EE 362K
Homework 5 Solutions

Q1

5 points

The system is in the reachable canonical form, so it is controllable.

Q2

5 points

- *3 points for observability matrix*
- *2 points for determining observability*

We can find the observability matrix to be

$$\begin{aligned} w_o &= \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 3 \\ 1 & 3 & 0 \\ -1 & -12 & -7 \end{bmatrix} \end{aligned}$$

w_o is full rank, so the system is observable.

Q3

12 points

- *6 points for explanation of what it means to be observable but not controllable or vice versa*
- *6 points for explanation for why the determined coordinate transformation cannot be used to formulate system in CCF*

Controllability refers to whether the input can be used to influence the state values, i.e., whether we can reach any point in the state space with the appropriate input. Observability is a measure of whether the output contains information about the states, i.e., whether we can infer state values by simply observing the output.

For a system to be controllable and not observable, the input should be able to influence all the states but the output should not be carrying information about all the state values. For the case we have an observable but not controllable system, we should be able to infer state values from the output but the input should not influence all the states.

The duality holds in the canonical spaces but we generally do not have the same relationships in the original state space form. Hence, the transformation used to arrive at the OCF cannot be used to arrive at the RCF. This can be seen from the equations for T_o and T_r where $\tilde{w}_o$ and $\tilde{w}_r$ are related but w_o and w_r do not have any such relationship in general.

Q4

27 points

(a)

5 points

- *2 points for approach*
- *1 points for dertermining observability for each C*

Using the formula for the observability matrix, we get that for

- $C = C_1$, $rank(w_o) = 2$
- $C = C_2$, $rank(w_o) = 3$
- $C = C_3$, $rank(w_o) = 2$

So, the system is only observable for $C = C_2$.

(b)

6 points

- *2 points for characteristic equantion*
- *2 points for OCF show steps*
- *1 points for formula for T*
- *1 points for T*

The characteristic polynomial of A is given by $det(\lambda I - A) = \lambda^3 - 4\lambda^2 + 2\lambda + 1$. So, the OCF is

$$\tilde{A} = \begin{bmatrix} 4 & 1 & 0 \\ -2 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}$$

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

$$\begin{aligned} \tilde{w_o} &= \begin{bmatrix} \tilde{C} \\ \tilde{C}\tilde{A} \\ \tilde{C}\tilde{A}^2 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 14 & 4 & 1 \end{bmatrix} \end{aligned}$$

The transformation, T $(z = Tx)$ is given by

$$\begin{aligned} T &= w_o^{-1} * \tilde{w_o} \\ &= \begin{bmatrix} -1 & -\frac{1}{3} & 0 \\ 1 & 0 & 0 \\ -\frac{1}{3} & 0 & -\frac{1}{3} \end{bmatrix} \end{aligned}$$

(c)

4 points

- *2 points for identifying restriction on eigenvalues*
- *2 points for explanation for why the observer is not functional*

Setting $L = 0$ means that the error in the state estimates follow the differential equation, $\frac{d\tilde{z}}{dt} = A\tilde{z}$. The eigenvalues of A are 1, 3.303 and -0.303, These are not non-negative, so the error in the state estimates will not converge to zero. Hence, $L = 0$ will not produce a functional observer.

(d)

12 points

- *1 point for eigenvalues that you want to attain*
- *2 points for CE*
- *2 points for CE in terms of L*
- *3 for correct L*
- *3 points for plot of estimator and true trajectory to verify functionality of observer*
- *1 point for labels*

The choice of eigenvalues that we want $A - LC$ to have is arbitrary but they must have negative real parts. Assuming them to be -3,-4 and -5 for this problem, we get that the desired characteristic equation of $A - LC$ is $p_d(\lambda) = \lambda^3 + 12\lambda^2 + 47\lambda + 60$. Taking $L = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}$, the characteristic equation for $A - LC$ is

$$\begin{aligned} p(\lambda) &= det(\lambda I - (A - LC) \\ &= \lambda^3 + (l_2 - 4)\lambda^2 + (2 - 3l_1 - 3l_2)\lambda - (l_2 + 3l_3 - 1) \end{aligned}$$

Comparing coefficients, we get $l_1 = -31, l_2 = 16, l_3 = -25$. The functionality of this observer is demonstrated by the plot below.

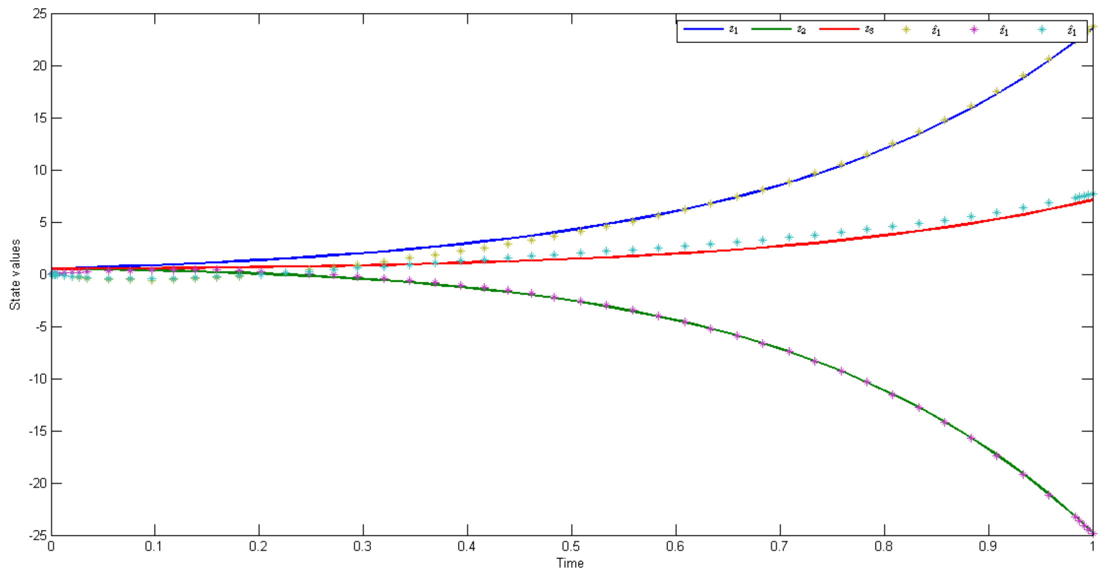

Figure 1: The state estimates converge quickly to the true state values.

```
%% Q4

clear all
close all

global A B C u L

A = [3 0 1; -3 1 0; 1 0 0];
B = [1 0 0]';
u = 1;
%% (a)
% C = [1 0 1];
C = [0 1 0];
% C = [0 0 1];

obs = [C; C*A; C*A^2];
r = rank(obs);

%% (b)
Ac = [4 1 0; -2 0 1; -1 0 0];
Cc = [1 0 0];
obs_c = [Cc; Cc*Ac; Cc*Ac^2];
T = inv(obs)*obs_c;

%% (c)
eigA = eig(A);

```

```
%% (d) verification
L = [-31 16 -25]';
eigobs = eig(A-L*C);

z_init = 0.5*ones(6,1);
[t,z] = ode45(@q4ss, [0 1], z_init);
figure;
z_est = z(:,1:3)-z(:,4:6);
plot(t,z(:,1:3),'Linewidth',1.5);

co = get(gca,'ColorOrder')
set(gca, 'ColorOrder', abs(co-0.75), 'NextPlot', 'replacechildren');
co = get(gca,'ColorOrder')
hold on
plot(t,z_est,'*')
legend({'$z_1$','$z_2$','$z_3$','$\hat{z_1}$','$\hat{z_1}$','$\hat{z_1}$'},'interpreter','latex','
    orientation','horizontal')
xlabel('Time')
ylabel('State values')
```

```
%% function for state space representtion, z and z_err are concatenated
function zprime = q4ss(t,z)
global A B C u L
zprime = zeros(6,1);

zprime(1:3) = A*z(1:3)+B*u; % Evolution of states
zprime(4:6) = (A-L*C)*z(4:6); % Evolution of error in state estimates

end
```

Q5

25 points

- *2 points for stating A,B,C,D*
- *4 points for choice of eigenvalues for observer and controller*
- *5 points for steps to derive K*
- *5 points for steps to derive L*
- *2 points for correct initalization of states and estimates*
- *3 points for formulation of equations governing system with both controller and observer*
- *3 points for graph demonstrating that the system with controller and observer performs as prescribed*
- *1 point for labels*
- *10 additional points for plot showing the convergence of observer to true trajectory*

Reusing the values from HW4, we have that

$$A = \begin{bmatrix} -0.2 & 0.1 \\ 0.5 & -0.5 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.5 \\ 0 \end{bmatrix}$$

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

$$D = 0$$

We can design the controller and observer separately. For the controller, we need to place the eigenvalues of $A - BK$. We find that the characteristic polynomial of $A - BK$ is $p_c(\lambda) = \lambda^2 + (1.5k_1 + 0.7)\lambda + (0.05 + 0.75k_1 + 0.75k_2)$ where $K = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$. Choosing $k_1 = 2, k_2 = 1$ gives us the eigenvalues -.2.91 and -0.79. Since both the eigenvalues are real, negative and distinct, this would be an overdamped system (no overshoot). We also find that $k_r = -\frac{1}{C(A-BK)^{-1}B} = 3.07$. Simulating this system with $y_d = 10$, we observe the following curve for the concentration in Vessel 2.

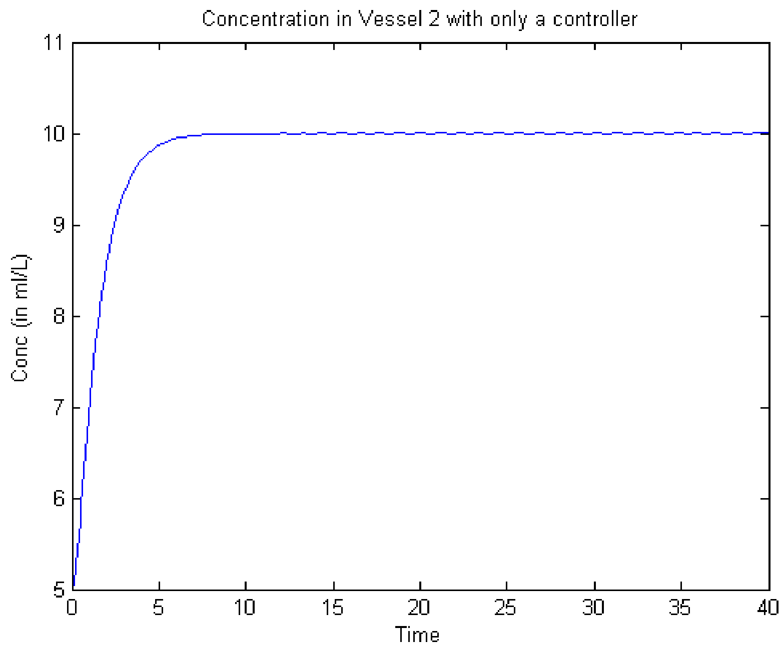

Figure 2: Concentration vs time with only a controller

This satisfies the performance requirements. We thus move onto designing the linear observer. We wish to set the eigenvalues of $A-LC$ to be 5-10 times larger than the eigenvalues in $A-BK$ in magnitude. Let us set them to be 6 times larger than the above eigenvalues, i.e., -17.46 and -4.74. We can find that the characteristic polynomial of $A-LC$ is $p_o(\lambda) = \lambda^2+(l_1+0.7)\lambda+(0.05+0.5l_1+0.1l_2)$ where $L = \begin{bmatrix} l_1 \\ l_2 \end{bmatrix}$. For the desired eigenvalues, we get that $l_1 = 21.5, l_2 = 720$.

When simulating the system with the observer and controller, we are following the equation

$$\frac{d}{dt}\begin{bmatrix} z \\ \tilde{z} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix}\begin{bmatrix} z \\ \tilde{z} \end{bmatrix} + \begin{bmatrix} Bk_r \\ 0 \end{bmatrix} y_d$$

```
%% Q5

clear all
close all

global A B C K kr yd L mode

A  = [-0.2  0.1;  0.5  -0.5];
B = [1.5;  0];
C = [1  0];  % Only concentration in vessel 1 can be measured
yd = 10;     % Desired concentration

%% Designing controller
K = [2  1];
```

```matlab
eigc = eig(A-B*K);
kr = -1/([0 1]*inv(A-B*K)*B);

mode = 1;

[t,z] = ode45(@q5ss, [0 40], [5 5]);
figure(1);
plot(t,z(:,2));
xlabel('Time')
ylabel('Conc (in ml/L)')
title('Concentration in Vessel 2 with only a controller');

%% Designing observer
b = -6*sum(eigc);
l1 = b-0.7;
c = 6^2*prod(eigc);
l2 = (c-0.05-0.5*l1)/0.1;
L = [l1 l2]';
eig(A-L*C);

mode = 2;
[t,z] = ode45(@q5ss, [0 40], [5 5 5 5]);    % Initial error is the same as initial state as state estimates are initialized to 0
figure(2);
plot(t,z(:,1:2));
legend('Vessel 1', 'Vessel 2');
xlabel('Time')
ylabel('Conc (in ml/L)')
title('Plot of concentrations in vessels vs time')

figure(3);
z_est = z(:,1:2)-z(:,3:4);  % State estimate as difference of state and error
plot(t,z(:,1:2),t, z_est);
legend({'$z_1$','$z_2$','$\hat{z_1}$','$\hat{z_2}$'},'interpreter','latex')
xlabel('Time')
ylabel('Conc (in ml/L)')
title('Plot of states and state estimates')
```

```matlab
% function to compute state derivate, the state vector is the state and
% error in state estimates concentenated columnwise
function zprime = q5ss(t,z)

global A B C K kr yd L mode

switch mode
    case 1
        zprime = (A-B*K)*z + B*kr*yd;

    case 2
        zprime = zeros(4,1);
        zprime(1:2) = (A-B*K)*z(1:2) + B*kr*yd;
        zprime(3:4) = (A-L*C)*z(3:4);
end

end
```

The state estimate, $\hat{z} = z - \tilde{z}$. Since the state estimates are initialized to 0, $\tilde{z}(0) = z(0)$. The plots for this system are shown below.

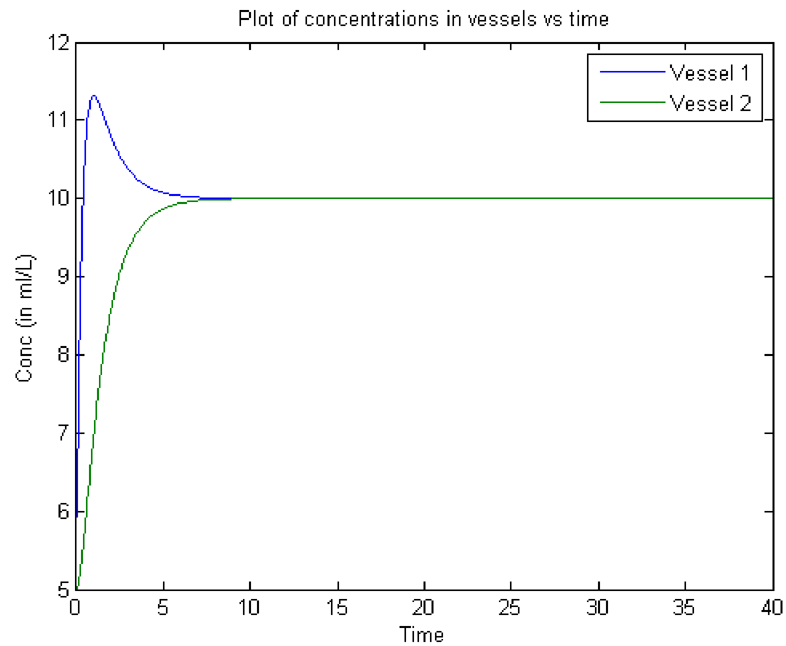

Figure 3: Concentration vs time with controller and observer

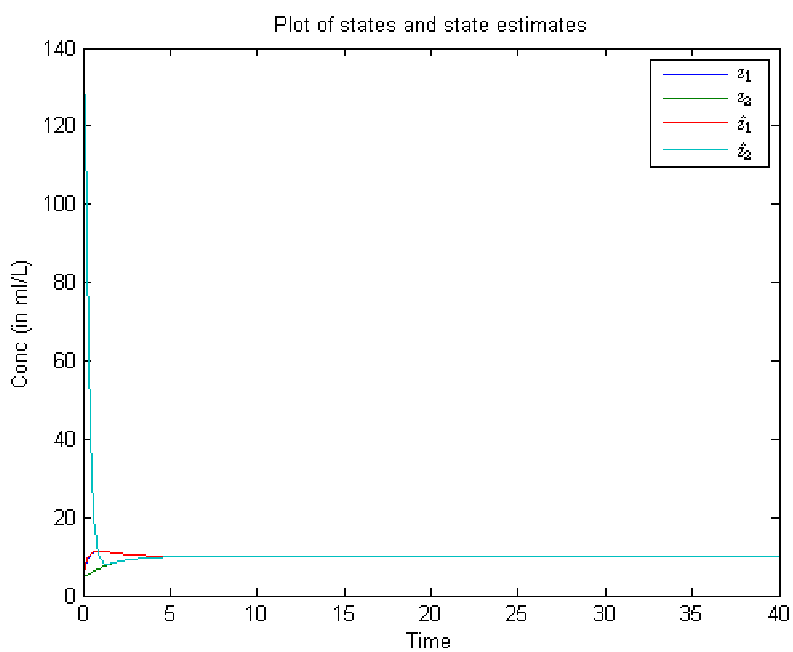

Figure 4: Stae estimates rapidly converge to the true value

Q6

26 points

- *3 points for correct state space representation*
- *4 points for choice of eigenvalues for observer and controller*

- *5 points for steps to derive K*
- *5 points for steps to derive L*
- *3 points for formulation of equations governing system with both controller and observer*
- *2 points for correct initalization of states and estimates*
- *3 points for graphs demonstrating that the system with controller and observer performs as prescribed*
- *1 point for labels*

The normalized state space model of the system gives us the following matrices.

$$A = \begin{bmatrix} 0 & 0 & \omega_0 & 0 \\ 0 & 0 & 0 & \omega_0 \\ -\frac{k}{J_1\omega_0} & \frac{k}{J_1\omega_0} & -\frac{c}{J_1} & \frac{c}{J_1} \\ \frac{k}{J_2\omega_0} & -\frac{k}{J_2\omega_0} & \frac{c}{J_2} & -\frac{c}{J_2} \end{bmatrix}$$
$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{k_i}{J_1\omega_0} \\ 0 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$
$$D = 0$$

Using the Ackerman's solution for designing the controller and observer (see the code), we get $K = \begin{bmatrix} 8.93 & 35.3 & 5.44 & 101 \end{bmatrix}$ and $L = \begin{bmatrix} 9.9 \\ 80.7 \\ 38 \\ 66.8 \end{bmatrix}$.

```matlab
%% Q6
clear all
close all

J1 = 10/9; J2 = 10; c = 0.1; k = 1; ki = 1;
w0 = sqrt( k*(J1+J2)/(J1*J2) );

global A B C D K L kr yd

A = [0 0 w0 0; 0 0 0 w0; -k/(J1*w0) k/(J1*w0) -c/J1 c/J1; k/(J2*w0) -k/(J2*w0) c/J2 -c/J2];
B = [0 0 ki/(J1*w0) 0]';
C = [1 0 0 0];
D = 0;
yd = 1;

%% Designing controller
eigc_des = [-1 -2 -1-1i -1+1i];
polyc_des = poly(eigc_des); % Finding characteristic polynomial for desired eigenvalues
pc = polyc_des(2:end);
% polyA = charpoly(A);
a = [0.1 1.0025 0 0];

% Rechability matrix
Wo = zeros(length(B));
Wo(:,1) = B;
for i=2:size(Wo,2)
```

```matlab
    Wo(:,i) = A*Wo(:,i-1);
end

% Reachable canonical form
Ac = [ -a(1) -a(2) -a(3) -a(4);
1 0 0 0;
0 1 0 0;
0 0 1 0; ];
Bc = [ 1; 0; 0; 0];

% Reachability matrix in canonical space
Wrc = zeros(length(Bc));
Wrc(:,1) = Bc;
for i=2:size(Wrc,2)
    Wrc(:,i) = Ac*Wrc(:,i-1);
end

% Ackerman's solution
Tc = Wrc*inv(Wo);
K = (pc-a)*Tc;
kr = -1/(C*inv(A-B*K)*B);

%% Designing observer
eigo_des = [-4 -2 -2-2i -2+2i];
polyo_des = poly(eigo_des); % Finding characteristic polynomial for desired eigenvalues
po = polyo_des(2:end);

% Observability matrix
Wo = zeros(length(C));
Wo(1,:) = C;
for i=2:size(Wo,1)
    Wo(i,:) = Wo(i-1,:)*A;
end

% Observable canonical form
Aoc = transpose(Ac);
Coc = [1 0 0 0];

% Observability matrix in canonical space
Woc = transpose(Wrc);

To = inv(Wo)*Woc;
L = To*transpose(po-a);

%% Simulating entire system

[t,z] = ode45(@q6ss, [0 10], 0.1*ones(8,1));    % Initial error is the same as initial state as state estimates are initialized to 0
figure(1);
plot(t,C*transpose(z(:,1:4)));
xlabel('Time')
ylabel('Angle')
title('Plot of angle of 1st mass vs time')

figure(2);
z_est = z(:,1:4)-z(:,5:8);  % State estimate as difference of state and error
plot(t,z(:,1:4));
co = get(gca,'ColorOrder');
set(gca, 'ColorOrder', rand(size(co)), 'NextPlot', 'replacechildren');
co = get(gca,'ColorOrder');
hold on
plot(t,z_est,'*')
legend({'$z_1$','$z_2$','$z_3$','$z_4$','$\hat{z_1}$','$\hat{z_2}$','$\hat{z_3}$','$\hat{z_4}$'},'interpreter','latex','orientation','horizontal')
xlabel('Time')
ylabel('Value')
title('Plot of states and state estimates')
```

```matlab
% function to compute state derivate, the state vector is the state and
% error in state estimates concentenated columnwise
function zprime = q6ss(t,z)

```

```
global A B C K kr yd L

zprime = zeros(8,1);
zprime(1:4) = (A-B*K)*z(1:4) + B*kr*yd;
zprime(5:8) = (A-L*C)*z(5:8);

end
```

The plots of the output (angle of first mass) and the performance of the state estimates have been added below.

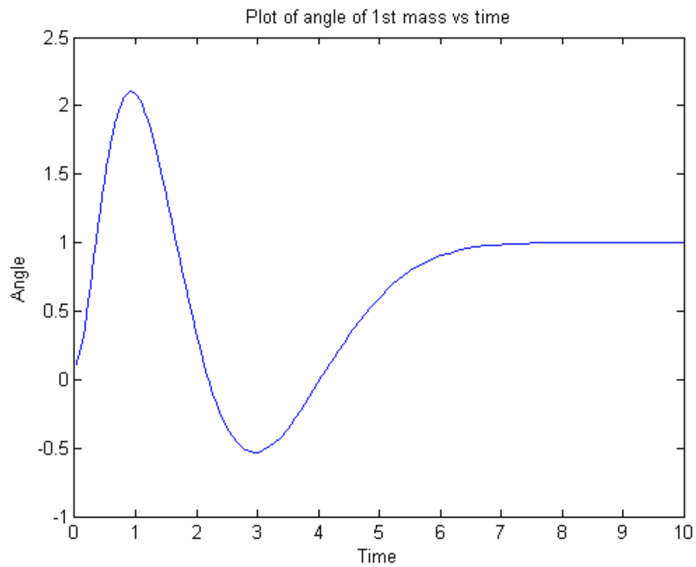

Figure 5: The output converges to the final value.

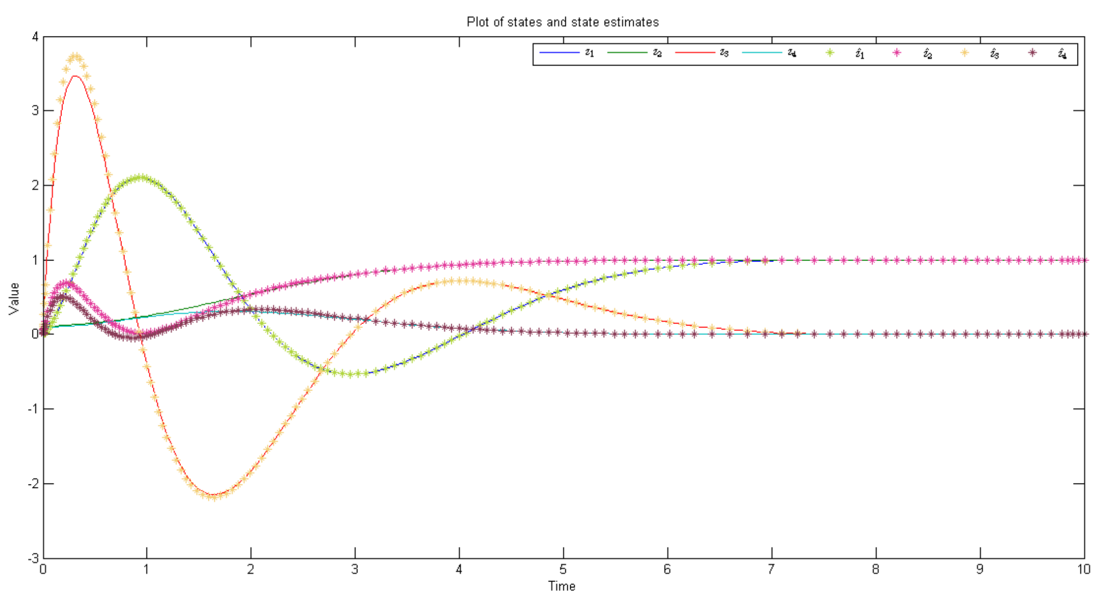

Figure 6: The state estimates converge to their true values despite differences in initialization.