Industrial Control

Nov 2021

Mahsa Seraji
Dr. Shasadeghi

INSTRUCTIONS

Experimental Modeling and Control Design for Industrial Systems:

- **Model Estimation**: Applied Standard Methods to Estimate First and Second-Order Models with Time Delay for a Process System
    - First-Order: Obtaining Time Constant Parameter τ through 4 Methods:
        - Entire Step Response
        - Tangent Intersection Method
        - 63% Final Value
        - Information from 2 Points (with Ranges of 0.2 and 0.63 of the Final Value)
    - Plotted Process Response Curve (with Model Curves from the Four Methods)
    - Estimated Second-Order Model with Time Delay for 3 States:
        1. Critical Overdamped State
        2. Supercritical Damped State
        3. Subcritical Damped State
- **Controller Design**: Selecting a First-Order Approximation and Designing a Controller with these Objectives:
    - Attaining a Phase Margin (PM) of 45 degrees by Determining the Intersection Point of the Nyquist Curve with a Unit Circle
    - Achieving Gain Margin (GM) of 6 decibels by Pinpointing the Frequency Cutting 180 degrees in the System's Bode Diagram
    - Pursuing both Previous PM and GM Goals
- **Experimental Z-N Method**
    - Implemented Controller Design in Closed and Open-Loop using the Ziegler-Nichols Method

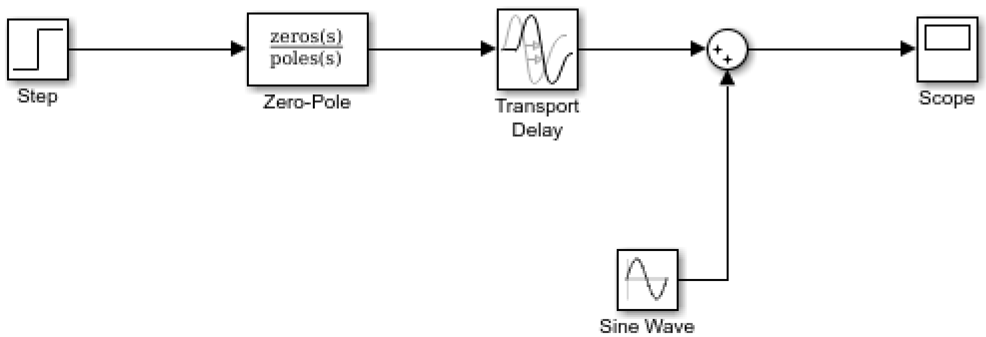
1) **Model Estimation Expertise:** I demonstrated proficiency in estimating first and second-order process models with time delay. This involved employing a spectrum of standard methods, including the analysis of entire step responses, tangent intersection methods with maximum slope, identifying the point where the output reaches 63% of the final value, and incorporating information from two strategically chosen points.

Model or Transformation Function in this Report:

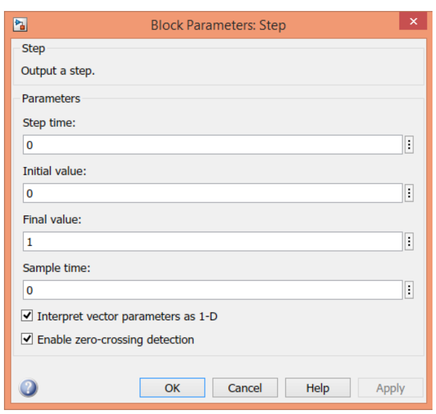
$$\frac{15}{(s+1)(s+2)(s+2)}e^{-0.6s}$$

First order

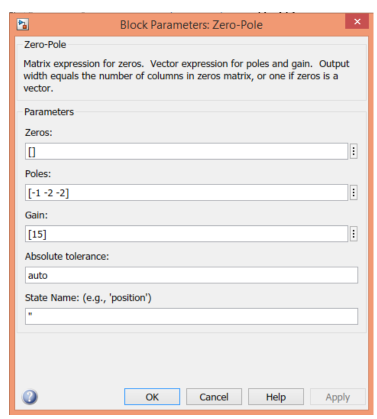
Simulink:



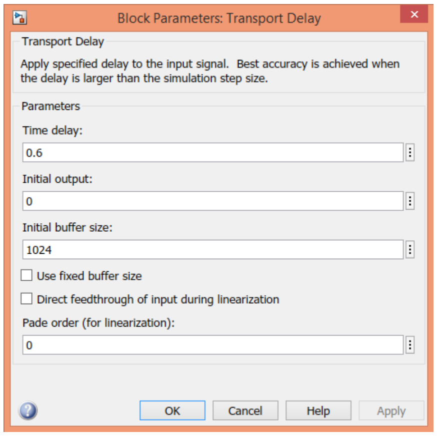
Settings:



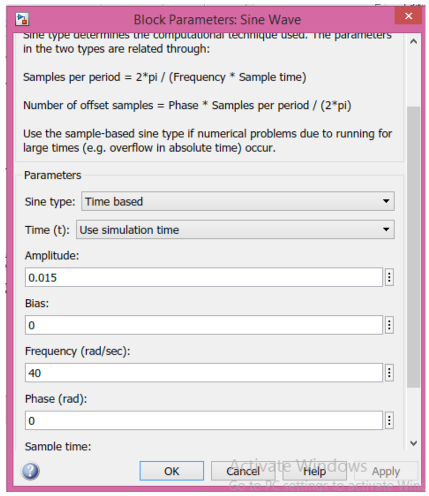
step time:0



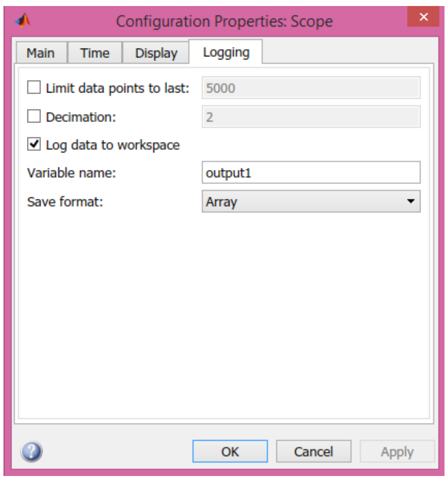
zeros : []
poles : [-1 -2 -2]
gain : [15]



time delay = 0.6



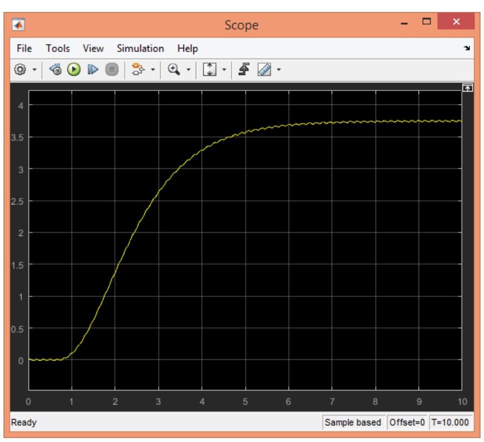
amplitude: 0.015
Frequency: 40



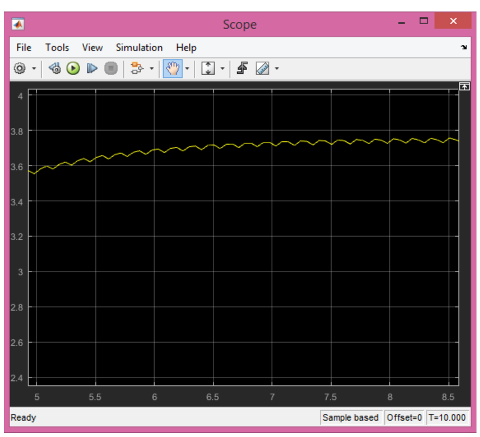
Variable name: output1

Format: array

Scope's input:



Closer look:

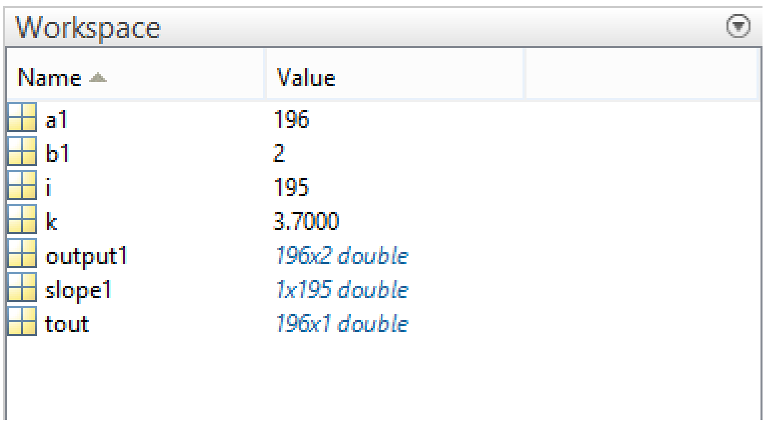


→ It can be seen that there are 196 data.

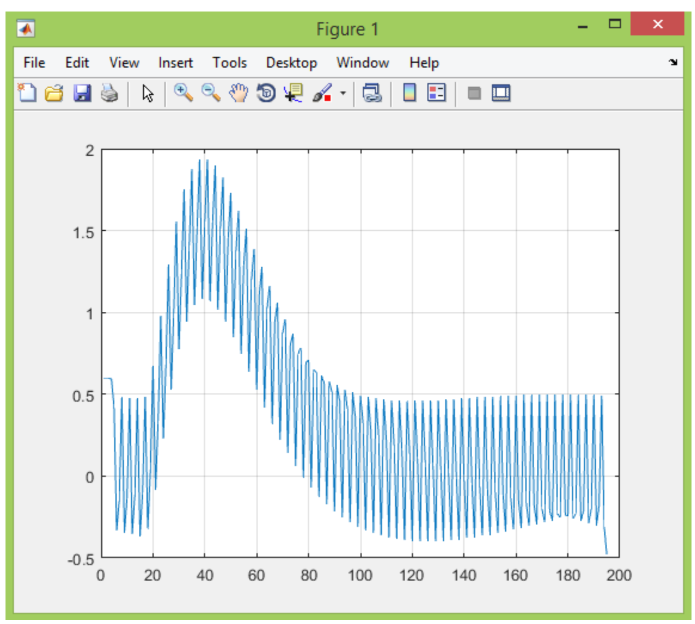
K=3.7

Now, we need to find the slope of the tangent line at each of these 196 points and select the maximum one and then plot it to find the τ_d
Codes:

```
k=3.7;
[a1,b1]=size(output1);
for i=1:a1-1
    slope1(1,i)=(output1(i+1,2)-output1(i,2))/(output1(i+1,1)-output1(i,1));
end
figure
plot(1:a1-1,slope1)
grid on
```



It can be seen:
a1=196, b1=2



The maximum point is at point 40.
Consequently, when we click on the slope, we have:

slope1

1x195 double

	38	39	40	41	42	43	44	45	46	47	48	49	50
1	1.9330	1.0826	1.5551	1.9355	1.0713	1.5528	1.8951	1.0218	1.5156	1.8229	0.9449	1.4539	1.
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													

If we click on output1:

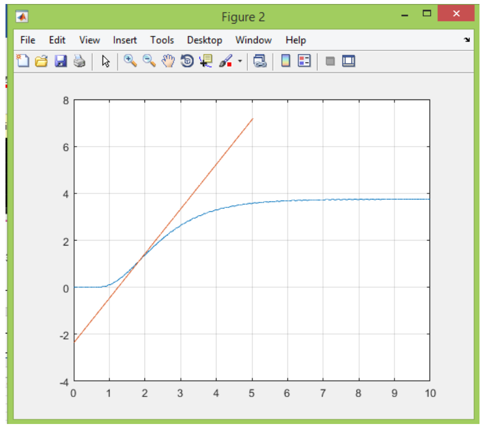
slope1 output1

196x2 double

	1	2	3
96	4.7682	3.5398	
97	4.8208	3.5249	
98	4.8735	3.5517	
99	4.9261	3.5708	
100	4.9787	3.5546	
101	5.0313	3.5806	
102	5.0840	3.5976	
103	5.1366	3.5802	
104	5.1892	3.6056	
105	5.2419	3.6205	
106	5.2945	3.6022	
107	5.3471	3.6272	

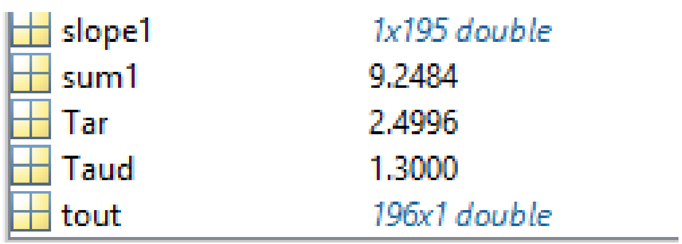
→ By the fifth second, it becomes 101

```
max_slope1=40;
mslope=1.9;
for j=1:101
    maxslopeline(j)=(mslope*output1(j,1))+(output1(max_slope1,2)-
mslope*output1(max_slope1,1));
end
figure
plot(output1(:,1),output1(:,2))
hold on
plot(output1(1:101,1),maxslopeline)
grid on
```



$$\tau_d = 1.3$$

```
Taud=1.3;%timedelayestimation
sum1=0;
for j=1:a1-1
    area1=(output1(j+1,1)-output1(j,1))*(2*k-output1(j+1,2)-output1(j,2))*0.5;
    sum1=sum1+area1;
end
Tar=sum1/k;
```

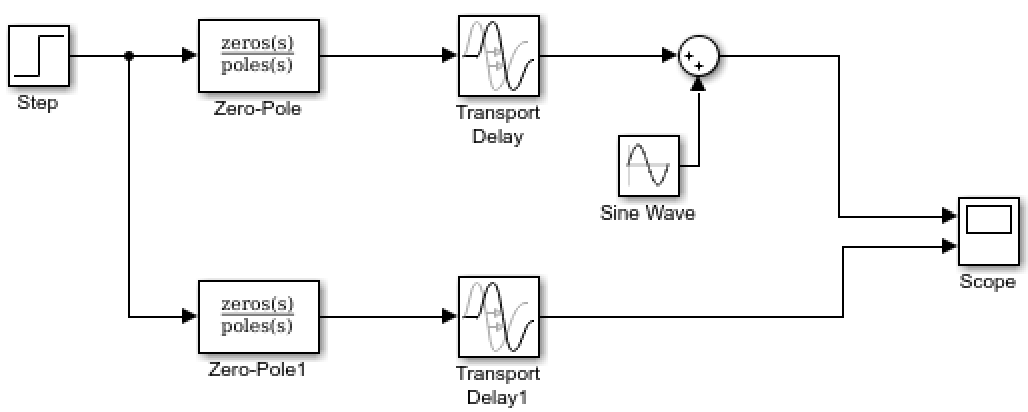


slope1	1x195 double
sum1	9.2484
Tar	2.4996
Taud	1.3000
tout	196x1 double

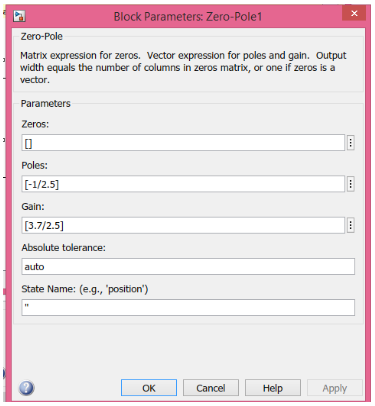
$$\tau = 2.4996 \sim 2.5$$

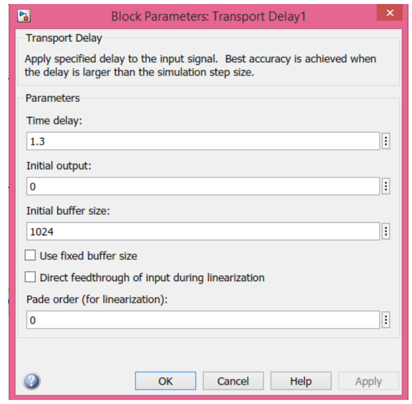
Now we have all the necessary parameters to estimate our transformation function.

$$k = 3.7 \quad \tau_d = 1.3 \quad \tau = 2.5$$

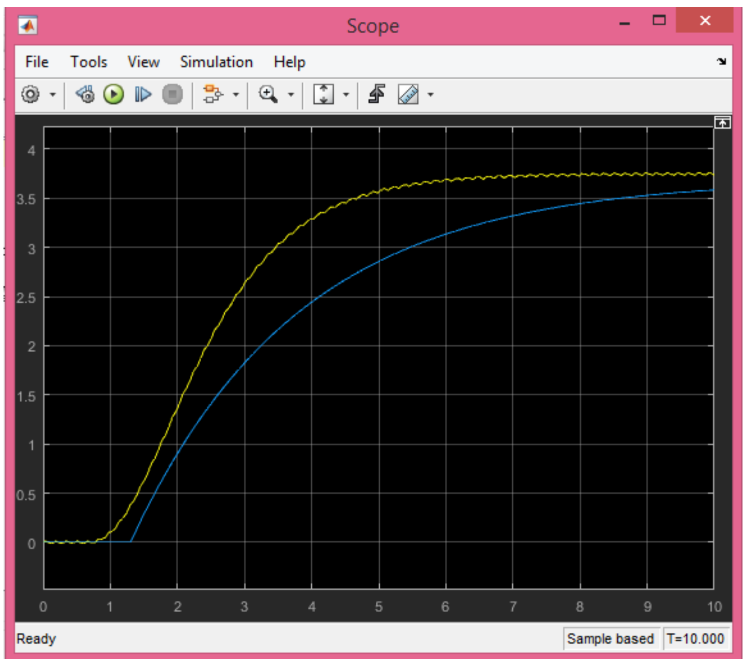


In which our new function lies:





Which we have in the final figure: (First method: Using the entire step response):



Now we need to obtain the transfer function using other methods:

Second method (Maximum slope intersect method) with K=3.7

```
Taud2=1.3;%ravesh2(shib-max)
Tau2=3.4-Taud2;
```

$$\tau_{d2} = 1.3$$

و

$$\tau_2 = 2.1$$

Third method (0.63 final value)

```
g = 0.63 * 3.7 ;%ravesh3-y(t0.63)=2.3-->t=3
Tau3 = 3-Taud;
```

$$\tau_3 = 3 - \tau_d = 1.7$$

Fourth method (2-point 0.63 and 0.28 final value)

```
g = 0.63 * 3.7; %ravesh4 , t =3
f = 0.28 * 3.7; %t(y=0.28k)=1.8
Tau4=1.5 * (3-1.8);
Taud4= (3-Tau4);
```

$$\tau_4 = 1.8$$
$$\tau_{d4} = 1.2$$

Now we need to plot the process response curve along with the response curves of the models obtained using these 4 methods:

$$G_1(s) = \frac{3.7/2.5}{s + (1/2.5)}$$

$$G_2(s) = \frac{3.7/2.1}{s + (1/2.1)}$$

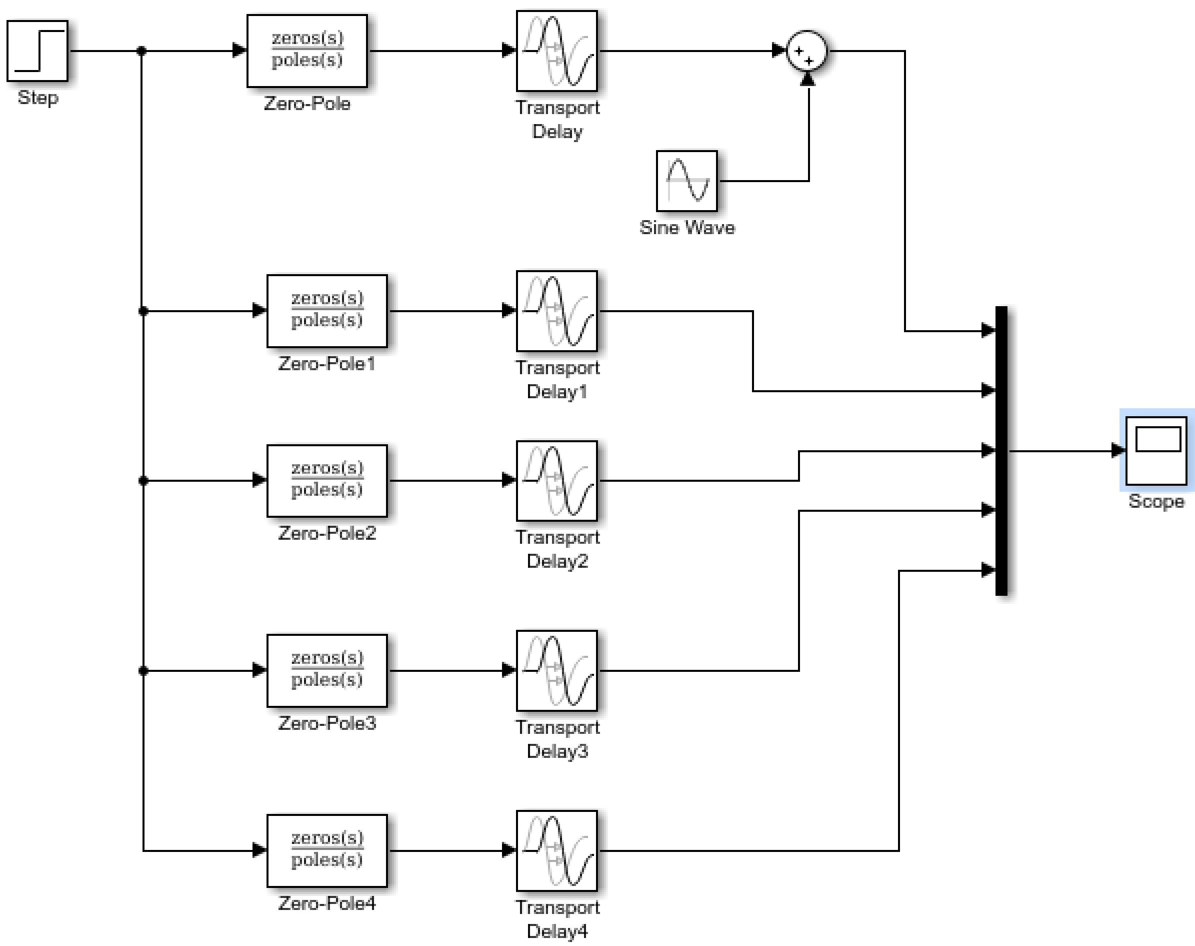
$$G_3(s) = \frac{3.7/1.7}{s + (1/1.7)}$$

$$\tau_{d1} = \tau_{d2} = \tau_{d3} = 1.3$$

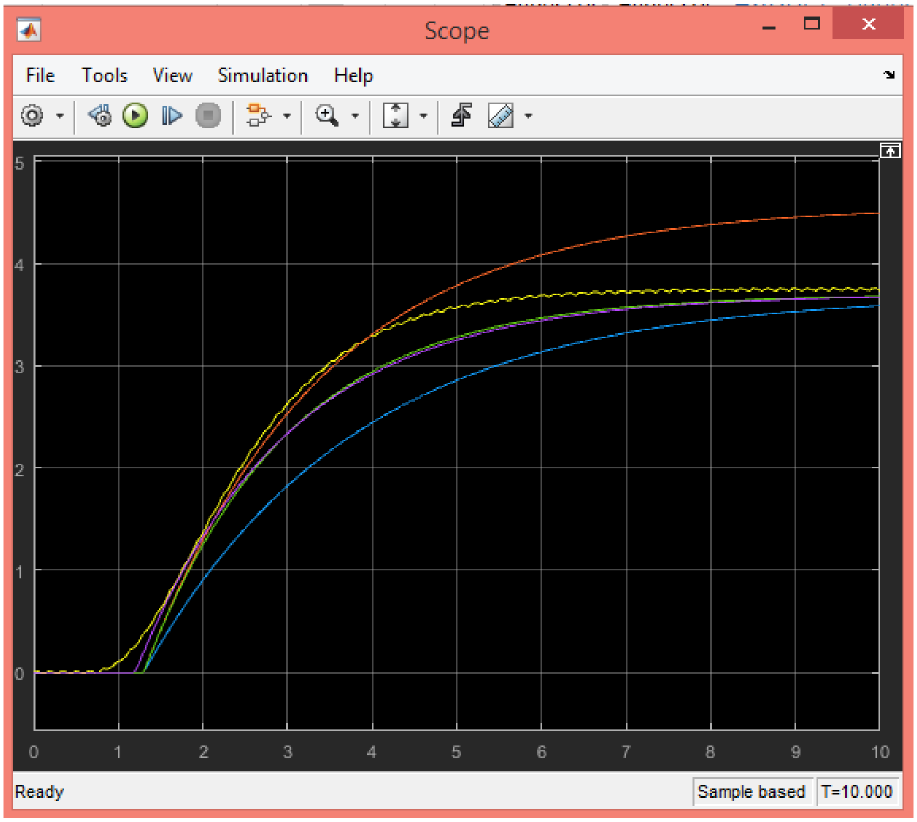
$$G_4(s) = \frac{3.7/1.8}{s + (1/1.8)}$$

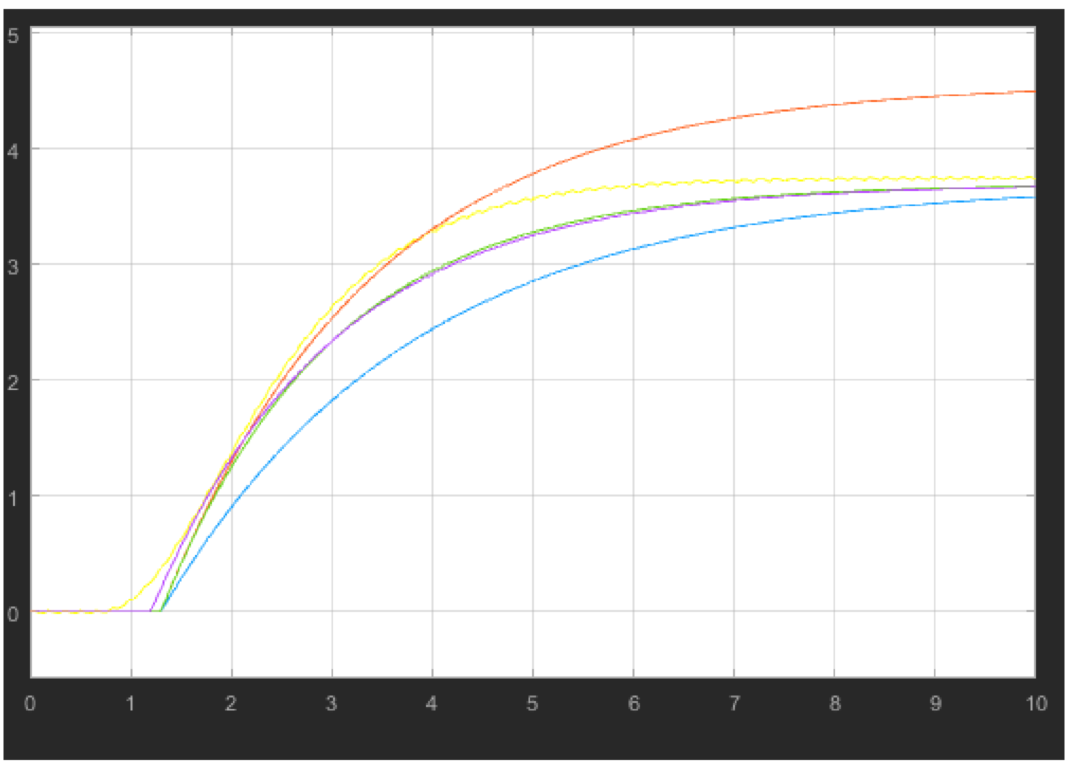
$$\tau_{d4} = 1.2$$

Symbolic representation of the obtained models:



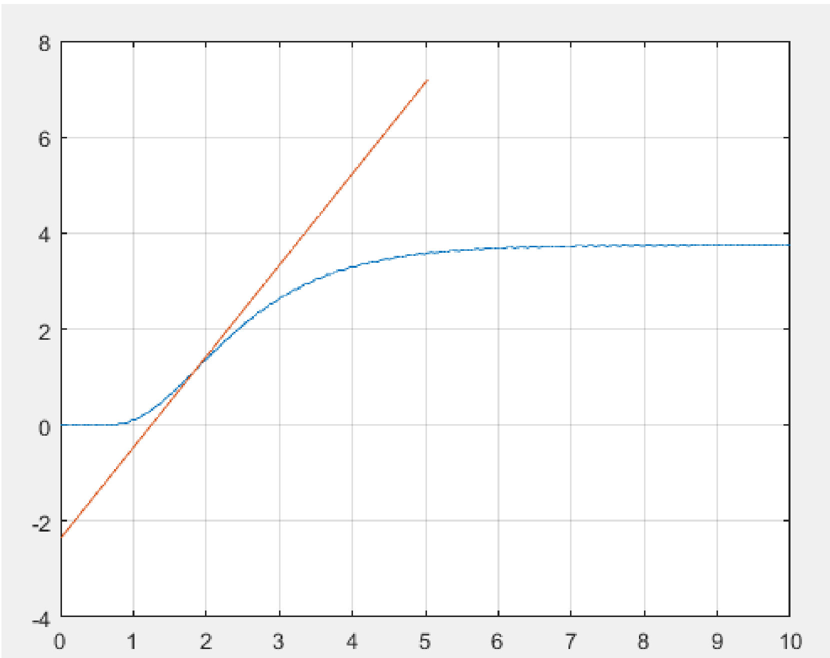
The symbolic representation is as follows: Output1





Second order:

First case: Critical Damped Second Order Model
Given the output shape and maximum slope from the previous section, we have:



We use point $y(\tau + \tau_d) = 0.264k$ to determine the parameters of this model

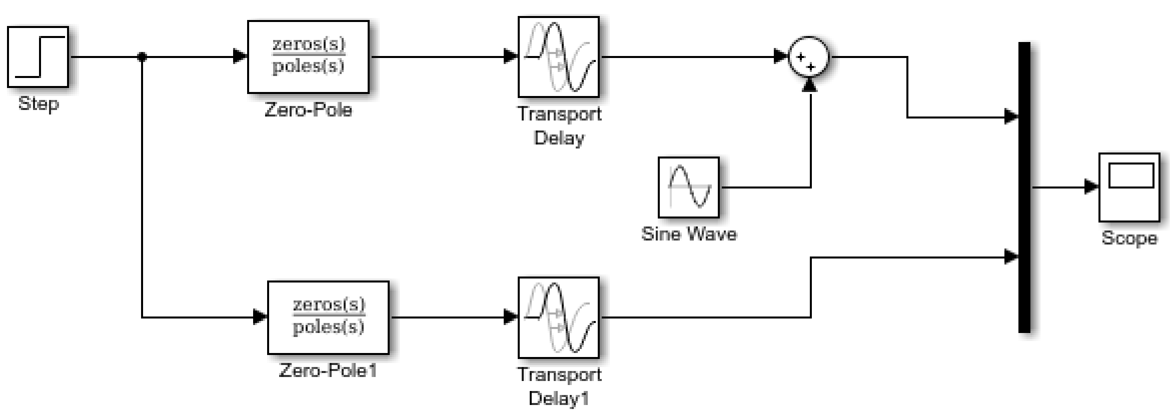
```
m = 0.264 * 3.7 ;%y(tau+taud)=y(0.9)=1.8
Tau =1.8 - 1.3;
```

$$\tau = 0.5$$

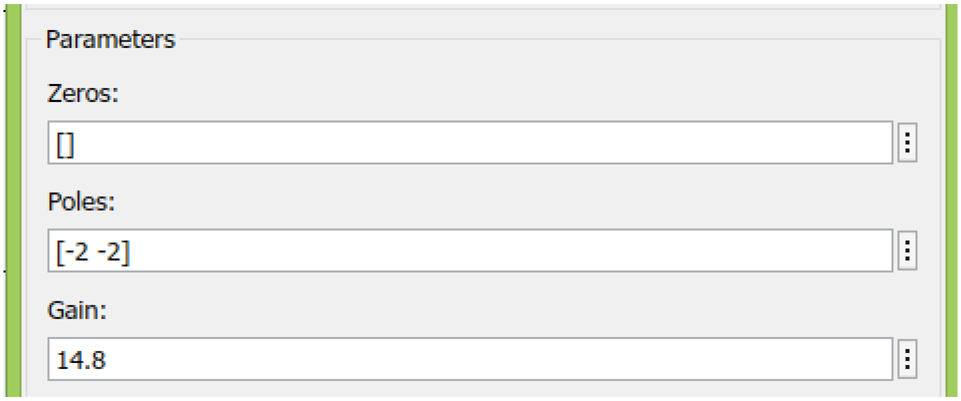
Consequently, the second-order model becomes as follows:

$$G(s) = \frac{k}{(\tau s + 1)^2} e^{-\tau_d s} = \frac{3.7}{0.25s^2 + s + 1} e^{-1.3s}$$

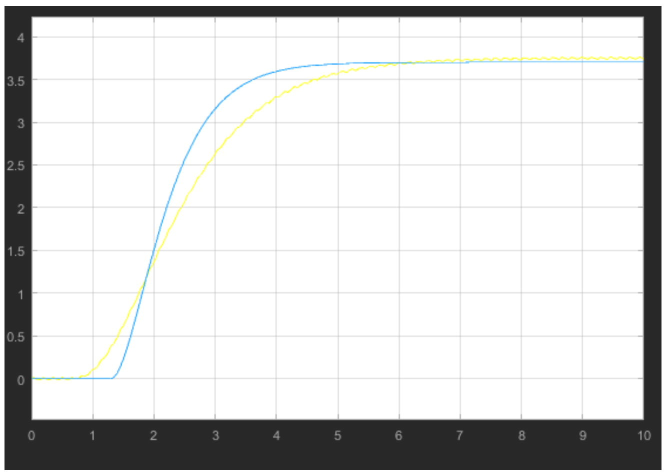
Scope's outputs:



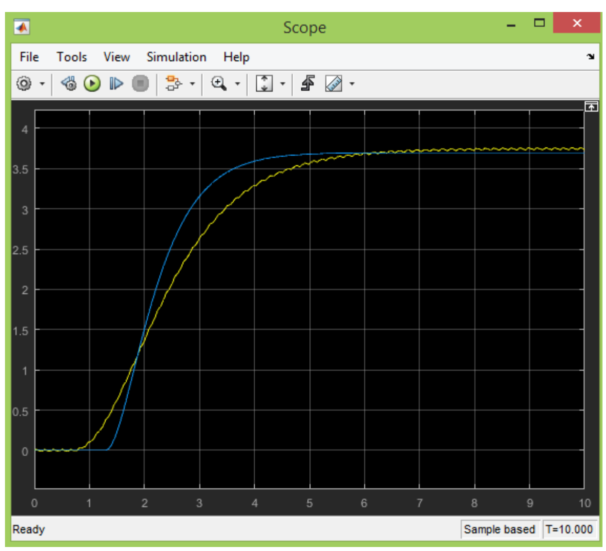
Settings for the first second-order model:



Output:



More specifically:



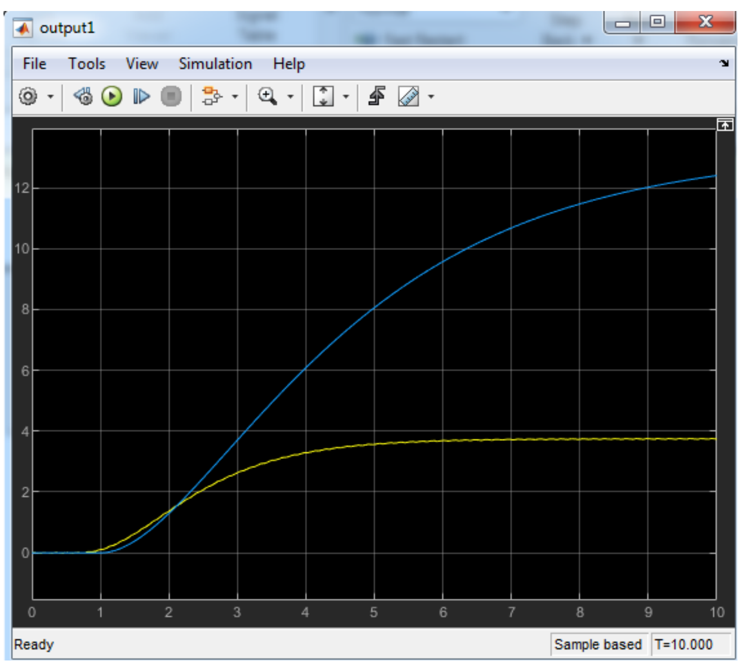
Second case: Overdamped Second Order Model

Second way $(T_{ar} = \tau_1 + \tau_2 + \tau_d)$

$$G(s) = \frac{k}{(\tau_1 s + 1)(\tau_2 s + 1)} e^{-\tau_d s}$$

To obtain the parameters in this method, we need to derive the following formulas:
Given the graph with maximum slope:

$$t_m = 5$$
$$t_x = 0.3$$
$$T_{ar} = sum1 = 2.5$$
$$\frac{k_s}{k} = \frac{1}{t_m - t_x} \rightarrow \frac{k_s}{3.7} = \frac{1}{4.7} \rightarrow k_s = 0.7$$
$$\lambda = \frac{(t_m - T_{ar})k_s}{k} = 0.4$$
$$\lambda = 0.4 = \chi e^{-\chi} \rightarrow \chi = \frac{\ln \eta}{\eta - 1} \rightarrow \eta \cong 0.6$$
$$\frac{k_s}{k} = \frac{\eta^{\frac{1}{1-\eta}}}{\tau_1} \rightarrow \tau_1 = 1.4$$
$$\eta = \frac{\tau_1}{\tau_2} \rightarrow \tau_2 = 2.3$$
$$\tau_d = T_{ar} - \tau_1 - \tau_2 = 1.02$$
$$G(s) = \frac{3.7}{(1.4s + 1)(2.3s + 1)} e^{-s}$$



Third method: Similar to the second method but without time delay

Third case (Fourth method): Subcritical Second Order Model

In this case, we have:

$$G(s) = \frac{k\omega^2}{s^2 + 2\xi\omega + \omega^2}$$

We must obtain ω و ξ :

$$\omega = \frac{2\pi}{T_P\sqrt{1-\xi^2}}$$

$$\xi = \frac{1}{\sqrt{1+(\frac{2\pi}{lnd})^2}}$$

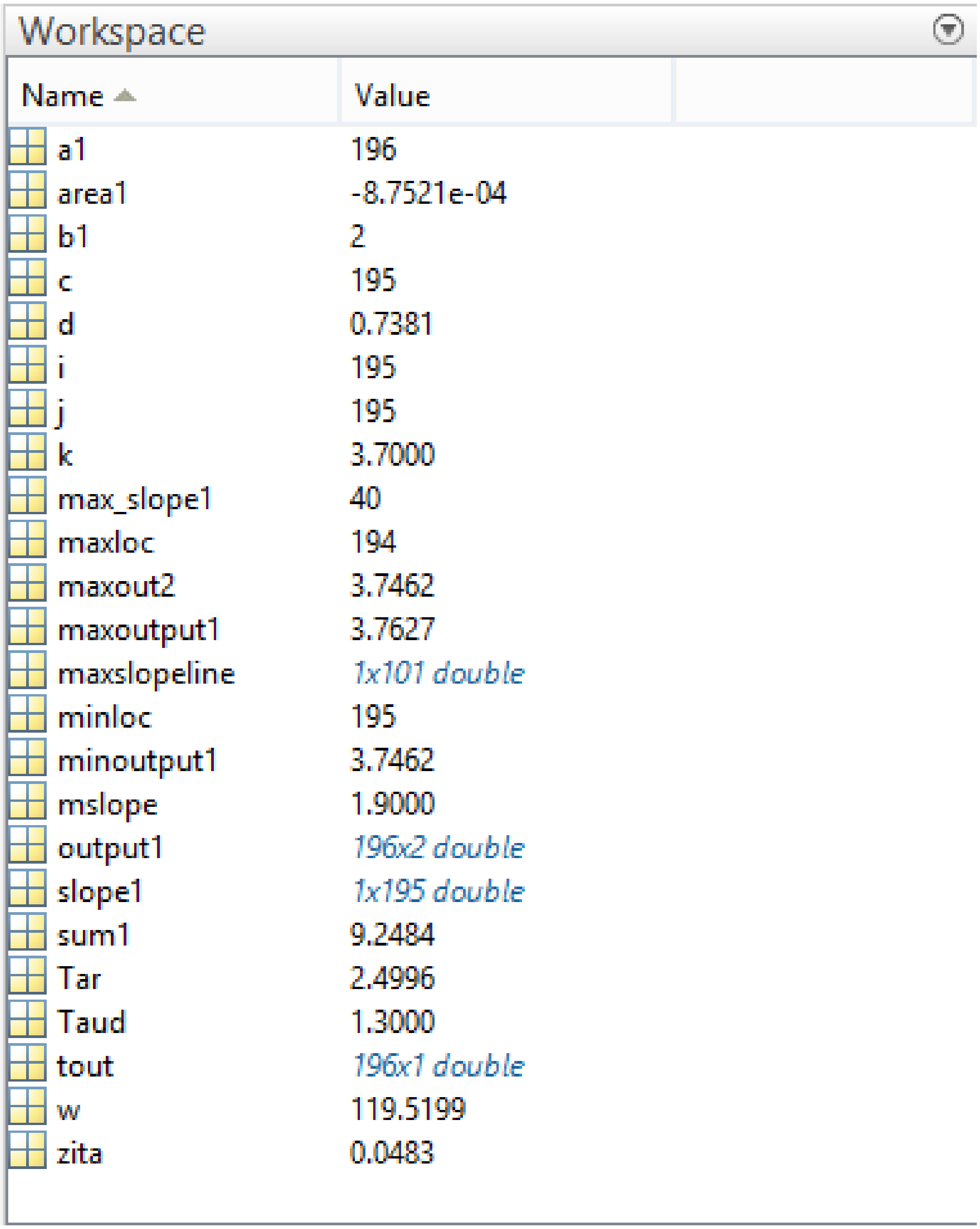
Now we should get d:

$$d = \frac{e_2}{e_1}$$

Codes:

```
maxoutput1=0;
for i=1:a1-1
    if output1(i,2)> maxoutput1
        maxoutput1=output1(i,2);
        maxloc=i;
    end
end
minoutput1=maxoutput1;
for j=maxloc : a1-1
    if output1(j,2)<minoutput1
    minoutput1=output1(j,2);
    minloc=j;
    end
end
maxout2=minoutput1;
for c = minloc:a1-1
    if output1(c,2)>maxout2
        maxout2=output1(c,2);
        maxloc2=c;
    end
end
d=(maxout2-3.7)/(maxoutput1-3.7);
zita=1/sqrt(1+(2*pi/log(d))^2);
w =(2*pi)/((output1(maxloc2,1)-output1(maxloc,1))*sqrt(1-zita^2));
```

results:



Workspace

Name	Value
a1	196
area1	-8.7521e-04
b1	2
c	195
d	0.7381
i	195
j	195
k	3.7000
max_slope1	40
maxloc	194
maxout2	3.7462
maxoutput1	3.7627
maxslopeline	1x101 double
minloc	195
minoutput1	3.7462
mslope	1.9000
output1	196x2 double
slope1	1x195 double
sum1	9.2484
Tar	2.4996
Taud	1.3000
tout	196x1 double
w	119.5199
zita	0.0483

$$d = 0.7$$
$$\xi = 0.048 \cong 0.05$$
$$\omega = 119$$

As a result, our approximate model looks like this:

$$G(s) = \frac{52395}{s^2 + 11.9s + 14161}$$

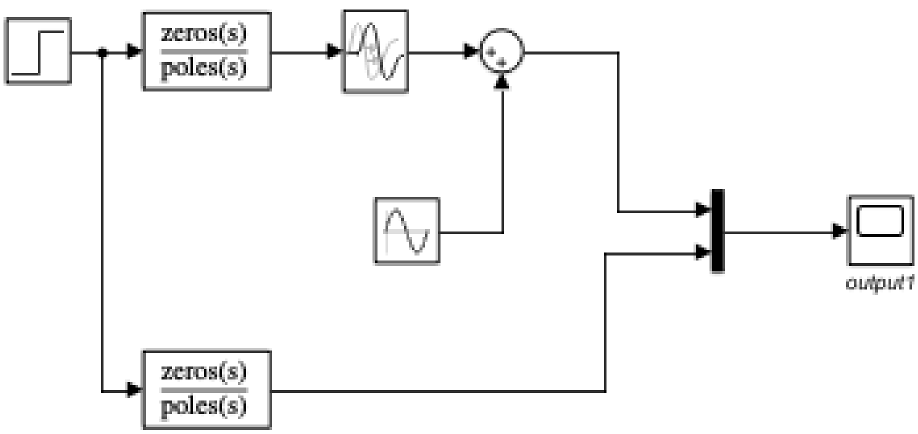
Maxloc2 error: Despite placing the program in MATLAB, it seems to output 'C' instead. Instead, the final answer appears to be incorrect and significantly larger than reasonable, so I calculated it manually and with an error of 'w'.

$$Tp \cong 140 \rightarrow \omega \cong 0.04$$

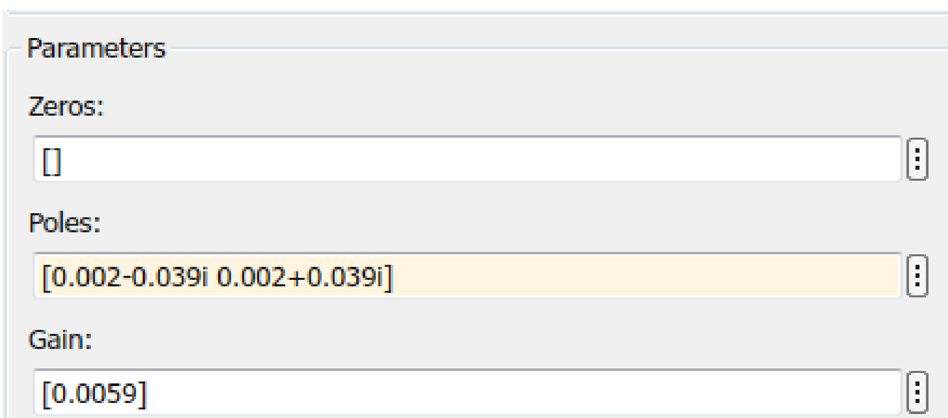
The final transformation function becomes as follows:

$$G(s) = \frac{0.0059}{s^2 + 0.004s + 0.0016}$$

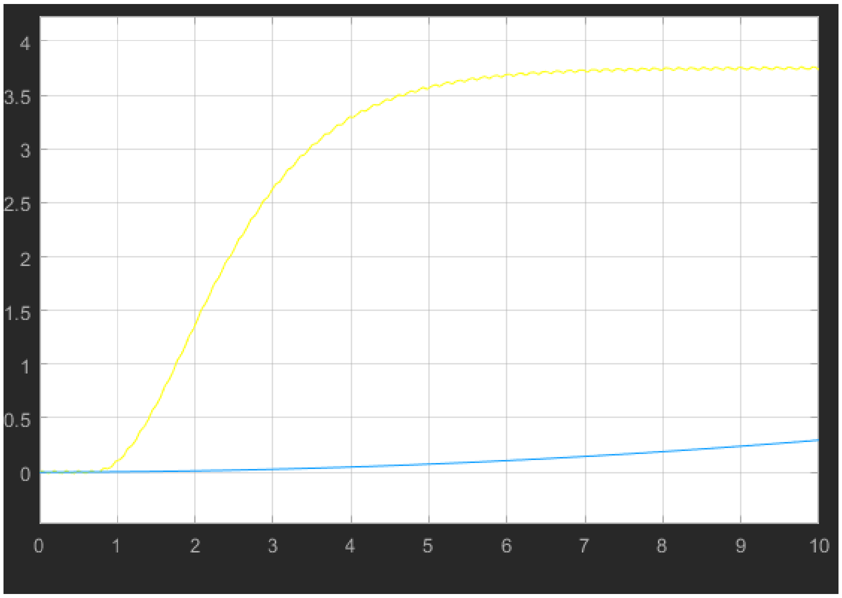
Simulink:



- As observed, my Simulink program displays the transformation function section as above, and I need to enter the zeros and poles in the settings. However, the poles of my transformation function come out as complex, and after running the program, it gives an error:


Parameters
Zeros:
[]
Poles:
[0.002-0.039i 0.002+0.039i]
Gain:
[0.0059]

And the output of the scope is:



➔ Modeling a second-order system for a third-order function is better. Some of the issues with first-order modeling include the fact that at the start, the derivative change in the output response is not zero, and also that a first-order model cannot capture oscillations effectively.

2) **Controller Design Mastery:** With a keen focus on precision, I excelled in selecting a first-order approximation and adeptly designing controllers to achieve specific objectives. This encompassed attaining a Phase Margin (PM) of 45 degrees by determining the intersection point of the Nyquist curve with a unit circle, achieving Gain Margin (GM) of 6 decibels by pinpointing the frequency cutting 180 degrees in the system's Bode diagram, and simultaneously pursuing both PM and GM goals.

First, we use one of the first and second-order approximations obtained in the first project.
Original system:

$$G(s) = \frac{15}{(s+1)(s+2)(s+2)} e^{-0.6s}$$

First-order approximation:

$$G_1(s) = \frac{3.7/2.5}{s + (1/2.5)} e^{-1.3s}$$

1) Obtain a Phase Margin (PM) of 45 degrees.
2) Obtain a Gain Margin (GM) of 6 dB.
3) Achieve both a PM of 45 degrees and a GM of 6 dB.

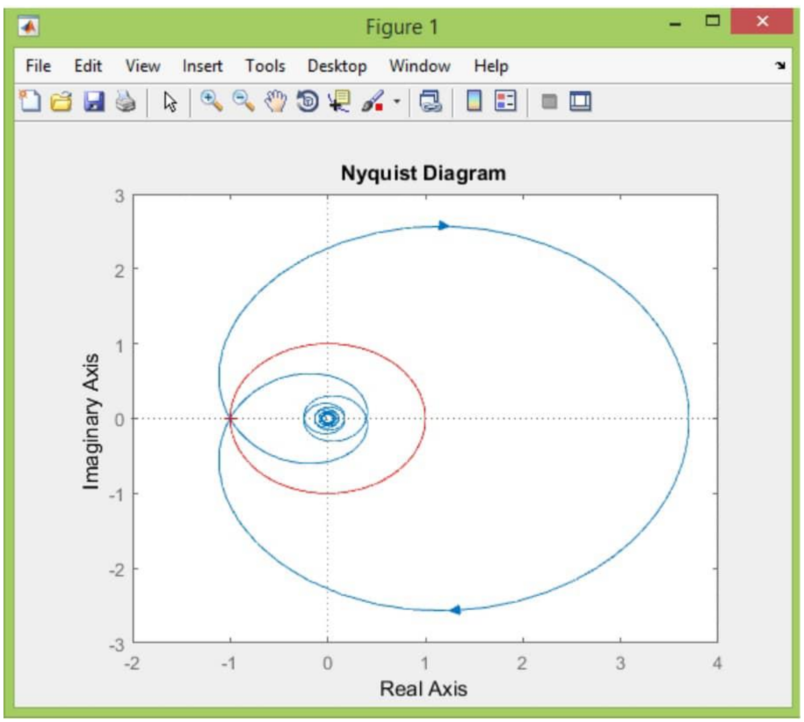
1-In the first scenario, where we aim to achieve a phase margin (PM) of 45 degrees, we first need to plot the Nyquist diagram of the system. We want to design a controller that can bring the PM of this system to 45 degrees and improve the system's performance. Therefore, we plot the unit circle on the

Nyquist diagram of the system to find the current PM. Then, we design the controller using the following mathematical relationships:

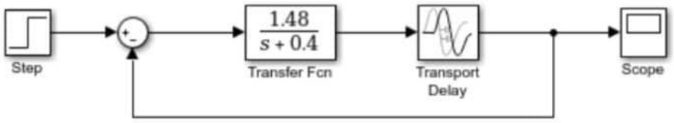
$$\left.\begin{array}{ll} A: & G(j\omega_0)=r_A e^{j\varphi_A} \\ B: & GC(j\omega_0)=r_B e^{j\varphi_B} \\ Controller: & C(j\omega_0)=r_C e^{j\varphi_C} \end{array}\right\} \Rightarrow \begin{cases} r_C=\dfrac{r_B}{r_A} \\ \varphi_C=\varphi_B-\varphi_A \end{cases}$$

```
s=tf('s');
G=3.7/(2.5*s+1);
G1=G*exp(-1.3*s);
nyquist(G1);
r=1;
for j=0:360
    x(j+1)=r*cos((pi/180)*j);
    y(j+1)=r*sin((pi/180)*j);
end
hold on
plot(x,y,'r')
```

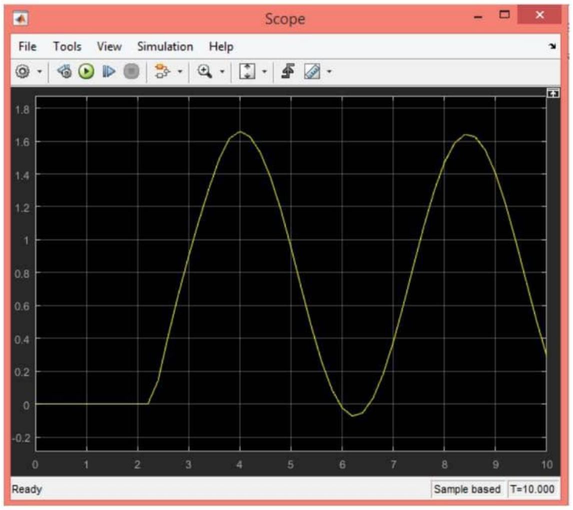
now we have:



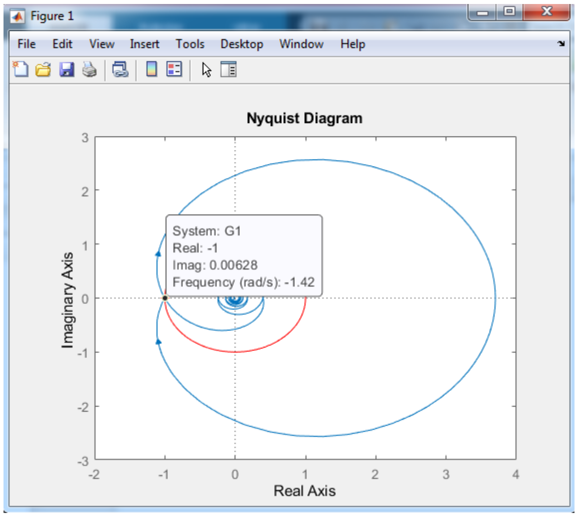
We need to find the intersection point of the curve with the unit circle, which is observed to be exactly at (x=-1, y=0). The reason for this is the instability of my system and its complete oscillation.



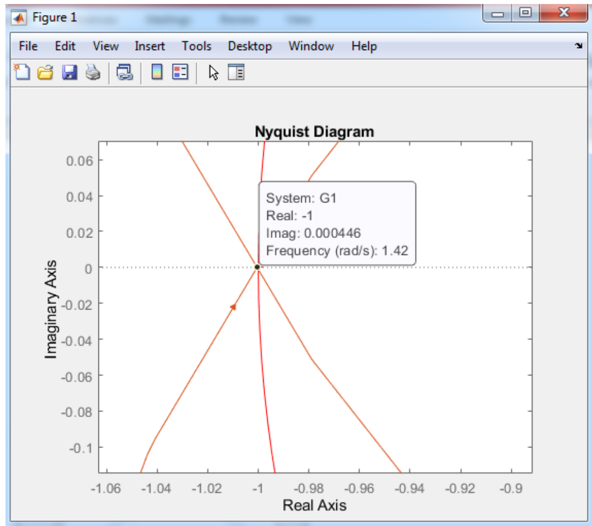
Output:



The intersection point of the Nyquist curve with the unit circle:

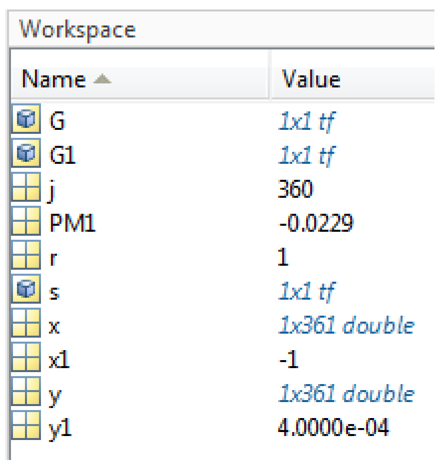


Closer look:




```
x1=-1;
y1=0.0004;
PM1=atan(y1/x1)*(180/pi);
```

So we have:



Workspace

Name	Value
G	1x1 tf
G1	1x1 tf
j	360
PM1	-0.0229
r	1
s	1x1 tf
x	1x361 double
x1	-1
y	1x361 double
y1	4.0000e-04

Now we will determine the PID coefficients based on the following formulas:

$$Kp = \frac{rB}{rA}\cos(\varphi B - \varphi A)$$

$$Ti = -\frac{1}{\tan(\varphi B - \varphi A)\,\omega 0}$$

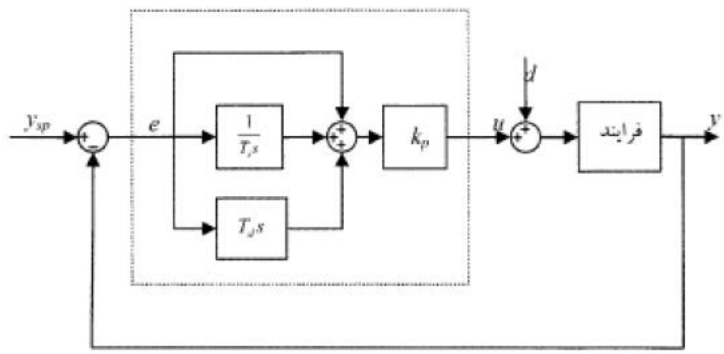
$$Td = 0.25Ti$$

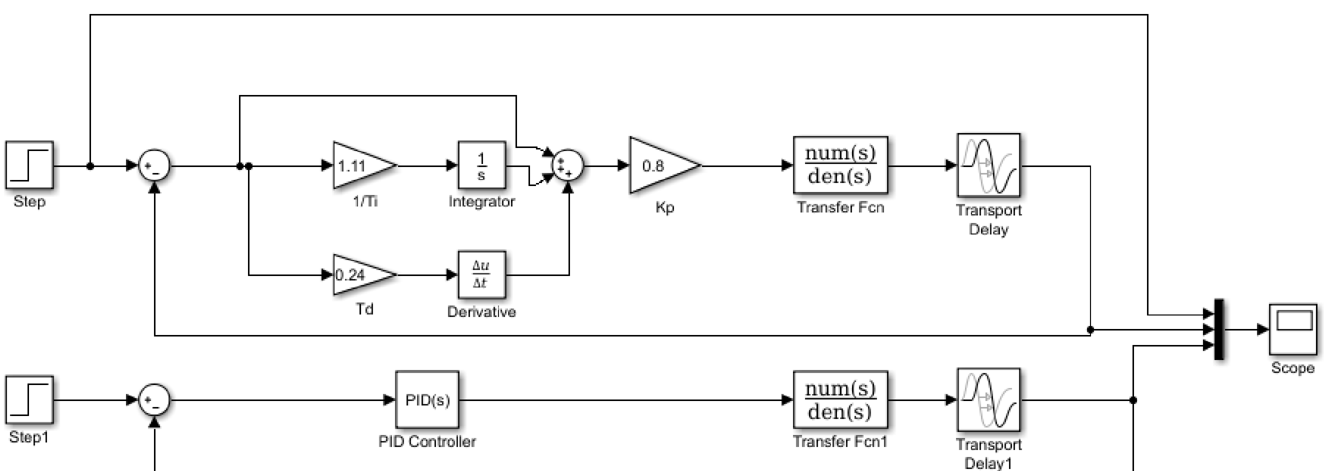
```
w0=1.42;
Kp=cos(36*(pi/180));
Ti=1/(w0*tan(36*(pi/180)));
Td=0.25*Ti;
```

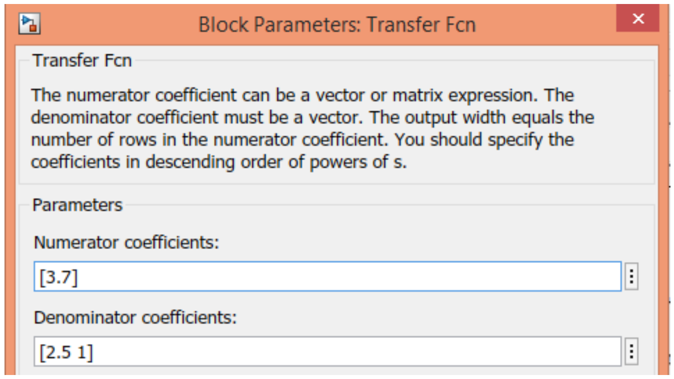
Kp	0.8090
PM1	-0.0229
r	1
s	*1x1 tf*
Td	0.2423
Ti	0.9693

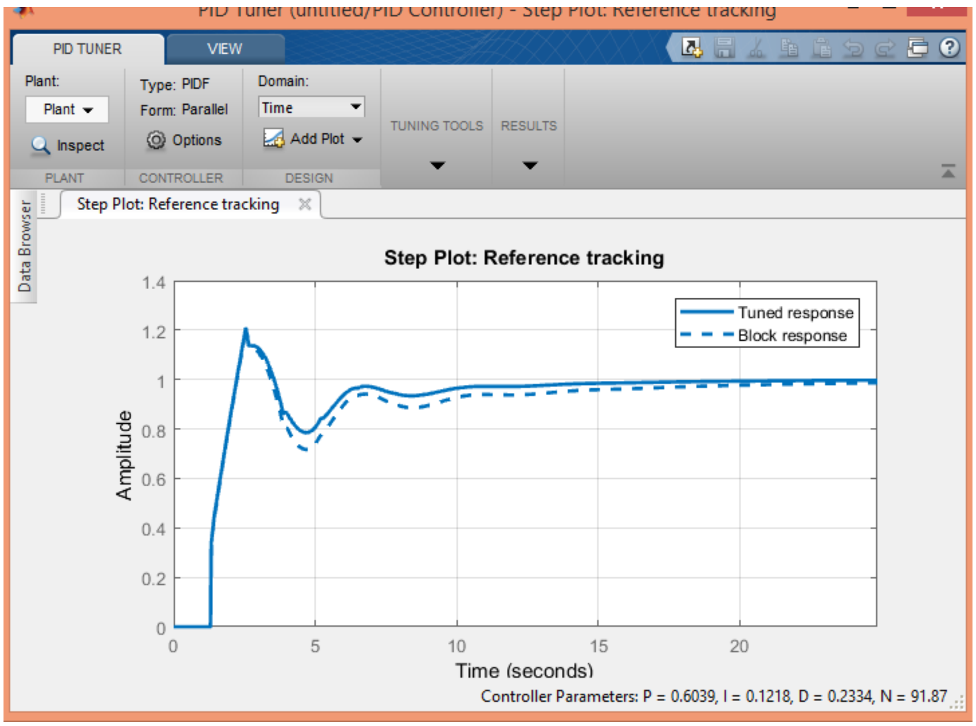
Now, in the Simulink section of MATLAB, the controller is added, and coefficient tuning is performed using the PID controller defined in MATLAB. We try to adjust the coefficients in such a way that the system response behaves similarly to the untuned system, ultimately resulting in responses as follows:

$$G(s)=k_p\left(1+\frac{1}{sT_i}+sT_d\right)$$



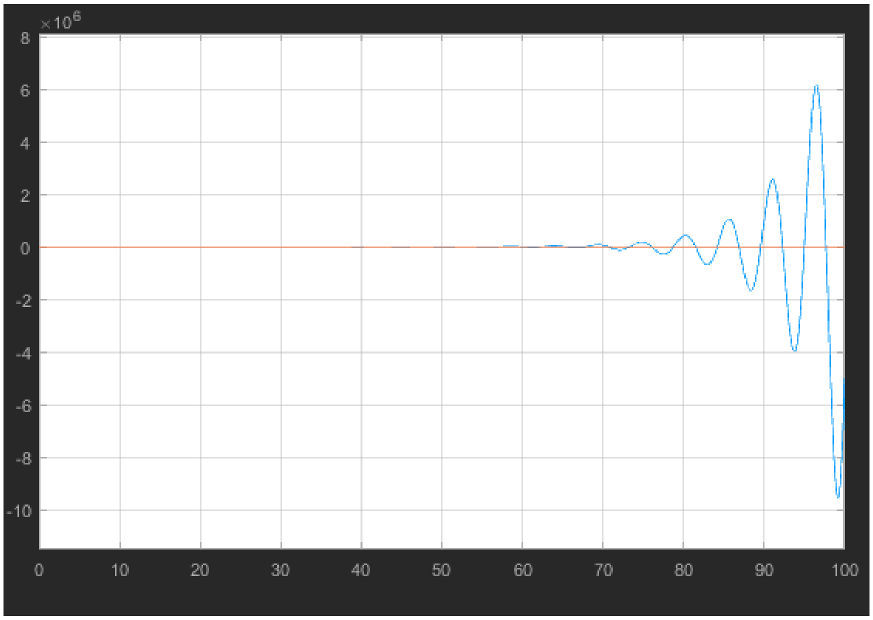






Source:	internal
Proportional (P):	0.622972825824049
Integral (I):	0.090825884216773
Derivative (D):	0.222984633718735
Filter coefficient (N):	96.19538044861

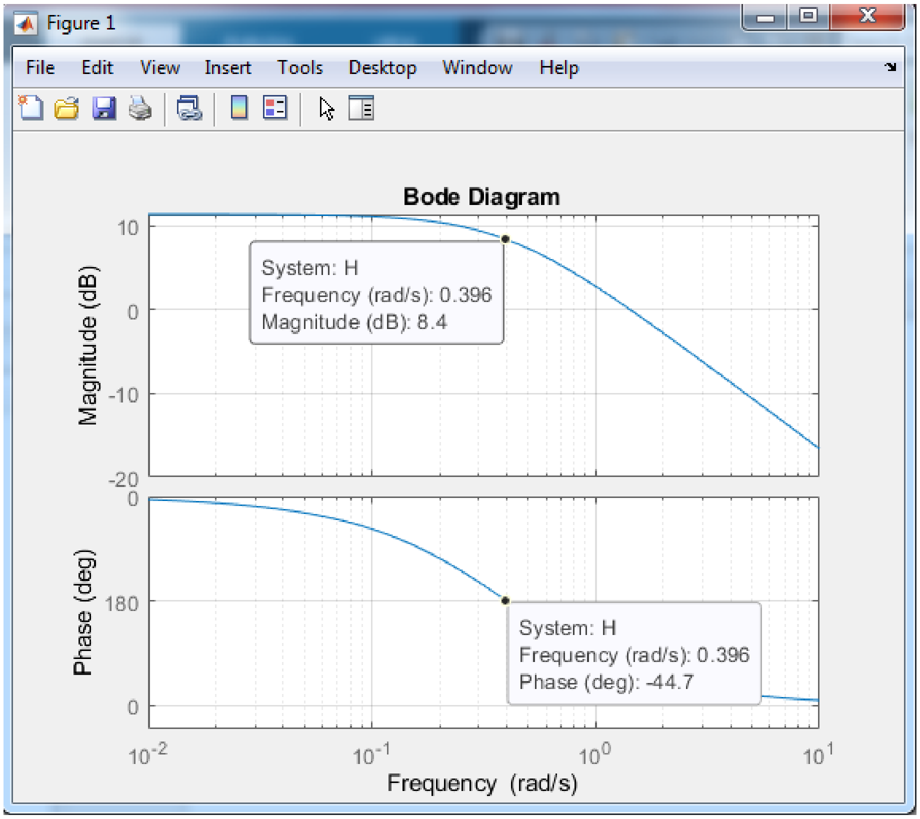
Output:



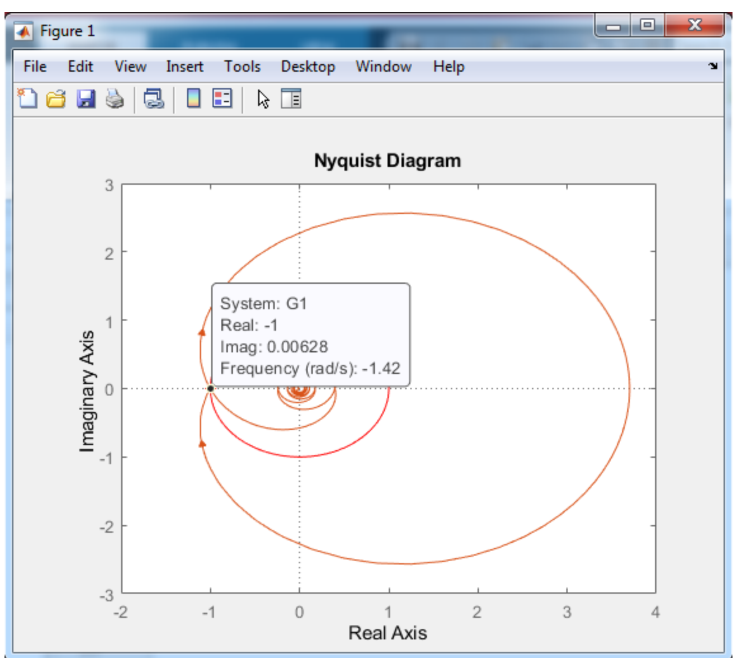
2: to increase the gain margin to 6 dB

In this scenario, we aim to adjust the system's gain margin to 6 dB. To do this, we first need to obtain the Bode plot of the system to find the frequency at which the phase margin crosses 180 degrees in the phase plot. Then, we evaluate the magnitude at that frequency in the magnitude plot and multiply it by a negative factor to find the current desired gain margin of the system:

```
clear all
close all
clc
s=tf('s');
H = (3.7/(2.5*s+1));
bode(H)
grid on
```



And also, we need to determine the intersection point with the real axis in the Nyquist diagram of the desired system. Additionally, we must find the intersection point with the real axis for the 6 dB magnitude in the controller we intend to design, which can be obtained using the following relationships:

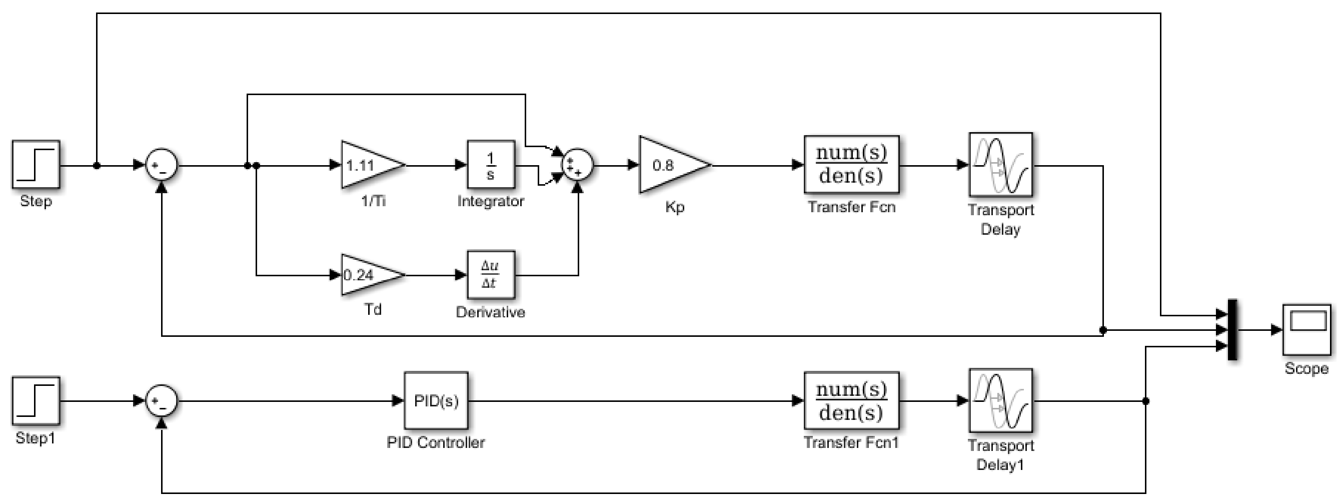



```
x1=-1;
GM1=8.4;
Ra=1;
```

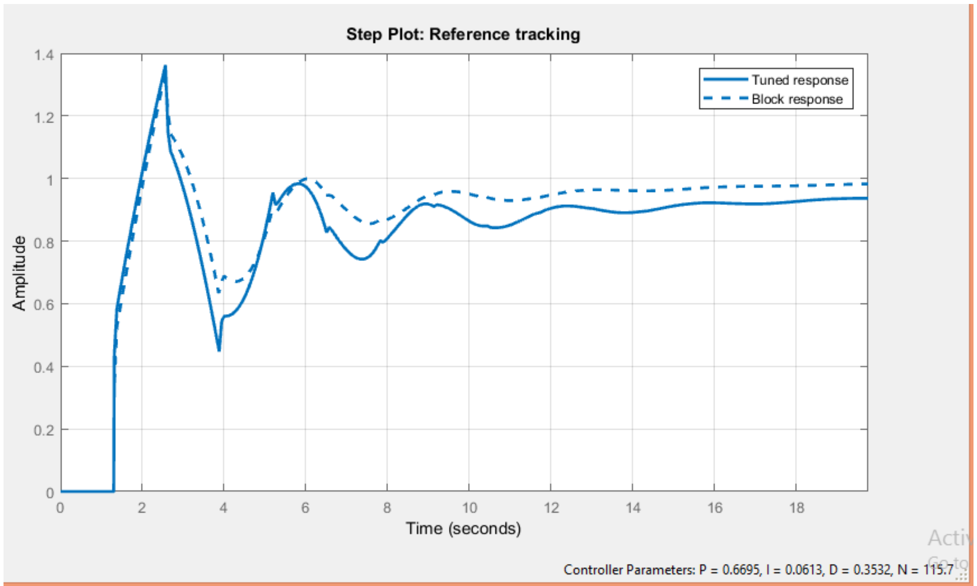
```
GM2=6;
Rb=0.5;
w0=1.42;
Kp=(Rb/Ra);
Td=1/(2*w0);
Ti=4*Td;
```

Result:

Kp	0.5000
PM1	-0.0229
r	1
Ra	1
Rb	0.5000
s	1x1 tf
Td	0.3521
Ti	1.4085

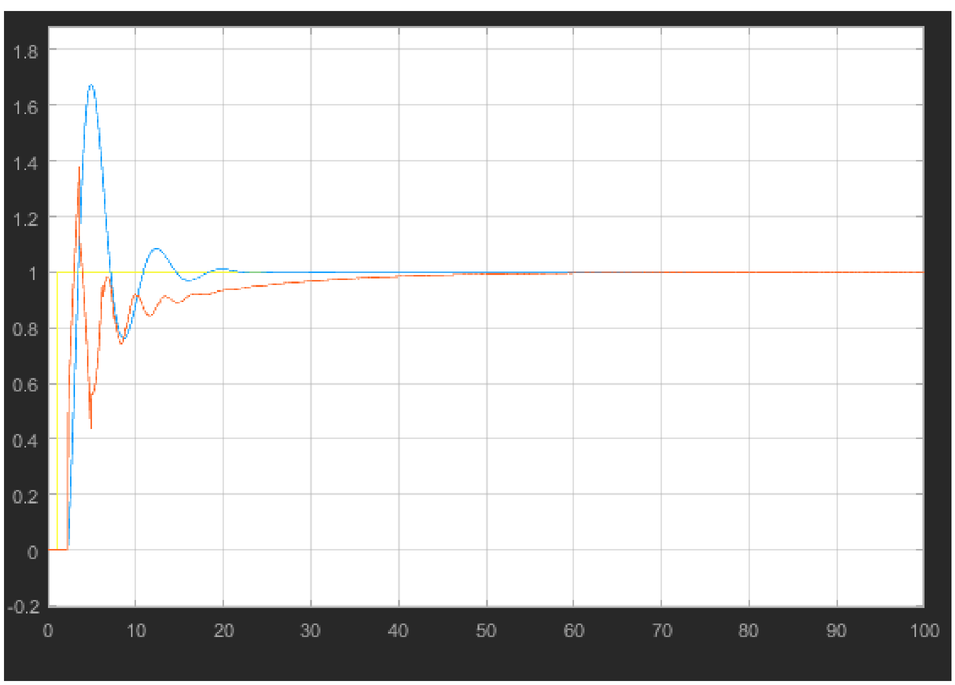


In the controller settings, we have:



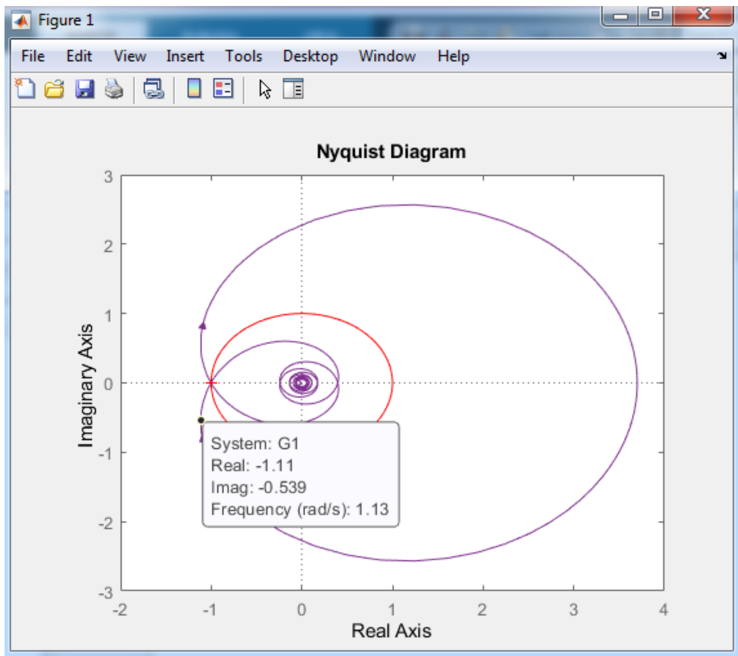
Source:	internal
Proportional (P):	0.669543854076909
Integral (I):	0.0613034641044271
Derivative (D):	0.353196183364406
Filter coefficient (N):	115.652284687855

Output:



Third case: A scenario where both 45 degrees phase margin and 6 dB gain margin are achieved.

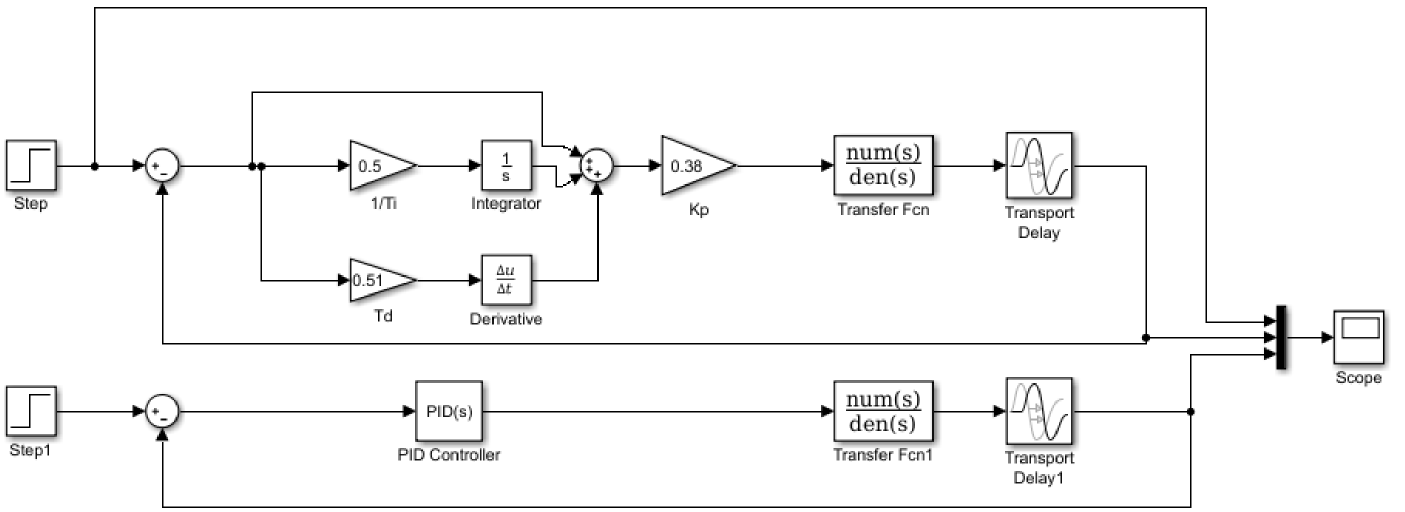
In the third part of this project, we need to adjust both GM = 6 dB and PM = 45 degrees. For this purpose, we selected a point between the intersection with the real axis and the unit circle. We obtained Phi A based on that point, and with the specifications of this point and the ratio Rb/Ra we have, we derive the controller coefficients. This point is defined as follows, and the corresponding code for this section is as follows:



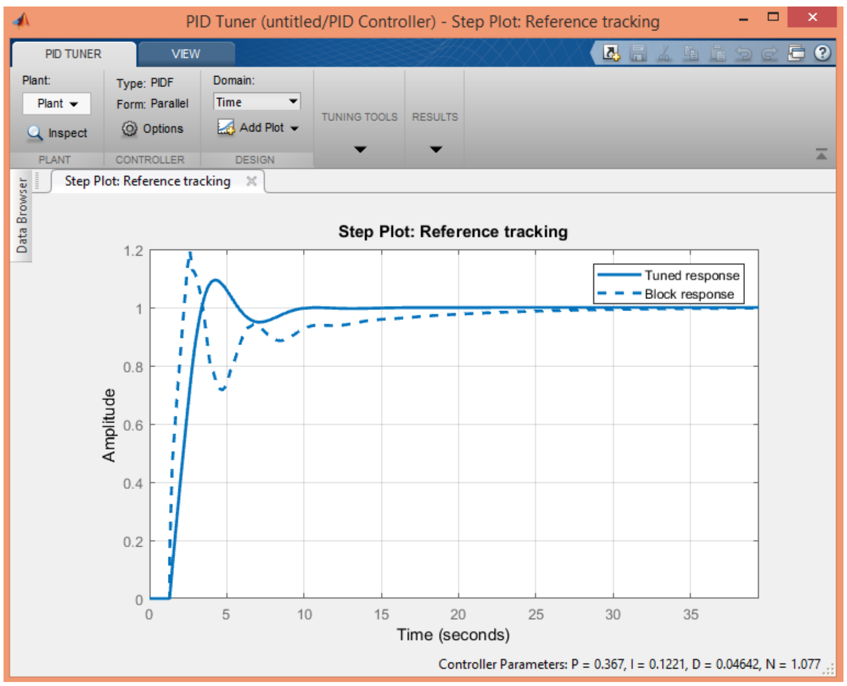

```
X2=-1.11;
Y2=-0.539;
W0=1.13;
phiA=atan(Y2/X2)*(180/pi);
Rb=0.5;
Ra=1.23;
Kp=(Rb/Ra)*cos(19*(pi/180));
Ti=1/(w0*tan(19*(pi/180)));
Td=0.25*Ti;
```

We have:

Kp	0.3844
phiA	25.9005
PM1	-0.0229
r	1
Ra	1.2300
Rb	0.5000
s	1x1 tf
Td	0.5113
Ti	2.0452

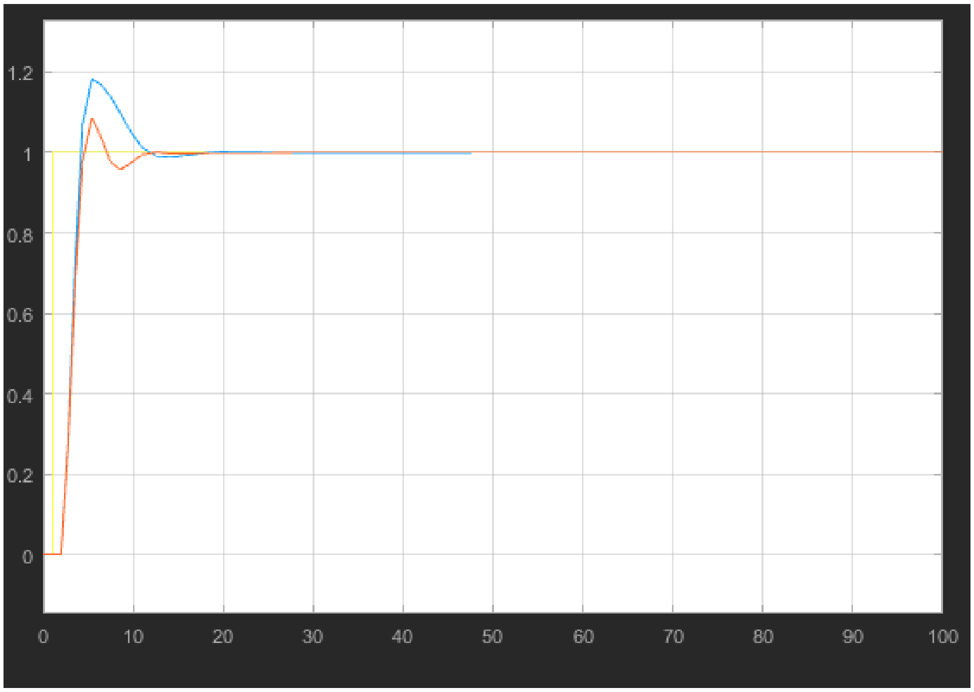


Settings:



Source:	internal
Proportional (P):	0.367046517528655
Integral (I):	0.122067040471561
Derivative (D):	0.0464171649393475
Filter coefficient (N):	1.07672704849522

Output:



3) **Implementation Prowess:** In the implementation phase, I successfully executed controller designs using the Ziegler-Nichols (ZN) method in both closed and open-loop configurations. This hands-on experience allowed for practical tuning and a comprehensive assessment of controller performance.

Conducting the experiment empirically:
In the next section, we select one of the controller design methods from the table and obtain the coefficients according to the table. We have chosen the Ziegler-Nichols open-loop method.

1) Open-loop method

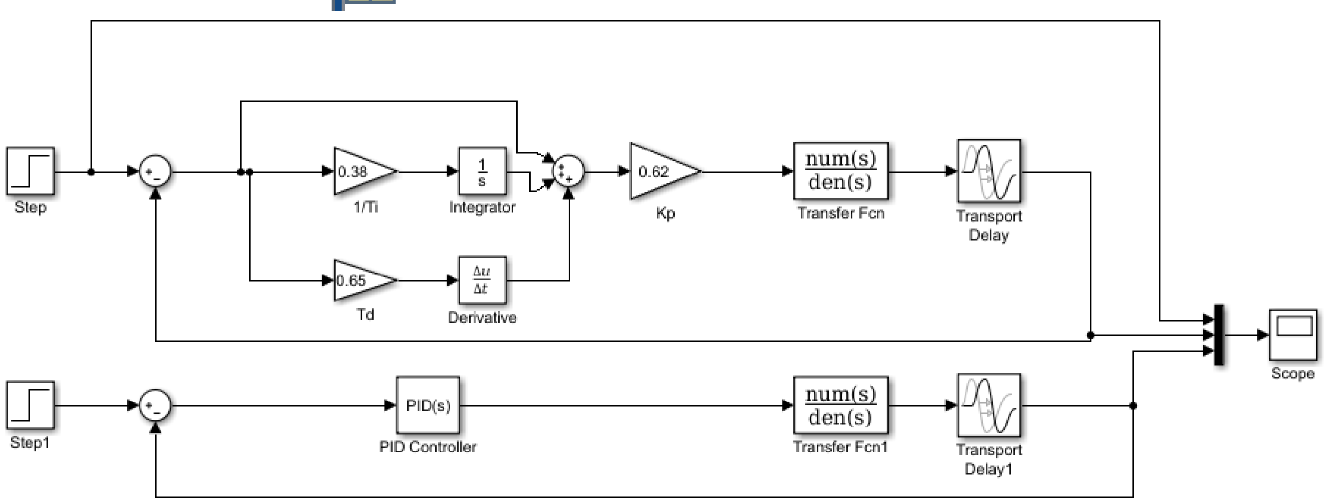
جدول ۵-۳: ضرایب کنترل کننده *PID* غیرتداخلی استاندارد بر اساس روش *Z-N* حلقه باز و مدل مرتبه یک با تاخیر

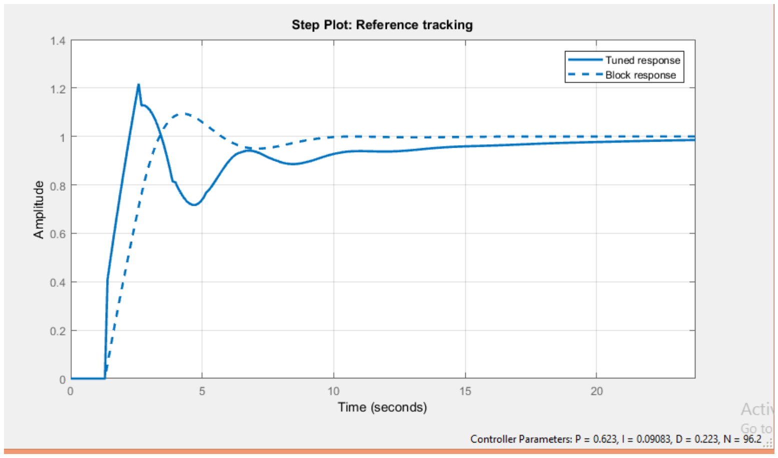
ZNsr	k_p	T_i	T_d
P	$\frac{۱}{k\alpha}$	–	–
PI	$\frac{۰.۹}{k\alpha}$	$۳.۳\tau_d$	–
PID	$\frac{۱.۲}{k\alpha}$	$۲\tau_d$	$۰.۵\tau_d$

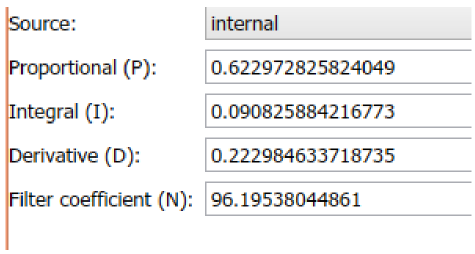
Open-loop ZN PID

```
k=3.7;
tau=2.5;
taud=1.3;
alpha=taud/tau;
Kp=1.2/(k*alpha);
Ti=2*taud;
Td=0.5*taud;
```

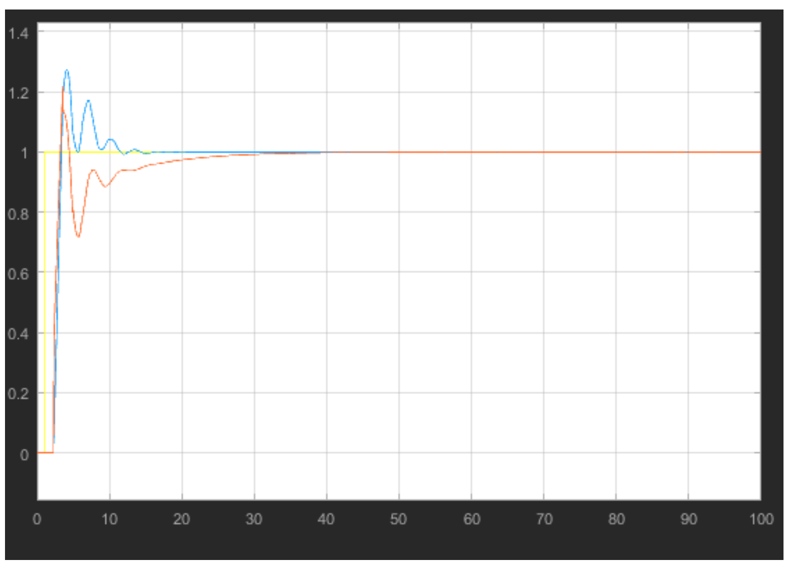
Kp	0.6237
Td	0.6500
Ti	2.6000







Source:	internal
Proportional (P):	0.622972825824049
Integral (I):	0.090825884216773
Derivative (D):	0.222984633718735
Filter coefficient (N):	96.19538044861



2) ZN Closed-loop method

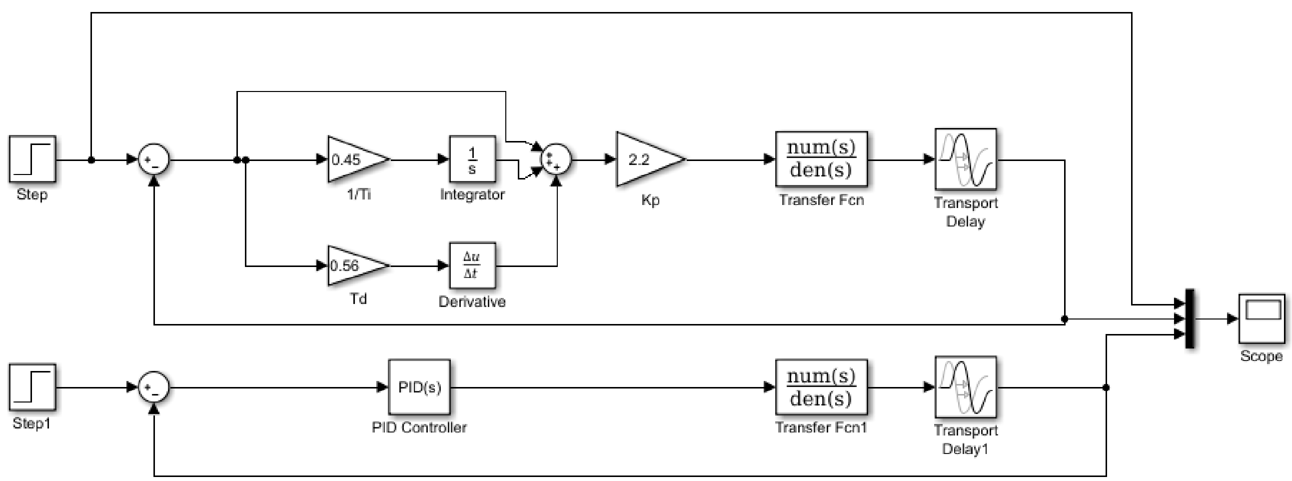
جدول ۵-۲: ضرایب کنترل کننده PID غیرتداخلی استاندارد بر اساس روش $Z\text{-}N$ حلقه بسته

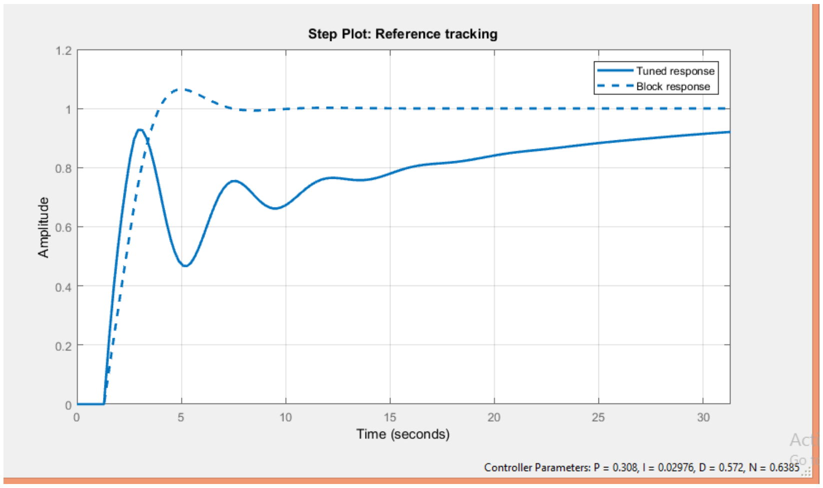
$ZNcl$	k_p	T_i	T_d
P	$۰.۵k_u$	–	–
PI	$۰.۴۵k_u$	$T_u/۱/۲$	–
PID	$۰.۶k_u$	$T_u/۲$	$T_u/۸$

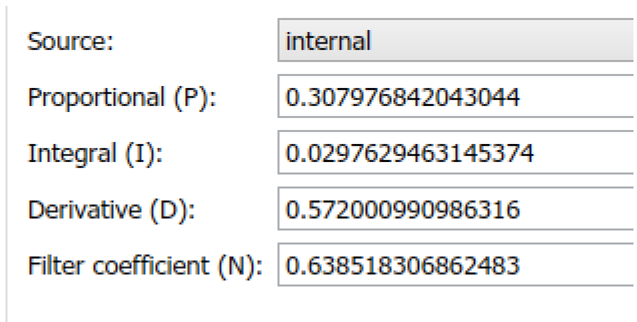
Closed-loop zn

```
Ku=3.7;
Tu=4.5;
Kp=0.6*Ku;
Ti=Tu/2;
Td=Tu/8;
```

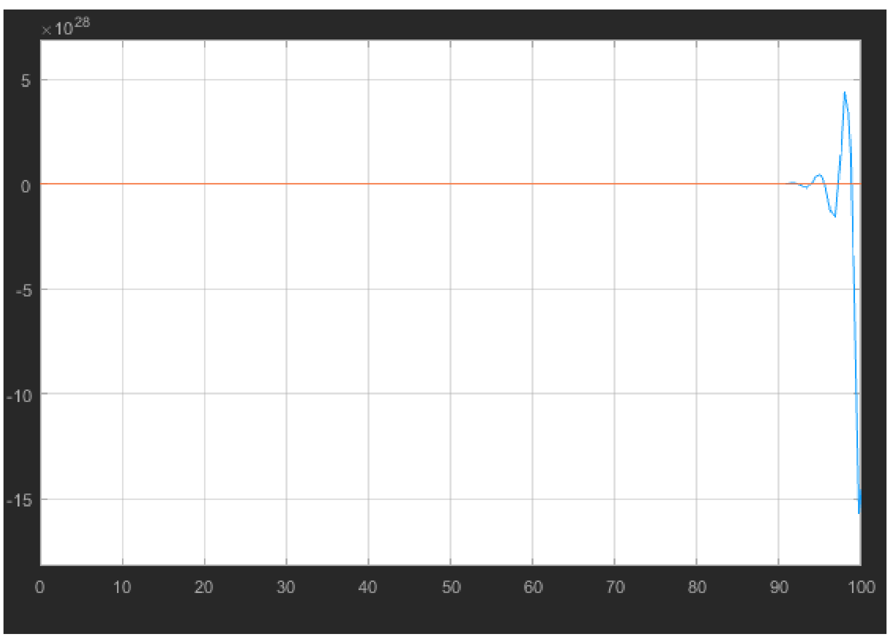
Kp	2.2200
Td	0.5625
Ti	2.2500







Source:	internal
Proportional (P):	0.307976842043044
Integral (I):	0.0297629463145374
Derivative (D):	0.572000990986316
Filter coefficient (N):	0.638518306862483



➔ Based on the results obtained from all controllers designed in all sections with varying GM, PM, and both, it is estimated that the system response for the controller obtained by the Ziegler-Nichols open-loop method provides the best response among all. Therefore, the ZN method, as obtained empirically, is a good and acceptable method for designing system controllers.