Simulation and Numerical Analysis and Comparative Study of a PID Controller Based on Ziegler-Nichols and Auto Turning Method

Andrea Scherlozer, Mestaro Orsini, Sulvane Patole

Simulation and Numerical Analysis and Comparative Study of a PID Controller Based on Ziegler-Nichols and Auto Turning Method

Andrea Scherlozer*, Mestaro Orsini, Sulvane Patole

Mechanical Engineering Department, Edinburgh University, Scotland

Abstract

Overall in any system the Proportional term, the Integral term as well as the Derivative term contribute to achieving a fast rise time, minimum overshoot, no oscillations and higher stability as well as no steady-state error. To achieve stability in typical PID systems, it is important to eliminate the steady state errors associated with such systems. MATLAB M-file was generated to plot responses of the transfer function with different integrator gains for auto tuning and both Ziegler-Nichols and auto tuning methods were used to the removal of steady state errors in PID systems. It was observed that both methods can be adopted for the elimination of steady state error in PID systems, but the drawback associated with Ziegler-Nichols method is that, it is time consuming and may delay while entering into an unstable region for the system. This paper proposed a comparative study of PID controller for these methods with simulation and numerical study.

Keywords: Control methods, Ziegler Nichols, Steady state error, Stability, PID controller, Simulation, Numerical study.

Introduction:

Control systems has been used extensively during past decades. One of the most important and successful control systems is PID controller, specifically in industries where more than 90 percent of industrial controllers were PID family on 2002. The three term PID controller has three basic modes for controlling the characteristics of a second order system. The three basic modes are: the Proportional term, the Integral term as well as the Derivative term. The PID controller is basically used in controlling closed loop form of an open loop system. But it is broadly applicable since a PID controller relies only on the measured process variable, not on knowledge of the underlying process. The proportional term of the controller is in proportion to the error in the system as the name implies, the integral term is proportional to the integral of the past errors while the derivative term is proportional to the rate of change of the error (Hunter, 1987; Krishnaswamy, 2011). Where is the integral time constant and the derivative time constant. The proportional part acts on the present value of the error, the integral represents an average of past errors and the derivative can be interpreted as a prediction of future errors based on linear extrapolation.The controller can also be parameterized mathematically as:

$$G(s) = k_p\left(e(t) + \frac{1}{T_i}e(t)\,dt + T_d\frac{de(t)}{dt}\right) \qquad (1)$$

ISBN: 978-1-5090-1738-6

It is worthy of note that the control signal u is formed entirely from the error e. Overall in any system the Proportional term, the Integral term as well as the derivative term contributes in achieving a fast rise time, minimum overshoot, no oscillations and higher stability as well as no steady-state error (Zilouchian and Jamshidi, 2001; Owunna et al., 2016). The role of each of the terms in a PID controller are tabulated in Table 1.

Table 1: Roles of PID controller terms

Term	Response	Rise Time	Overshoot	Settling Time	S-S Error
Proportional		Decrease	Increase	Small Change	Decrease
Integral	(—)	Decrease	Increase	Increase	Eliminate
Derivative		Small Change	Decrease	Decrease	No Change

The proportional controller often times reduces the rise time and often reduces the steady state error of the system but never does it eliminate the error completely. The integral control on the other hand eliminates completely the steady state error of a system. A derivative control however increases the stability of the system, reduces the overshoot as well as improves the transient response of the system. It does not in any way alter the steady state error of the system. In this paper we compare our methods simulation result with the fuzzy system controllers implemented on aerial and surface vessels (Abbasi et al., 2013; Yazdanpanah et al., 2013). Our model has been developed based on the results of these papers and comparison with their model.

Type 1 Systems

Type 1 systems are systems that do not have any steady state error. Since these systems have no steady state error, the presence of integrators will be superfluous to the system. This is because the major function of integrator in controllers is to eliminate steady state errors which are absent in type 1 open loop systems (Cooper, 2007). The significant challenge connected with the derivative controller noise issues and sensitivities that, a large frequency within a system associated with large changes in the system error may cause the derivative of the signal to amplify the signal significantly. Thus little levels of noise present in the system may cause the output of the system to increase greatly. In other words, the sensitivity of derivative controllers to noise may result in significant changes in the value of the output as a result of small level of noise in the system. In these circumstances, it is often sensible to use a PI controller or set the derivative action of a PID controller to zero. To eliminate/minimize this downside, an electronic signal filter may be enclosed within the loop. Electronic signal filters are unit electronic circuits that perform signal process functions, specifically supposed to get rid of unwanted signal components and/or enhance needed ones. Electronic filters can be: passive or active, analogue or digital, discrete-time (sampled) or continuous-time, linear or non-linear, etc. The most common types of electronic filters are linear filters, regardless of other aspects of their design.

Ziegler-Nichols Closed-loop Tuning Method

In the Ziegler-Nichols closed-loop tuning method, the ultimate gain K_u and the ultimate period of oscillations P_u is employed in calculating the needed K_c which is the value of the proportional gain required for effective tuning of the system. The Ziegler-Nichols closed-loop tuning method is only applicable in closed-loop systems and cannot be applied in open loop systems. To determine the value of K_c, the value of the proportional gain that will produce a steady oscillation in the system is first obtained. The gains for the integrator and the derivative controllers are initially set at zero for the procedure. When the systems oscillates steadily, the period of oscillation must therefore be obtained as it is required in calculating the integral and derivative times. The ultimate period is the time required to complete one full oscillation while the system is at steady state. To find the values of the PID parameters from the values of K_c and period obtained, the following procedures must be adopted.

Closed Loop (Feedback Loop)

i. The derivative controller gain and the integral gains must be set at zero.
ii. The proportional gain should then be varied till the system oscillates at constant amplitude.
iii. The values of K_u and the period of the oscillation can then be recordedP_u.

To obtain the various values for the PID controllers Ziegler-Nichols equation must be used and the equation is presented in the Table 2.

Table 2: Ziegler-Nichols closed-loop tuning formula

Rule Name	Tuning Parameters		
	Kp	Ki	Kd
P	0.5 Ku		
PI	0.45Ku	1.2Kp/Pu	
PID	0.6 Ku	2Kp/Pu	KpPu/8

Closed Loop Systems-P Controller

P controllers are often used in first order systems to stabilize unstable responses. P controller helps to majorly reduce the steady state error of the system. An increase in the proportional gain factor K of the P controller reduces the steady state error of the system (Ogata, 1997). It is however worthy of note that P controllers can reduce but not eliminate totally the steady state error of a system. As the proportional gain of the P controller increases, smaller amplitudes as well as smaller phase margin are introduced to the system. The dynamic of the system also becomes faster and the sensitivity of the system to noise reduces as the proportional gain increases. The system is applicable only in instances where the system can tolerate constant steady state error (Taeib and Chaari, 2015).

P-I Controller

P-I controller are used majorly in the elimination of steady state errors arising from P controllers. The PI controllers have a negative effect on the stability of a system as well as the response speed of the system. It is therefore important to note that P-I controllers are useful in systems where the response speed is insignificant. P-I controllers have no effect on the rise time and cannot eliminate oscillations in a system because they cannot accurately predict future errors within the system.

P-I-D Controller

With PID controllers, zero state errors are possible. The response of the system can be improved to achieve a fast response, oscillations in the system can be removed and the stability of the system can be improved. A derivative of the output response is often added to a PI controller to remove overshoot and oscillations in the system. PID controllers have the advantage of use in higher order systems.

P-D Controller

P-D controllers are used majorly to increase system stability as the controller is able to predict future error that can occur in the systems response. A derivative of the output response is often used instead of using the error in the signal to ensure there is no abrupt change in the value of the error of the signal (Padula and Visioli, 2011; Taeib and Chaari, 2015). The derivative controllers is often not used alone to prevent amplification of noise in the system.

Research Methodology

For optimum performance of control systems, the steady state errors must be eliminated to enable stability of the close loop systems which are basically controlled by PID. The closed loop transfer function was designed in Simulink and MATLAB M-file was generated to plot responses of the transfer function with different proportional gain. Ziegler-Nichols closed-loop tuning method and auto tuning system method to determine PID values and a MATLAB command was generated and simulated for both tuning methods. Result obtained from the MATLAB simulation was used to determine if the steady state error has been eliminated or not, and how effective each method is.

Steady State Error (P controllers)

For the given transfer function

$$G(s) = \frac{10}{s^2 + 5s + 6}$$

Recall that for a closed loop system,

$$e_{ss} = \lim_{s \to 0} \left[s \frac{1}{1+G(s)} \theta_i(s) \right] \qquad (2)$$

For unit step input,

$$\theta_i(s) = \frac{1}{s} \qquad (3)$$

Thus,

$$e_{ss} = \lim_{s\to 0}\left[s\frac{1}{1+G(s)}\theta_i(s)\right] = \lim_{s\to 0}\left[s\frac{1}{1+G(s)}.\frac{1}{s}\right]$$
(4)

$$= \lim_{s\to 0}\left[\frac{1}{1+G(s)}\right]$$
(5)

For

$$G(s) = \frac{10}{s^2+5s+6}$$

$$1+G(s) = 1+\frac{10}{s^2+5s+6}$$

$$= \frac{s^2+5s+6+10}{s^2+5s+6}$$

$$= \frac{s^2+5s+16}{s^2+5s+6}$$

$$\frac{1}{1+G(s)} = \frac{1}{\left(\frac{s^2+5s+16}{s^2+5s+6}\right)}$$

$$= \frac{s^2+5s+6}{s^2+5s+16}$$

Therefore,

$$e_{ss} = \lim_{s\to 0}\left[s\frac{1}{1+G(s)}\theta_i(s)\right]$$

$$= \lim_{s\to 0}\left[\frac{1}{1+G(s)}\right]$$

$$= \lim_{s\to 0}\left[\frac{s^2+5s+6}{s^2+5s+16}\right]$$

$$= \frac{0^2+5(0)+6}{0^2+5(0)+16}$$

$$= \frac{6}{16} = 0.375$$

For P controller with gain K_p the closed loop transfer function becomes;

$$G_0(s) = \frac{K_pG(s)}{1+K_pG(s)} \qquad (6)$$

$$= \frac{K_p \times \frac{10}{s^2 + 5s + 6}}{1 + K_p \times \frac{10}{s^2 + 5s + 6}}$$

$$= \frac{\frac{10K_p}{s^2 + 5s + 6}}{1 + \frac{10K_p}{s^2 + 5s + 6}}$$

$$= \frac{\frac{10K_p}{s^2 + 5s + 6}}{\frac{s^2 + 5s + 6 + 10K_p}{s^2 + 5s + 6}}$$

$$= \frac{10K_p}{s^2 + 5s + 6 + 10K_p}$$

For steady state error

$$1 + G(s) = 1 + \frac{10K_p}{s^2 + 5s + 6 + 10K_p}$$

$$= \frac{s^2 + 5s + 6 + 10K_p + 10K_p}{s^2 + 5s + 6 + 10K_p}$$

$$= \frac{s^2 + 5s + 6 + 20K_p}{s^2 + 5s + 6 + 10K_p}$$

$$\frac{1}{1 + G(s)} = \frac{1}{\left(\frac{s^2 + 5s + 6 + 20K_p}{s^2 + 5s + 6 + 10K_p}\right)} = \frac{s^2 + 5s + 6 + 10K_p}{s^2 + 5s + 6 + 20K_p}$$

Therefore,

$$e_{ss} = \lim_{s \to 0} \left[s \frac{1}{1 + G(s)} \theta_i(s) \right]$$

$$= \lim_{s \to 0} \left[\frac{1}{1 + G(s)} \right]$$

$$= \lim_{s \to 0} \left[\frac{s^2 + 5s + 6 + 10K_p}{s^2 + 5s + 6 + 20K_p} \right]$$

$$= \frac{0^2 + 5(0) + 6 + 10K_p}{0^2 + 5(0) + 6 + 20K_p}$$

$$= \frac{6 + 10K_p}{6 + 20K_p}$$

For stability of the system, using Routh Hurwitz criterion,

s^2	1	$6 + 10K_p$
s	5	0
s^0	$6 + 10K_p$	

For stability,

$$6 + 10K_p > 0$$

$$10K_p > -6$$

$$K_p > \frac{-6}{10}$$

$$K_p > -0.6$$

But

$$e_{ss} = \frac{6+10K_p}{6+20K_p}$$

For $K_p > -0.6$,

$$e_{ss} = \frac{6+10K_p}{6+20K_p}$$

This implies that if the system will remain stable, there will remain within the system steady state error if a proportional controller is used. To establish the fact that there will remain within the system steady state error if a proportional controller is used, a MATLAB M-file was generated to plot responses of the transfer function with different proportional gain. The MATLAB code for the simulation is written as follows:

```
% To obtain the Unit-Step Response of the System

num=[0 0 10]; % num_sys is the numerator of the system transfer function
den=[1 5 6]; % den_sys is the denominator of the system transfer function
K=1;
K2=2;
K3=5;
% specify proportional controller
% To obtain the transfer function
s=tf('s');
```

```
Gs=tf(num,den); %Gs is the system transfer function
%To plot the step response
figure(1) %specifies figure number
step(Gs, feedback (Gs*K,1),feedback (Gs*K2,1),feedback (Gs*K3,1))
hold on
plot([-0.1,0,0,4],[0,0,1,1],'r');
axis([-0.1 4 0 1.8]);
legend('Gs','K=1','K2=2','K3=5', 'target');
%specify title and grid
grid on
title ('Unit-Step Response of the system')
```

The plot generated from the system shows plots of step response from the real transfer function and the closed loop systems with different proportional gains as well as the target response. Figure 1 shows the plots with different proportional gains, from the plots shown in Figure 1 it is obvious that with increasing value of proportional gain, the steady state gain remained.

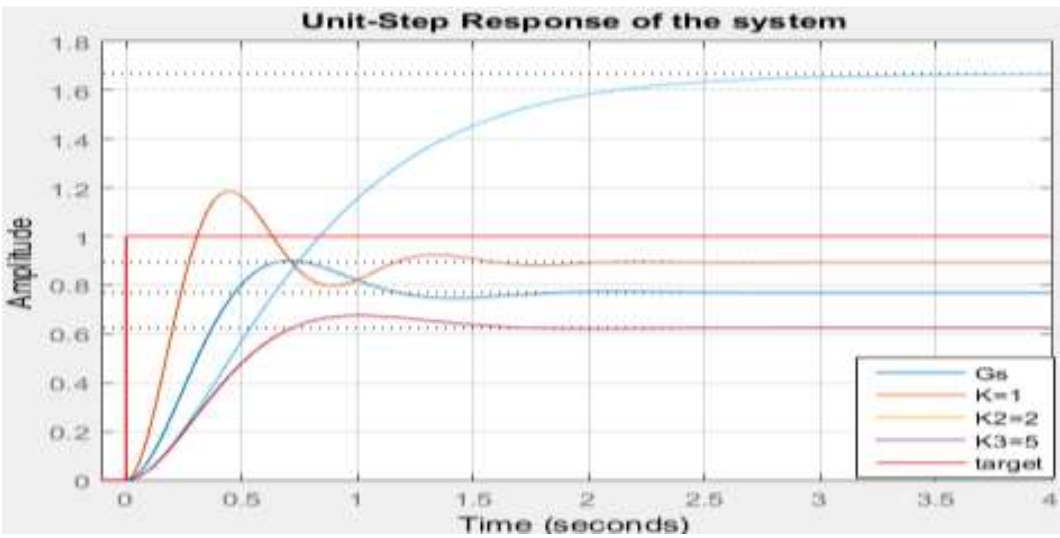

Figure 1: Plot of Transfer Function with Proportional Controller

PI Controllers

For PI controller with proportional gain K_p and integral gain $\frac{K_i}{s}$ the closed loop transfer function becomes;

$$G_0(s) = \frac{K_p\left[s+\frac{1}{T_i}\right]G(s)}{s} \quad (7)$$

$$= \frac{K_p\left[s+\frac{1}{T_i}\right]\frac{10}{s^2+5s+6}}{s}$$

$$= \frac{10K_p\left[s+\frac{1}{T_i}\right]}{s(s^2 + 5s + 6)}$$

For steady state error

$$1 + G(s) = 1 + \frac{10K_p\left[s+\frac{1}{T_i}\right]}{s(s^2 + 5s + 6)} = \frac{s(s^2 + 5s + 6) + 10K_p\left[s+\frac{1}{T_i}\right]}{s(s^2 + 5s + 6)}$$

$$\frac{1}{1+G(s)} = \frac{1}{\left(\frac{s(s^2 + 5s + 6) + 10K_p\left[s+\frac{1}{T_i}\right]}{s(s^2 + 5s + 6)}\right)}$$

$$= \frac{s(s^2 + 5s + 6)}{s(s^2 + 5s + 6) + 10K_p\left[s+\frac{1}{T_i}\right]}$$

Therefore,

$$e_{ss} = \lim_{s\to 0}\left[s\frac{1}{1+G(s)}\theta_i(s)\right]$$

$$= \lim_{s\to 0}\left[\frac{1}{1+G(s)}\right]$$

$$= \lim_{s\to 0}\left[\frac{s(s^2 + 5s + 6)}{s(s^2 + 5s + 6) + 10K_p\left[s+\frac{1}{T_i}\right]}\right]$$

$$= \frac{0(0^2 + 5(0) + 6)}{0(0^2 + 5(0) + 6) + 10K_p\left[0+\frac{1}{T_i}\right]} = \frac{0}{10K_p\frac{1}{T_i}} = 0$$

This implies that at any value of with proportional gain K_p and integral gain $\frac{K_i}{s}$, there will be no steady state error within the system. To establish the fact that there will no remains of any steady state error within the system if a proportional-Integrator controller is used, a MATLAB M-file was generated to plot responses of the transfer function with different integrator gains. The MATLAB code for the simulation is written as follows:

```
% To obtain the Unit-Step Response of the System

num=[0 0 10]; % num_sys is the numerator of the system transfer function

den=[1 5 6]; % den_sys is the denominator of the system transfer function
% specify proportional and integrator controllers
```

```
Kp=1;
Ki=1;
Ki2=2;
Ki3=5;
%To combine the P-I controllers
K=pid(Kp,Ki);
K2=pid(Kp,Ki2);
K3=pid(Kp,Ki2);

% To obtain the transfer function

s=tf('s');
Gs=tf(num,den); %Gs is the system transfer function

%To plot the step response

figure(1) %specifies figure number

step(Gs, feedback (Gs*K,1),feedback (Gs*K2,1),feedback (Gs*K3,1))%to plot
%all feedback functions
hold on
plot([-0.1,0,0,4],[0,0,1,1],'r');
axis([-0.1 4 0 1.8]);

legend('Gs','Ki=1','Ki2=2','Ki3=5', 'target');
%specify title and grid
grid on
title ('Unit-Step Response of the system')
```

The plot generated from the system shows plots of step response from the real transfer function and the closed loop systems with different proportional-Integrator gains as well as the target response. From the plots shown in Figure 2, it is obvious that at whatever value of the integrator gain, the steady state error is always eliminated (Messner and Tilbury, 2015).

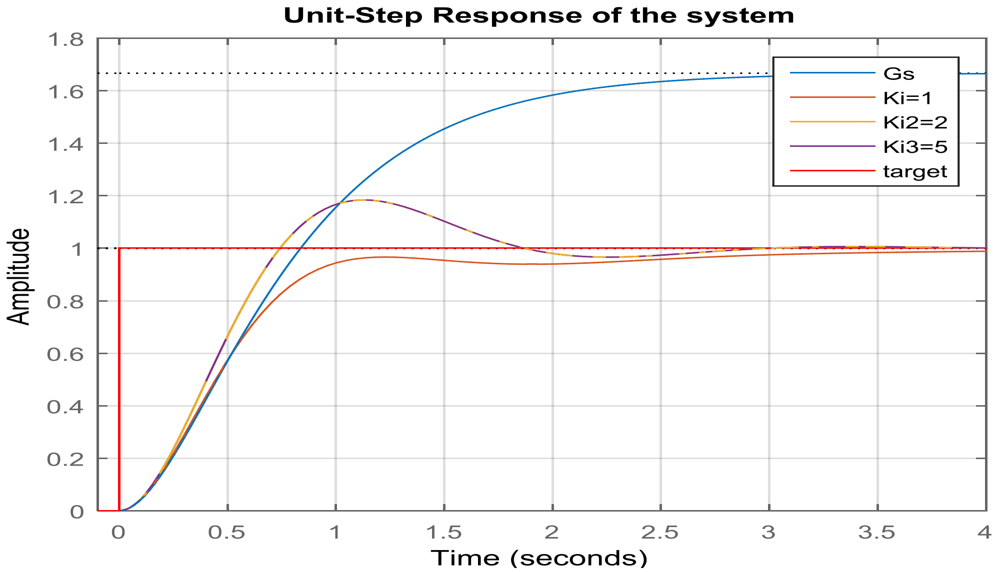

Figure 2: Plot of Transfer Function with Proportional-Integrator Controller

Ziegler-Nichols Closed-loop Tuning Method

The derivative controller gain and the integral gains were set at zero. The proportional gain was varied, until a relatively stable system was obtained at Ku=8.5. The response obtained is shown in Figure 3.

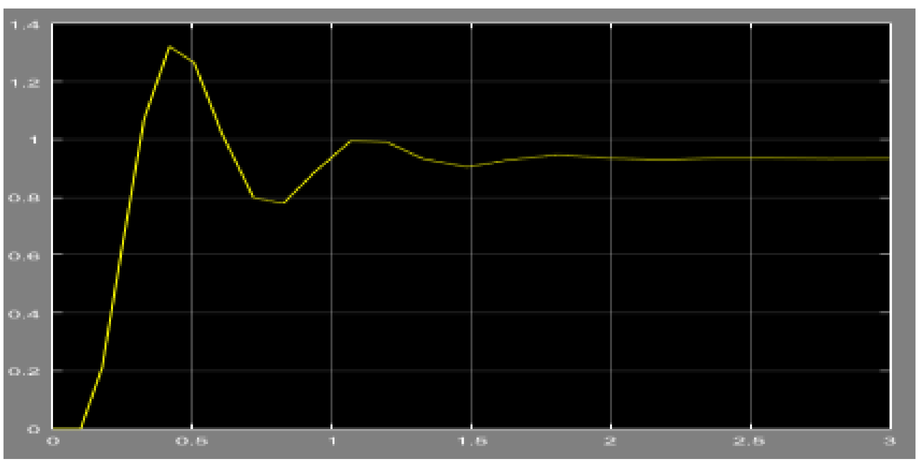

Figure 3: System response at Ku=8.5

At this gain, the period of oscillation is obtained as 0.7. From Ziegler-Nichols method, the PID values are given as shown in Table 3.

Table 3: Ziegler-Nichols tuning parameters

Rule Name	Tuning Parameters		
	Kp	Ki	Kd
P	4.25		

PI	3.825	14.6	
PID	5.1	24.28	0.74375

The PID Values if used in plotting the response of the system will give a situation as shown in Figure 4.

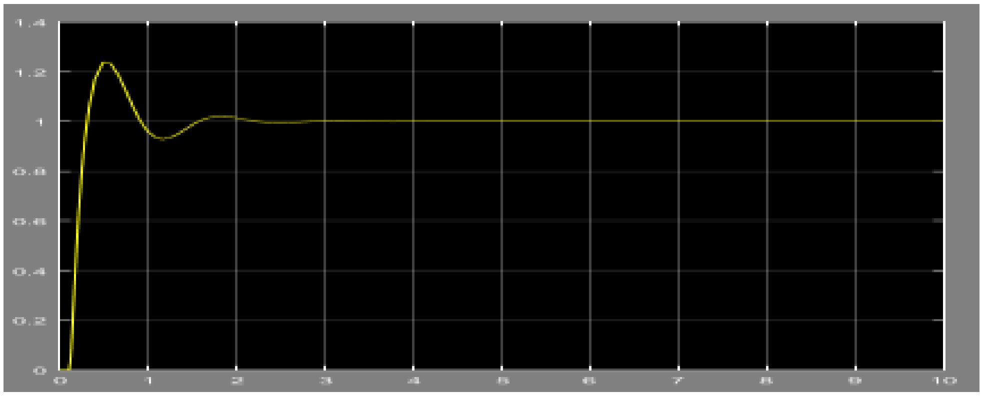

Figure 4: System response with Zeigler-Nichols Tuning

Comparing the initial response and the final response, it is obvious that the steady state error is eliminated using the Ziegler-Nichols method of the analysis. A backdrop of this method is that, it is time consuming and may delay while entering into an unstable region for the system.

MATLAB PID Tuning

It is necessary to tune the system represented above using automatic Simulink tuning. The different steps used are as follows (Nguyen, 2015);

i. Design the closed loop transfer function in Simulink

The design as generated from Simulink is shown in Figure 5.

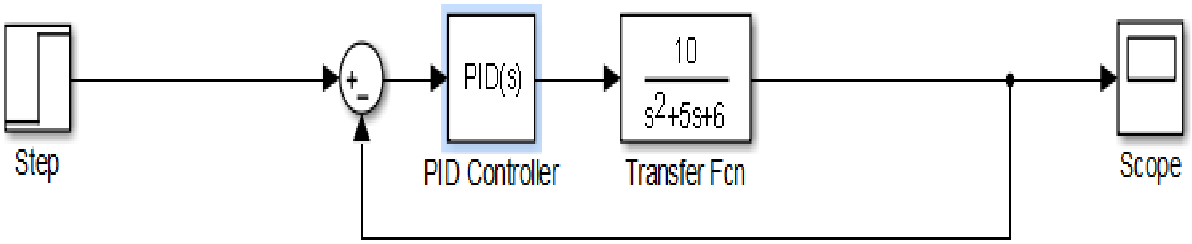

Figure 5: Simulink design for the closed loop transfer function

ii. run the simulation

iii. open the tune panel and increase the system response and increase the system robustness to highest

The result of the tuning is shown in Figure 6.

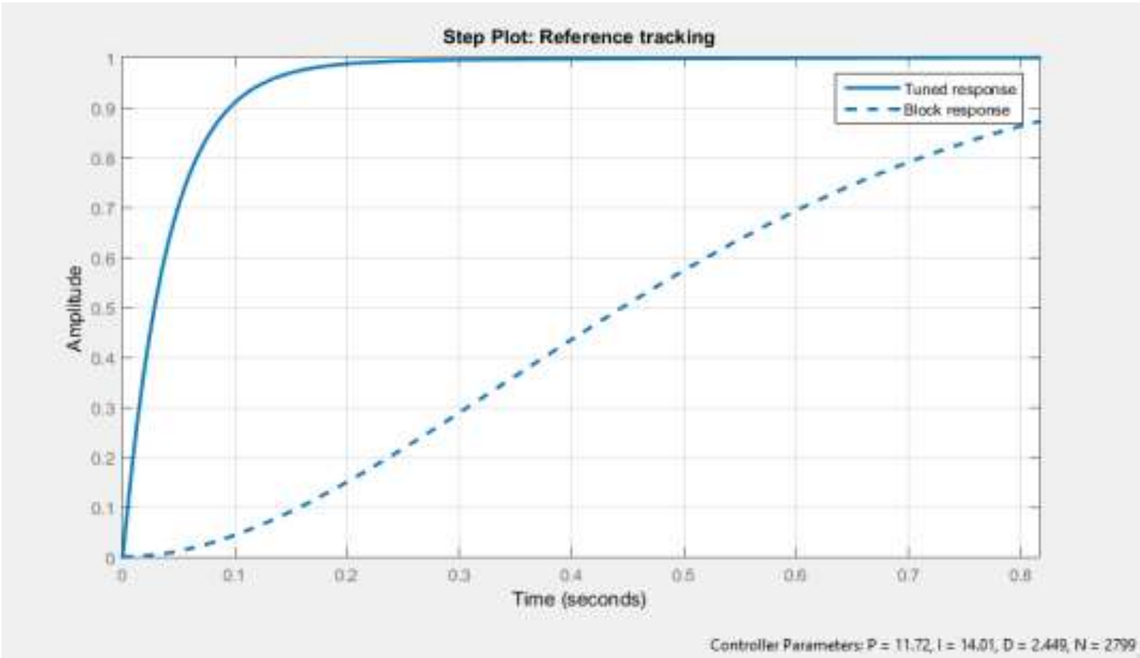

Figure 6: Tuned Response with Auto-tuning

iv. The values obtained for the system is Kp=11.72, Ki=14.01, Kd=2.449

v. At this value, the settling time reduced to 0.172s while the rise time reduced to 0.0909s.The system is very stable.

Comparing the tuned system to the initial transfer function response, Figure 7 was generated using MATLAB M-Codes

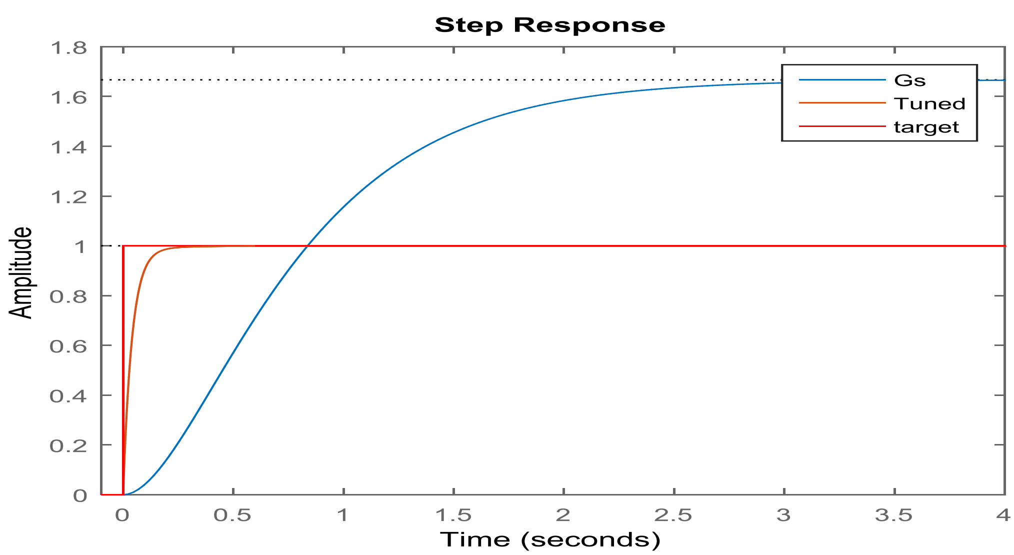

Figure 7: Tuned Response with Auto-tuning compared to

To compare the results from the Ziegler-Nichols closed-loop tuning method with the auto tuning system, a MATLAB m-file was written. The codes written for the comparism are given below:

```
% To obtain the Unit-Step Response of the System

num=[0 0 10]; % num_sys is the numerator of the system transfer function

den=[1 5 6]; % den_sys is the denominator of the system transfer function
% specify proportional and integrator controllers
Kp=11.72;
Ki=14.01;
Kd=2.449;

%To combine the P-I controllers
K=pid(Kp,Ki,Kd);

% To obtain the transfer function

s=tf('s');
Gs=tf(num,den); %Gs is the system transfer function

%To plot the step response
figure(1) %specifies figure number
step(Gs, feedback (Gs*K,1))%to plot
%all feedback functions
hold on
plot([-0.1,0,0,4],[0,0,1,1],'r');
axis([-0.1 4 0 1.8]);

legend('Gs','Tuned', 'target');

%specify title and grid
grid on
title ('Unit-Step Response of the system')
```

The result of the simulation is given in Figure 8

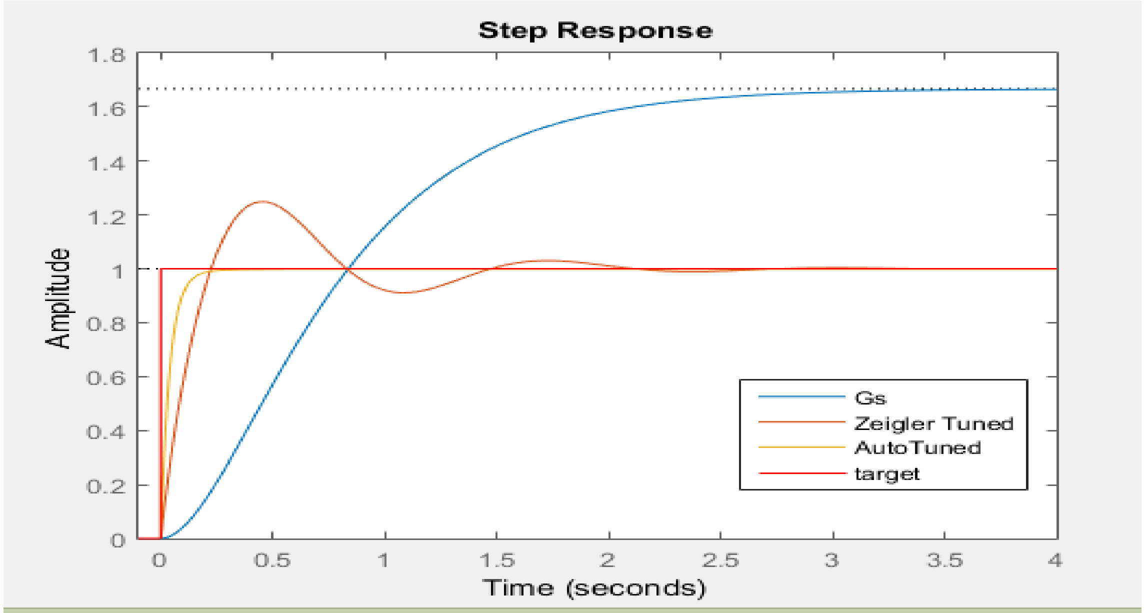

Figure 8: Comparism of Ziegler-Nichols tuned system with Auto-tuned system

The results from both tuning methods are tabulated in Table 4

Table 4: Table showing the Results for both Tuning Methods

	Ziegler-Nichols	Auto tuning
P	Kp=5.1	Kp=11.72
I	Ki=24.28	Ki=14.01
D	Kd=0.74375	Kd=2.449

Conclusion

From the graph in Figure 8, it can be observed that both tuning methods removed the steady state errors within the system, thereby indicating stability of the close loop system. It is however easy to move the system in any direction with the auto tuning method than the Ziegler method, as the Ziegler method is a quick approximation of results.

Limitations of PID control

Although PID controllers can be used in many control situations with satisfactory performance, their performance in other applications may be relatively poor with no optimal performance. PID controllers, when used in cases that are non-linear may be unable to respond to the fluctuations in process behaviour and may ultimately lag in their response to large disturbances. To solve such discrepancies, a knowledge of the control system can help include a feedforward control allowing the PID controller to only deal with the steady state error (Foley et al., 2005). Another serious challenge with the use of PID controllers is that they are linear and symmetric. Their performance in non-linear systems is unpredictable. This means that overshoot cannot easily be corrected like in linear systems. In trying to reduce overshoot in non-linear systems, the performance of the system may be compromised. Another issue with PID systems is that the derivative term can amplify high frequencies in a system and for non-linear systems, any large frequencies may be amplified to cause large discrepancies in the output of the system.

References:

[1] J. D. Cooper, "Car Cruise Control Loop Block Diagram," *online*, 2007.

[2] M. W. Foley, R. H. Julien, and B. R. Copeland, "A comparison of PID controller tuning methods," *The Canadian Journal of Chemical Engineering*, vol. 83, no. 4, pp. 712-722, 2005.

[3] R. P. Hunter, *Automated process control systems: concepts and hardware*. Prentice Hall PTR, 1987.

[4] K. Krishnaswamy, *Process control*. New Age International, 2007.

[5] B. Messner and D. Tilbury, "Control Tutorials for MATLAB and Simulink - Introduction: PID Controller Design," 2015.

[6] E. Abbasi, M. Mahjoob, and R. Yazdanpanah, "Controlling of quadrotor uav using a fuzzy system for tuning the pid gains in hovering mode," in *10th Int. Conf. Adv. Comput. Entertain. Technol*, 2013, pp. 1-6.

[7] R. Yazdanpanah, M. Mahjoob, and E. Abbasi, "Fuzzy LQR controller for heading control of an unmanned surface vessel," in *International Conference in Electrical and Electronics Engineering*, 2013, pp. 73-78.

[8] T. Nguyen, "Simulink TUTORIAL," 2015.

[9] K. Ogata and Y. Yang, "Modern control engineering," 1970.

[10] O. Ikechukwu, I. Aniekan, P. Ebunilo, and E. Ikpe, "Determination of Close Loop System Stability in Automobile Adaptive Cruise Control Systems."

[11] F. Padula and A. Visioli, "Tuning rules for optimal PID and fractional-order PID controllers," *Journal of process control*, vol. 21, no. 1, pp. 69-81, 2011.

[12] A. Taeib and A. Chaari, "Tuning optimal PID controller," *International Journal of Modelling, Identification and Control*, vol. 23, no. 2, pp. 140-147, 2015.

[13] A. Zilouchian and M. Jamshidi, *Intelligent control systems using soft computing methodologies*. CRC Press, Inc., 2000.

[14] P. Meshram and R. G. Kanojiya, "Tuning of PID controller using Ziegler-Nichols method for speed control of DC motor," in *Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on*, 2012, pp. 117-122: IEEE.

[15] P. Cominos and N. Munro, "PID controllers: recent tuning methods and design to specification," *IEE Proceedings-Control Theory and Applications*, vol. 149, no. 1, pp. 46-53, 2002.

[16] A. Visioli, "Fuzzy logic based set-point weight tuning of PID controllers," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: systems and humans*, vol. 29, no. 6, pp. 587-592, 1999.

[17] D. Vrančić, Y. Peng, and S. Strmčnik, "A new PID controller tuning method based on multiple integrations," *Control Engineering Practice*, vol. 7, no. 5, pp. 623-633, 1999.