Contents

[CONTINUOUS SYSTEM ANALYSIS AND CONTROL DESIGN USING THE TRANSFER FUNCTION APPROACH 5](#_Toc135475171)

[1. Find the transfer function . Plot the root locus and find the stability range of a gain controller. 5](#_Toc135475172)

[MATLAB CODE TO OBTAIN THE TRANSFER FUNCTION OF A SYSTEM USING ITS STATE SPACE REPRESENTATION: 5](#_Toc135475173)

[OUTPUT 5](#_Toc135475174)

[ROOT LOCUS FOR THE TRANSFER FUNCTION AND STABILITY RANGE: 6](#_Toc135475175)

[POLES & ZEROS 6](#_Toc135475176)

[STABILTY RANGE 6](#_Toc135475177)

[SYSTEM ARCHITECTURE 7](#_Toc135475178)

[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](#_Toc135475179)

[POLES & ZEROS 7](#_Toc135475180)

[DESIGN REQUIREMENTS 8](#_Toc135475181)

[DESIGN REGION 8](#_Toc135475182)

[STEP RESPONSE 8](#_Toc135475183)

[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](#_Toc135475184)

[POLES & ZEROS 9](#_Toc135475185)

[BODE PLOT 10](#_Toc135475186)

[STEP RESPONSE 11](#_Toc135475187)

[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](#_Toc135475188)

[MODEL 11](#_Toc135475189)

[DISTURBANCE PLOT 12](#_Toc135475190)

[STEP INPUT 13](#_Toc135475191)

[OUTPUT SIGNAL & CONTROLLER ACTION (ROOT LOCUS) & DISTURBANCE PULSE 14](#_Toc135475192)

[OUTPUT SIGNAL & CONTROLLER ACTION (BODE PLOT) & DISTURBANCE PULSE 15](#_Toc135475193)

[Digital Control Design Using the Transfer Function Approach 16](#_Toc135475194)

[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](#_Toc135475195)

[POLES & ZEROS 18](#_Toc135475196)

[STABILTY RANGE 18](#_Toc135475197)

[POLES & ZEROS 19](#_Toc135475198)

[STABILTY RANGE 19](#_Toc135475199)

[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](#_Toc135475200)

[DESIGN REQUIREMENTS 20](#_Toc135475201)

[DESIGN REGION 21](#_Toc135475202)

[STEP RESPONSE 21](#_Toc135475203)

[ROOT-LOCUS AFTER COMPENSATION 22](#_Toc135475204)

[3. Repeat in question #2 using a design by emulation. 22](#_Toc135475205)

[POLES & ZEROS 23](#_Toc135475206)

[STABILTY RANGE 23](#_Toc135475207)

[DESIGN REGION 24](#_Toc135475208)

[STEP RESPONSE 24](#_Toc135475209)

[ROOT-LOCUS AFTER COMPENSATION 25](#_Toc135475210)

[4. Compare the results due to the design by emulation to those obtained by the direct design. Comment on the results. 26](#_Toc135475211)

[COMMENT 27](#_Toc135475212)

[Continuous State Space Representation 28](#_Toc135475213)

[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](#_Toc135475214)

[System state space representation (**repA**) 28](#_Toc135475215)

[Simulink model 28](#_Toc135475216)

[29](#_Toc135475217)

[SYSTEM STATES 29](#_Toc135475218)

[X1 29](#_Toc135475219)

[X2 30](#_Toc135475220)

[Y=X3 31](#_Toc135475221)

[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](#_Toc135475222)

[DESIRED POLES 31](#_Toc135475223)

[CODE 31](#_Toc135475224)

[3. Implement the feedback signals using the Simulink. Draw the states and output versus time. Comment on the results. 32](#_Toc135475225)

[SIMLINK MODEL WITH STATE FEEDBACK 32](#_Toc135475226)

[SYSTEM STATES 32](#_Toc135475227)

[X1 32](#_Toc135475228)

[COMMENT 34](#_Toc135475229)

[Discrete State Space Representation 34](#_Toc135475230)

[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](#_Toc135475231)

[DISCRETIZATION 34](#_Toc135475232)

[SYSTEM STATES 35](#_Toc135475233)

[X1 35](#_Toc135475234)

[X2 35](#_Toc135475235)

[X3=Y 36](#_Toc135475236)

[CONTROLLABLE CANONICAL FORM 37](#_Toc135475237)

[SIMULINK MODEL 37](#_Toc135475238)

[SYSTEM STATES 38](#_Toc135475239)

[X1 38](#_Toc135475240)

[X2 38](#_Toc135475241)

[X3 39](#_Toc135475242)

[Y 39](#_Toc135475243)

[COMMENT 40](#_Toc135475244)

[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](#_Toc135475245)

[repB 40](#_Toc135475246)

[POLE PLACEMENT 40](#_Toc135475247)

[RESPONSE 40](#_Toc135475248)

[X1 40](#_Toc135475249)

[X2 41](#_Toc135475250)

[X3=Y 41](#_Toc135475251)

[repA 42](#_Toc135475252)

[MODEL 42](#_Toc135475253)

[RESPONSE 42](#_Toc135475254)

[COMMENT 44](#_Toc135475255)

[3. Implement same controller designed on (2) using the states measurements from (rep C). Draw the system states, and comment on the results. 44](#_Toc135475256)

[repC 44](#_Toc135475257)

[MODEL 44](#_Toc135475258)

[POLE PLACEMENT 45](#_Toc135475259)

[RESPONSE 45](#_Toc135475260)

[X1 45](#_Toc135475261)

[X2 46](#_Toc135475262)

[X3 46](#_Toc135475263)

[Y 47](#_Toc135475264)

[COMMENT 47](#_Toc135475265)

[4. Using rep A and the controller designed in 2, assuming that the only measurement available is the output 𝜔2 , 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](#_Toc135475266)

[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](#_Toc135475267)

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

## Find the transfer function . 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

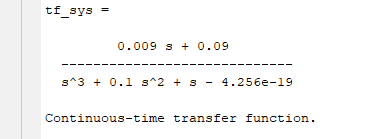
% Convert the state space representation to transfer function

sys = ss(Anew, B, C, D);

tf\_sys = tf(sys);

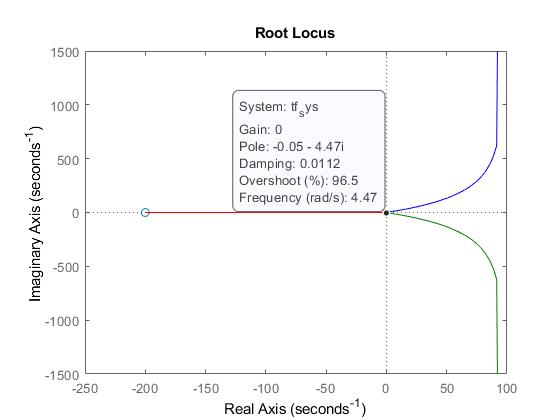
sisotool(tf\_sys)

### OUTPUT



### 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

A screenshot of a computer

Description automatically generated

## 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)

A screenshot of a computer

Description automatically generated

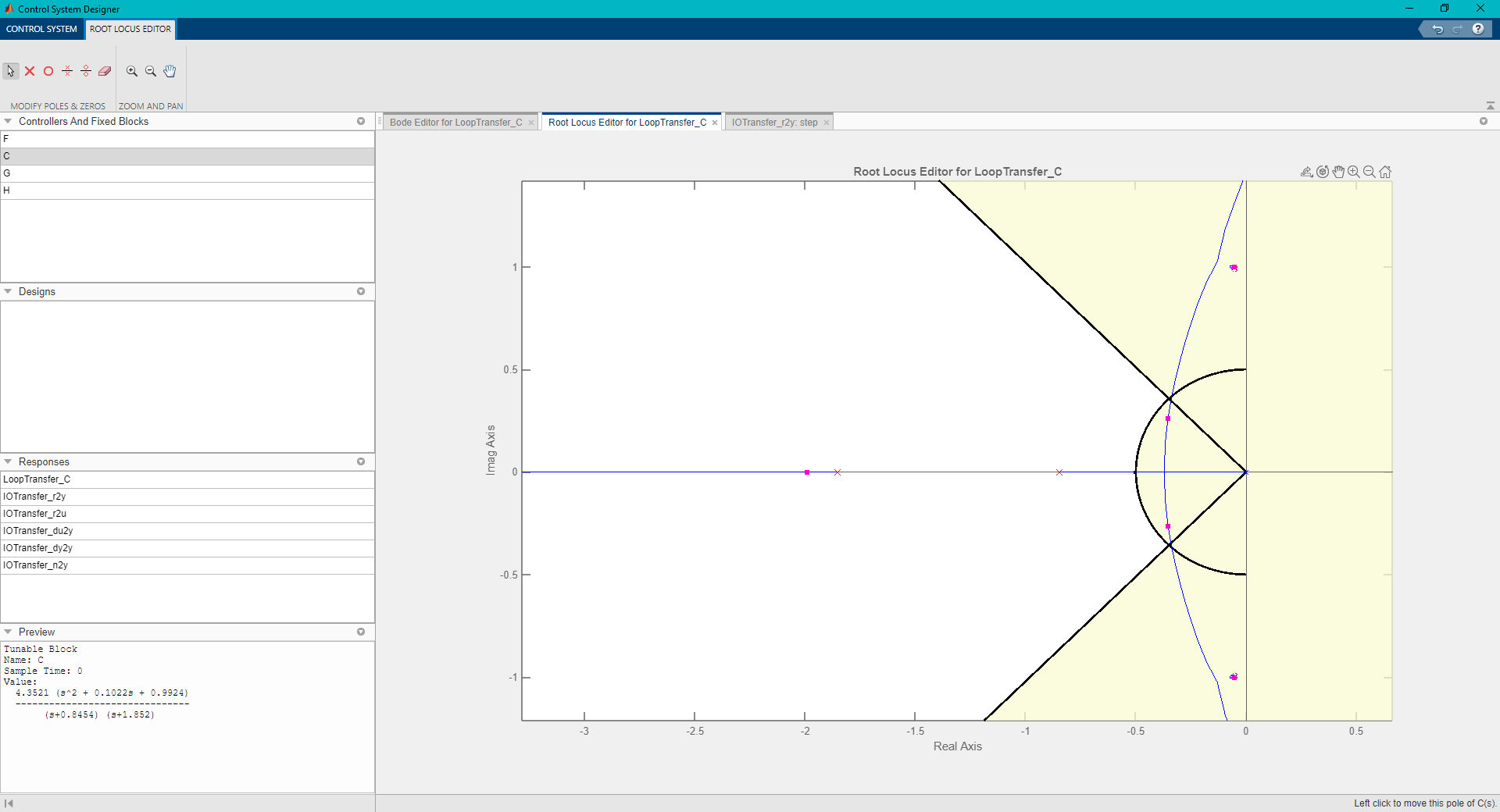
### DESIGN REQUIREMENTS

A screenshot of a computer

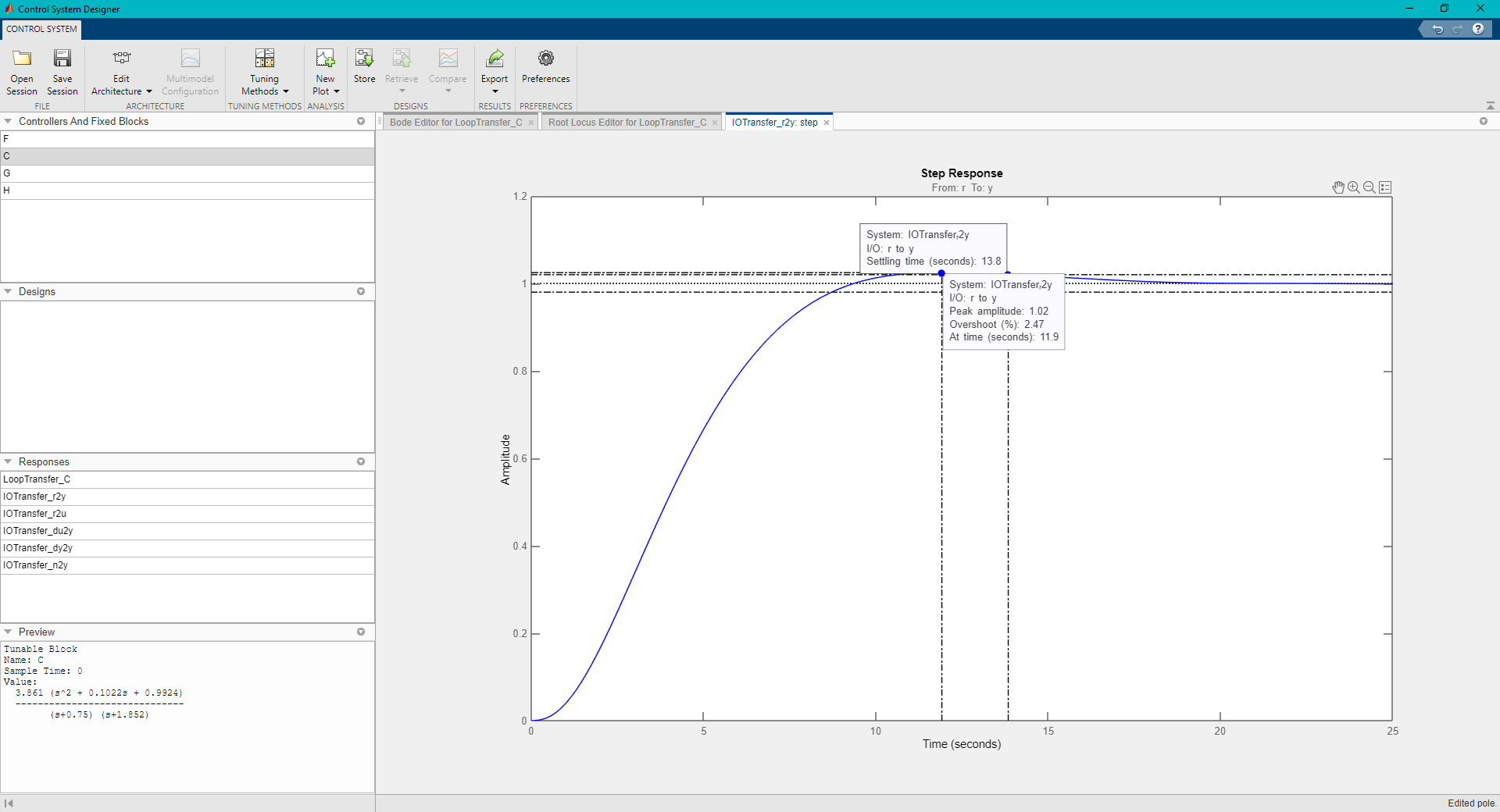
Description automatically generated with medium confidenceA screenshot of a computer screen

Description automatically generated with low confidence

### DESIGN REGION



### STEP RESPONSE



## 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:

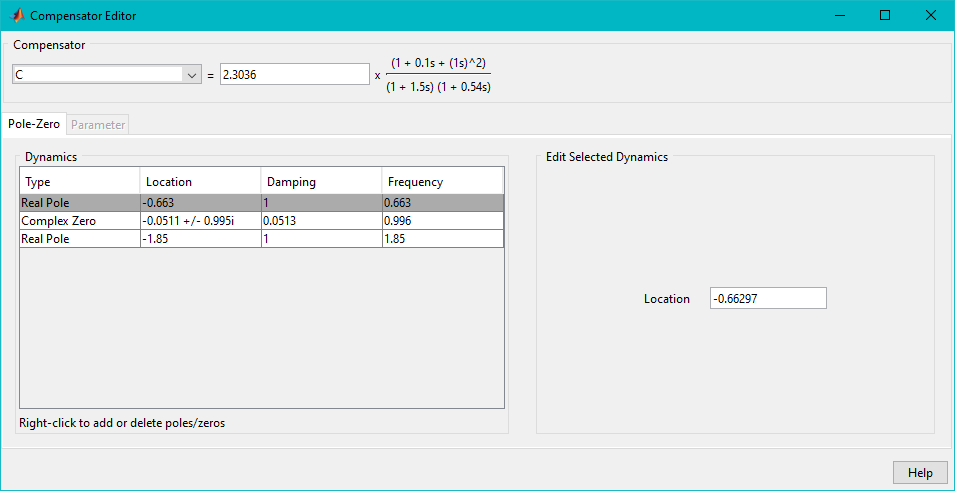
Zero at s = -0.051121+ j0.99487 (complex conjugate pole)

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)



### BODE PLOT

A screenshot of a computer

Description automatically generated

### STEP RESPONSE

A screenshot of a computer

Description automatically generated with medium confidence

## 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

A computer screen shot of a diagram

Description automatically generated with low confidence

### DISTURBANCE PLOT

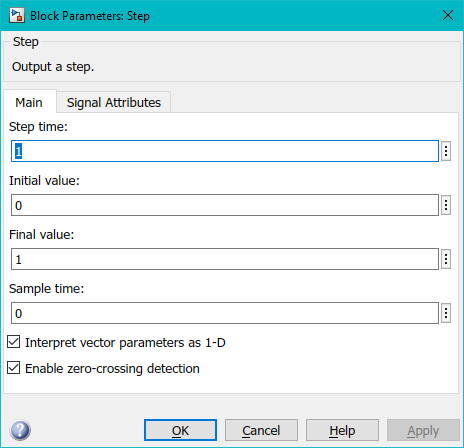
We create a pulse signal using two pulse generators in Simulink, where the signal is active for a time range from t=20 to t=22 and the two pulse generators are activated at different times. The output of the difference block will be a pulse signal that starts at time = 20 sec and ends at time = 23 sec, 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.

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence

### STEP INPUT



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

A screen shot of a computer

Description automatically generated with medium confidence

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

A screen shot of a graph

Description automatically generated with medium confidence

# 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:
3. 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:

A picture containing text, screenshot, font, number

Description automatically generated

A picture containing text, screenshot, font, number

Description automatically generated

1. Simulate the system in discrete time using the root locus method and find the step response.
2. Get the desired operating region and design a lead compensator C(Z) = to satisfy the operating conditions.
3. 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.
4. 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.

## 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) =

### 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)

A screen shot of a graph

Description automatically generated with medium confidence

and when Ts = 0.01 sec (i.e. short sampling interval), G(z) =

### 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)

A screen shot of a graph

Description automatically generated with medium confidence

## 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)

A screenshot of a computer

Description automatically generated

### DESIGN REQUIREMENTS

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer screen

Description automatically generated with low confidence

### DESIGN REGION

A screenshot of a computer

Description automatically generated with medium confidence

### STEP RESPONSE

A screenshot of a computer

Description automatically generated with medium confidence

### ROOT-LOCUS AFTER COMPENSATION

A screenshot of a computer

Description automatically generated with medium confidence

## 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 This is the formula for backward transformation.

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)

A screenshot of a computer

Description automatically generated

### DESIGN REGION

A screenshot of a computer

Description automatically generated with medium confidence

### STEP RESPONSE

A screenshot of a computer

Description automatically generated with medium confidence

### ROOT-LOCUS AFTER COMPENSATION

A screenshot of a computer

Description automatically generated with medium confidence

## 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

A screenshot of a computer

Description automatically generated with medium confidence

### RESULTS FROM EMULATION

A screenshot of a computer

Description automatically generated with medium confidence

### 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

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

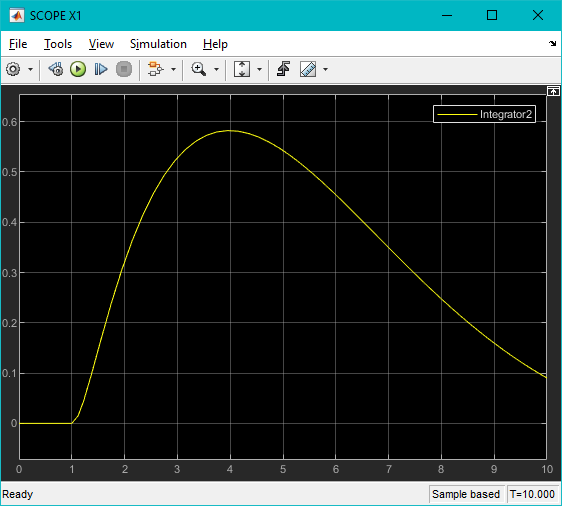
### System state space representation (**repA**) A picture containing text, screenshot, font, number Description automatically generated

### Simulink model

### A screenshot of a computer Description automatically generated

### SYSTEM STATES

### X1



### X2

A screen shot of a graph

Description automatically generated with medium confidence

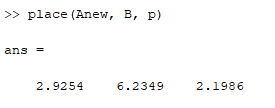
### Y=X3 A screen shot of a graph Description automatically generated with medium confidence

## 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  
A picture containing text, screenshot, font, line

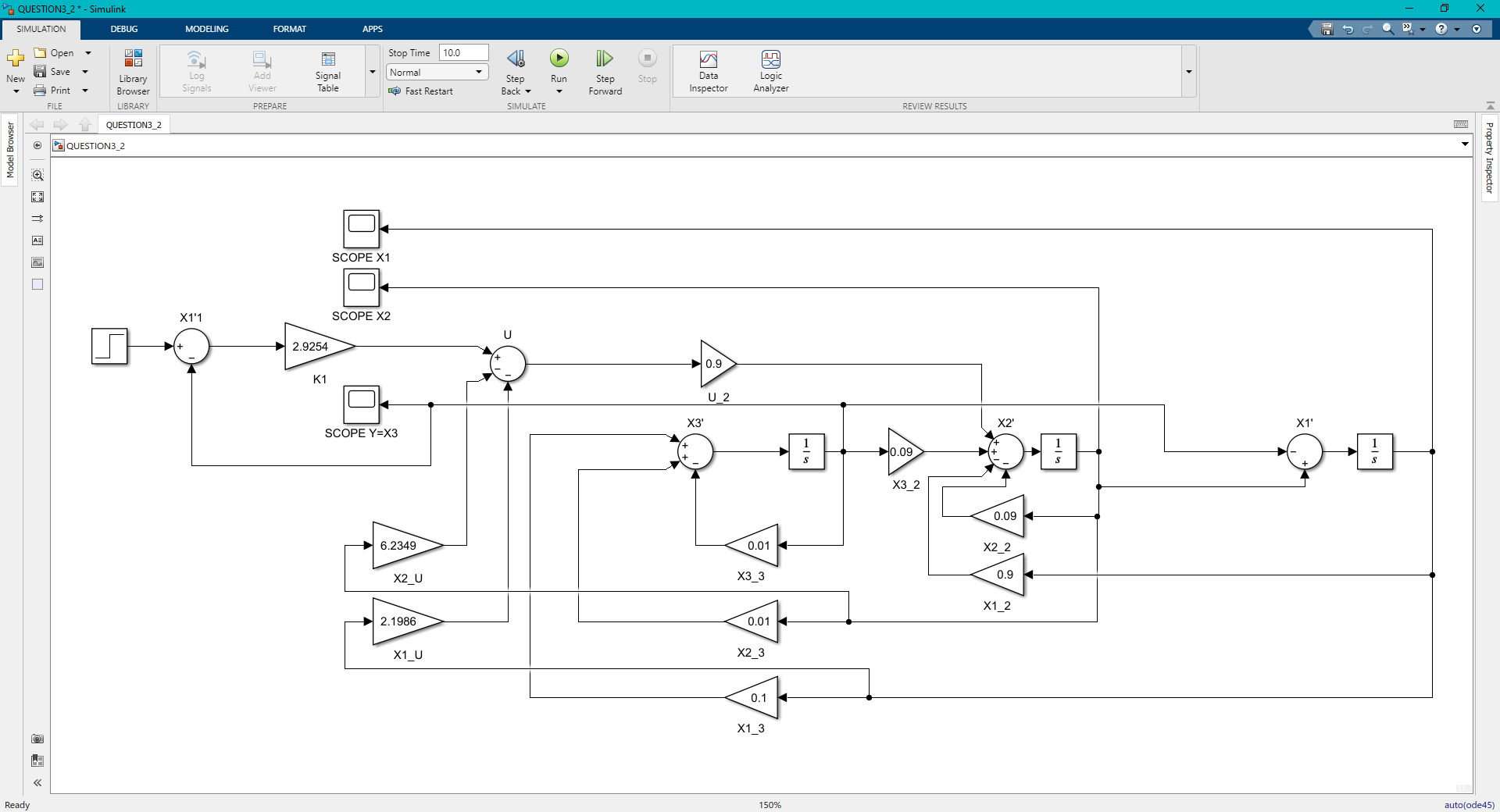
Description automatically generated

### CODE



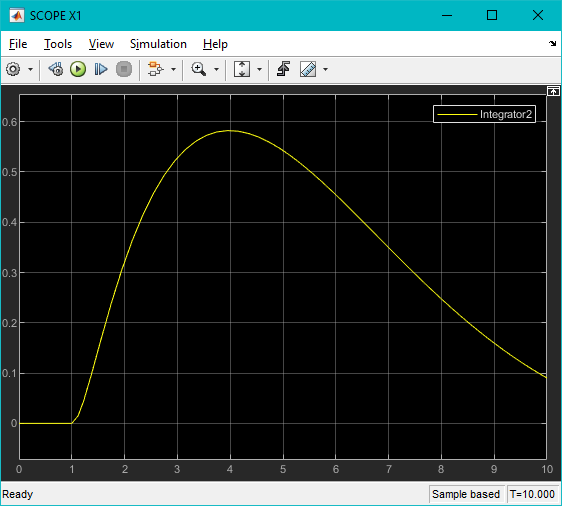
## Implement the feedback signals using the Simulink. Draw the states and output versus time. Comment on the results.

### SIMLINK MODEL WITH STATE FEEDBACK



### SYSTEM STATES

### X1



X2

A screen shot of a graph

Description automatically generated with medium confidence

Y=X3  
A screen shot of a graph

Description automatically generated with medium confidence

### 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

## 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

A picture containing text, screenshot, font, number

Description automatically generated

### SYSTEM STATES

### X1

A screen shot of a graph

Description automatically generated with medium confidence

### X2

A screen shot of a graph

Description automatically generated with medium confidence

### X3=Y

A screen shot of a graph

Description automatically generated with medium confidence

### CONTROLLABLE CANONICAL FORM

A picture containing text, screenshot, font, number

Description automatically generated

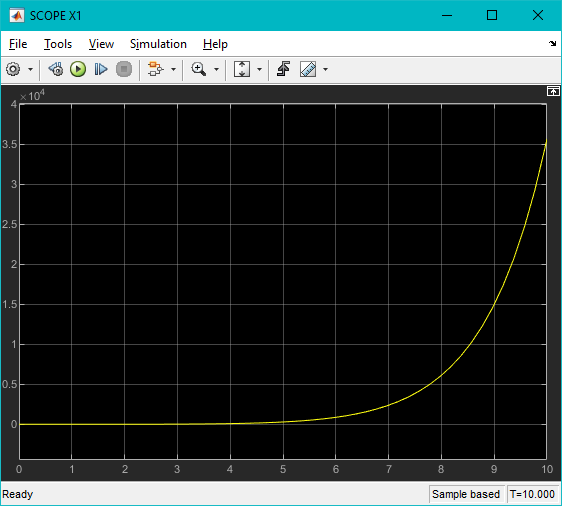
### SIMULINK MODEL

A computer screen shot of a diagram

Description automatically generated with low confidence

### SYSTEM STATES

### X1



### X2

A screen shot of a graph

Description automatically generated with medium confidence

### X3

A screen shot of a graph

Description automatically generated with medium confidence

### Y

A screen shot of a graph

Description automatically generated with medium confidence

### COMMENT

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

## 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

A picture containing text, screenshot, font, line

Description automatically generated

### RESPONSE

### X1

A screen shot of a graph

Description automatically generated with medium confidence

### X2

A screen shot of a graph

Description automatically generated

### X3=Y

A screen shot of a graph

Description automatically generated with medium confidence

### repA

### MODEL

A screenshot of a computer

Description automatically generated

### RESPONSE

X1

A screen shot of a graph

Description automatically generated with medium confidence

X2

A screen shot of a graph

Description automatically generated with medium confidence

Y=X3

A screen shot of a graph

Description automatically generated with medium confidence

### 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.

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

### repC

### MODEL

A screenshot of a computer

Description automatically generated

### POLE PLACEMENT

### RESPONSE

### X1

A screen shot of a graph

Description automatically generated with medium confidence

### X2

A screen shot of a graph

Description automatically generated with medium confidence

### X3

A screen shot of a graph

Description automatically generated with medium confidence

### Y

A screen shot of a graph

Description automatically generated with medium confidence

### COMMENT

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

## Using rep A and the controller designed in 2, assuming that the only measurement available is the output 𝜔2 , 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.

## 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.