```
clc;clear;close all
```

Problem 1 - State Space Derivation

Formula provided in the project document,

$$\tau_B(t) = K_{\mathrm{PB}}\left(\delta_{\mathrm{AB}}(t) - \overline{\delta}_{\mathrm{AB}}\right)$$
$$\tau_C(t) = K_{\mathrm{PC}}\left(\delta_{\mathrm{BC}}(t) - \overline{\delta}_{\mathrm{BC}}\right) \quad (1)$$

$$\tau_A(t) = -K_{\mathrm{PA}}\left(S_A(t) - \overline{S}_A\right) \quad (2)$$

$$m_A \ddot{S}_A + b_A \dot{S}_A = R\,\tau_A$$
$$m_B \ddot{S}_B + b_B \dot{S}_B = R\,\tau_B \quad (3)$$
$$m_C \ddot{S}_C + b_C \dot{S}_C = R\,\tau_C$$

$$S_A(t) = \overline{S}_A + \widetilde{S}_A(t)$$
$$S_B(t) = \overline{S}_B + \widetilde{S}_B(t) \quad (4)$$
$$S_C(t) = \overline{S}_C + \widetilde{S}_C(t)$$

$$\delta_{\mathrm{AB}}(t) = \overline{\delta}_{\mathrm{AB}} + \widetilde{\delta}_{\mathrm{AB}}(t)$$
$$\delta_{\mathrm{BC}}(t) = \overline{\delta}_{\mathrm{BC}} + \widetilde{\delta}_{\mathrm{BC}}(t) \quad (5)$$

$$\dot{S}_A(t) = \dot{\widetilde{S}}_A(t)$$
$$\dot{S}_B(t) = \dot{\widetilde{S}}_B(t)$$
$$\dot{S}_C(t) = \dot{\widetilde{S}}_B(t)$$
$$\ddot{S}_A(t) = \ddot{\widetilde{S}}_A(t) \quad (6)$$
$$\ddot{S}_B(t) = \ddot{\widetilde{S}}_B(t)$$
$$\ddot{S}_C(t) = \ddot{\widetilde{S}}_C(t)$$

$$m_A \ddot{\widetilde{S}}_A + b_A \dot{\widetilde{S}}_A = R\,\tau_A$$
$$m_B \ddot{\widetilde{S}}_A + b_B \dot{\widetilde{S}}_B = R\,\tau_B \quad (7)$$
$$m_C \ddot{\widetilde{S}}_A + b_C \dot{\widetilde{S}}_B = R\,\tau_C$$

$$\tau_A(t) = -K_{\mathrm{PA}}\left(S_A(t) - \overline{S}_A\right) + \tau_{\mathrm{SFA}}(t)$$
$$\tau_B(t) = K_{\mathrm{PB}}\left(\delta_{\mathrm{AB}}(t) - \overline{\delta}_{\mathrm{AB}}\right) + \tau_{\mathrm{SFB}}(t) \quad (8)$$
$$\tau_C(t) = K_{\mathrm{PC}}\left(\delta_{\mathrm{BC}}(t) - \overline{\delta}_{\mathrm{BC}}\right) + \tau_{\mathrm{SFC}}(t)$$

Subsititute (4) and (5) into (8)

$$\tau_A(t) = -K_{PA}\widetilde{S}_A(t) + \tau_{SFA}(t)$$
$$\tau_B(t) = K_{PB}\widetilde{\delta}_{AB}(t) + \tau_{SFB}(t) \qquad (9)$$
$$\tau_C(t) = K_{PC}\widetilde{\delta}_{BC}(t) + \tau_{SFC}(t)$$

For steady state,

$$S_A(t) = \overline{S}_A$$
$$S_B(t) = \overline{S}_B \qquad (10)$$
$$S_C(t) = \overline{S}_C$$

From the geometry,

$$\delta_{AB}(t) = S_A(t) - S_B(t) - L$$
$$\delta_{BC}(t) = S_B(t) - S_C(t) - L \qquad (11)$$

Because of steady state relationship (11), we have

$$\overline{\delta}_{AB} = \overline{S}_A - \overline{S}_B - L$$
$$\overline{\delta}_{BC} = \overline{S}_B - \overline{S}_C - L \qquad (12)$$

Take the difference between (11) and (12)

$$\widetilde{\delta}_{AB}(t) = \widetilde{S}_A(t) - \widetilde{S}_B(t)$$
$$\widetilde{\delta}_{BC}(t) = \widetilde{S}_B(t) - \widetilde{S}_C(t) \qquad (13)$$

subsitute (9) into (7)

$$m_A\ddot{\widetilde{S}}_A + b_A\dot{\widetilde{S}}_A = -R\,K_{PA}\widetilde{S}_A(t) + R\tau_{SFA}(t)$$
$$m_B\ddot{\widetilde{S}}_A + b_B\dot{\widetilde{S}}_B = \mathrm{RK}_{PB}\widetilde{\delta}_{AB}(t) + R\tau_{SFB}(t) \qquad (14)$$
$$m_C\ddot{\widetilde{S}}_A + b_C\dot{\widetilde{S}}_B = \mathrm{RK}_{PC}\widetilde{\delta}_{BC}(t) + R\tau_{SFC}(t)$$

subsitiute (13) to (14)

$$m_A\ddot{\widetilde{S}}_A + b_A\dot{\widetilde{S}}_A = -R\,K_{PA}\widetilde{S}_A(t) + R\tau_{SFA}(t)$$
$$m_B\ddot{\widetilde{S}}_B + b_B\dot{\widetilde{S}}_B = \mathrm{RK}_{PB}\widetilde{S}_A(t) - \mathrm{RK}_{PB}\widetilde{S}_B(t) + R\tau_{SFB}(t) \qquad (15)$$
$$m_C\ddot{\widetilde{S}}_C + b_C\dot{\widetilde{S}}_C = \mathrm{RK}_{PC}\widetilde{S}_B(t) - \mathrm{RK}_{PC}\widetilde{S}_C(t) + R\tau_{SFC}(t)$$

Finally, reorganize (15)

$$
\begin{aligned}
\ddot{\tilde{S}}_A &= \frac{-R\,K_{\mathrm{PA}}}{m_A}\tilde{S}_A(t) + \frac{-\,b_A}{m_A}\dot{\tilde{S}}_A + 0 + 0 + 0 + 0\; + \frac{R}{m_A}\tau_{\mathrm{SFA}}(t) \\
\ddot{\tilde{S}}_B &= \frac{\mathrm{RK}_{\mathrm{PB}}}{m_B}\tilde{S}_A(t) + 0 + \frac{-\mathrm{RK}_{\mathrm{PB}}}{m_B}\tilde{S}_B(t) + \frac{-b_B}{m_B}\dot{\tilde{S}}_B + 0 + 0 + \frac{R}{m_B}\tau_{\mathrm{SFB}}(t) \quad (16) \\
\ddot{\tilde{S}}_C &= 0 + 0 + \frac{\mathrm{RK}_{\mathrm{PC}}}{m_C}\tilde{S}_B(t) + 0 + \frac{-\mathrm{RK}_{\mathrm{PC}}}{m_C}\tilde{S}_C(t) + \frac{-b_C}{m_C}\dot{\tilde{S}}_C + \frac{R}{m_C}\tau_{\mathrm{SFC}}(t)
\end{aligned}
$$

Problem 2 - Define State Space Parameters

Define the parameters from the project assginment.

```
mA = 2018; %kg
mB = 1907; %kg
mC = 1796; %kg

bA = 100; %N*s/m
bB = 150; %N*s/m
bC = 180; %N*s/m

KPA = 1E3;
KPB = 2E2;
KPC = 2e2;

L = 5; %m
R = 0.358; %m
```

Probelm 3 - Construct State Space Equations

Construct the state equation from (14)

and the output equation from the project assignment.

The state vector is

$$
\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} \widetilde{S}_A \\ \dot{\widetilde{S}}_A \\ \widetilde{S}_B \\ \dot{\widetilde{S}}_B \\ \widetilde{S}_C \\ \dot{\widetilde{S}}_C \end{bmatrix} \qquad \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} \dot{\widetilde{S}}_A \\ \ddot{\widetilde{S}}_A \\ \dot{\widetilde{S}}_B \\ \ddot{\widetilde{S}}_B \\ \dot{\widetilde{S}}_C \\ \ddot{\widetilde{S}}_C \end{bmatrix}
$$

$$
\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \tau_{\mathrm{SFA}} \\ \tau_{\mathrm{SF}B} \\ \tau_{\mathrm{SF}C} \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \widetilde{S}_A \\ \widetilde{\delta}_{\mathrm{AB}} \\ \widetilde{\delta}_{\mathrm{BC}} \end{bmatrix} = \begin{bmatrix} \widetilde{S}_A \\ \widetilde{S}_A - \widetilde{S}_B \\ \widetilde{S}_B - \widetilde{S}_C \end{bmatrix}
$$

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{-R\,K_{\mathrm{P}A}}{m_A} & \frac{-\,b_A}{m_A} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{\mathrm{RK_{PB}}}{m_B} & 0 & \frac{-\mathrm{RK_{PB}}}{m_B} & \frac{-b_B}{m_B} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{\mathrm{RK_{PC}}}{m_C} & 0 & \frac{-\mathrm{RK_{PC}}}{m_C} & \frac{-b_C}{m_C} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ \frac{R}{m_A} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{R}{m_B} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{R}{m_C} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}
$$

$$
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + [0]
$$

```
A = [0 1 0 0 0 0;
```

```
    -R*KPA/mA -bA/mA 0 0 0 0;
    0 0 0 1 0 0;
    R*KPB/mB 0 -R*KPB/mB -bB/mB 0 0;
    0 0 0 0 0 1;
    0 0 R*KPC/mC 0 -R*KPC/mC -bC/mC];

B = [0 0 0;
    R/mA 0 0;
    0 0 0;
    0 R/mB 0;
    0 0 0;
    0 0 R/mC];

C = [1 0 0 0 0 0;
    1 0 -1 0 0 0;
    0 0 1 0 -1 0];

D = 0;

%construct state space model
ol_sys = ss(A,B,C,D)
```

```
ol_sys =

  A = 
             x1        x2        x3        x4        x5        x6
   x1         0         1         0         0         0         0
   x2   -0.1774  -0.04955         0         0         0         0
   x3         0         0         0         1         0         0
   x4   0.03755         0  -0.03755  -0.07866         0         0
   x5         0         0         0         0         0         1
   x6         0         0   0.03987         0  -0.03987   -0.1002

  B = 
               u1          u2          u3
   x1           0           0           0
   x2   0.0001774           0           0
   x3           0           0           0
   x4           0   0.0001877           0
   x5           0           0           0
   x6           0           0   0.0001993

  C = 
       x1  x2  x3  x4  x5  x6
   y1   1   0   0   0   0   0
   y2   1   0  -1   0   0   0
```

```
    y3   0   0   1   0  -1   0

  D = 
       u1  u2  u3
    y1   0   0   0
    y2   0   0   0
    y3   0   0   0

Continuous-time state-space model.
```

```
eig(ol_sys)%check eigen value
```

```
ans = 6×1 complex
  -0.0248 + 0.4205i
  -0.0248 - 0.4205i
  -0.0501 + 0.1933i
  -0.0501 - 0.1933i
  -0.0393 + 0.1897i
  -0.0393 - 0.1897i
```

Problem 4 - Minimum Realization

```
mr_sys = minreal(ol_sys);
size_mr_sys = size(mr_sys)
```

```
size_mr_sys = 1×2
     3     3
```

```
size_ol_sys = size(ol_sys)
```

```
size_ol_sys = 1×2
     3     3
```

The number of states are the same between the two systems.

Therefore, our state space model is the minimum realization.

Problem 5 - Controllability

When the rank of the controllability matrixis 6, the system is controllable, and when the rank is smaller than 6, the system is uncontrollable. Therefore, the whole system is controllable.

```
sys_rank = rank(ctrb(ol_sys))
```

```
sys_rank = 6
```

Controllability for each single input, and store the rank into an array.

```
for c = 1:3
    sub_sys = ol_sys(:,c);
    rank_sub_sys(c) = rank(ctrb(sub_sys))
end
```

```
rank_sub_sys = 6
rank_sub_sys = 1×2
     6     4
rank_sub_sys = 1×3
     6     4     2
```

Controllability for 2 combinations of inputs.

```
sub_sys = ol_sys(:,[1 2]);
rank_sub_sys(4) = rank(ctrb(ol_sys(:,[1 2])))
```

```
rank_sub_sys = 1×4
     6     4     2     6
```

Sub-system with input 1 and 2 is controllable.

```
sub_sys = ol_sys(:,[2 3]);
rank_sub_sys(5) = rank(ctrb(ol_sys(:,[2 3])))
```

```
rank_sub_sys = 1×5
     6     4     2     6     4
```

Sub-system with input 2 and 3 is uncontrollable

```
sub_sys = ol_sys(:,[1 3]);
rank_sub_sys(6) = rank(ctrb(ol_sys(:,[1 3])))
```

```
rank_sub_sys = 1×6
     6     4     2     6     4     6
```

Sub-system with input 1 and 3 is controllable.

Therefore, only the systems that contains the fist input(Tau_SFA) is controllable. This happened becuase only vehicle A has the knowledge to the absolute location.

Problem 6 - Open Loop Poles, Natural Frequency and Damping Ratio

```
Open_Loop_Poles = pole(ol_sys);
[Wn_Hz,Zeta] = damp(ol_sys); % Wn is already in Hz
T = table(Open_Loop_Poles,Wn_Hz,Zeta);
disp(T)
```

Open_Loop_Poles	Wn_Hz	Zeta
-0.024777+0.42046i	0.19377	0.20297
-0.024777-0.42046i	0.19377	0.20297
-0.050111+0.19327i	0.19967	0.25098
-0.050111-0.19327i	0.19967	0.25098
-0.039329+0.18973i	0.42119	0.058826
-0.039329-0.18973i	0.42119	0.058826

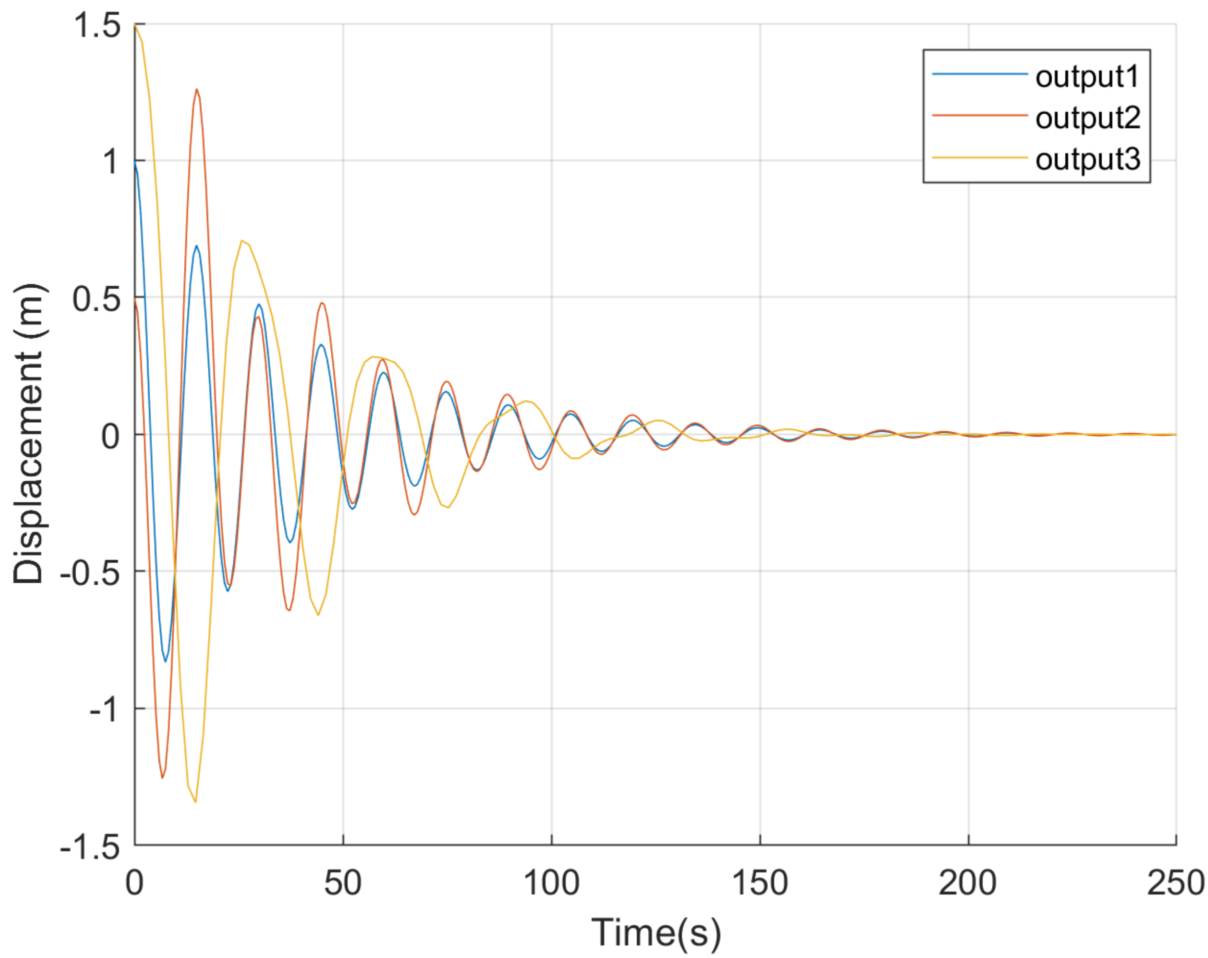
Problem 7 - Initial Condition Reponse of Open Loop System

Initial Condition is

```
x0 = [1;0;0.5;0;-1;0];
```

Plot The Inital condition reponse.

```
figure
for c = 1:3
hold on
[y,t,x] = initial(ol_sys(c,:),x0,250);
plot(t,y)
end
xlim([0,250])
legend('output1','output2','output3')
grid on
hold off
xlabel('Time(s)')
ylabel('Displacement (m)')
set(gca,'fontsize',11) % change font size
```



Problem 8 - Pole placement

Iteration 1

<u>Basic Principle: the more we put the poles on the left half plane, the faster the reponse is, but the bigger control torque is required.</u>

```
% pzmap(ol_sys) %original open loop pole
```

Desired closed loop poles.

```
p1 = -0.0393 + 0.1897i;
p2 = -0.0393 - 0.1897i;

p3 = -0.0501 + 0.1933i;
p4 = -0.0501 - 0.1933i;

% place the original dominant pole away from the imaginary axis
p5 =  -0.06 + 0.4205i;
p6 =  -0.06 - 0.4205i;
```

Construct Close loop system and use initial condition reponse.

```
G1 = place(A,B,[p1 p2 p3 p4 p5 p6]);

cl_sys1 = ss(A-B*G1,[],C - D * G1,D);

[y_cl1,t_cl1,x_cl1] = initial(cl_sys1,x0);
S1 = lsiminfo(y_cl1,t_cl1,'SettlingTimeThreshold',0.05);
```

Compute maximum sellting time.

```
for c = 1:3
    SettlingTimeArray1(c) = S1(c).SettlingTime;
end
maxSettlingTime1 = max(SettlingTimeArray1);
disp(['maxSettlingTime1 = ', num2str(maxSettlingTime1),' s'])
```

```
maxSettlingTime1 = 68.2618 s
```

Settling time becomes much better than the open loop plant.

Compute maximum control torque.

```
[m,n] = size(x_cl1);
for c = 1 : m
u1(:,c) = - G1 * x_cl1(c,:)';
end

%compute max input - control torque
umax1 = max(abs(u1(:)));
disp(['umax1 =', num2str(umax1),' N*m'])
```

```
umax1 =768.3886 N*m
```

Control torque is under-untilized.

Iteration 2

Make the real part of the real poles slightly bigger.

```
p1 = -0.1 + 0.2i;
p2 = -0.1 - 0.2i;

p3 = -0.2 + 0.2i;
p4 = -0.2 - 0.2i;
```

```
p5 =  -0.4 + 0.4i;
p6 =  -0.4 - 0.4i;
```

Construct Close loop system and use initial condition reponse.

```
G2 = place(A,B,[p1 p2 p3 p4 p5 p6]);

cl_sys2 = ss(A-B*G2,[],C - D * G2,[]);

[y_cl2,t_cl2,x_cl2] = initial(cl_sys2,x0);
S2 = lsiminfo(y_cl2,t_cl2,'SettlingTimeThreshold',0.05);
```

Compute maximum sellting time.

```
c = 1;
for c = 1:3
    SettlingTimeArray2(c) = S2(c).SettlingTime;
end

%find max settling time
maxSettlingTime2 = max(SettlingTimeArray2);
disp(['maxSettlingTime2 = ', num2str(maxSettlingTime2),' s'])
```

```
maxSettlingTime2 = 27.3797 s
```

The sellting time improves but is still too high.

Compute maximum control torque.

```
[m,n] = size(x_cl2);

c = 1;
for c = 1 : m
u2(:,c) = - G2 * x_cl2(c,:)';
end
%compute max input - control torque
umax2 = max(abs(u2(:)))
```

```
umax2 = 952.2905
```

```
disp(['umax2 =', num2str(umax2),' N*m'])
```

```
umax2 =952.2905 N*m
```

The control torque is still low, so more control torque can be use.

Iteration 3

double the real part and add damping.

```
p1 = -0.2 + 0i;
p2 = -0.2 - 0i;

p3 = -0.4 + 0.1i;
p4 = -0.4 - 0.1i;

p5 =  -0.8 + 0.2i;
p6 =  -0.8 - 0.2i;
```

Construct Close loop system and use initial condition reponse.

```
G3 = place(A,B,[p1 p2 p3 p4 p5 p6]);

cl_sys3 = ss(A-B*G3,[],C - D * G3,[]);

[y_cl3,t_cl3,x_cl3] = initial(cl_sys3,x0);
S3 = lsiminfo(y_cl3,t_cl3,'SettlingTimeThreshold',0.05);
```

Compute maximum sellting time.

```
c = 1;
for c = 1:3
    SettlingTimeArray3(c) = S3(c).SettlingTime;
end

%find max settling time
maxSettlingTime3 = max(SettlingTimeArray3);
disp(['maxSettlingTime3 = ', num2str(maxSettlingTime3),' s'])
```

```
maxSettlingTime3 = 16.2492 s
```

The settling time becomes closed to the goal.

Compute maximum control torque.

```
[m,n] = size(x_cl3);

c = 1;
for c = 1 : m
u3(:,c) = - G3 * x_cl3(c,:)';
end
```

```
%compute max input - control torque
umax3 = max(abs(u3(:)));
disp(['umax3 =', num2str(umax3),' N*m'])
```

```
umax3 =766.9778 N*m
```

Control torques slightly increased, but still under-utilized.

Iteration 4

Double the real part again.

```
p1 = -0.4 + 0i;
p2 = -0.5 - 0i;

p3 = -0.6 + 0.1i;
p4 = -0.6 - 0.1i;

p5 = -0.8 + 0.2i;
p6 = -0.8 - 0.2i;
```

Construct Close loop system and use initial condition reponse.

```
G4 = place(A,B,[p1 p2 p3 p4 p5 p6]);

cl_sys4 = ss(A-B*G4,[],C - D * G4,[]);

[y_cl4,t_cl4,x_cl4] = initial(cl_sys4,x0);
S4 = lsiminfo(y_cl4,t_cl4,'SettlingTimeThreshold',0.05);
```

Compute maximum sellting time.

```
c = 1;
for c = 1:3
    SettlingTimeArray4(c) = S4(c).SettlingTime;
end

%find max settling time
maxSettlingTime4 = max(SettlingTimeArray4) ;
disp(['maxSettlingTime4 = ', num2str(maxSettlingTime4),' s'])
```

```
maxSettlingTime4 = 9.3875 s
```

The settling time is finally within the design requirement.

Compute maximum control torque.

```
[m,n] = size(x_cl4);

c = 1;
for c = 1 : m
u4(:,c) = - G4 * x_cl4(c,:)';
end
%compute max input - control torque
umax4 = max(abs(u4(:)));
disp(['umax4 =', num2str(umax4),' N*m'])
```

```
umax4 =1747.7017 N*m
```

The control torque is also within the limit

Iteration 5

Slightly move the poles to the left to fully utilize the control torque we have

```
p1 = -0.6 + 0i;
p2 = -0.7 - 0i;

p3 = -0.6 + 0.1i;
p4 = -0.6 - 0.1i;

p5 =  -0.7 + 0.1i;
p6 =  -0.7 - 0.1i;
```

Construct Close loop system and use initial condition reponse.

```
G5 = place(A,B,[p1 p2 p3 p4 p5 p6]);

cl_sys5 = ss(A-B*G5,[],C - D * G5,[]);

[y_cl5,t_cl5,x_cl5] = initial(cl_sys5,x0);
S5 = lsiminfo(y_cl5,t_cl5,'SettlingTimeThreshold',0.05);
```

Compute maximum sellting time.

```
c = 1;
for c = 1:3
    SettlingTimeArray5(c) = S5(c).SettlingTime;
end

%find max settling time
```

```
maxSettlingTime5 = max(SettlingTimeArray5) ;
disp(['maxSettlingTime5 = ', num2str(maxSettlingTime5),' s'])
```

```
maxSettlingTime5 = 7.2614 s
```

The settling time is further reduced.

Compute maximum control torque.

```
[m,n] = size(x_cl5);

c = 1;
for c = 1 : m
u5(:,c) = - G5 * x_cl5(c,:)';
end
%compute max input - control torque
umax5 = max(abs(u5(:)));
disp(['umax5 =', num2str(umax5),' N*m'])
```

```
umax5 =1801.5727 N*m
```

The control torque is almost fully utilized and is within 2100 N*m.

Final Design Inputs and Output Plots

```
figure
```

Generate the output plots

```
subplot(2,1,1)
plot(t_cl5,y_cl5)
xlim([0 12])
legend('output1','output2','output3')
grid on
xlabel('Time(s)')
ylabel('Displacement (m)')
set(gca,'fontsize',11) % change font size
```

Generate the input plots

```
subplot(2,1,2)
plot(t_cl5,u5)
xlim([0 12])
legend('input1','input2','input3')
grid on
hold off
```

```
xlabel('Time(s)')
ylabel('Torque(N*m)')
set(gca,'fontsize',11) % change font size
```