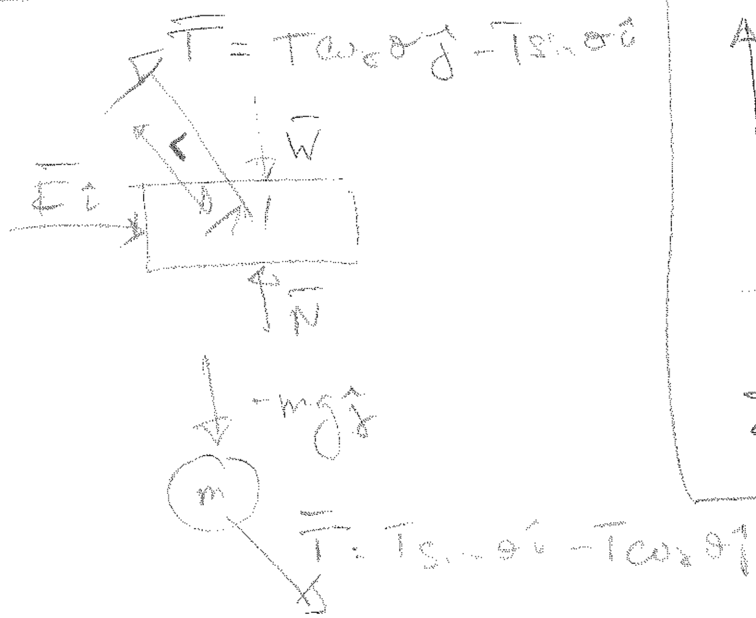
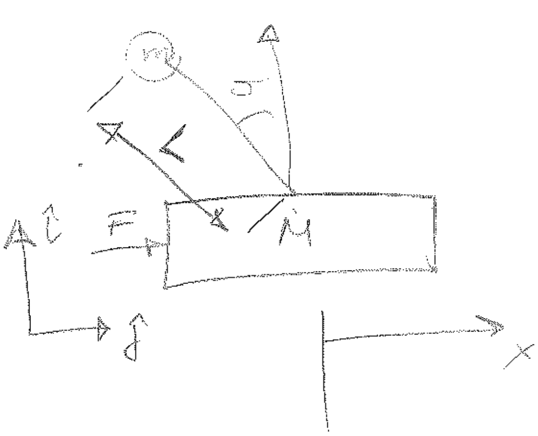
Project Part I

inverted pendulum





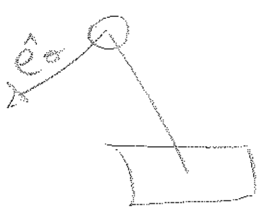
Cart:

$$\sum F_{\hat{i}} : F - T\sin\theta = M\ddot{X}_M(t)$$

Ball:

$$\sum F_{\hat{j}} : -T\cos\theta - mg = m\ddot{y}_m(t)$$

$$\sum F_{\hat{i}} : T\sin\theta = m\ddot{X}_m(t)$$



acceleration of the ball:

$$\bar{a}_p = \ddot{X}_M(t)\,\hat{i} + \left[L\ddot{\theta}\,\hat{e}_\theta - L\dot{\theta}^2\,\hat{e}_r\right]$$

$$= \ddot{X}_M(t)\,\hat{i} + L\ddot{\theta}\left[-\cos\theta\,\hat{i} - \sin\theta\,\hat{j}\right] - L\dot{\theta}^2\left[-\sin\theta\,\hat{i} + \cos\theta\,\hat{j}\right]$$

$\rightarrow T\sin\theta = m\ddot{X}_M - mL\ddot{\theta}\cos\theta + mL\dot{\theta}^2\sin\theta \leftarrow \hat{i}$ coordinate

$T\cos\theta + mg = mL\ddot{\theta}\sin\theta + mL\dot{\theta}^2\cos\theta \leftarrow \hat{j}$ coordinate

$$T\cos\theta\sin\theta = m\ddot{X}_M\cos\theta - mL\ddot{\theta}\cos^2\theta + mL\dot{\theta}^2\sin\theta\cos\theta$$

$$T\cos\theta\sin\theta + (\sin\theta)mg = mL\ddot{\theta}\sin^2\theta + mL\dot{\theta}^2\sin\theta\cos\theta$$

$$\rightarrow \underline{L\ddot{\theta} = g\sin\theta + \ddot{X}_M\cos\theta}$$

$$\ddot{\theta}L - g\sin\theta + \ddot{X}_M\cos\theta = 0$$

$$\ddot{\theta}L = g\sin\theta - \ddot{X}_M\cos\theta$$

$$\Rightarrow (M+m)\ddot{X}_M + m\left(g\sin\theta - \ddot{X}\cos\theta\right)\cos\theta + m\dot{\theta}^2\sin\theta = f(t)$$

$$(M+m)\ddot{X}_M + mg\sin\theta\cos\theta - m\ddot{X}_M\cos^2\theta + m\dot{\theta}^2\sin\theta = f(t)$$

$$(M+m-m\cos^2\theta)\ddot{X}_M + mg\sin\theta\cos\theta + m\dot{\theta}^2\sin\theta = f(t)$$

$$(M+m\sin^2\theta)\ddot{X}_M + mg\sin\theta\cos\theta + m\dot{\theta}^2\sin\theta = f(t)$$

$$\ddot{X}_M(t) = \frac{-mg\sin\theta\cos\theta - m\dot{\theta}^2\sin\theta + f(t)}{M+m\sin^2\theta}$$

$$\ddot{\theta} = \frac{g\sin\theta}{L} + \cos\theta\left(\frac{mg\sin\theta\cos\theta + m\dot{\theta}^2\sin\theta + f(t)}{M_M + m\sin^2\theta}\right)$$

Need to use small angle approximations.

Single input

$$\begin{cases} L\ddot{\theta} - g\sin\theta + \ddot{X}_M\cos\theta = 0 \\ (M+m)\ddot{X}_M + mL\ddot{\theta}\cos\theta + m\dot{\theta}^2\sin\theta = f(t) \end{cases}$$

Double input

$$\begin{cases} L\ddot{\theta} - g\sin\theta + \ddot{x}_M\cos\theta = \tau(t) \\ (M+m)\ddot{x}_M + mL\ddot{\theta}\cos\theta + m\dot{\theta}^2\sin\theta = f(t) \end{cases}$$

Let $x_1(t) = x$; $x_2(t) = \dot{x}$; $x_3(t) = \theta$; $x_4(t) = \dot{\theta}$.

$$L\dot{x}_4(t) - g\sin(x_3(t)) - \dot{x}_2(t)\cos(x_3(t)) = u_2(t)$$

$$(M+m)\dot{x}_2(t) + mL\dot{x}_4(t)\cos(x_3(t)) - m x_4^2(t)\sin(x_3(t)) = u_1(t)$$

Project part 2 a_i(a):

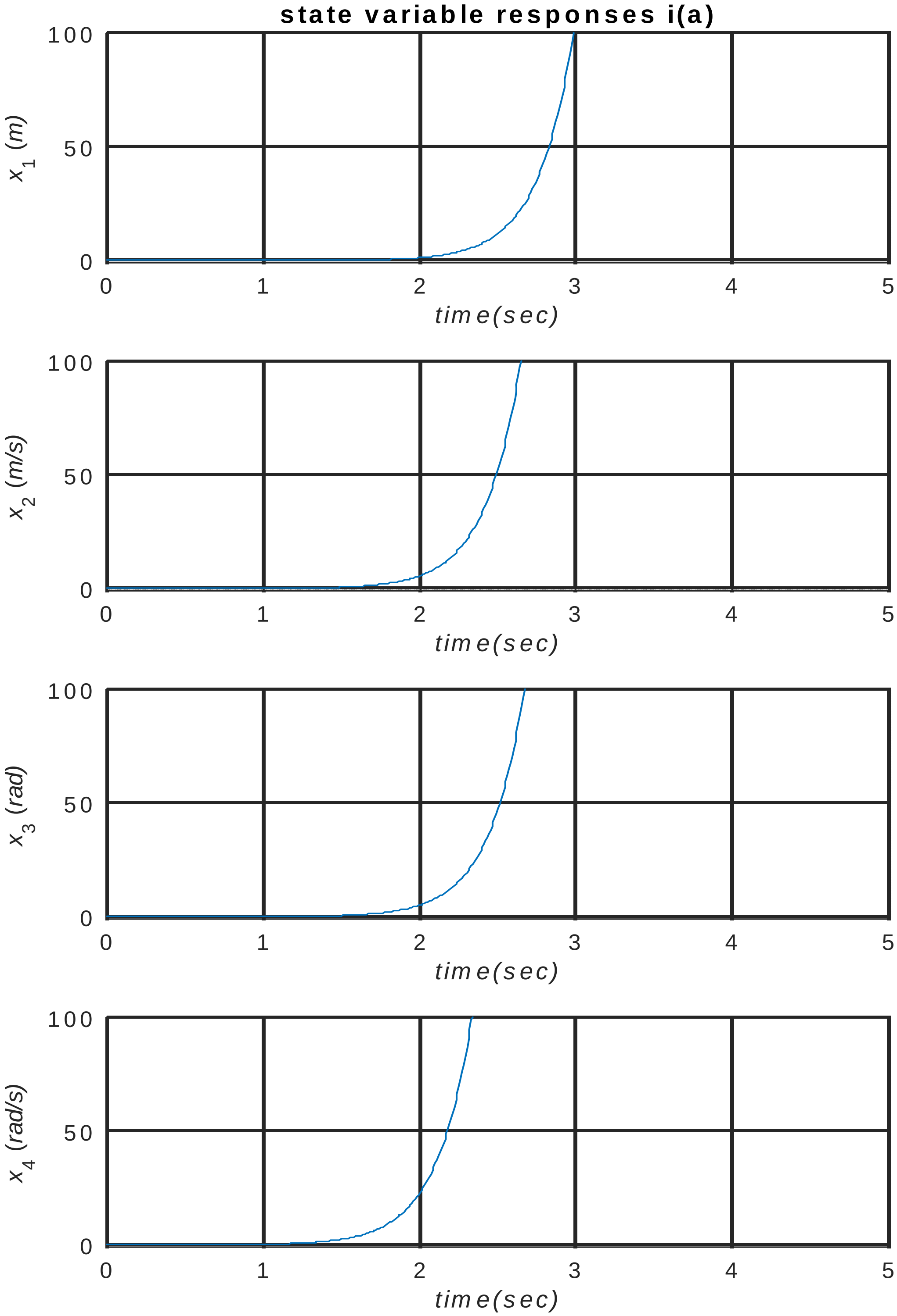
unit impulse input f(t) and zero initial state.

```
%Yevgen Solodkyy
%CE2.2a part i.a:

clear all;
clc;
m1=2; m2=1; L=.75; g=9.81;

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0];
EigenValues = eig(A)

D = 0;

t=[0:.01:7];

x0 = [0;0;0;0];

impulse = t==0;

B = [0; 1/m1; 0; 1/(L*m1)]; % need to verify what this matrix looks
like.
C = [0 0 1 0];

u=[impulse]; %zeros(size(t)) is t x 1 dimentions.

sys_ia = ss(A,B,C,D)% create a state space system based on the system
matrices

[Y0,t,X0] = lsim(sys_ia,u,t,x0); % simulate the system output solution
Y0 & system state solution X0.

% I have 4 state variables and one system solution. So I should have 5
% plots all together for each case.

%PLOTTING PORTION

figure(1);
set(gcf,'numbertitle','on','name','state variable responses i(a)') %
figure title
subplot(411),
plot(t,X0(:,1)); grid; %plot the first state variable
title('state variable responses i(a)')
axis([0 5 -1 100]);
set(gca, 'FontSize',12);
xlabel('\ittime(sec)'); ylabel('{\itx}_1 (\itm)'); % might want to
remove the x-lalel here

subplot(412),
```

```
plot(t,X0(:,2)); grid;  %plot the second state variable
axis([0 5 -1 100]);
set(gca,'FontSize',12);
xlabel('\ittime(sec)'); ylabel('{\itx}_2 (\itm/s)');

subplot(413),
plot(t,X0(:,3)); grid; %plot the third state variable
axis([0 5 -1 100]);
set(gca, 'FontSize',12);
xlabel('\ittime(sec)'); ylabel('{\itx}_3 (\itrad)'); % might want to
remove the x-lalel here

subplot(414),
plot(t,X0(:,4)); grid; %plot the second variable
axis([0 5 -1 100]);
set(gca, 'FontSize',12);
xlabel('\ittime(sec)'); ylabel('{\itx}_4 (\itrad/s)'); % might want to
remove the x-lalel here
```

Command Output:

```
sys_ia =

  A =
           x1      x2      x3      x4
   x1       0       1       0       0
   x2       0       0   4.905       0
   x3       0       0       0       1
   x4       0       0   19.62       0

  B =
           u1
   x1       0
   x2     0.5
   x3       0
   x4  0.6667

  C =
       x1  x2  x3  x4
   y1   0   0   1   0

  D =
       u1
   y1   0

Continuous-time state-space model.

>>
```


state variable responses i(a)
100
50
0
x_1 (m)
0
1
2
3
4
5
time(sec)
100
50
0
x_2 (m/s)
0
1
2
3
4
5
time(sec)
100
50
0
x_3 (rad)
0
1
2
3
4
5
time(sec)
100
50
0
x_4 (rad/s)
0
1
2
3
4
5
time(sec)

Project Part 2 a_i(b):
zero iput f(t) and initial conditions theta(0)= 0.1 rad

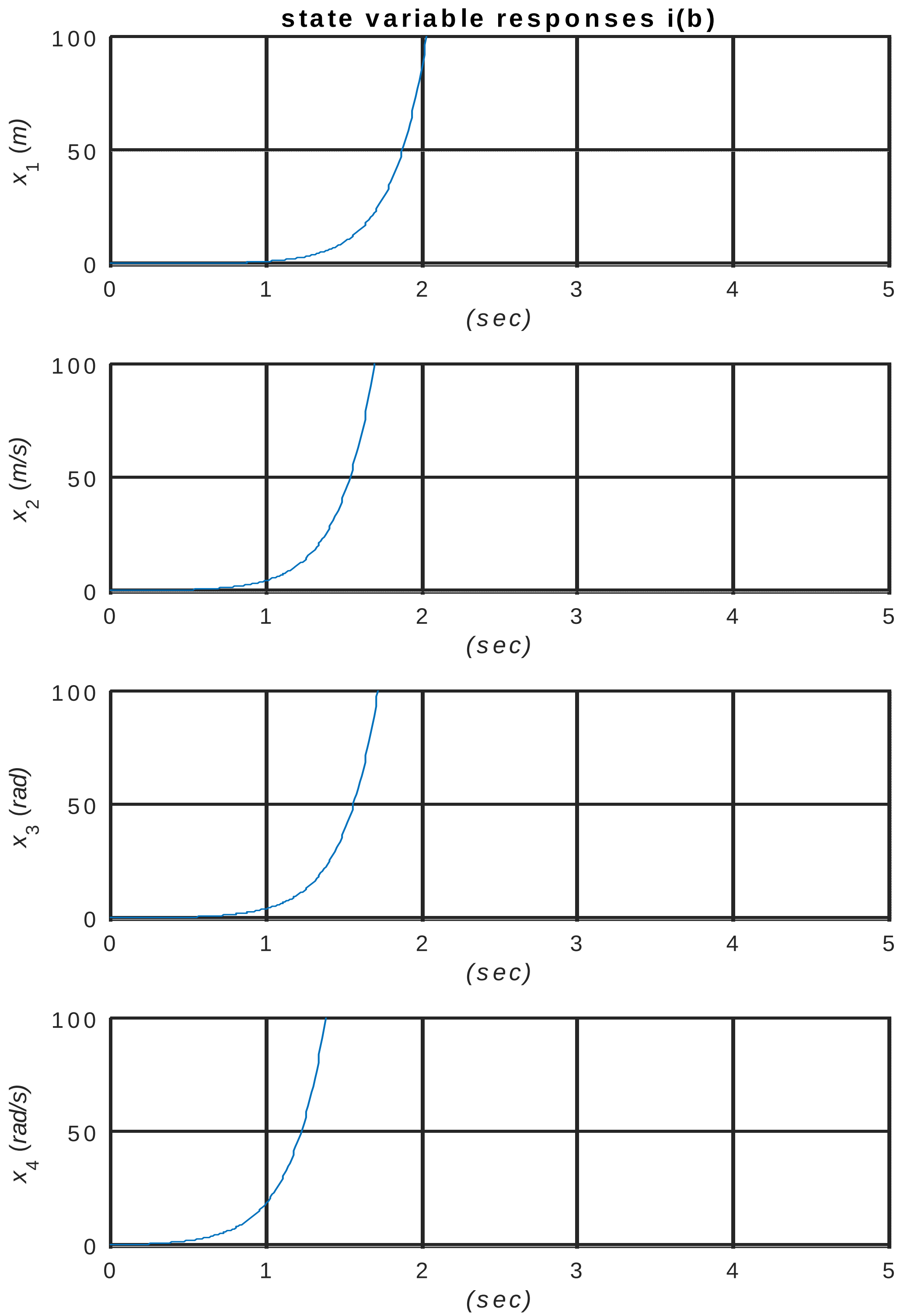
Code:

```
%CE2.2a, part i.b.

clear all;
clc;
m1=2; m2=1; L=.75; g=9.81;

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0];
EigenValues = eig(A)

D = 0;

t=[0:.01:7];

impulse = t==0;

B = [0; 1/m1; 0; 1/(L*m1)];
C = [0 0 1 0];
x0_ib = [0;0;.1;0];
u_ib=[zeros(size(t))];

sys_ib = ss(A,B,C,D)

[Y0_ib,t,X0_ib] = lsim(sys_ib,u_ib,t,x0_ib);

%PLOTTING PORTION

figure(2);
set(gcf,'numbertitle','off','name','state variable responses i(b)') %
figure title
subplot(411),
plot(t,X0_ib(:,1)); grid; %plot the first state variable
title('state variable responses i(b)')
axis([0 5 -1 100]);
set(gca, 'FontSize',12);
xlabel('\it(sec)'); ylabel('{\itx}_1 (\itm)'); % might want to remove
the x-lalel here

subplot(412), plot(t,X0_ib(:,2)); grid;  %plot the second state variable
axis([0 5 -1 100]);
set(gca,'FontSize',12);
xlabel('\it(sec)'); ylabel('{\itx}_2 (\itm/s)');

subplot(413), plot(t,X0_ib(:,3)); grid; %plot the third state variable
axis([0 5 -1 100]);
set(gca, 'FontSize',12);
```

```
xlabel('\it(sec)'); ylabel('{\itx}_3 (\itrad)'); % might want to remove
the x-lalel here

subplot(414), plot(t,X0_ib(:,4)); grid; %plot the second variable
axis([0 5 -1 100]);
set(gca, 'FontSize',12);
xlabel('\it(sec)'); ylabel('{\itx}_4 (\itrad/s)'); % might want to
remove the x-lalel here
```

Command output:

```
sys_ib =

  A =
           x1      x2      x3      x4
   x1       0       1       0       0
   x2       0       0   4.905       0
   x3       0       0       0       1
   x4       0       0   19.62       0

  B =
            u1
   x1        0
   x2      0.5
   x3        0
   x4   0.6667

  C =
       x1  x2  x3  x4
   y1   0   0   1   0

  D =
       u1
   y1   0

Continuous-time state-space model.

>>
```



state variable responses i(b)
x_1 (m)
x_2 (m/s)
x_3 (rad)
x_4 (rad/s)
100
50
0
0
1
2
3
4
5
(sec)

Project Part 2 a_ii:
Single iput, multiple output: impulse input f(t) & two outputs

Code:

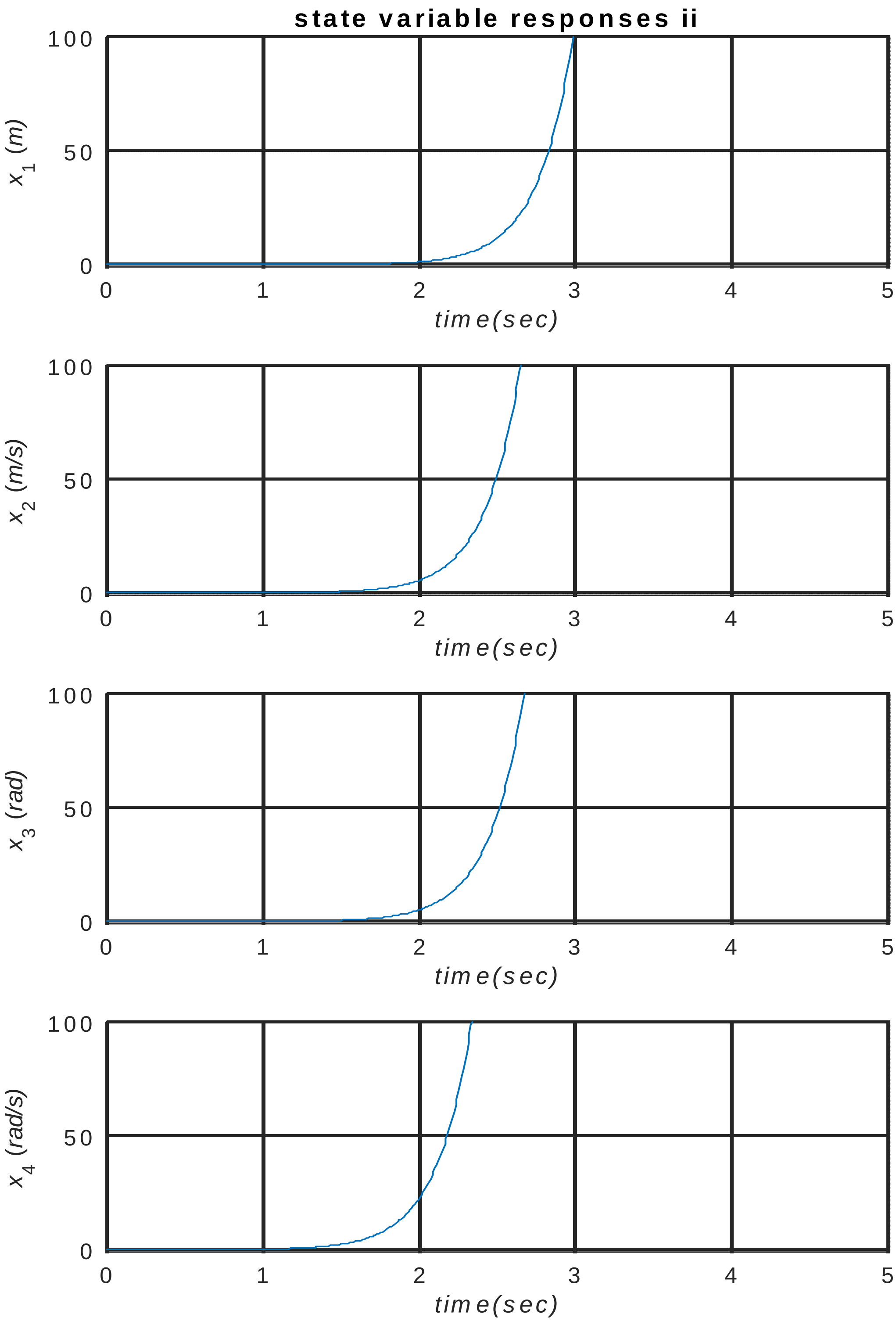
```
%Yevgen Solodkyy
%CE2.2a, part ii

clear all;
clc;
m1=2; m2=1; L=.75; g=9.81;

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0];

D = 0;

t=[0:.01:7];

x0 = [0;0;0;0];

impulse = t==0;



B = [0; 1/m1; 0; 1/(L*m1)]; % need to verify what this matrix looks
like.
C = [1 0 0 0;0 0 1 0]; % two outputs

u=[impulse];

sys_ii = ss(A,B,C,D)

[Y0,t,X0] = lsim(sys_ii,u,t,x0);

%PLOTTING PORTION

figure(3);
set(gcf,'numbertitle','off','name','state variable responses ii') %
figure title
subplot(411),
plot(t,X0(:,1)); grid; %plot the first state variable
title('state variable responses ii')
axis([0 5 -1 100]);
set(gca, 'FontSize',12);
xlabel('\ittime(sec)'); ylabel('{\itx}_1 (\itm)'); % might want to
remove the x-lalel here

subplot(412), plot(t,X0(:,2)); grid;  %plot the second state variable
```

```
axis([0 5 -1 100]);
set(gca,'FontSize',12);
xlabel('\ittime(sec)'); ylabel('{\itx}_2 (\itm/s)');

subplot(413), plot(t,X0(:,3)); grid; %plot the third state variable
axis([0 5 -1 100]);
set(gca, 'FontSize',12);
xlabel('\ittime(sec)'); ylabel('{\itx}_3 (\itrad)'); % might want to
remove the x-lalel here

subplot(414), plot(t,X0(:,4)); grid; %plot the second variable
axis([0 5 -1 100]);
set(gca, 'FontSize',12);
xlabel('\ittime(sec)'); ylabel('{\itx}_4 (\itrad/s)'); % might want to
remove the x-lalel here
```

Command Output:

```
sys_ii =

  A =
           x1      x2      x3      x4
   x1       0       1       0       0
   x2       0       0   4.905       0
   x3       0       0       0       1
   x4       0       0   19.62       0

  B =
            u1
   x1        0
   x2      0.5
   x3        0
   x4   0.6667

  C =
        x1  x2  x3  x4
   y1    1   0   0   0
   y2    0   0   1   0

  D =
        u1
   y1    0
   y2    0

Continuous-time state-space model.
```

part a_ii plots:



state variable responses ii
100
50
0
0
1
2
3
4
5
time(sec)
x_1 (m)
x_2 (m/s)
x_3 (rad)
x_4 (rad/s)

Project Part 2_iii:

Multiple-input multiple-output :
two unit step inputs & two outputs.

Code:

```
%CE2.2a, part III

clear all;
clc;
m1=2; m2=1; L=.75; g=9.81;

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0];

D = 0;

t=[0:.01:3600];

x0 = [0;0;0;0];

%impulse = t==0;

B = [0 0; 1/m1 1/(L*m1); 0 0; 1/m1*L (m2+m1)/(m1*m2*L^2)] % need to
verify what this matrix looks like.
C = [1 0 0 0;0 0 1 0]; % two outputs

u=[ones(size(t)); ones(size(t))];%[zeros(size(t)); zeros(size(t))];

sys_iii = ss(A,B,C,D)



[Y0,t,X0] = lsim(sys_iii,u,t,x0);



%PLOTTING PORTION%{
figure(4);
set(gcf,'numbertitle','off','name','state variable responses III') %
figure title
subplot(411),
plot(t,X0(:,1)); grid; %plot the first state variable
title('state variable responses iii')
axis([0 5 -1 100]);
set(gca, 'FontSize',12);
xlabel('\ittime(sec)'); ylabel('{\itx}_1 (\itm)'); % might want to
remove the x-lalel here

subplot(412), plot(t,X0(:,2)); grid;  %plot the second state variable
```

```
axis([0 5 -1 100]);
set(gca,'FontSize',12);
xlabel('\ittime(sec)'); ylabel('{\itx}_2 (\itm/s)');

subplot(413), plot(t,X0(:,3)); grid; %plot the third state variable
axis([0 5 -1 100]);
set(gca, 'FontSize',12);
xlabel('\ittime(sec)'); ylabel('{\itx}_3 (\itrad)'); % might want to
remove the x-lalel here

subplot(414), plot(t,X0(:,4)); grid; %plot the second variable
axis([0 5 -1 100]);
set(gca, 'FontSize',12);
xlabel('\ittime(sec)'); ylabel('{\itx}_4 (\itrad/s)'); % might want to
remove the x-lalel here%}
```

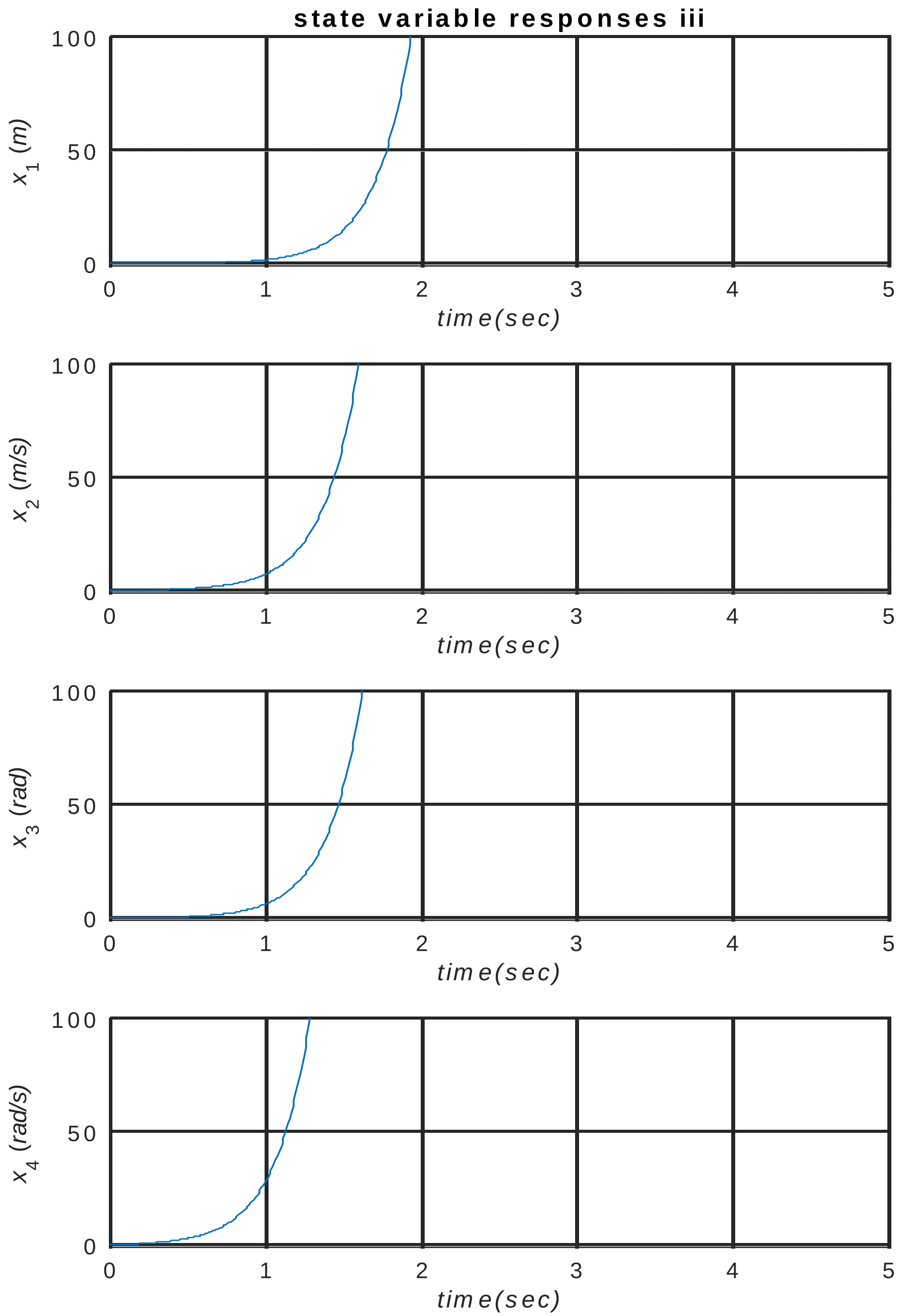
Command Output:

```
B =

         0         0
    0.5000    0.6667
         0         0
    0.3750    2.6667


sys_iii =

  A =
          x1      x2      x3      x4
   x1      0       1       0       0
   x2      0       0   4.905       0
   x3      0       0       0       1
   x4      0       0   19.62       0

  B =
           u1      u2
   x1       0       0
   x2     0.5  0.6667
   x3       0       0
   x4   0.375   2.667

  C =
       x1  x2  x3  x4
   y1   1   0   0   0
   y2   0   0   1   0

  D =
```

	u1	u2
y1	0	0
y2	0	0

Continuous-time state-space model.

Part a_iii Plots:


state variable responses iii
100
50
0
x_1 (m)
0
1
2
3
4
5
time(sec)
100
50
0
x_2 (m/s)
0
1
2
3
4
5
time(sec)
100
50
0
x_3 (rad)
0
1
2
3
4
5
time(sec)
100
50
0
x_4 (rad/s)
0
1
2
3
4
5
time(sec)

Project Part 3:

Code:

```
%Yevgen Solodkyy
%Project Part 3

clc;
m1=2; m2=1; L=.75; g=9.81;

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0];


D = 0;

t=[0:.01:7];

x0 = [0;0;0;0];

impulse = t==0;

B = [0; 1/m1; 0; 1/(L*m1)]; % need to verify what this matrix looks
like.
C = [0 0 1 0];

u=[impulse; zeros(size(t))]; %zeros(size(t)) is t x 1 dimentions.

sys_ia = ss(A,B,C,D)

det_A= det(A)% |A| =0 -> cannont be diagonalized. The diagonal canonical
form cannot be obtained.

display('|A| =0 -> Diagonal Canonitcal Form cannot be obtained.')
```

Command Output:

```
sys_ia =

  A =
          x1      x2      x3      x4
   x1      0       1       0       0
   x2      0       0   4.905       0
   x3      0       0       0       1
   x4      0       0   19.62       0

  B =
           u1
```

```
   x1        0
   x2      0.5
   x3        0
   x4   0.6667

  C =
        x1   x2   x3   x4
   y1    0    0    1    0

  D =
        u1
   y1    0

Continuous-time state-space model.


det_A =

     0

|A| =0 -> Diagonal Canonitcal Form cannot be obtained.
>>
```

Project Part 4: assess system controllability

Code:

```
%Yevgen Solodkyy
%project part 4:
clear all;
clc;
m1=2; m2=1; L=.75; g=9.81;
display('is the system controllable? If rank(P) = n yes; if rank(P) < n, no.')

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0];

%part i-ii
B = [0; 1/m1; 0; 1/(L*m1)];

P = [B A*B A^2*B A^3*B]

RANK =rank(P)
display('rank(P) = 4, so the system is controllable')
%part iii
clear B;

B = [0 0; 1/m1 1/(L*m1); 0 0; 1/m1*L (m2+m1)/(m1*m2*L^2)];
P = [B A*B A^2*B A^3*B]
```

```
RANK=rank(P)
display('rank(P) = 4, so the system is controllable')
```

Command Output:

```
is the system controllable? If rank(P) = n yes; if rank(P) < n,
no.

P =

         0    0.5000         0    3.2700
    0.5000         0    3.2700         0
         0    0.6667         0   13.0800
    0.6667         0   13.0800         0


RANK =

     4

rank(P) = 4, so the system is controllable

P =

         0         0    0.5000    0.6667         0         0
1.8394   13.0800
    0.5000    0.6667         0         0    1.8394   13.0800
0         0
         0         0    0.3750    2.6667         0         0
7.3575   52.3200
    0.3750    2.6667         0         0    7.3575   52.3200
0         0


RANK =

     4

rank(P) = 4, so the system is controllable
>>
```

Project Part 5:

Calculate the CCF for the sytem
Assessing system observability

Code:

```
%Yevgen Solodkyy
%Project part 5;

clear all; clc;

display(' Problem CME3.2b: calculate the CCF for the sytem')
%calculate cotroller canonical form for the system
m1=2; m2=1; L=.75; g=9.81;

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0];
B = [0; 1/m1; 0; 1/(L*m1)];
C = [0 0 1 0];
C_ii = [1 0 0 0;0 0 1 0]
C_iii= [1 0 0 0;0 0 1 0]




CharPoly = poly(A)
a3 = CharPoly(4)
a2 = CharPoly(3)
a1 = CharPoly(2)
a0 = CharPoly(1)

Pccf_i=[a1 a2 a3 1;a2 a3 1 0; a3 1 0 0; 1 0 0 0]
P=[B A*B A^2*B A^3*B]
Tccf=P*Pccf_i




Accf=Tccf^-1*A*Tccf
Bccf=Tccf^-1*B
Cccf=C*Tccf




%observability

display('CME 4.2a: Assessing system observability')

Q = [C;C*A;C*A^2; C*A^3]

detQ=det(Q)

display('|Q|=0, the system in cae (i) is not observeble. The system can
be reduced to an observable rank(Q)-size system:')
rank_Q = rank(Q)


T_tilda= [0,0,1,0;0,0,0,1;0,1,19.62,0;1,0,0,19.62]
```

```
T =(T_tilda')^-1

A_tilda = T^-1*A*T
C_tilda = C*T
display('The observable portion of the system is:')

A11= [0 19.62; 1 0]
C1 = [0 1]
B1=[0;1]

ObsrSYS = ss(A11,B1,C1,0)

%observability of parts (ii) & (iii)

display('the systems in cases (ii) & (iii) are observable as both Qii &
Qiii are rank 4:')


Qii = [C_ii; C_ii*A; C_ii*A^2; C_ii*A^3]
display('rank(Qii):')
rank(Qii)

Qiii = [C_iii; C_iii*A; C_iii*A^2; C_iii*A^3]
display('rank(Qiii):')
rank(Qiii)
```

command output:

Problem CME3.2b: calculate the CCF for the sytem

```
C_ii =

     1     0     0     0
     0     0     1     0


C_iii =

     1     0     0     0
     0     0     1     0


CharPoly =

    1.0000         0  -19.6200         0         0


a3 =

     0
```

```
a2 =

  -19.6200


a1 =

     0


a0 =

     1


Pccf_i =

         0  -19.6200         0    1.0000
  -19.6200         0    1.0000         0
         0    1.0000         0         0
    1.0000         0         0         0


P =

         0    0.5000         0    3.2700
    0.5000         0    3.2700         0
         0    0.6667         0   13.0800
    0.6667         0   13.0800         0


Tccf =

   -6.5400         0    0.5000         0
         0   -6.5400         0    0.5000
         0         0    0.6667         0
         0         0         0    0.6667


Accf =

         0    1.0000         0         0
         0         0    1.0000         0
         0         0         0    1.0000
         0         0   19.6200         0


Bccf =

     0
     0
     0
```

```
     1


Cccf =

         0         0    0.6667         0
```

CME 4.2a: Assessing system observability

```
Q =

         0         0    1.0000         0
         0         0         0    1.0000
         0         0   19.6200         0
         0         0         0   19.6200


detQ =

     0
```

|Q|=0, the system in cae (i) is not observeble. The system can be reduced to an observable rank(Q)-size system:

```
rank_Q =

     2


T_tilda =

         0         0    1.0000         0
         0         0         0    1.0000
         0    1.0000   19.6200         0
    1.0000         0         0   19.6200


T =

         0  -19.6200    1.0000         0
  -19.6200         0         0    1.0000
         0    1.0000         0         0
    1.0000         0         0         0


A_tilda =

         0   19.6200         0         0
    1.0000         0         0         0
         0         0         0    1.0000
         0  389.8494         0         0
```

```
C_tilda =

     0     1     0     0

The observable portion of the system is:

A11 =

         0   19.6200
    1.0000         0


C1 =

     0     1


B1 =

     0
     1


ObsrSYS =

  A =
          x1     x2
   x1      0  19.62
   x2      1      0

  B =
       u1
   x1   0
   x2   1

  C =
       x1  x2
   y1   0   1

  D =
       u1
   y1   0

Continuous-time state-space model.

the systems in cases (ii) & (iii) are observable as both Qii & Qiii are
rank 4:

Qii =

    1.0000         0         0         0
         0         0    1.0000         0
         0    1.0000         0         0
```

```
         0         0         0    1.0000
         0         0    4.9050         0
         0         0   19.6200         0
         0         0         0    4.9050
         0         0         0   19.6200
```

rank(Qii):

```
ans =

     4


Qiii =

    1.0000         0         0         0
         0         0    1.0000         0
         0    1.0000         0         0
         0         0         0    1.0000
         0         0    4.9050         0
         0         0   19.6200         0
         0         0         0    4.9050
         0         0         0   19.6200
```

rank(Qiii):

```
ans =

     4

>>



>>
```

Project Part 6:

Assess the stability properties of the CE2.2 system

```
%Yevgen Solodkyy
%project part 6
%assess the stability properties of the CE2.2 system
```

Code:

```
clear all;
clc;

display('assess the stability properties of the system')
```

```
m1=2; m2=1; L=.75; g=9.81;

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0]

display('Eigen value approach:')

eig(A)

display('because Re{eigen values} are not strictly negative, the
Lyapunov approach will fail.')


display('Lyapunov approach:')

Q = eye(4)
S=lyap(A',Q)
display('|s(1,1)|:')
det(S(1,1))
display('|s(1:2,1:2)|:')
det(S(1:2,1:2))
display('|s(1:3,1:3)|:')
det(S(1:3,1:3))
display('|s(1:3,1:3)|:')
det(S)
```

Command Output:

assess the stability properties of the system

A =

0	1.0000	0	0
0	0	4.9050	0
0	0	0	1.0000
0	0	19.6200	0

Eigen value approach:

ans =

0

0

4.4294

-4.4294

because **Re{eigen values}** are not strictly negative, the Lyapunov approach will fail.

Lyapunov approach:

Q =

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Error using lyap (line 69)

The solution of this Lyapunov equation does not exist or is not unique.

Error in Project_part6 (line 25)

S=lyap(A',Q)

>>

Project part 7:

`Design state feedback control laws for closed loop eigenvalues`

7-i(a):

code:

```
%Yevgen Solodkyy
%Project part 7;
%design state feedback control laws for closed loop eigenvalues.

clear all; clc;

display('Design state feedback control laws for closed loop
eigenvalues')

clear all;
clc;
m1=2; m2=1; L=.75; g=9.81;

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0];
B_i = [0; 1/m1; 0; 1/(L*m1)];
B_iii = [0 0; 1/m1 1/(L*m1); 0 0; 1/m1*L (m2+m1)/(m1*m2*L^2)];  % need
to verify again the last value of B.
C = [0 0 1 0];
C_ii = [1 0 0 0;0 0 1 0];
C_iii= [1 0 0 0;0 0 1 0];
D=0;



%desired eigven values
l1= -1.27+3.79i;
l2= -1.27-3.79i;
l3= -1.88+1.24i;
l4= -1.88-1.24i;


DesEig = [l1; l2; l3; l4]

%open loop state space for part i:
Sys_ol_i = ss(A,B_i,C,D)

% K for part i:
K_i=place(A,B_i,DesEig)

Ac = A-B_i*K_i;

%closed loop state space for part i:
Sys_cl_i = ss(Ac,B_i,C,D)

[Yc,t,Xc] = impulse(Sys_cl_i);
[YO,tO,XO] = impulse(Sys_ol_i);


figure(1)
```

```
subplot(411),
plot(t,Xc(:,1),'green',tO,XO(:,1),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_1 (\itm)');
legend('Closed-loop','Open-loop');
title('7 i(a): Open vs. closed-loop responses', 'FontSize',14);

subplot(412),
plot(t,Xc(:,2),'green',tO,XO(:,2),'red');
grid;
axis([0 12 -.6 1]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_2 (\itm/s)');
legend('Closed-loop','Open-loop');


subplot(413),
plot(t,Xc(:,3),'green',tO,XO(:,3),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_3 (\itrad)');
legend('Closed-loop','Open-loop');



subplot(414),
plot(t,Xc(:,3),'green',tO,XO(:,3),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_3 (\itrad)');
legend('Closed-loop','Open-loop');


figure(2)
%subplot(411),
plot(t,Yc(:,1),'green',tO,YO(:,1),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\ity}_1 (\itm)');
legend('Closed-loop','Open-loop');
title('7 i(a): Open/closed-loop responses', 'FontSize',14);
```

command output:

DesEig =

-1.2700 + 3.7900i

-1.2700 - 3.7900i

-1.8800 + 1.2400i

-1.8800 - 1.2400i

Sys_ol_i =

A =

	x1	x2	x3	x4
x1	0	1	0	0
x2	0	0	4.905	0
x3	0	0	0	1
x4	0	0	19.62	0

B =

	u1
x1	0
x2	0.5

x3 0

x4 0.6667

C =

	x1	x2	x3	x4
y1	0	0	1	0

D =

	u1
y1	0

Continuous-time state-space model.

K_i =

-12.3907 -11.1554 84.6221 17.8166

Sys_cl_i =

A =

	x1	x2	x3	x4
x1	0	1	0	0

x2 6.195 5.578 -37.41 -8.908

x3 0 0 0 1

x4 8.26 7.437 -36.79 -11.88

B =

	u1
x1	0
x2	0.5
x3	0
x4	0.6667

C =

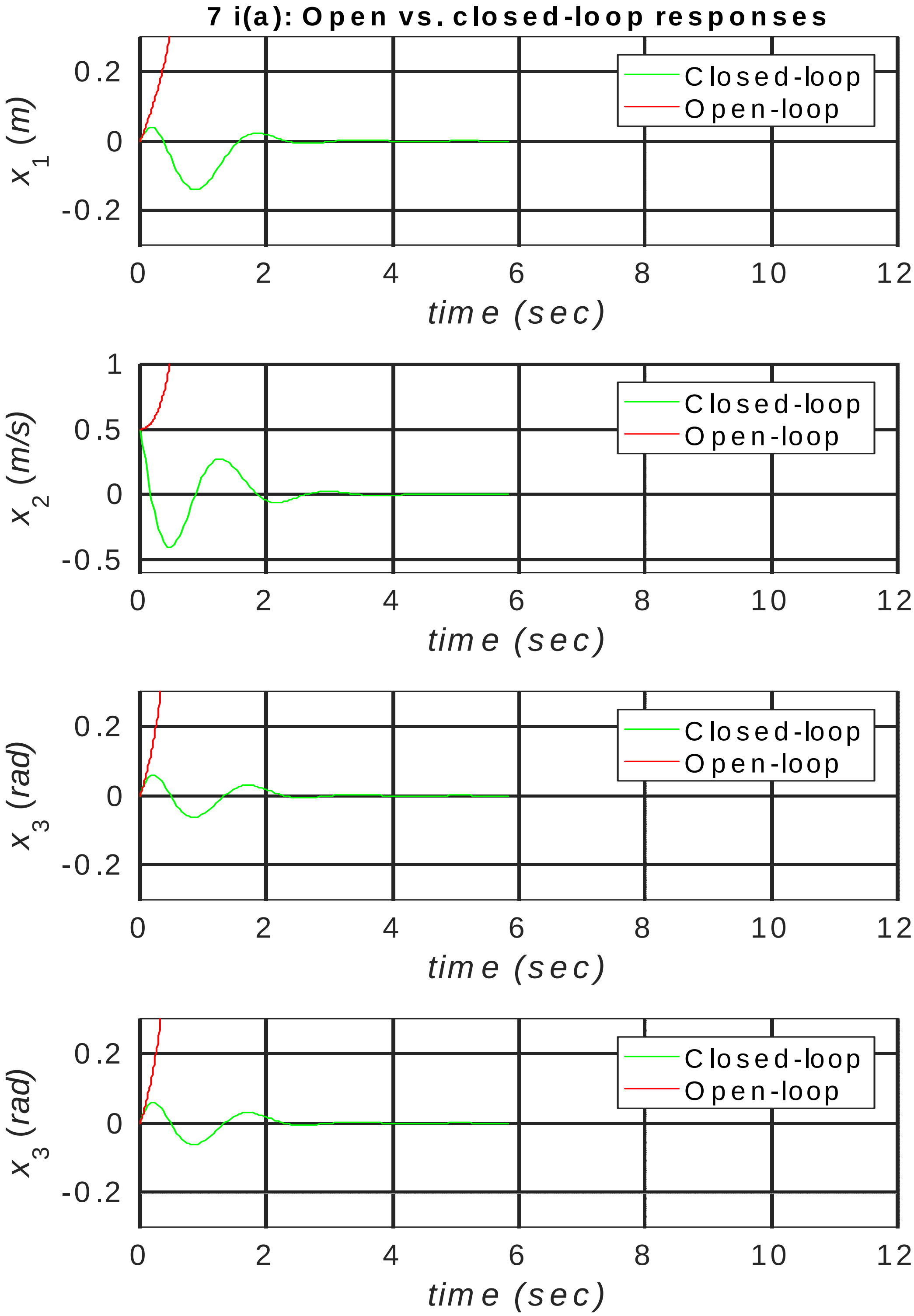
	x1	x2	x3	x4
y1	0	0	1	0

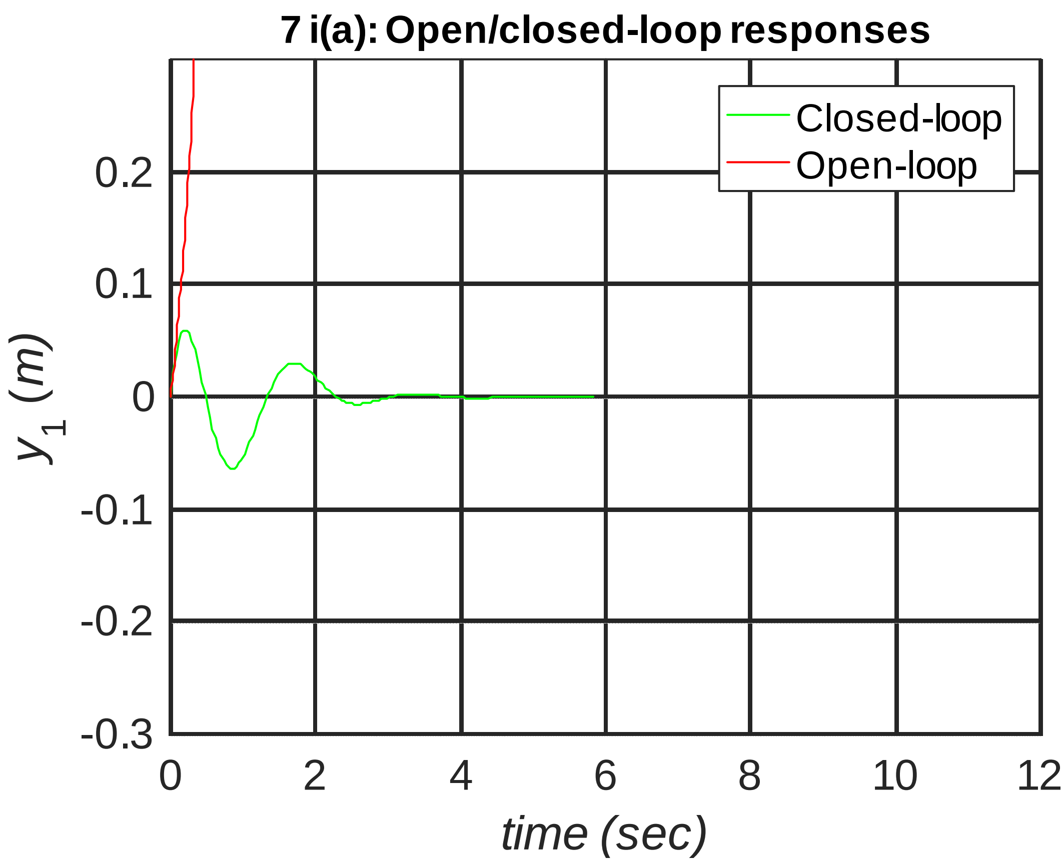
D =

	u1
y1	0

Continuous-time state-space model.

7-i(a) Plots:


7 i(a): Open vs. closed-loop responses
Closed-loop
Open-loop
x_1 (m)
x_2 (m/s)
x_3 (rad)
x_3 (rad)
time (sec)
0
2
4
6
8
10
12
0.2
0
-0.2
1
0.5
-0.5



part 7-i(b):

code:

```
%Yevgen Solodkyy
%Project part 7;
%design state feedback control laws for closed loop eigenvalues.

clear all; clc;

display('Design state feedback control laws for closed loop
eigenvalues')

clear all;
clc;
m1=2; m2=1; L=.75; g=9.81;

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0];
B_i = [0; 1/m1; 0; 1/(L*m1)];
%B_iii = [0 0; 1/m1 1/(L*m1); 0 0; 1/m1*L (m2+m1)/(m1*m2*L^2)];  % need
to verify again the last value of B.
C = [0 0 1 0];
```

```
%C_ii = [1 0 0 0;0 0 1 0]
%C_iii= [1 0 0 0;0 0 1 0]
D=0;



%desired eigven values
l1= -1.27+3.79i;
l2= -1.27-3.79i;
l3= -1.88+1.24i;
l4= -1.88-1.24i;


DesEig = [l1; l2; l3; l4]

%open loop state space for part i:
Sys_oli = ss(A,B_i,C,D)

% K for part i:
K_i=place(A,B_i,DesEig)

Ac = A-B_i*K_i;

%closed loop state space for part i:
Sys_cl = ss(Ac,B_i,C,D)



t1 = 0:.1:7;

X0 = [0;0;0.1;0];

u=[zeros(size(t1))];



[Yc,t,Xc] = lsim(Sys_cl,u,t1,X0);

[YO,tO,XO] = lsim(Sys_oli,u,t1,X0);

figure(1);
subplot(411),
plot(t,Xc(:,1),'green',tO,XO(:,1),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_1 (\itm)');
legend('Closed-loop','Open-loop');
title('7 i-(b): Open/closed-loop responses', 'FontSize',14);

subplot(412),
```

```
plot(t,Xc(:,2),'green',tO,XO(:,2),'red');
grid;
axis([0 12 -.6 .5]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_2 (\itm/s)');
legend('Closed-loop','Open-loop');


subplot(413),
plot(t,Xc(:,3),'green',tO,XO(:,3),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_3 (\itrad)');
legend('Closed-loop','Open-loop');



subplot(414),
plot(t,Xc(:,3),'green',tO,XO(:,3),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_3 (\itrad)');
legend('Closed-loop','Open-loop');

figure(2)

plot(t,Yc(:,1),'green',tO,YO(:,1),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\ity}_1 (\itm)');
legend('Closed-loop','Open-loop');
title('7 i(b): Open/closed-loop responses', 'FontSize',14);
```

command output:

DesEig =

-1.2700 + 3.7900i

-1.2700 - 3.7900i

-1.8800 + 1.2400i

-1.8800 - 1.2400i

Sys_oli =

A =

	x1	x2	x3	x4
x1	0	1	0	0
x2	0	0	4.905	0
x3	0	0	0	1
x4	0	0	19.62	0

B =

	u1
x1	0
x2	0.5
x3	0
x4	0.6667

C =

	x1	x2	x3	x4
y1	0	0	1	0

D =

	u1
y1	0

Continuous-time state-space model.

K_i =

-12.3907 -11.1554 84.6221 17.8166

Sys_cl =

A =

	x1	x2	x3	x4
x1	0	1	0	0
x2	6.195	5.578	-37.41	-8.908
x3	0	0	0	1
x4	8.26	7.437	-36.79	-11.88

B =

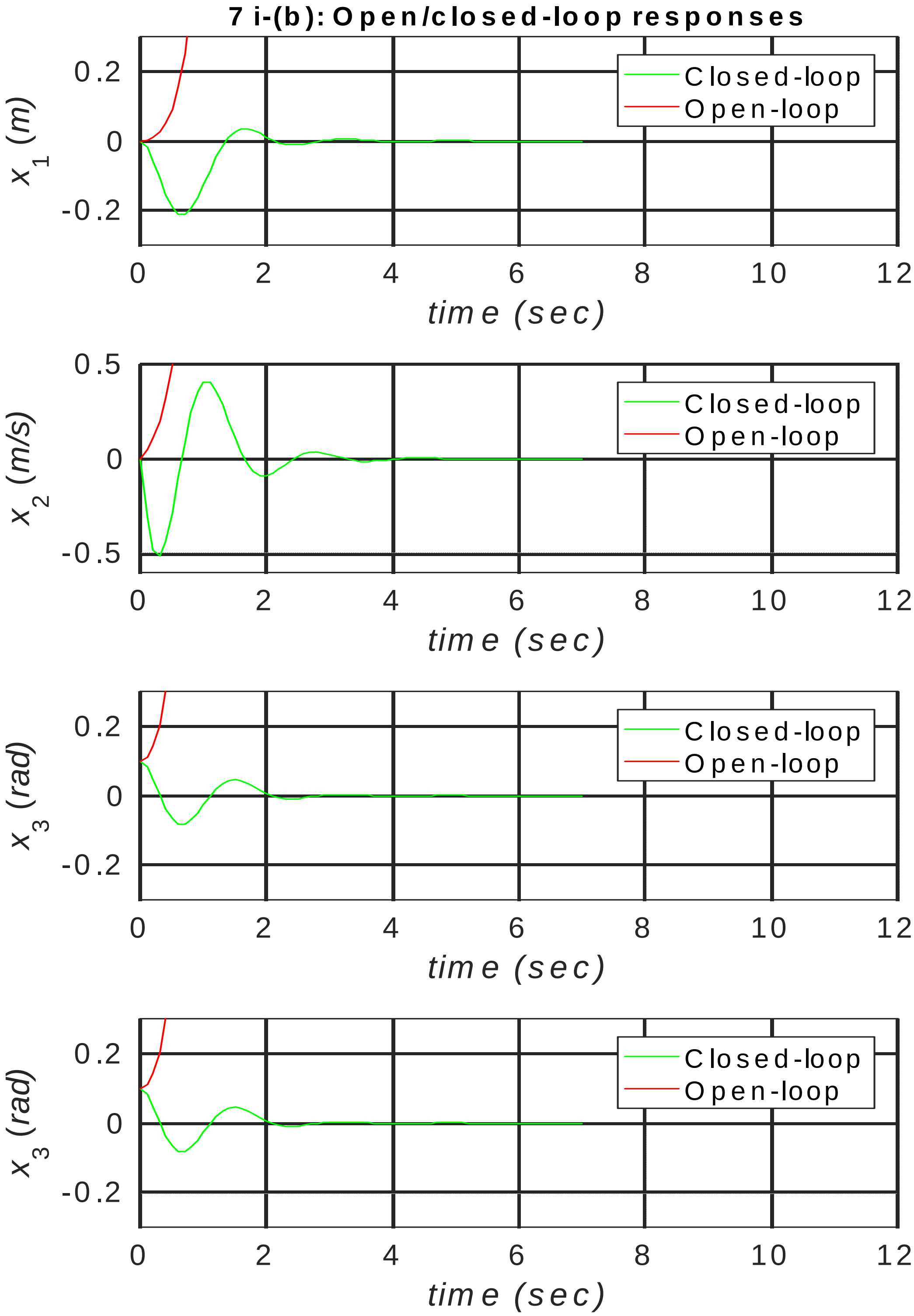
u1

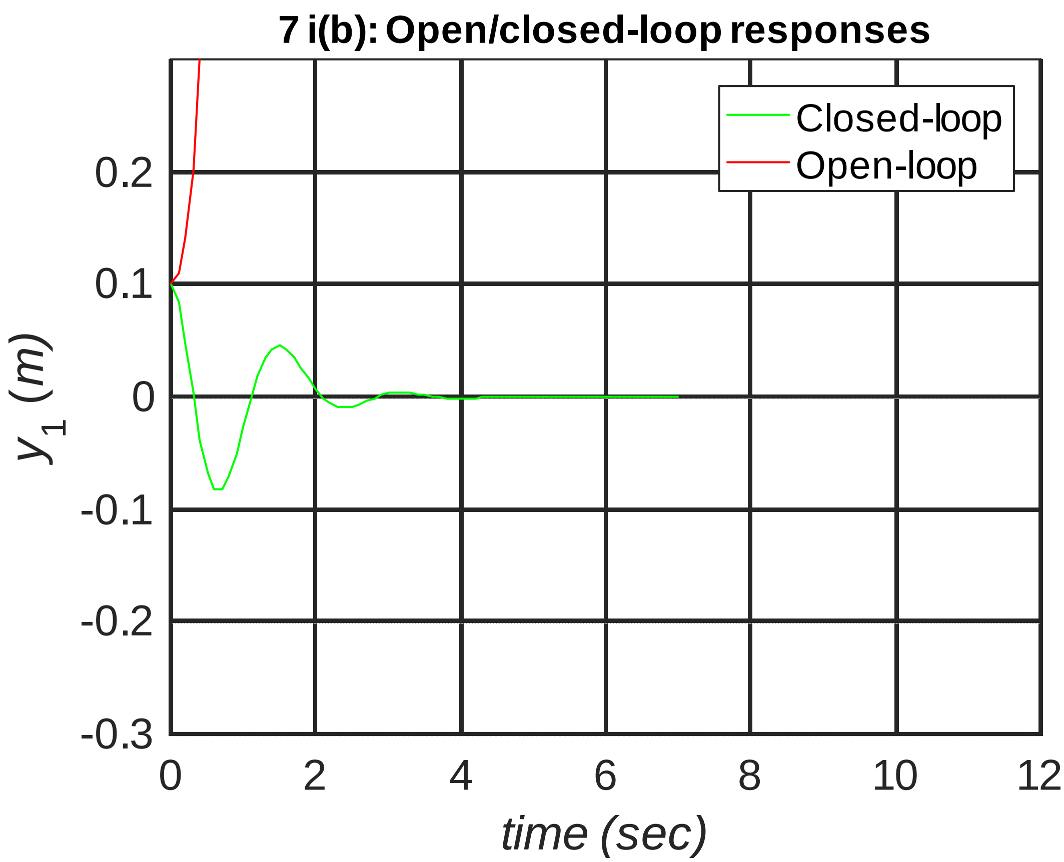
```
  x1       0
  x2     0.5
  x3       0
  x4  0.6667


 C =

     x1  x2  x3  x4
  y1   0   0   1   0


 D =

     u1
  y1   0
```

Continuous-time state-space model.

>>

Plots:


7 i-(b): Open/closed-loop responses
Closed-loop
Open-loop
x1 (m)
0.2
0
-0.2
0
2
4
6
8
10
12
time (sec)
x2 (m/s)
0.5
0
-0.5
x3 (rad)
0.2
0
-0.2
x3 (rad)



part 7-ii:

```
%Yevgen Solodkyy
%Project part 7;
%design state feedback control laws for closed loop eigenvalues.

clear all; clc;

display('Design state feedback control laws for closed loop
eigenvalues')

clear all;
clc;
m1=2; m2=1; L=.75; g=9.81;

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0];
B_i = [0; 1/m1; 0; 1/(L*m1)];
B_iii = [0 0; 1/m1 1/(L*m1); 0 0; 1/m1*L (m2+m1)/(m1*m2*L^2)];  % need
to verify again the last value of B.
C = [0 0 1 0];
```

```
C_ii = [1 0 0 0;0 0 1 0]
C_iii= [1 0 0 0;0 0 1 0]
D=0;



%desired eigven values
l1= -1.27+3.79i;
l2= -1.27-3.79i;
l3= -1.88+1.24i;
l4= -1.88-1.24i;


DesEig = [l1; l2; l3; l4]


% K for part i:
K_i=place(A,B_i,DesEig)



K_ii = K_i % parts i & ii have the same B matrix.

%open look state space for part ii:
Sys_ol_ii = ss(A,B_i,C_ii,D)

% K for part i:

Ac_ii = A-B_i*K_ii;

Sys_cl_ii = ss(Ac_ii,B_i,C_ii,D)

[Yc,t,Xc] = impulse(Sys_cl_ii);
[YO,tO,XO] = impulse(Sys_ol_ii);


figure(1)
subplot(411),
plot(t,Xc(:,1),'green',tO,XO(:,1),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_1 (\itm)');
legend('Closed-loop','Open-loop');
title('7 (i): Open vs. closed-loop responses', 'FontSize',14);

subplot(412),
plot(t,Xc(:,2),'green',tO,XO(:,2),'red');
grid;
axis([-1 12 -.5 1]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
```

```
ylabel('{\itx}_2 (\itm/s)');
legend('Closed-loop','Open-loop');

subplot(413),
plot(t,Xc(:,3),'green',tO,XO(:,3),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_3 (\itrad)');
legend('Closed-loop','Open-loop');

subplot(414),
plot(t,Xc(:,3),'green',tO,XO(:,3),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_3 (\itrad)');
legend('Closed-loop','Open-loop');


figure(2)
subplot(211),
plot(t,Yc(:,1),'green',tO,YO(:,1),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\ity}_1 (\itm)');
legend('Closed-loop','Open-loop');
title('7 ii: Open/closed-loop responses', 'FontSize',14);

subplot(212),
plot(t,Yc(:,2),'green',tO,YO(:,2),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\ity}_2 (\itm)');
legend('Closed-loop','Open-loop');
title('7 ii: Open/closed-loop responses', 'FontSize',14);
```

command output:

C_ii =

```
    1     0     0     0

    0     0     1     0


DesEig =

  -1.2700 + 3.7900i
  -1.2700 - 3.7900i
  -1.8800 + 1.2400i
  -1.8800 - 1.2400i


K_i =

  -12.3907  -11.1554   84.6221   17.8166


K_ii =

  -12.3907  -11.1554   84.6221   17.8166


Sys_ol_ii =

  A =
        x1    x2    x3    x4
   x1    0     1     0     0
```

```
 x2    0    0  4.905    0
 x3    0    0    0    1
 x4    0    0  19.62    0

 B =
        u1
 x1      0
 x2    0.5
 x3      0
 x4  0.6667

 C =
     x1  x2  x3  x4
 y1   1   0   0   0
 y2   0   0   1   0

 D =
     u1
 y1   0
 y2   0

Continuous-time state-space model.



Sys_cl_ii =
```

A =

	x1	**x2**	**x3**	**x4**
x1	**0**	**1**	**0**	**0**
x2	**6.195**	**5.578**	**-37.41**	**-8.908**
x3	**0**	**0**	**0**	**1**
x4	**8.26**	**7.437**	**-36.79**	**-11.88**

B =

	u1
x1	**0**
x2	**0.5**
x3	**0**
x4	**0.6667**

C =

	x1	**x2**	**x3**	**x4**
y1	**1**	**0**	**0**	**0**
y2	**0**	**0**	**1**	**0**

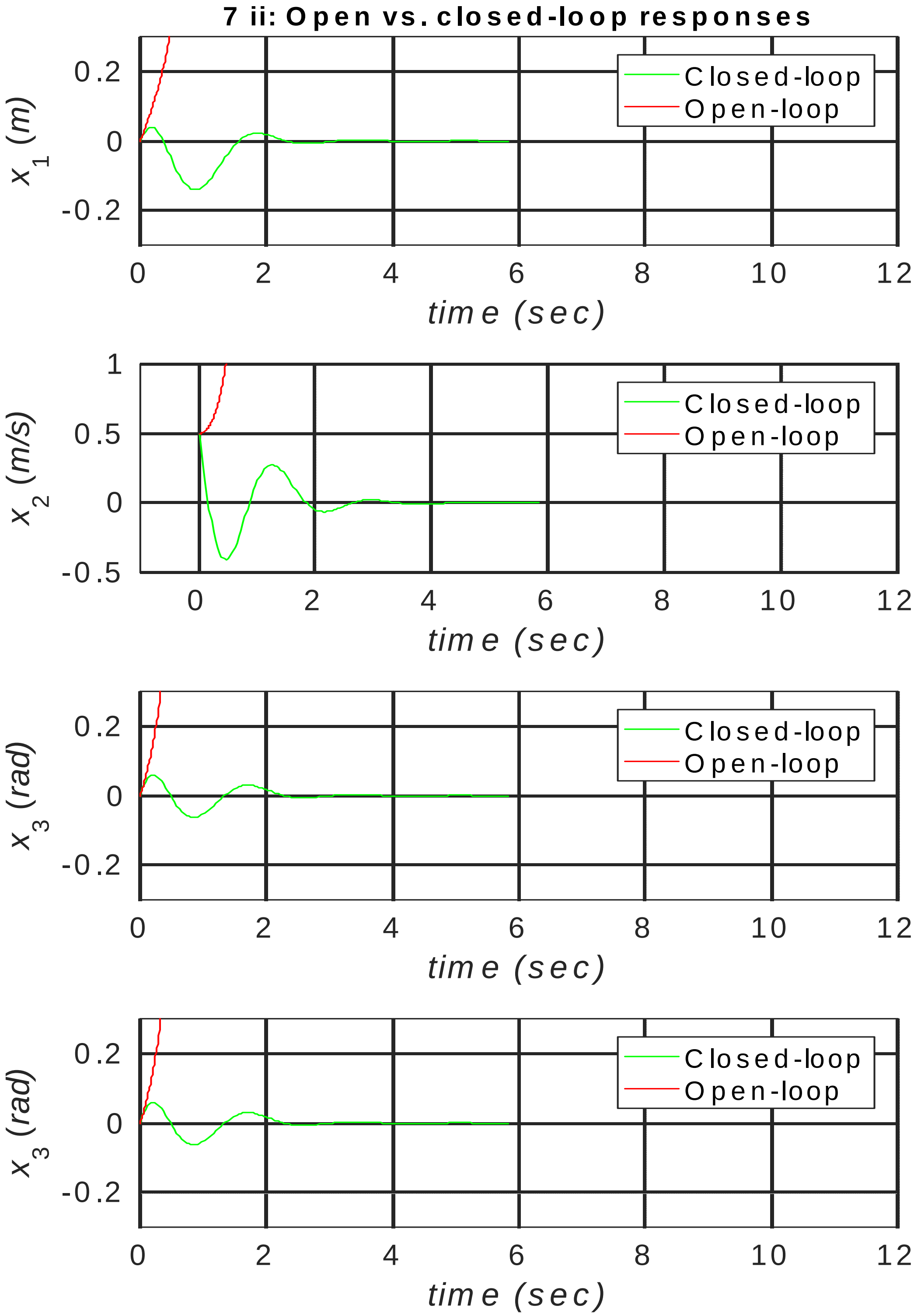
D =

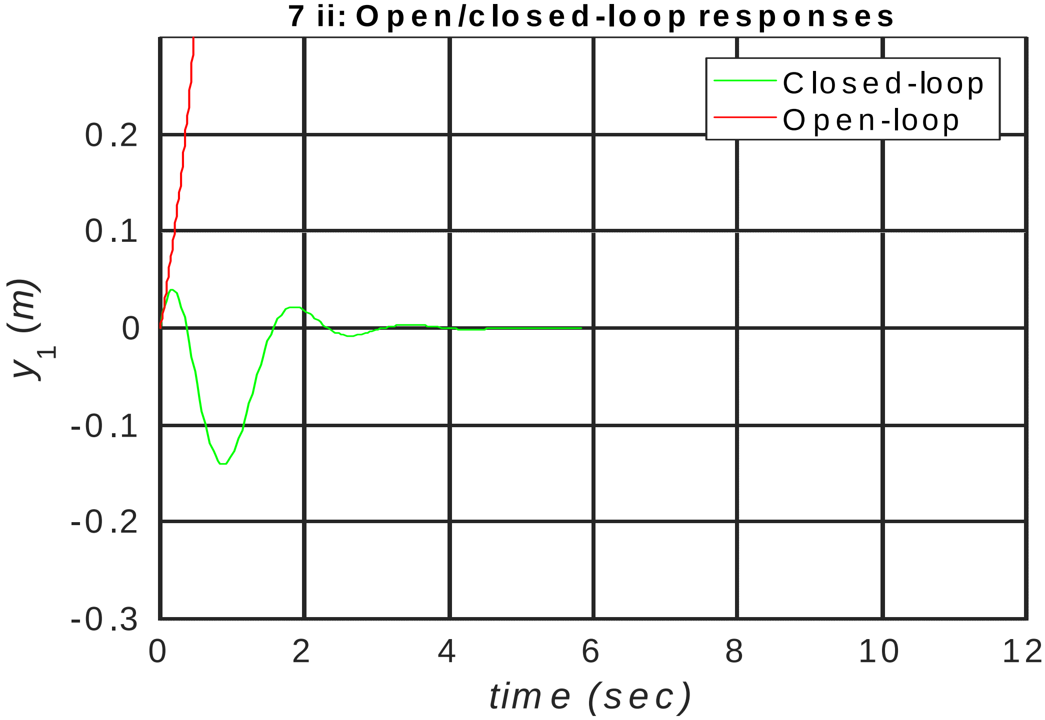
	u1
y1	**0**
y2	**0**

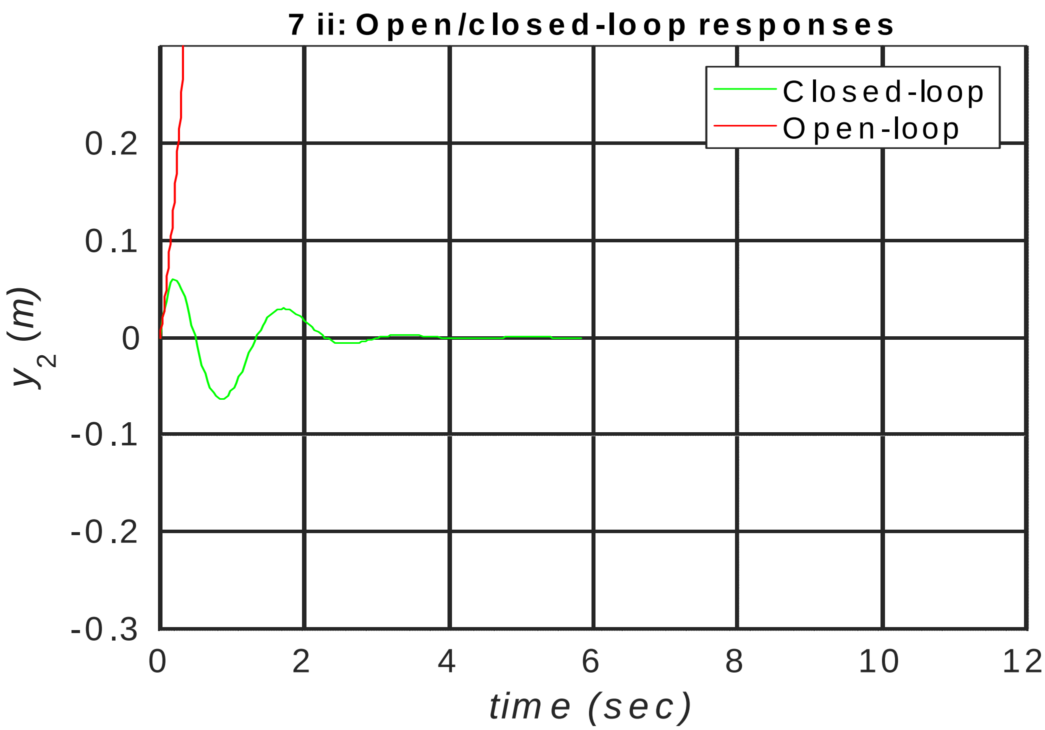
Continuous-time state-space model.

>>

Plots:


7 ii: Open vs. closed-loop responses
Closed-loop
Open-loop
0.2
0
-0.2
x 1 (m)
0
2
4
6
8
10
12
time (sec)
1
0.5
0
-0.5
x 2 (m/s)
x 3 (rad)
x 3 (rad)


7 ii: Open/closed-loop responses
Closed-loop
Open-loop
0.2
0.1
0
-0.1
-0.2
-0.3
0
2
4
6
8
10
12
time (sec)
y1 (m)


7 ii: Open/closed-loop responses
Closed-loop
Open-loop
0.2
0.1
0
-0.1
-0.2
-0.3
0
2
4
6
8
10
12
time (sec)
y2 (m)

part 7_iii:

code:

```
%Yevgen Solodkyy
%Project part 7;
%design state feedback control laws for closed loop eigenvalues.

clear all; clc;

display('Design state feedback control laws for closed loop
eigenvalues')

clear all;
clc;
m1=2; m2=1; L=.75; g=9.81;

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0];
B_i = [0; 1/m1; 0; 1/(L*m1)];
B_iii = [0 0; 1/m1 1/(L*m1); 0 0; 1/m1*L (m2+m1)/(m1*m2*L^2)];  % need
to verify again the last value of B.
C = [0 0 1 0];
C_ii = [1 0 0 0;0 0 1 0]
C_iii= [1 0 0 0;0 0 1 0]
D=0;



%desired eigven values
l1= -1.27+3.79i;
l2= -1.27-3.79i;
l3= -1.88+1.24i;
l4= -1.88-1.24i;


DesEig = [l1; l2; l3; l4]



K_iii=place(A,B_iii,DesEig)

Sys_ol_iii = ss(A,B_iii,C_iii,D)
Ac_iii = A-B_iii*K_iii;
Sys_cl_iii = ss(Ac_iii,B_iii,C_iii,D)


[Yc,t,Xc] = step(Sys_cl_iii);
[YO,tO,XO] = step(Sys_ol_iii);
```

```
figure()
subplot(411),
plot(t,Xc(:,1),'green',tO,XO(:,1),'red');
grid;
axis([0 12 -.2 .2]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_1 (\itm)');
legend('Closed-loop','Open-loop');
title('7 iii: Open/closed-loop responses', 'FontSize',14);

subplot(412),
plot(t,Xc(:,2),'green',tO,XO(:,2),'red');
grid;
axis([0 12 -.2 .2]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_2 (\itm/s)');
legend('Closed-loop','Open-loop');


subplot(413),
plot(t,Xc(:,3),'green',tO,XO(:,3),'red');
grid;
axis([0 12 -.2 .2]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_3 (\itrad)');
legend('Closed-loop','Open-loop');



subplot(414),
plot(t,Xc(:,3),'green',tO,XO(:,3),'red');
grid;
axis([0 12 -.2 .2]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_3 (\itrad)');
legend('Closed-loop','Open-loop');

figure(2)
subplot(211),
plot(t,Yc(:,1),'green',tO,YO(:,1),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\ity}_1 (\itm)');
legend('Closed-loop','Open-loop');
title('7 ii: Open/closed-loop responses', 'FontSize',14);
```

```
subplot(212),
plot(t,Yc(:,2),'green',tO,YO(:,2),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\ity}_2 (\itm)');
legend('Closed-loop','Open-loop');
title('7 iii: Open/closed-loop responses', 'FontSize',14);
```

command output:

C_iii =

1 0 0 0

0 0 1 0

DesEig =

-1.2700 + 3.7900i

-1.2700 - 3.7900i

-1.8800 + 1.2400i

-1.8800 - 1.2400i

K_iii =

20.8408	**9.3247**	**9.3042**	**4.3430**
-5.0110	**-2.2669**	**8.7096**	**0.5701**

Sys_ol_iii =

A =

	x1	x2	x3	x4
x1	0	1	0	0
x2	0	0	4.905	0
x3	0	0	0	1
x4	0	0	19.62	0

B =

	u1	u2
x1	0	0
x2	0.5	0.6667
x3	0	0
x4	0.375	2.667

C =

	x1	x2	x3	x4
y1	1	0	0	0
y2	0	0	1	0

D =

	u1	u2
y1	0	0
y2	0	0

Continuous-time state-space model.

Sys_cl_iii =

A =

	x1	x2	x3	x4
x1	0	1	0	0
x2	-7.08	-3.151	-5.554	-2.552
x3	0	0	0	1
x4	5.547	2.548	-7.095	-3.149

B =

	u1	u2
x1	0	0
x2	0.5	0.6667
x3	0	0
x4	0.375	2.667

C =

	x1	x2	x3	x4
y1	1	0	0	0
y2	0	0	1	0

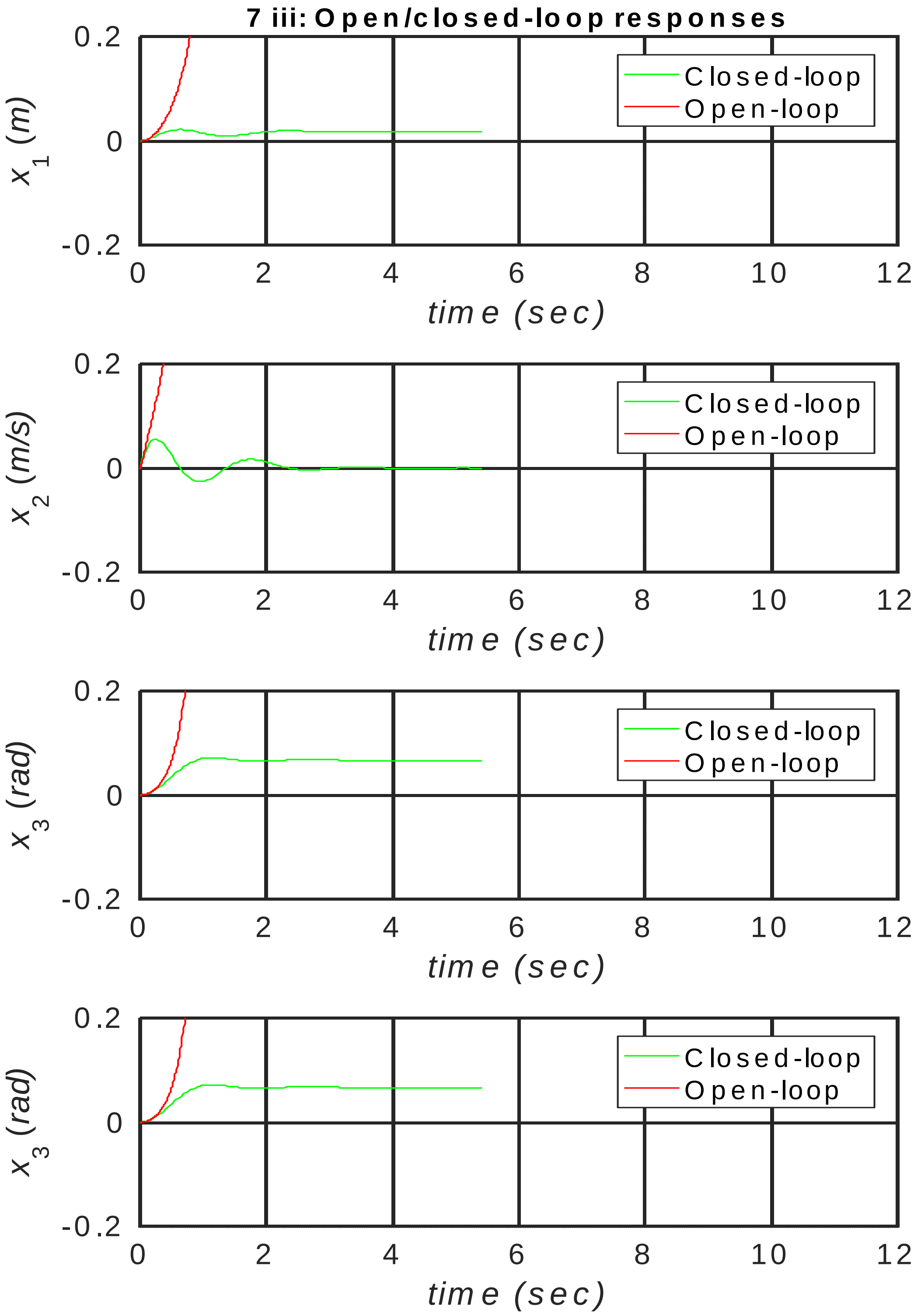
D =

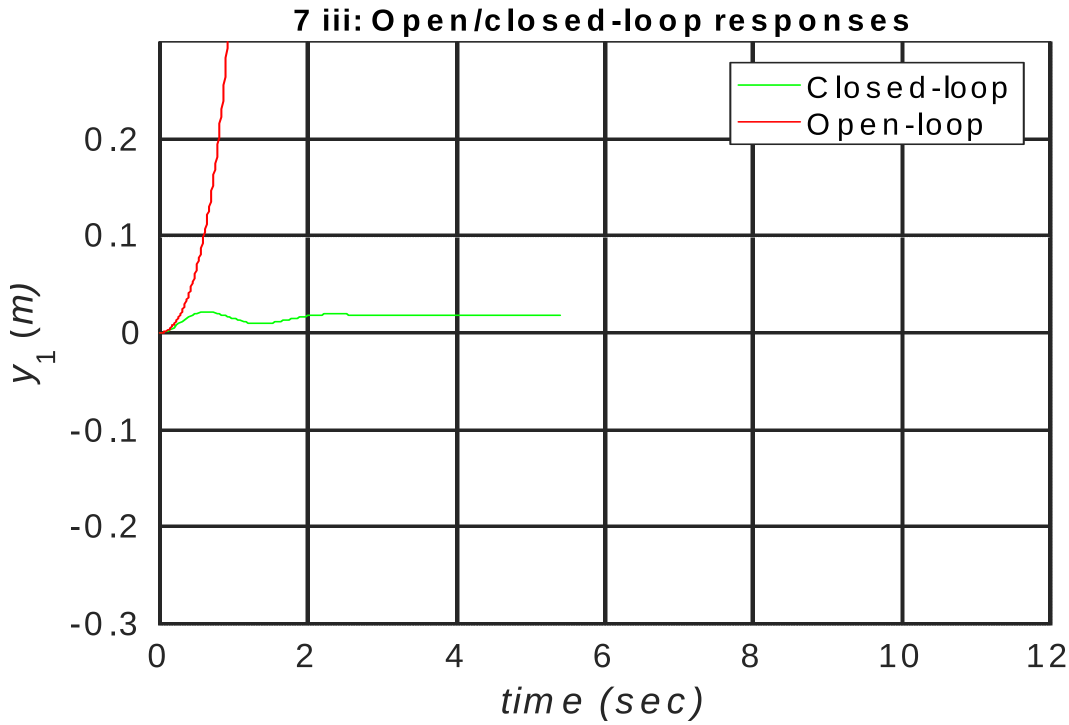
	u1	u2
y1	0	0
y2	0	0

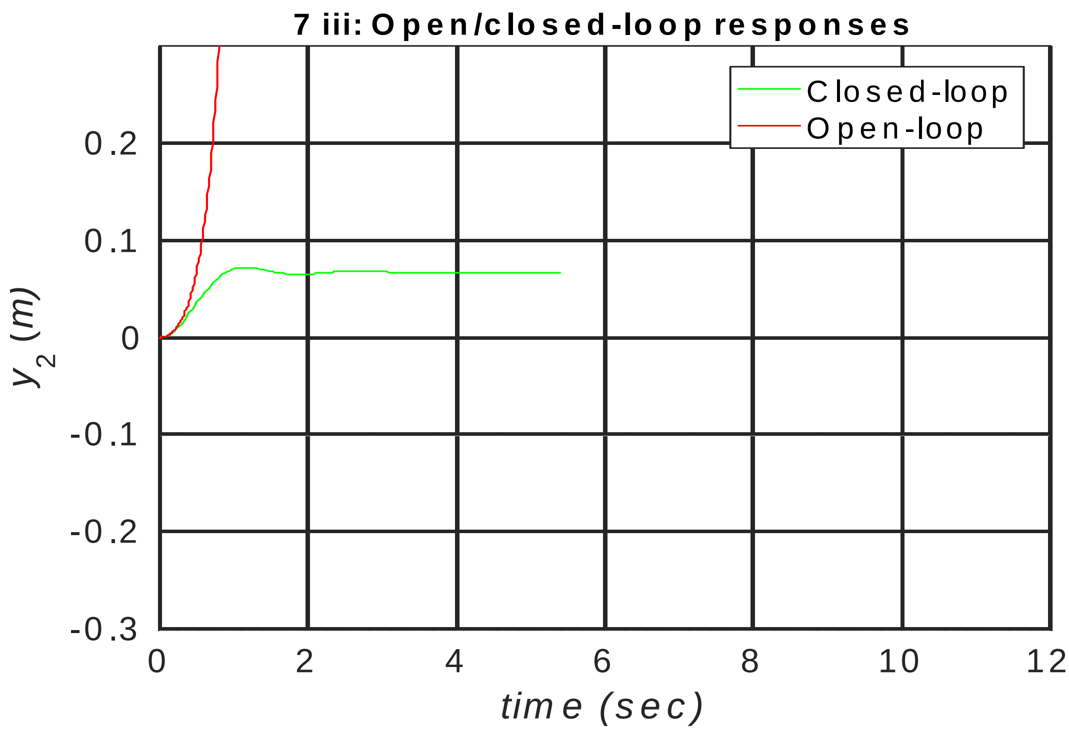
Continuous-time state-space model.

>>

Plots:


7 iii: Open/closed-loop responses
0.2
0
-0.2
x_1 (m)
0 2 4 6 8 10 12
time (sec)
Closed-loop
Open-loop
x_2 (m/s)
x_3 (rad)
x_3 (rad)


7 iii: Open/closed-loop responses
Closed-loop
Open-loop
0.2
0.1
0
-0.1
-0.2
-0.3
y_1 (m)
0
2
4
6
8
10
12
time (sec)


7 iii: Open/closed-loop responses
Closed-loop
Open-loop
0.2
0.1
0
-0.1
-0.2
-0.3
y_2 (m)
0
2
4
6
8
10
12
time (sec)

Project part 8: Observer Compensator

part 8(i) system is not observable as a result, I had difficulty creating a reduced observer compensator for the sytem.

Systems 8(ii) & 8(iii) were observable, and the following observer based compensators were created:

Part 8(i):

code:

```
%Yevgen Solodkyy
%Project part 8 ii;
%

clear all; clc;



clear all;
clc;
m1=2; m2=1; L=.75; g=9.81;

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0];
B_i = [0; 1/m1; 0; 1/(L*m1)];
B_iii = [0 0; 1/m1 1/(L*m1); 0 0; 1/m1*L (m2+m1)/(m1*m2*L^2)];  % need
to verify again the last value of B.
C = [0 0 1 0];
C_ii = [1 0 0 0;0 0 1 0]
C_iii= [1 0 0 0;0 0 1 0]
D=0;



%desired eigven values
l1= -1.27+3.79i;
l2= -1.27-3.79i;
l3= -1.88+1.24i;
l4= -1.88-1.24i;


DesEig = [l1; l2; l3; l4]
```

```
% K for part i:
K_i=place(A,B_i,DesEig)



K_ii = K_i % parts i & ii have the same B matrix.

%open look state space for part ii:
Sys_ol_ii = ss(A,B_i,C_ii,D)

% K for part i:

Ac_ii = A-B_i*K_ii;

Sys_cl_ii = ss(Ac_ii,B_i,C_ii,D)

[Yc,t,Xc] = impulse(Sys_cl_ii);
[YO,tO,XO] = impulse(Sys_ol_ii);






%% Observer portion


ObsEig = 100*DesEig; % observer eighen values 10x desired eigen values

%Q= [C; C*A; C*A^2; C*A^3];



L=place(A',C_ii',ObsEig)';
Ahat=A-L*C_ii  % declaring oberver state matrix






Ar = [(A-B_i*K_i) B_i*K_i;zeros(size(A)) (A-L*C_ii)];
Br = [B_i;zeros(size(B_i))];
Cr = [C_ii zeros(size(C_ii))];
Dr = D;

Sys_r = ss(Ar,Br,Cr,Dr);

tr = [0:0.01:10];

r = [zeros(size(tr))];
Xr0 = [0.1;0.1;0.1;0.1;0.1;0.1;0.1;0.1]; %initial error conditions
```

```
%create vecros like these for the ope and closed loop systems as well.
see
%page 345 for reference

[Yr,tr,Xr] = lsim(Sys_r,r,tr,Xr0);




%%
figure(1)
subplot(411),
plot(t,Xc(:,1),'green',tO,XO(:,1),'red',tr,Xr(:,1),'blue')     %plot
(tO,YO(:1),'r',tc,Yc,'g',t,Yr,'b'); grid;
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_1 (\itm)');
legend('Closed-loop','Open-loop','Observer');
title('8 ii: Open vs. closed-loop responses , & observer, '); %
%legend('open loop','closee loop','w/ Observer');

subplot(412),
plot(t,Xc(:,2),'green',tO,XO(:,2),'red',tr,Xr(:,2),'b'); %plot(t,Xc
(:,2),'green',tO,XO(:,2),'red');
grid;
axis([0 12 -.5 1]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_2 (\itm/s)');
legend('Closed-loop','Open-loop','Observer');

subplot(413),
plot(t,Xc(:,3),'green',tO,XO(:,3),'red',tr,Xr(:,3),'b'); %plot(t,Xc
(:,3),'green',tO,XO(:,3),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_3 (\itrad)');
legend('Closed-loop','Open-loop','Observer');

subplot(414),
plot(t,Xc(:,3),'green',tO,XO(:,3),'red',tr,Xr(:,3),'b'); %plot(t,Xc
(:,3),'green',tO,XO(:,3),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_3 (\itrad)');
legend('Closed-loop','Open-loop','Observer');
```

```
figure(2)
subplot(211),
plot(t,Xc(:,1),'green',tO,XO(:,1),'red',tr,Yr(:,1),'b'); %plot(t,Yc(:,1),'green',tO,YO(:,1),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\ity}_1 (\itm)');
legend('Closed-loop','Open-loop','Observer');
title('8 ii: Open/closed-loop responses', 'FontSize',14);

subplot(212),
plot(t,Xc(:,2),'green',tO,XO(:,2),'red',tr,Yr(:,2),'b');%plot(t,Yc(:,2),'green',tO,YO(:,2),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\ity}_2 (\itm)');
legend('Closed-loop','Open-loop','Observer');
title('8 ii: Open/closed-loop responses', 'FontSize',14);
```

Command Output:

C_ii =

1 0 0 0

0 0 1 0

C_iii =

1 0 0 0

0 0 1 0

DesEig =

-1.2700 + 3.7900i

-1.2700 - 3.7900i

-1.8800 + 1.2400i

-1.8800 - 1.2400i

K_i =

-12.3907 -11.1554 84.6221 17.8166

K_ii =

-12.3907 -11.1554 84.6221 17.8166

Sys_ol_ii =

A =

	x1	x2	x3	x4
x1	0	1	0	0
x2	0	0	4.905	0
x3	0	0	0	1

```
 x4      0      0  19.62      0


 B =

        u1

  x1      0

  x2    0.5

  x3      0

  x4  0.6667


 C =

     x1  x2  x3  x4

  y1   1   0   0   0

  y2   0   0   1   0


 D =

     u1

  y1   0

  y2   0

Continuous-time state-space model.



Sys_cl_ii =


 A =

         x1      x2      x3      x4
```

x1	**0**	**1**	**0**	**0**
x2	**6.195**	**5.578**	**-37.41**	**-8.908**
x3	**0**	**0**	**0**	**1**
x4	**8.26**	**7.437**	**-36.79**	**-11.88**

B =

	u1
x1	**0**
x2	**0.5**
x3	**0**
x4	**0.6667**

C =

	x1	**x2**	**x3**	**x4**
y1	**1**	**0**	**0**	**0**
y2	**0**	**0**	**1**	**0**

D =

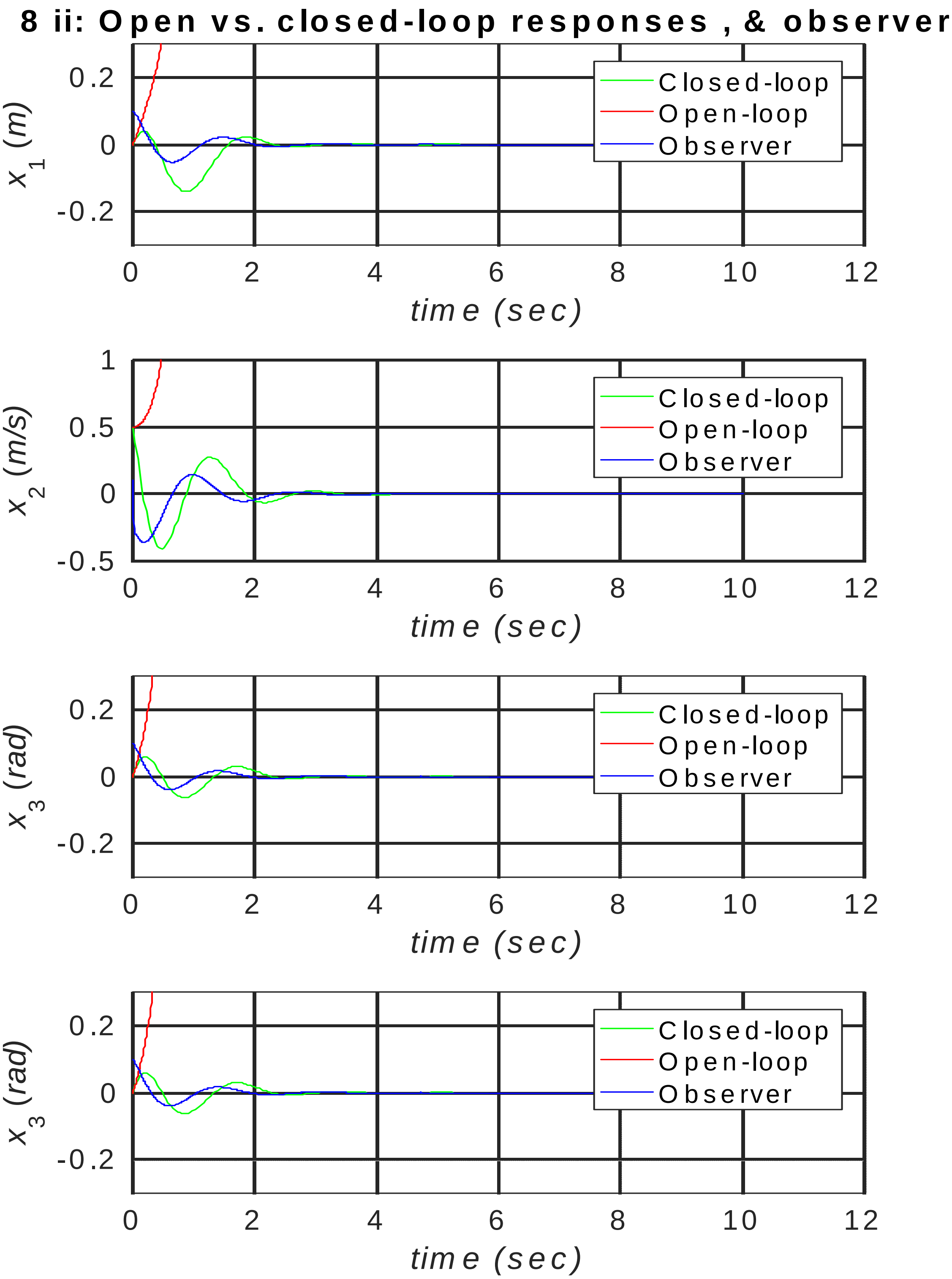
	u1
y1	**0**
y2	**0**

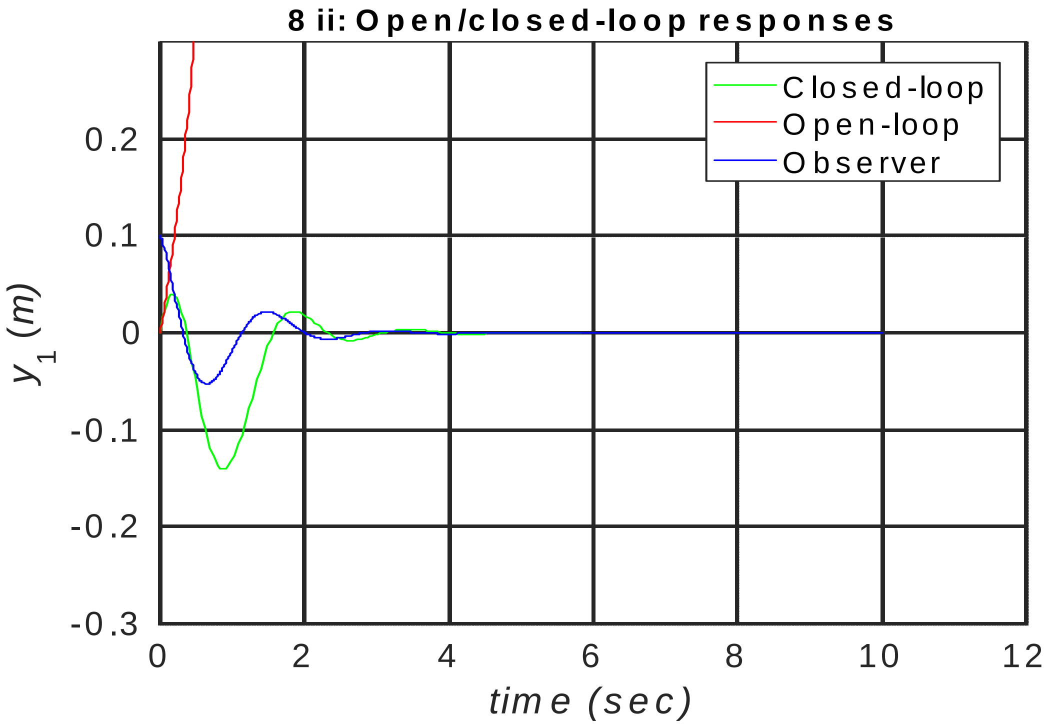
Continuous-time state-space model.

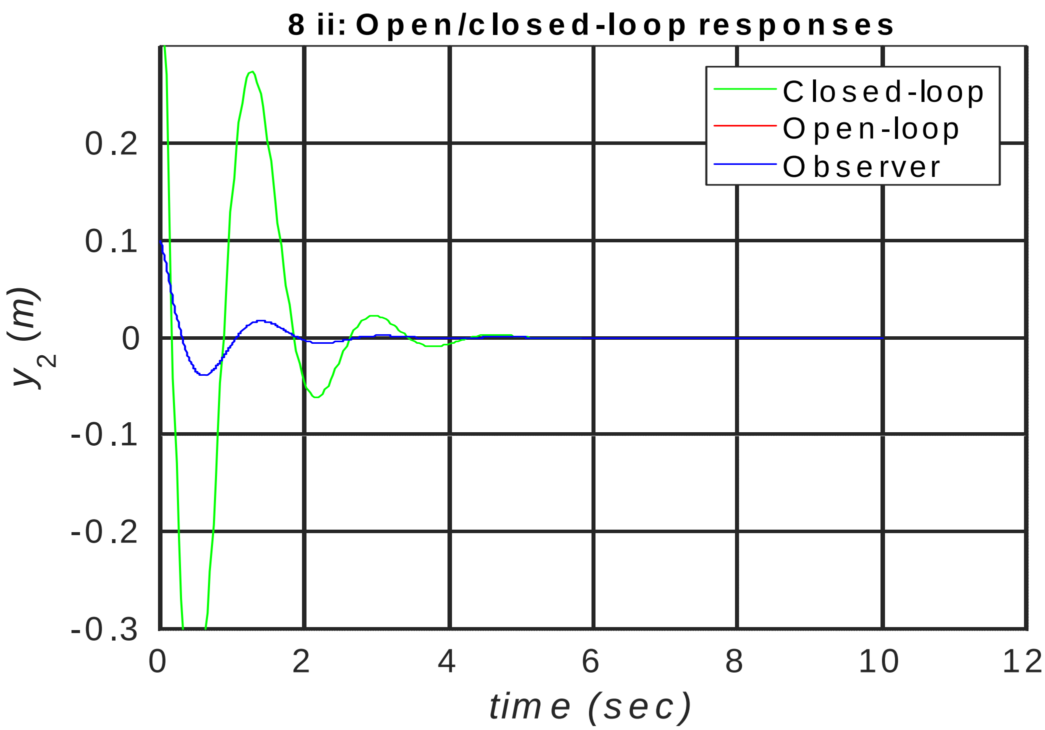
Ahat =

```
   1.0e+04 *

   -0.0315    0.0001    0.0254         0
   -7.1171         0    5.5228         0
   -0.0256         0   -0.0315    0.0001
   -5.5782         0   -7.0574         0

>>
```

Plots:


8 ii: Open vs. closed-loop responses , & observer
Closed-loop
Open-loop
Observer
x 1 (m)
x 2 (m/s)
x 3 (rad)
x 3 (rad)
0.2
0
-0.2
1
0.5
-0.5
0 2 4 6 8 10 12
time (sec)


8 ii: Open/closed-loop responses
Closed-loop
Open-loop
Observer
0.2
0.1
0
-0.1
-0.2
-0.3
y_1 (m)
0
2
4
6
8
10
12
time (sec)


8 ii: Open/closed-loop responses
Closed-loop
Open-loop
Observer
0.2
0.1
0
-0.1
-0.2
-0.3
y_2 (m)
0
2
4
6
8
10
12
time (sec)

Project part 8(iii):

code:

```
%Yevgen Solodkyy
%Project part 8 iii;


clear all; clc;

display('Design state feedback control laws for closed loop
eigenvalues')

clear all;
clc;
m1=2; m2=1; L=.75; g=9.81;

A = [0 1 0 0; 0 0 m2*g/m1 0; 0 0 0 1; 0 0 (m1+m2)*g/(m1*L) 0];
B_i = [0; 1/m1; 0; 1/(L*m1)];
B_iii = [0 0; 1/m1 1/(L*m1); 0 0; 1/m1*L (m2+m1)/(m1*m2*L^2)];  % need
to verify again the last value of B.
C = [0 0 1 0];
C_ii = [1 0 0 0;0 0 1 0]
C_iii= [1 0 0 0;0 0 1 0]
D=0;



%desired eigven values
l1= -1.27+3.79i;
l2= -1.27-3.79i;
l3= -1.88+1.24i;
l4= -1.88-1.24i;


DesEig = [l1; l2; l3; l4]



K_iii=place(A,B_iii,DesEig)

Sys_ol_iii = ss(A,B_iii,C_iii,D)
Ac_iii = A-B_iii*K_iii;
Sys_cl_iii = ss(Ac_iii,B_iii,C_iii,D)
```

```
[Yc,t,Xc] = step(Sys_cl_iii);
[YO,tO,XO] = step(Sys_ol_iii);



%% Observer portion


ObsEig = 100*DesEig; % observer eighen values 10x desired eigen values

%Q= [C; C*A; C*A^2; C*A^3];



L=place(A',C_iii',ObsEig)';
Ahat=A-L*C_iii  % declaring oberver state matrix







Ar = [(A-B_iii*K_iii) B_iii*K_iii;zeros(size(A)) (A-L*C_iii)];
Br = [B_iii;zeros(size(B_iii))];
Cr = [C_iii zeros(size(C_iii))];
Dr = D;

Sys_r = ss(Ar,Br,Cr,Dr);

tr = [0:0.01:10];

r = [zeros(size(tr));zeros(size(tr))];
Xr0 = [0.1;0.1;0.1;0.1;0.1;0.1;0.1;0.1]; %initial error conditions

%create vecros like these for the ope and closed loop systems as well.
see
%page 345 for reference

[Yr,tr,Xr] = lsim(Sys_r,r,tr,Xr0);



%%
figure(1)
subplot(411),
plot(t,Xc(:,1),'green',tO,XO(:,1),'red',tr,Xr(:,1),'blue')     %plot
(tO,YO(:1),'r',tc,Yc,'g',t,Yr,'b'); grid;
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_1 (\itm)');
```

```
legend('Closed-loop','Open-loop','Observer');
title('8 iii: Open vs. closed-loop responses , & observer, '); %
%legend('open loop','closee loop','w/ Observer');

subplot(412),
plot(t,Xc(:,2),'green',tO,XO(:,2),'red',tr,Xr(:,2),'b'); %plot(t,Xc
(:,2),'green',tO,XO(:,2),'red');
grid;
axis([0 12 -.5 1]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_2 (\itm/s)');
legend('Closed-loop','Open-loop','Observer');

subplot(413),
plot(t,Xc(:,3),'green',tO,XO(:,3),'red',tr,Xr(:,3),'b'); %plot(t,Xc
(:,3),'green',tO,XO(:,3),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_3 (\itrad)');
legend('Closed-loop','Open-loop','Observer');

subplot(414),
plot(t,Xc(:,3),'green',tO,XO(:,3),'red',tr,Xr(:,3),'b'); %plot(t,Xc
(:,3),'green',tO,XO(:,3),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\itx}_3 (\itrad)');
legend('Closed-loop','Open-loop','Observer');


figure(2)
subplot(211),
plot(t,Xc(:,1),'green',tO,XO(:,1),'red',tr,Yr(:,1),'b'); %plot(t,Yc
(:,1),'green',tO,YO(:,1),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\ity}_1 (\itm)');
legend('Closed-loop','Open-loop','Observer');
title('8 iii: Open/closed-loop responses', 'FontSize',14);

subplot(212),
plot(t,Xc(:,2),'green',tO,XO(:,2),'red',tr,Yr(:,2),'b');%plot(t,Yc
(:,2),'green',tO,YO(:,2),'red');
grid;
axis([0 12 -.3 .3]);
set(gca,'FontSize',16);
xlabel('\ittime (sec)');
ylabel('{\ity}_2 (\itm)');
```

```
legend('Closed-loop','Open-loop','Observer');
title('8 iii: Open/closed-loop responses', 'FontSize',14);
```

command output:

C_ii =

```
    1     0     0     0
    0     0     1     0
```

C_iii =

```
    1     0     0     0
    0     0     1     0
```

DesEig =

```
  -1.2700 + 3.7900i
  -1.2700 - 3.7900i
  -1.8800 + 1.2400i
  -1.8800 - 1.2400i
```

K_iii =

20.8408 9.3247 9.3042 4.3430

-5.0110 -2.2669 8.7096 0.5701

Sys_ol_iii =

A =

	x1	x2	x3	x4
x1	0	1	0	0
x2	0	0	4.905	0
x3	0	0	0	1
x4	0	0	19.62	0

B =

	u1	u2
x1	0	0
x2	0.5	0.6667
x3	0	0
x4	0.375	2.667

C =

	x1	x2	x3	x4
y1	1	0	0	0
y2	0	0	1	0

D =

	u1	u2
y1	0	0
y2	0	0

Continuous-time state-space model.

Sys_cl_iii =

A =

	x1	x2	x3	x4
x1	0	1	0	0
x2	-7.08	-3.151	-5.554	-2.552
x3	0	0	0	1
x4	5.547	2.548	-7.095	-3.149

B =

	u1	u2
x1	0	0
x2	0.5	0.6667
x3	0	0
x4	0.375	2.667

C =

	x1	x2	x3	x4
y1	1	0	0	0

y2 0 0 1 0

D =

	u1	**u2**
y1	**0**	**0**
y2	**0**	**0**

Continuous-time state-space model.

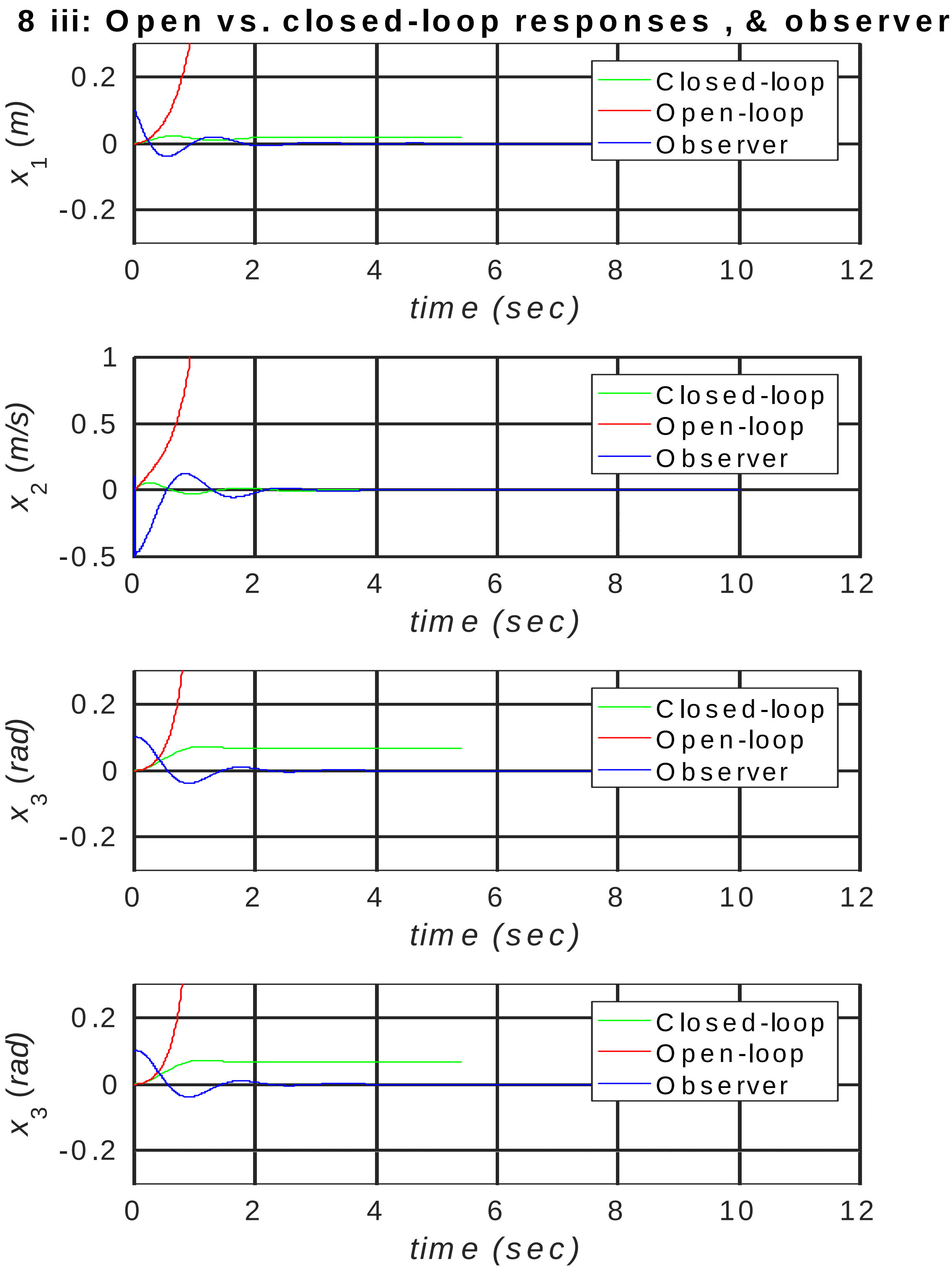
Ahat =

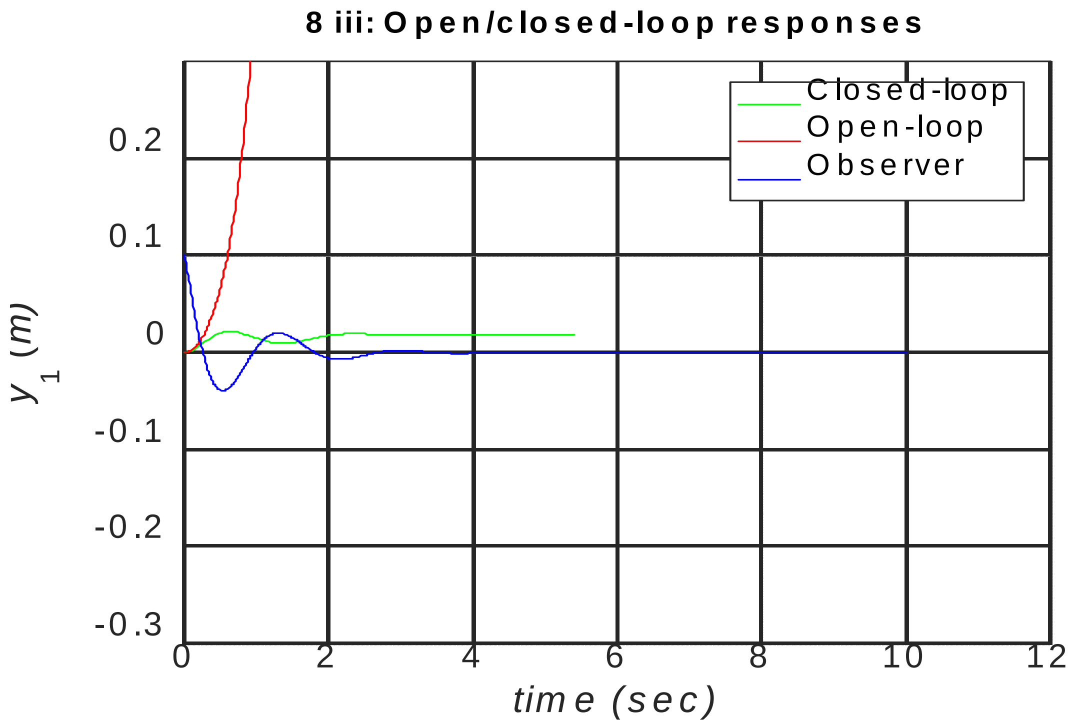
1.0e+04 *

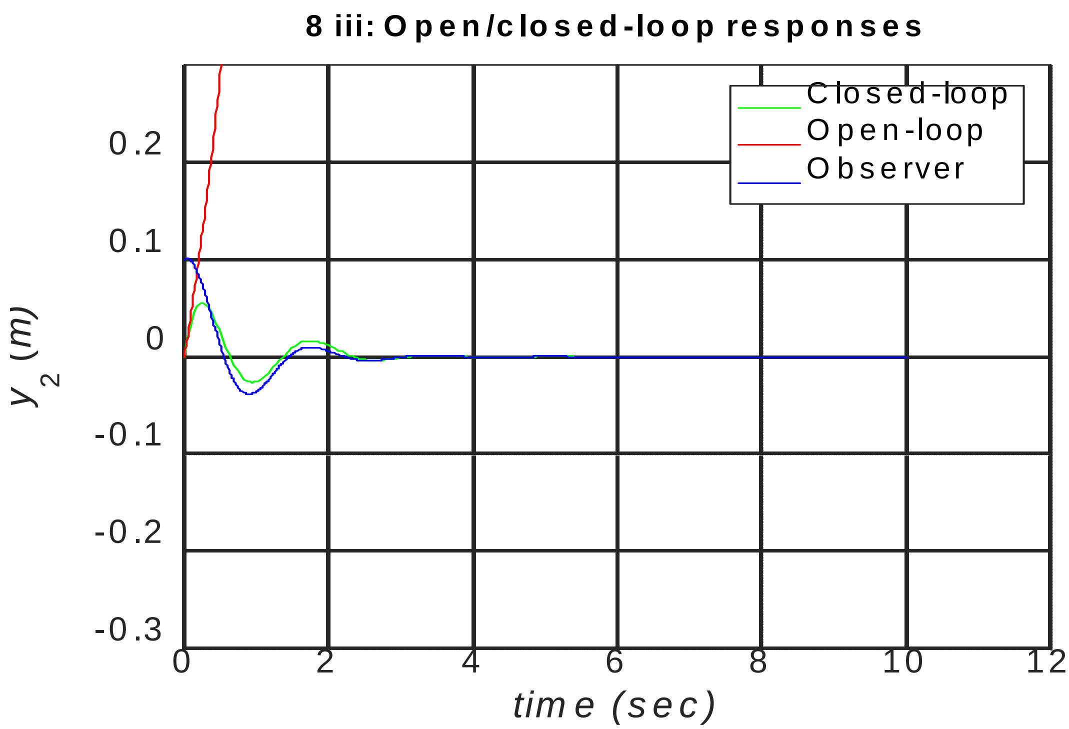
-0.0315	**0.0001**	**0.0254**	**0**
-7.1171	**0**	**5.5228**	**0**
-0.0256	**0**	**-0.0315**	**0.0001**
-5.5782	**0**	**-7.0574**	**0**

>>

PLots:


8 iii: Open vs. closed-loop responses , & observer
x1 (m)
x2 (m/s)
x3 (rad)
x3 (rad)
time (sec)
0.2
0
-0.2
1
0.5
-0.5
0
2
4
6
8
10
12
Closed-loop
Open-loop
Observer


8 iii: Open/closed-loop responses
Closed-loop
Open-loop
Observer
0.2
0.1
0
-0.1
-0.2
-0.3
y_1 (m)
0
2
4
6
8
10
12
time (sec)


8 iii: Open/closed-loop responses
Closed-loop
Open-loop
Observer
0.2
0.1
0
-0.1
-0.2
-0.3
y_2 (m)
0
2
4
6
8
10
12
time (sec)