EE352: Advanced Control Systems

Term Project Report

SET-27

Shivam Vishwakarma 210102080

Neelima Pulipati 210108032
Zainab Ali 210108062
Drishti Agarwal 210108066
Katla Rohith Kumar Reddy 210108067

Department of Electronics and Electrical Engineering

26th April 2024

Contents

Question 1

Given the present pandemic, the faculties have to record their lecture via camera. The idea being oated around is that there is need for a recording system with a camera that follows the faculty when he or she is writing on the blackboard. Hence, there is need of a controller to keep the camera erect when it moves, i.e., the camera needs to be balanced on a stick. The configuration is shown in Figure (1).

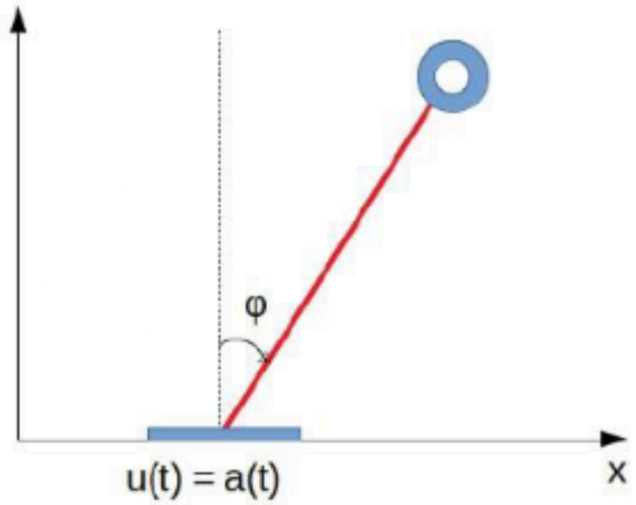


Figure (1): A camera system

The bottom end of the pointer is moving along the x-axis with your input u(t) being the acceleration of this point: u(t) = a(t). The position of the camera is given by x(t). The length of the stick is L; assume that its mass is concentrated at the top end. Assuming ϕ is small, the differential equations governing the system are found to be:

$$mg\phi(t) = mx^{\cdot\cdot}(t) \text{ and } x(t) = a(t) + L\phi(t) \qquad (1)$$

- Can the system be stabilized by proportional feedback:

  u(t) = - ky(t) ?

- Determine a state-feedback law so that the closed-loop system has both the poles at-2.

Instructions:- This problem will be solved using MATLAB, and needs to be explained very well

Solution 1

To solve this problem of stabilizing a camera on a stick, where the stick's base is controlled by acceleration, we followed the following steps :

1. Identifying the System Dynamics

Given:

- **x(t)** is the position of the base along the x-axis.
- **ϕ(t)** is the angle of the stick with the vertical.
- **u(t) = a(t)** is the acceleration applied at the base.
- **L** is the length of the stick.
- **m** is the mass of the stick's concentrated mass at the top.
- **g** is the gravitational constant.

The key dynamic equations

1. **Angle Equation:** The gravitational force creates a torque that is balanced by the movement of the camera along the x-axis. This can be expressed as:

$$mg \cdot \phi(t) = m \cdot x''(t)$$

2. **Position Equation:** The acceleration of the camera, **x''(t)** or **a(t)**, affects its position. The relationship between acceleration and the angle of the stick is:

$$x(t) = a(t) + L \cdot \phi(t)$$

To control the system, we need a controller that can stabilize the stick while allowing movement along the x-axis.

2. State-Space Representation

The state-space representation is a mathematical model of the system, where we describe the system's state variables, inputs, outputs, and how they evolve over time. Let's identify the state variables and derive the system's state-space form.

- **State Variables:** To represent the system in state-space form, we need state variables. The system has position x(t) and velocity x'(t). Define these as:

  - $x_1 = x(t)$
  - $x_2 = x'(t)$

- **State-space equations:** Based on the differential equations, the state-space equations are:

  - $x_1' = x_2,$
  - $x_2' = \frac{g}{L} \cdot x_1 + u(t),$

- **State-space Matrices:** This leads to a state-space representation with matrices:

- $A = \begin{bmatrix} 0 & 1 \\ \frac{g}{L} & 0 \end{bmatrix}$
- $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

- $C = \begin{bmatrix} -g/L & 0 \end{bmatrix}$
- $D = \begin{bmatrix} 0 \end{bmatrix}.$

Open Loop Transfer Function

The open-loop transfer function can be derived using the dynamic equations or the following formula:

$$G(s) = C \cdot (sI - A)^{-1} \cdot B + D$$

$$G(s) = C \cdot (sI - A)^{-1} \cdot B = \frac{-g/L}{s^2 - g/L}.$$

4. Q1. (i) Stability Analysis:

To determine stability, we examine the system's response to proportional feedback. The key is to find whether the feedback can keep the system's poles (roots of the characteristic polynomial) in the left half-plane, ensuring stability.

Open Loop Transfer Function

The open-loop transfer function, derived from the differential equation, relates the system output y(t) to the control input u(t)

$$G(s) = \frac{-g/L}{s^2 - g/L}.$$

The Characteristic Polynomial and Poles

The characteristic polynomial is:

$$s^2 - g/L$$

With poles at

$$\pm\sqrt{g/L}.$$

Closed-Loop Transfer Function with Proportional Feedback

With proportional feedback u(t)=−k⋅y(t), the control input u(t) is directly related to the output y(t). The closed-loop transfer function for a feedback system can be derived from the open-loop transfer function using this feedback:

$$G_{\text{cl}}(s) = \frac{G(s)}{1 + k \cdot G(s)}.$$

Substituting G(s), we get

$$G_{\text{cl}}(s) = \frac{-g/L}{s^2 - g/L} \cdot \frac{1}{1 - k \cdot \frac{-g/L}{s^2 - g/L}}.$$

Simplifying we get

$$G_{\text{cl}}(s) = \frac{-g/L}{s^2 - g/L - k \cdot g/L} = \frac{-g/L}{s^2 - (g/L + k \cdot g/L)}.$$

Characteristic Polynomial and Stability

The denominator of the transfer function gives the characteristic polynomial:

$$s^2 - (g/L + k \cdot g/L).$$

For stability, the characteristic polynomial should have roots with negative real parts. To find the roots of this polynomial, we need to examine the values of s:

$$s = \pm\sqrt{g/L + k \cdot g/L} = \pm\sqrt{(1 + k) \cdot g/L}.$$

With proportional feedback, the roots of the characteristic polynomial depend on k. If k>0, one root is positive, indicating instability. This result implies that proportional feedback leads to one pole in the right half-plane, indicating that the system is unstable.

Conclusion

Based on this analysis, the system cannot be stabilized with proportional feedback u(t)=−k⋅y(t). The characteristic polynomial derived from the closed-loop transfer function shows that a proportional gain results in at least one positive root, indicating an unstable system. More complex feedback control strategies are needed to stabilize this system.

4. Q1. (ii) State Feedback Law for Pole placement at -2(both):

Designing a State-Feedback Controller

State-feedback control allows us to manipulate the system's behaviour by altering the control input based on the current state. The goal is to find a feedback gain

K=[k 1 ,k 2]

such that the closed-loop system has the desired stability and pole placement.

With state-feedback, the control input is:

$$u(t) = -K \cdot x(t) = -k_1 \cdot x_1 - k_2 \cdot x_2.$$

This control law creates a closed-loop system, where the closed-loop state-space matrix A_{cl} becomes:

$$A_{\text{cl}} = A - B \cdot K = \begin{bmatrix} 0 & 1 \\ g/L - k_1 & -k_2 \end{bmatrix}$$

Characteristic Equation and Pole Placement

The characteristic equation of a system determines its stability. For a stable system, all poles (roots of the characteristic equation) must be in the left half-plane, indicating that the system's behaviour decays over time.

To find the characteristic equation, we determine

$$sI - A_{\text{cl}} = \begin{bmatrix} s & -1 \\ -(g/L - k_1) & s + k_2 \end{bmatrix}$$

The determinant of this matrix gives the characteristic equation for the closed-loop system:

$$s^2 + k_2 \cdot s + k_1 - g/L$$

Desired Characteristic Polynomial

To place both poles at -2, the desired characteristic polynomial is:

$$s^2 + 4 \cdot s + 4$$

This polynomial has both poles at -2, which indicates a stable system with a specific damping behavior.

Finding State-Feedback Gain K

To achieve the desired characteristic polynomial, we can compare coefficients between the derived characteristic polynomial and the desired one:

We get

k1=g/L +4

And

K2 = 4

Given g=9.81 and L=1

- $k_1 = 5.81,$
- $k_2 = 4.$

Conclusion

$$K = [5.81, 4]$$

The closed-loop characteristic polynomial becomes:

$$s^2 + 4 \cdot s + 5.81 - g/L,$$

ensuring both poles are at -2, indicating a stable system. This state-feedback law can be used to control the system and achieve the desired stability and dynamic response.

5. MATLAB Implementation

Code

```
clc;
clear;

% System parameters
g = 9.81; % Gravitational acceleration (m/s^2)
L = sym('L', 'real'); % Length of the pendulum (m) as a symbolic variable

% State-space representation
A = [0 1; g/L 0];
B = [0; 1];
C = [-g/L 0]; % Assuming full-state feedback
D = 0;

% Check controllability
controllability = rank([B A*B]);
disp('The controllability matrix is');
[B A*B]
if controllability == length(A)
    disp('The system is fully controllable for all L.');
else
    disp('The system is not fully controllable.');
end
```

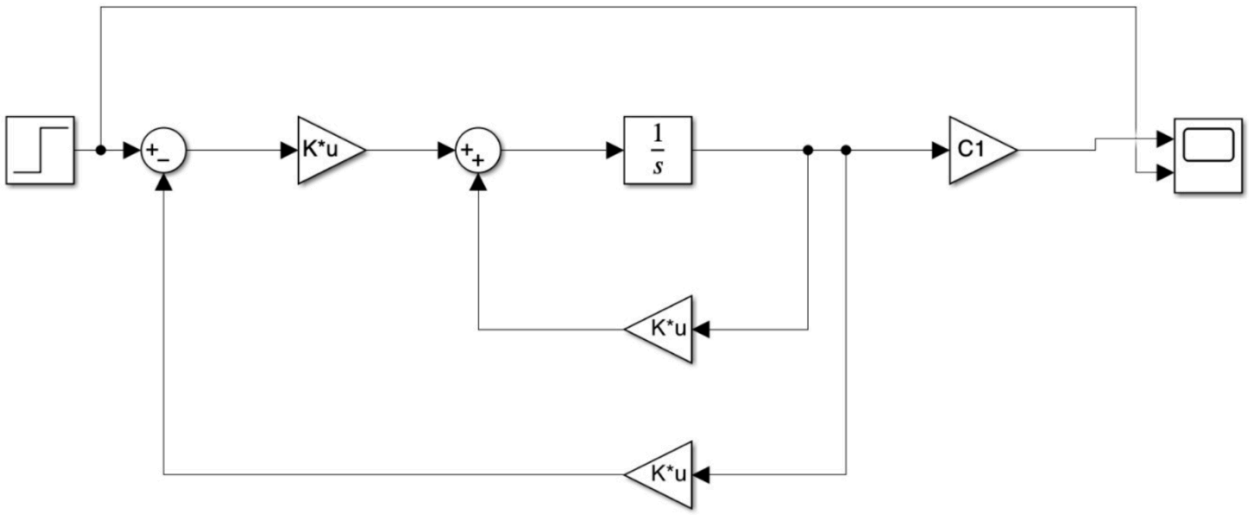
```

% Check observability
observability = rank([C; C*A]);
disp('The observability matrix is');
[C; C*A]
if observability == length(A)
    disp('The system is fully observable for all L.');
else
    disp('The system is not fully observable.');
end

disp('For K=[k1 k2]');
disp('A-BK=Acl');
k1 = sym('k1', 'real');
k2 = sym('k2', 'real');
Acl=A-B*[k1 k2]
s = sym('s', 'real');
det([s 0;0 s]-Acl)

```

```
42
43  % Desired pole locations
44  p1 = -2;
45  p2 = -2;
46
47  % Desired characteristic equation
48  des_poles_eq = s^2 + 4*s+4
49  % Calculation of the feedback gain matrix
50
51
52  disp('The system from here assumes L=1.');
53  %For L=1 metres
54  A1 = [0 1; g/1 0];
55  B1 = [0; 1];
56  C1 = [-g/1 0]; % Assuming full-state feedback
57  D1 = 0;
58  des_poles=[-2 -2];
59  K1 = acker(A1, B1, des_poles);
60  % Closed-loop system matrix
61  A1_cl = A1 - B1*K1;
```

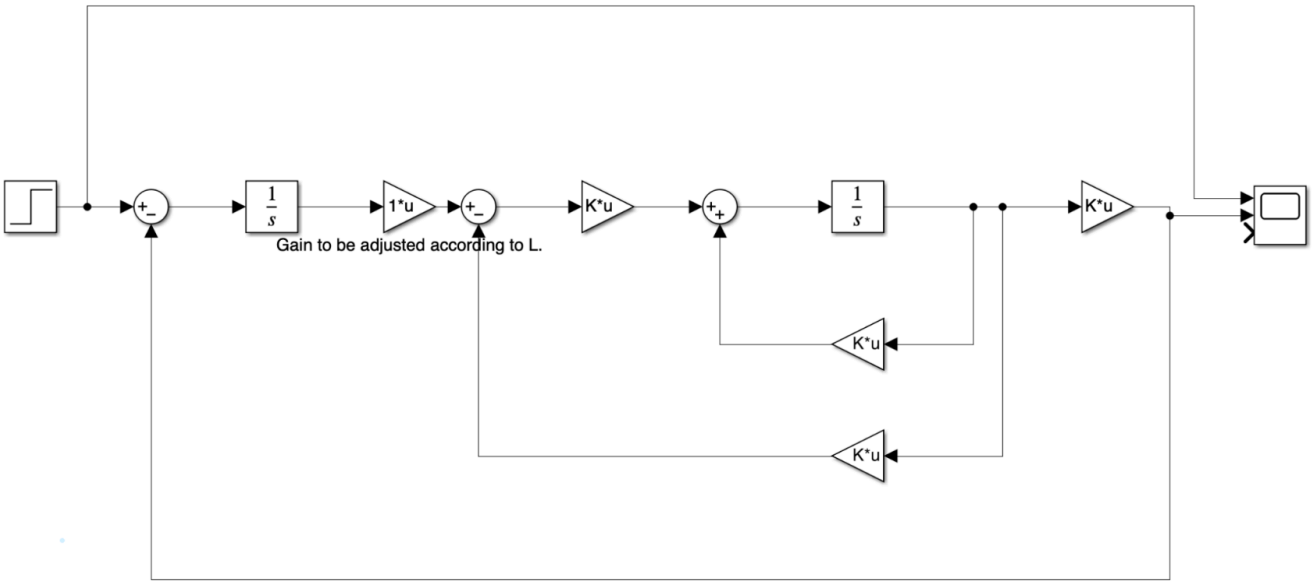
```
62
63  % Verifying the closed-loop pole locations
64  eig_cl = eig(A1_cl);
65
66  % Displaying the results
67  disp('System parameters:');
68  disp(['g = ' num2str(g) ' m/s^2, L = 1 m']);
69  disp('State-space representation:');
70  disp('A = ');
71  disp(A1);
72  disp('B = ');
73  disp(B1);
74  disp('Desired pole locations:');
75  disp(['p1 = ' num2str(p1) ', p2 = ' num2str(p2)]);
76  disp('Feedback gain matrix:');
77  disp('K = ');
78  disp(K1);
79  disp('Closed-loop system poles:');
80  disp(eig_cl);
```

Block Diagram

Open Loop


K*u
1
s
C1
K*u
K*u

Closed Loop


1
s
1*u
Gain to be adjusted according to L.
K*u
1
s
K*u
K*u
K*u

Simulation Results

```
The system is fully observable for all L.
For K=[k1 k2]
A-BK=Acl

Acl =

[                0,   1]
[981/(100*L) - k1, -k2]


ans =

(100*L*s^2 + 100*L*k2*s + 100*L*k1 - 981)/(100*L)


des_poles_eq =

s^2 + 4*s + 4

The system from here assumes L=1.
System parameters:
g = 9.81 m/s^2, L = 1 m
State-space representation:
A =
         0    1.0000
    9.8100         0

B =
     0
     1

Desired pole locations:
p1 = -2, p2 = -2
Feedback gain matrix:
K =
   13.8100    4.0000

Closed-loop system poles:
    -2
    -2
```

```
The controllability matrix is

ans =

[0, 1]
[1, 0]

The system is fully controllable for all L.
The observability matrix is

ans =

[-981/(100*L),            0]
[           0, -981/(100*L)]

The system is fully observable for all L.
For K=[k1 k2]
A-BK=Acl

Acl =

[               0,   1]
[981/(100*L) - k1, -k2]


ans =

(100*L*s^2 + 100*L*k2*s + 100*L*k1 - 981)/(100*L)


des_poles_eq =

s^2 + 4*s + 4

The system from here assumes L=1.
System parameters:
g = 9.81 m/s^2, L = 1 m
State-space representation:
A =
         0    1.0000
    9.8100         0
```

6. Analysis:

This code creates a state-space model for the given system, computes the state-feedback gain K̀ to place the poles at -2, and creates a closed-loop state-space system. The result can be used to analyze the closed-loop behavior, simulate the response, or further design additional controllers, observers, or state estimators.

Question 2

Design a PID controller using Ziegler-Nichols tuning rules For the OLTF

$$\frac{20}{s^3 + 15s^2 + 10s}$$

With the help of MATLAB(Simulink/code). Take input as a step, and show responses.

Solution 2

To solve this problem of stabilizing a camera on a stick, where the stick's base is controlled by acceleration, we followed the following steps :

1. Identifying the System Dynamics

The OLTF of the given system is:

$$\frac{20}{s^3 + 15s^2 + 10s}$$

2. Zeigler-Nichols Method Selection

For this problem, we are opting to utilize the First Method of Ziegler-Nichols tuning. The choice of the First Method stems from our strategy of adjusting the gain until steady oscillations are observed, followed by fine-tuning to reduce overshoot. This method provides a systematic approach to tune the PID controller parameters based on the system's response characteristics.

Theoretical Explanation:

The Ziegler-Nichols tuning method offers two main approaches: the First Method (Ultimate Gain) and the Second Method (Ultimate Period). The selection between these methods hinges on the system's behavior in response to a step input.

In the First Method, the proportional gain (Kp) is incrementally increased until the system exhibits steady oscillations with constant amplitude. This critical gain value (Kcu) and the corresponding oscillation period (Pcu) are then noted. By utilizing these parameters, we derive the appropriate proportional, integral, and derivative terms for the PID controller.

This method is preferred when the system response to a step input allows for clear observation of steady oscillations, enabling straightforward identification of the critical parameters. Additionally, the First Method facilitates a stepwise tuning process, making it suitable for systems where stability and overshoot are primary concerns.

By opting for the First Method in this context, we aim to leverage its structured approach to efficiently tune the PID controller and achieve desirable system performance, ensuring stability while minimizing overshoot and settling time.

3. First Method (Ultimate Gain)

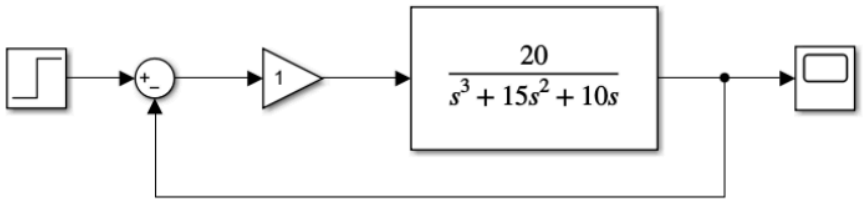
Procedure:

- Apply a step input to the system.
- Increase the proportional gain (Kp) until the system oscillates steadily with constant amplitude.
- Note the critical gain (Kcu) and the corresponding oscillation period (Pcu).

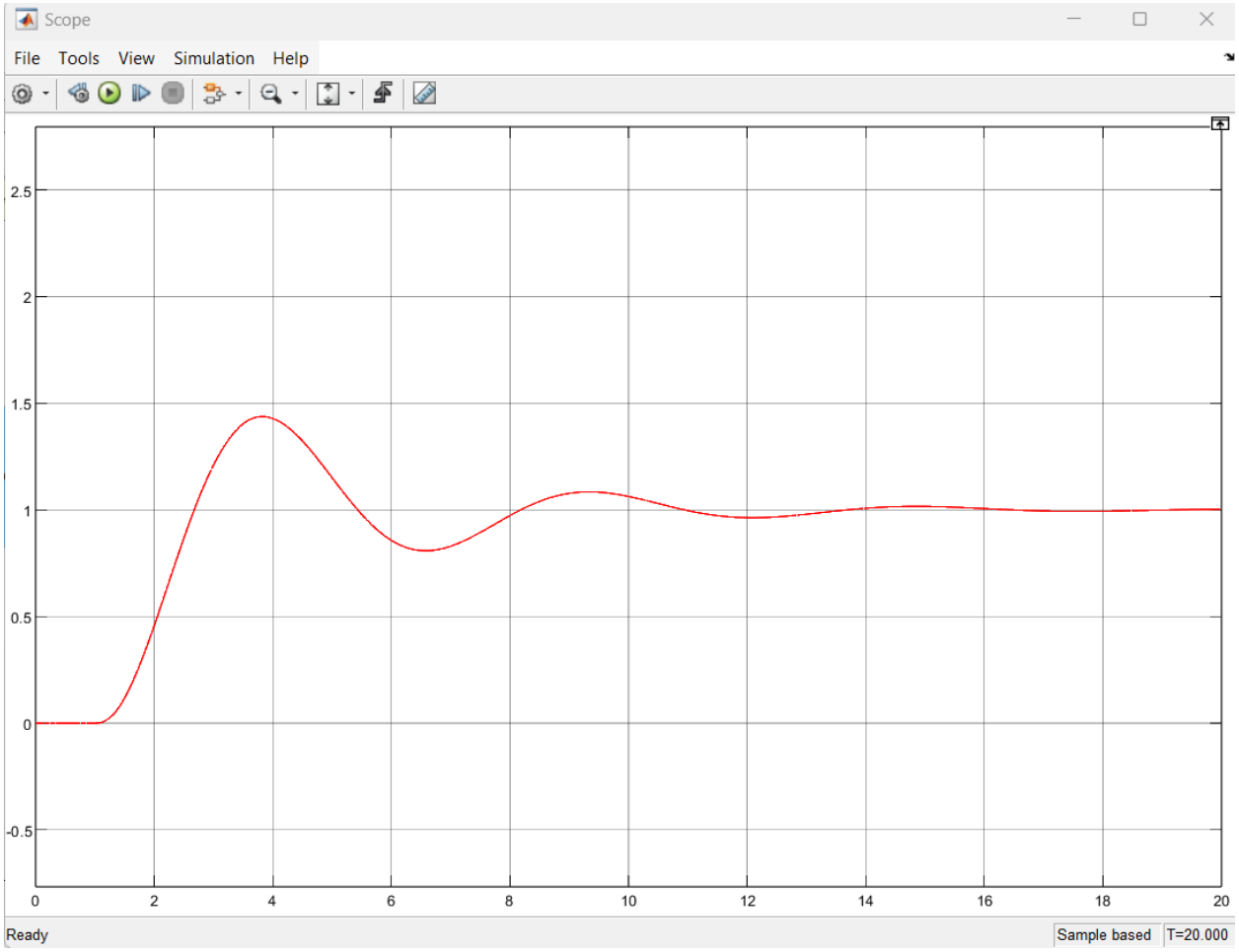
Controller Parameters:

- For P control: $Kp = 0.5 \times Kcu$
- For PI control: $Kp = 0.45 \times Kcu$ and $Ti = 0.85 \times Pcu$
- For PID control: $Kp=0.6\times Kcu$, $Ti=0.5\times Pcu$, and $Td=0.125\times Pcu$

4. MATLAB Implementation

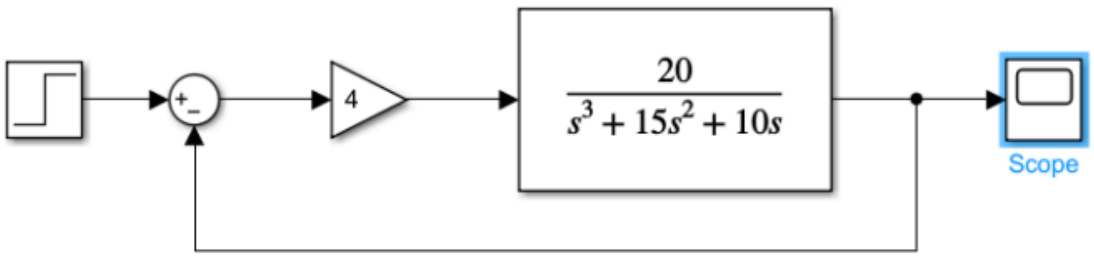


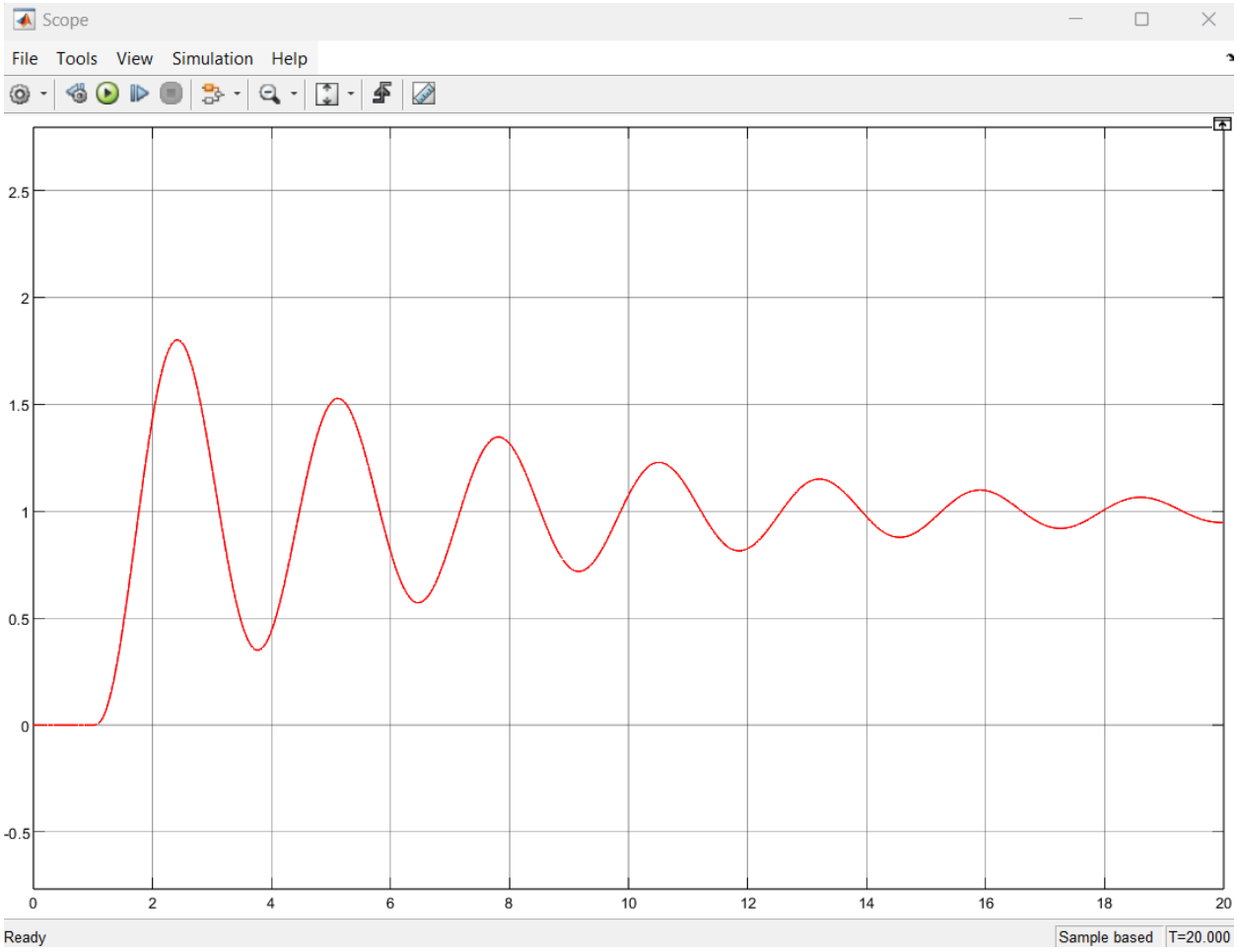
The response of closed loop system with the OLTF G(s) given is:



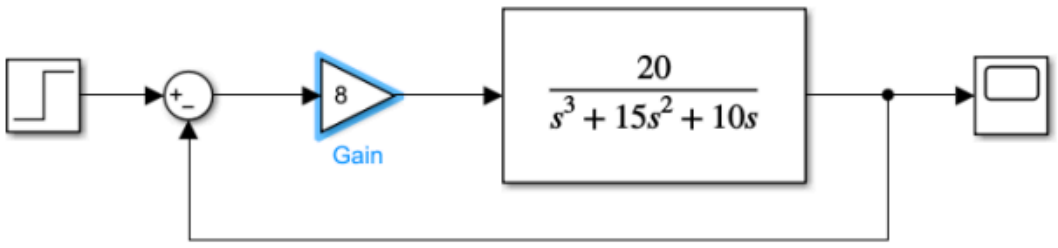
Designing of PID controller:

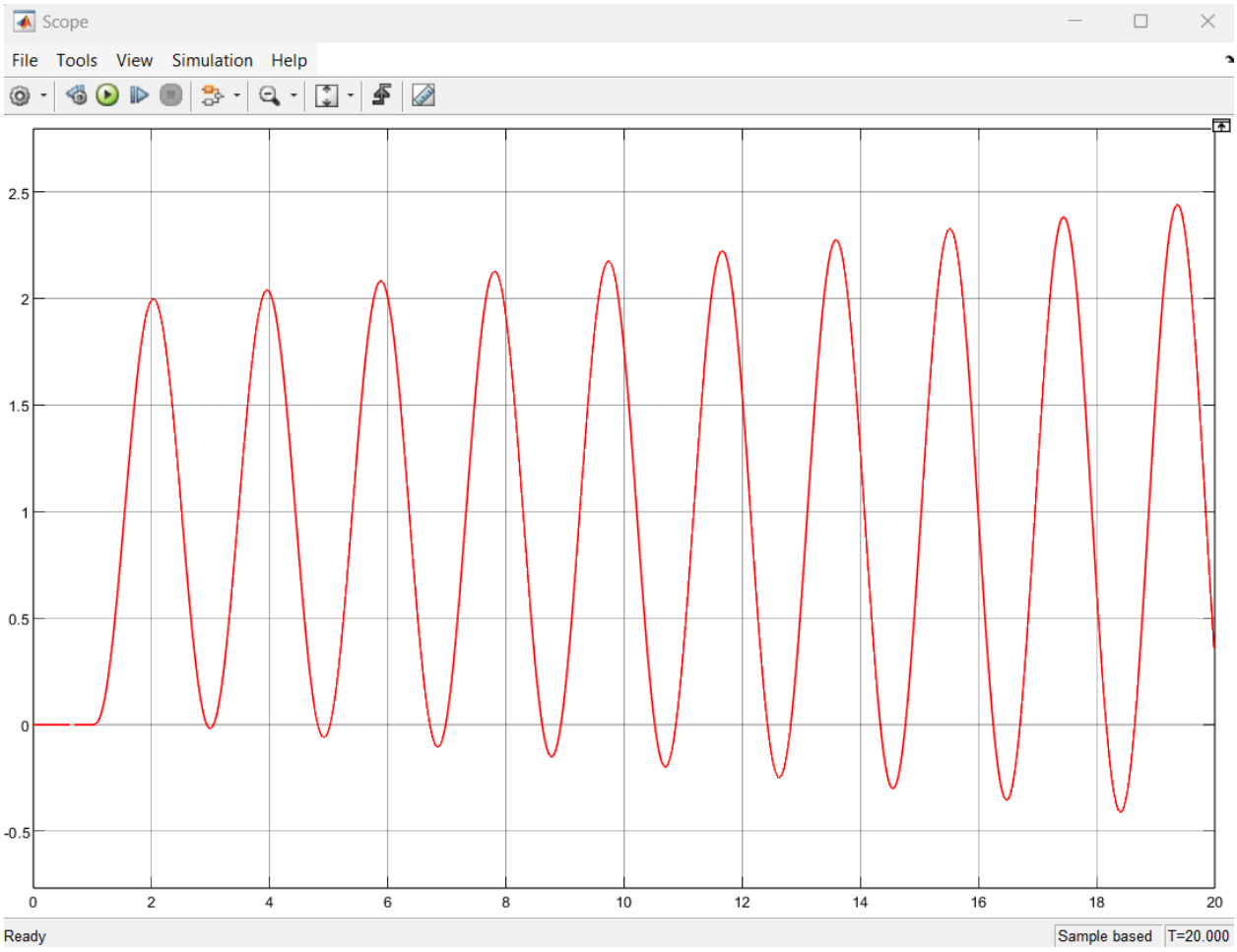
- Set the value of P to attain the sustained oscillations
    1. At k=4 :



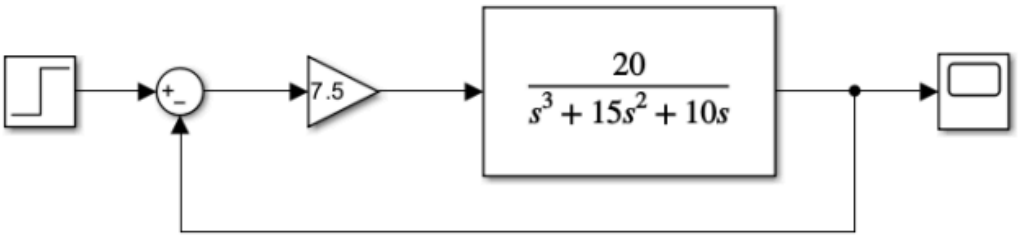

Scope
File Tools View Simulation Help
2.5
2
1.5
1
0.5
0
-0.5
0 2 4 6 8 10 12 14 16 18 20
Ready
Sample based
T=20.000

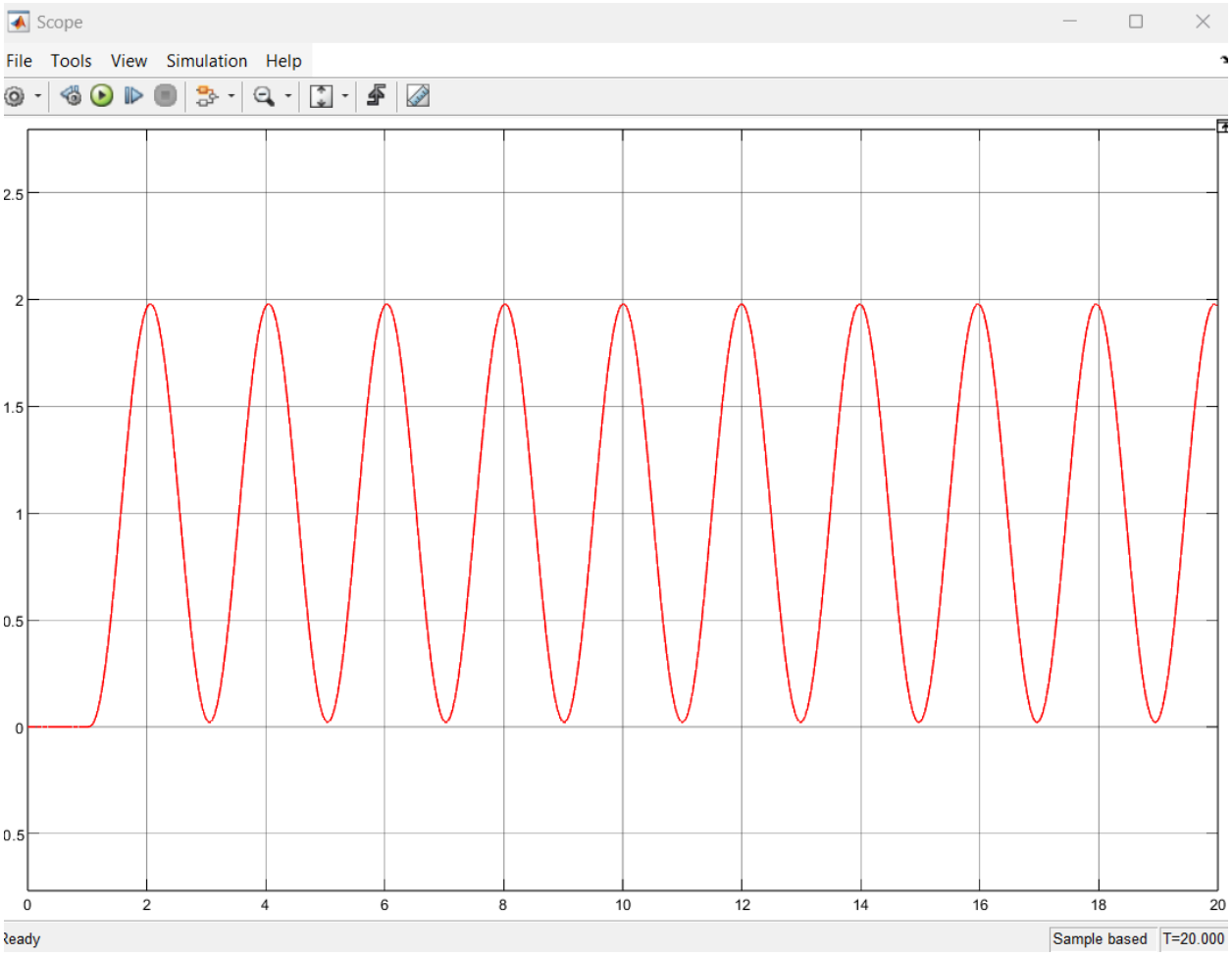
2. At k=8 :


8
Gain
20
$s^3 + 15s^2 + 10s$


Scope
File Tools View Simulation Help
2.5
2
1.5
1
0.5
0
-0.5
0 2 4 6 8 10 12 14 16 18 20
Ready
Sample based
T=20.000

3. At k=7.5:

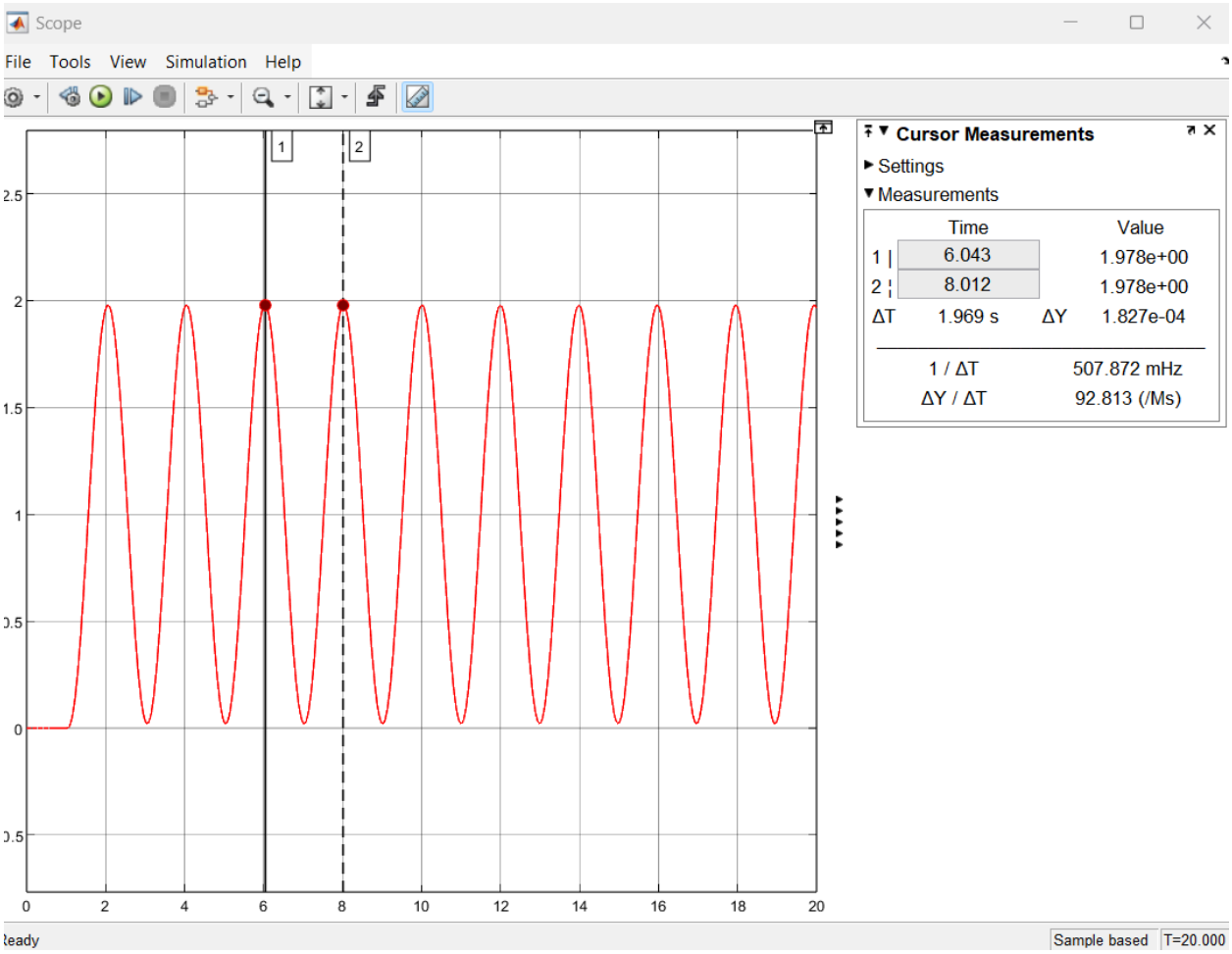

+
−
7.5
20
$s^3 + 15s^2 + 10s$



Hence the gain that give sustained oscillations is:

$$critical\ gain\ (Kcu)\ = 7.5$$

- Now calculate the Period of Oscillations:



$$Pcu = 1.969s$$

- **PID parameters:**

  Hence,

Controller	K_p	T_i	T_d
P	K_p=0.5×K_{cu}	-	-
PI	K_p=0.45×K_{cu}	K_p=0.85×K_{cu}	-
PID	K_p=0.6×K_{cu}	K_p=0.5×K_{cu}	K_p=0.125×K_{cu}

$$Kp = 7.5/1.7 = 4.412$$

$Ti = 1.969/2 = 0.984$

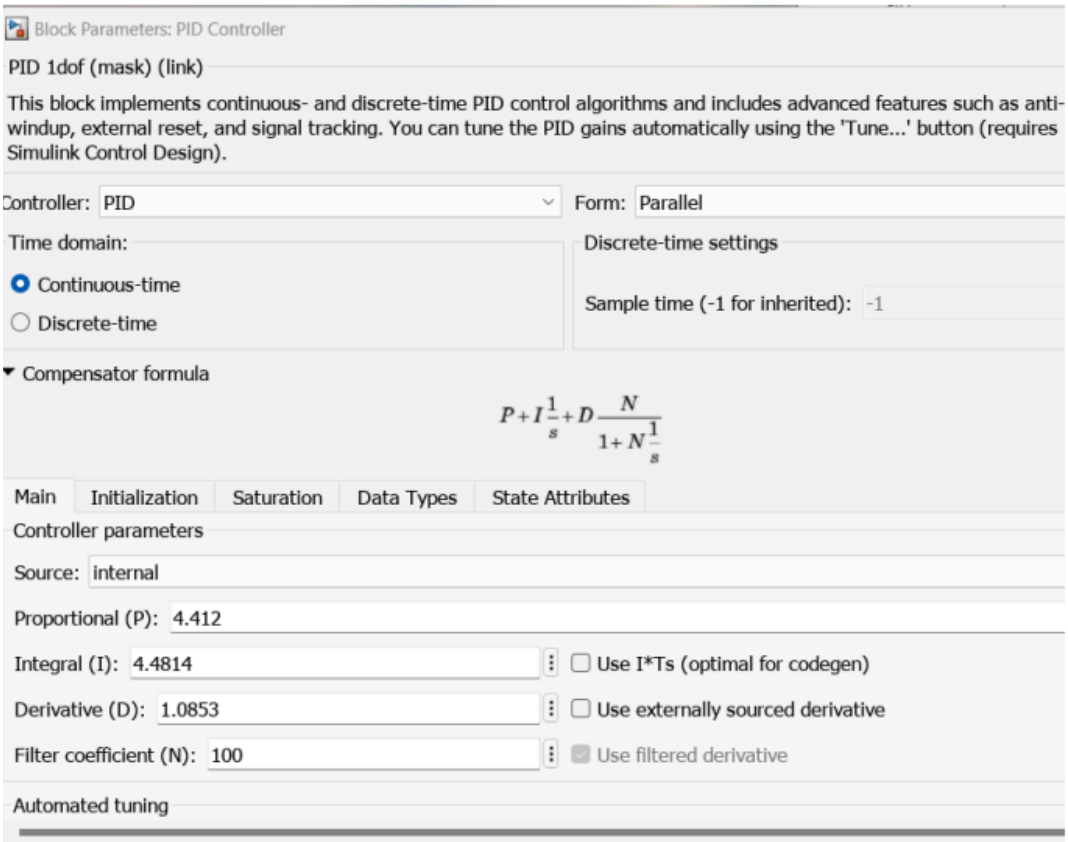
$Td = 1.969/8 = 0.246$

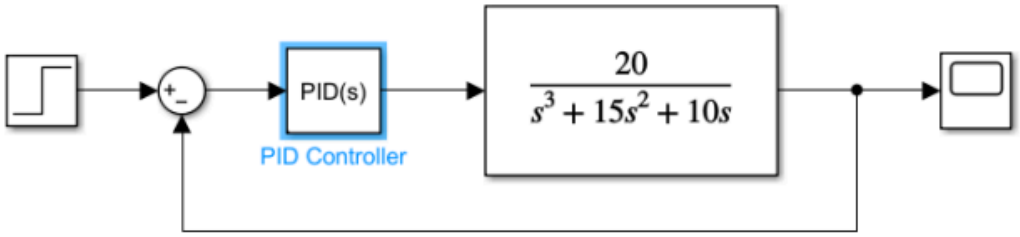
Now,

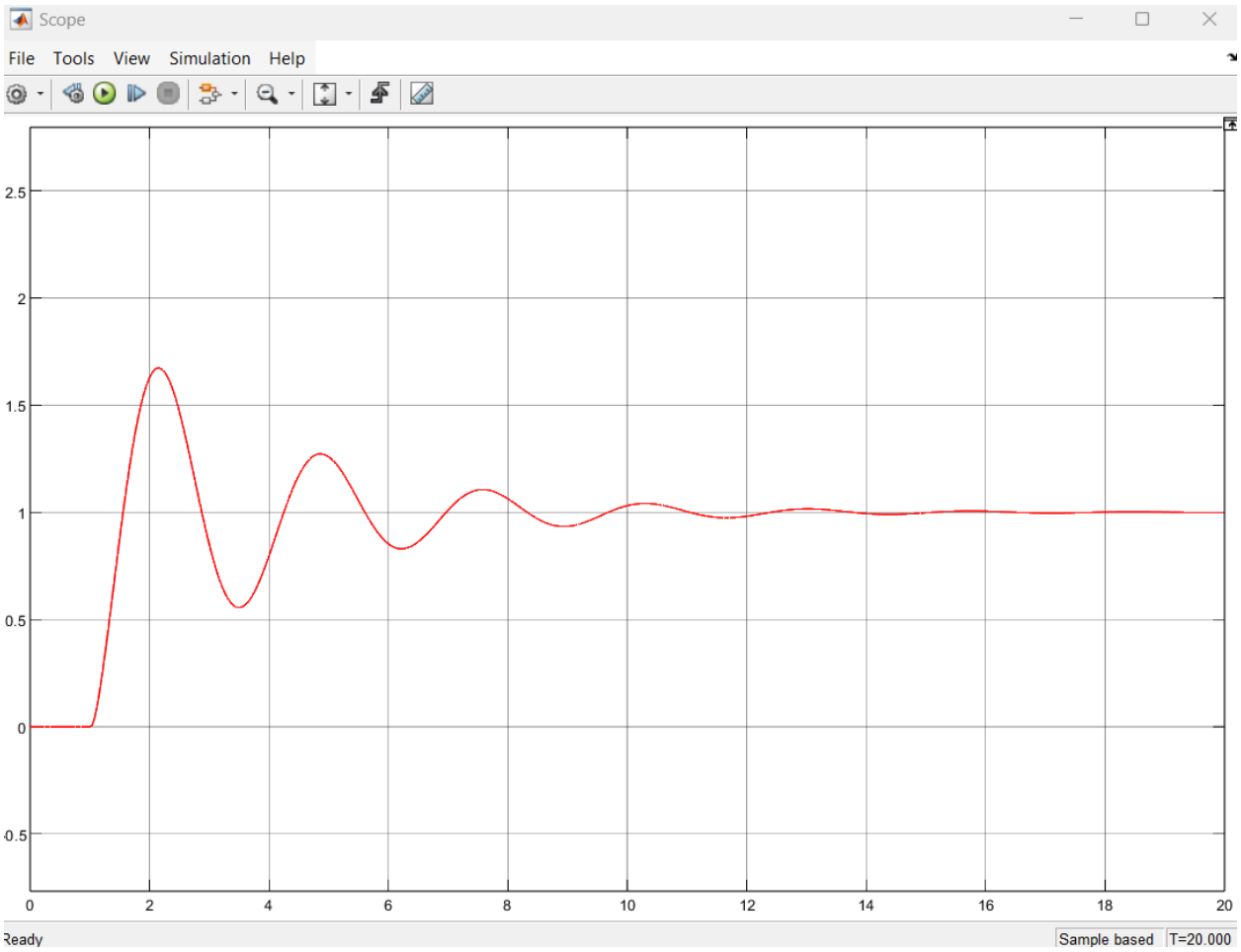
$Ki = Kp/Ti = 4.412/0.9845 = 4.4814$

$Kd = Kp * Td = 4.412 * 0.246 = 1.0853$

- **Now set the parameters by including a PID block**

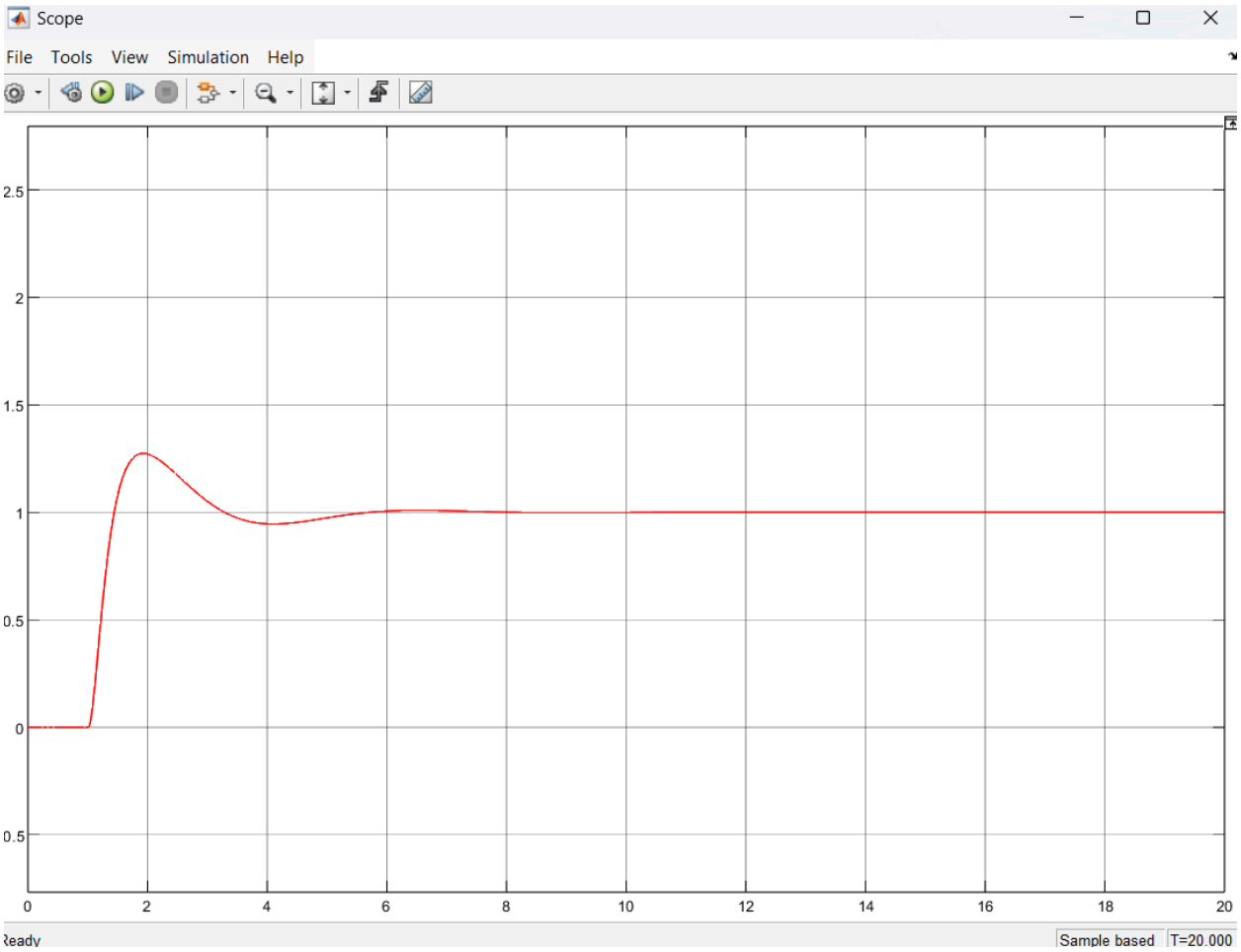






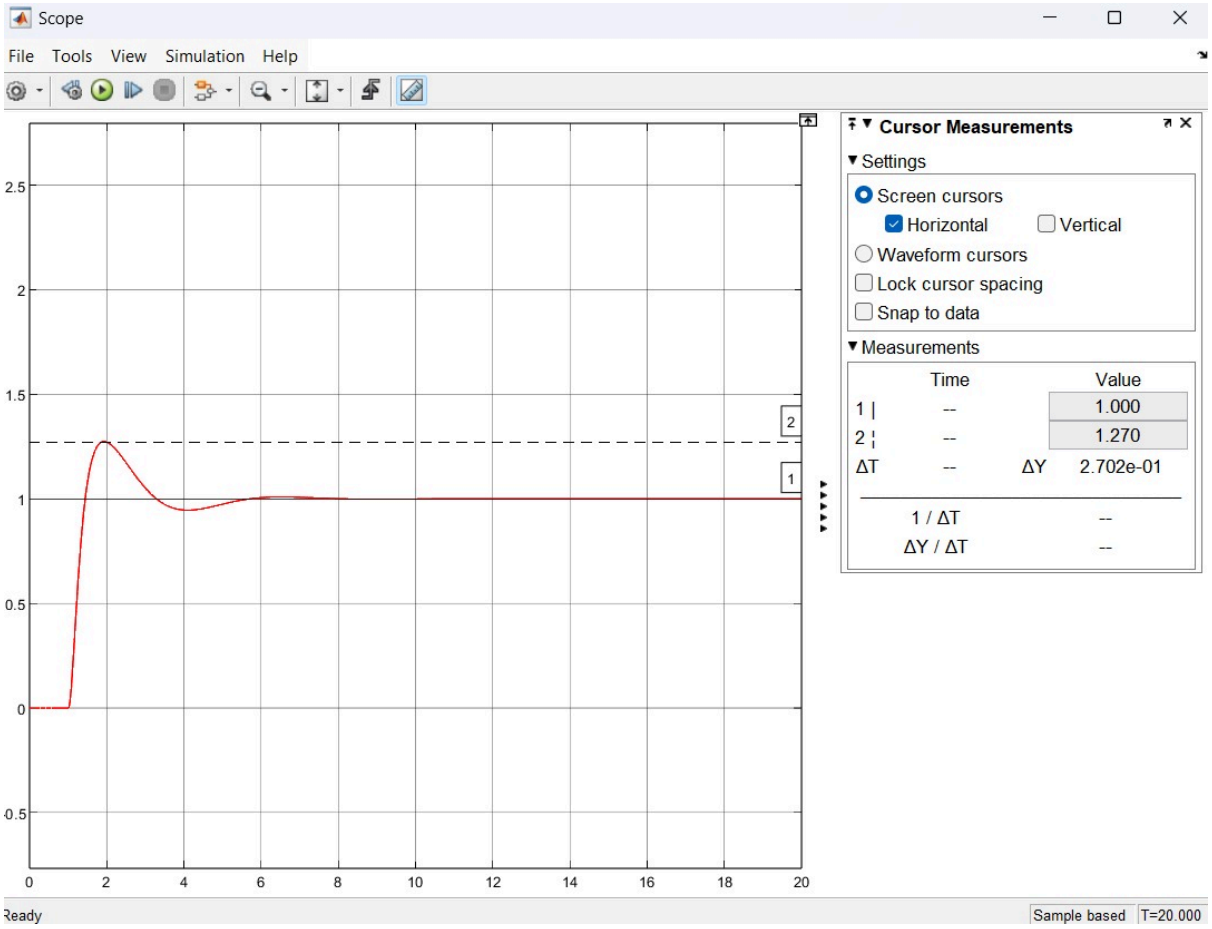
- Now other adjustments like reducing the overshoot can be done,

Therefore, the final output after adding a PID controller is:



5. Analysis

- **Steady-State Error:** The analysis revealed that all controller configurations achieved a steady-state error of zero. This indicates that the system output closely tracks the desired setpoint with negligible deviation once the system has settled.
- **Overshoot:** It was observed that the system exhibited an overshoot of 27% for all controller configurations. This overshoot represents the percentage by which the system's response exceeds the desired setpoint before settling. While overshoot can lead to transient oscillations, the observed value remains within an acceptable range for many applications.



$$Overshoot = 0.270/1 * 100 = 27\%$$

6. Conclusion

In this study, we designed and evaluated P, PI, and PID controllers using the Ziegler-Nichols tuning method for the given system represented by the open-loop transfer function

$$\frac{20}{s^3 + 15s^2 + 10s}$$

. The controllers were simulated in MATLAB/Simulink, and their responses to a step input were analyzed to assess their performance.