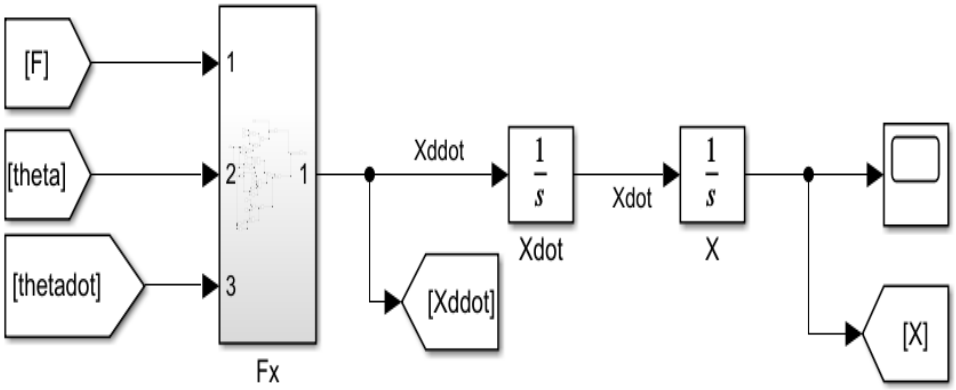
CUFE
Cairo university –
Faculty of engineering

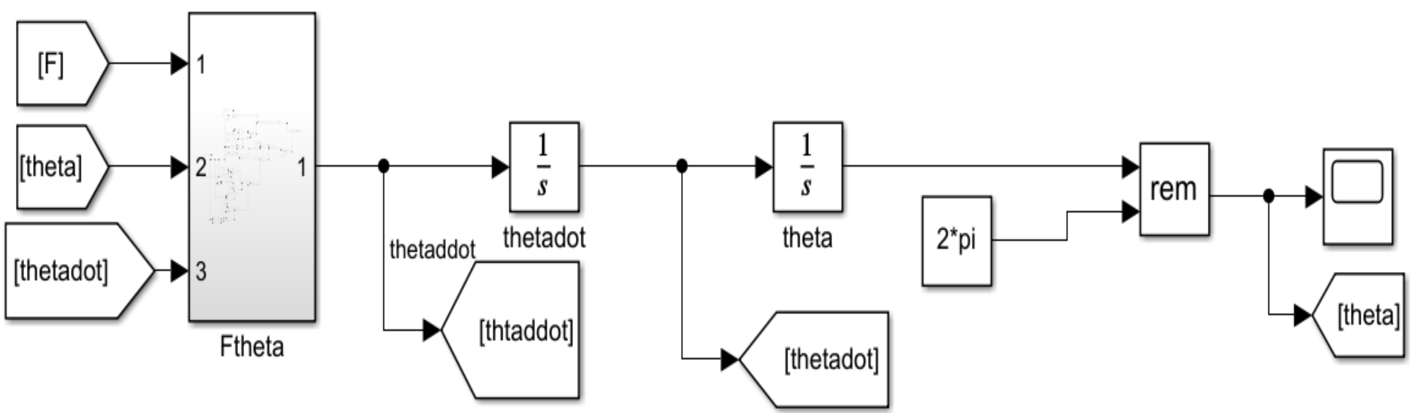
Inverted Pendulum

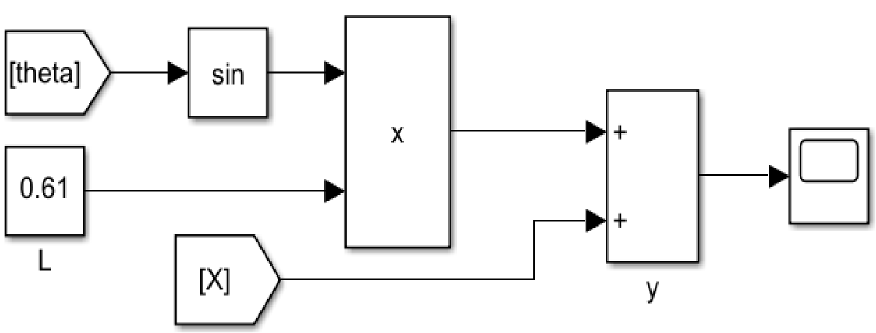
B.N	Sec	Name
10	2	خالد طارق عاطف
11	2	خالد محمد امام

Continuous System Analysis and Control Design Using the Transfer Function Approach:

1- Model the nonlinear system. Plot θ and x assuming the equilibrium conditions stated in (6) are satisfied. Repeat the simulation and the plotting assuming the initial conditions given in (7).



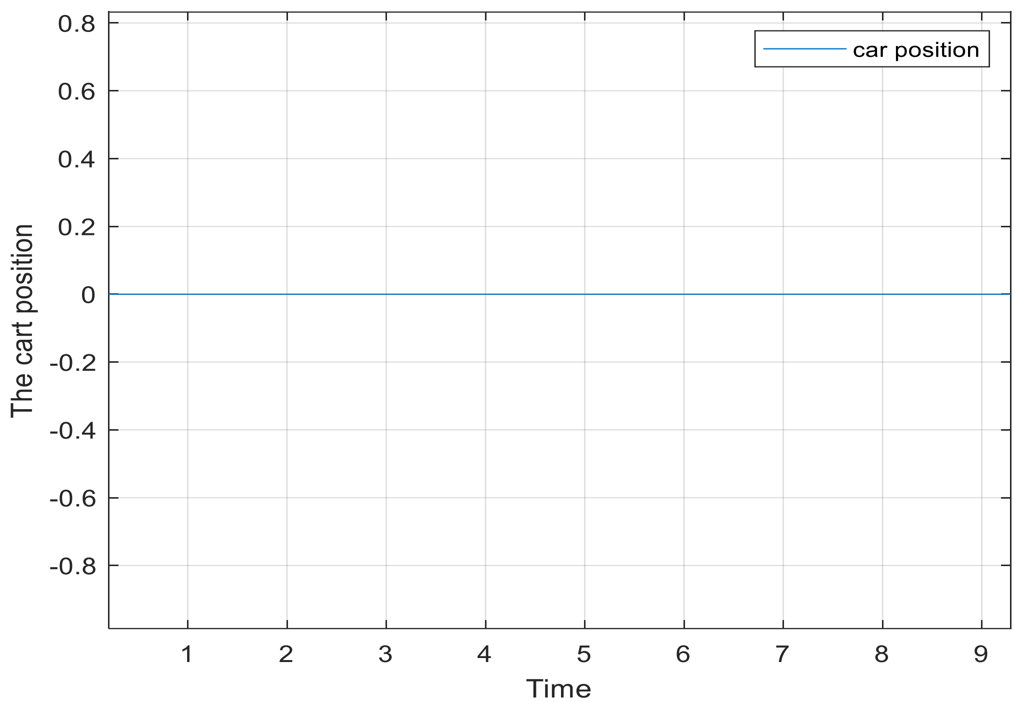


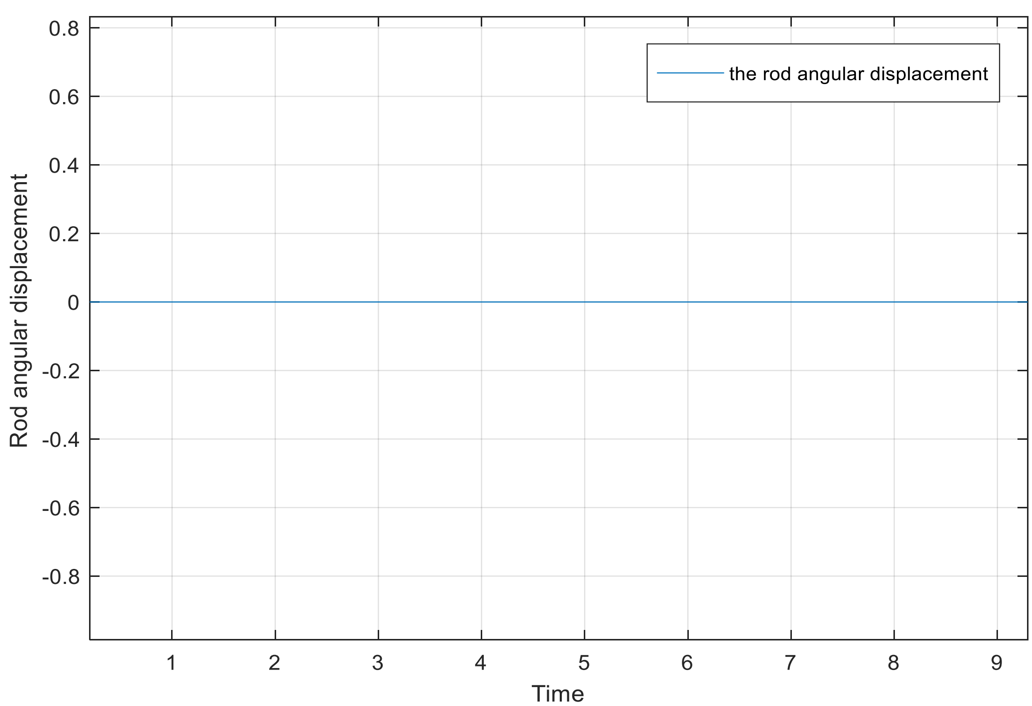


- Condition 1 : The equilibrium point:

  $x = 0\,m \qquad \dot{x} = 0\ m/sec$

  $\theta = 0\,rad \qquad \dot{\theta} = 0\,rad/sec$

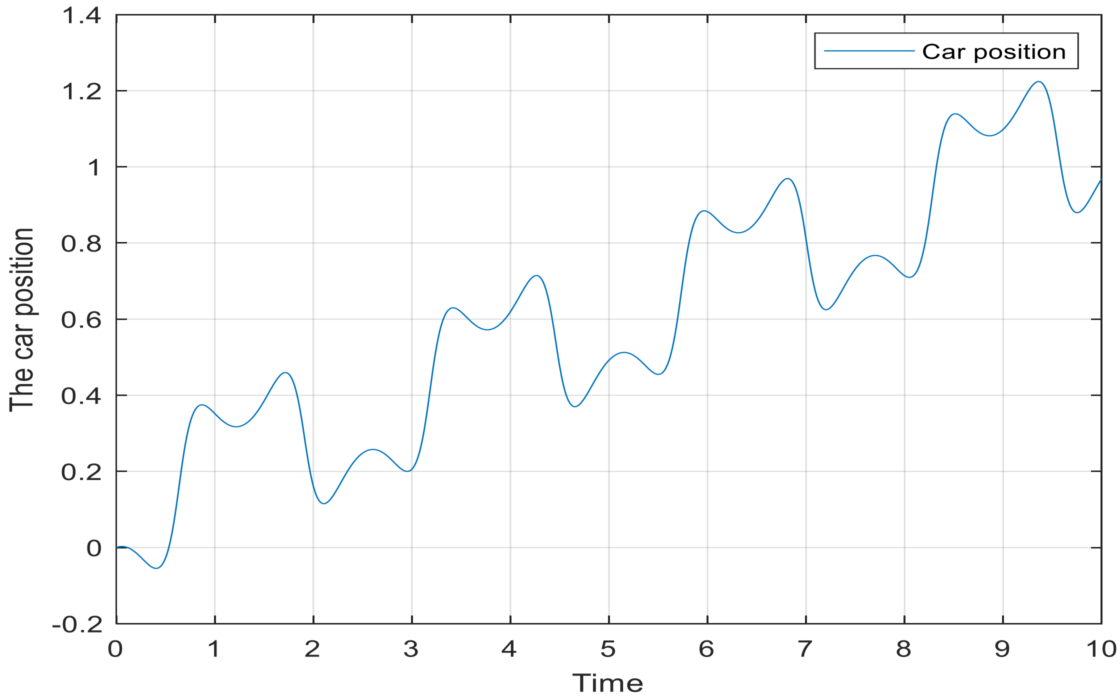


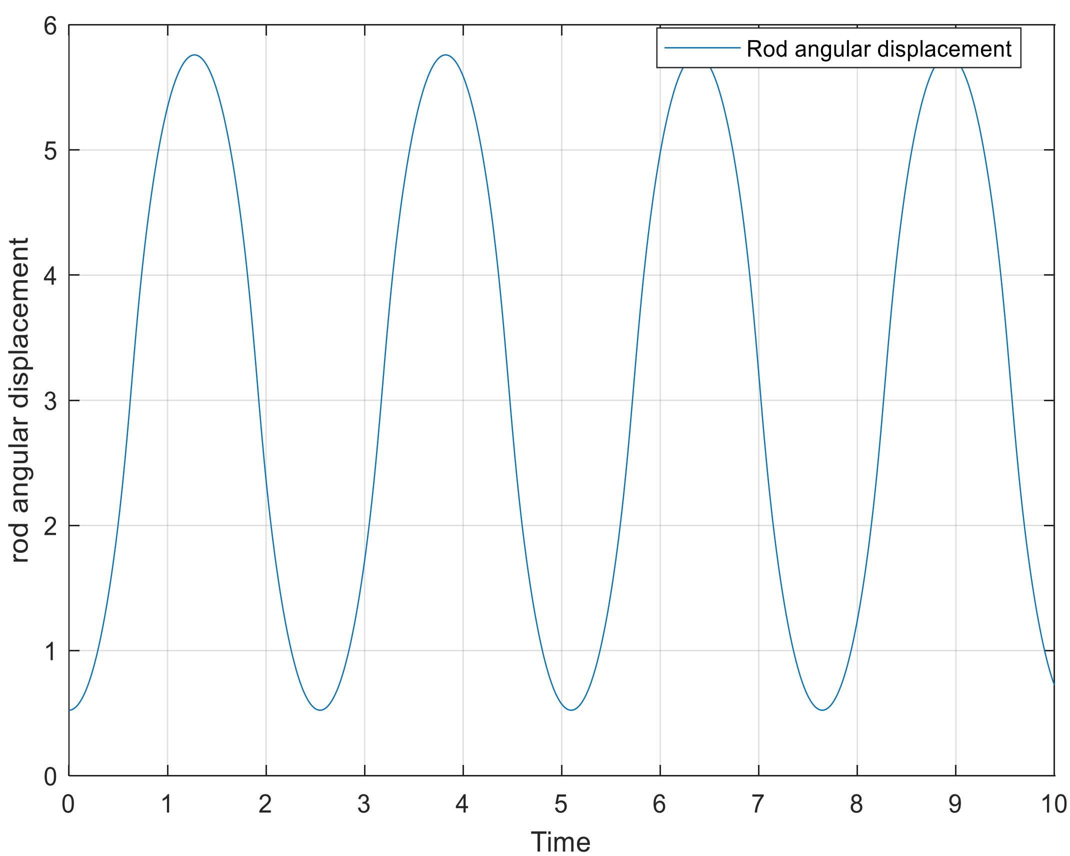


- Condition 2 : The deviated initial condition:

$x = 0\,m \qquad \dot{x} = 0.1\ m/sec$

$\theta = \pi/6\,rad \qquad \dot{\theta} = 0\,rad/sec$





2- Starting from the nonlinear model, find the linearized transfer function $\Delta\theta/\Delta F$ using the equilibrium point condition in (6). Calculate the system poles and zeros. Plot the system root-locus.

Linearize and getting the transfer function using MATLAB

code

```
clear
clc
sys=linearize('pendulum',getlinio('pendulum'));
[K1 , K2]=ss2tf(sys.A,sys.B,sys.C,sys.D);
K1=round(K1,4);
K2=round(K2,4);
Dtheta_DF=tf(K1,K2);
rlocus=rlocusplot(Dtheta_DF);
```

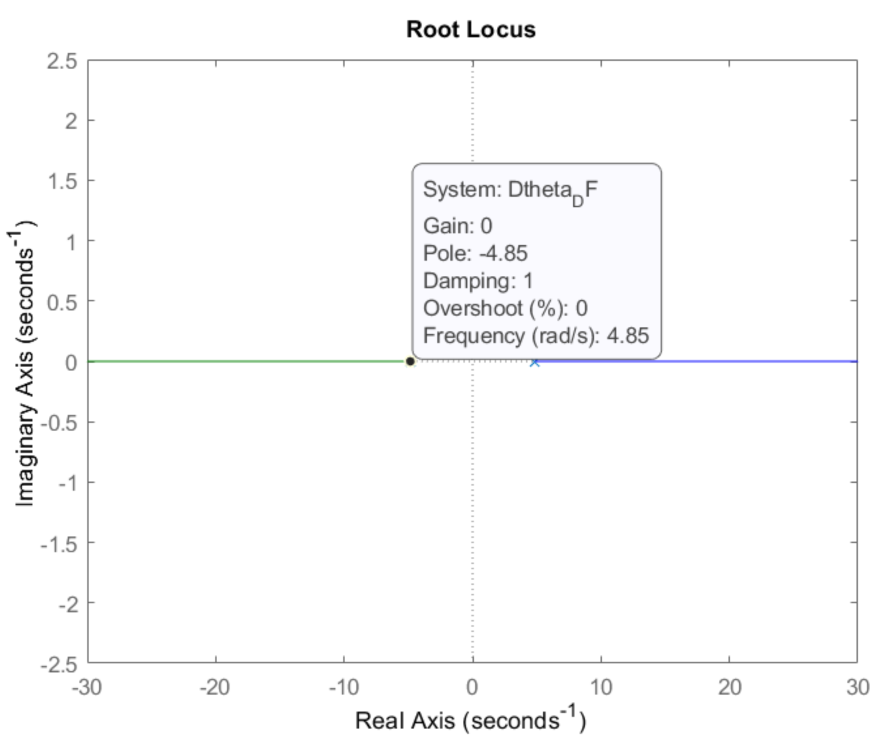
Transfer function

-3.603

---------- poles at (±4.85), no zeros

s^2 - 23.5

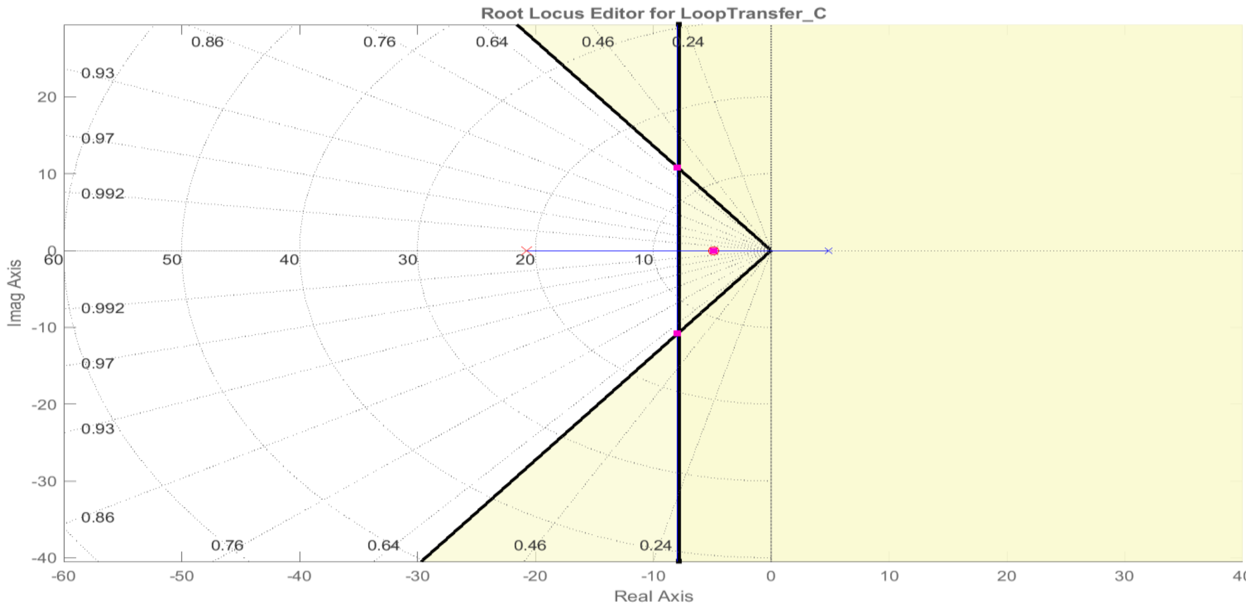
Root locus

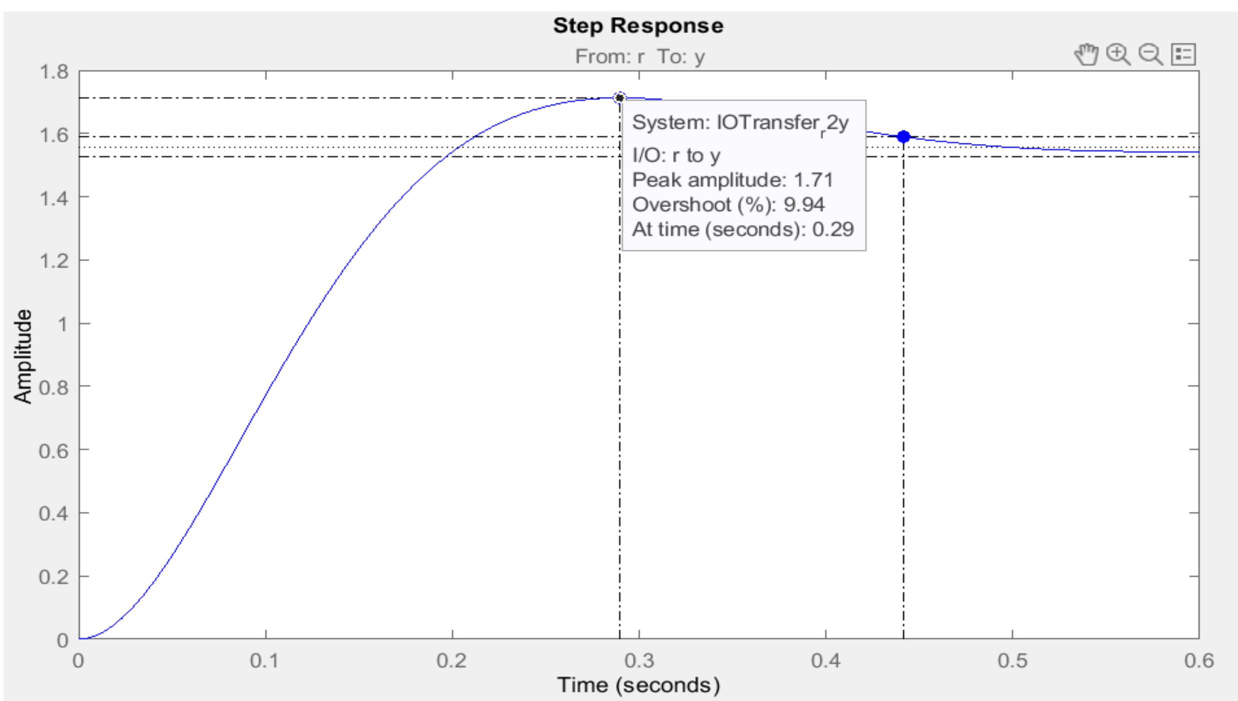


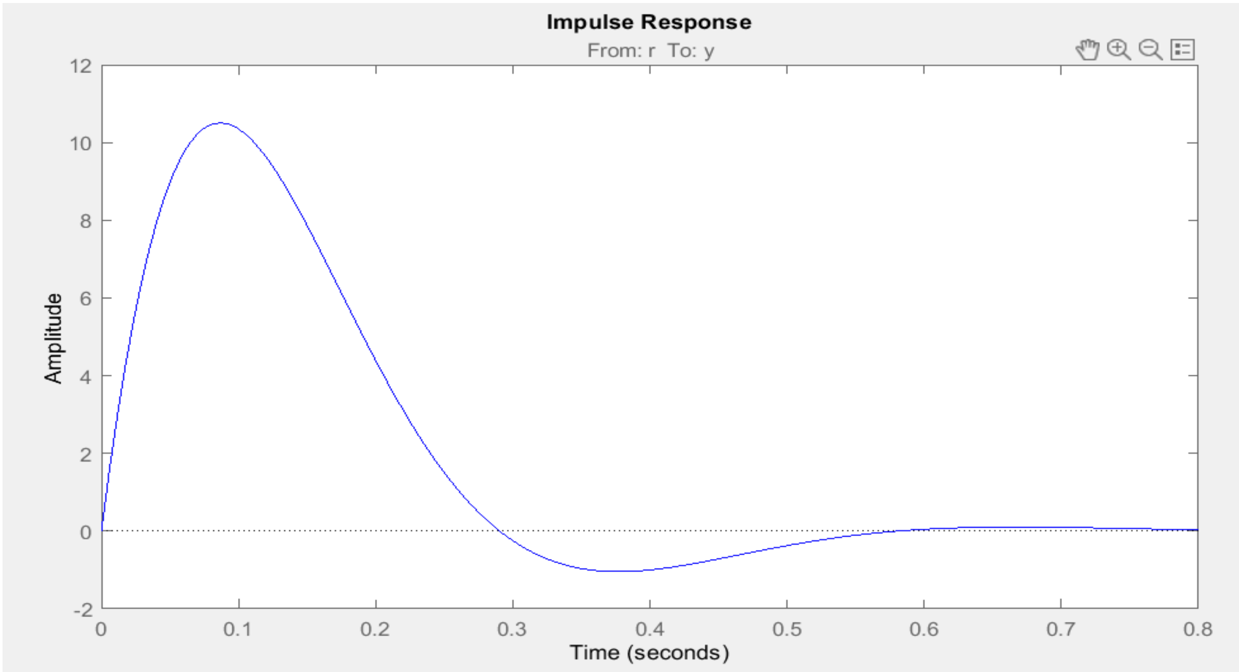
3- Design a lead compensator to achieve a settling time less than 0.5 sec and the maximum overshoot to be less than 10%. Plot the root-locus after compensation. Specify the closed loop poles. Plot the impulse response. Comment on the results.

```
C =

  77.955 (s+4.848)
  ----------------
     (s+20.75)
```

The closed loop poles are -7.95±10.8j



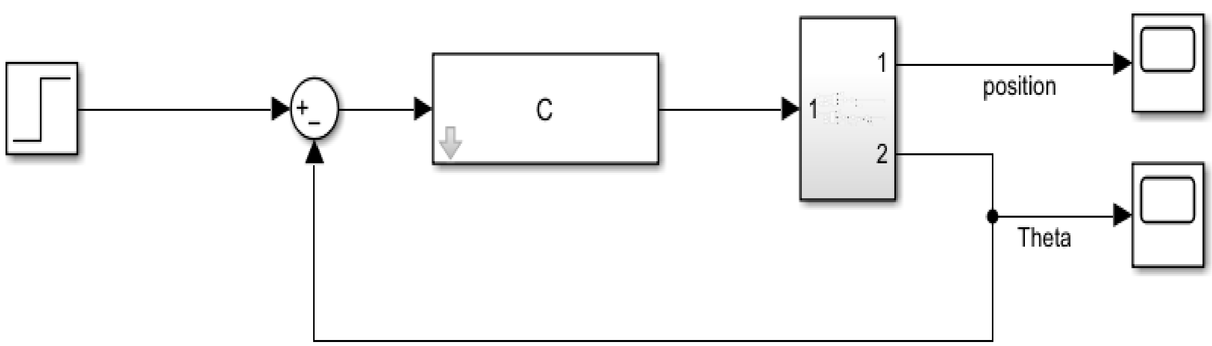




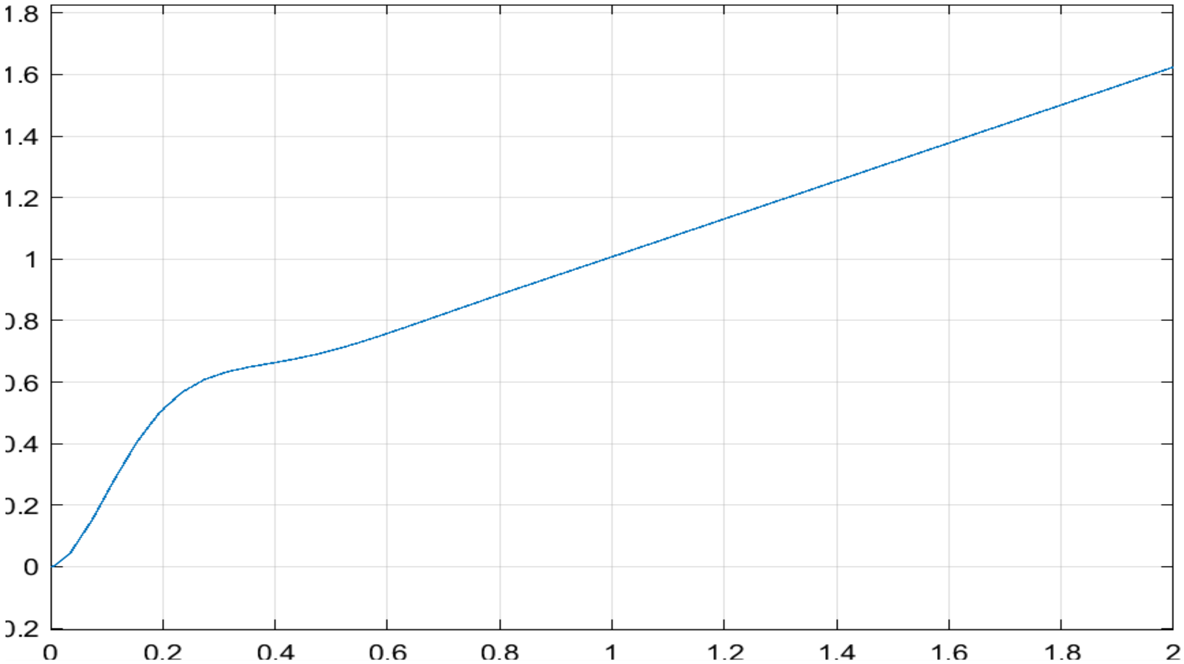
Comment:

We can achieve the required over shoot and settling time using lead compensator.

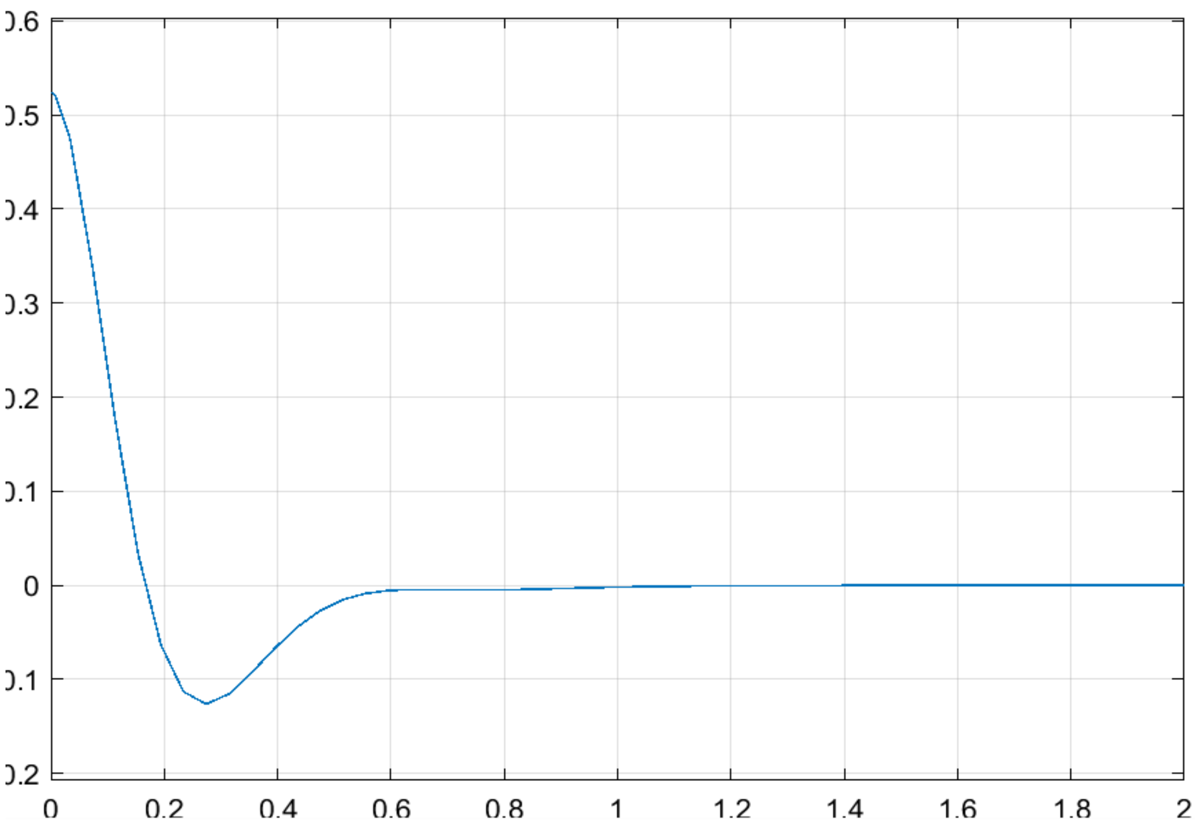
4- Simulate the nonlinear system with the lead compensator. Plot θ and x assuming the initial condition in (7).



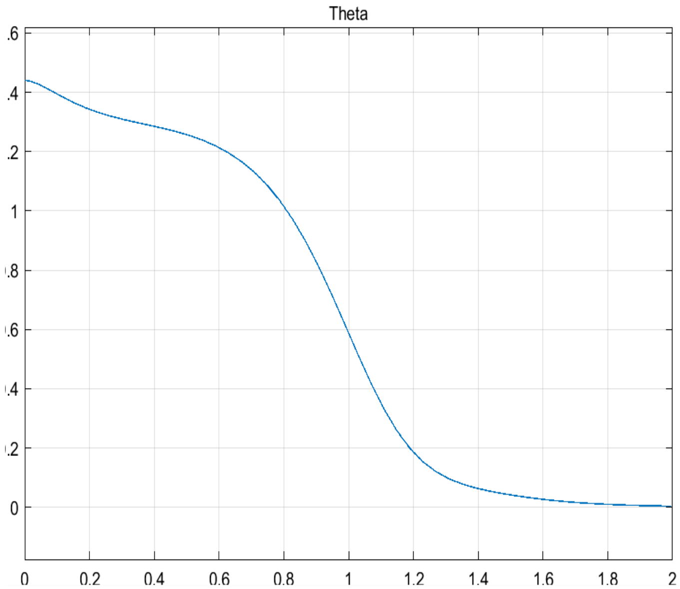
position



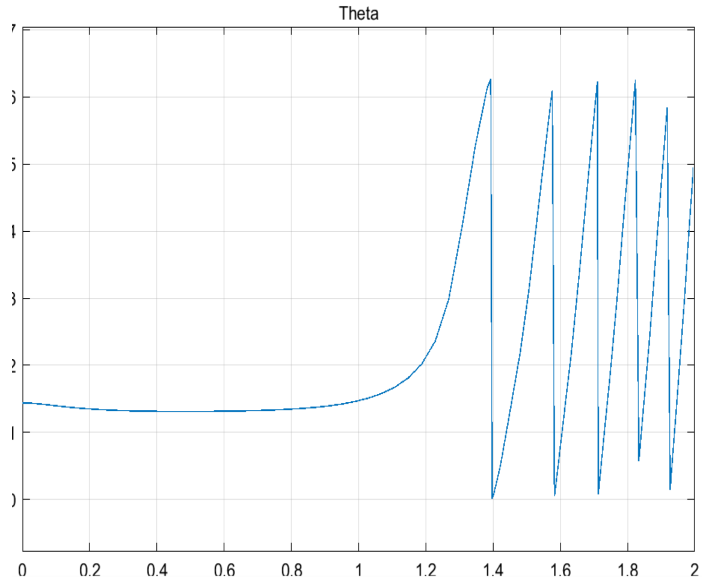
Theta



5- Using the lead compensator developed above, test the maximum deviations that the controller can deal with and keep the pendulum vertically up.



Response at theta initial=82.5 degree



Response at theta initial=82.6 degree

So, the maximum deviations that the controller can deal with and keep the pendulum vertically up is 82.6 degree.

<u>Digital Control Design Using the Transfer Function Approach:</u>

6- Select an appropriate sampling interval and design a digital controller (using the emulation technique) to stabilize the system and ensure a settling time less than 0.5 seconds. Apply the digital controller to the nonlinear model using the initial condition in (7). Plot θ, x and the control signal.

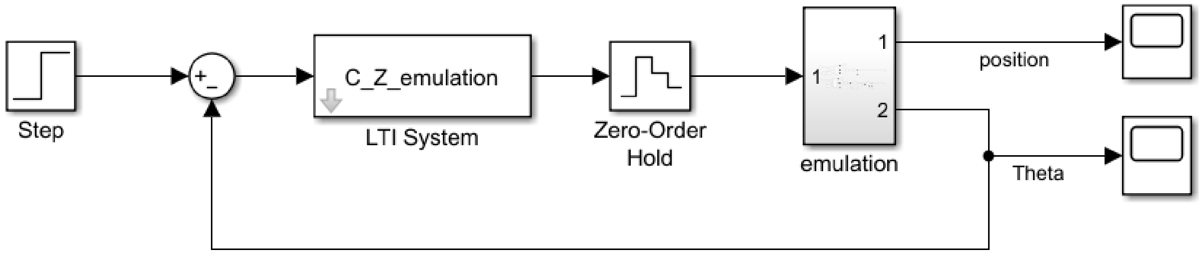
Taking the sampling interval 10 times from rising time

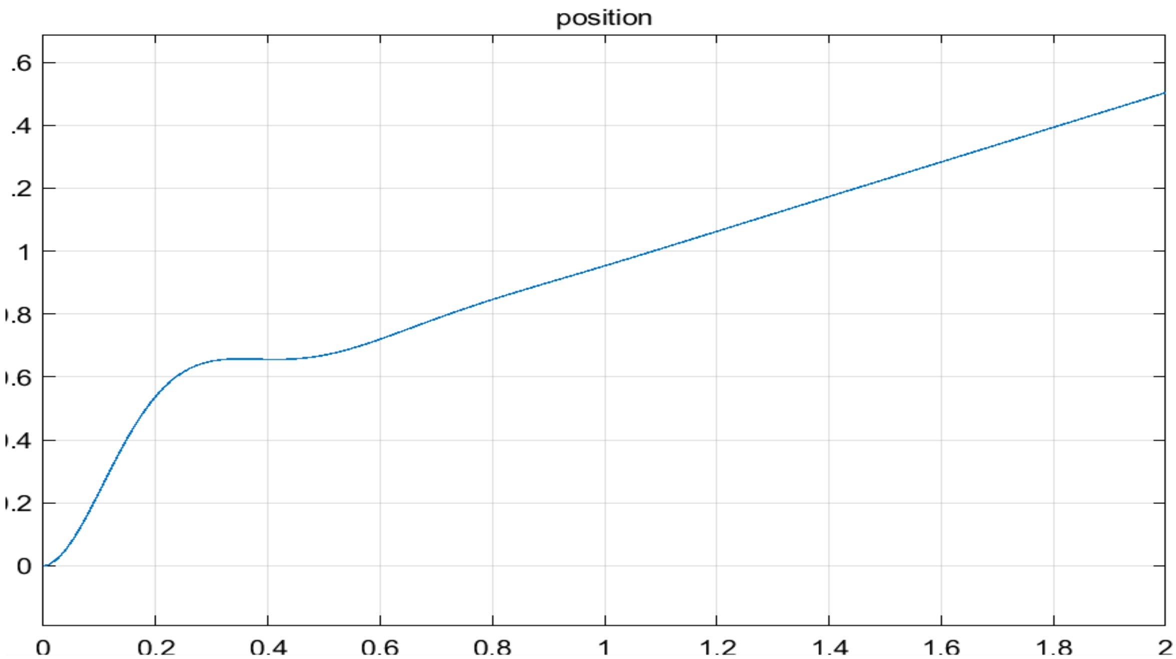
T_s=0.0137 s

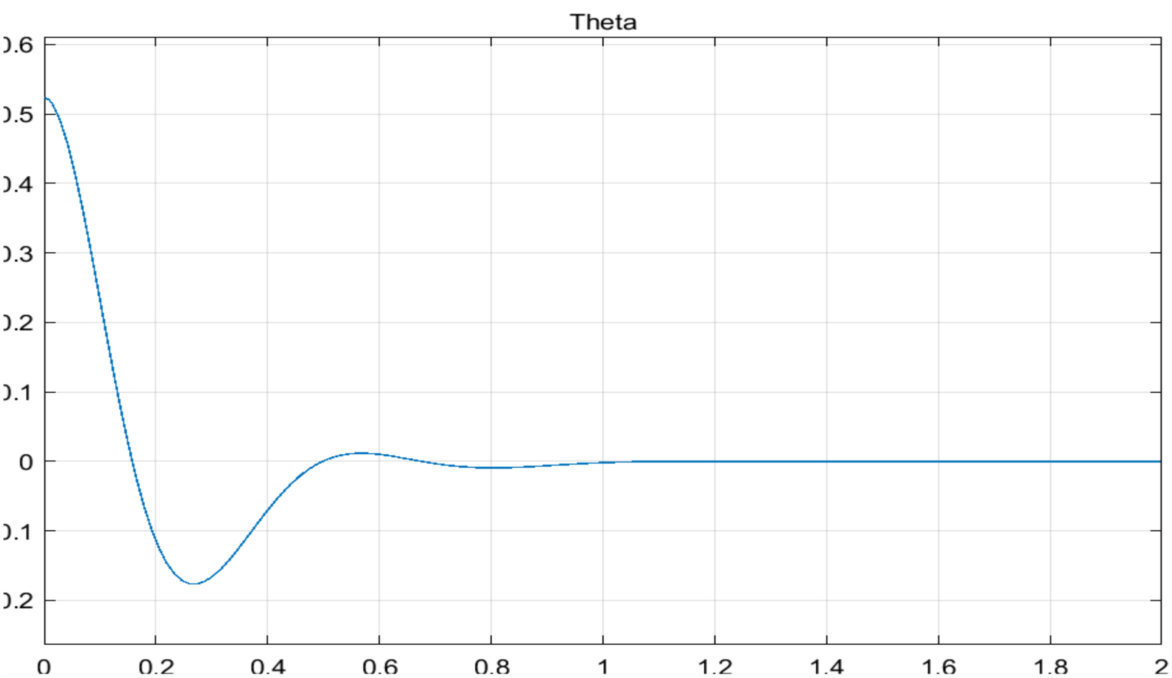
```
C_Z_emulation =

  -64.728 (z-0.9377)
  ------------------
      (z-0.7786)
```

```
a=0.937699;  %Backward emulation done manually
b=0.7786;
k=-64.72834;
C_Z_emulation=zpk(a,b,k,0.0137);
```

```
Sample time: 0.0137 seconds
Discrete-time zero/pole/gain model.
```







Comment:

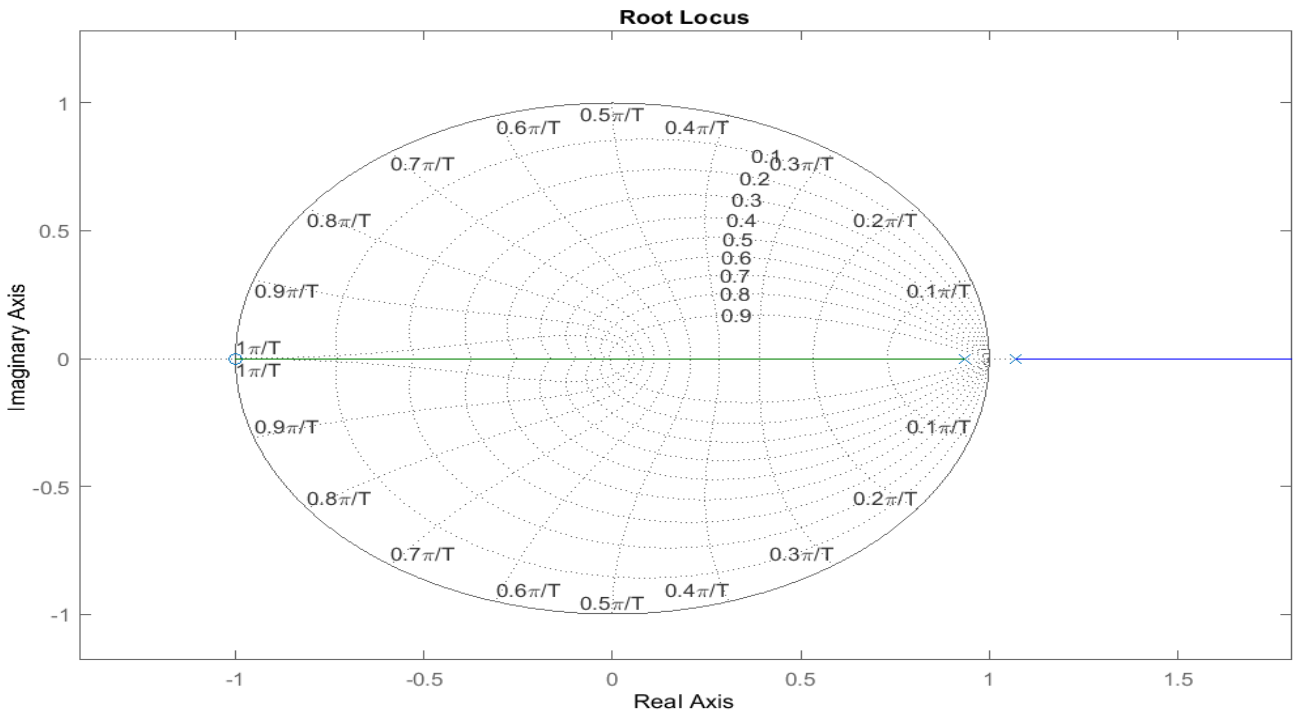
In emulation technique the settling time and overshoot is greater than in continuous technique.

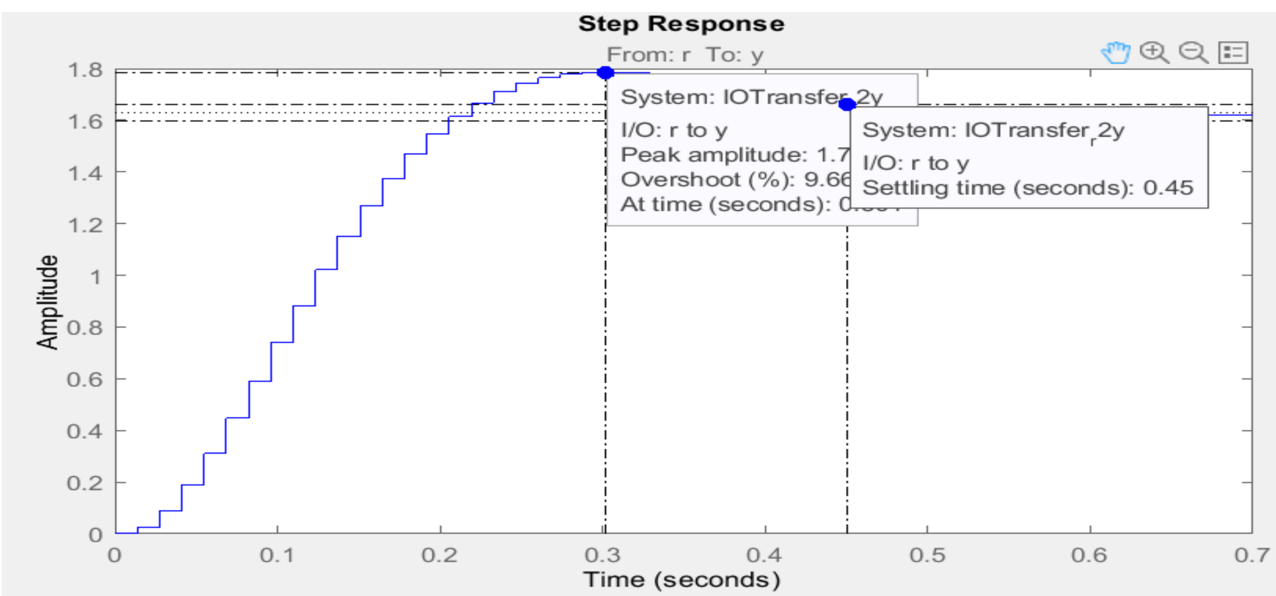
7- Repeat in question #6 using direct digital control design. Compare the results from 6 and 7. Comment on the results.

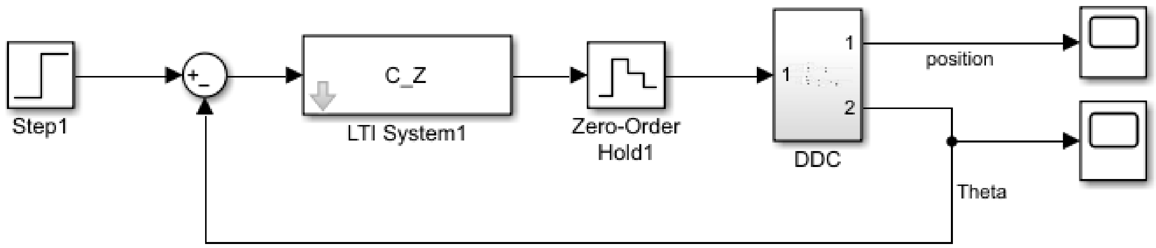
T_s=0.0137 s

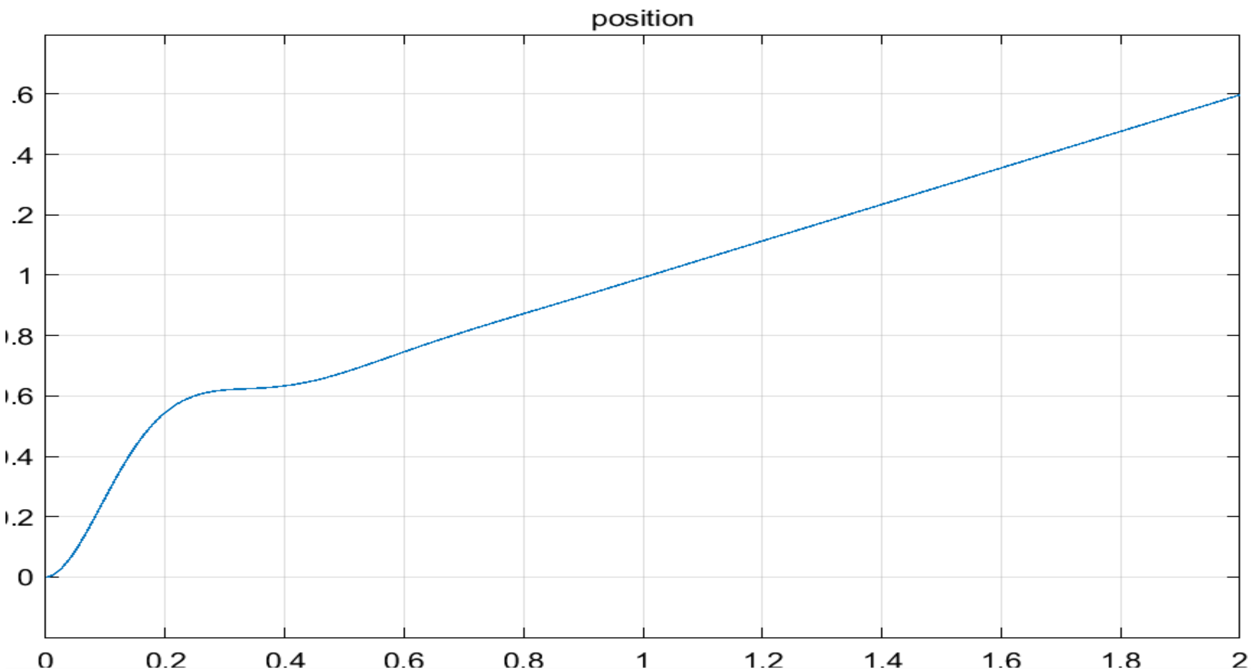
closed loop Poles: 0.963, 0.888±0.13j

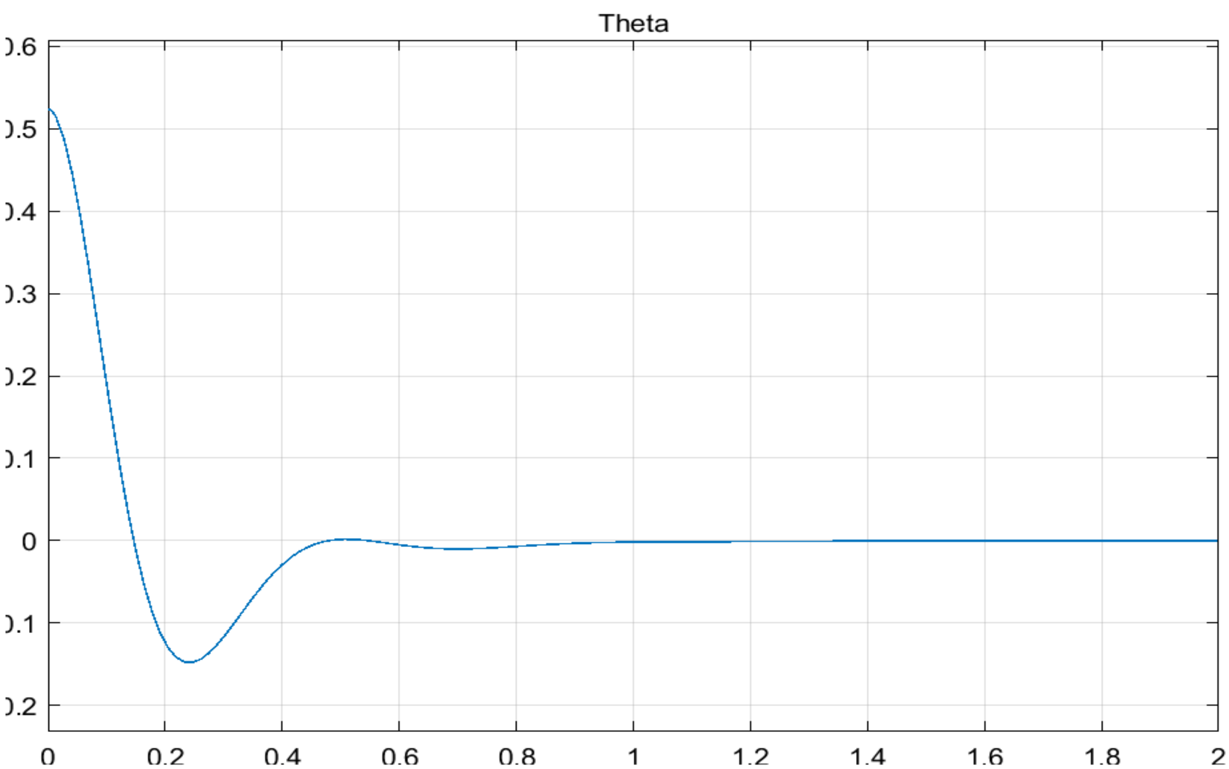
```
G_Z =

  -0.0003382 z - 0.0003382
  ------------------------
     z^2 - 2.004 z + 1
```






Step1
C_Z
LTI System1
Zero-Order
Hold1
DDC
1
1
2
position
Theta


position
.6
.4
.2
1
).8
).6
).4
).2
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


Theta
).6
).5
).4
).3
).2
).1
0
).1
).2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

Comment:

We notice that the over shoot and settling time in emulation technique is greater than in DDC technique.

Continuous State Space Representation

8- Using the system dynamics in (3), Choose appropriate signals to be the states of the system and write the corresponding linear state space representation (Call this representation “rep A”).

$$x_1 = x \quad x_2 = \dot{x} \quad x_3 = \theta \quad x_4 = \dot{\theta}$$

```
sys=linearize ('pendulem',getlinio('pendulem'));
```

Continuous-time state-space model.

A =

	X	theta	thetadot	Xdot
X	0	0	0	1
theta	0	0	1	0
thetadot	0	23.5	0	0
Xdot	0	-4.528	0	0

B =

	Add
X	0
theta	0
thetadot	-3.603
Xdot	2.198

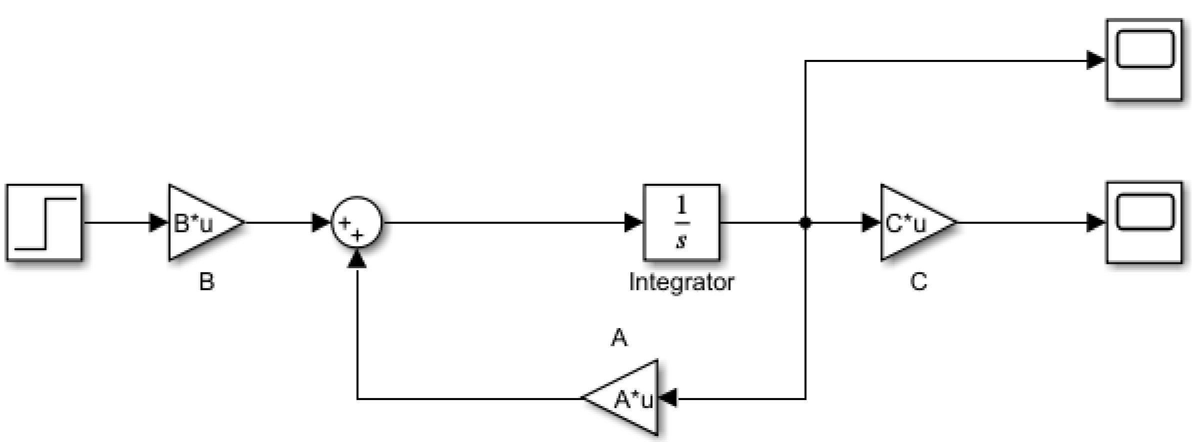
C =

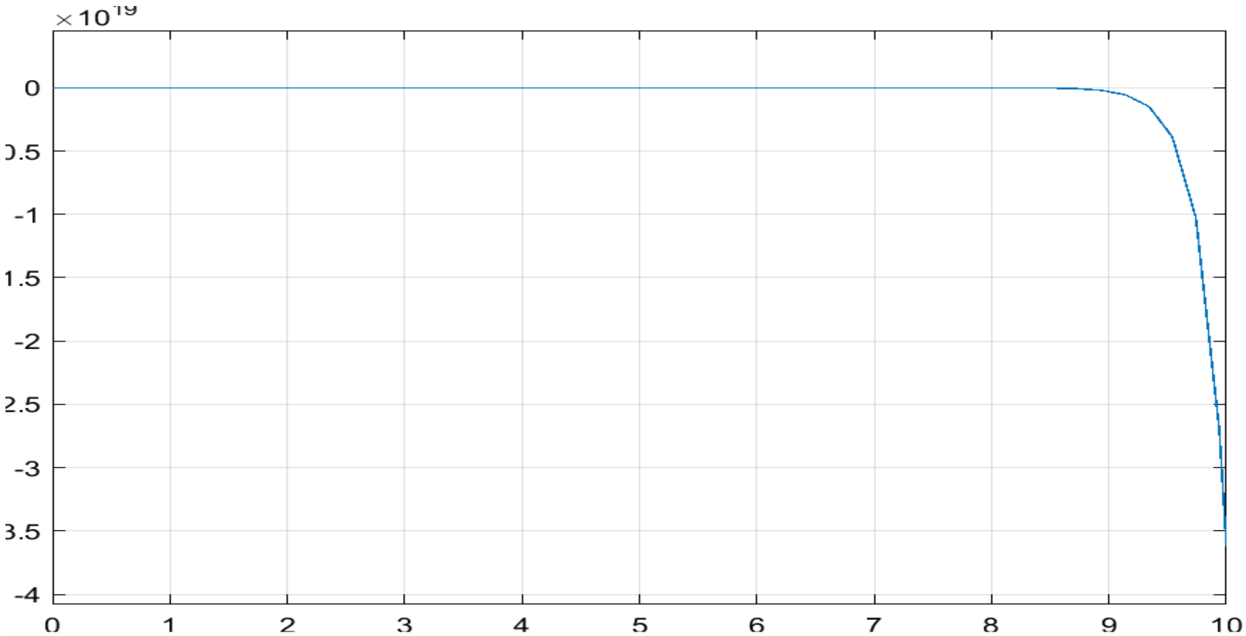
	X	theta	thetadot	Xdot
y	1	0.61	0	0

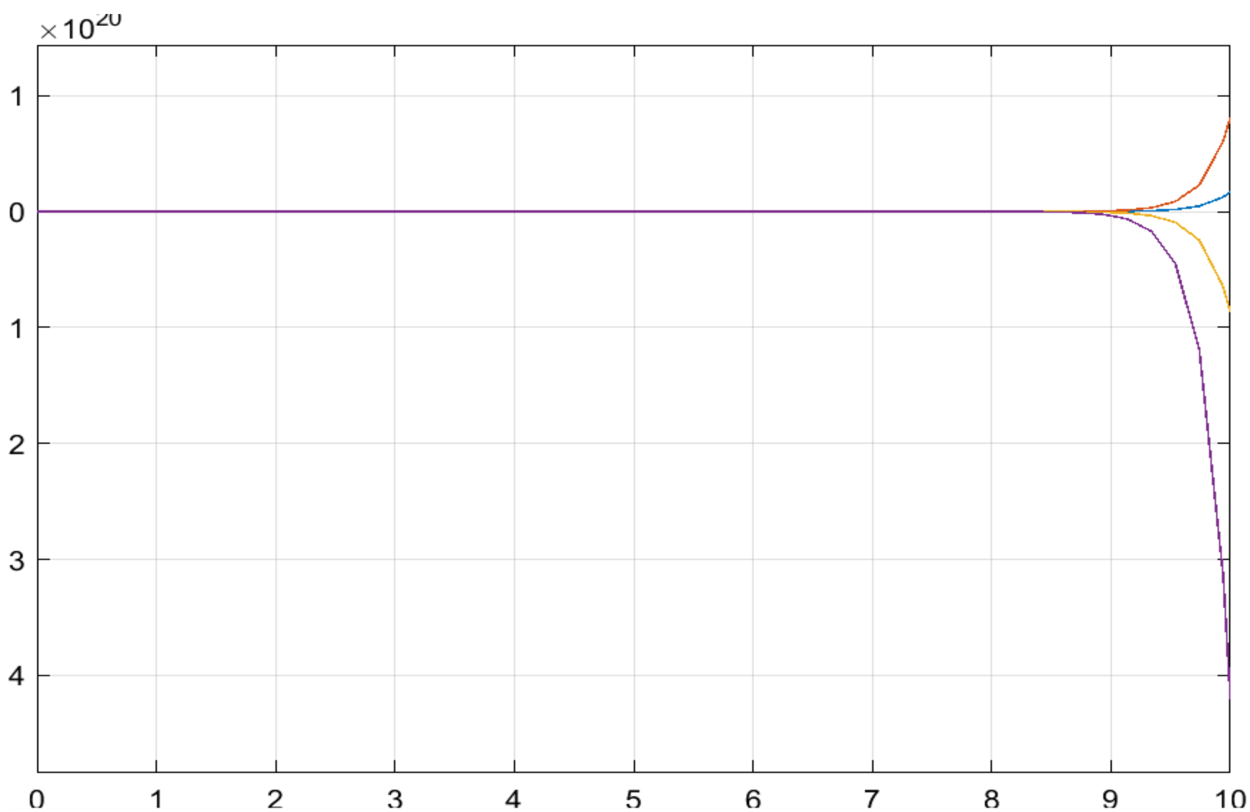
D =

	Add
y	0

9- Develop a Simulink model for “rep A” and draw XA(t) and y(t) versus t as the system starts at condition in 2. Compare between the results of “rep A” and the nonlinear system obtained in 2. Comment on the results.







Comment:

The output Y and states are unstable. The difference between the results of “rep A” and the nonlinear system obtained is fast response due to linearization.

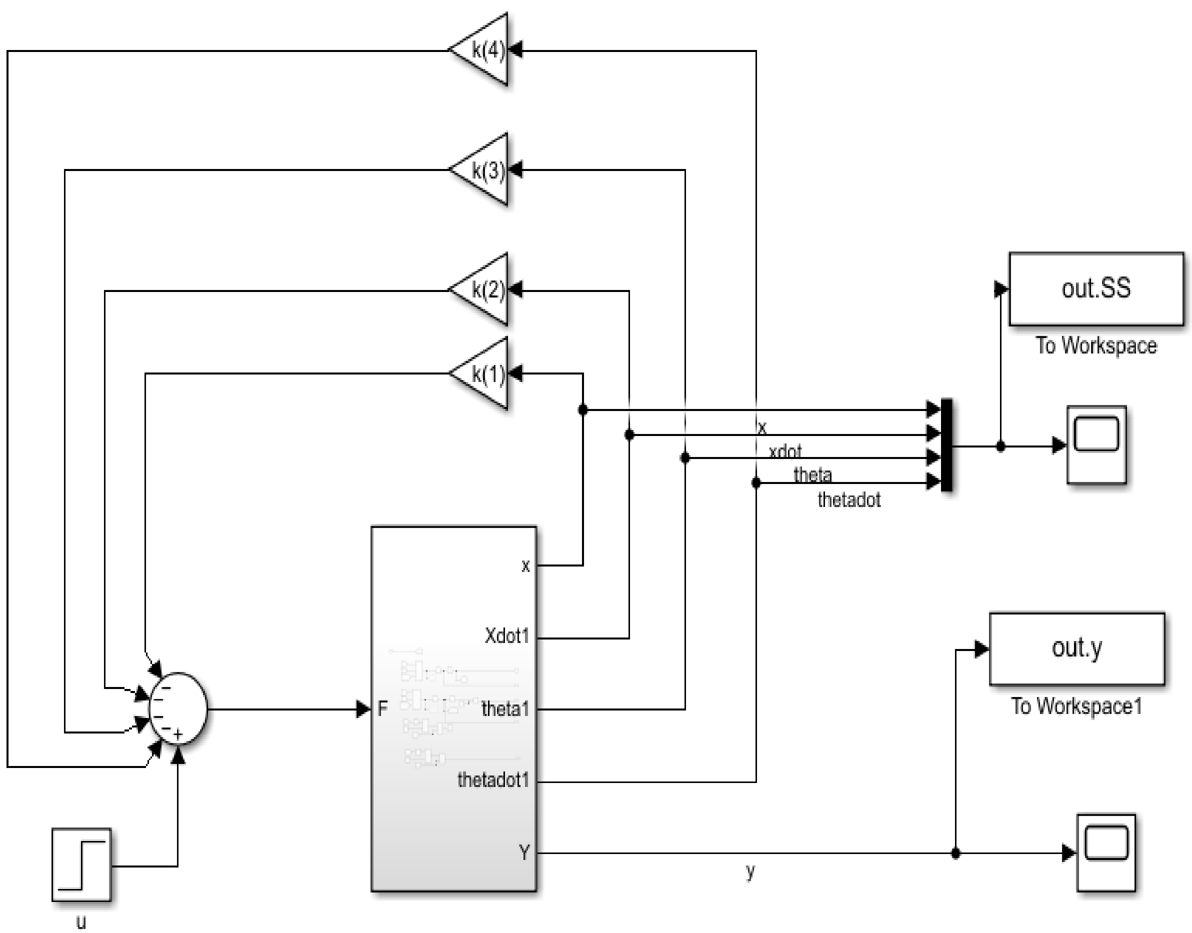
10- Find appropriate state feedback vector to place the poles of the system in suitable places (You can use the same requirements as mentioned in the transfer function approach). Implement the feedback signals using the nonlinear system (same initial conditions as in (9)) and draw the new states and outputs versus time. Comment on the results.

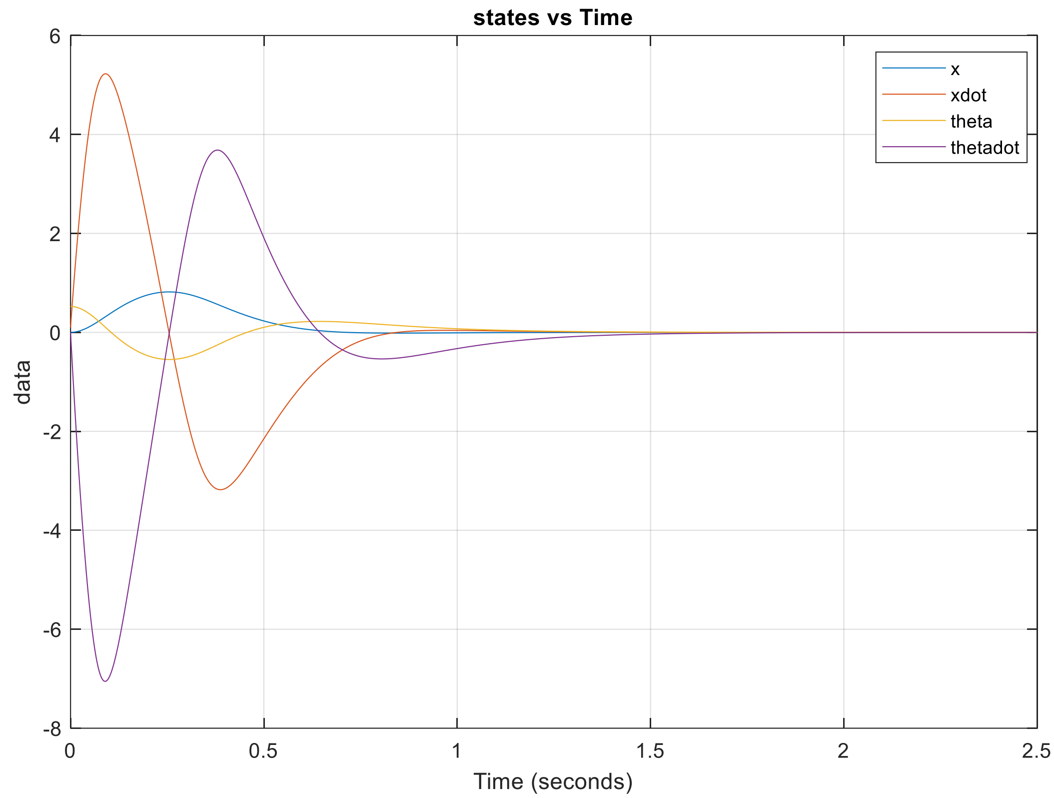
Let desired poles = [-4 -6 -7 -9]

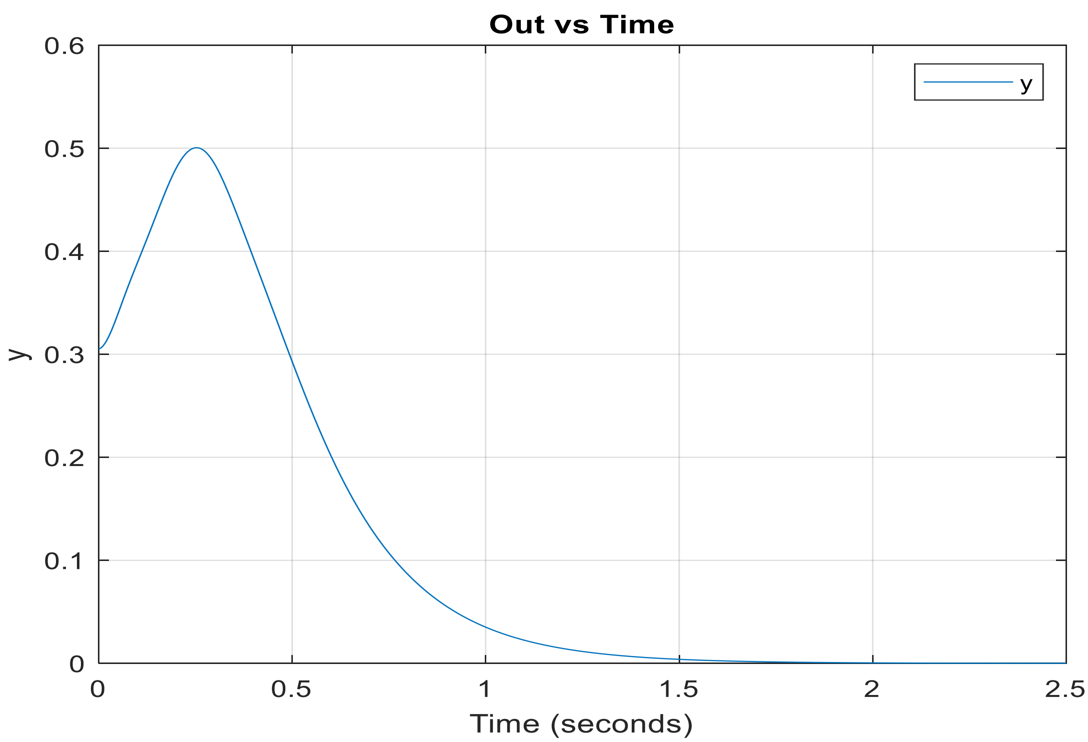
code

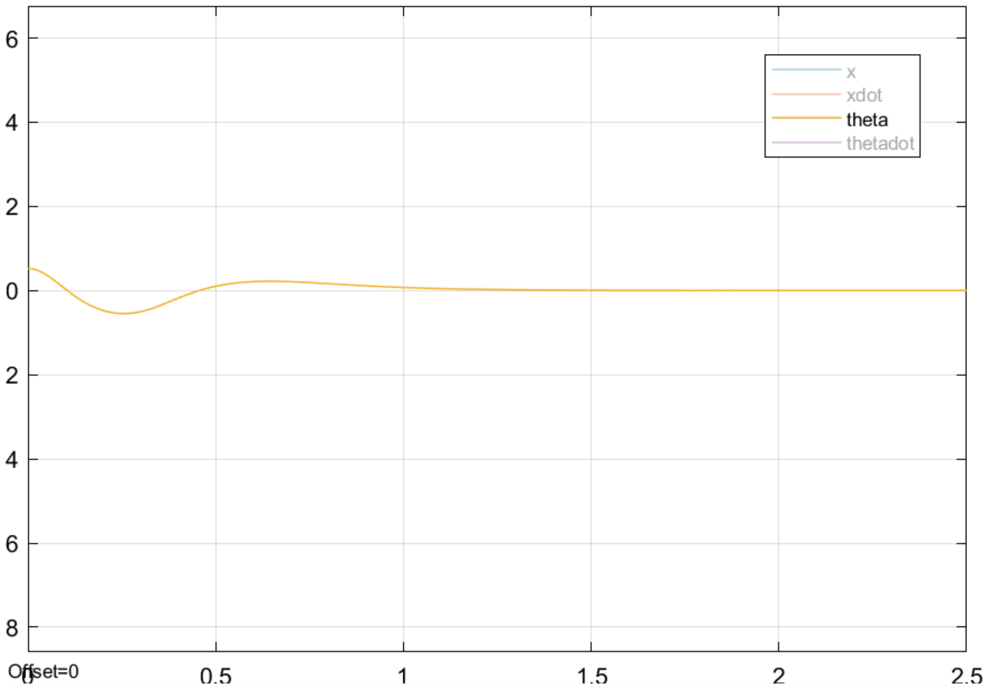
```
sys=linearize ('pendulem',getlinio('pendulem'));
A=sys.A;
B=sys.B;
C=sys.C;
D=sys.D;
repA=ss(A,B,C,D);
p=[-4 -6 -7 -9];
k=place(A,B,p);
```

```
k =

  -42.7861  -28.6938 -101.1778  -24.7208
```




states vs Time
x
xdot
theta
thetadot
data
6
4
2
0
-2
-4
-6
-8
0
0.5
1
1.5
2
2.5
Time (seconds)


Out vs Time
y
y
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
Time (seconds)



Comment:

The system stable and there is no over shoot and as shown in the above fig. the settling time is around 0.5 sec.

11-Choose a suitable sampling period and find the discrete form for “rep A” (Call this representation “rep B”). Write the state space representation in the controllable canonical form (Call this representation “rep C”).

```
clear
clc
sys=linearize ('pendulem',getlinio('pendulem'));
repA=ss(sys.A,sys.B,sys.C,sys.D);
repB=c2d(repA,0.0137);
G_Z=tf(repB)
```

```
repB =

  A =
             x1          x2          x3          x4
   x1         1      0.0137  -0.0004251  -1.941e-06
   x2         0           1    -0.06208  -0.0004251
   x3         0           0       1.002     0.01371
   x4         0           0      0.3222       1.002
```

```
B =

            u1
 x1   0.0002063
 x2     0.03012
 x3  -0.0003382
 x4     -0.0494
```

```
C =

       x1    x2    x3    x4
 y1     1     0  0.61     0

D =

     u1
 y1   0
```

```
G_Z =

  -5.189e-08 z^3 - 5.709e-07 z^2 - 5.709e-07 z - 5.189e-08
  --------------------------------------------------------
        z^4 - 4.004 z^3 + 6.009 z^2 - 4.004 z + 1

Sample time: 0.0137 seconds
Discrete-time transfer function.
```

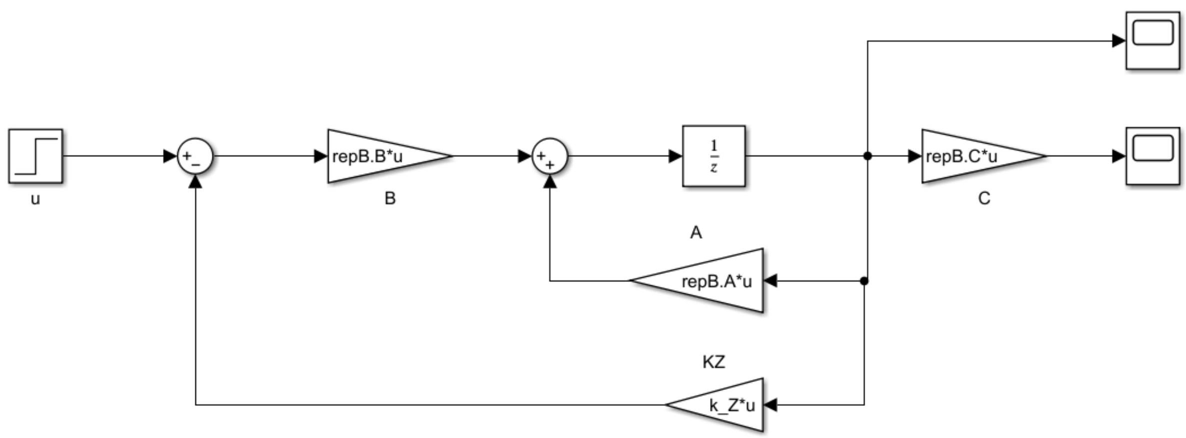
```
[A, B, C, D]=tf2ss(num,den);  % after getting ABCD I convert in in controllable form manually
AC=[0 1  0 0;0 0 1 0;0 0 0 1;-1.0000 4.0044 -6.0088 4.0044];
BC=[0;0;0;1];
CC=[-5.189e-08,- 5.709e-07,- 5.709e-07,- 5.189e-08];
DC=[0]
repC=ss(AC,BC,CC,DC)
```

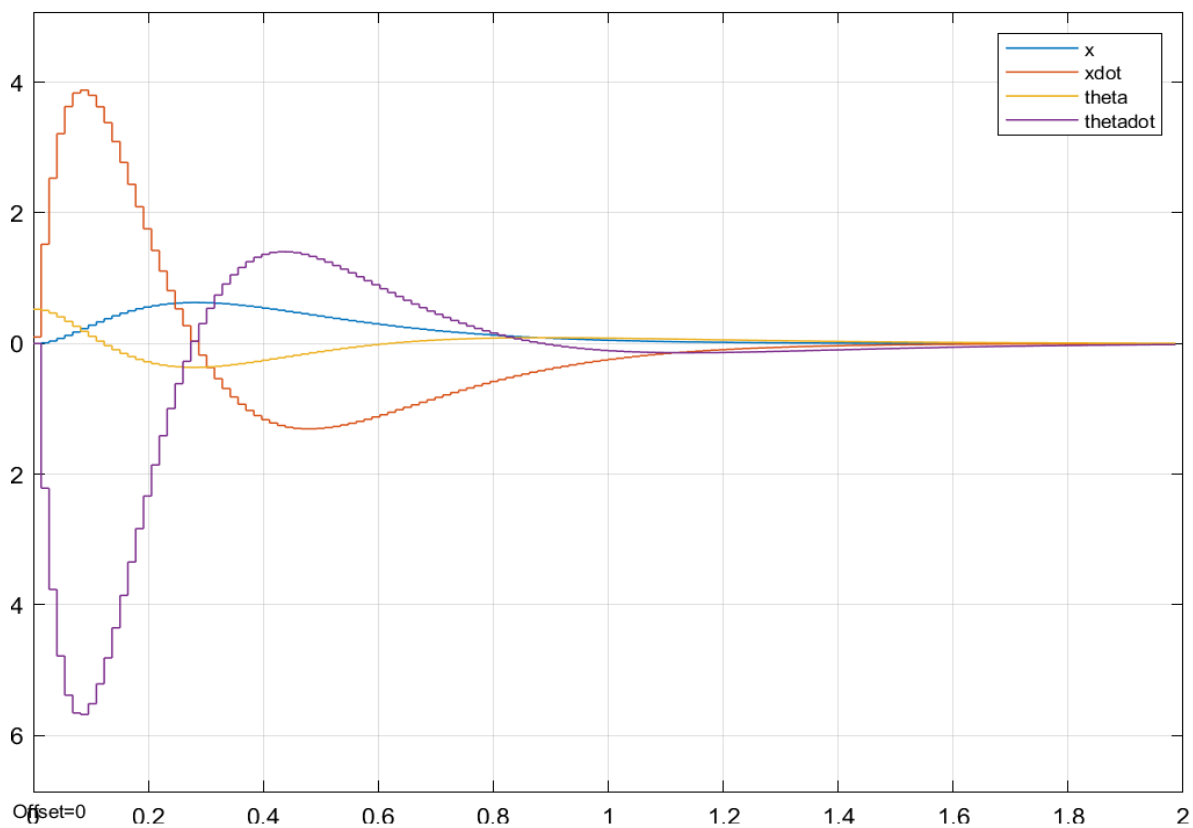
```
repC =

  A =
           x1      x2      x3      x4
   x1       0       1       0       0
   x2       0       0       1       0
   x3       0       0       0       1
   x4      -1   4.004  -6.009   4.004

  B =
       u1
   x1   0
   x2   0
   x3   0
   x4   1

C =
            x1          x2          x3          x4
 y1 -5.189e-08  -5.709e-07  -5.709e-07  -5.189e-08

  D =
       u1
   y1   0
```

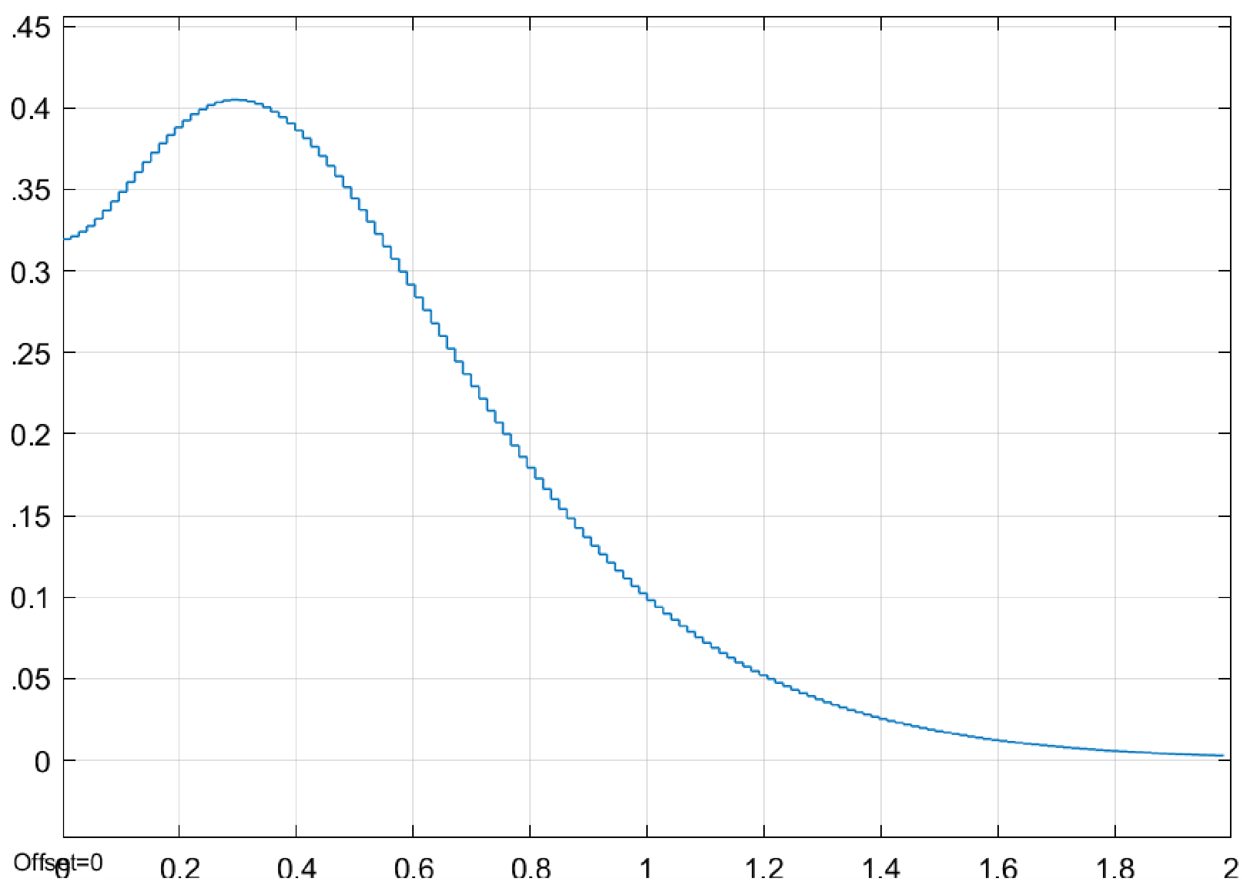
12- Design state feedback vector for “rep B” to achieve appropriate transient response specifications. Implement your designed values using “rep B” and “rep C”. In each case draw the states and the output versus K. Comment on your results.

RepB

```
P_Z=exp(0.0137.*p)  %same poles in discrete form
k_Z=place(repB.A,repB.B,P_Z)
```

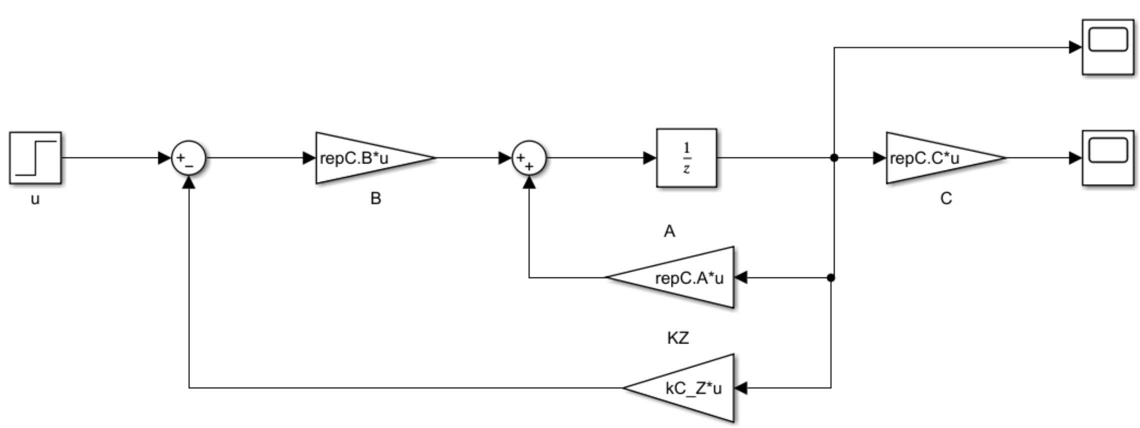


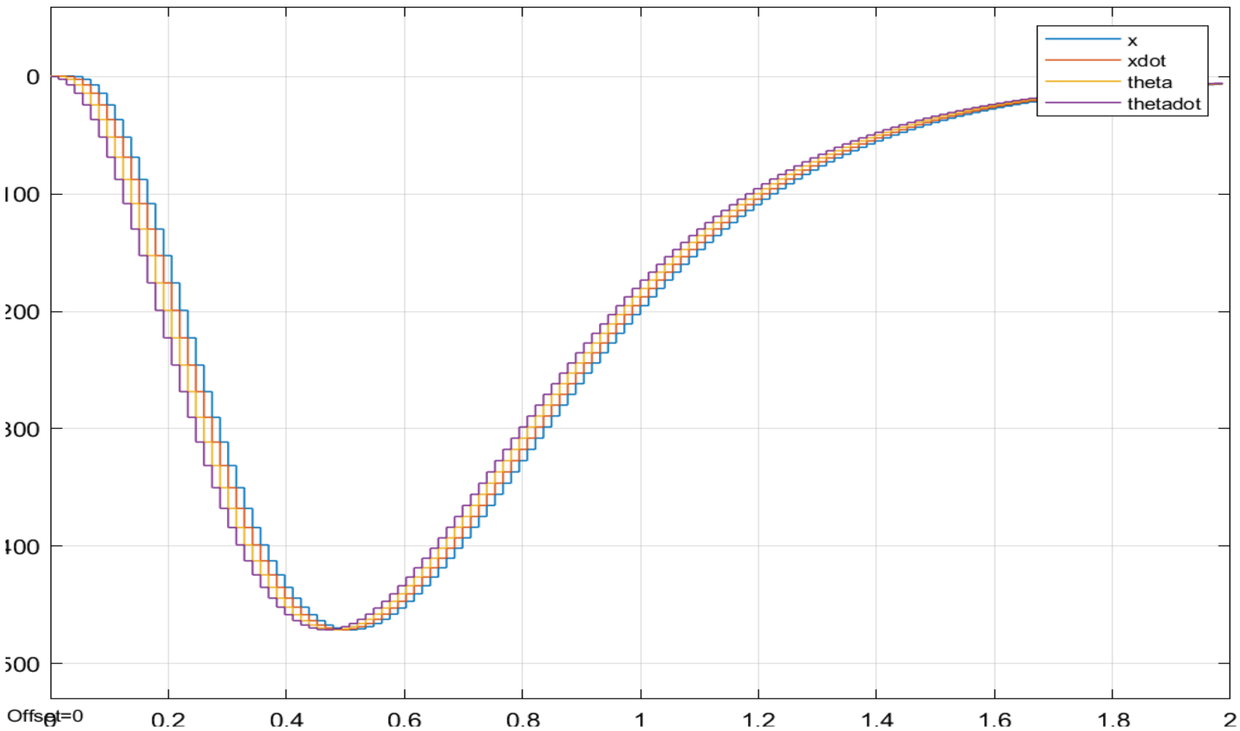


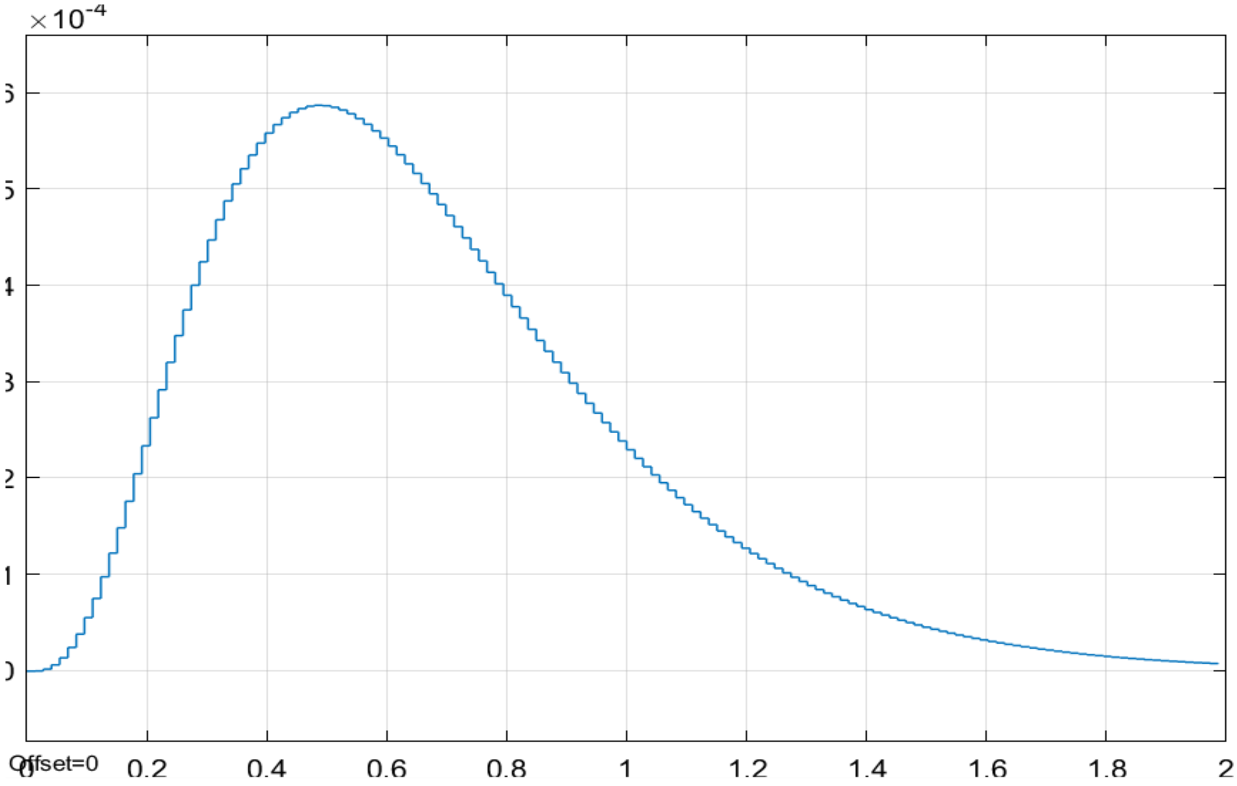


RepC

```
kC_Z=place(repC.A,repC.B,P_Z)

kC_Z =

   -0.2997    0.9412   -0.9856    0.3441
```







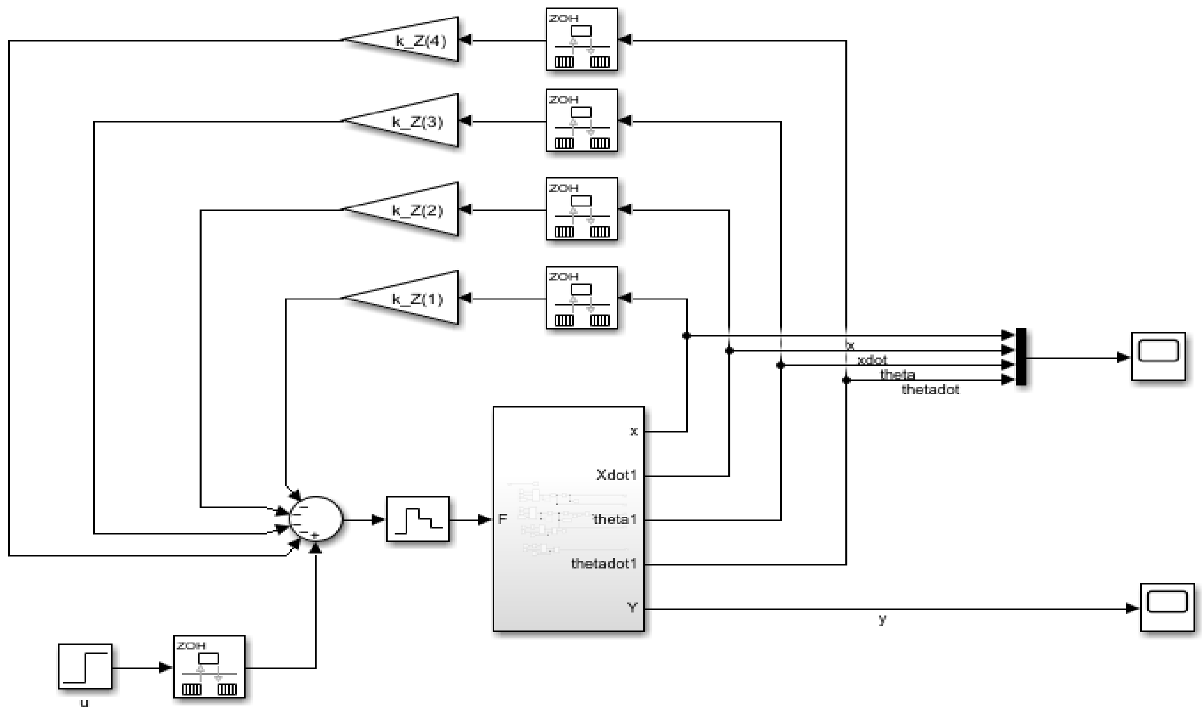
Comment:

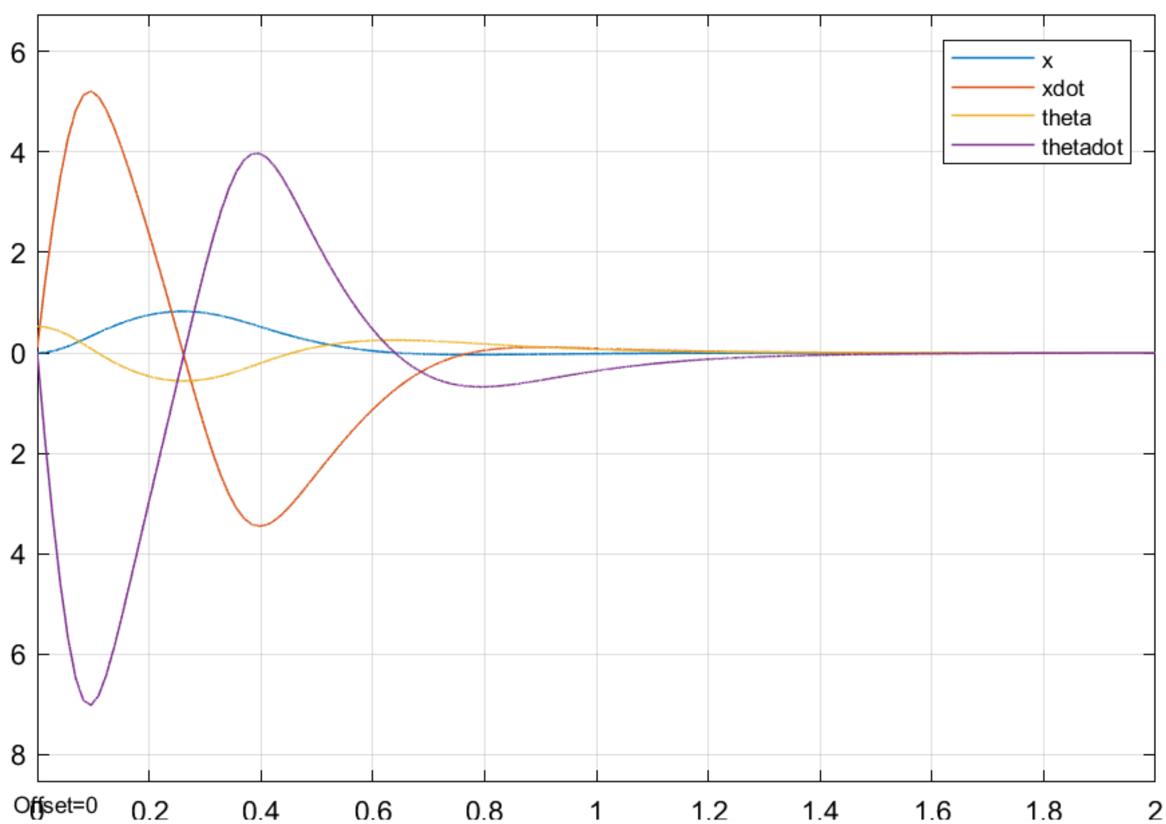
We noted that the system using repB similar to the continuous system in response.

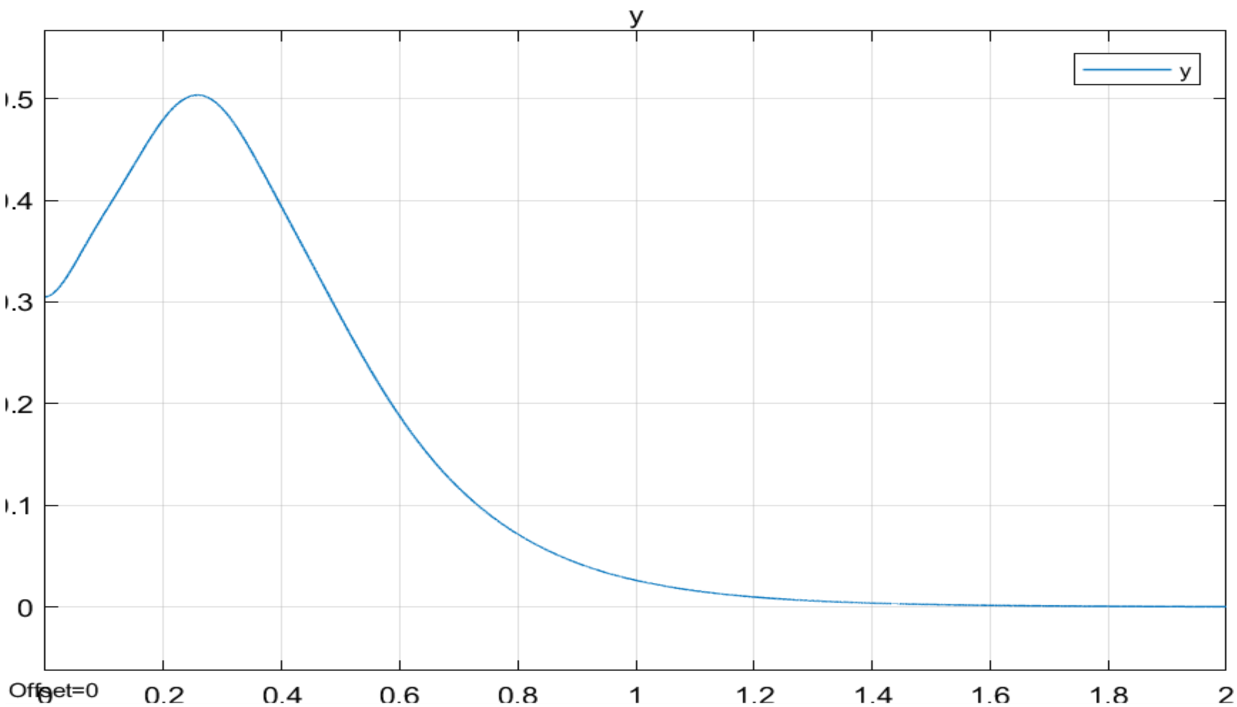
In repC the system is stable but has huge O.S.

13-Implement the state feedback signals using nonlinear system and use condition 2 as an initial condition. Plot the system states and output. Comment on your results.

<u>RepB</u>



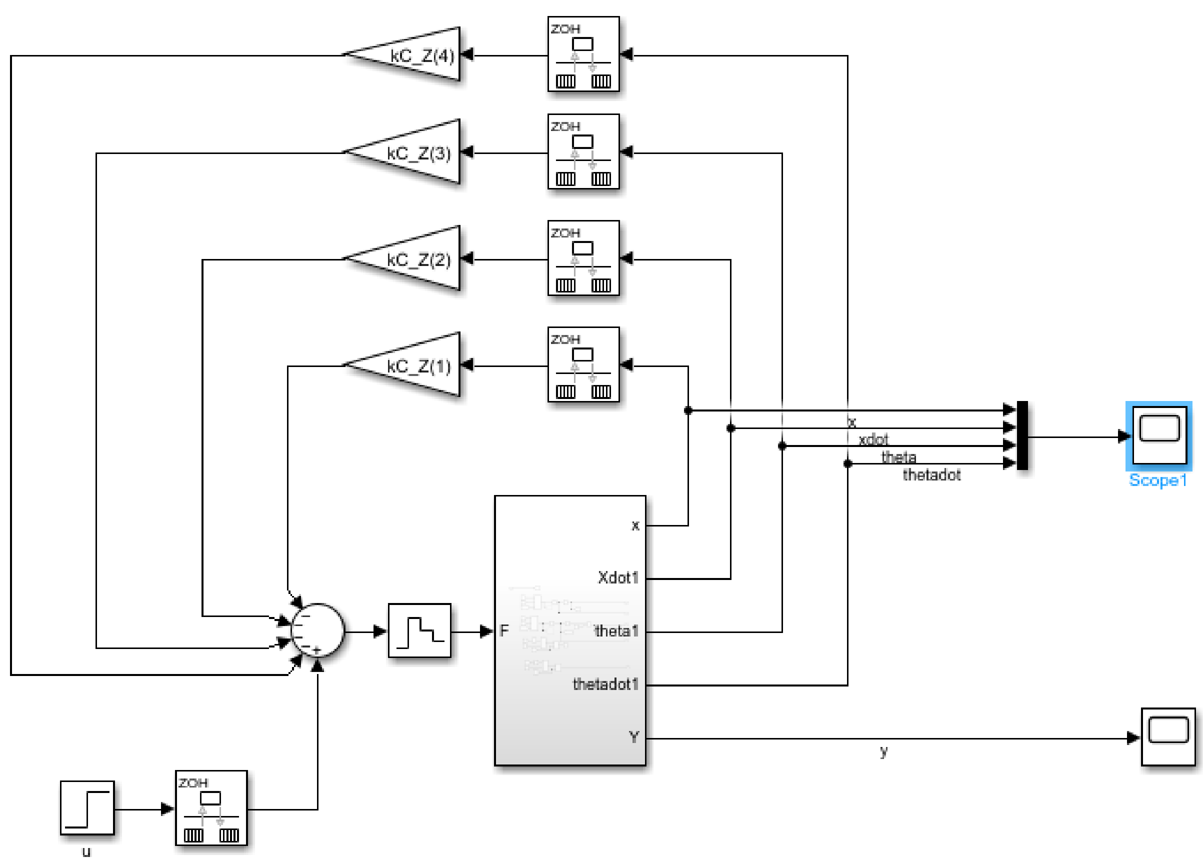


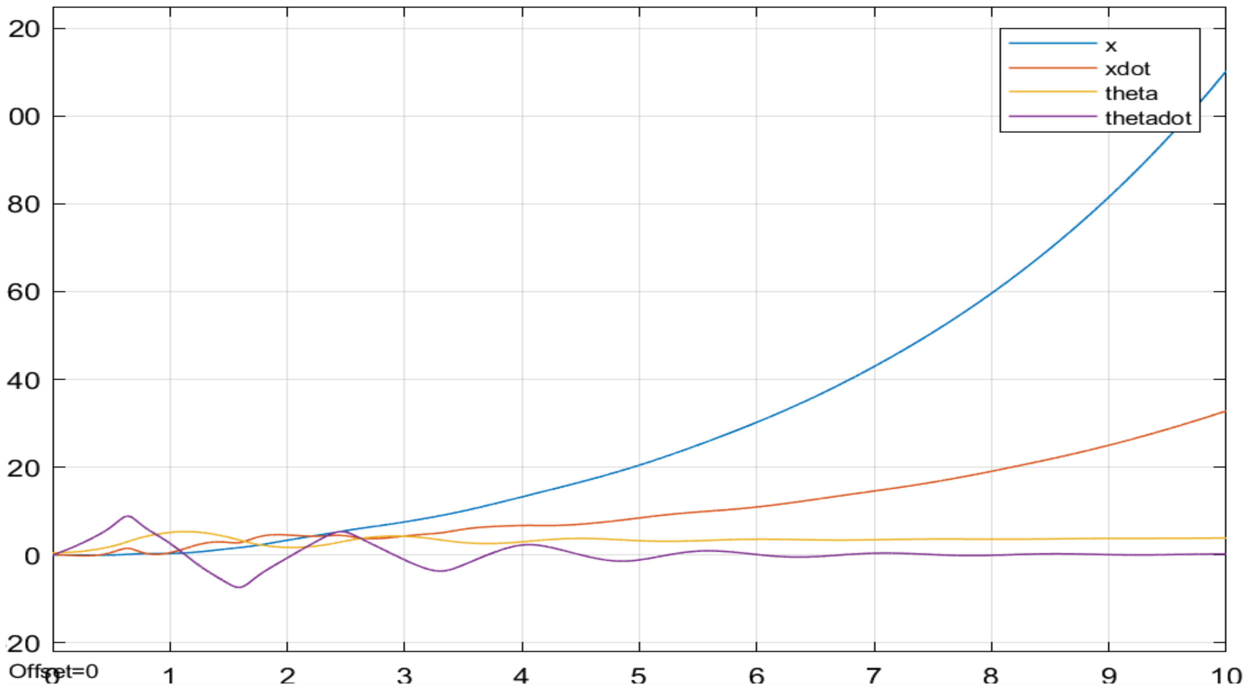


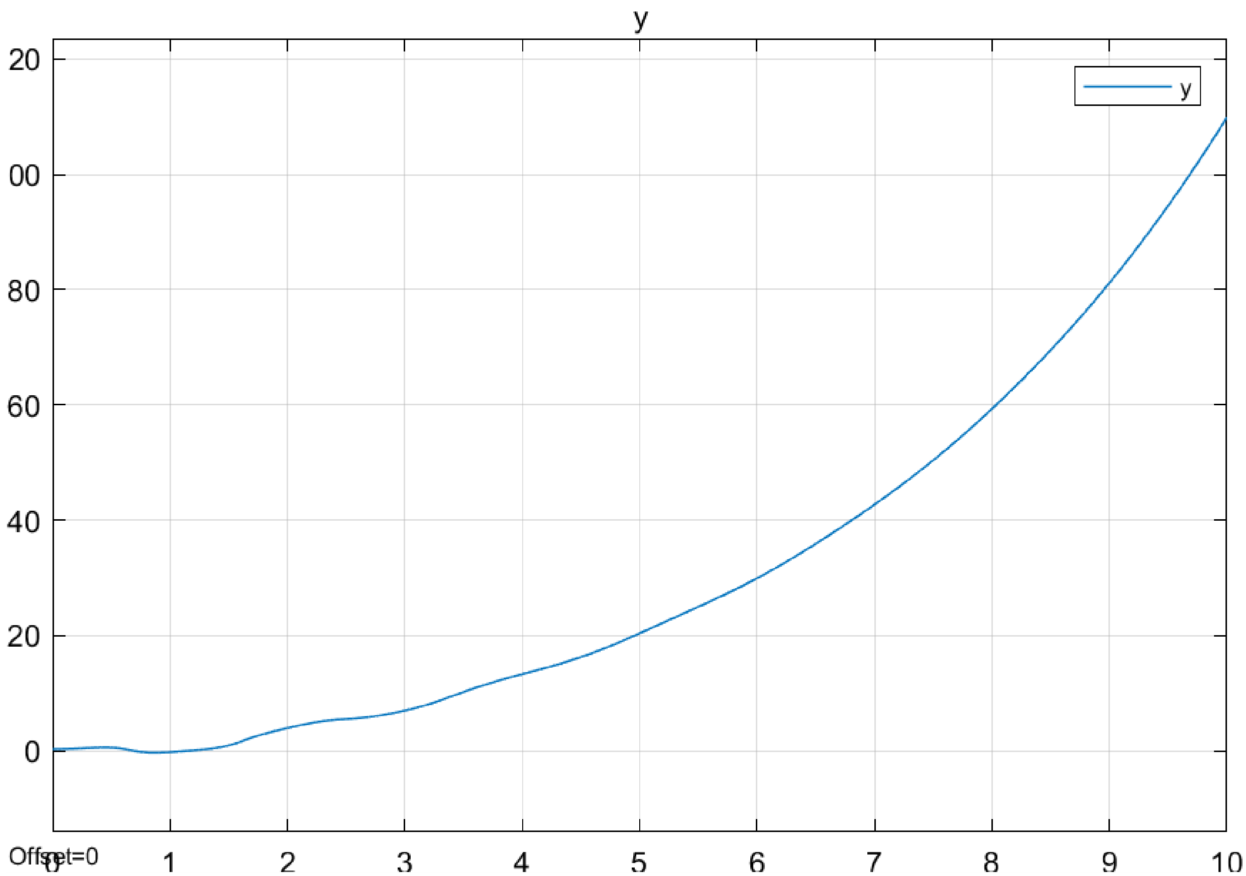
Comment:

Same response as nonlinear system.

RepC







Comment:

We noted that theta is stable but has SSE and the other states is unstable.