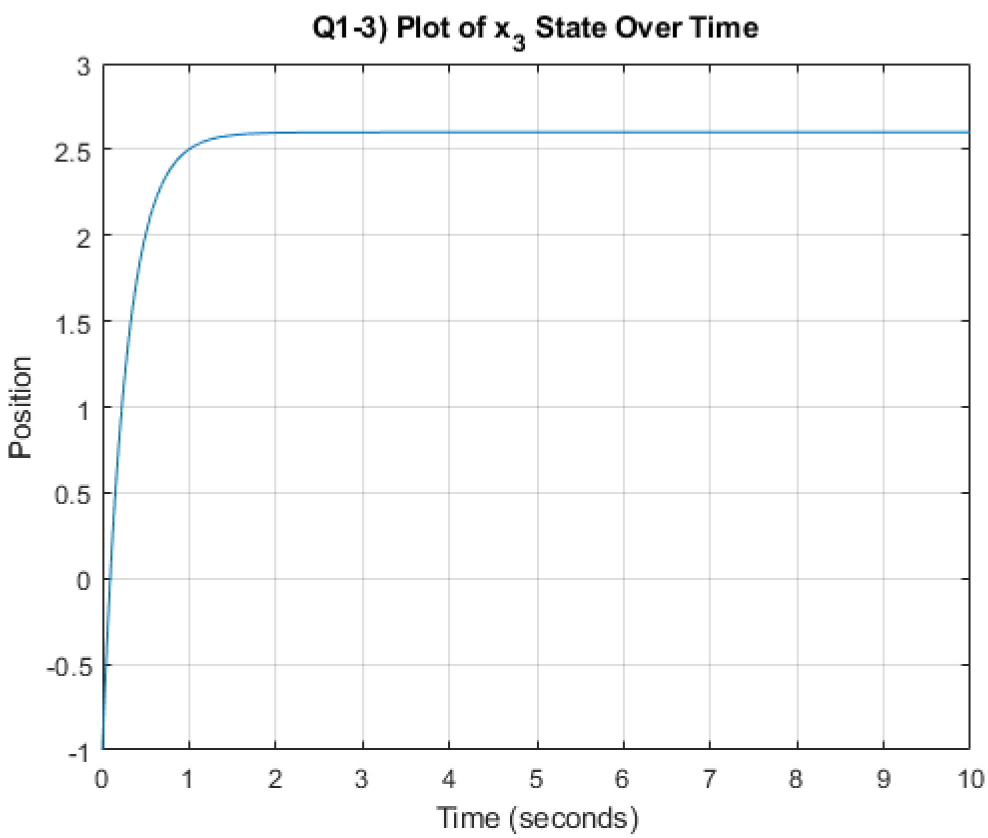
```
Agreement Value From Lecture:
    5.8138

Speed of Convergence From Lecture:
```

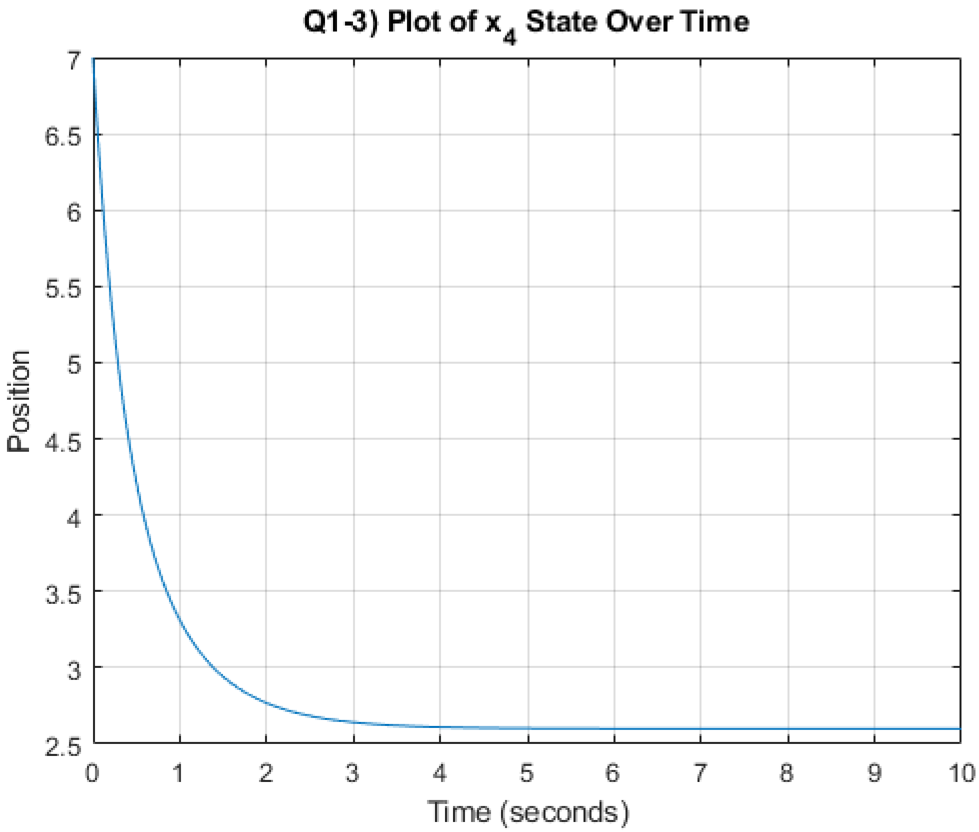
```
    0.7236

Agreement Value (From Integration of Agreement Protocol - rounded to 3 decimal places):
    2.6000

Agreement Time (From Integration of Agreement Protocol):
    6.2100

3)
4)
Kronecker Product of L and $I_D:
     2     0    -1     0     0     0     0     0    -1     0
     0     2     0    -1     0     0     0     0     0    -1
    -1     0     2     0    -1     0     0     0     0     0
     0    -1     0     2     0    -1     0     0     0     0
     0     0    -1     0     2     0    -1     0     0     0
     0     0     0    -1     0     2     0    -1     0     0
     0     0     0     0    -1     0     2     0    -1     0
     0     0     0     0     0    -1     0     2     0    -1
    -1     0     0     0     0     0    -1     0     2     0
     0    -1     0     0     0     0     0    -1     0     2

5)
Agreement Value for x_2 and x_4 (From Integration of Agreement Protocol - rounded to 3 decimal
places):
   -0.0000    0.8001

Agreement Time for x_2 and x_4 (From Integration of Agreement Protocol):
    7.1750

Agreement Value for x_1, x_3, and x_5 (From Integration of Agreement Protocol - rounded to 3 decimal
places):
    0.8000   -0.0000

Agreement Time for x_1, x_3, and x_5 (From Integration of Agreement Protocol):
    7.3500

6)
```

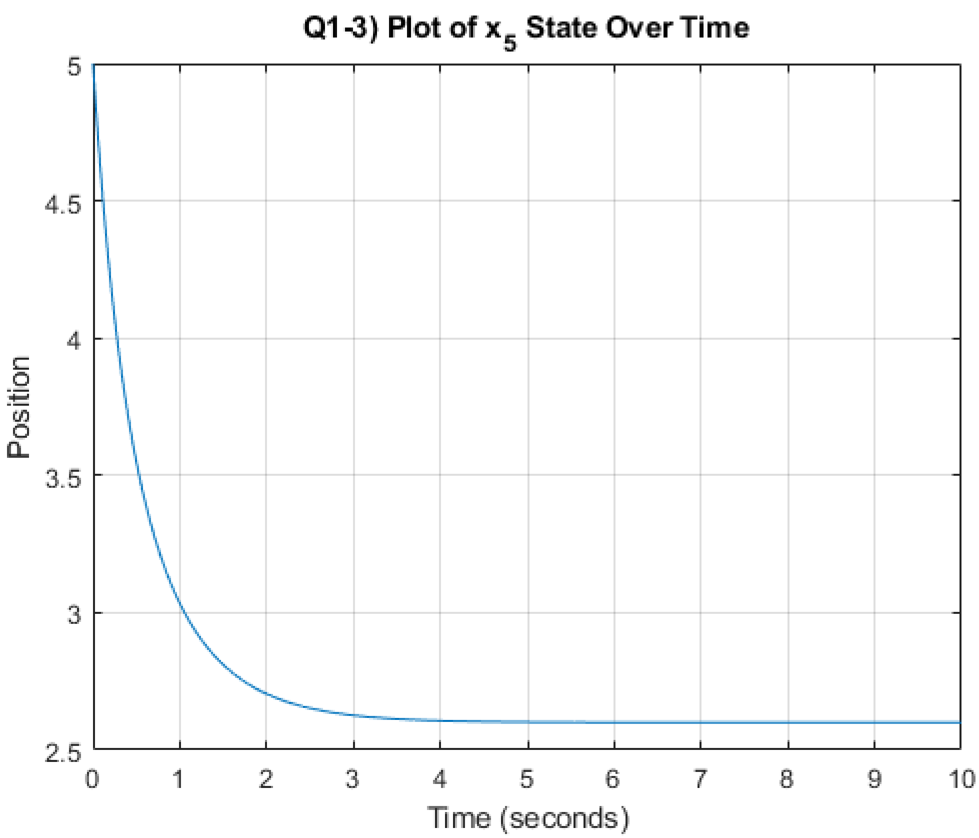

Q1-3) Plot of Each Node's State Over Time
Position
Time (seconds)
x_1
x_2
x_3
x_4
x_5

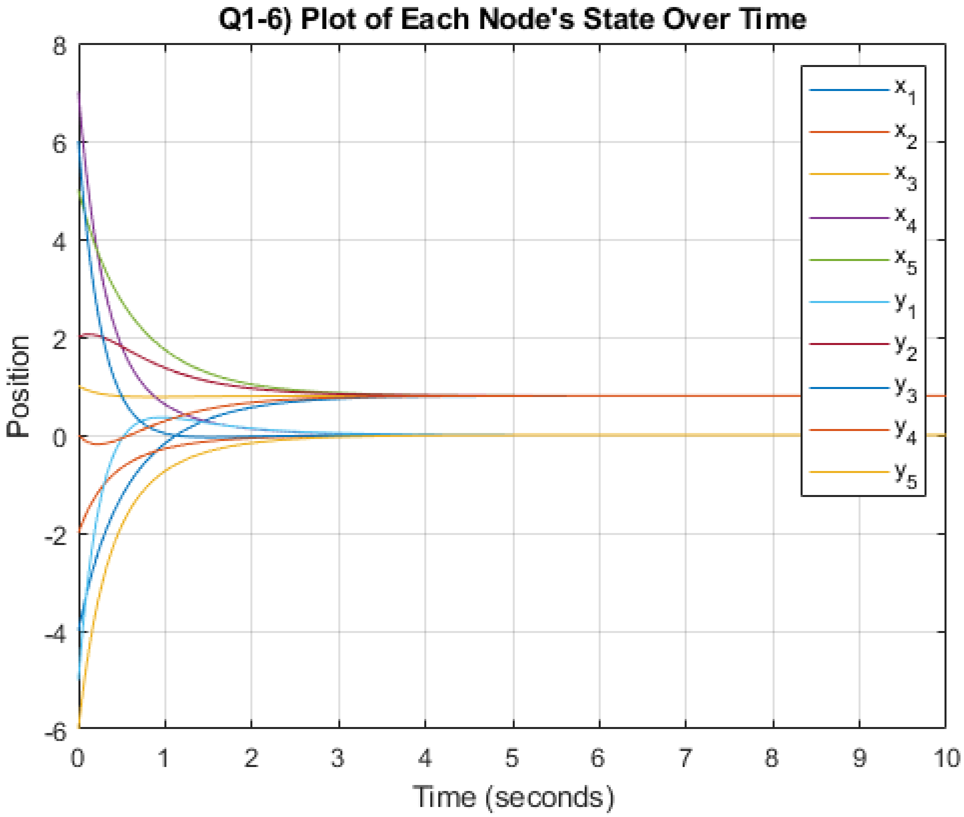

Q1-3) Plot of x_1 State Over Time
Position
Time (seconds)

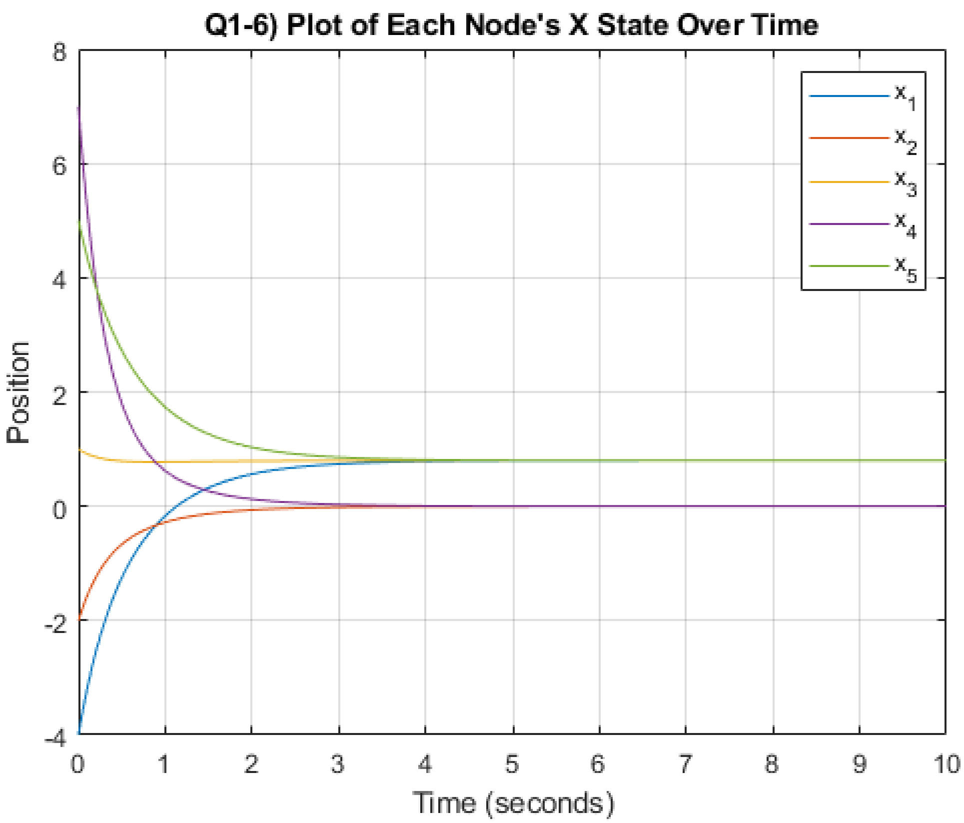

Q1-3) Plot of x2 State Over Time
Position
Time (seconds)
4
3.5
3
2.5
2
1.5
0
1
2
3
4
5
6
7
8
9
10

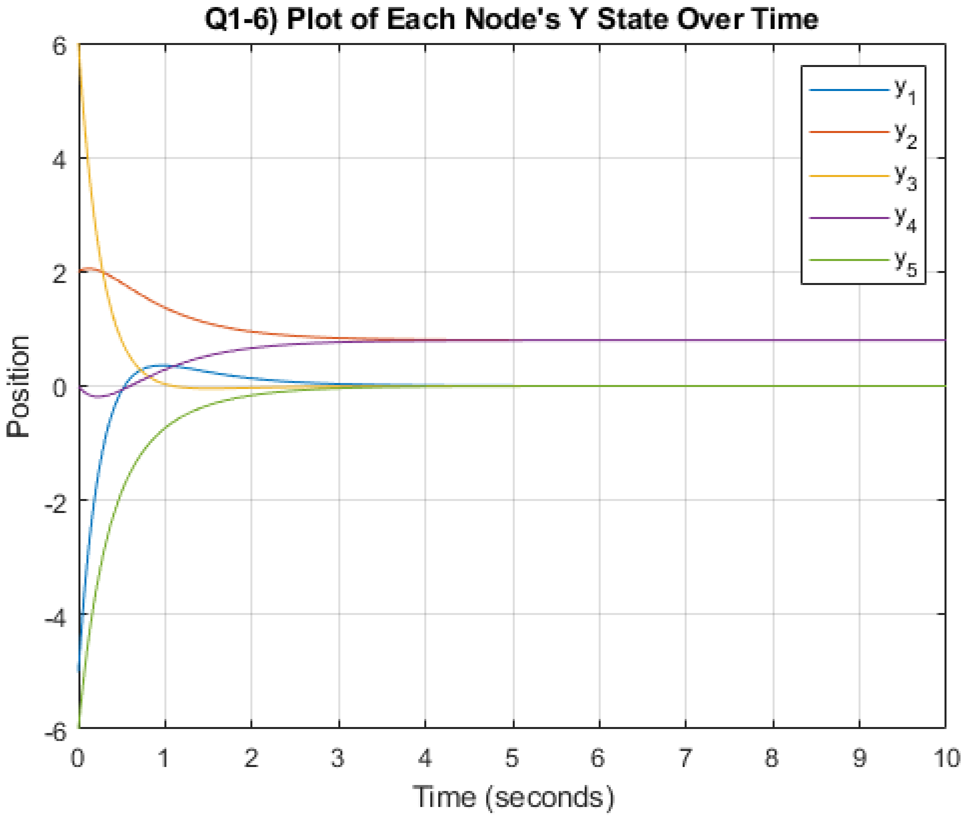

Q1-3) Plot of x3 State Over Time
Position
Time (seconds)
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
1
2
3
4
5
6
7
8
9
10

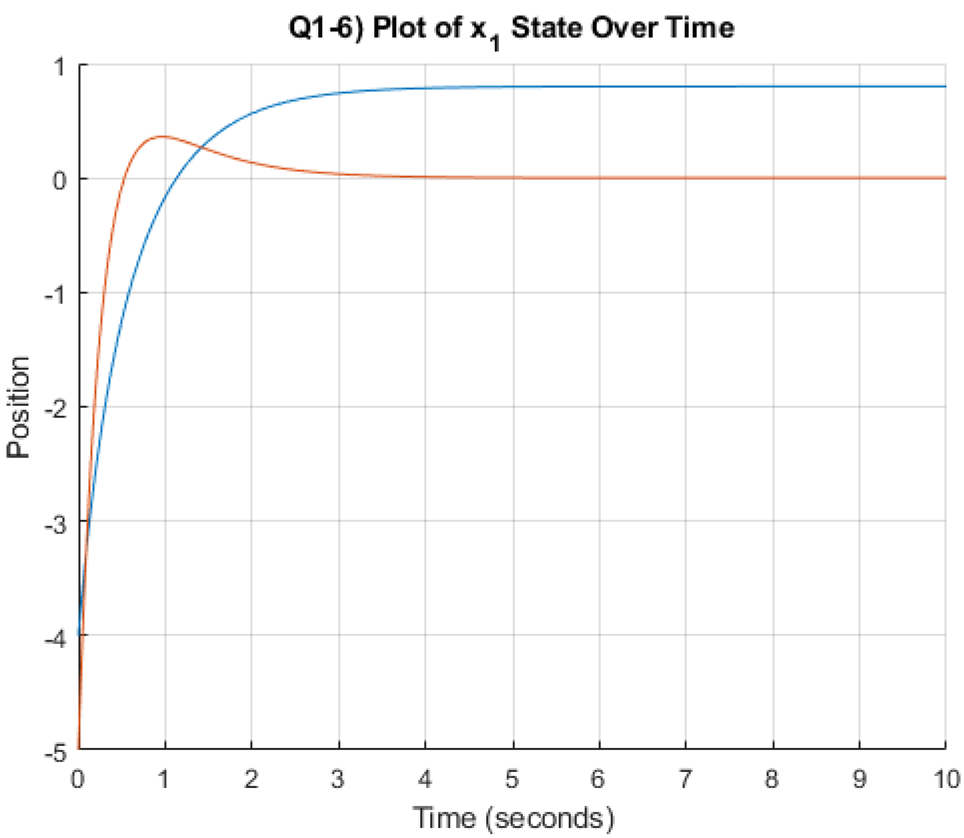

Q1-3) Plot of x4 State Over Time
Position
Time (seconds)
7
6.5
6
5.5
5
4.5
4
3.5
3
2.5
0
1
2
3
4
5
6
7
8
9
10

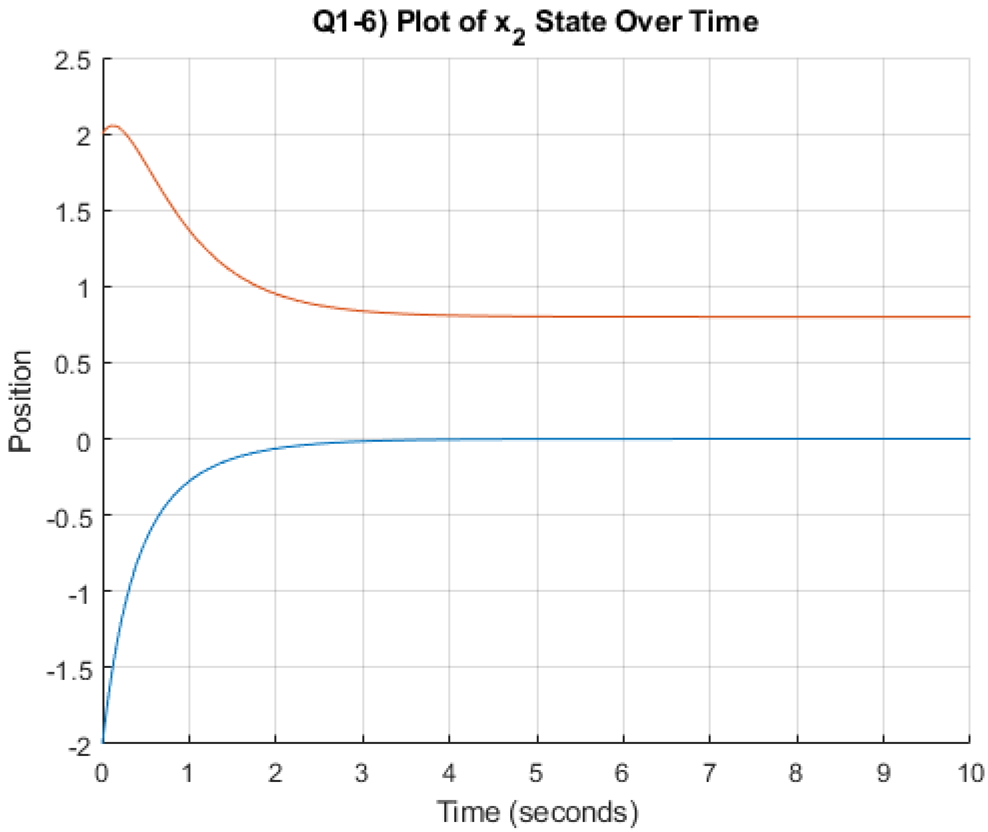

Q1-3) Plot of x5 State Over Time
Position
Time (seconds)
5
4.5
4
3.5
3
2.5
0
1
2
3
4
5
6
7
8
9
10

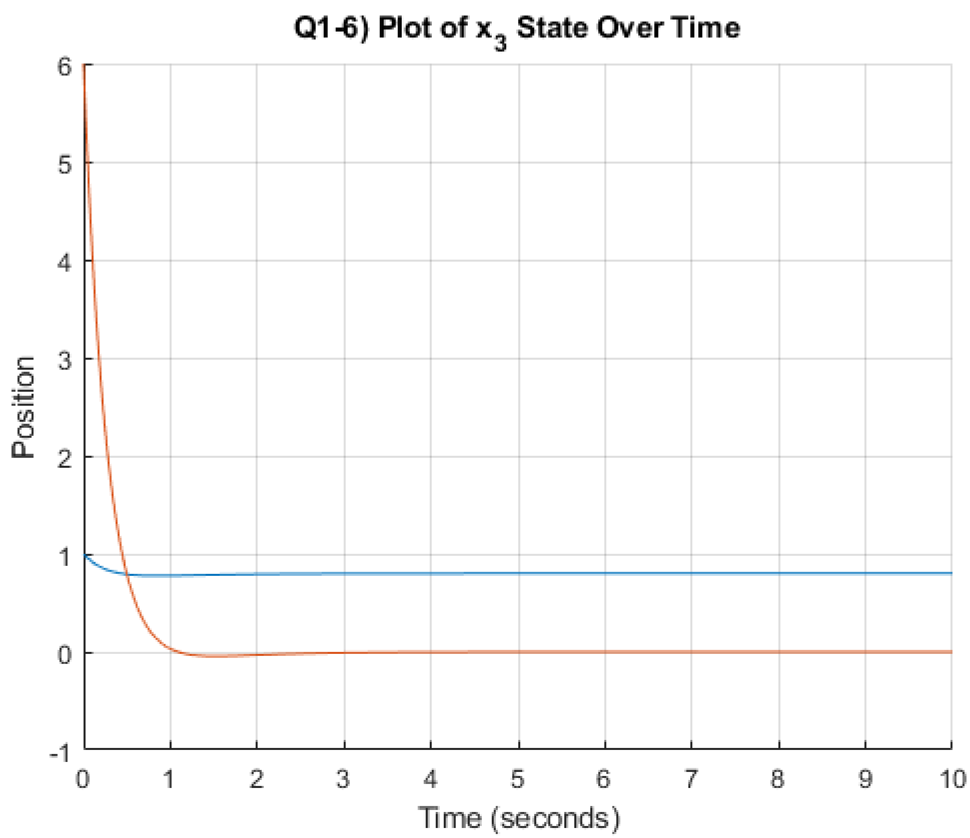

Q1-6) Plot of Each Node's State Over Time
Position
Time (seconds)
8
6
4
2
0
-2
-4
-6
0
1
2
3
4
5
6
7
8
9
10
x1
x2
x3
x4
x5
y1
y2
y3
y4
y5

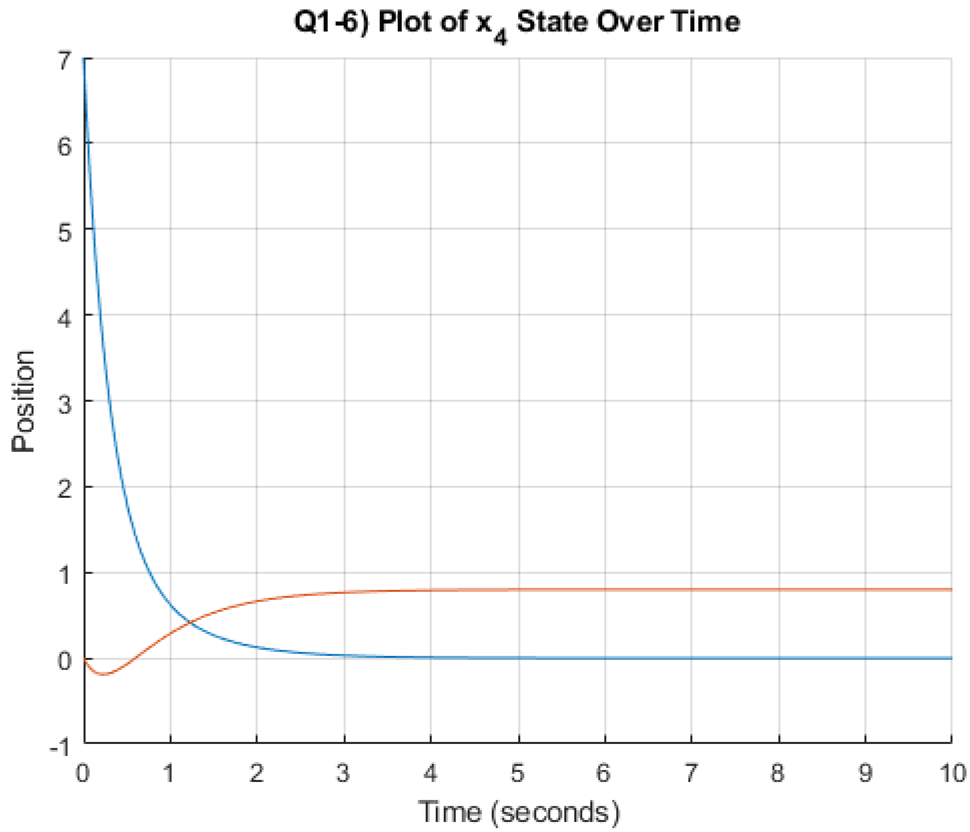

Q1-6) Plot of Each Node's X State Over Time
Position
Time (seconds)
8
6
4
2
0
-2
-4
0
1
2
3
4
5
6
7
8
9
10
x1
x2
x3
x4
x5


Q1-6) Plot of Each Node's Y State Over Time
Position
Time (seconds)
y1
y2
y3
y4
y5
6
4
2
0
-2
-4
-6
0
1
2
3
4
5
6
7
8
9
10

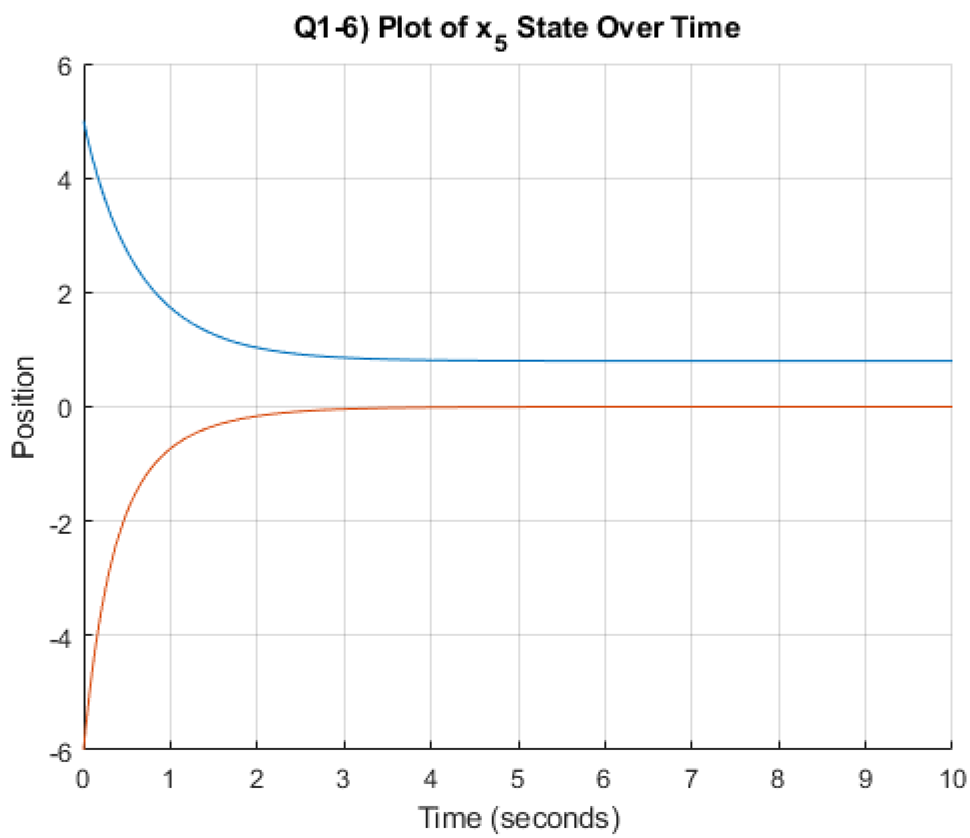

Q1-6) Plot of x1 State Over Time
Position
Time (seconds)
1
0
-1
-2
-3
-4
-5
0
1
2
3
4
5
6
7
8
9
10


Q1-6) Plot of x2 State Over Time
Position
Time (seconds)
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
1
2
3
4
5
6
7
8
9
10

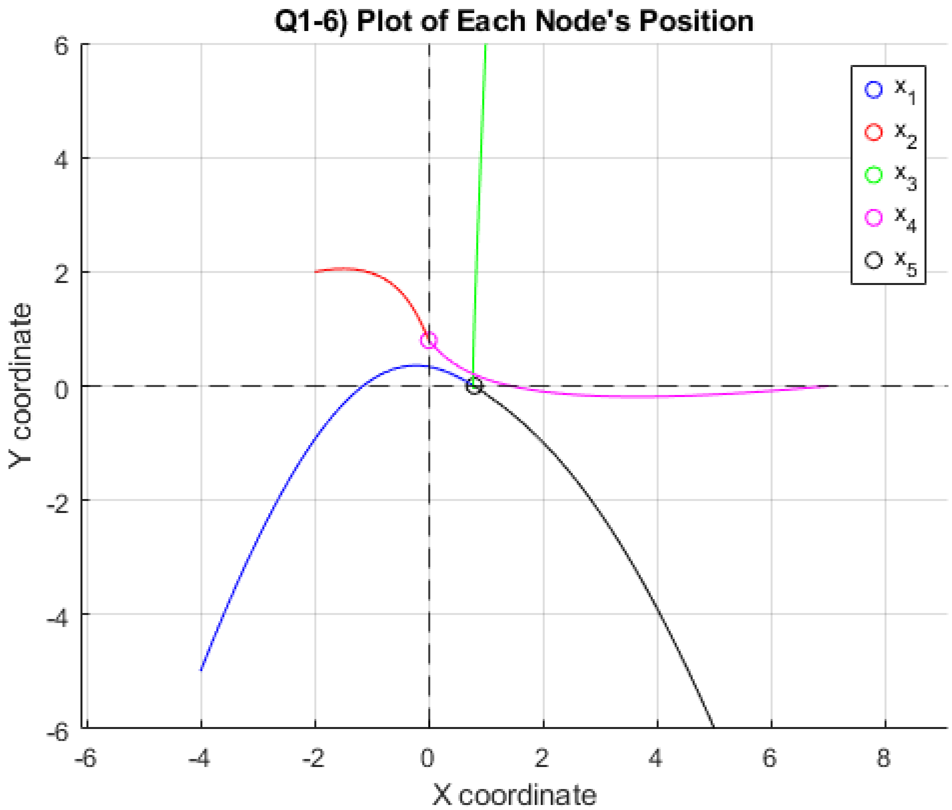

Q1-6) Plot of x3 State Over Time
Position
Time (seconds)
6
5
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9
10



Q1-6) Plot of x4 State Over Time
Position
Time (seconds)
7
6
5
4
3
2
1
0
-1
0
1
2
3
4
5
6
7
8
9
10



Q1-6) Plot of x5 State Over Time
Position
Time (seconds)
6
4
2
0
-2
-4
-6
0
1
2
3
4
5
6
7
8
9
10



Q1-6) Plot of Each Node's Position
x1
x2
x3
x4
x5
Y coordinate
X coordinate
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6
8