Digital control systems EPE 3090

Prof. Dr. Abdellatif ElShafey
Dr. Mahmoud ElNaggar
Eng. Sherif Fathalla

DIGITAL CONTROL PROJECT

FLEXIBLE ROBOTIC ARM

MADE BY:

محمد أشرف مبروك العبد	سكشن 3	بنش 33
محمد ثروت إبراهيم إمام	سكشن 3	بنش 34
محمد جمال حب الدين تركي	سكشن 3	بنش 35

Contents

CONTINUOUS SYSTEM ANALYSIS AND CONTROL DESIGN USING THE TRANSFER FUNCTION APPROACH 5

1. Find the transfer function $\boldsymbol{YU}$. Plot the root locus and find the stability range of a gain controller. 5

MATLAB CODE TO OBTAIN THE TRANSFER FUNCTION OF A SYSTEM USING ITS STATE SPACE REPRESENTATION: 5

OUTPUT 5

ROOT LOCUS FOR THE TRANSFER FUNCTION AND STABILITY RANGE: 6

POLES & ZEROS 6

STABILTY RANGE 6

SYSTEM ARCHITECTURE 7

2. Use the root locus to design a lead compensator that achieves a damping ratio 0.7 and a natural frequency 0.5 rad/sec. 7

POLES & ZEROS 7

DESIGN REQUIREMENTS 8

DESIGN REGION 8

STEP RESPONSE 8

3. Use the Bode plots to design a lead compensator that achieves a damping ratio 0.7 and a natural frequency 0.5 rad/sec. 9

POLES & ZEROS 9

BODE PLOT 10

STEP RESPONSE 11

.4 Simulate the system with both controllers (designed in parts 2 and 3 above) assuming the reference is a unit step and the disturbance is a pulse of magnitude 1 that appears from 20 < t < 22 secs. 11

MODEL 11

DISTURBANCE PLOT 12

STEP INPUT 13

OUTPUT SIGNAL & CONTROLLER ACTION (ROOT LOCUS) & DISTURBANCE PULSE 14

OUTPUT SIGNAL & CONTROLLER ACTION (BODE PLOT) & DISTURBANCE PULSE 15

Digital Control Design Using the Transfer Function Approach 16

1. Show the effect of selecting short and long sampling interval on the resulting poles and zeros in the discrete model. Comment on your results. 18
POLES & ZEROS 18
STABILTY RANGE 18
POLES & ZEROS 19
STABILTY RANGE 19
2. Select an appropriate sampling interval and design a digital controller (using the direct digital control design) to stabilize the system and ensure a damping ratio 0.7 and a natural frequency 0.5 rad/sec. 19
DESIGN REQUIREMENTS 20
DESIGN REGION 21
STEP RESPONSE 21
ROOT-LOCUS AFTER COMPENSATION 22
3. Repeat in question #2 using a design by emulation. 22
POLES & ZEROS 23
STABILTY RANGE 23
DESIGN REGION 24
STEP RESPONSE 24
ROOT-LOCUS AFTER COMPENSATION 25
.4 Compare the results due to the design by emulation to those obtained by the direct design. Comment on the results. 26
COMMENT 27
Continuous State Space Representation 28
1. Using the continuous system dynamics given, develop and Simulink model for open loop system. Draw the system states (Call this representation “rep A”). 28
System state space representation (**repA**) 28
Simulink model 28
29
SYSTEM STATES 29
X1 29
X2 30
Y=X3 31

2. Find appropriate state feedback gain to place the poles of the system in suitable places (You can use the same requirements as mentioned in the transfer function approach). 31
DESIRED POLES 31
CODE 31
3. Implement the feedback signals using the Simulink. Draw the states and output versus time. Comment on the results. 32
SIMLINK MODEL WITH STATE FEEDBACK 32
SYSTEM STATES....... 32
X1 32
COMMENT 34
Discrete State Space Representation 34
1. 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”). Draw the system states for both representations for open loop case. Comment on the results. 34
DISCRETIZATION 34
SYSTEM STATES....... 35
X1 35
X2 35
X3=Y 36
CONTROLLABLE CANONICAL FORM 37
SIMULINK MODEL....... 37
SYSTEM STATES....... 38
X1 38
X2 38
X3 39
Y 39
COMMENT 40
2. Design state feedback vector for “rep B” to achieve same transient response specifications as before and use reference manipulation gain to achieve zero steady state error. Implement your controller on the continuous-time model (rep A). Draw the states and output versus time. Comment on the results. 40
repB....... 40

POLE PLACEMENT 40
RESPONSE 40
X1 40
X2 41
X3=Y 41
repA 42
MODEL 42
RESPONSE 42
COMMENT 44
3. Implement same controller designed on (2) using the states measurements from (rep C). Draw the system states, and comment on the results. 44
repC........ 44
MODEL 44
POLE PLACEMENT 45
RESPONSE 45
X1 45
X2 46
X3 46
Y 47
COMMENT 47
4. Using rep A and the controller designed in 2, assuming that the only measurement available is the output $\omega 2$, design an appropriate observer for the system and implement the state feedback. Draw the system states (starting from initial condition x = [3 5 0]^T and the estimated states. Comment on the results. 47
5. For the controller in 4, if the sampling interval is changed to double its value, what will be the effect of this change on the feed-back system response with the same state feedback gain? Study this effect and draw the system states. 47

CONTINUOUS SYSTEM ANALYSIS AND CONTROL DESIGN USING THE TRANSFER FUNCTION APPROACH

1. Find the transfer function $\frac{Y}{U}$. Plot the root locus and find the stability range of a gain controller.

MATLAB CODE TO OBTAIN THE TRANSFER FUNCTION OF A SYSTEM USING ITS STATE SPACE REPRESENTATION:

CODE

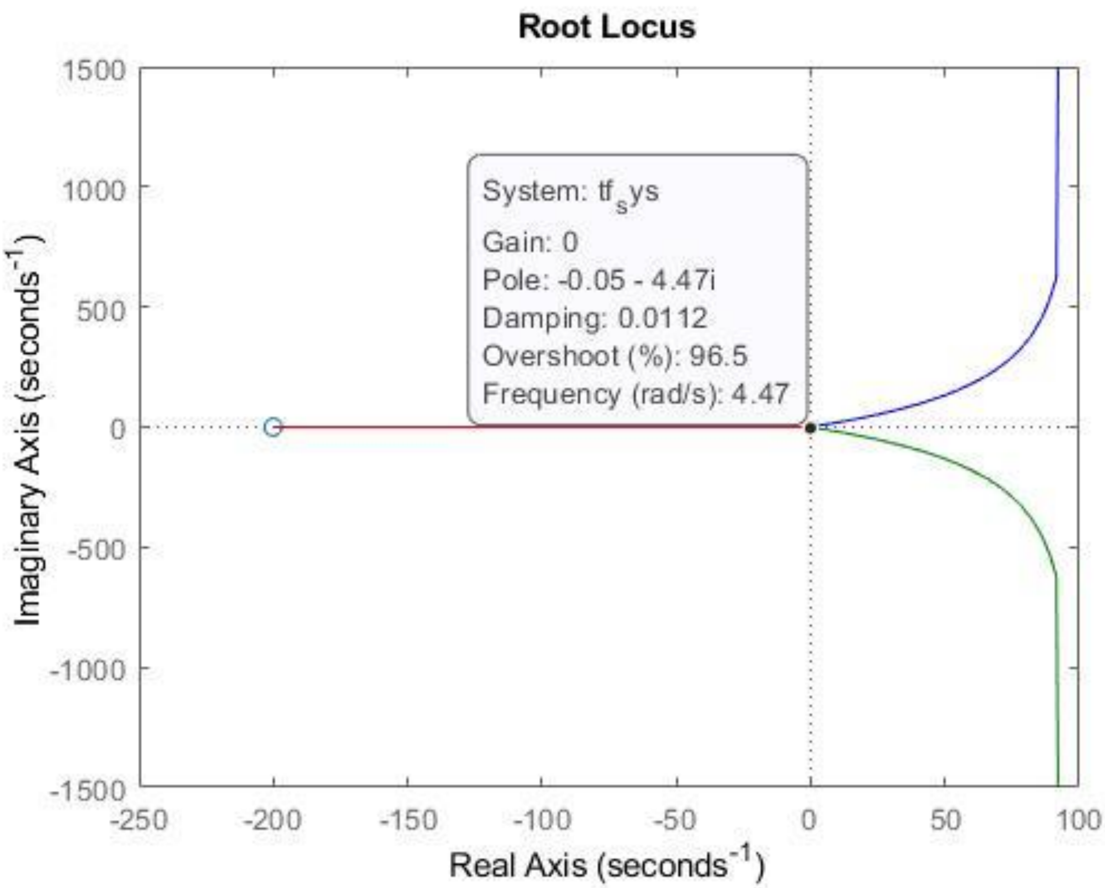
```
J1 = 10/9;
J2 = 10;
k = 1;
d= 0.1;
ki=1;
w0 = sqrt(k*(J1 + J2)/(J1*J2));

alpha=J1/(J1+J2);
beta1=d/(J1*w0);
beta2=d/(J2*w0);
gamma=ki/(J1*w0);
delta=1/(J1*w0);

% Define the system's state space representation
A = [0 1 -1; alpha-1 -beta1 beta1; alpha beta2 -beta2]; % state matrix
Anew=w0*A;
B = [0; gamma; 0;]; % input matrix
C = [0 0 w0]; % output matrix
D = 0; % feedthrough matrix

% Convert the state space representation to transfer function
sys = ss(Anew, B, C, D);
tf_sys = tf(sys);
sisotool(tf_sys)
```

OUTPUT

```
tf_sys =

        0.009 s + 0.09
  -----------------------------
  s^3 + 0.1 s^2 + s - 4.256e-19

Continuous-time transfer function.
```

ROOT LOCUS FOR THE TRANSFER FUNCTION AND STABILITY RANGE:

ROOT LOCUS



POLES & ZEROS

Zeros:

Zero at s = -0.09/0.009 = -10

Poles:

Pole at s3 = -0.00233 (real pole)

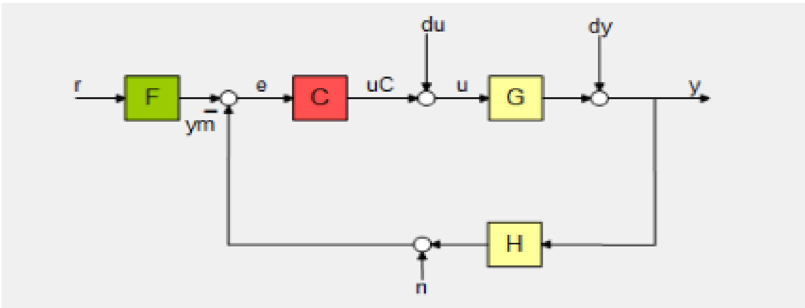
Pole at s2 = -0.0499 – 0.999i (complex conjugate pole)

Pole at s1 = -0.0499 + 0.999i (complex conjugate pole)

STABILTY RANGE

0<Kp< 1.1529 (CRITICALLY STABLE)

SYSTEM ARCHITECTURE



2. Use the root locus to design a lead compensator that achieves a damping ratio 0.7 and a natural frequency 0.5 rad/sec.

POLES & ZEROS

We will use a compensator that cancels the TF poles of the system near the imaginary axis (i.e. faster poles) which are at s = -0.00169 + 1.322j & s = -0.00169 - 1.322j. So the compensator poles and zeros are:

Controller zeros:

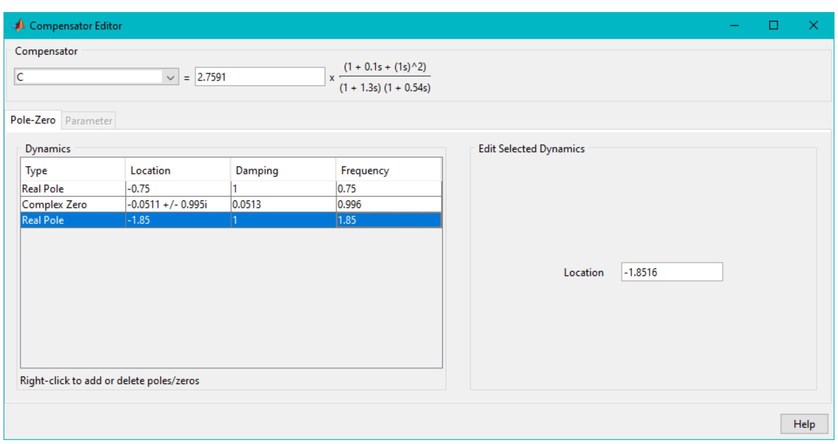
Zero at s = -0.0021301+j0.9987 (complex conjugate pole)

Zero at s = -0.0021301-j0.9987 (complex conjugate pole)

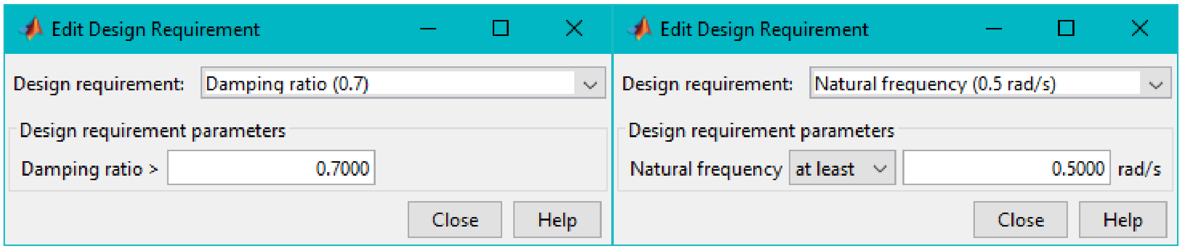
Controller Poles:

Pole at s = -0.75 (real pole)

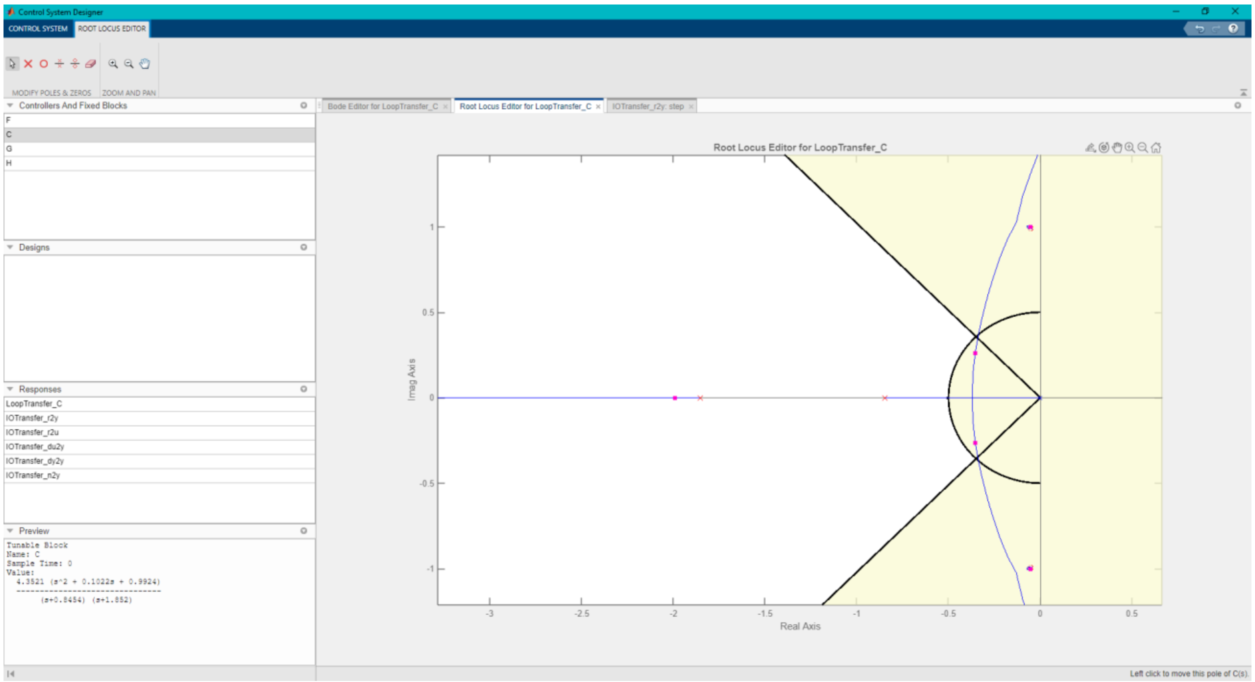
Pole at s = - 1.8516 (real pole)



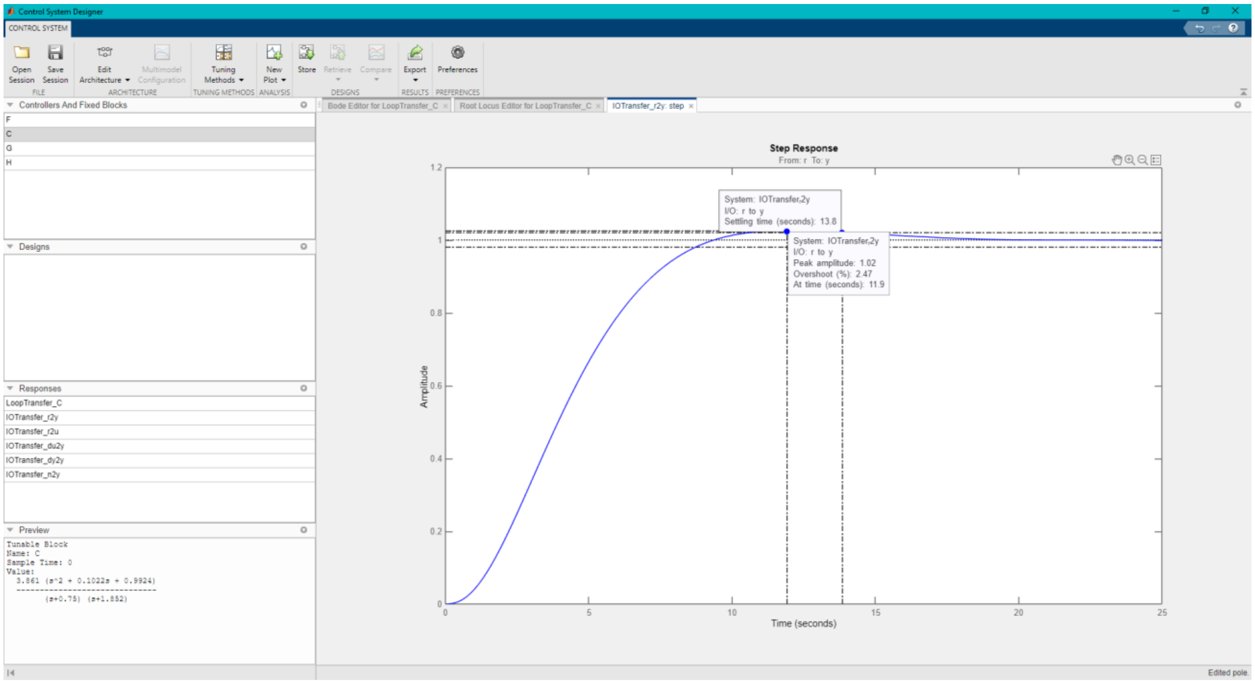
DESIGN REQUIREMENTS



DESIGN REGION



STEP RESPONSE



3. Use the Bode plots to design a lead compensator that achieves a damping ratio 0.7 and a natural frequency 0.5 rad/sec.

POLES & ZEROS

We will use a compensator that cancels the TF poles of the system near the imaginary axis (i.e. faster poles) which are at s = -0.00169 + 1.322j & s = -0.00169 - 1.322j. So the compensator poles and zeros are:

Controller zeros:

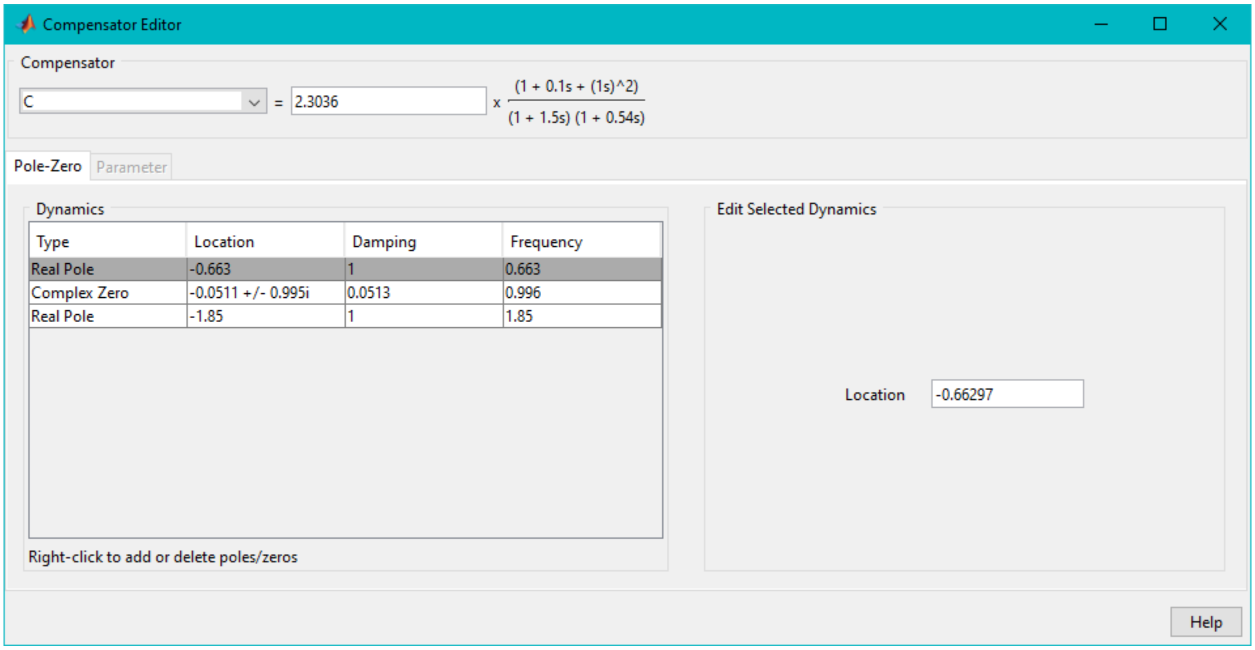
Zero at s = -0.051121+ j0.99487 (complex conjugate pole)

Zero at s = -0.051121- j0.99487 (complex conjugate pole)

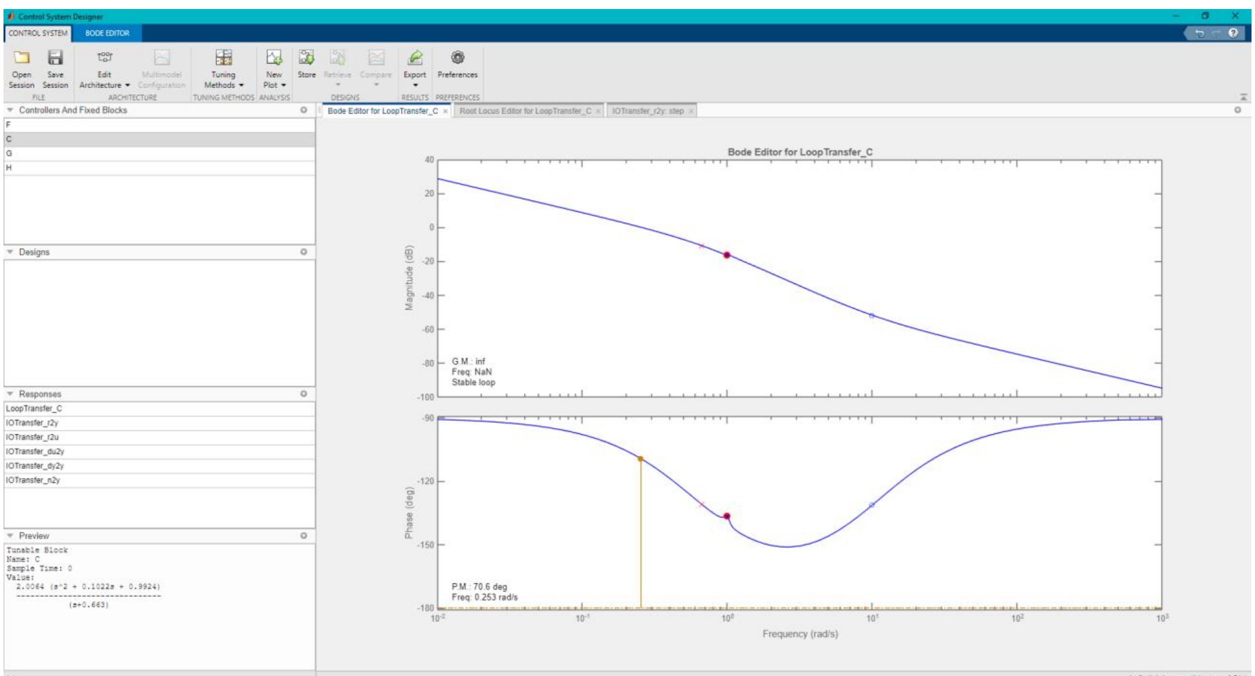
Controller Poles:

Pole at s = -0.66297 (real pole)

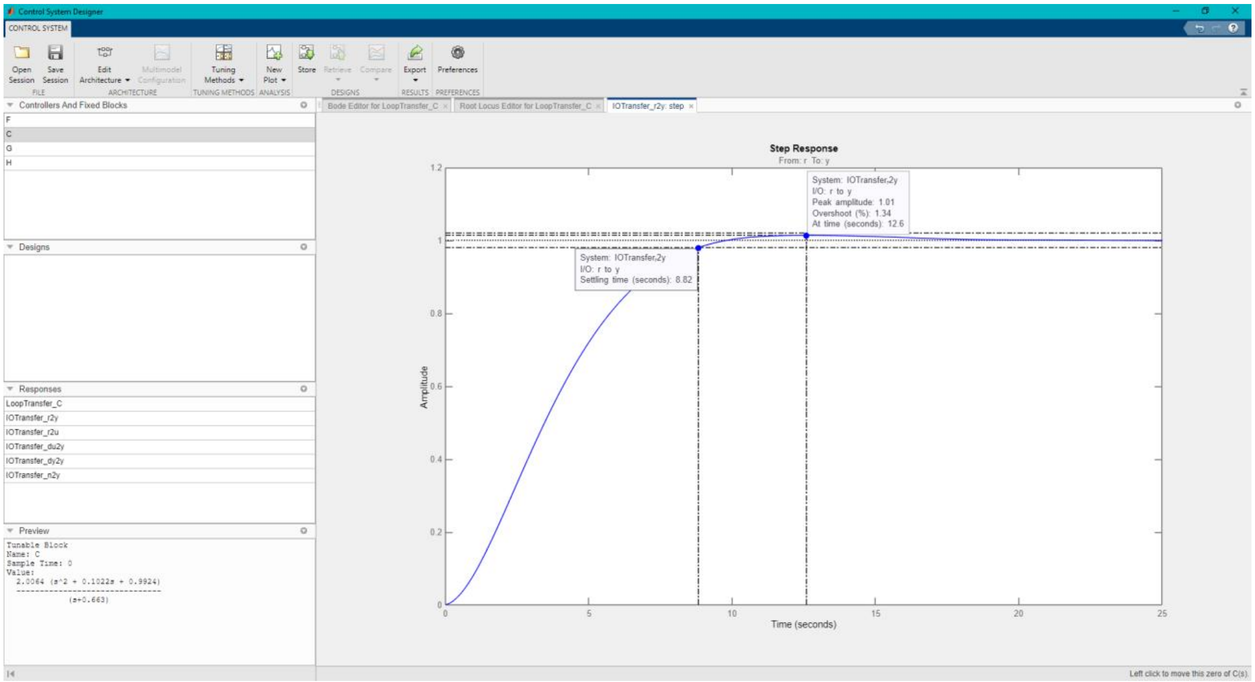
Pole at s = - 1.8516 (real pole)


Compensator Editor
Compensator
C = 2.3036 x (1 + 0.1s + (1s)^2) / (1 + 1.5s) (1 + 0.54s)
Pole-Zero
Parameter
Dynamics
Type | Location | Damping | Frequency
Real Pole | -0.663 | 1 | 0.663
Complex Zero | -0.0511 +/- 0.995i | 0.0513 | 0.996
Real Pole | -1.85 | 1 | 1.85
Right-click to add or delete poles/zeros
Edit Selected Dynamics
Location -0.66297
Help

BODE PLOT

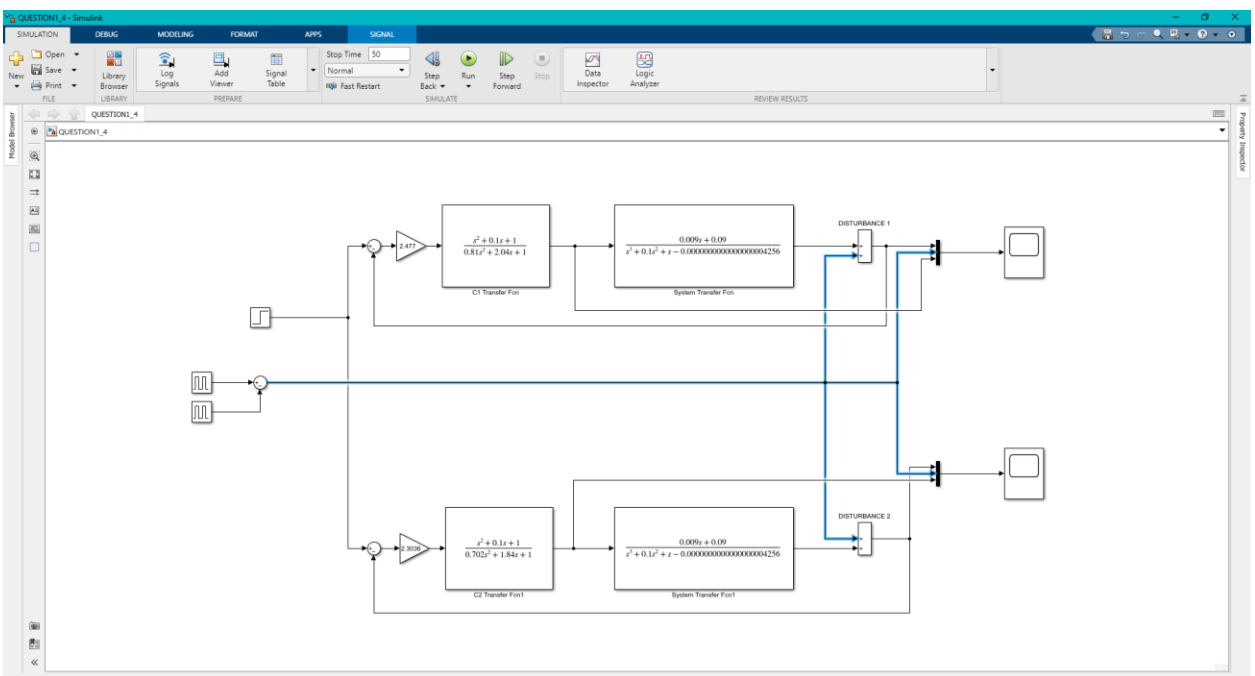

Control System Designer
Bode Editor for LoopTransfer_C
Magnitude (dB)
G.M.: inf
Freq: NaN
Stable loop
Phase (deg)
P.M.: 70.6 deg
Freq: 0.253 rad/s
Frequency (rad/s)

STEP RESPONSE



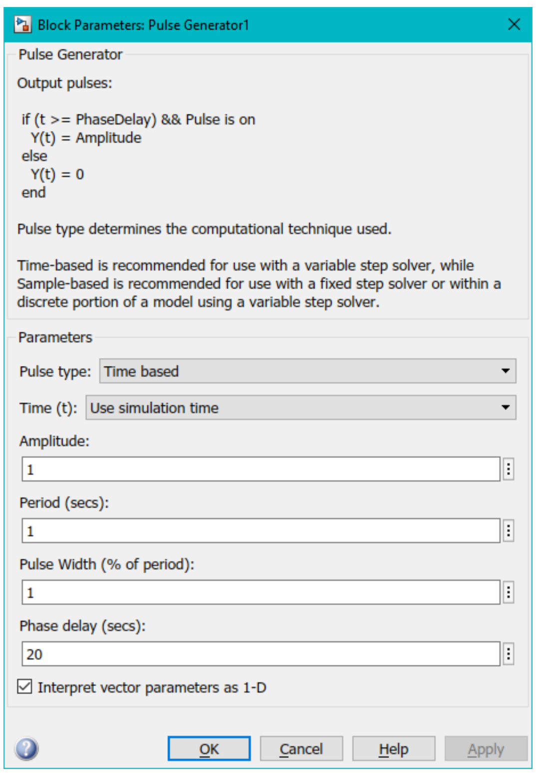
4. Simulate the system with both controllers (designed in parts 2 and 3 above) assuming the reference is a unit step and the disturbance is a pulse of magnitude 1 that appears from 20 < *t* < 22 secs.

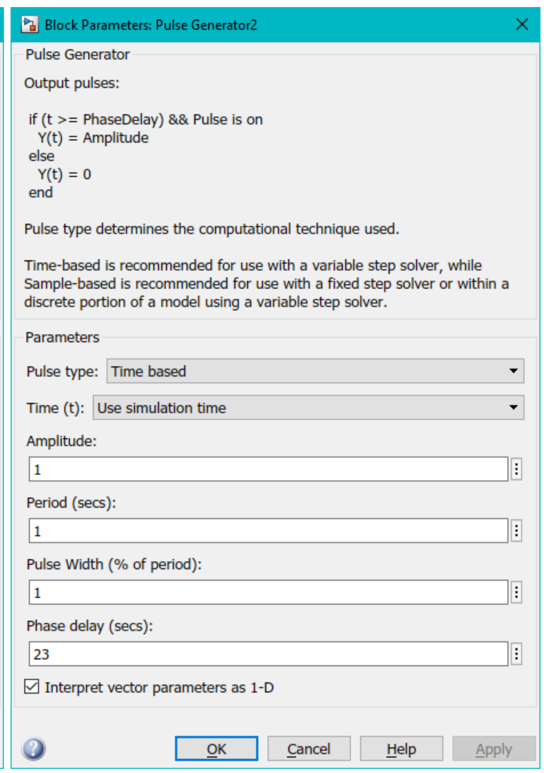
MODEL



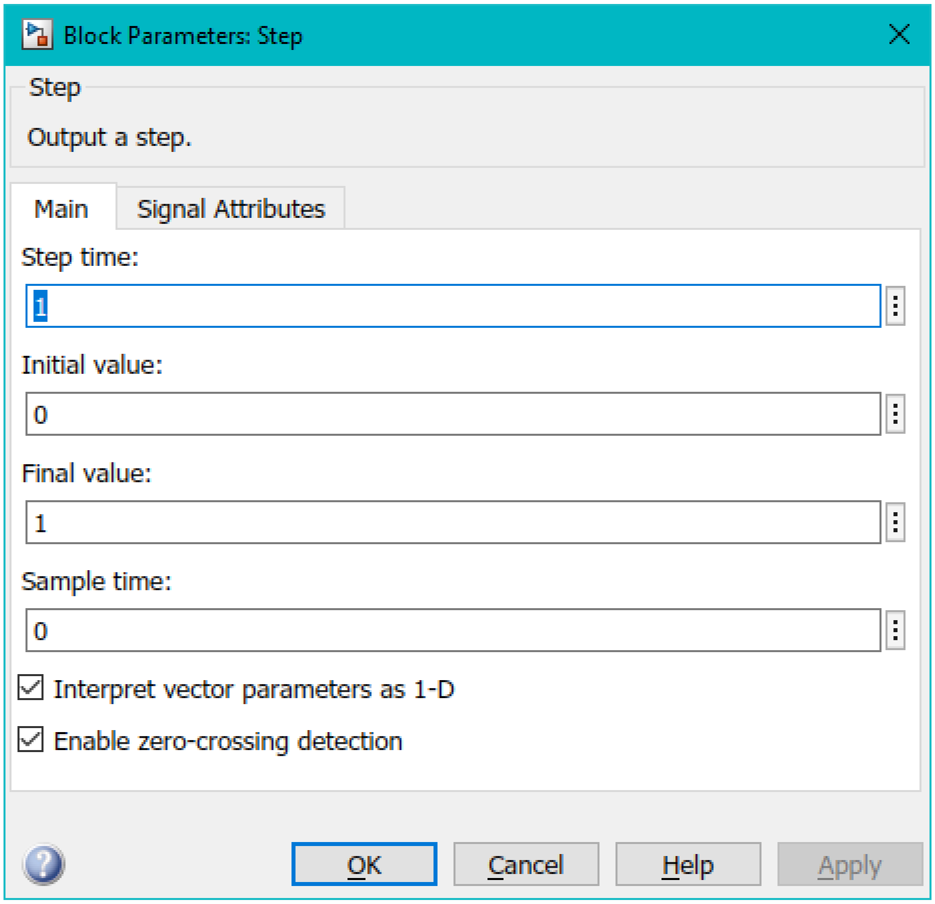
DISTURBANCE PLOT

We create a pulse signal using two pulse generators in Simulink, where the signal is active for a time range from t=20 to t=22 and the two pulse generators are activated at different times. The output of the difference block will be a pulse signal that starts at time = 20 sec and ends at time = 23 sec, with a pulse width of 0.1 seconds. Note that by subtracting the two pulse signals from each other, the resulting output signal will have a negative pulse of the same width and amplitude as the positive pulse. The amplitude of the negative pulse will depend on the amplitude of the positive pulse and the timing of the second pulse generator. If the second pulse generator is delayed too much, the negative pulse may not completely cancel out the positive pulse, resulting in a smaller amplitude negative pulse. If the second pulse generator is delayed too little, the negative pulse may overlap with the positive pulse, resulting in a larger amplitude negative pulse.

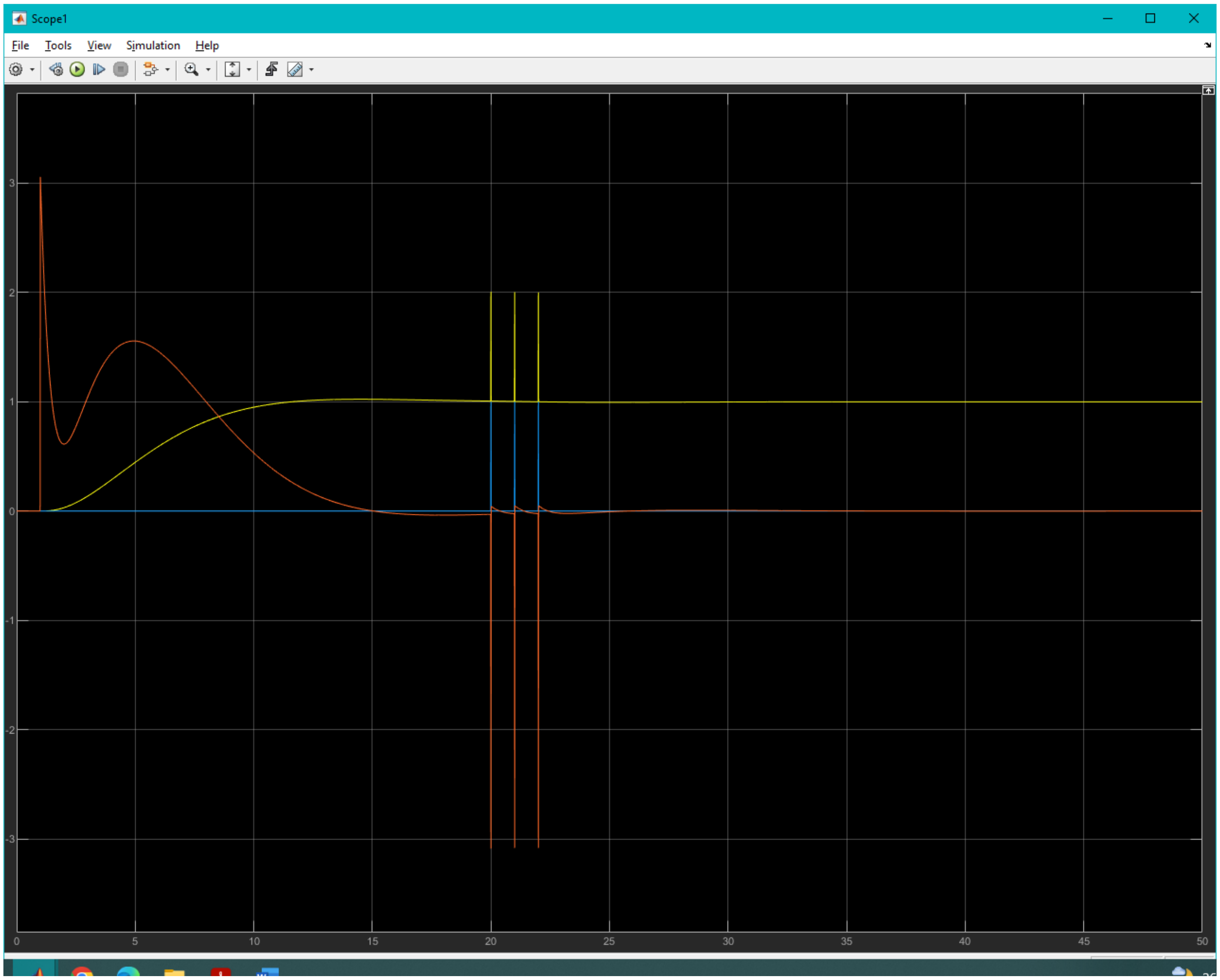




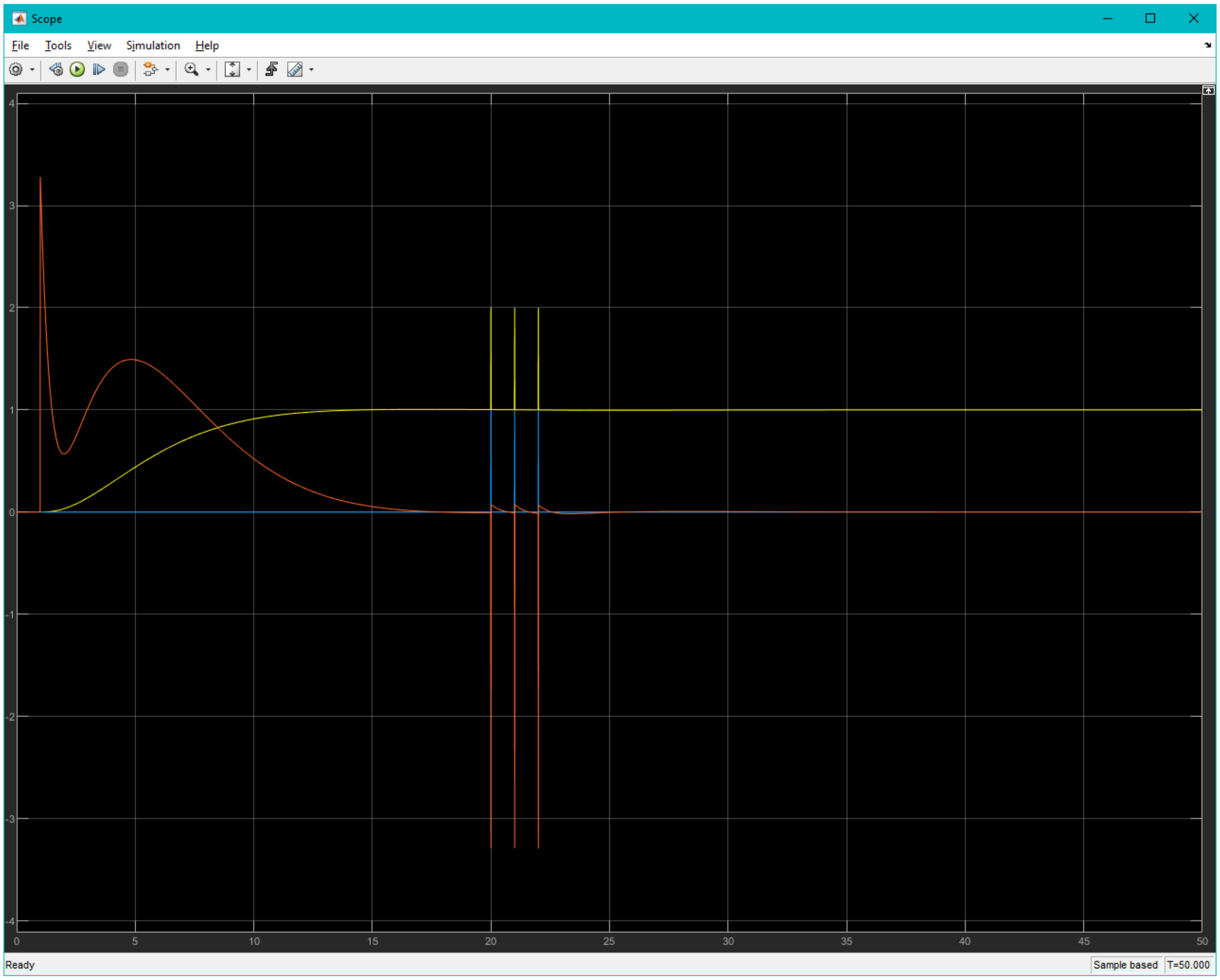
STEP INPUT


Block Parameters: Step
Step
Output a step.
Main
Signal Attributes
Step time:
1
Initial value:
0
Final value:
1
Sample time:
0
Interpret vector parameters as 1-D
Enable zero-crossing detection
OK
Cancel
Help
Apply

OUTPUT SIGNAL & CONTROLLER ACTION (ROOT LOCUS) & DISTURBANCE PULSE



OUTPUT SIGNAL & CONTROLLER ACTION (BODE PLOT) & DISTURBANCE PULSE



Digital Control Design Using the Transfer Function Approach

To solve this problem, first we need to get the discrete form of the transfer function

we need to follow the following steps:

1. Determine the sampling interval: Once we have the continuous-time model, we need to determine the appropriate sampling interval. A general rule of thumb is to choose a sampling interval that is 10 to 30 times the system's bandwidth.

2. Design a digital controller: We need to design a digital controller to stabilize the system and meet the desired damping ratio and natural frequency. The direct digital control design method could be used for this purpose.

3. Apply zero-order-hold: Before applying the output of the controller to the system, we need to pass it through a zero-order-hold to convert the discrete-time signal back to continuous-time.

Here are the detailed steps for designing the digital controller:

1. Determine the sampling interval: Let's assume that we choose a sampling interval of Ts seconds. Here we use 2 sampling intervals namely 0.1 sec and 0.01 sec.

2. Design a digital controller: To design the digital controller, we can use the direct digital control design method as follows:

    a. Convert the continuous-time transfer function G(s) to a discrete-time transfer function G(z) using the c2d function in MATLAB. The resulting transfer function will have the form:

```
>> c2d(G, 0.1)

ans =

  4.631e-05 z^2 + 5.815e-06 z - 4.318e-05
  ---------------------------------------
      z^3 - 2.98 z^2 + 2.97 z - 0.99

Sample time: 0.1 seconds
Discrete-time transfer function.
```

```
>> c2d(G, 0.01)

ans =

  4.513e-07 z^2 + 5.847e-09 z - 4.482e-07
  ---------------------------------------
     z^3 - 2.999 z^2 + 2.998 z - 0.999

Sample time: 0.01 seconds
Discrete-time transfer function.
```

3. Simulate the system in discrete time using the root locus method and find the step response.

4. Get the desired operating region and design a lead compensator C(Z) = $\kappa \frac{z-\beta}{z-p}$ to satisfy the operating conditions.

5. Apply zero-order-hold: Pass the output of the controller through a zero-order-hold to convert the discrete-time signal back to continuous-time.

6. Simulate the system: Simulate the closed-loop system using the continuous-time model of the system and the designed digital controller to ensure that it meets the desired specifications.

1. Show the effect of selecting short and long sampling interval on the resulting poles and zeros in the discrete model. Comment on your results.

As we have seen, when transferring the system to the z domain and choosing different sampling intervals, the transfer function changes.

Thus, when we chose the Ts = 0.1 sec (i.e. long sampling interval), G(z) = $\frac{4.631*10^{-5}z^{2}+5.815*10^{-6}z-4.318*10^{-5}}{z^{3}-2.98z^{2}+2.97z-0.99}$

POLES & ZEROS

Zeros:

Zero at z1 = -1.0304

Zero at z2 = 0.9048

Poles:

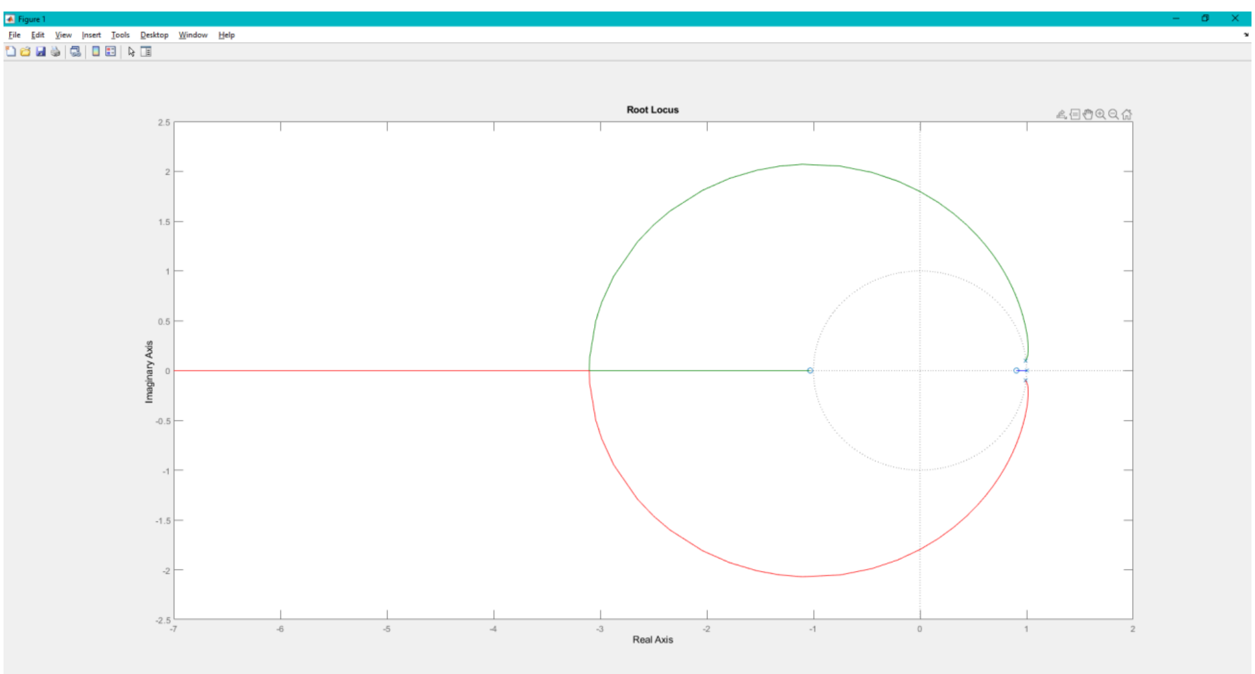
Pole at p3 = 1 (real pole)

Pole at p2 = 0.9901 + 0.0992i (complex conjugate pole)

Pole at p1 = 0.9901 - 0.0992i (complex conjugate pole)

STABILTY RANGE

0<Kp< 11.432 (CRITICALLY STABLE)



and when Ts = 0.01 sec (i.e. short sampling interval), G(z) = $\frac{4.513*10^{-7}z^2+5.847*10^{-9}z-4.482*10^{-7}}{z^3-2.999z^2+2.998z-0.999}$

POLES & ZEROS

Zeros:

Zero at z1 = -1.0030

Zero at z2 = 0.9900

Poles:

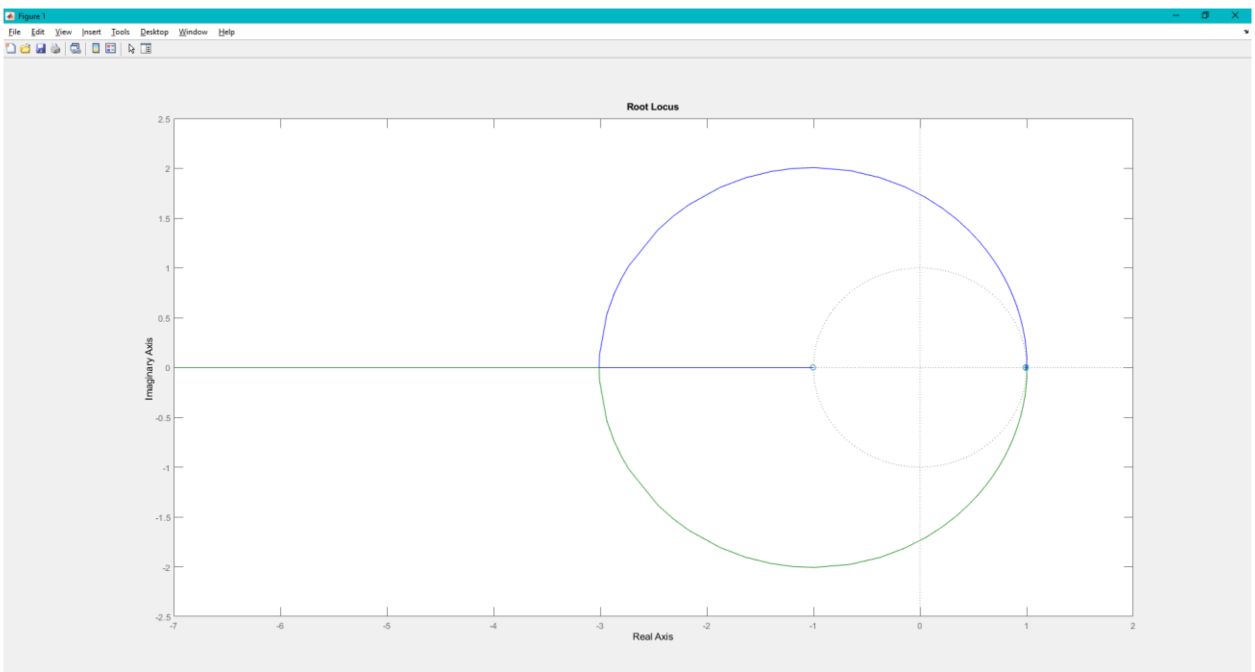
Pole at p3 = 1 (real pole)

Pole at p2 = 0.9995 + 0.0100i (complex conjugate pole)

Pole at p1 = 0.9995 - 0.0100i (complex conjugate pole)

STABILTY RANGE

0<Kp< 13.554 (CRITICALLY STABLE)



2. Select an appropriate sampling interval and design a digital controller (using the direct digital control design) to stabilize the system and ensure a damping ratio 0.7 and a natural frequency 0.5 rad/sec.

WE WILL BE USING Ts=0.01 sec to design a controller using DDC method:

Controller zeros:

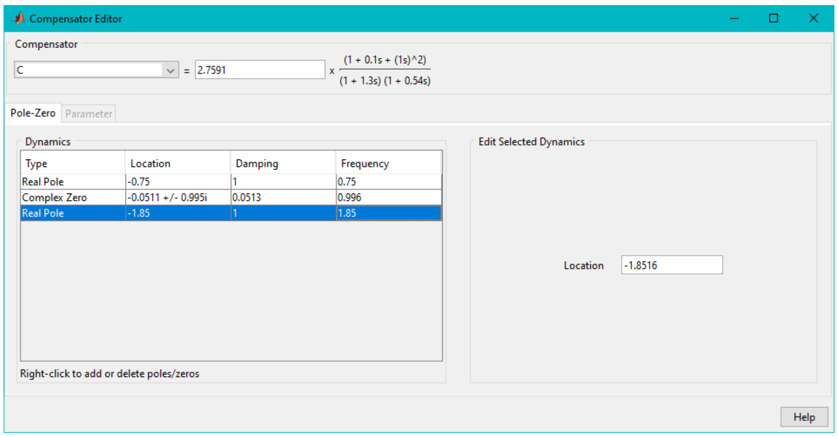
Zero at z = 0.99945+j 0.0099757 (complex conjugate pole)

Zero at z = 0.99945-j 0.0099757 (complex conjugate pole)

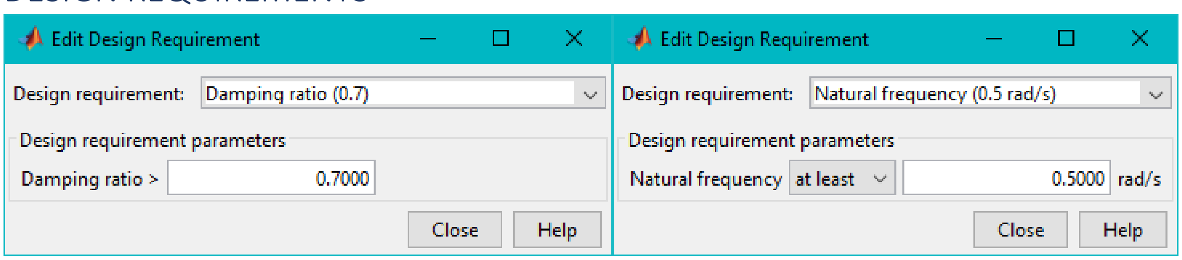
Controller Poles:

Pole at z = 0.99005 (real pole)

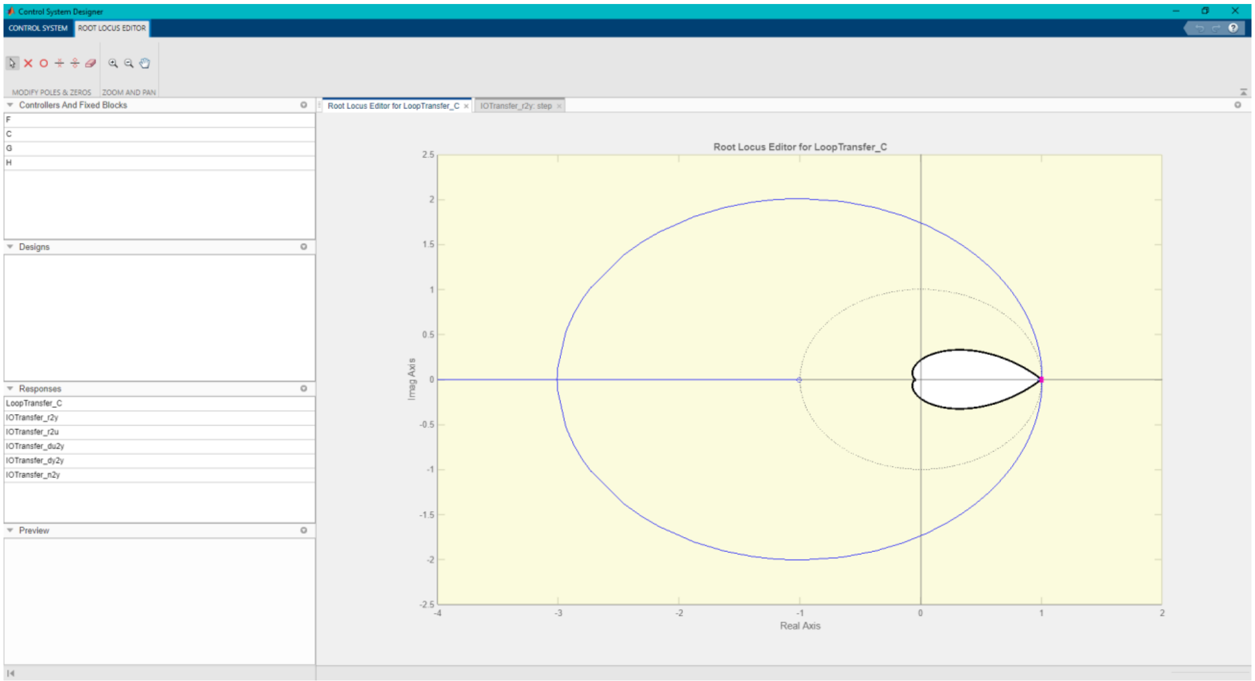
Pole at z = 0.99304 (real pole)



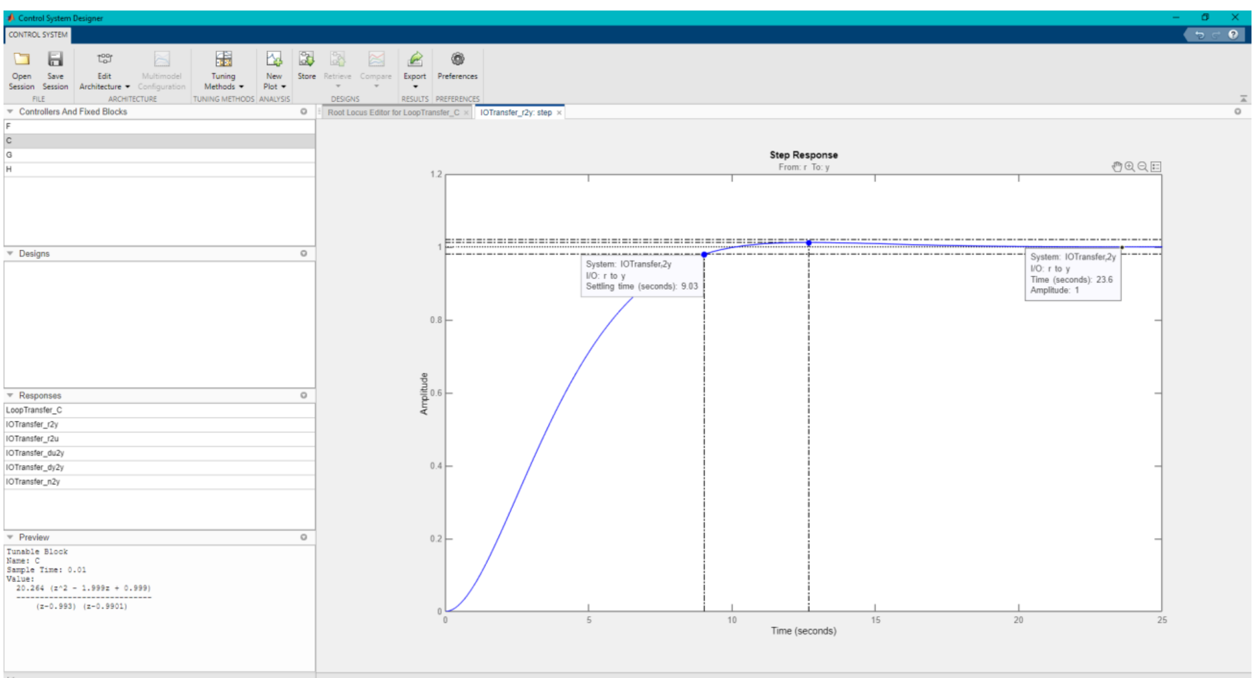
DESIGN REQUIREMENTS



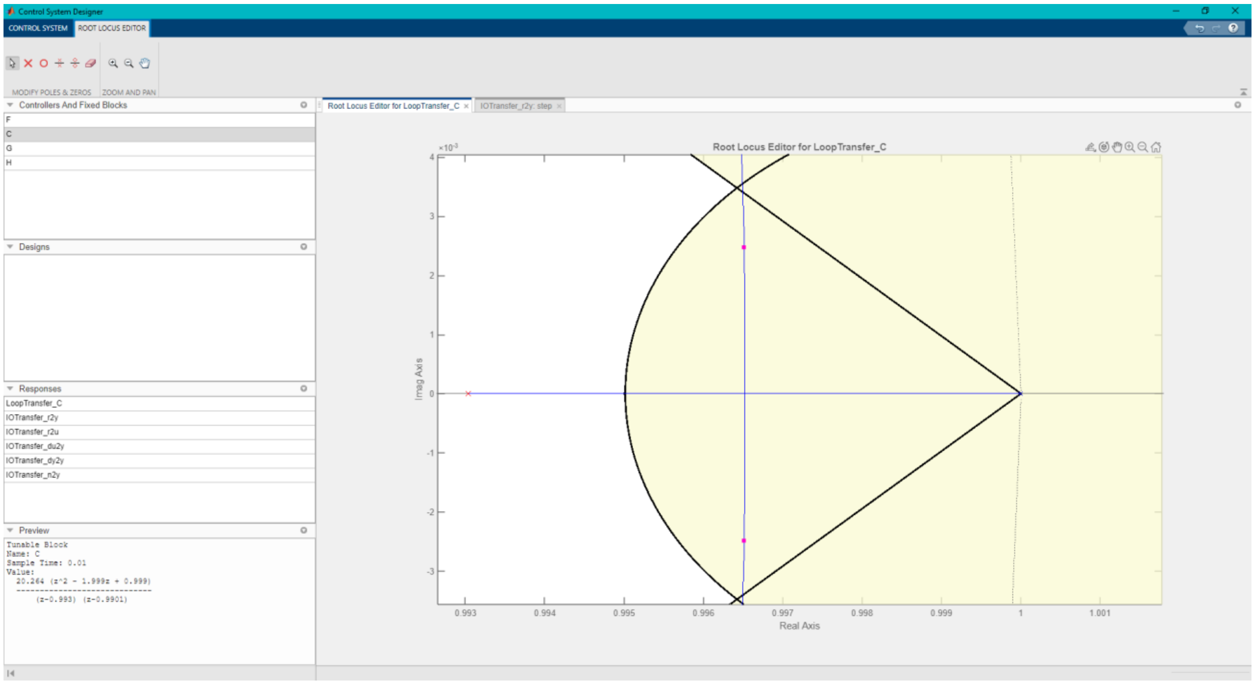
DESIGN REGION



STEP RESPONSE



ROOT-LOCUS AFTER COMPENSATION



3. Repeat in question #2 using a design by emulation.

WE WILL BE USING Ts=0.01 sec to design a controller using EMULATION method:

By substituting $S \longrightarrow \dfrac{z-1}{0.01z}$ This is the formula for backward transformation.

$$G(s) = \frac{0.009\,s\ +\ 0.009}{s^3\ +\ 0.1\,s^2\ +\ s\ -\ 4.256 * 10^{-19}}$$

$$G(z) = \frac{0.00005z^3 - 0.000045z^2}{50.01z^3 - 150.015z^2 + 150.005z - 50}$$

To enter the new G(z) into MATLAB, we use the following code:

```
num = [0.00005 -0.000045 0 0];
den = [50.01 -150.015 150.005 -50];
G = tf(num, den, 1, 'variable', 'z')
sisotool(G)
```

POLES & ZEROS

Zeros:

Zero at z1 = 0

Zero at z2 = 0

Zero at z2 = 0.9000

Poles:

Pole at p3 = 1 (real pole)

Pole at p2 = 0.9999 + 0.0100i (complex conjugate pole)

Pole at p1 = 0.9999 - 0.0100i (complex conjugate pole)

STABILTY RANGE

0<Kp< 0.22309 (CRITICALLY STABLE)

Controller zeros:

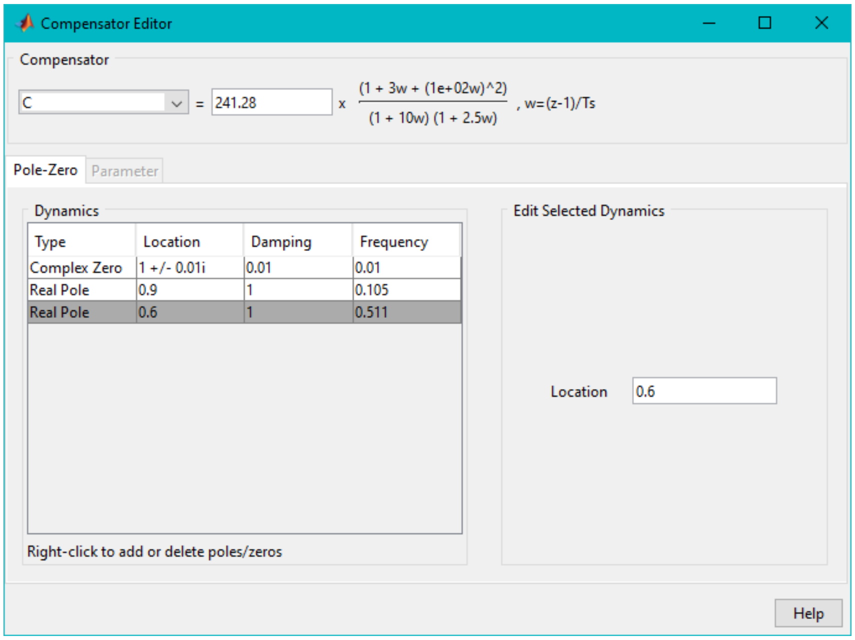
Zero at z = 0.99945+j 0.0099757 (complex conjugate pole)

Zero at z = 0.99945-j 0.0099757 (complex conjugate pole)

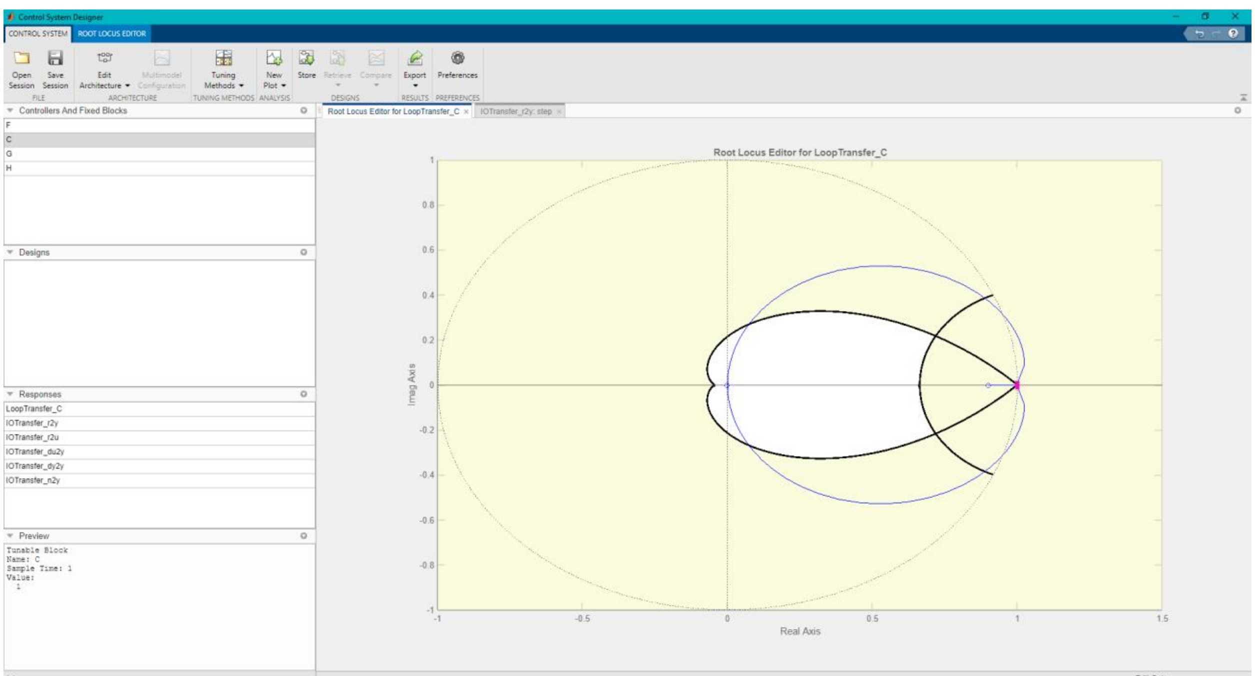
Controller Poles:

Pole at z = 0.99005 (real pole)

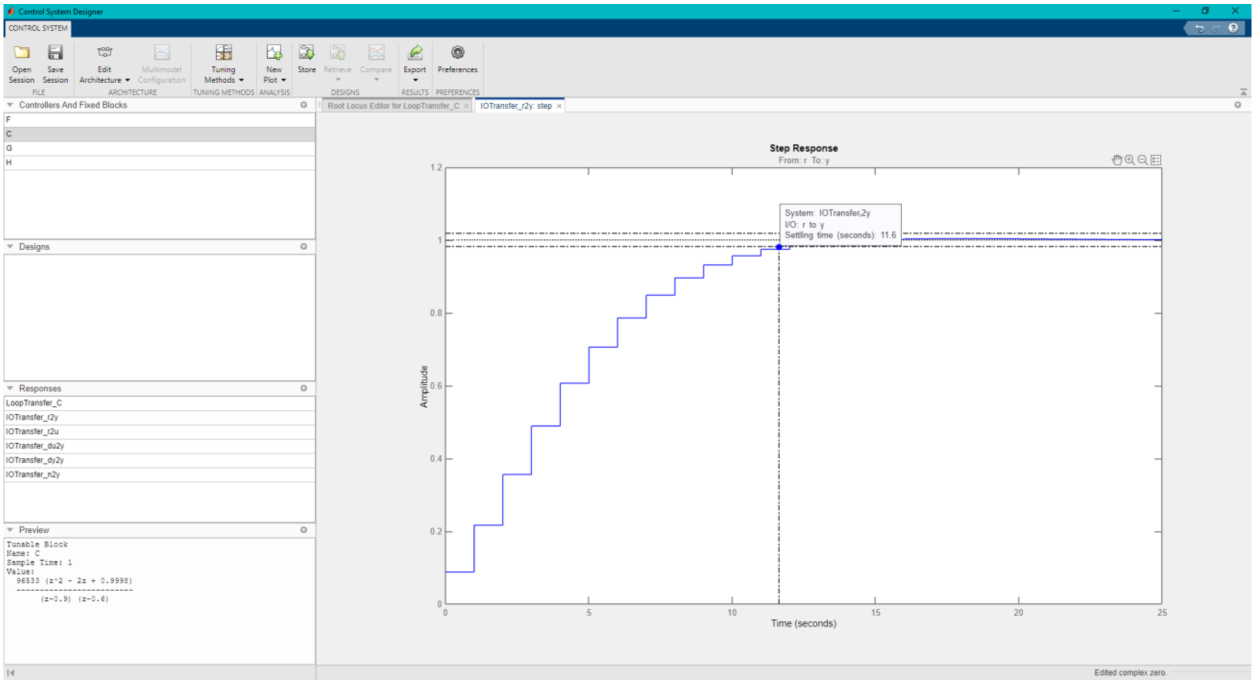
Pole at z = 0.99304 (real pole)



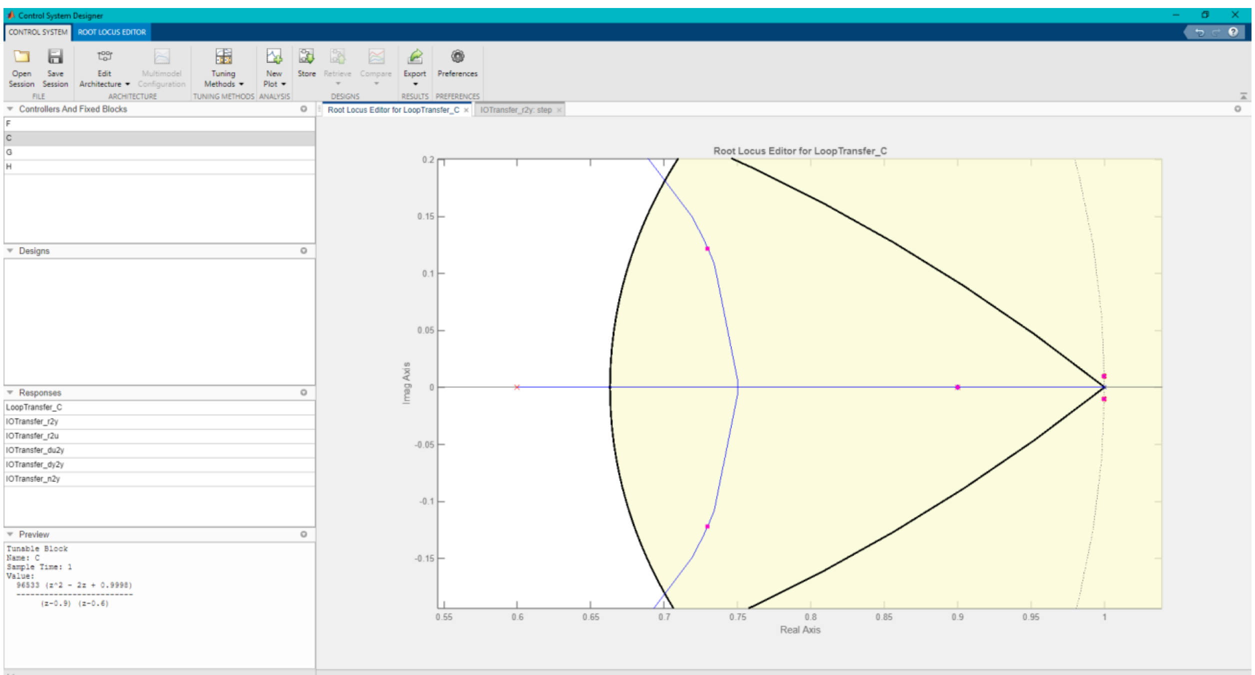
DESIGN REGION



STEP RESPONSE

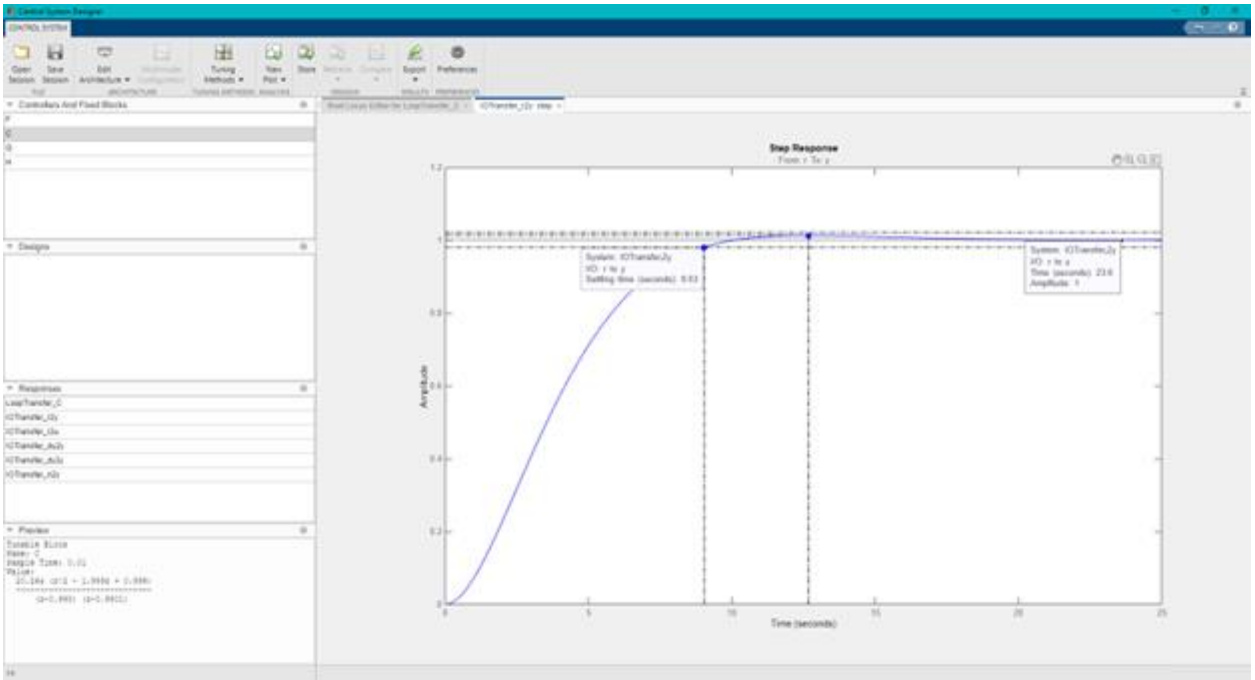


ROOT-LOCUS AFTER COMPENSATION

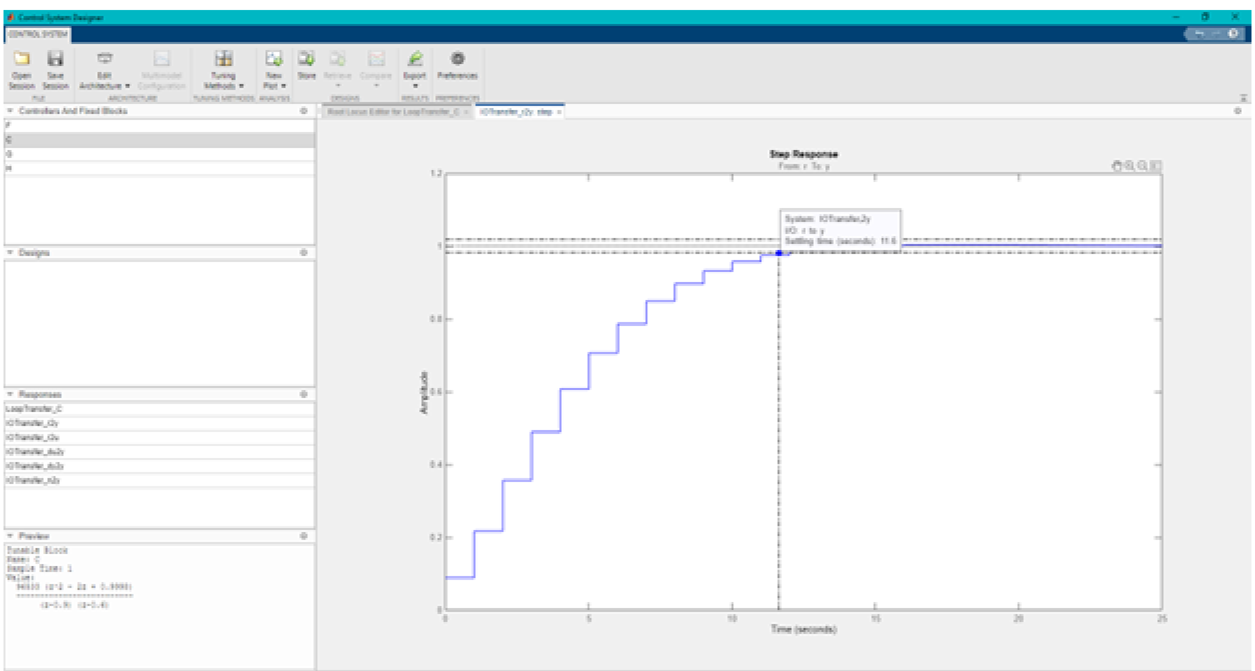


4. Compare the results due to the design by emulation to those obtained by the direct design. Comment on the results.

RESULTS FROM DIRECT DIGITAL DESIGN



RESULTS FROM EMULATION



COMMENT

One advantage of DDC is that it provides real-time control of the process being controlled, which can be important in applications where rapid response times are necessary. Additionally, DDC allows for more precise control of the process, as the controller can continuously monitor and adjust the process based on real-time data.

Emulation techniques, on the other hand, have the advantage of being more flexible and less expensive than DDC. They can be used to simulate a wide range of processes and control algorithms, allowing designers to test and refine their designs without the need for expensive hardware. Emulation can also be used to quickly evaluate different control strategies and to optimize the performance of existing control systems.

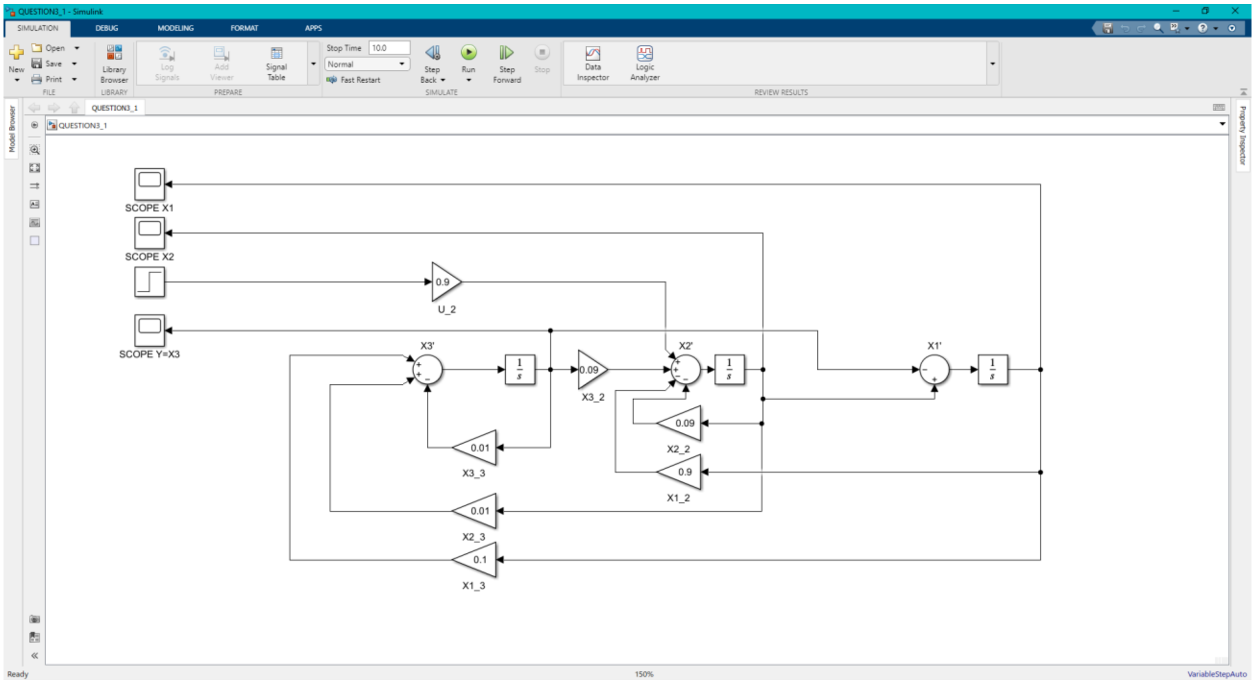
Continuous State Space Representation

1. **Using the continuous system dynamics given, develop and Simulink model for open loop system. Draw the system states (Call this representation “rep A”).**

System state space representation (repA)

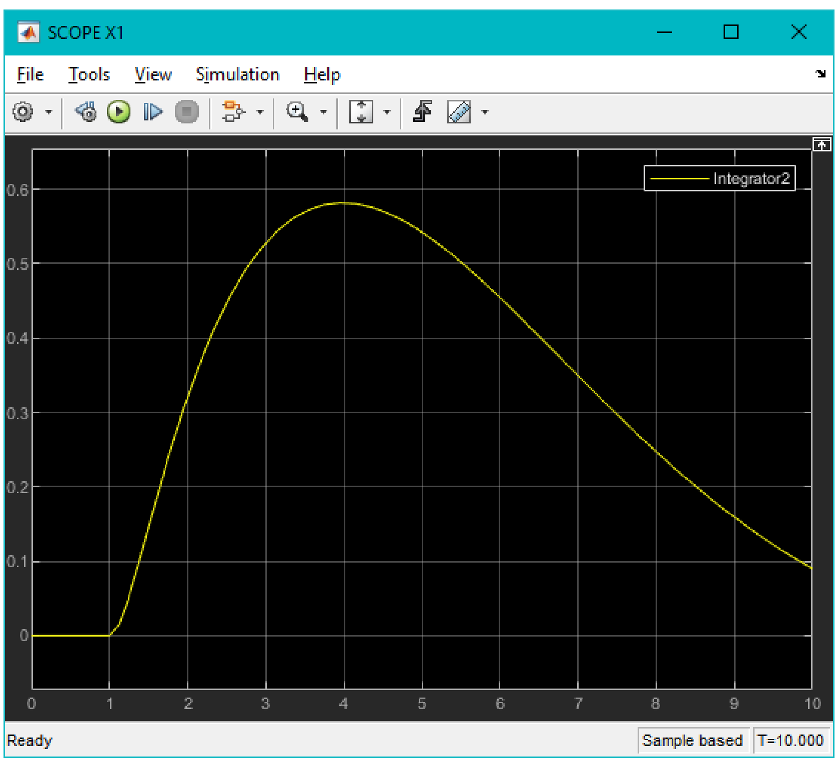
```
ss =

  A =
           x1     x2     x3
   x1       0      1     -1
   x2    -0.9  -0.09   0.09
   x3     0.1   0.01  -0.01

  B =
          u1
   x1      0
   x2    0.9
   x3      0

  C =
         x1  x2  x3
   y1     0   0   1

  D =
         u1
   y1     0

Continuous-time state-space model.
```

Simulink model

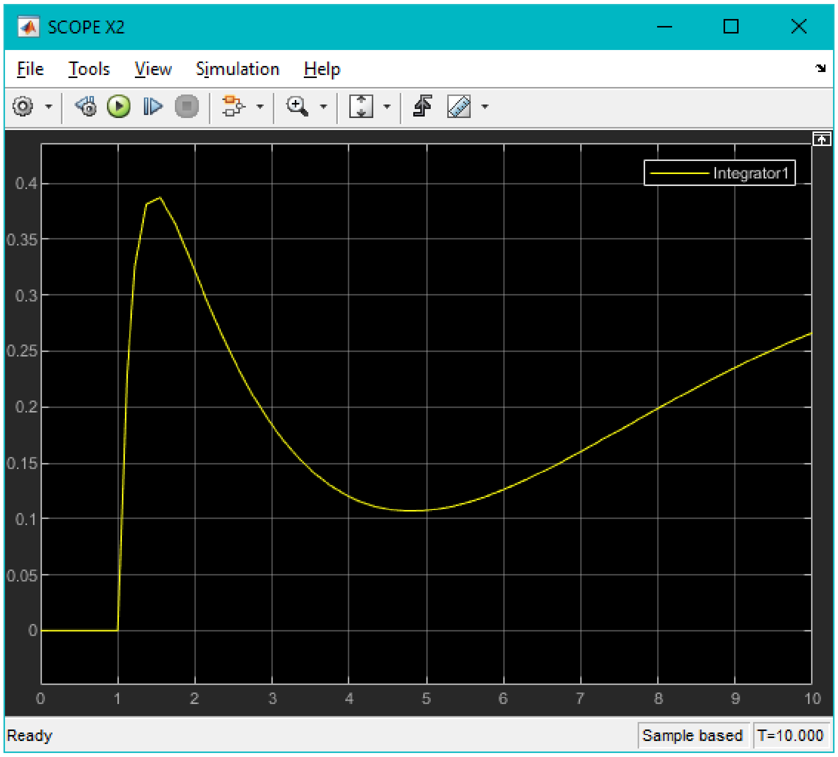


SYSTEM STATES

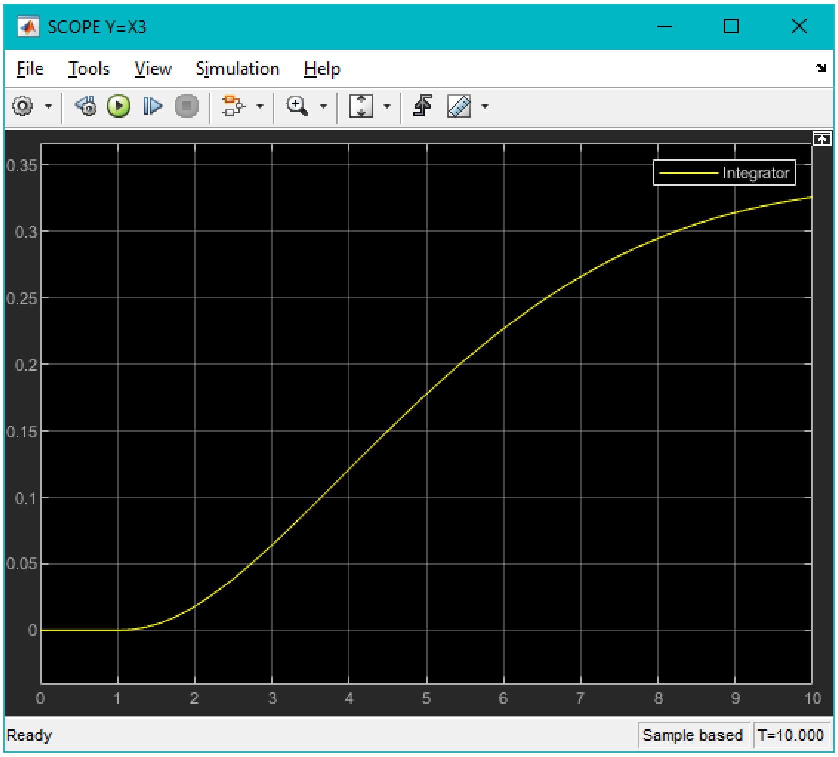
X1



X2


SCOPE X2
File
Tools
View
Simulation
Help
Integrator1
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
1
2
3
4
5
6
7
8
9
10
Ready
Sample based
T=10.000

Y=X3



2. Find appropriate state feedback gain to place the poles of the system in suitable places (You can use the same requirements as mentioned in the transfer function approach).

DESIRED POLES

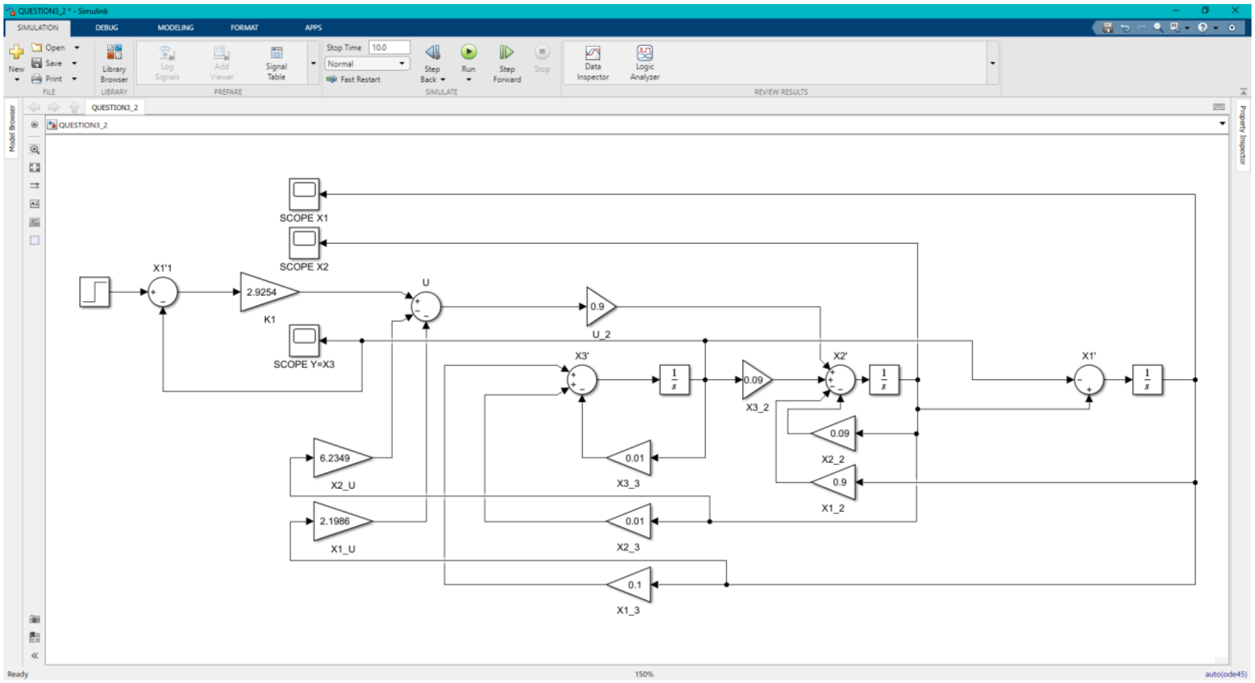
```
p =

  -0.3557 + 0.1590i  -0.3557 - 0.1590i  -5.0000 + 0.0000i
```

CODE

```
>> place(Anew, B, p)

ans =

    2.9254    6.2349    2.1986
```

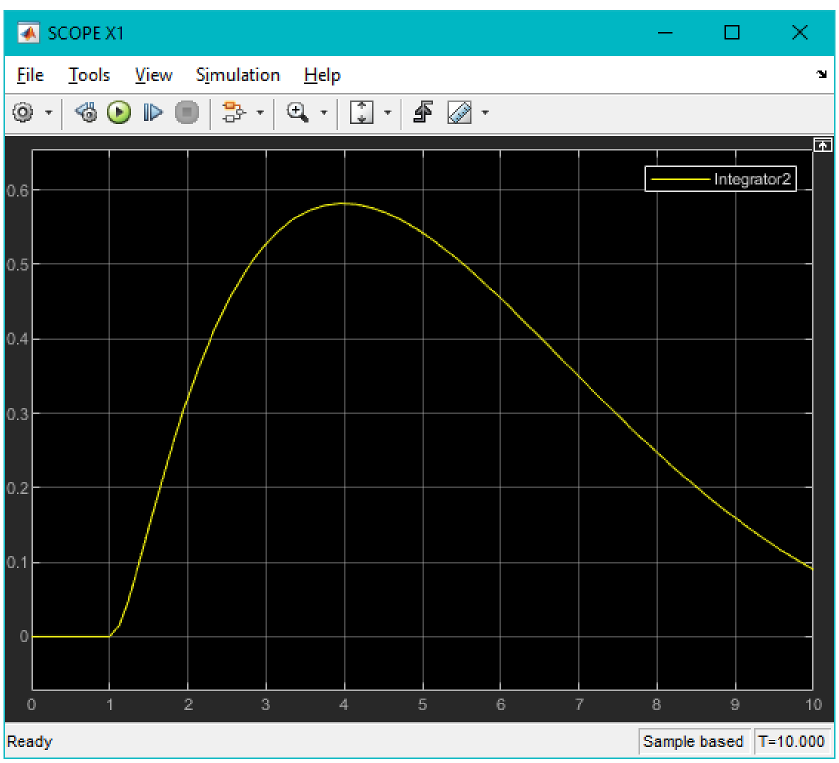
3. Implement the feedback signals using the Simulink. Draw the states and output versus time. Comment on the results.

SIMLINK MODEL WITH STATE FEEDBACK

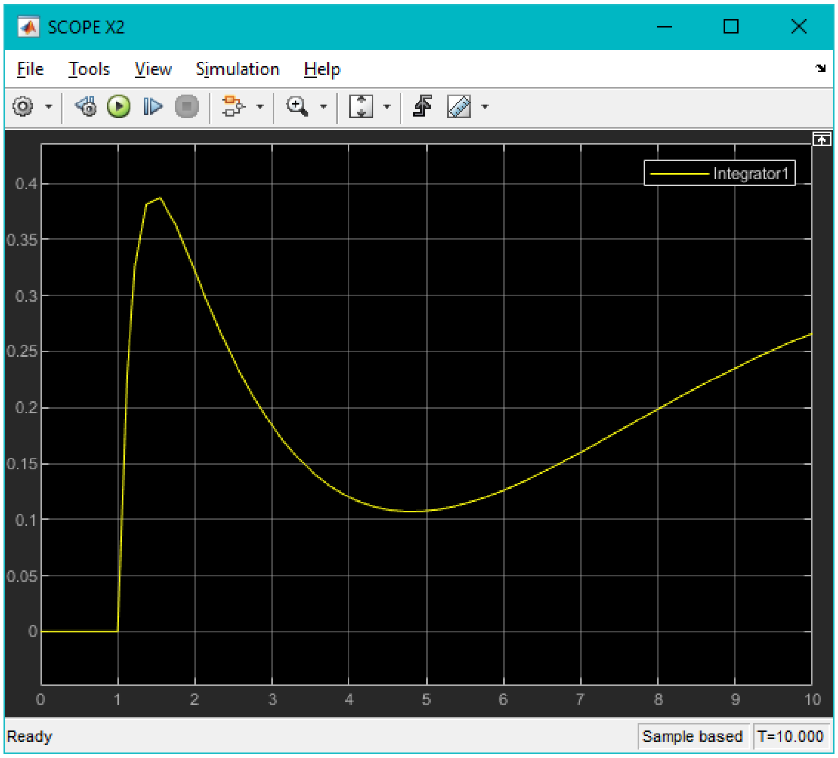


SYSTEM STATES

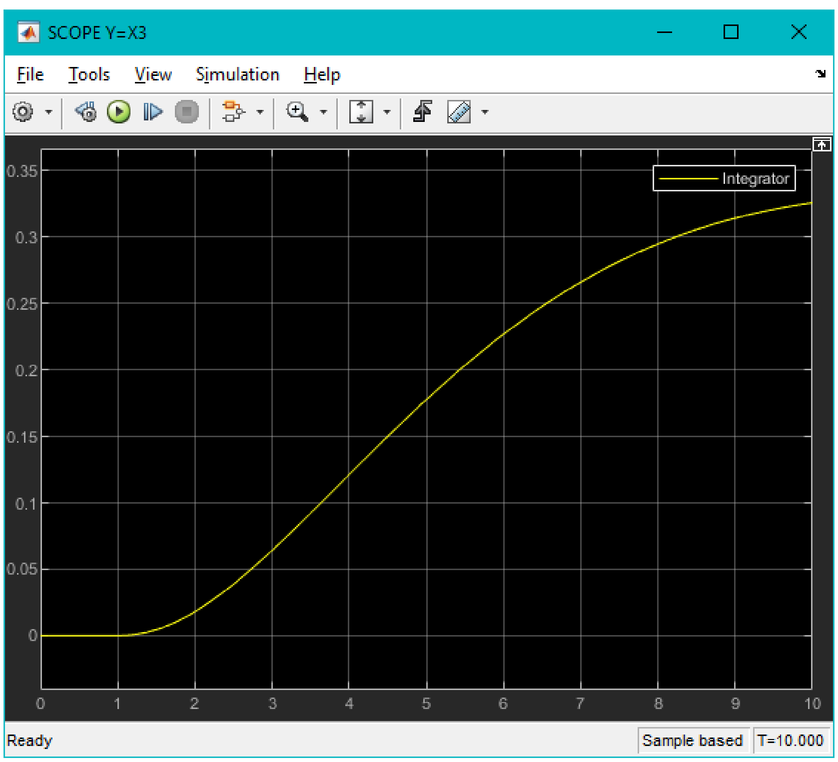
X1



X2


SCOPE X2
File Tools View Simulation Help
Integrator1
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0 1 2 3 4 5 6 7 8 9 10
Ready
Sample based
T=10.000

Y=X3


SCOPE Y=X3
File Tools View Simulation Help
Integrator
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0 1 2 3 4 5 6 7 8 9 10
Ready
Sample based
T=10.000

COMMENT

When we used pole placement method to change the system feedback, we obtained better system characteristics in terms of MOS% and Natural frequency.

Discrete State Space Representation

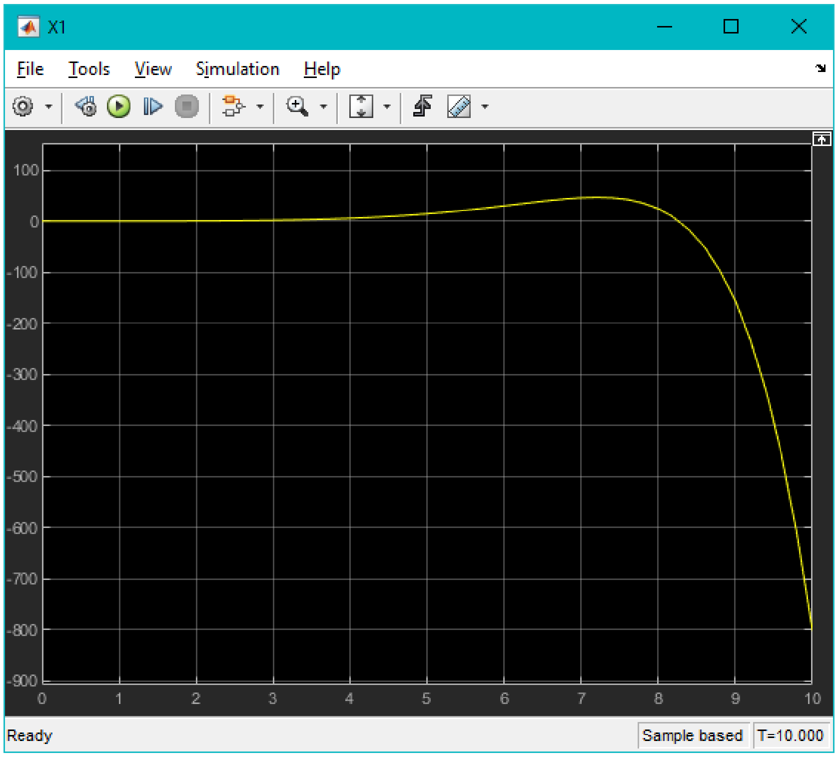
1. Choose a suitable sampling period and find the discrete form for "rep A" (Call this representation "rep B"). Write the state space representation in the controllable canonical form (Call this representation "rep C"). Draw the system states for both representations for open loop case. Comment on the results.

DISCRETIZATION

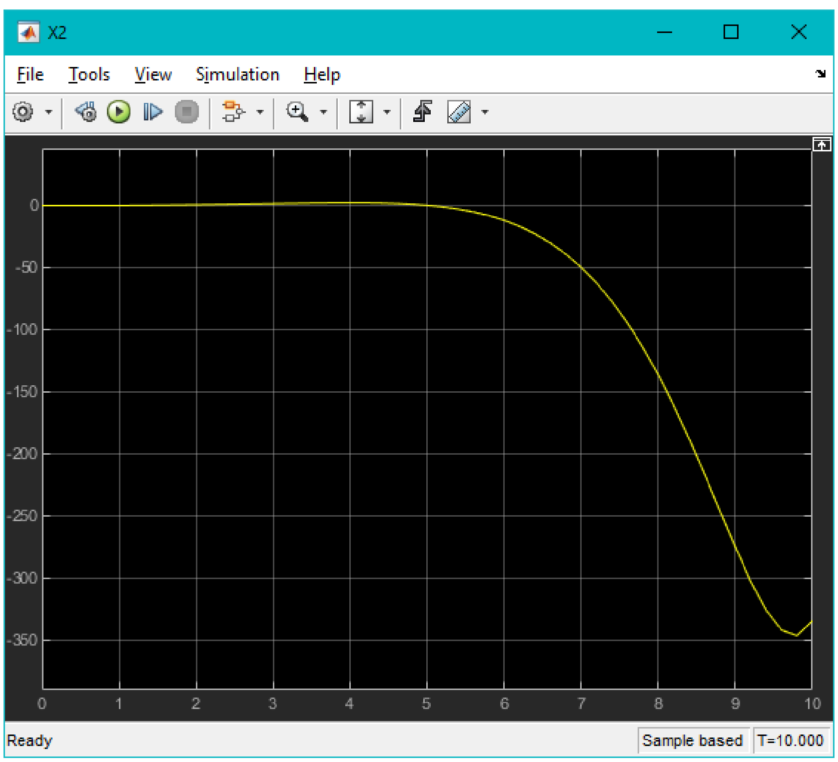
```
>> c2d(SS, 0.5)

ans =

  A =
             x1        x2        x3
   x1    0.8796    0.4676   -0.4676
   x2   -0.4209    0.8495    0.1505
   x3   0.04676   0.01672    0.9833

  B =
              u1
   x1     0.1084
   x2     0.4238
   x3   0.002913

  C =
        x1  x2  x3
   y1    0   0   1

  D =
        u1
   y1    0

Sample time: 0.5 seconds
Discrete-time state-space model.
```

SYSTEM STATES

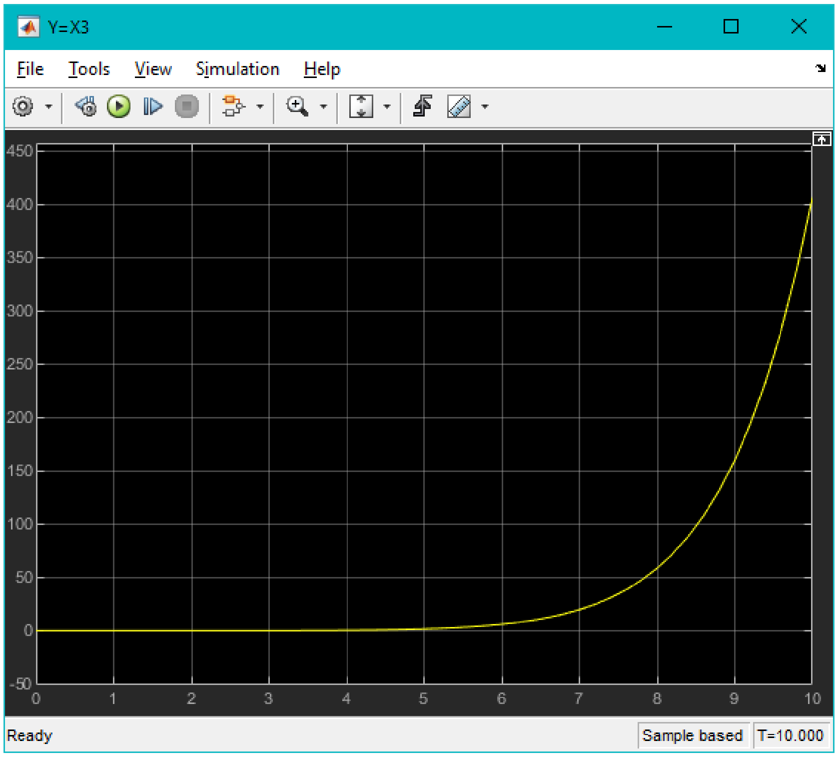
X1



X2



X3=Y

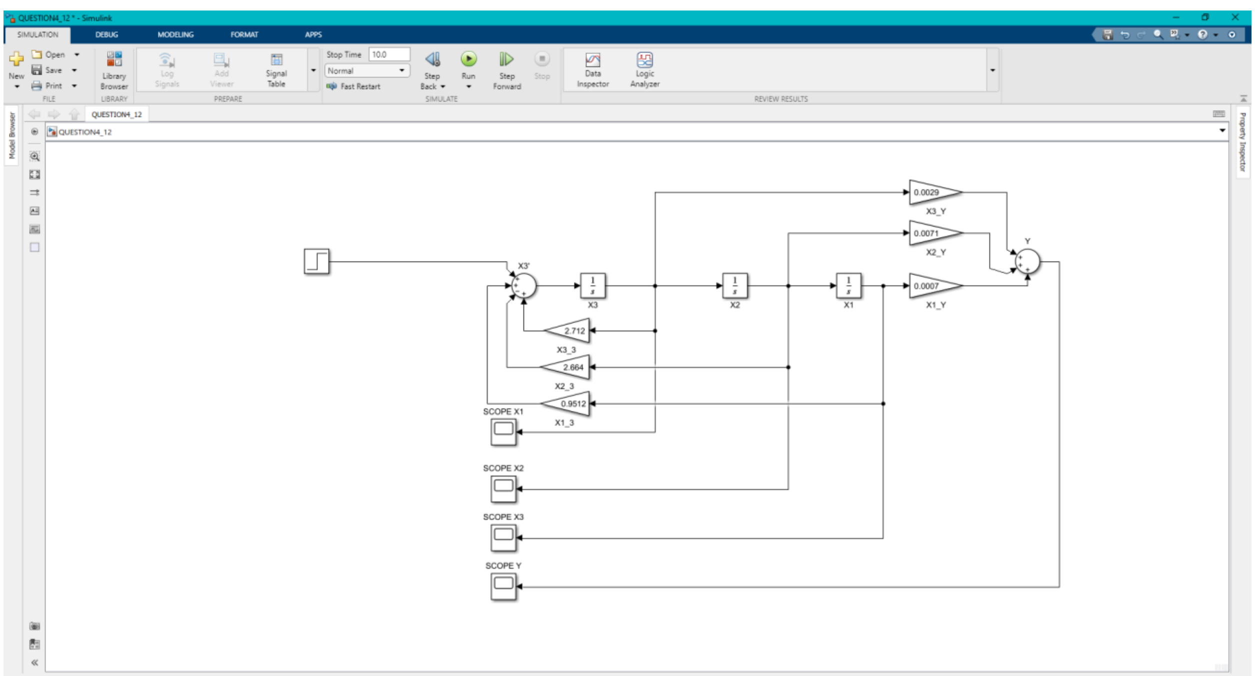


Y=X3
File
Tools
View
Simulation
Help
450
400
350
300
250
200
150
100
50
0
-50
0
1
2
3
4
5
6
7
8
9
10
Ready
Sample based
T=10.000

CONTROLLABLE CANONICAL FORM

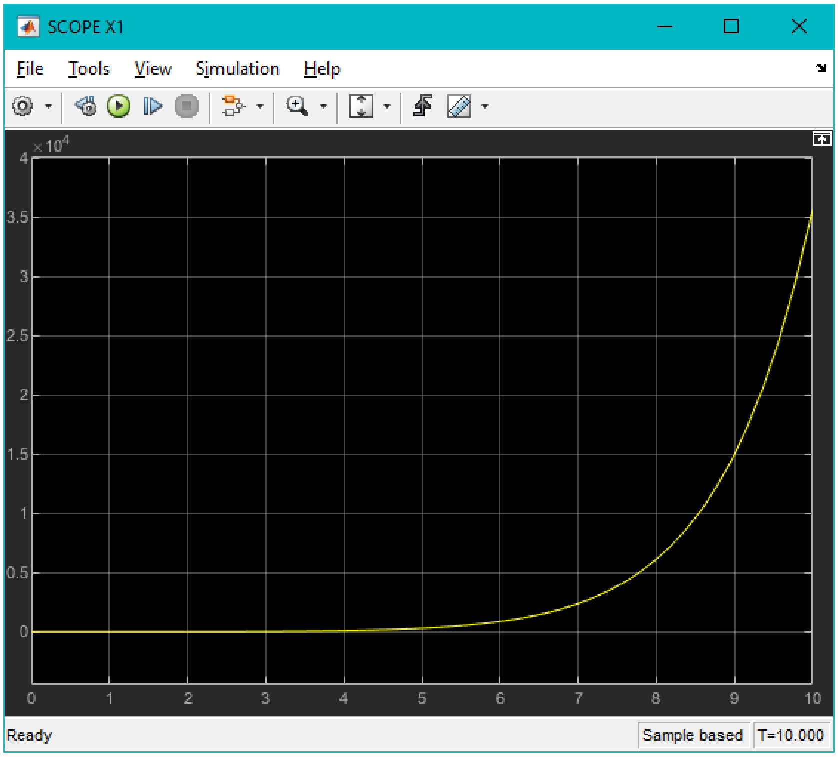
```
SS_CCF_disc =

  A =
             x1       x2       x3
   x1         0        1        0
   x2         0        0        1
   x3    0.9512   -2.664   -2.712

  B =
         u1
   x1     0
   x2     0
   x3     1

  C =
             x1       x2       x3
   y1    0.0007   0.0071   0.0029

  D =
         u1
   y1     0

Continuous-time state-space model.
```

SIMULINK MODEL

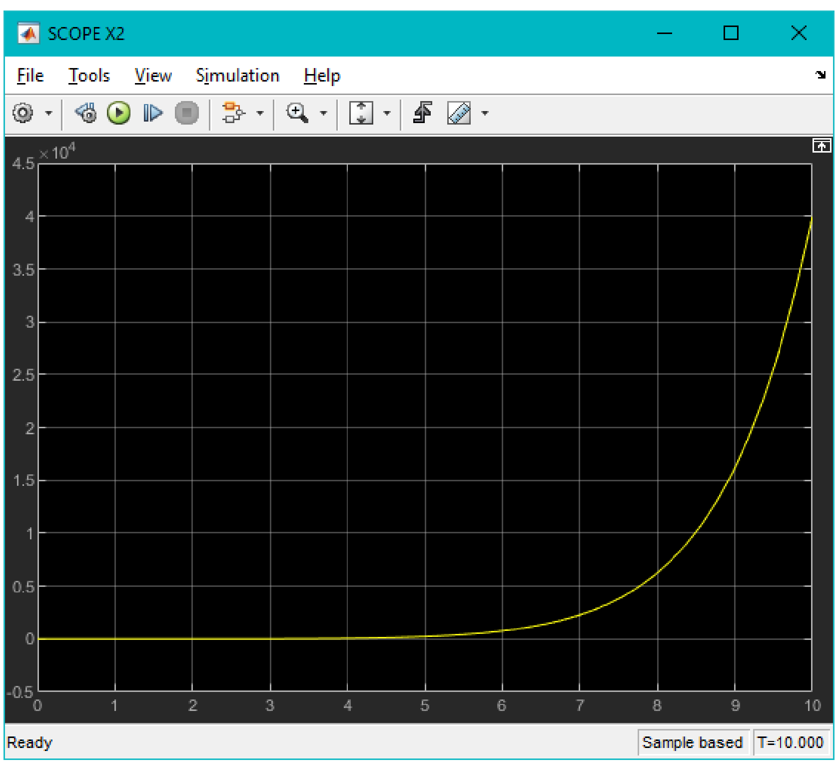


SYSTEM STATES

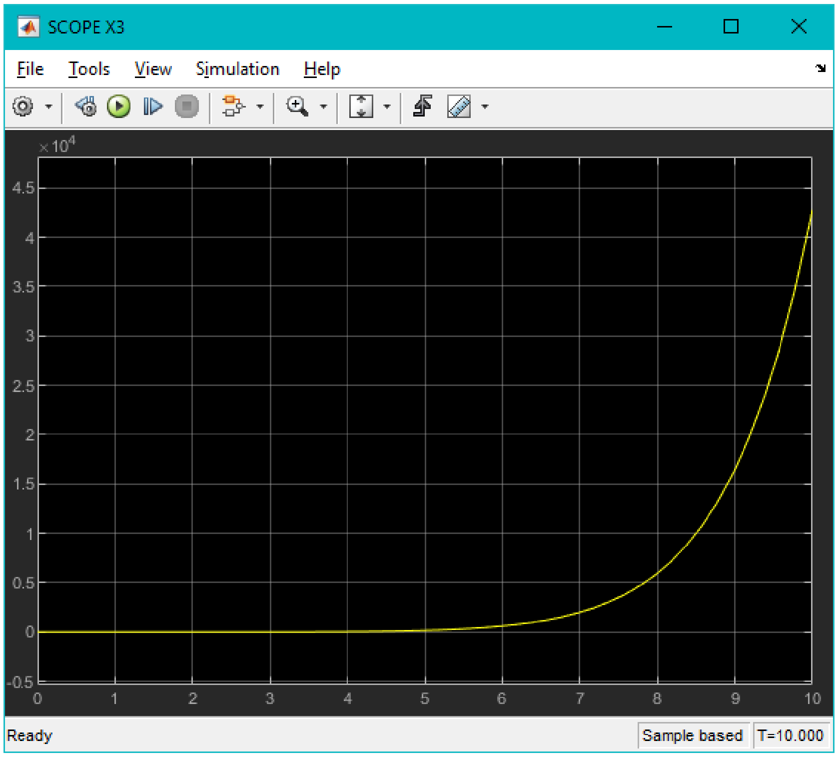
X1



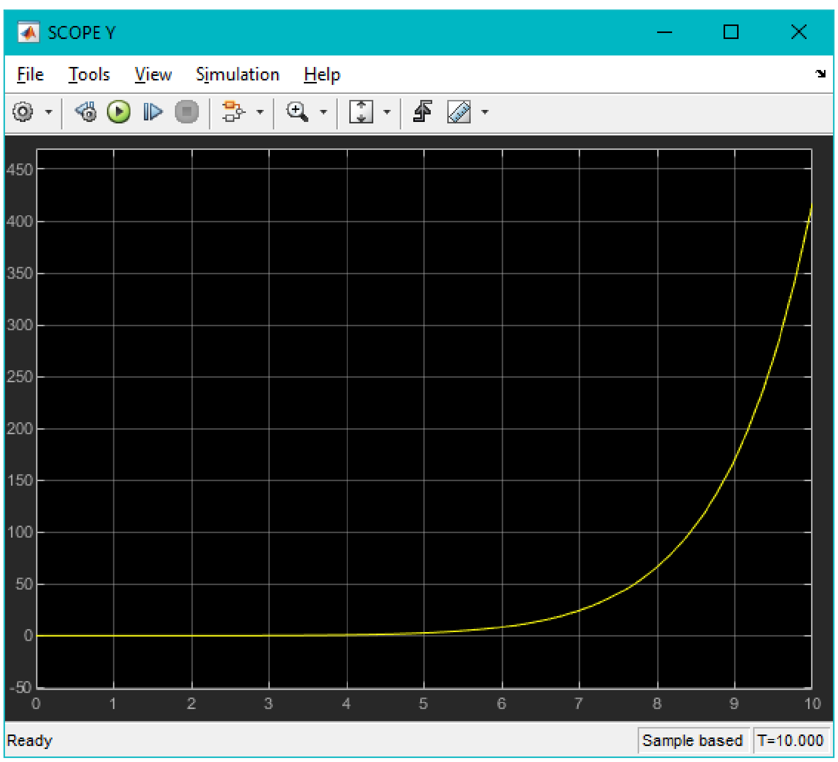
X2



X3


SCOPE X3
File Tools View Simulation Help
×10^4
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0 1 2 3 4 5 6 7 8 9 10
Ready
Sample based
T=10.000

Y


SCOPE Y
File Tools View Simulation Help
450
400
350
300
250
200
150
100
50
0
-50
0 1 2 3 4 5 6 7 8 9 10
Ready
Sample based
T=10.000

COMMENT

We notice that for both systems repB and repA, no matter how we chose a different sampling period, the response is unstable.

2. Design state feedback vector for "rep B" to achieve same transient response specifications as before and use reference manipulation gain to achieve zero steady state error. Implement your controller on the continuous-time model (rep A). Draw the states and output versus time. Comment on the results.

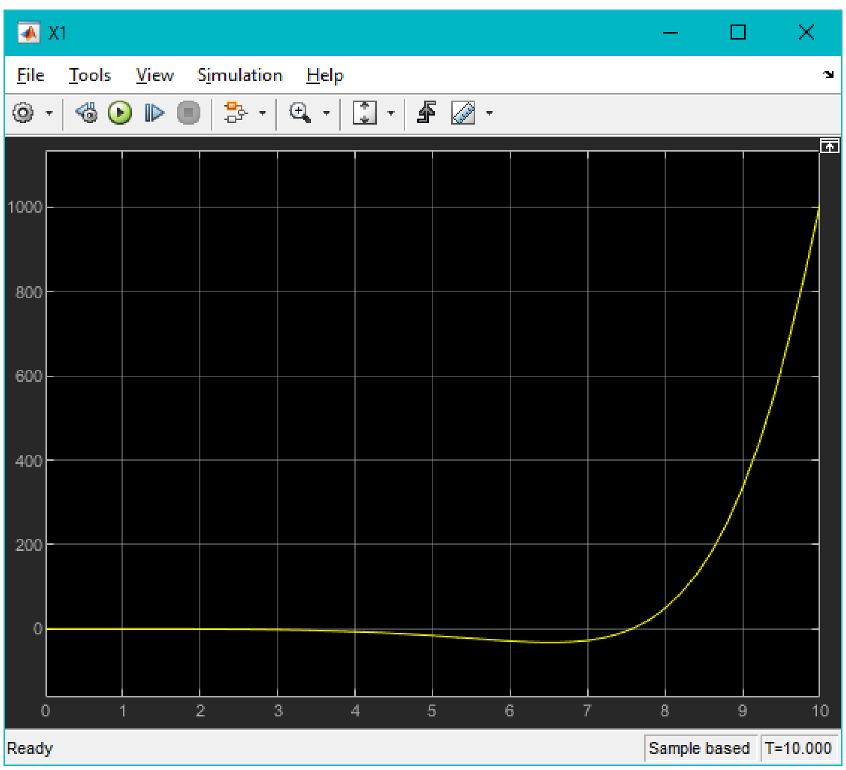
repB

POLE PLACEMENT

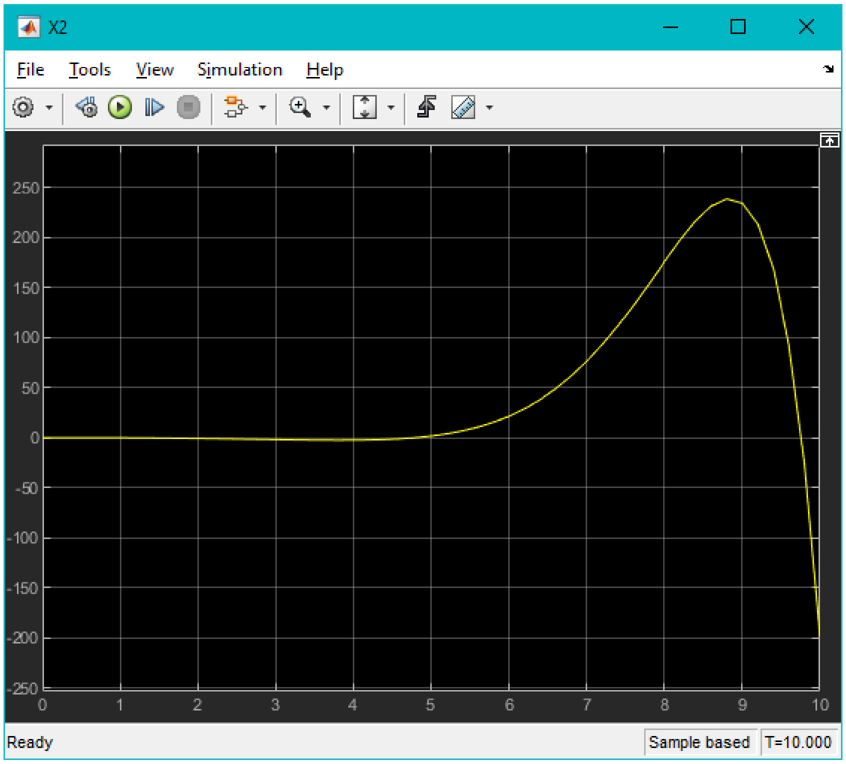
```
>> p_disc=[0.997+0.00248i 0.997-0.00248i 0.997];
place(A_disc, B_disc, p_disc)

ans =

   -1.1110   -0.3758    0.3758
```

RESPONSE

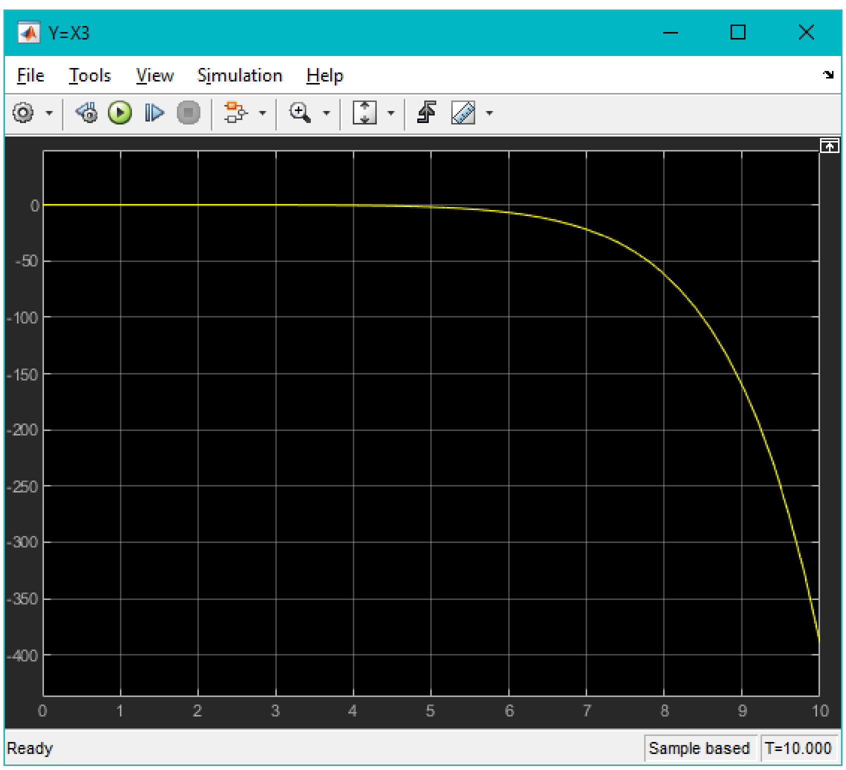
X1



X2

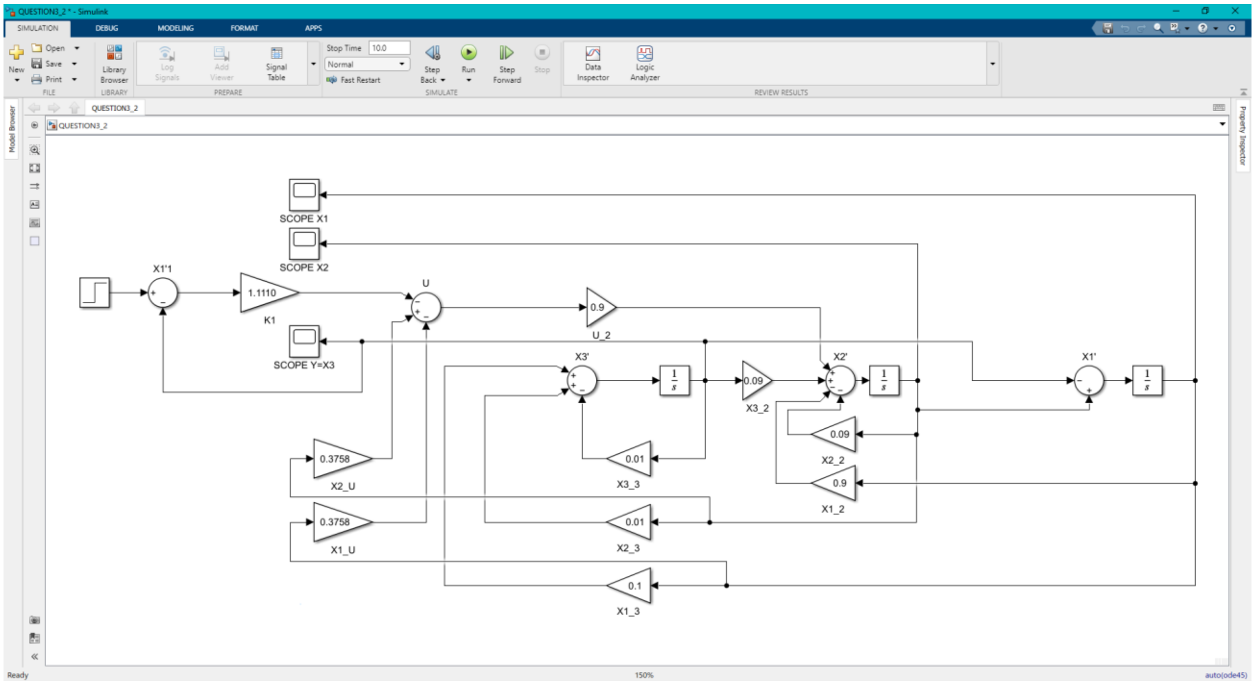

X2
File
Tools
View
Simulation
Help
250
200
150
100
50
0
-50
-100
-150
-200
-250
0 1 2 3 4 5 6 7 8 9 10
Ready
Sample based
T=10.000

X3=Y


Y=X3
File
Tools
View
Simulation
Help
0
-50
-100
-150
-200
-250
-300
-350
-400
0 1 2 3 4 5 6 7 8 9 10
Ready
Sample based
T=10.000

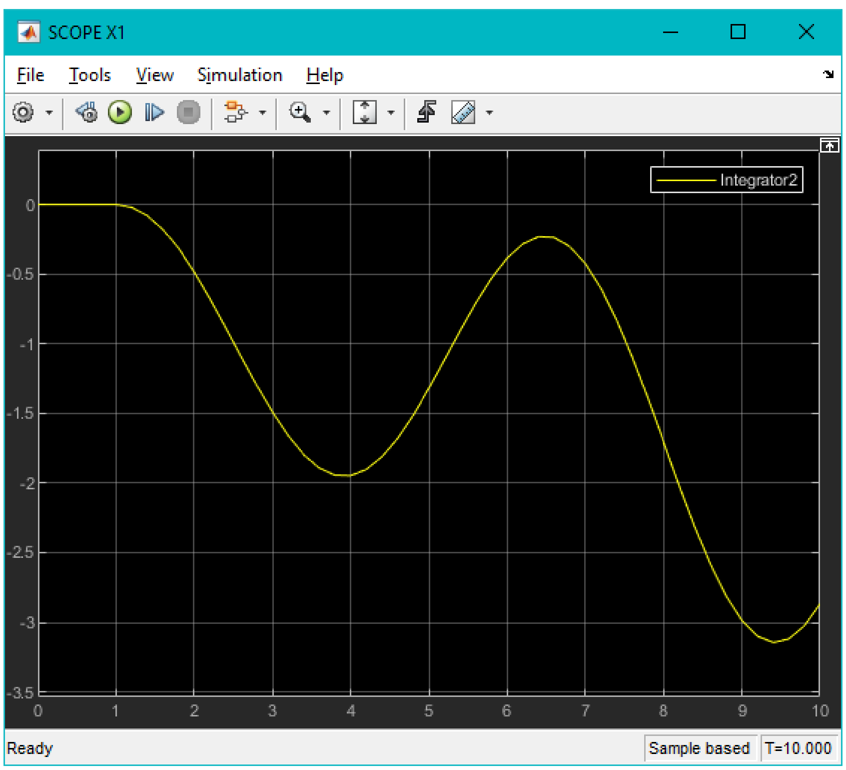
repA

MODEL

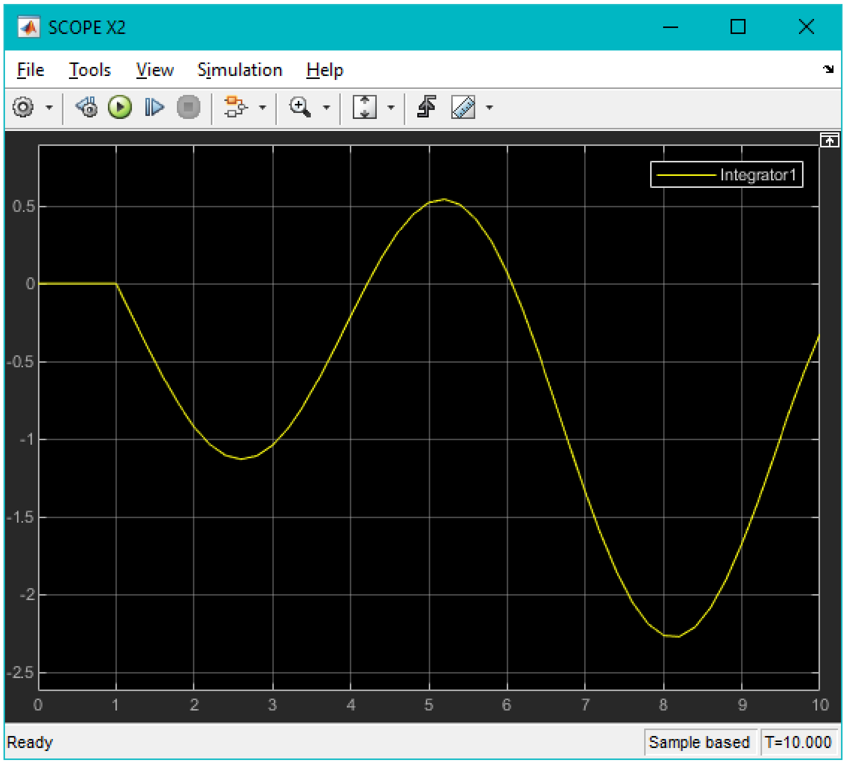


RESPONSE

X1

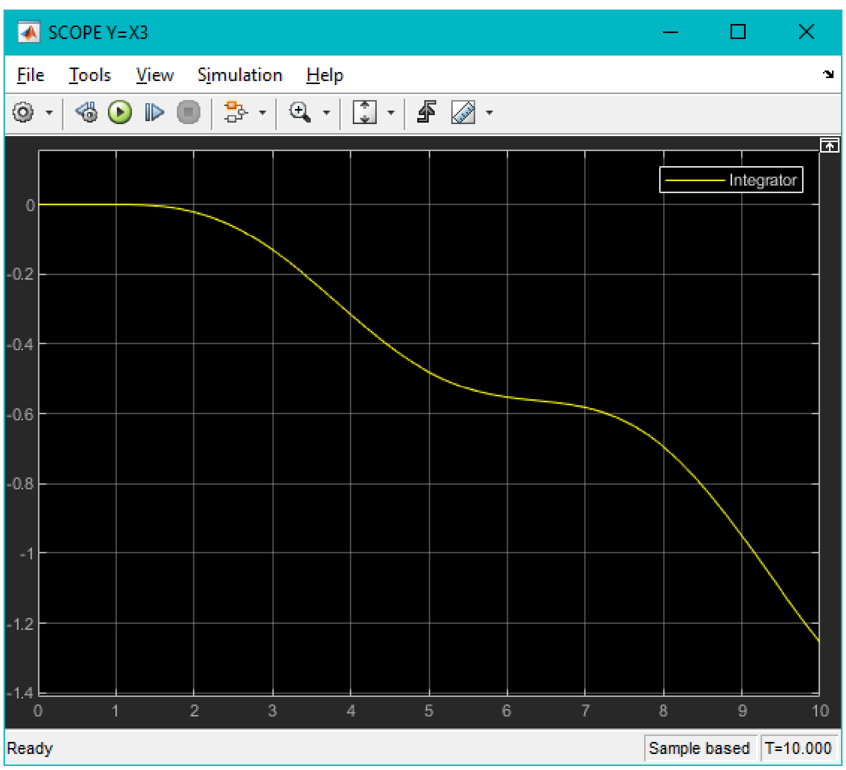


X2



SCOPE X2
File
Tools
View
Simulation
Help
Integrator1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
1
2
3
4
5
6
7
8
9
10
Ready
Sample based
T=10.000

Y=X3



SCOPE Y=X3
File
Tools
View
Simulation
Help
Integrator
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
0
1
2
3
4
5
6
7
8
9
10
Ready
Sample based
T=10.000

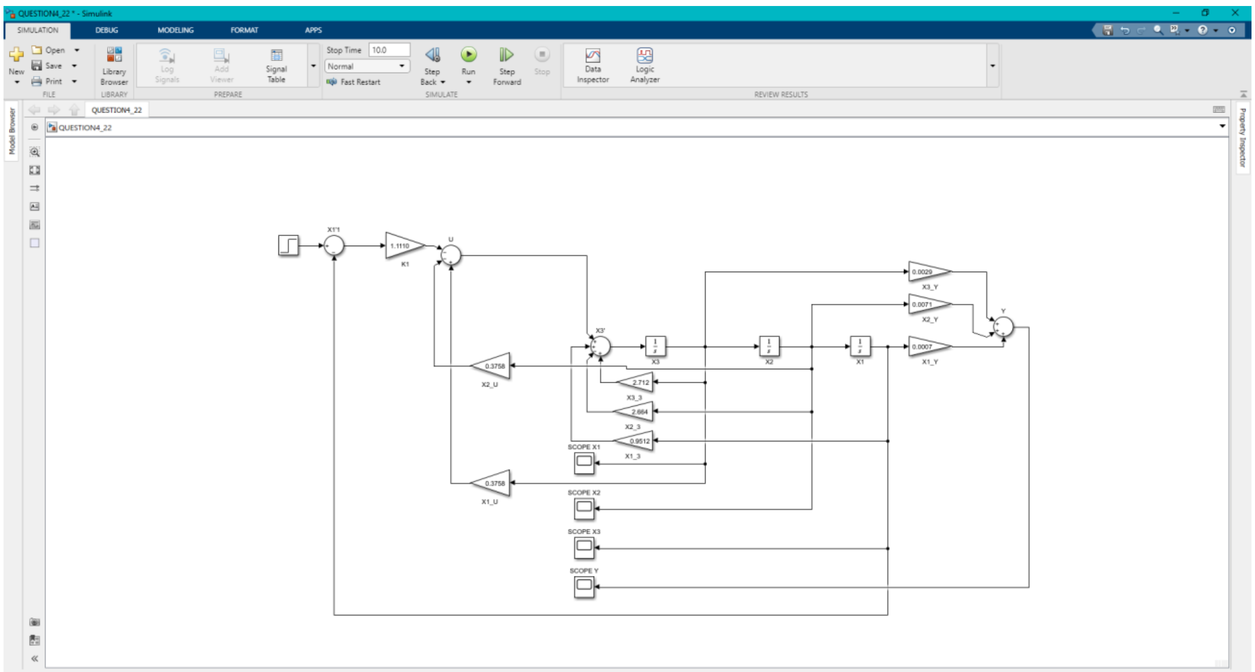
COMMENT

Unfortunately we were unable to reach a stable system after using state feedback gain, it was unknown why the system could not be stabilized: whether it was an unstabilizable system or there was an error in design. We assume that there is an error in the Simulink model which prevented the system from reaching stability.

3. Implement same controller designed on (2) using the states measurements from (rep C). Draw the system states, and comment on the results.

repC

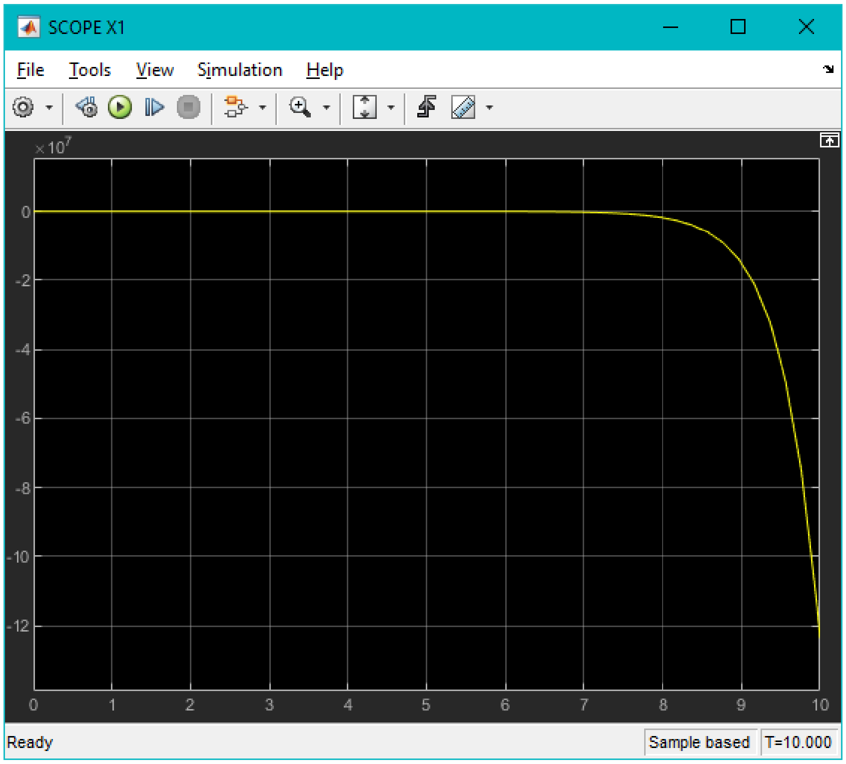
MODEL



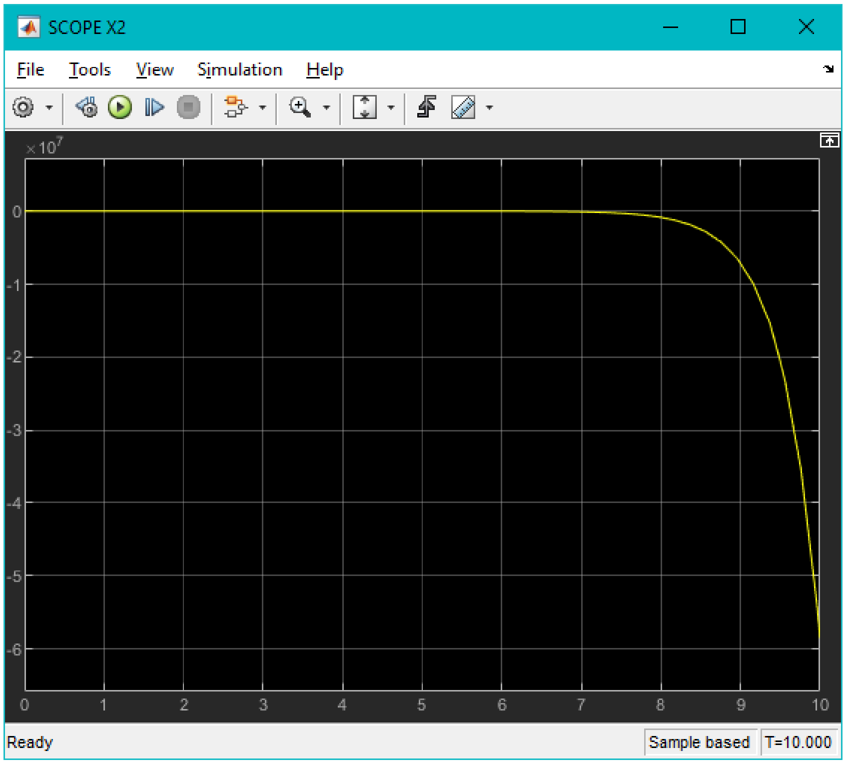
POLE PLACEMENT

RESPONSE

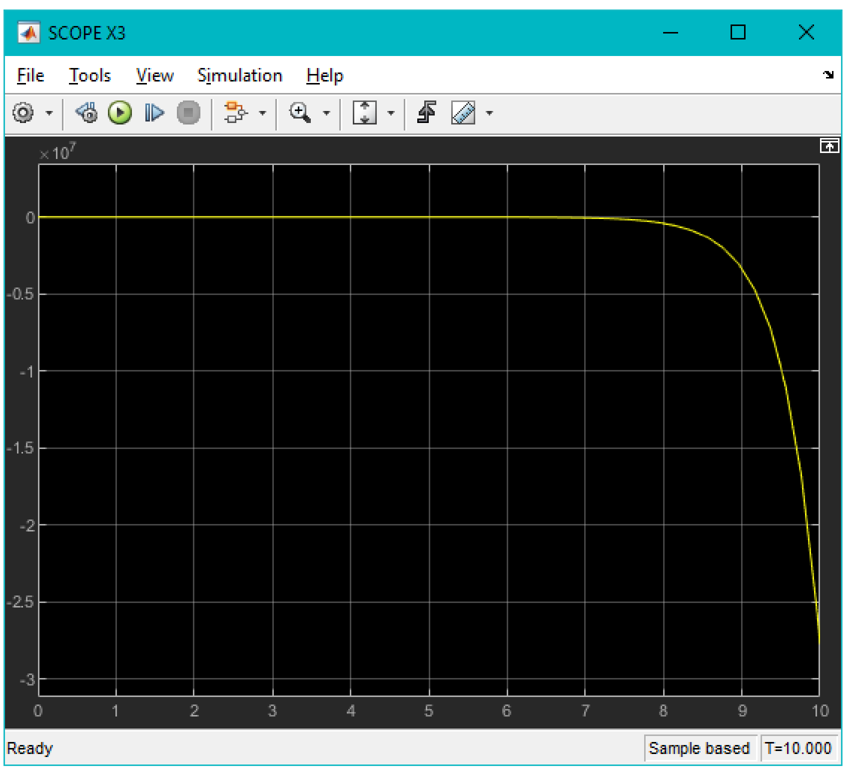
X1



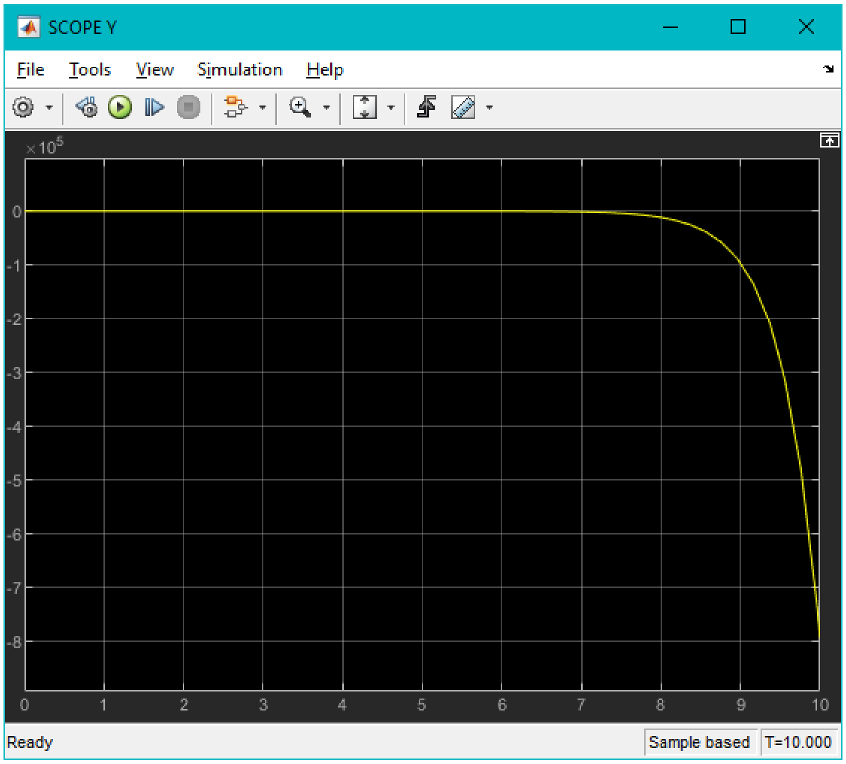
X2


SCOPE X2
File Tools View Simulation Help
×10^7
0
-1
-2
-3
-4
-5
-6
0 1 2 3 4 5 6 7 8 9 10
Ready
Sample based
T=10.000

X3


SCOPE X3
File Tools View Simulation Help
×10^7
0
-0.5
-1
-1.5
-2
-2.5
-3
0 1 2 3 4 5 6 7 8 9 10
Ready
Sample based
T=10.000

Y



COMMENT

Again, the state feedback gain could not stabilize the controllable canonical form of the system.

4. Using rep A and the controller designed in 2, assuming that the only measurement available is the output $\omega 2$, design an appropriate observer for the system and implement the state feedback. Draw the system states (starting from initial condition $\boldsymbol{x}$ = $[3\ 5\ 0]^T$ and the estimated states. Comment on the results.

5. For the controller in 4, if the sampling interval is changed to double its value, what will be the effect of this change on the feed-back system response with the same state feedback gain? Study this effect and draw the system states.