Assignment Report

on

“Optimal Nash tuning rules for robust PID controllers”

By

Group No. - 31

Joyoti Disha Rajbongshi (2017A8PS0418P)

Ateeksha Mittal (2017A8PS0431P)

Nagesh Samane (2017A8PS0612P)

Table of contents

1. Introduction:

The control system can be described by the objective of control, control system components and output of the system. Two major components of the control system are the controller and controlled process itself. An input signal or command r, is applied to the controller whose output acts as the actuating signal u which then controls the controlled process so that the controlled variable y will perform according to some predefined way. Through the IIC coursework we have learnt different types of controller and control structure. Amongst all PID controllers are observed to be widely used and accepted in the industry. The PID controllers have been in the industry for a long time and designers also have gained experience in its application specific design. The tuning of parameters are critical as slight change in one of the parameters might bring the entire system towards instability. Therefore to tune PID controllers we have learnt Z-N, Cohen-Coon and Tyres-Luyben tuning during lectures.

The controllers are tuned according to the objective of the control system, which can be broadly classified into two categories: servo action and regulatory action. The paper describes a new method of tuning PID controllers with the aim of having balanced servo and regulatory action. Authors have explored Multi-Objective Optimization Design for tuning the parameters. Further they have used Nash Bargaining technique to choose optimum design vector from a set of acceptable solutions. Through this assignment we understood the tuning method proposed in the paper and attempted to compare this method with other tuning methods (Z-N and Tyres-Luyben) for the systems presented in the paper. The comparative analysis tuning methods, results obtained and our understanding of the paper is presented in this report.

2. Problem formulation:

The figure below shows the general block diagram of a feedback control system with controller (K), process (P), reference signal (r), error signal (e), disturbance (d) and output (y). Author has modelled process P as a first order plus dead time (FOPDT) system and controller K is assumed to be one degree of freedom PID with a derivative filter. The presented system can work in two different modes: servo operation mode (for tracking of r) and regulation operation mode (for rejection of d) and controller K has to support the process to achieve this operation.

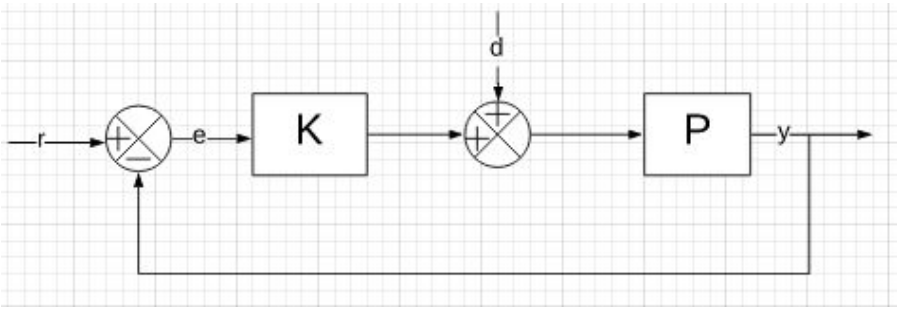


Figure 1. General block diagram of feedback control system

The objective of the paper is to tune the PID controller in such a way that it will support both the operation modes with maintaining desired level robustness. To quantify the performance in servo action and regulatory action Integral Squared Error (ISEs) is considered and robustness measured by the **Sensitivity.** The tuning problem can be visualized as the optimization process having two objectives (to minimize ISEs for both actions) with one constraint (to maintain sensitivity in desired range). This is the typical case of the Multi-Objective Optimization Design problem.

2.1. Formulating control problem as the Multi-Objective Optimization problem:

Multi-Objective Optimization Design can be modularized into 3 major steps.

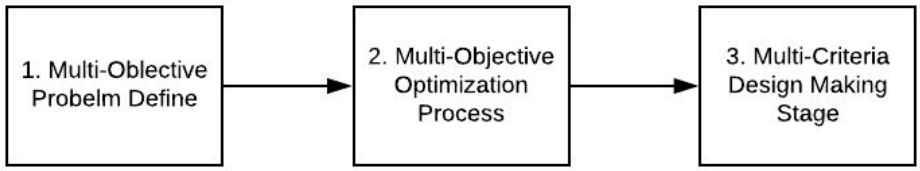


Figure 2. Flow chart of Multi-Objective Optimization design Procedure

2.1.1. Multi-Objective Problem Definition

We define the target vector as [kp, Ti, Td] tuning parameters of the PID controller. The two objective (ISE) functions are then defined in terms of target vector. The constraint of robustness is also defined in terms of the target vector.

2.1.2. Multi-Objective Optimization Process

The author has employed Normalized Normal Constraint (NNC) algorithm for the given bi-objective problem to find Pareto front. After the optimization process we obtain a set of solutions on the Pareto front which satisfies our objectives and constraints. To pick optimum solution from the Pareto front final step of Multi-Criteria Design Making is carried out.

2.1.3. Multi-Criteria Design Making stage

At this stage the author states two ways to get optimum solution from the obtained Pareto front. First method is to run an optimization process with additional criteria which is of secondary importance. In the second method we choose a fair point which balances two cost functions. Author has used a second approach with **Nash bargaining solution** to get optimum tuning parameters.

3. Design Approach:

The design of a robust PID controller is summarized in the following flowchart.

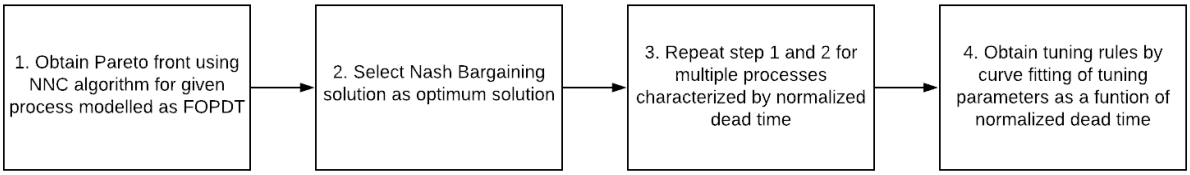


Figure 3.1. Flow chart of PID controller design steps

By applying NNC algorithm for bi-obejctive problem with one constraint, we get set of solutions satisfying our objectives as well as staying within the constraints. Such a set of solutions is denoted by pareto front. Here all the solutions are acceptable but they differ in degree of performance as measured in terms of cost functions.

A typical pareto front is shown in figure 3 of the paper, in which the x and y axis are the cost function values as a function of design vector. The important points on the figure are Utopia point (U), Disagreement point (D), Compromise Solution (CS), Nash Solutions (NS). U is the point where both ISEs (cost functions) attain their individual minimum which is not a feasible solution. It is not possible to minimize both the cost functions simultaneously. Hence the first approach to get a feasible solution is to locate a point on the pareto front at minimum distance from Utopia point (U), called as Compromise Solution (CS). But this method is not practical as we start from a non-attainable solution (U). Second approach is to traverse on the pareto front and find a point which maximizes the area of the rectangle formed by sides as “profit” (longitudinal and transverse distance from point D). In this way we try to minimize the product of cost associated with both ISEs. Author has used a second practical approach in the paper calling it a “Nash Bargaining Solution”.

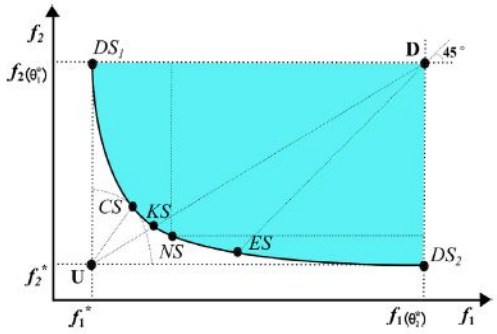


Figure 3.2. - Location of the different solutions on the Pareto front. (***Figure-3 : directly used from the paper***)

Step 1 and 2 are repeated for multiple processes modelled as a FOPDT system characterized by normalized dead time ($\boldsymbol{\tau}$).

We have gathered data points i.e. design vector [kp, Ti, Td] values as a function of normalized dead time ($\boldsymbol{\tau}$) from step c, plotted in figure 6 of the paper. The different lines represent the Pareto front for processes from $\boldsymbol{\tau}$ value 0.1 to 2. The red dot on the figure corresponds to Nash Bargaining Solution which is optimized in ISE plane and lies within robustness requirements. Figure 8 from the paper shows curve fitting process for obtained data set and tuning equations are summarized in equation 17, 18, 19 along with parameters for tuning in table 2.

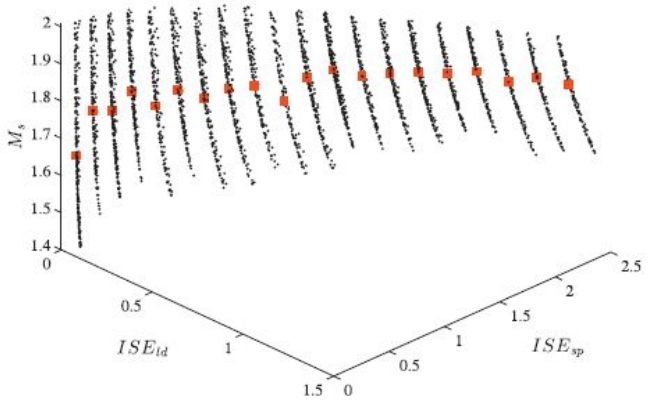


Figure 3.3. -Pareto fronts for the robust case for different normalized dead times and the corresponding Nash solutions (red square) **(*Figure-6 : directly used from the paper*)**

From these 4 steps PID controllers can be tuned for robust behavior balancing servo and regulatory operation.

4. Implementation:

As stated in the paper, we have made an attempt to compare the different tuning methods like Nash Tuning, Zieglar-Nichols and Tyres-Luyben. Though these methods are related to different controller structures, we have mainly experimented with a PID controller and applied it to various process models and for different control tasks that they address.

For Nash tuning the values in Table 3, 4 and 5 (Kp, Ti, Td) were used to calculate the various parameters of the PID controllers for each of the three examples given.
For Zieglar-Nichols and Tyres-Luyben the parameters were found out from the characteristic equations of the three examples respectively (This has been explained in the later portion of the report).

The circuit diagrams were appropriately designed on simulink and the PID was updated according to the parameters as discussed above. Graphs were then plotted and compared with those given in the paper for both set point and load disturbance step response to observe the differences and similarities between the three tuning methods.
The results obtained from each of the tuning methods used are compared on the basis of ISE and relevant inferences are drawn accordingly.

5. Results and Discussion:

We have plotted the set point tracking and disturbance reduction graphs for the processes (P1, P2, P3) mentioned in the paper.

5.1. Nash Tuning:

5.1.1 Process 1:

$$P_1(s) = \frac{1}{10s+1}e^{-4s}$$

Nash Solution: Kp = 1.88, Ti =6.60, Td = 1.97

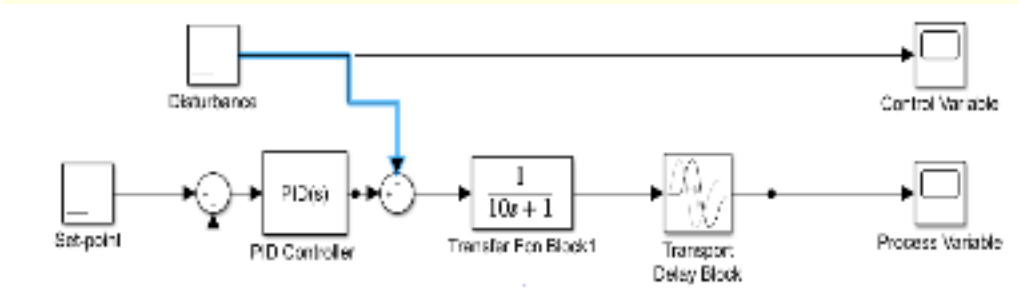


Figure 4. Simulink Block diagram for set-point and load disturbance step response for Process P1.

Set-point Step Responses for the process P1

a. Control Variable:

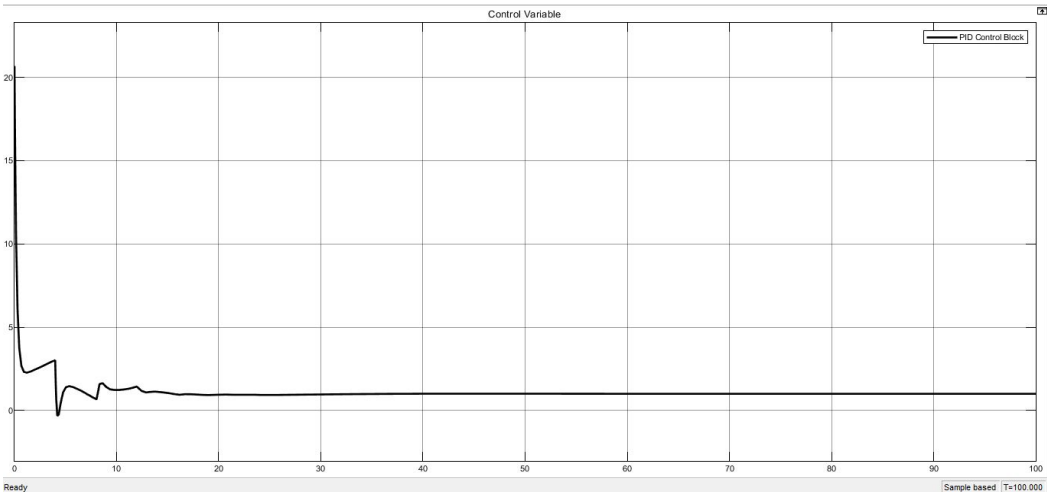


Figure 5. Set-point step response for Process P1- Control Variable

b. Process Variable:

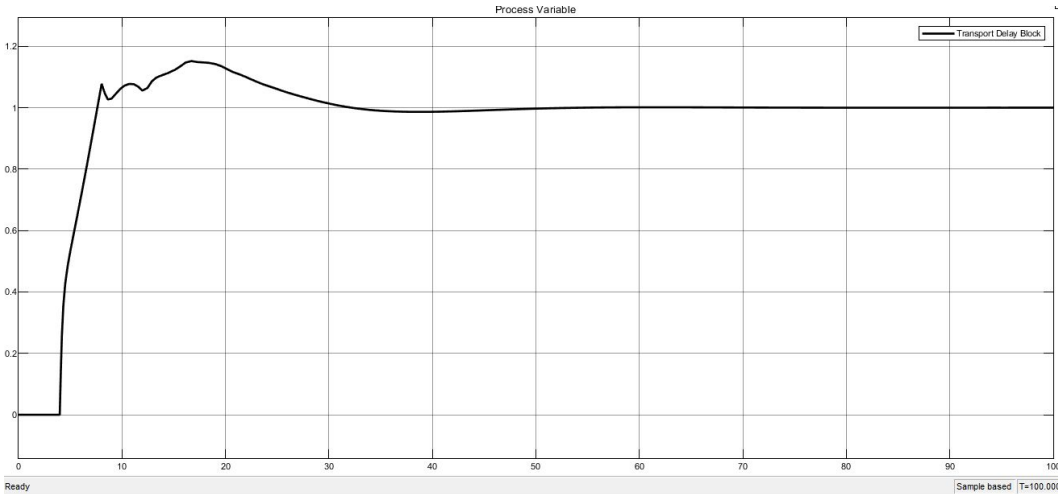


Figure 6. Set-point step response for Process P1- ProcessVariable

Load Disturbance Step Response for the process P1

a. Control Variable:

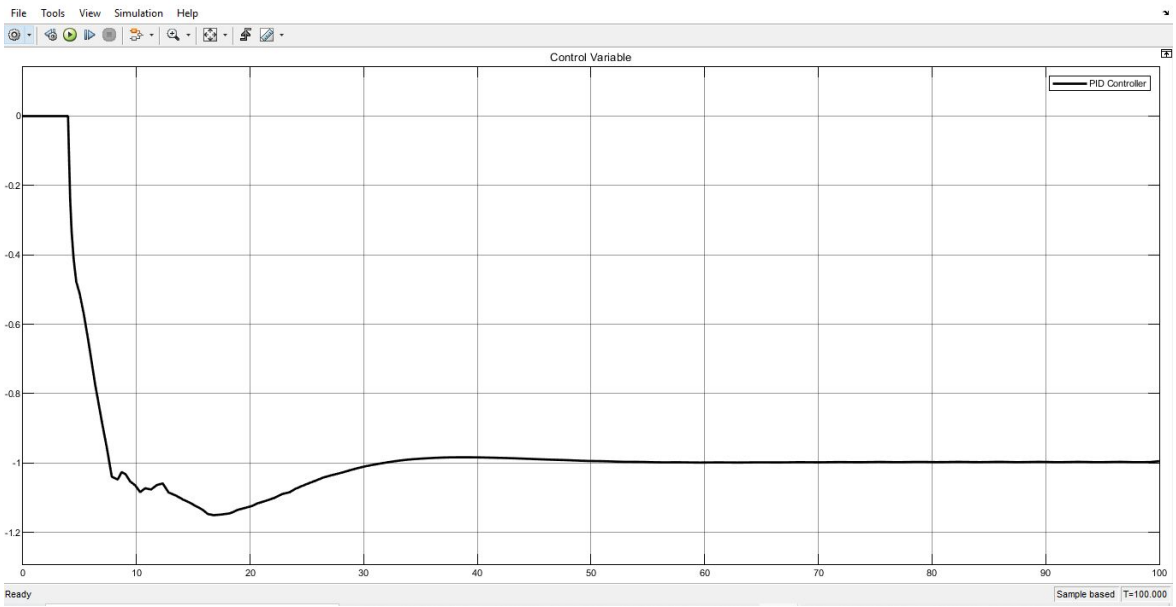


Figure 7. Load Disturbance step response for Process P1- Control Variable

b. Process Variable

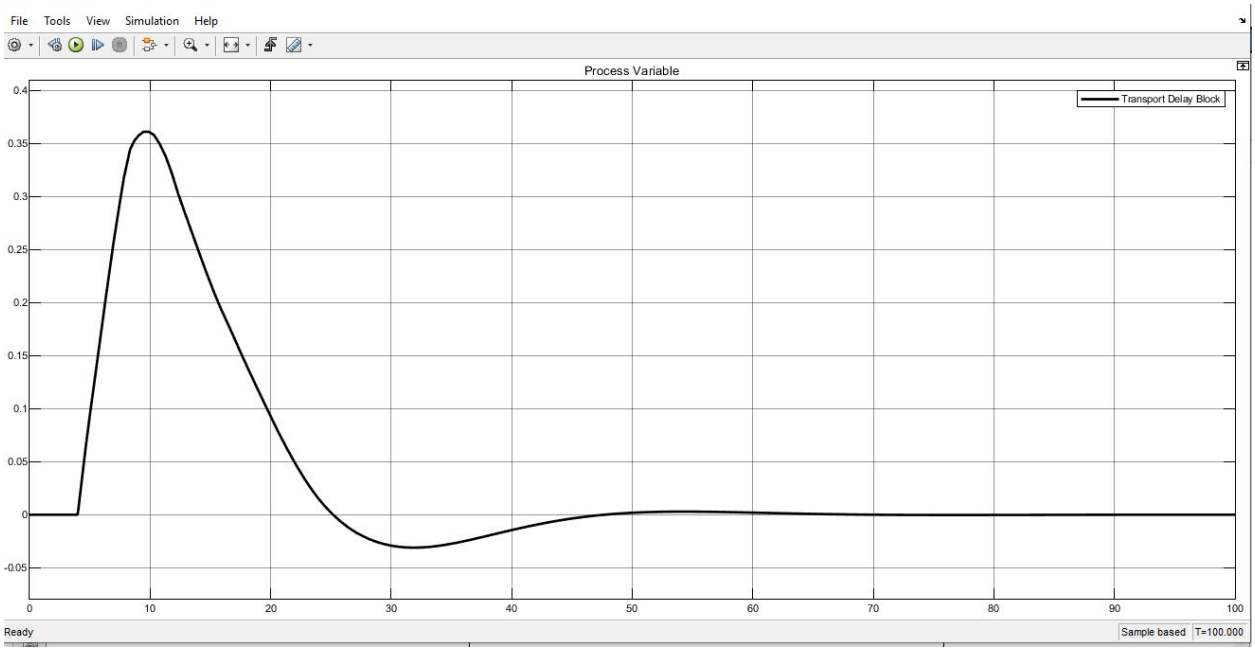
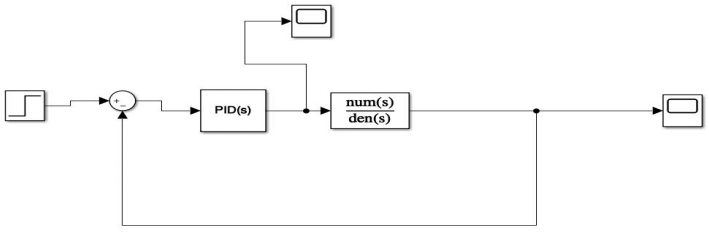


Figure 8. Load Disturbance step response for Process P1- ProcessVariable

5.1.2. Process 2:

$$P_2(s) = \frac{1}{(s+1)^4}$$

Nash Solution:- Kp = 1.60, Ti =2.060, Td = 0.69



num(s) = 1, den(s) = s4 + 4s3 + 6s2 + 4s + 1

Figure 9. Simulink Block diagram for set-point and load disturbance step response for Process P2

Set-point Step Responses for the process P1

a. Control Variable:

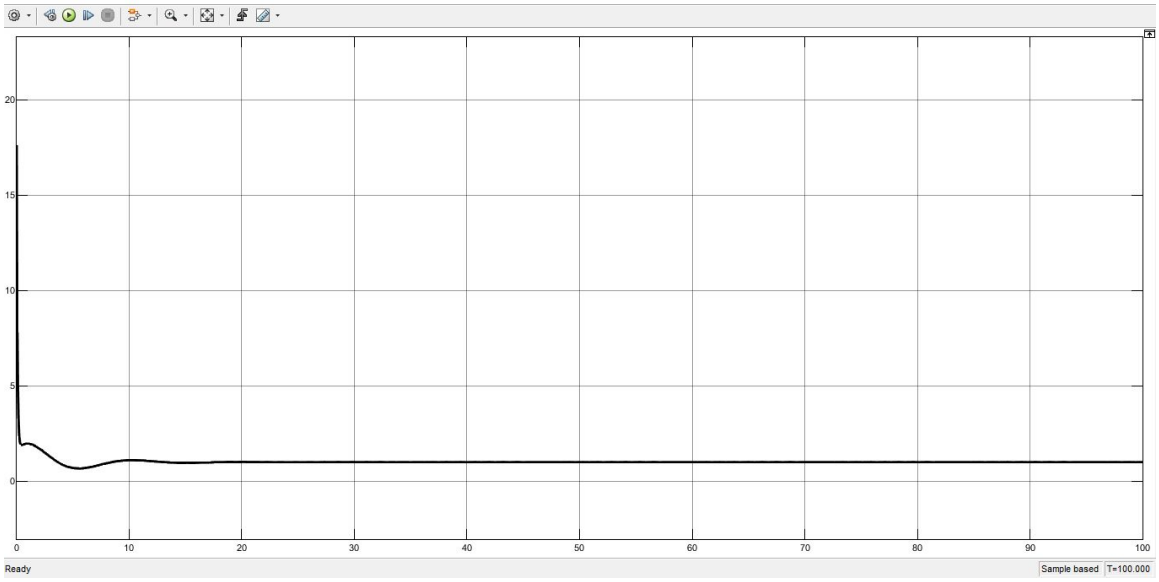


Figure 10.: Set-point step response for Process P2- Control Variable

b. Process Variable:

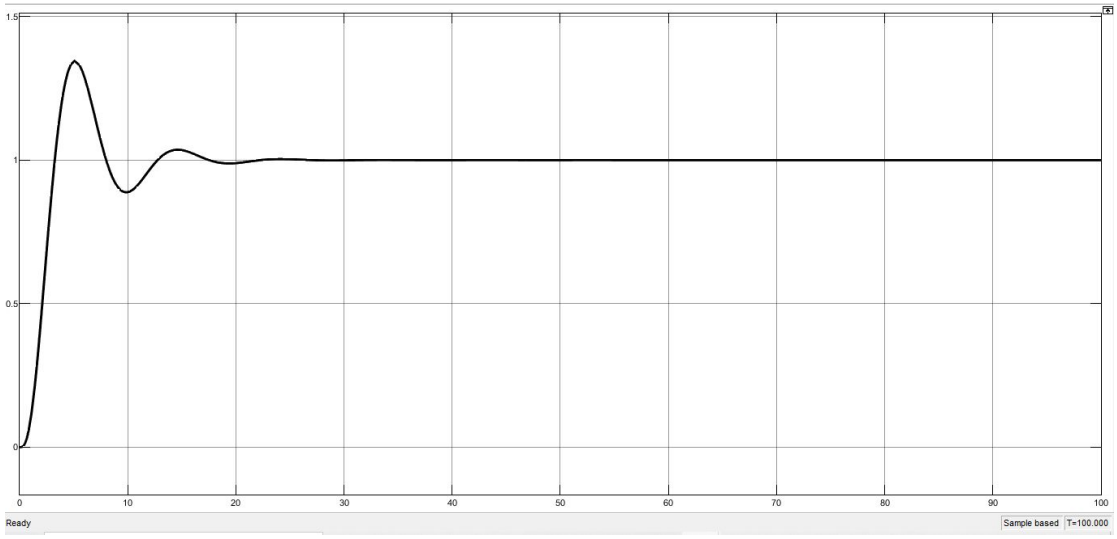


Figure 11. Set-point step response for Process P2- ProcessVariable

Load Disturbance Step Response for the process P2

a. Control Variable:

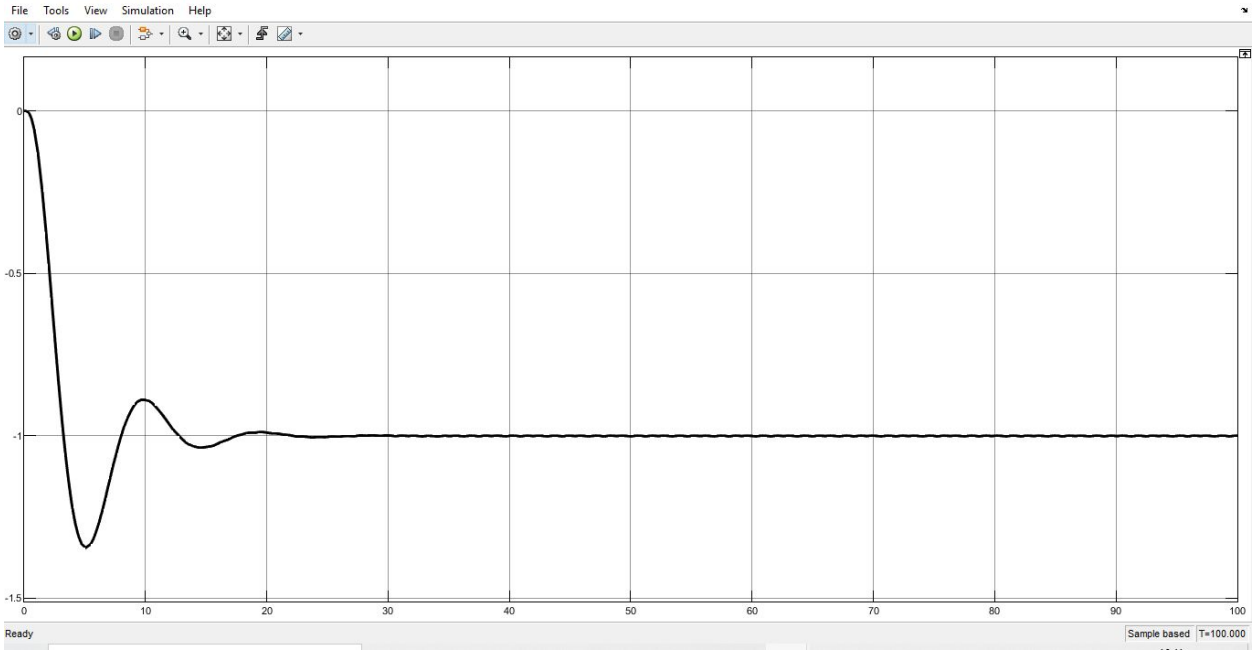


Figure 12. Load Disturbance step response for Process P2- Control Variable

b. Process Variable

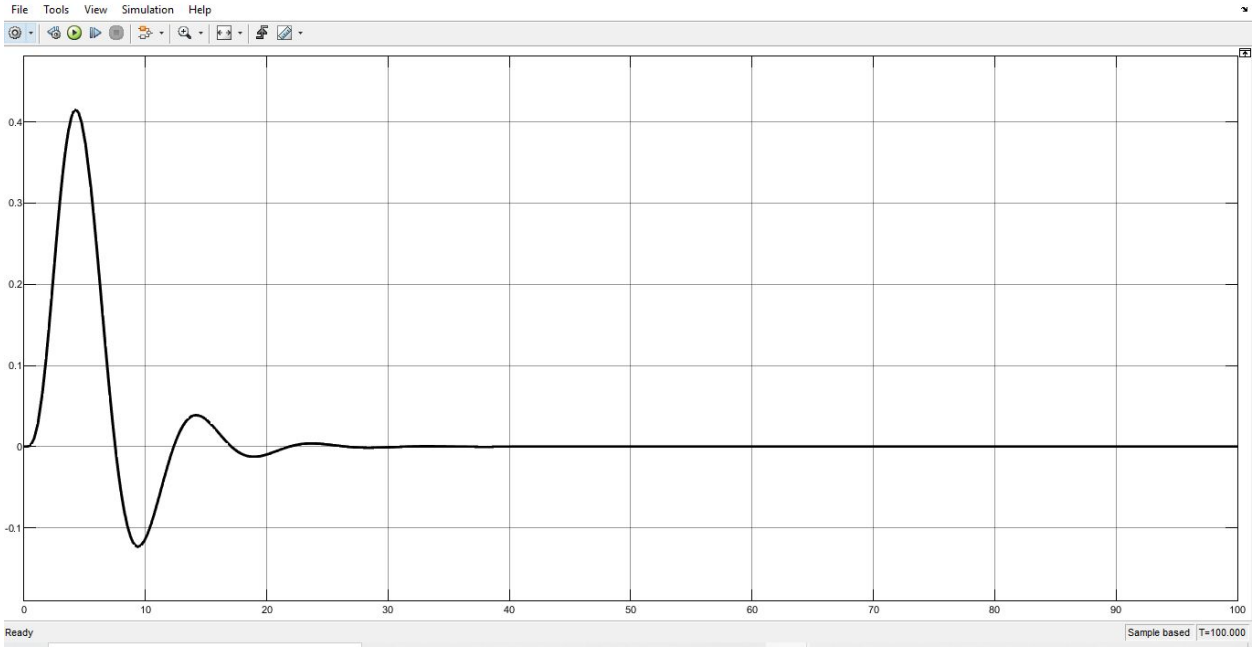


Figure 13. Load Disturbance step response for Process P2- ProcessVariable

5.1.3. Process 3:

$$P_3(s) = \frac{1}{(s+1)^{20}}$$

Nash Solution:-Kp = 0.76, Ti =10.35, Td = 4.69

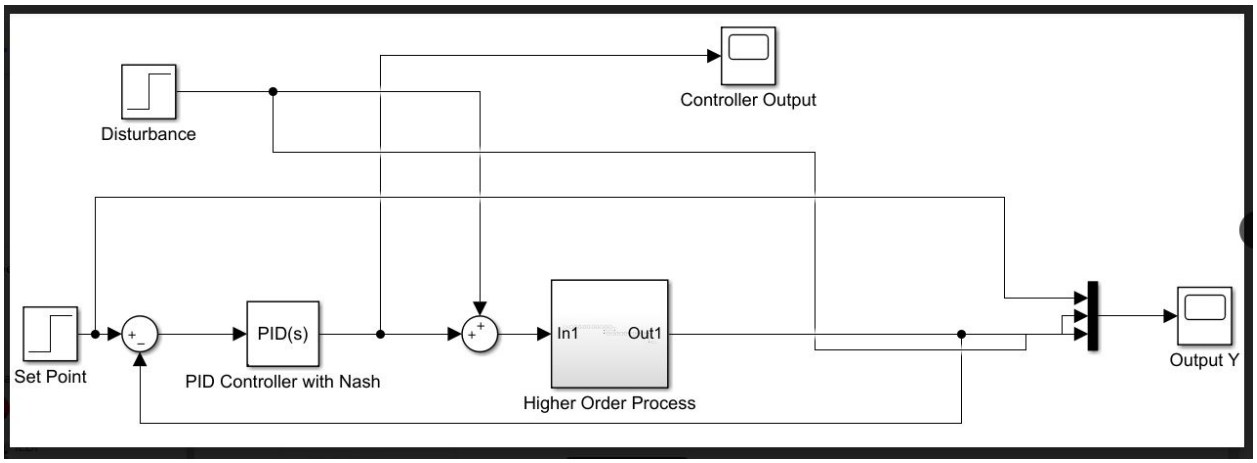


Figure 14. Simulink Block diagram for set-point and load disturbance step response for Process P3

Set-point Step Responses for the process P3

a. Control Variable:

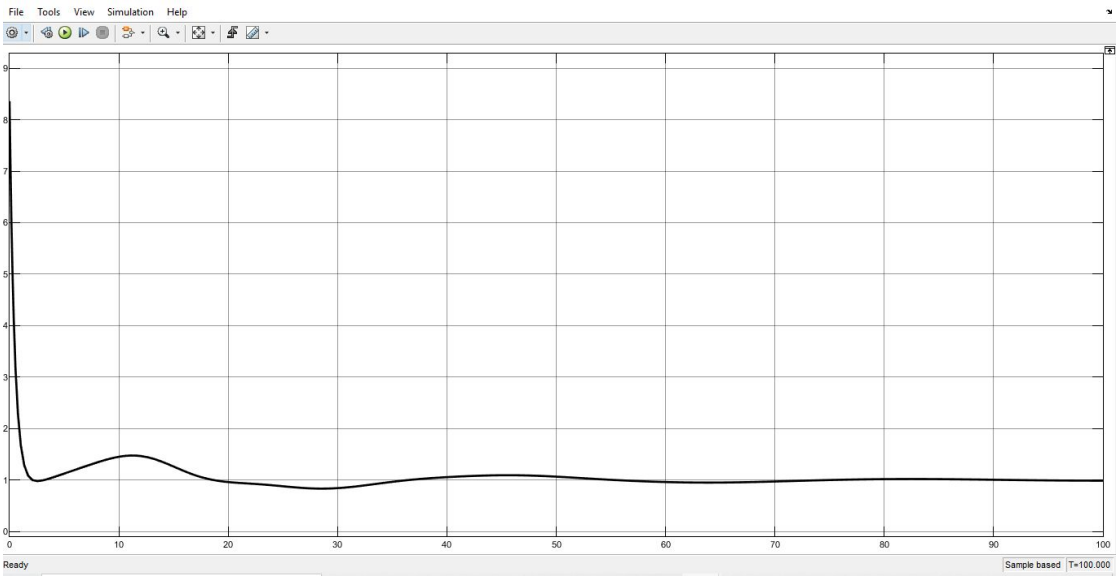


Figure 15. Set-point step response for Process P3- Control Variable

b. Process Variable

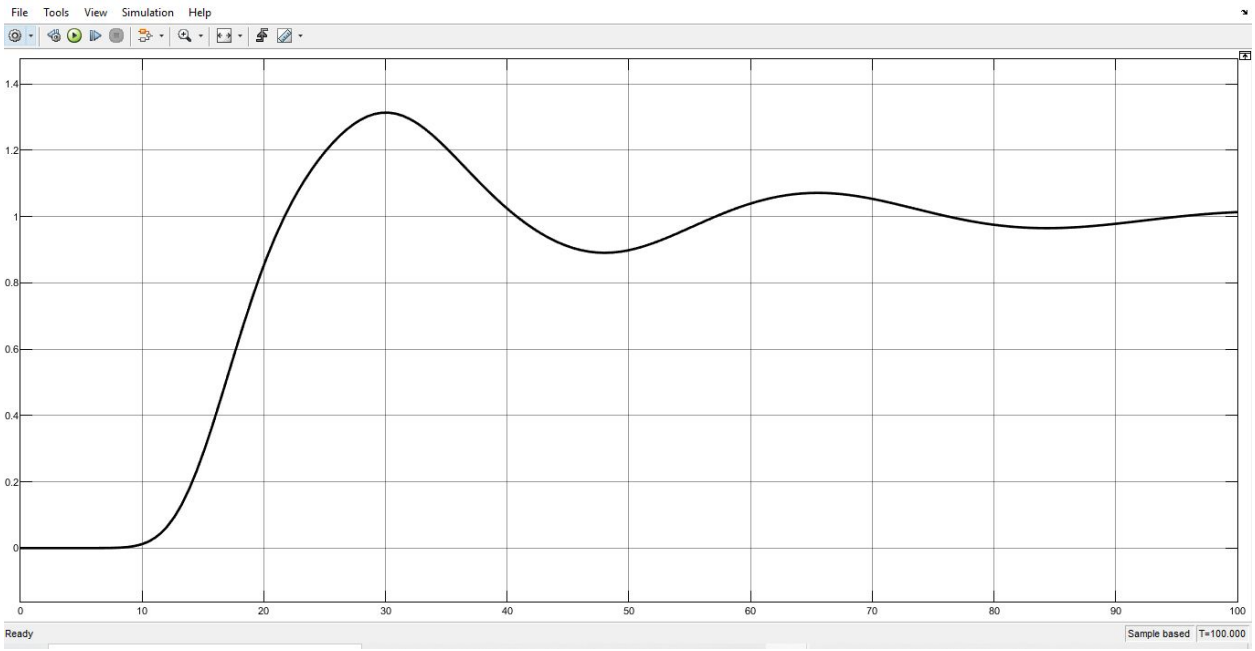


Figure 16. Set-point step response for Process P3- ProcessVariable

Load Disturbance Step Response for the process P3

a. Control Variable:

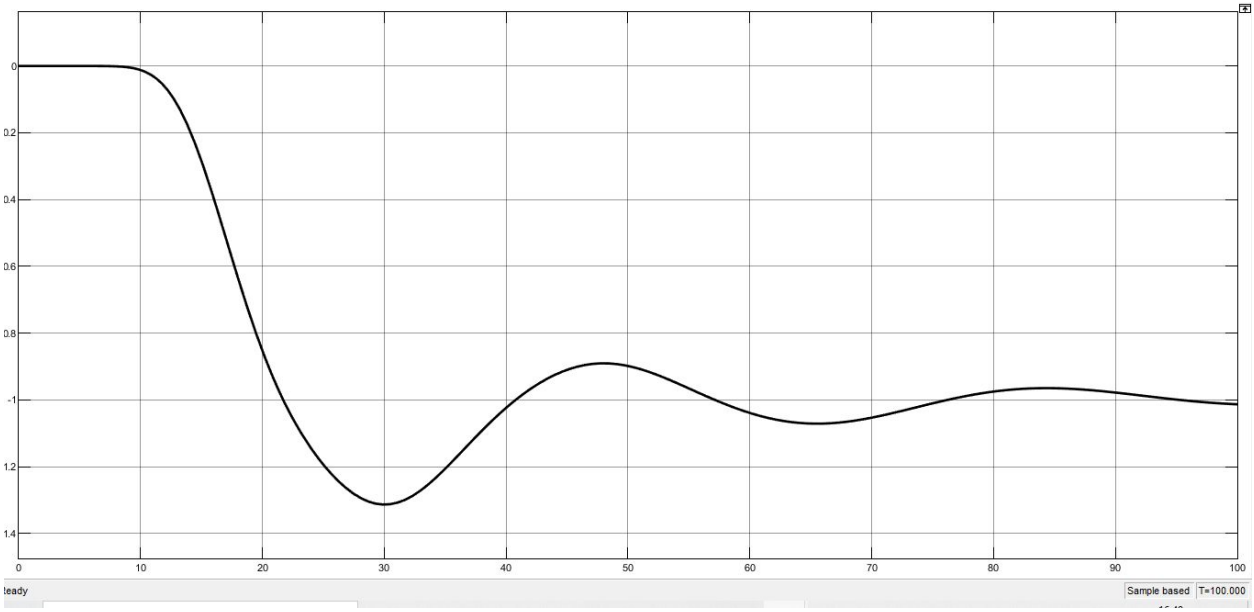


Figure 17. Load Disturbance step response for Process P3- Control Variable

b. Process Variable

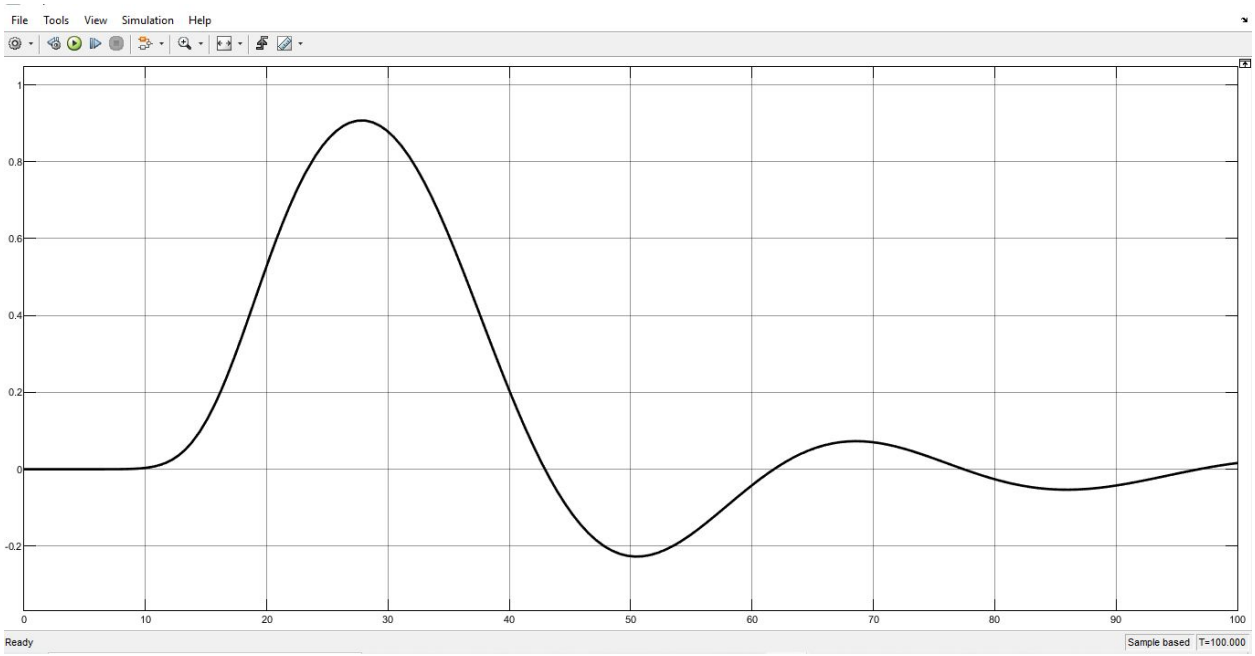


Figure 18. Load Disturbance step response for Process P3- ProcessVariable

5.2. Calculation of PID parameters for ZN tuning and Tyres-Luyben tuning:

Firstly the characteristic equations were found for every example and accordingly the Ku was found out (using Routh's Stability Criterion).

Controller type	Kc	T_I(min)	T_D(min)
P-only	$K_u/2$	-	-
PI Controller	$K_u/2.2$	$P_u/1.2$	-
PID Controller	$K_u/1.7$	$P_u/2$	$P_u/8$

This table was used to find the constants using Zieglar-Nicoles Tuning.

Controller type	Kc	T_I(min)	T_I(min)
PI Controller	$K_u/3.2$	$2.2P_u$	-
PID Controller	$K_u/2.2$	$2.2P_u$	$P_u/6.3$

This table was used to find the constants using Tyres-Luyben Tuning

5.3. Ziegler-Nichols Tuning:

5.3.1. Process-1:

$$P_1(s) = \frac{1}{10s+1}e^{-4s}$$

ZN tuning Parameters Calculated: P = 2.676 , I = 0.449, D = 3.99, N = 6.711

Set-point Step Responses for the process P1

a. Control Variable:-

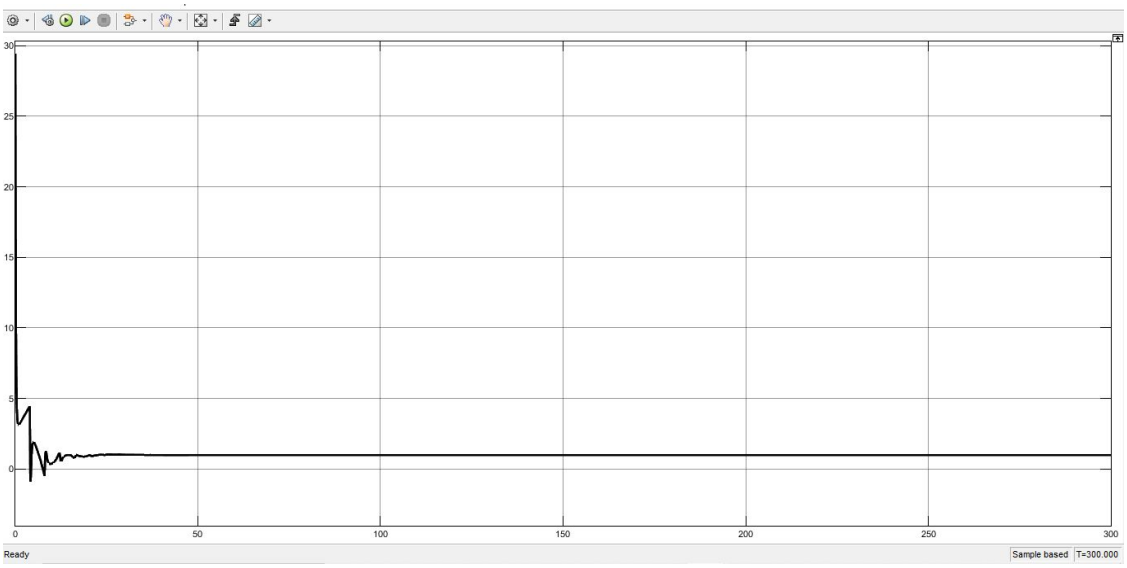


Figure 19. Set-point step response for Process P1- Control Variable

b. Process Variable

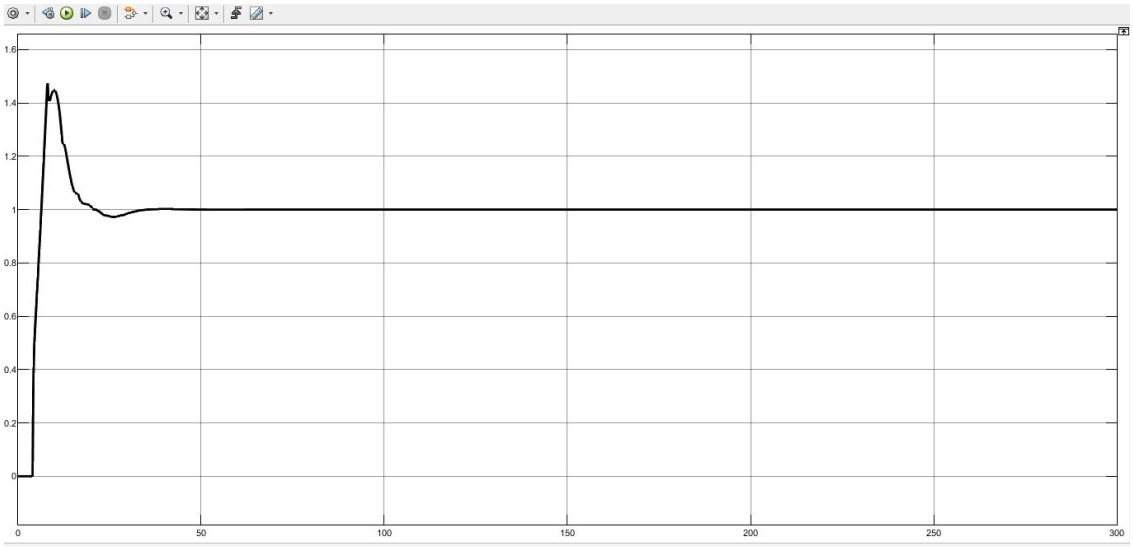


Figure 20. Set-point step response for Process P1- ProcessVariable

Load Disturbance Step Response for the process P1

a. Control Variable:

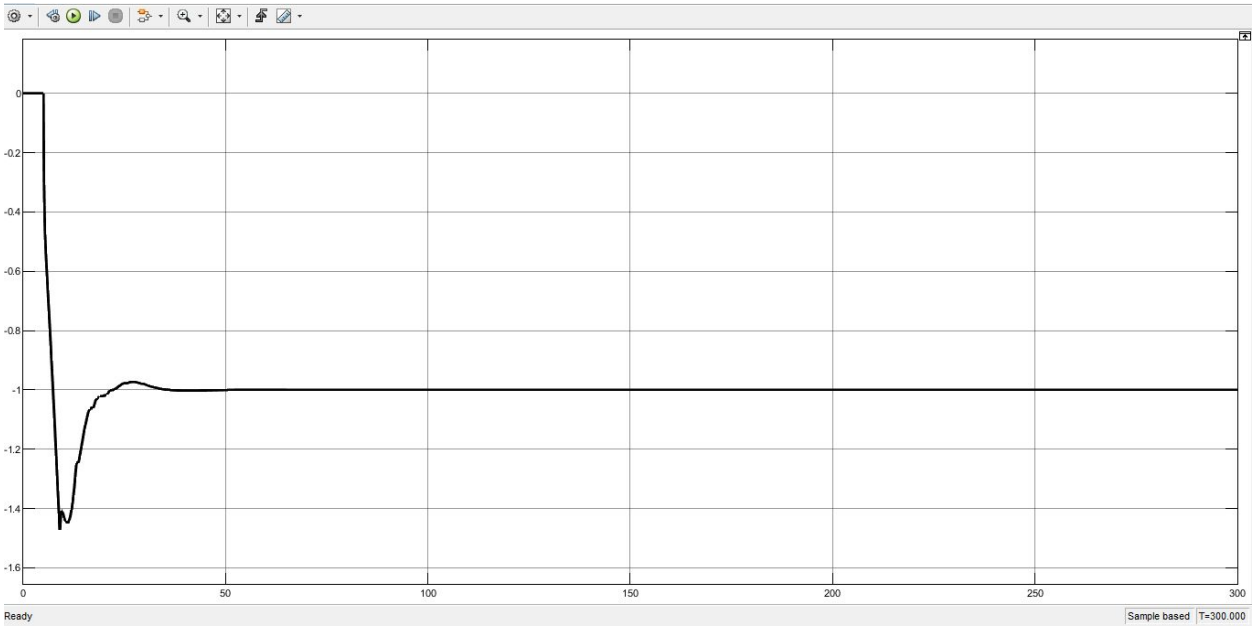


Figure 21. Load Disturbance step response for Process P1- Control Variable

b. Process Variable

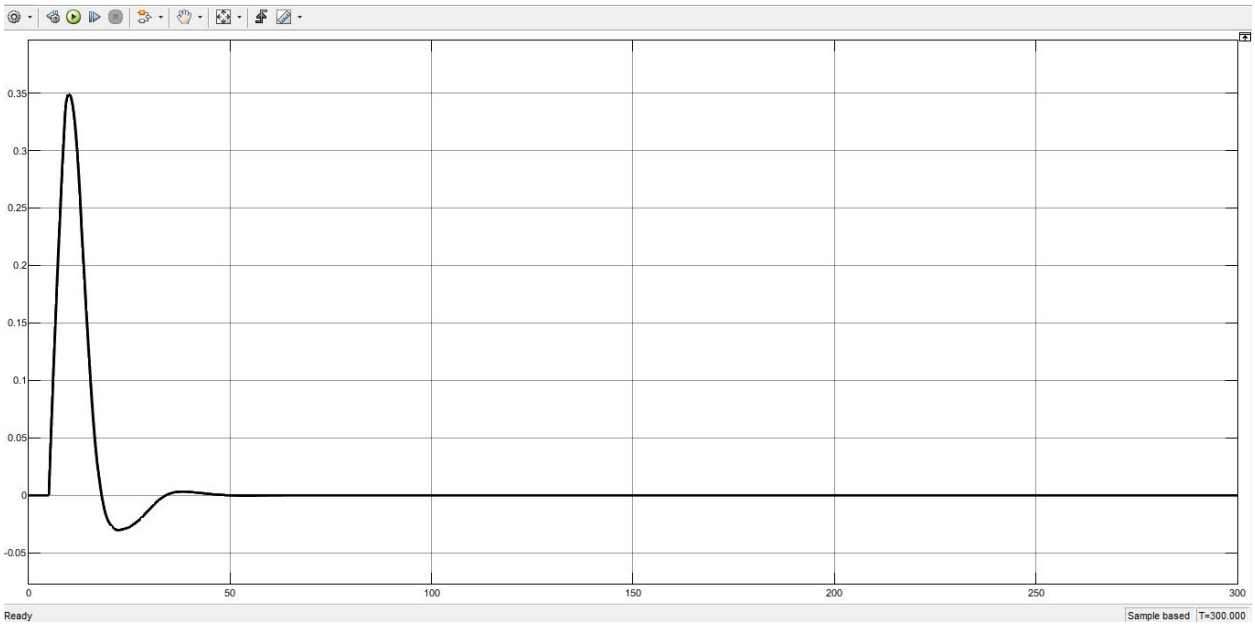


Figure 22. Load Disturbance step response for Process P1- ProcessVariable

5.3.2. Process-2:

$$P_2(s) = \frac{1}{(s+1)^4}$$

Parameters Calculated: PID: P = 2.353 , I = 0.75, D = 1.847, N = 12.74

Set-point Step Responses for the process P2

a. Control Variable:-

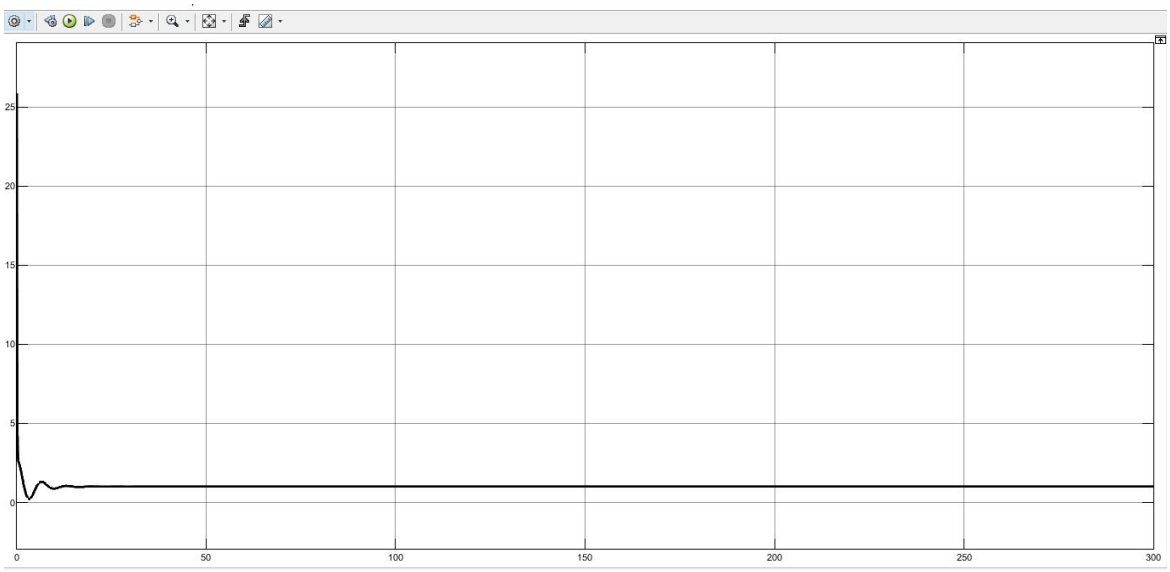


Figure 23. Set-point step response for Process P2- Control Variable

b. Process Variable

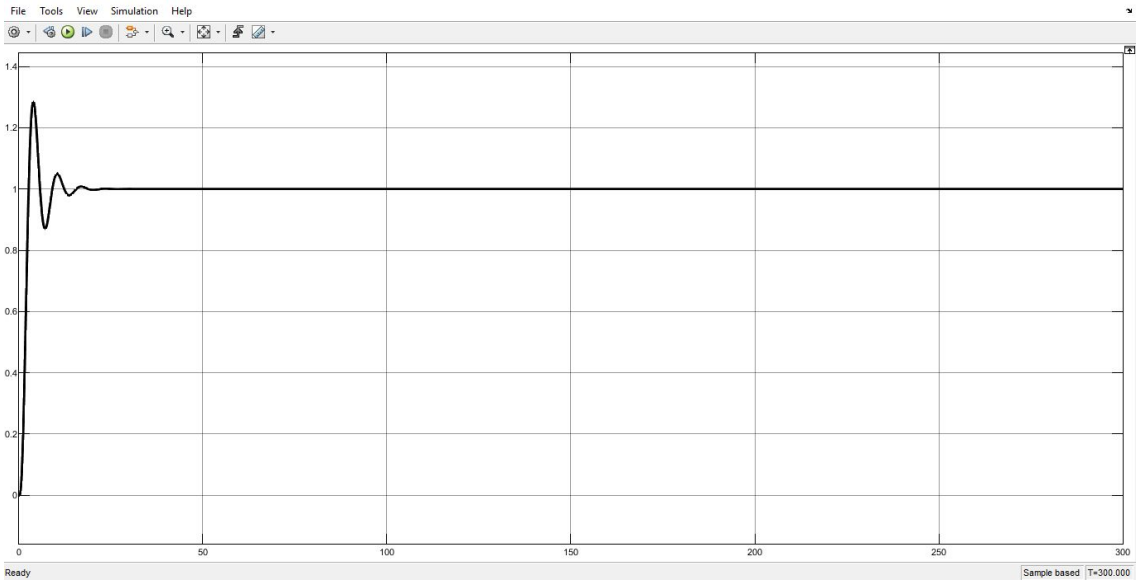


Figure 24. Set-point step response for Process P2- ProcessVariable

Load Disturbance Step Response for the process P2

a. Control Variable:-

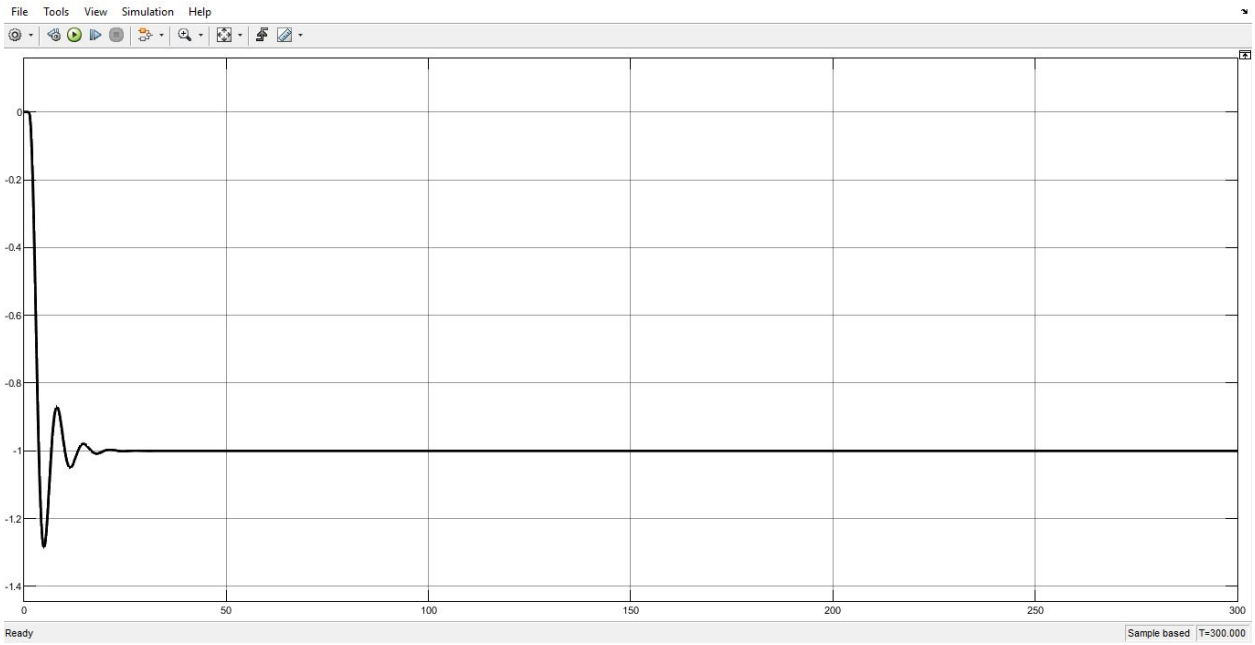


Figure 25. Load Disturbance step response for Process P2- Control Variable

b. Process Variable

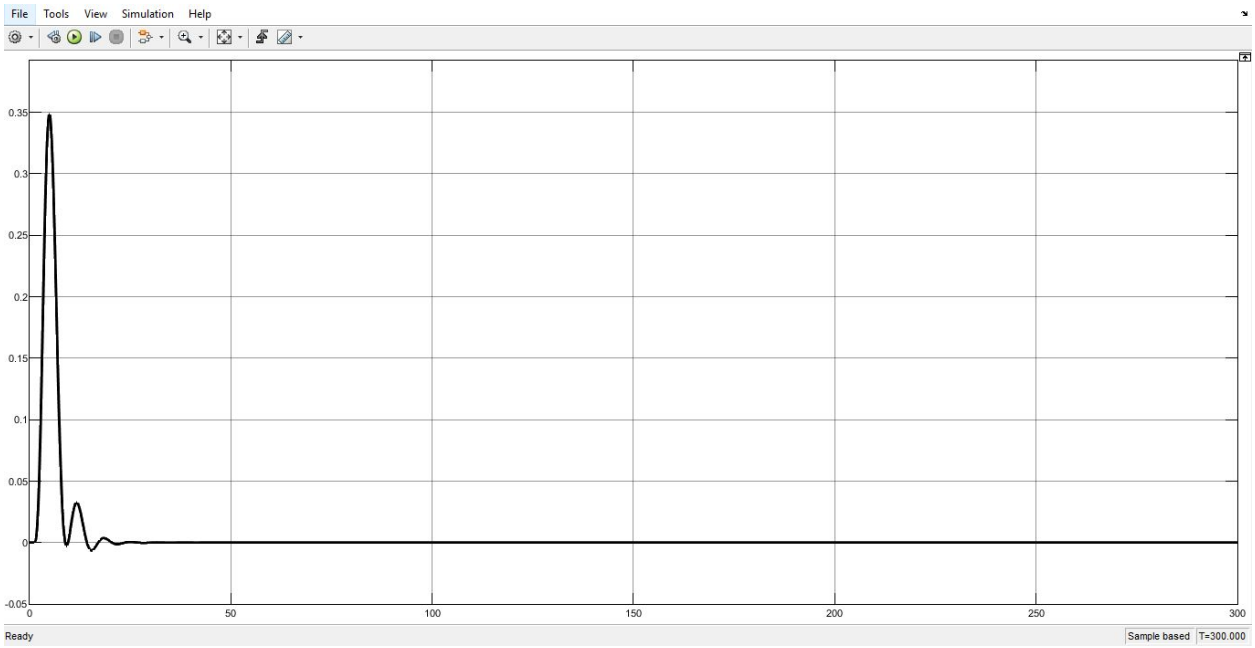


Figure 26. Load Disturbance step response for Process P2- ProcessVariable

5.3.3. Process-3

$$P_3(s) = \frac{1}{(s+1)^{20}}$$

Parameters Calculated: PID: P = 0.75 , I = 0.0382, D = 3.6825, N = 2.03666

Set-point Step Responses for the process P3

a. Control Variable:-

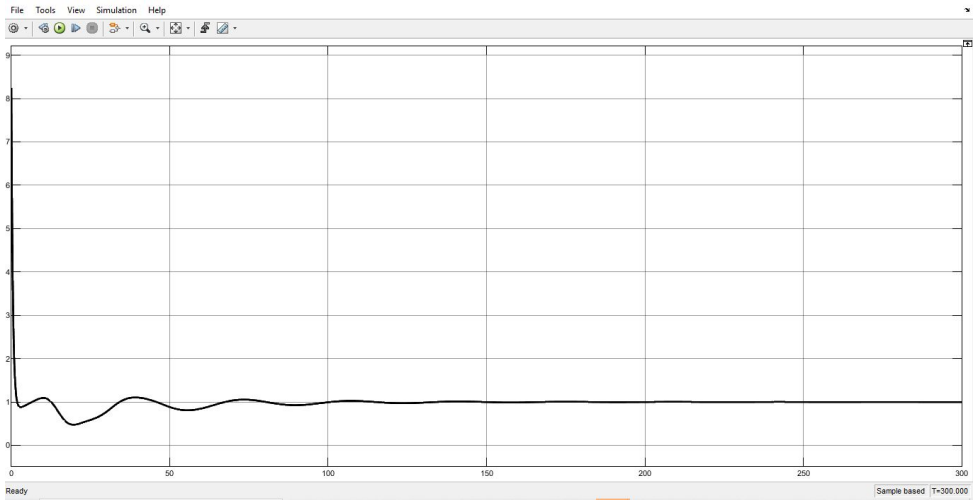


Figure 27. Set-point step response for Process P3- Control Variable

b. Process Variable

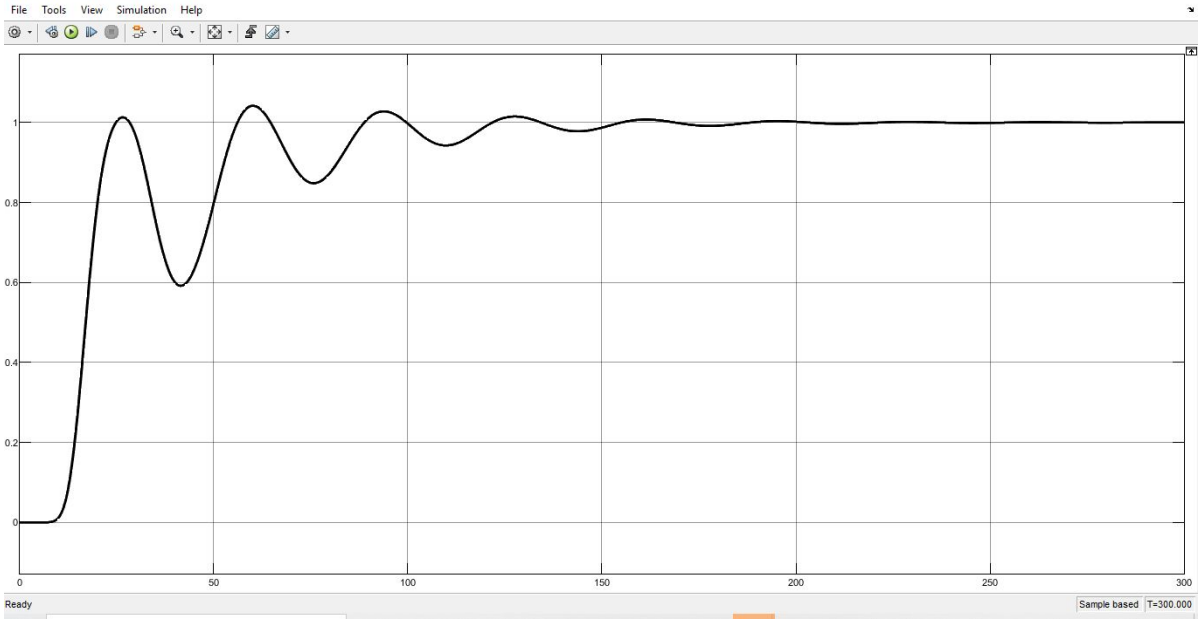


Figure 28. Set-point step response for Process P3- ProcessVariable

Load Disturbance Step Response for the process P3

a. Control Variable:-

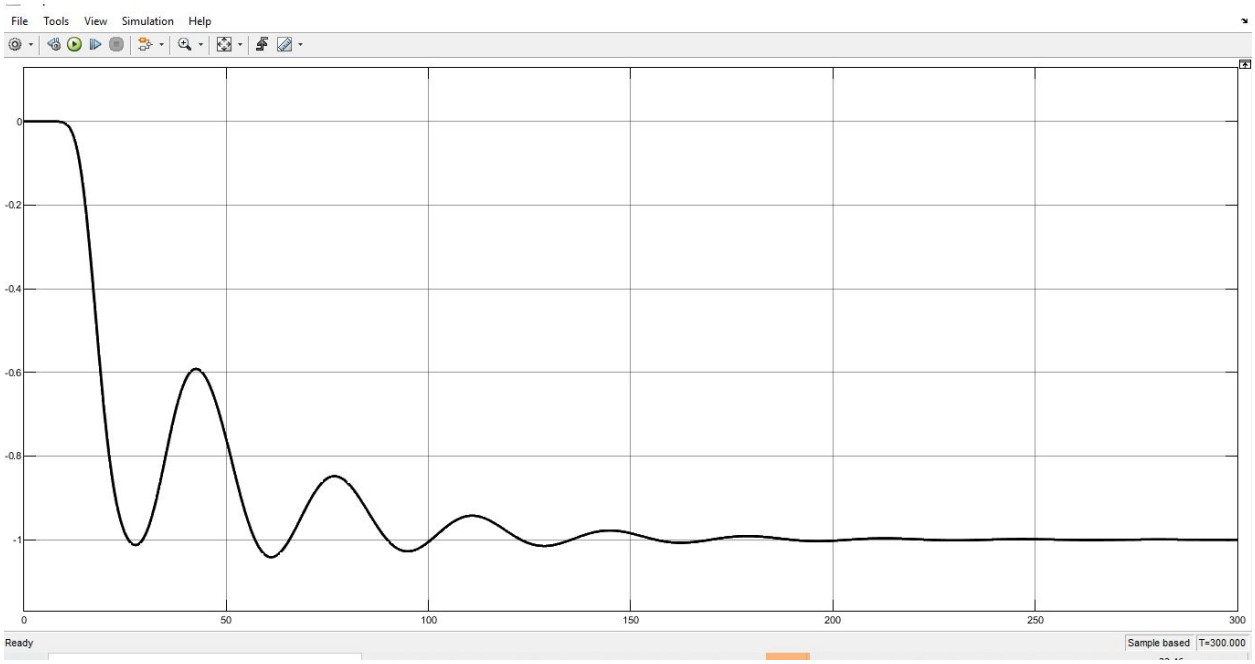


Figure 29. Load Disturbance step response for Process P3- Control Variable

b. Process Variable

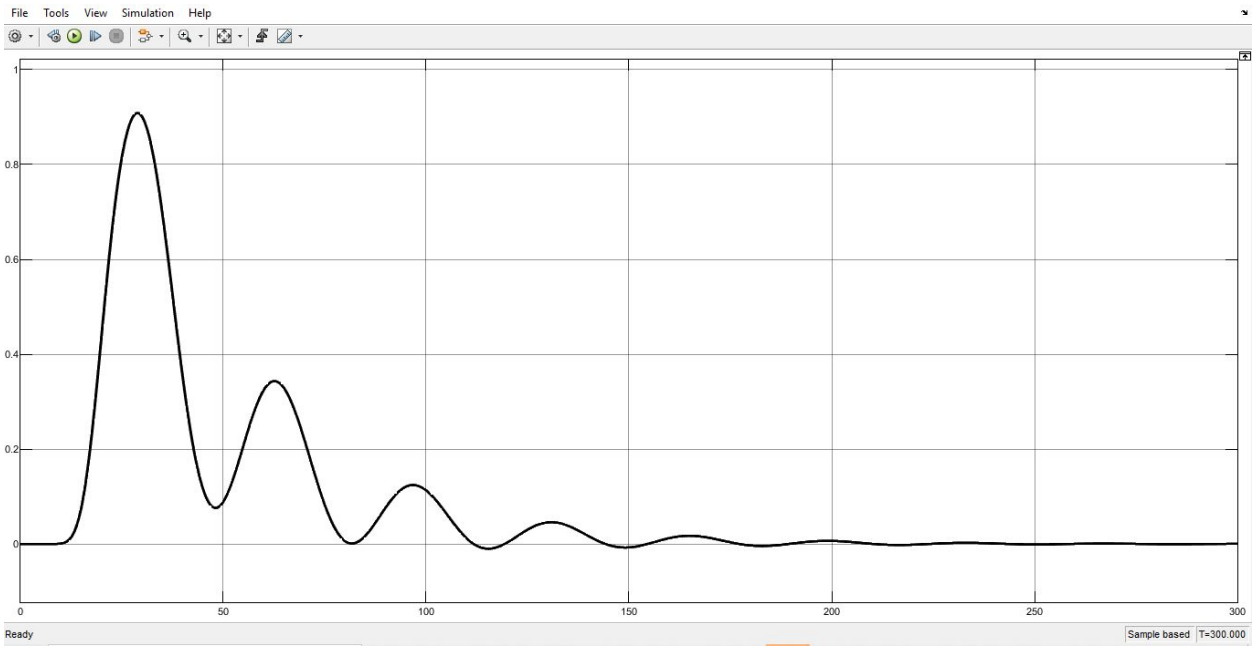


Figure 30. Load Disturbance step response for Process P3- ProcessVariable

5.4. Tyreus-Luyben Tuning:

5.4.1. Process-1:

$$P_1(s) = \frac{1}{10s+1}e^{-4s}$$

Parameters Calculated:-PID: P = 2.068 , I = 0.0789, D = 3.68, N = 5.618

Set-point Step Responses for the process P1

a. Control Variable

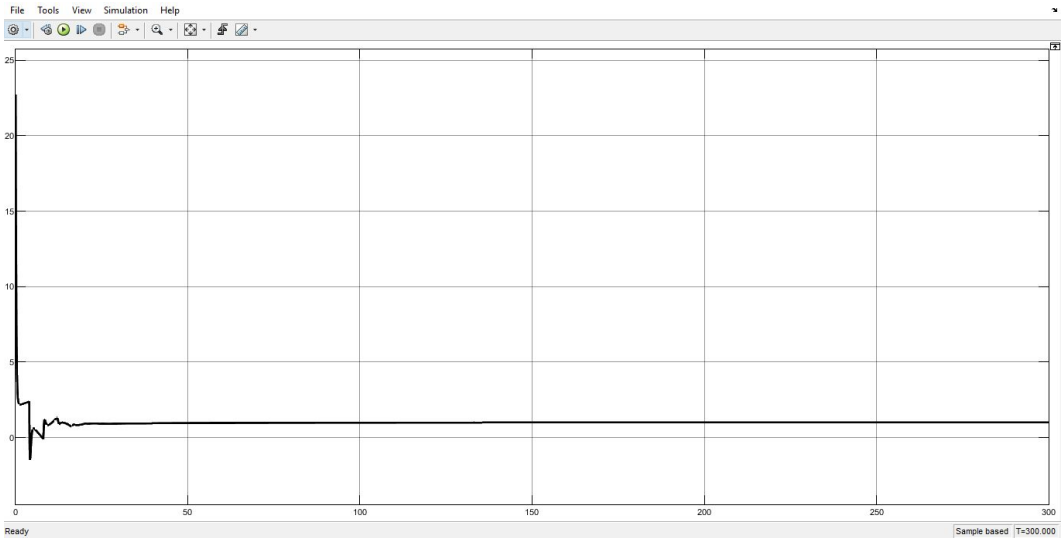


Figure 31. Set-point step response for Process P1- Control Variable

b. Process Variable

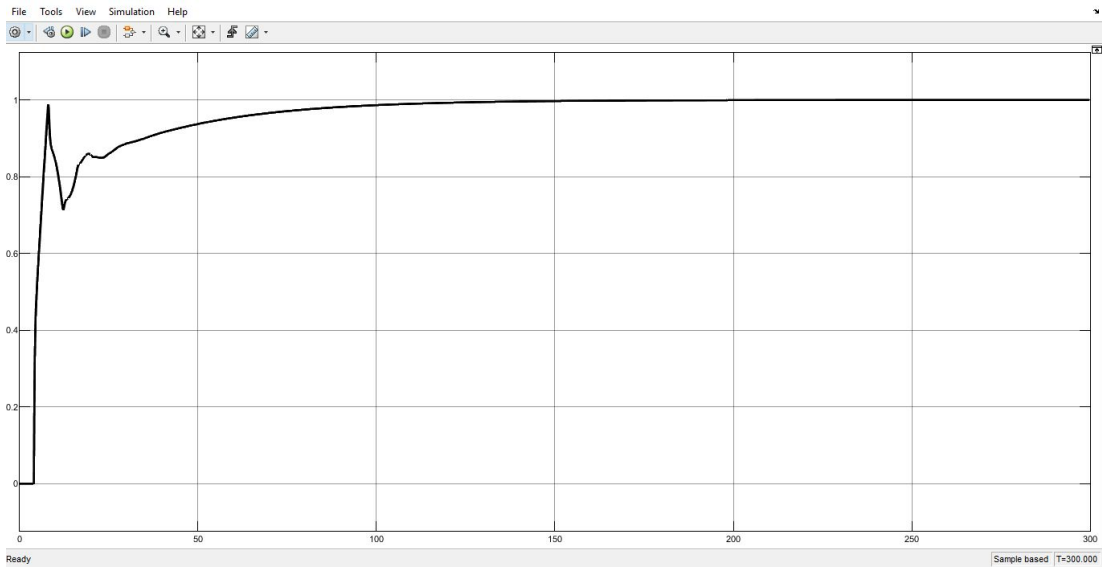


Figure 32. Set-point step response for Process P1- ProcessVariable

Load Disturbance Step Response for the process P1

a. Control Variable:-

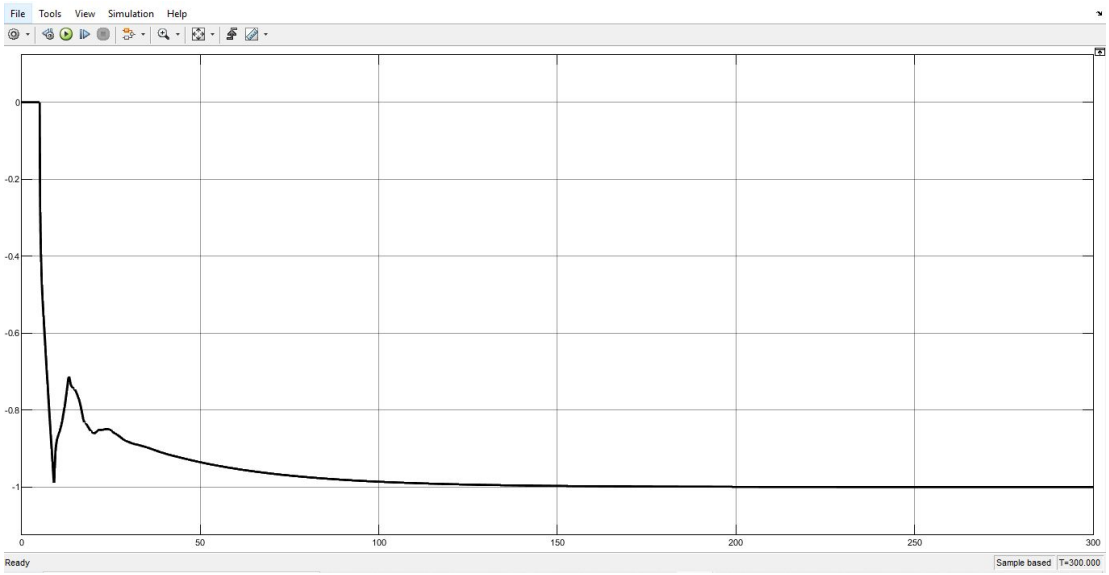


Figure 33. Load Disturbance step response for Process P1- Control Variable

b. Process Variable

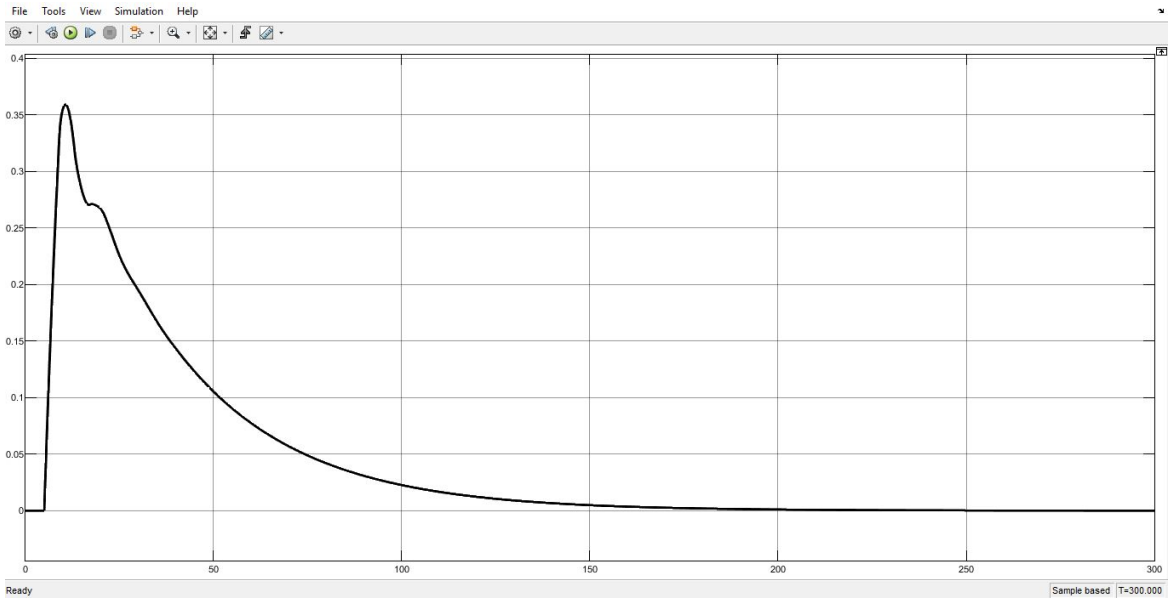


Figure 34. Load Disturbance step response for Process P1- ProcessVariable

5.4.2. Process 2:

$$P_2(s) = \frac{1}{(s+1)^4}$$

Parameters Calculated:-PID: P = 1.82 , I = 0.132, D = 1.81454, N = 10.03

Set-point Step Responses for the process P2

a. Control Variable:-

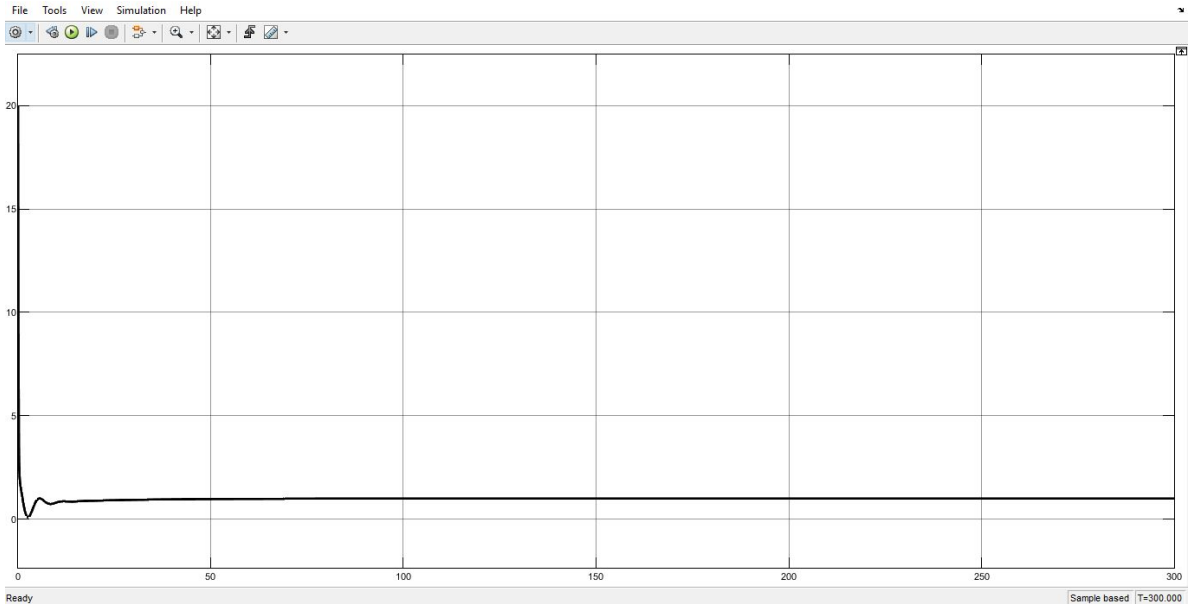


Figure 35. Set-point step response for Process P2- Control Variable

b. Process Variable

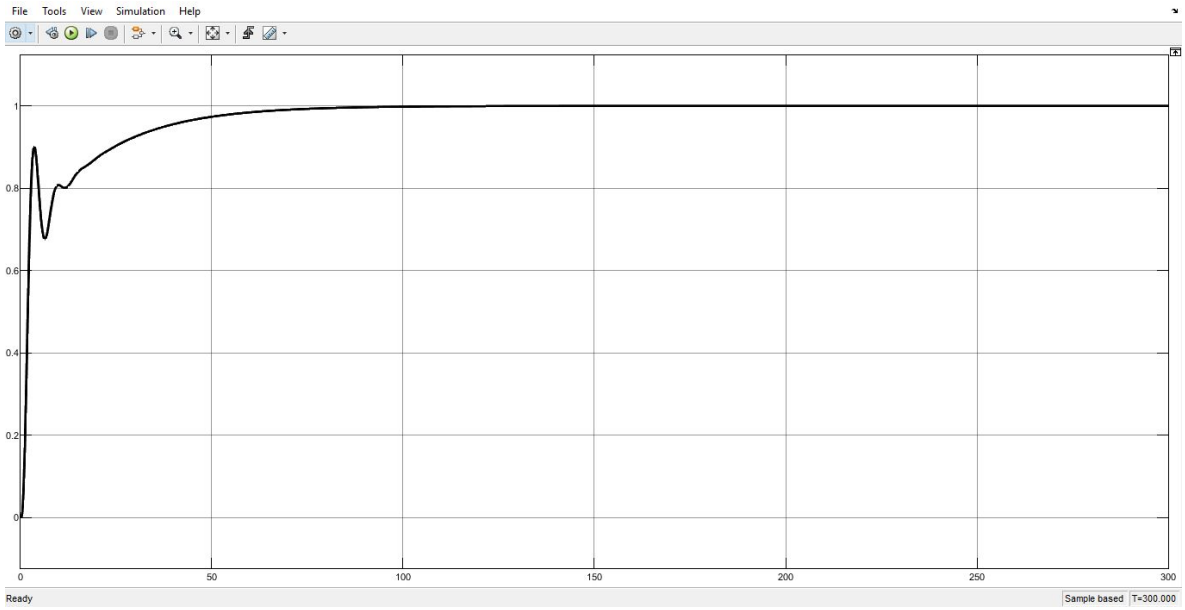


Figure 36.Set-point step response for Process P2- ProcessVariable

Load Disturbance Step Response for the process P2

a. Control Variable:-

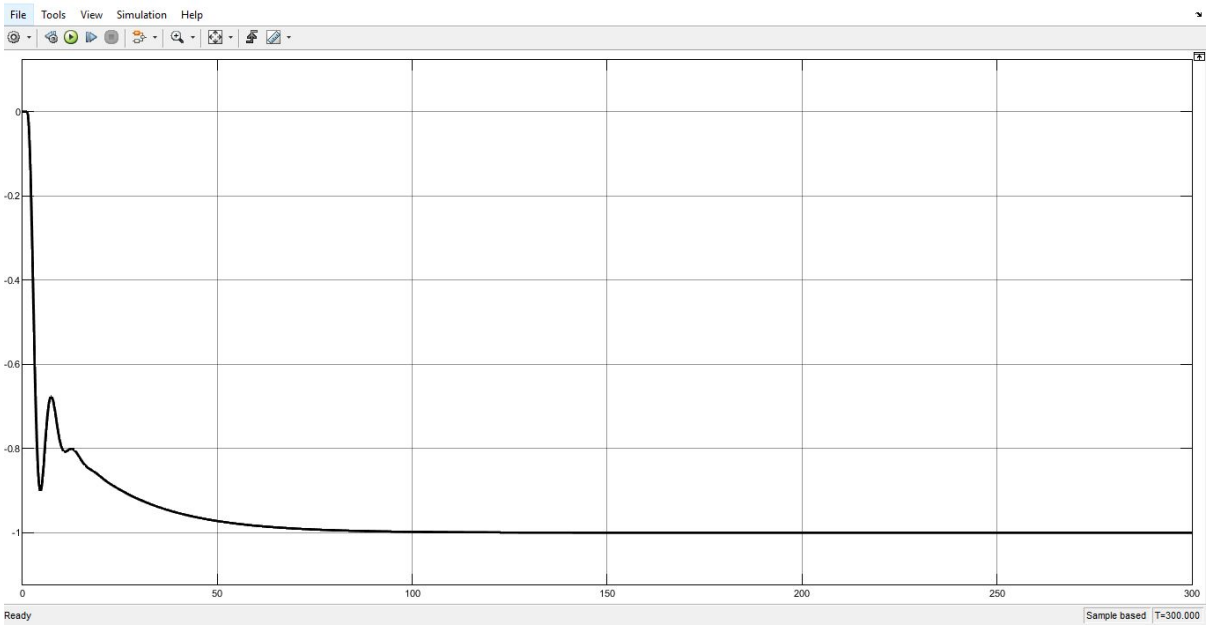


Figure 37. Load Disturbance step response for Process P2- Control Variable

b. Process Variable

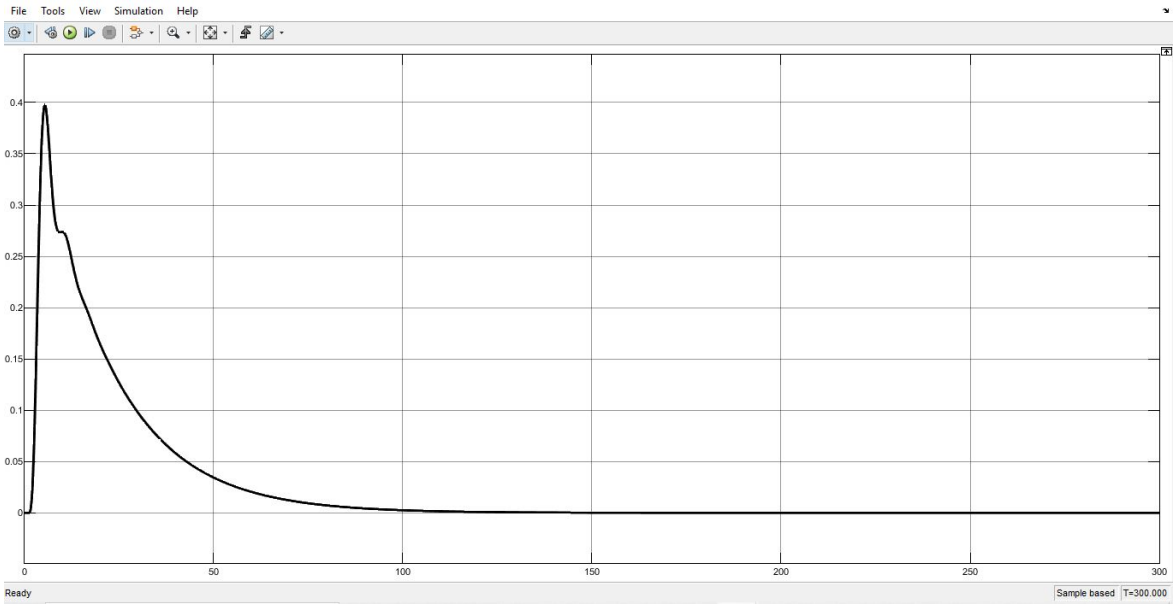


Figure 38. Load Disturbance step response for Process P2- ProcessVariable

5.4.3. Process-3

$$P_3(s) = \frac{1}{(s+1)^{20}}$$

PID: P = 0.58 , I = 0.0067, D = 3.61514, N = 1.604

Set-point Step Responses for the process P3

a. Control Variable:-

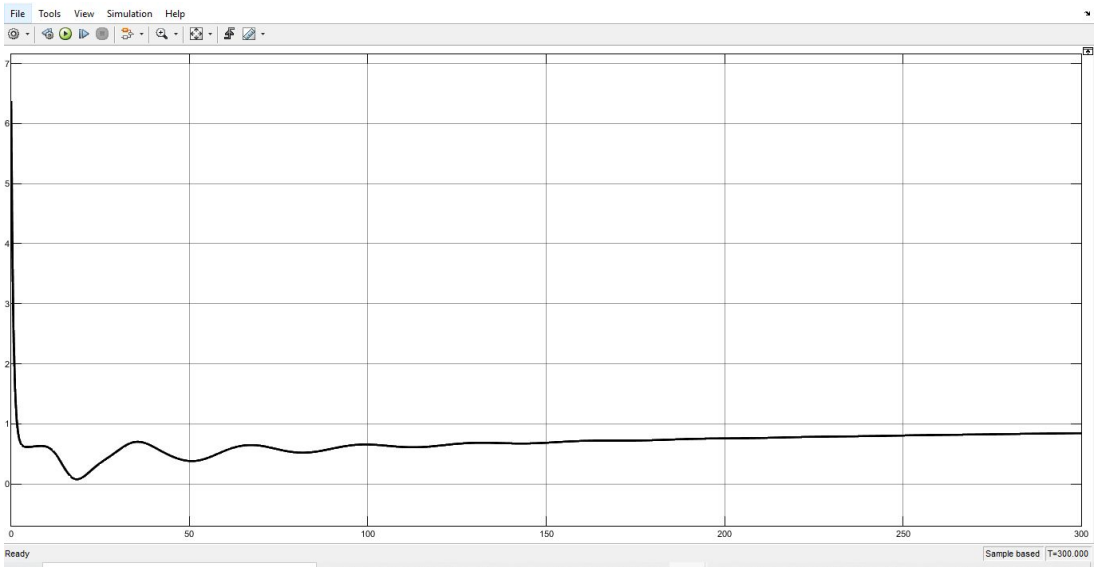


Figure 39. Set-point step response for Process P3- Control Variable

b. Process Variable

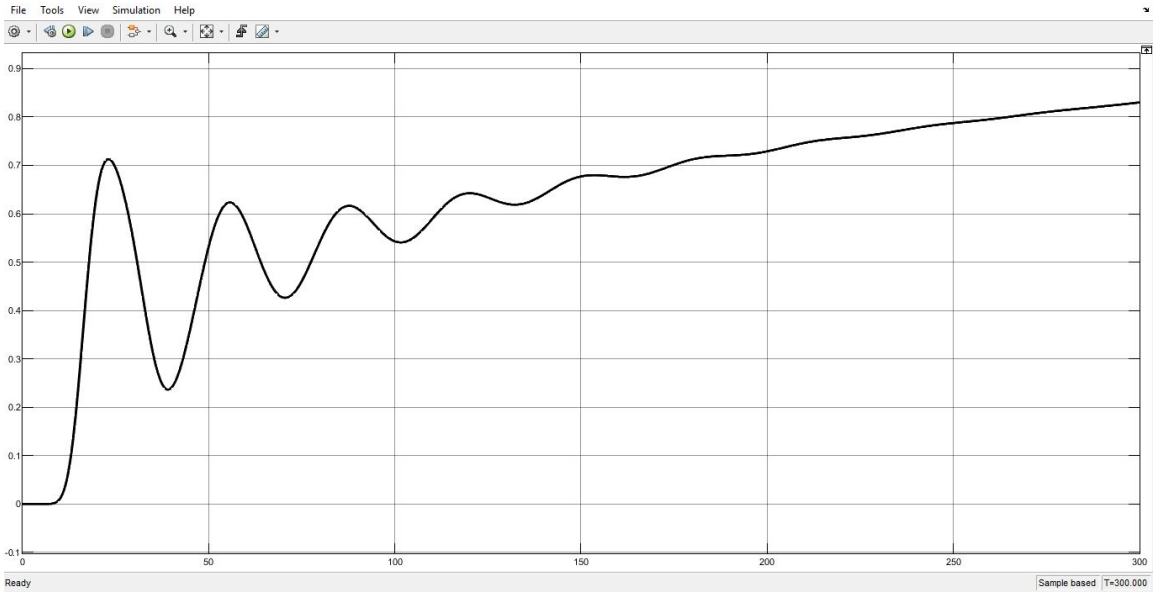


Figure 40. Set-point step response for Process P3- ProcessVariable

Load Disturbance Step Response for the process P3

a. Control Variable:-

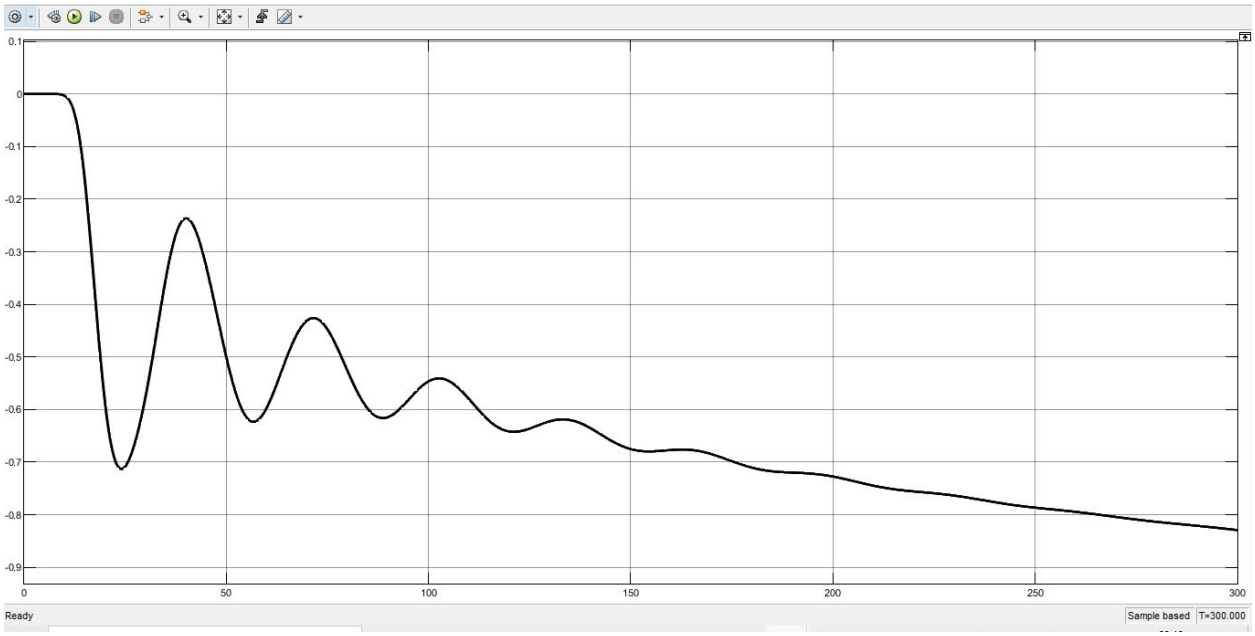


Figure 41. Load Disturbance step response for Process P3- Control Variable

b. Process Variable

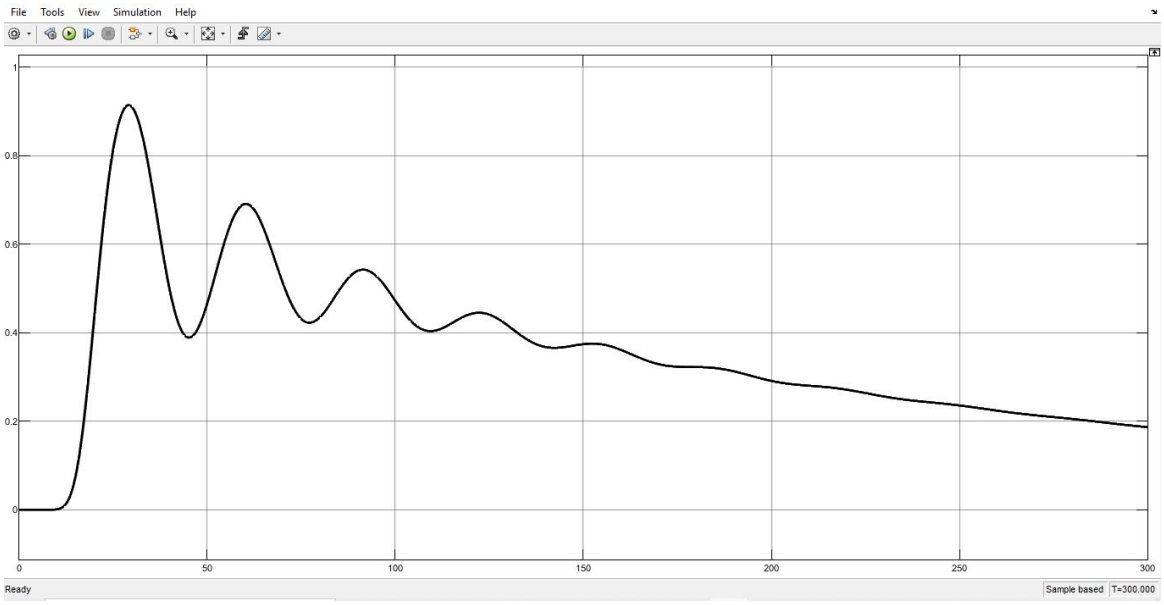


Figure 42. Load Disturbance step response for Process P3- ProcessVariable

5.5. Ziegler Nichols PRC method:

5.5.1. Process-1:

$$P_1(s) = \frac{1}{10s+1}e^{-4s}$$

Parameters obtained using PRC Method:
L =3.9921, T = 10.6201, a =0.3759, Kp =3.1923 ,Ti = 7.9843, Td =1.9961

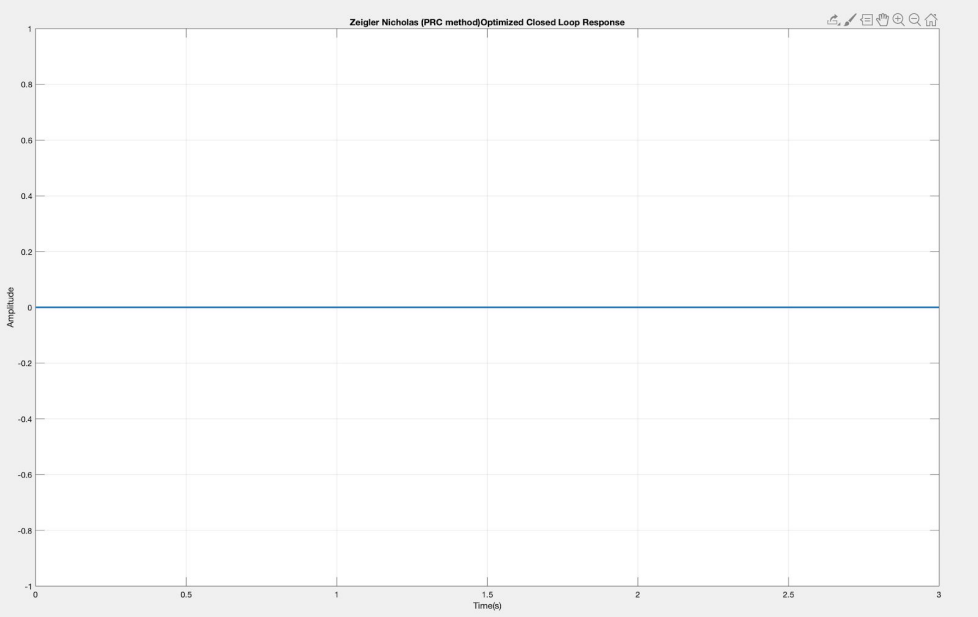


Figure 43. Optimised Closed Loop Response for P1

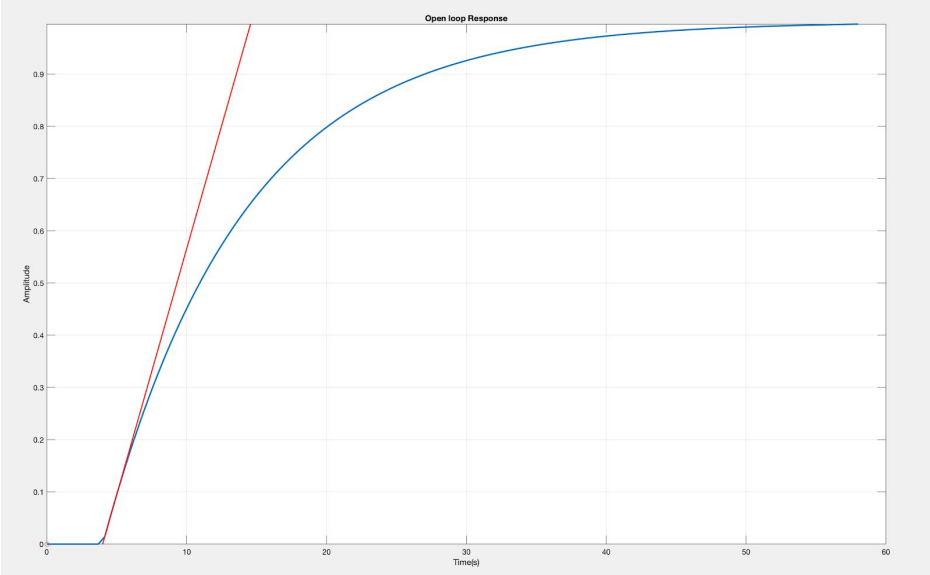


Figure 44. Closed Loop Response for P1

5.5.2. Process-2:

$$P_2(s) = \frac{1}{(s+1)^4}$$

Parameters obtained using PRC Method
t_infl =0.0060, y_infl = 2.4997e-10, L =1.4254, T = 4.4635, a = 0.3194, Kp = 3.7576, Ti = 2.8509, Td = 0.7127

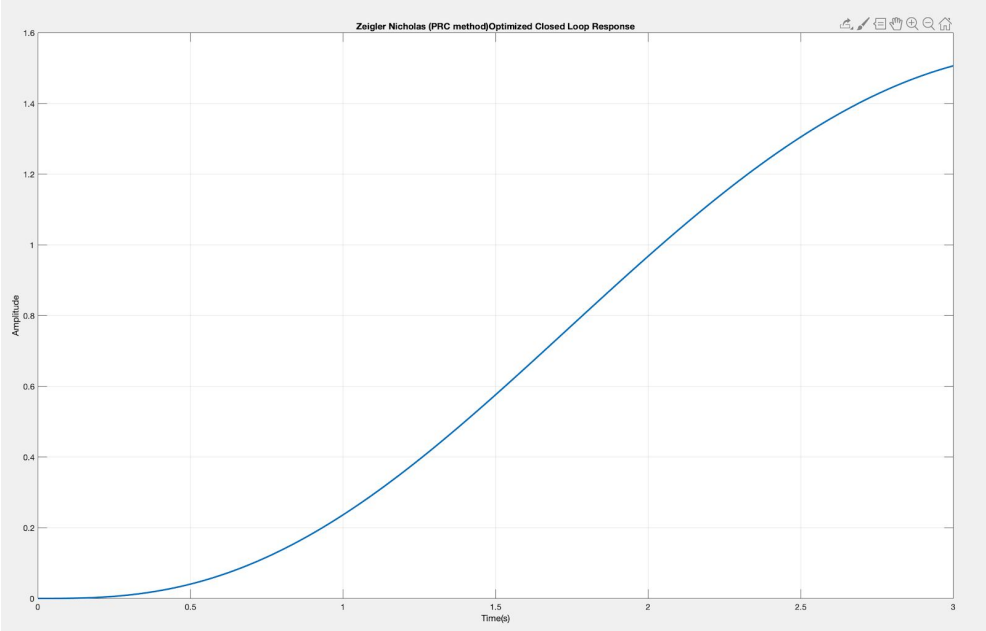


Figure 45. Optimised Closed Loop Response for P2

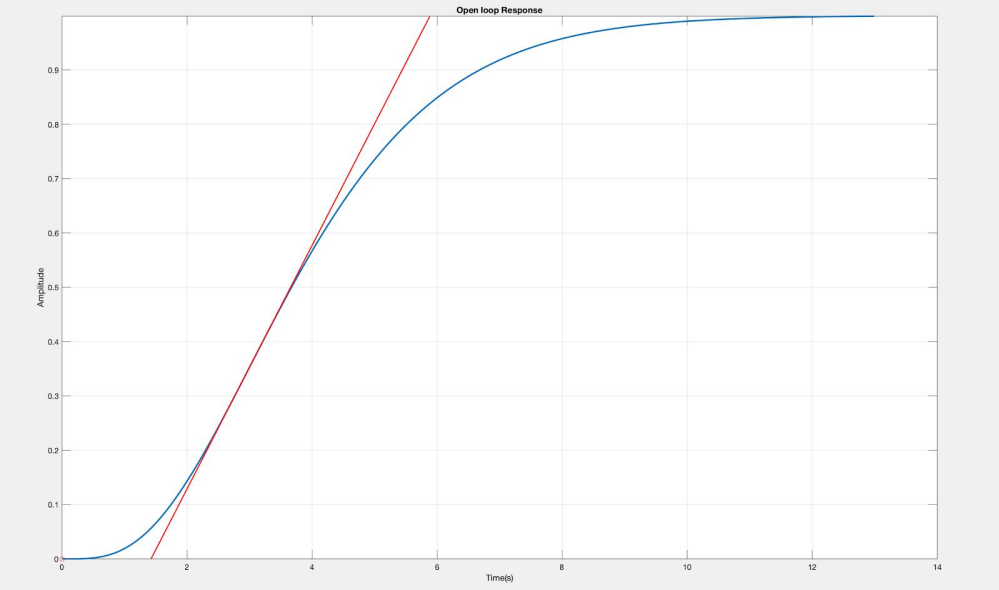


Figure 46. Closed Loop Response for P2

5.5.3. Process-3:

$$P_3(s) = \frac{1}{(s+1)^{20}}$$

Parameters obtained using PRC Method
t_infl =0.0100, y_infl =2.8281e-58, L =14.1780, T = 10.9742, a = 1.2919, Kp =0.9288, Ti =28.3561, Td =7.0890

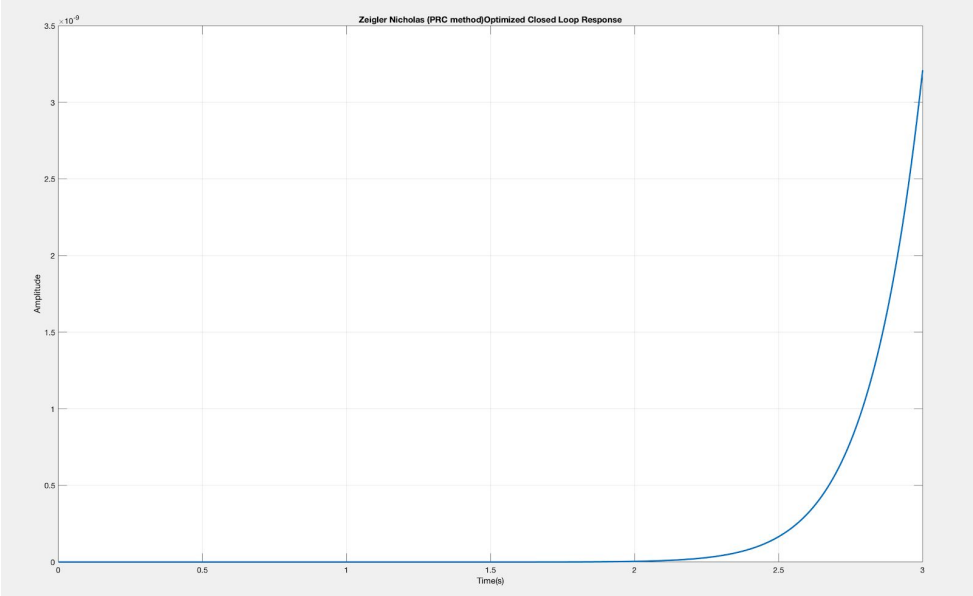


Figure 47. Optimised Closed Loop Response for P3

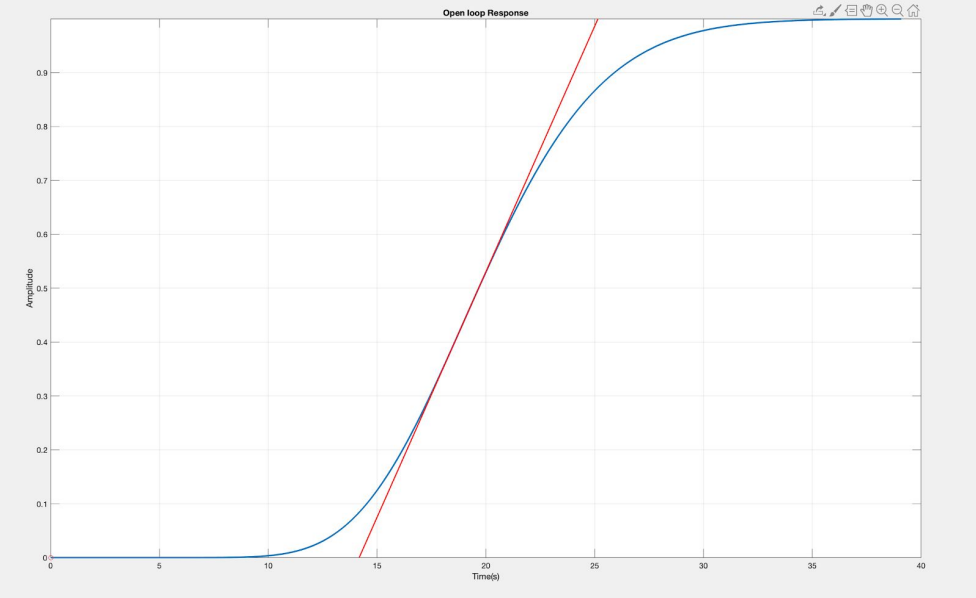


Figure 48. Closed Loop Response for P3

5.6. ISE Calculation and Comparison of the tuning methods used for PID Controller:

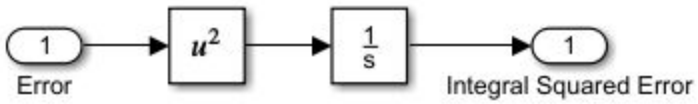


Figure 49. Simulink Block Used for ISE Calculation

Processes	Nash ISE (Set Point)	Nash ISE (Load Disturbance)	ZN (Set Point)	ZN (Load Disturbance)	TL (Set Point)	TL (Load Disturbance)
P1	**4.80**	**1.00**	**5.29**	**0.661**	**5.464**	**2.337**
P2	**1.94**	**0.53**	**1.512**	**0.3249**	**2.321**	**1.451**
P3	**16.10**	**11.62**	**17.48**	**13.29**	**52.41**	**48.11**

6. Observations and learnings from the paper:

- Paper introduces a new approach of looking at control problems for minimizing both the ISEs (load disturbance reduction and set point tracking). Such an approach is needed when we do not want to be biased towards one operation mode.

- First we implemented a first order process and then gradually moved towards a higher order process for design in MATLAB Simulink. We learnt to simulate the PID controlled process and could comparatively analyse the results obtained from different tuning methods graphically.

- By comparing ISEs for servo action and regulatory action for various tuning methods we could relate the calculated ISEs and order of the process.

- The novelty of Nash Bargaining solution intrigued us to look further into proving the tuning constants given in the paper.

7. Conclusion:

The project tries to make an attempt to bring forward the significance of using Nash Tuning. As described in the earlier sections of the report regarding the process of arriving at the constants for Nash Equilibrium, the method has its own differences as compared to the other Tuning methods,In particular, the tuning allows the minimization of the integrated square error for the set-point following and load disturbance rejection task subject to a constraint on the **maximum sensitivity**. The paper tries to develop tuning rules on the basis of Nash Solution. They are also found to be easily implemented in **standard Industrial controllers.** Moreover, the paper tries to implement robustness explicitly in the optimisation procedure.

Through the project, we have tried to replicate the results published in the referred paper using our own simulink block diagram model. Furthermore, in order to gauge deeper into the significance of nash tuning, we tried to compare the tuning results with that of the results obtained from Ziegler-Nicholas and Tyres-Luyben tuning methods. In all the cases, we tried to model the PID controller for the three example processes of different orders similar to what is proposed in the paper for Nash Solution. The comparison table lists the Integral Squared Error for each case. As can be observed from the comparison table, for the set-point response, the ISE is minimum in the case of Nash Tuning, followed by Z-N and T-L for process P1. On the other hand, the ISE for load disturbance is found to be minimum for Z-N followed by Nash and T-L. Similarly, for process P2, the ISE is found to be least for Z-N Tuning for both set point and load disturbance response. For higher order processes like P3, Nash solution proves to be more efficient with minimum ISE values. In sum, through the presented examples and simulation graphs it can be observed that it is extremely fundamental for the devised tuning rules to offer a balance between both **operation modes with a smooth response** and **acceptable values of performance and robustness.**

8. References:

1. O. Arrieta , A. Visioli , R. Vilanova, **PID autotuning for weighted servo/regulation control operation**, J. Process Control 20 (4) (2010) 472–480.

2. A. Messac , A. Ismail-Yahaya , C.A. Mattson, **The normalized normal constraint method for generating the Pareto frontier**, Struct. Multidiscip. Optim. 25 (2) (2003) 86–98.

3. H. Sánchez , A. Visioli , R. Vilanova, **Nash tuning for optimal balance of the servo/regulation operation in robust PID control**, in: Proceedings of the 23rd Mediterranean Conference on Control and Automation, Torremolinos, Spain, 2015, pp. 715–721.

4. Normalized Normal Constraint (NNC) algorithm for multi-objective optimization, Mathworks, File exchange [Blog post]. Retrieved from:
   https://in.mathworks.com/matlabcentral/fileexchange/38976-normalized-normal-constraint-nnc-algorithm-for-multi-objective-optimization

9. Appendix:

Matlab Code for PRC Method Implementation with open and closed loop response:

```
close all; clear all; clc;
% define pant transfer function
s = tf('s');
% 2nd order system
%sys = (exp(-4*s))/(10*s+1)   %Example-1
%sys = 1/(s^4 + 4*s^3+ 6*s^2+ 4*s+ 1); %Example-2
sys = 1/(s+1)^20;%Example-3

%sys = 1/(s^2+20*s+20) %random example
% Obtain step response of the system
[y,t] = step(sys);
plot(t,y,'LineWidth',2); grid on; xlabel('Time(s)'); ylabel('Amplitude');
title('Open loop Response');

yp = diff(y);
ypp = diff(y,2);
%plot (ypp)
% Find the root using FZERO(Finds Inflection point ----- verify pakka pakka)
t_infl = fzero(@(T) interp1(t(2:end-1),ypp,T,'linear','extrap'),0)
y_infl = interp1(t,y,t_infl,'linear')
hold on;
plot(t_infl,y_infl,'ro');

%code to plot tangent
h = mean(diff(t));
dy = gradient(y, h);
[~,idx] = max(dy);
b = [t([idx-1,idx+1]) ones(2,1)] \ y([idx-1,idx+1]);          % Regression Line Around Maximum
Derivative
tv = [-b(2)/b(1); (1-b(2))/b(1)];                    % Independent Variable Range For Tangent Line Plot
f = [tv ones(2,1)] * b;                              % Calculate Tangent Line
```

```
plot(tv, f, '-r','LineWidth',1.5)
ylim([0 max(y)]);

L = tv(1)%equal to delay time for us(verified)
T = tv(2) - tv(1)% should be equal to tp/K in order to verify the coefficients

% PID parameters
a = L/T
Kp = 1.2/a
Ti = 2*L
Td = L/2

% cont = Kp(1 + 1/(s*Ti) + s*Td);
cont = Kp + Kp/(s*Ti) + Kp*Td*s;

cl_sys = feedback(cont*sys,1);
t = [0:0.01:3];
[yc,tc] = step(cl_sys,t);
figure;
plot(tc,yc,'LineWidth',2); xlabel('Time(s)'); ylabel('Amplitude');
title('Zeigler Nicholas (PRC method)Optimized Closed Loop Response');
grid on;
```

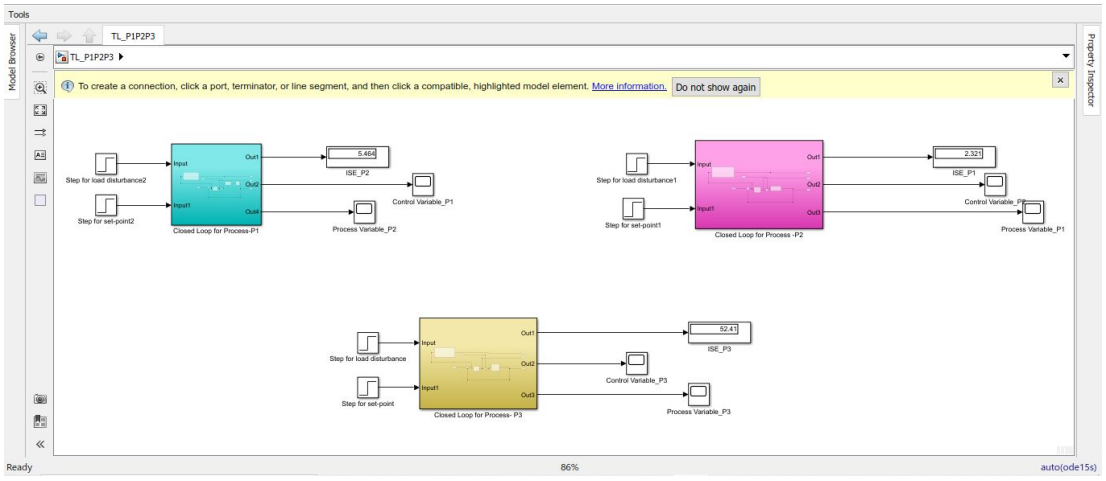


Figure:50. Screenshot of the Simulink Block Diagram for different Tuning Methods used