CHEMICAL ENGINEERING DEPARTMENT

4/24/2023

Control System

P, PD, PI, PID CONTROLLERS

GROUP MEMBERS:- Ayushi Mohanty, Manisha Kumari Patra, Nitya Kasera
COLLEGE NAME:- Indira Gandhi Institute of Technology & Indore Institute of Technology

Table of Contents

Introduction:

This report is written to analyze the recitation that was presented on 02.04.2013. The recitation was presented by Sena TEMEL, Semih YAĞLI and Semih GÖREN. It was mainly about P, P-D, P-I and P-I-D controllers, their digital versus continuous time realizations and their characteristics including sampling period effects on the response of digital ones. Moreover, position and velocity form of P-I-D control was modeled on the 'Gate' project. Apart from these topics, P-I-D tuning methods such as manual tuning, Ziegler-Nichols tuning, Cohen-Coon tuning and MATLAB tuning method were discussed. Transient performances of P, P-D, P-I and P-I-D controllers were explained in detail. Modeling a discrete time P-I-D controller to control a continuous time plant was explained over a MATLAB code introducing the effect of sampling time and the choice of s*-domain to z-domain transformation method on MATLAB. It was explained how to remove poles that cause instability in discrete time by adding a new pole. Finally, it was shown how one could control the speed and position of the vehicle using discrete time P-I-D controller on the 'Gate' project.

Aim of the Recitation:

Aim of the recitation was to introduce the concept of Discrete Time P-I-D controllers and how they can be implemented on real life projects.

It was first intended to explain the usage of continuous time P-I-D controllers. In the first part of the recitation, it was aimed to show the how P, P-I, P-I-D controllers change the steady state response of the closed loop systems. Moreover, the methods to tune P-I-D controllers were introduced. It was meant to show that how hard it could get to properly tune a P-I-D controller. Secondly, it was intended to show how P, P-D, P-I, and P-I-D controllers affect the transient response of the closed loop system. It was meant to show how one can gain a feature but lose the other. Thirdly, it was intended to show how one should estimate the dynamics of the continuous time plant and use proper sampling time for discrete time P-I-D controller. It was also meant to show how changing transformation method may cause different pole locations on the z-plane. Lastly, it was intended to show how one could control the velocity and the position of the vehicle of the 'Gate' project by implementing a discrete time P-I-D controller in that project.

P Controller:

P controller is mostly used in first order processes with single energy storage to stabilize the unstable process. The main usage of the P controller is to decrease the steady state error of the system. As the proportional gain factor K increases, the steady state error of the system decreases. However, despite the reduction, P control can never manage to eliminate the steady state error of the system. As we increase the proportional gain, it provides smaller amplitude and phase margin, faster dynamics satisfying wider frequency band and larger sensitivity to the noise. We can use this controller only when our system is tolerable to a constant steady state error. In addition, it can be easily concluded that applying P controller decreases the rise time and after a certain value of reduction on the steady state error, increasing K only leads to overshoot of the system response. P control also causes oscillation if sufficiently aggressive in the presence of lags and/or dead time. The more lags (higher order), the more problem it leads. Plus, it directly amplifies process noise.

P-I Controller:

P-I controller is mainly used to eliminate the steady state error resulting from P controller. However, in terms of the speed of the response and overall stability of the system, it has a negative impact. This controller is mostly used in areas where speed of the system is not an issue. Since P-I controller has no ability to predict the future errors of the system it cannot decrease the rise time and eliminate the oscillations. If applied, any amount of I guarantees set point overshoot.

P-D Controller:

The aim of using P-D controller is to increase the stability of the system by improving control since it has an ability to predict the future error of the system response. In order to avoid effects of the sudden change in the value of the error signal, the derivative is taken from the output response of the system variable instead of the error signal. Therefore, D mode is designed to be proportional to the change of the output variable to prevent the sudden changes occurring in the control output resulting from sudden changes in the error signal. In addition D directly amplifies process noise therefore D-only control is not used.

P-I-D Controller:

P-I-D controller has the optimum control dynamics including zero steady state error, fast response (short rise time), no oscillations and higher stability. The necessity of using a derivative gain component in addition to the PI controller is to eliminate the overshoot and the oscillations occurring in the output response of the system. One of the main advantages of the P-I-D controller is that it can be used with higher order processes including more than single energy storage.
In order to observe the basic impacts, described above, of the proportional, integrative and derivative gain to the system response, see the simulations below prepared on MATLAB in continuous time with a transfer function $1/S^2+10S+20$ unit step input. The results will lead to turning methods.

Simulations and Results to Find the Constraints on Loop Tuning:

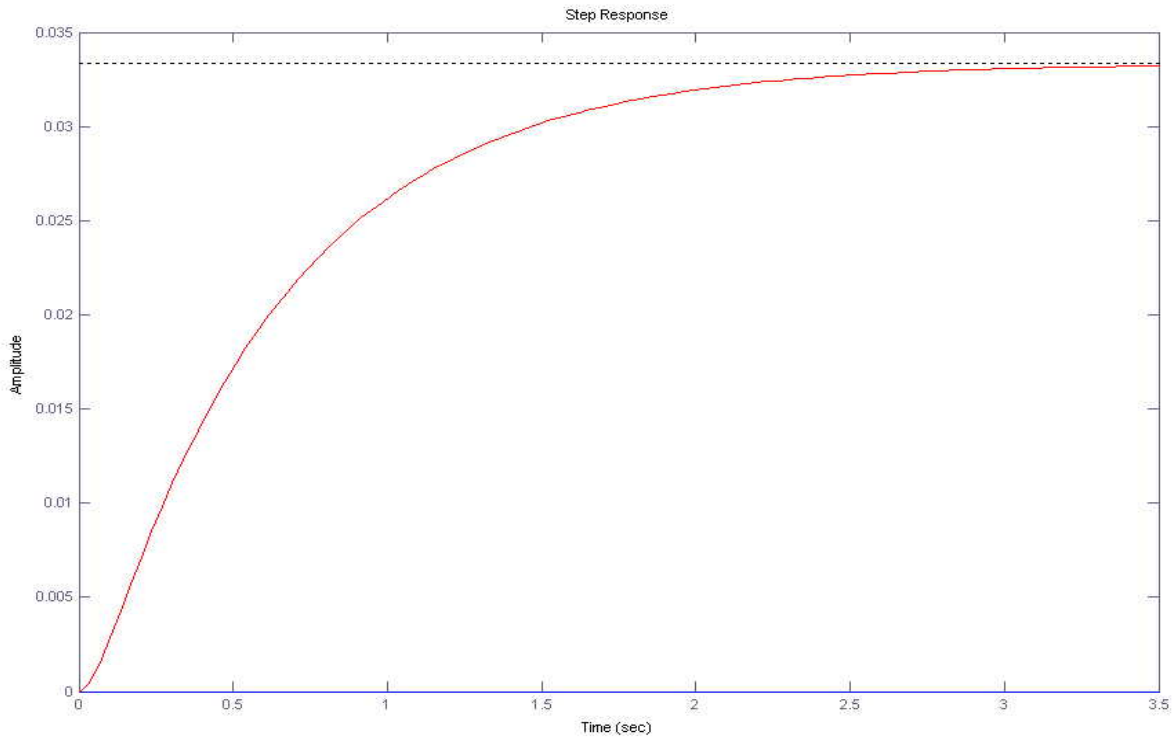

Figure 1: Step response without any controller

Result:

Steady state error= e_{ss}= 0.9625(too high)

Rise Time= t_r =~ 3 second

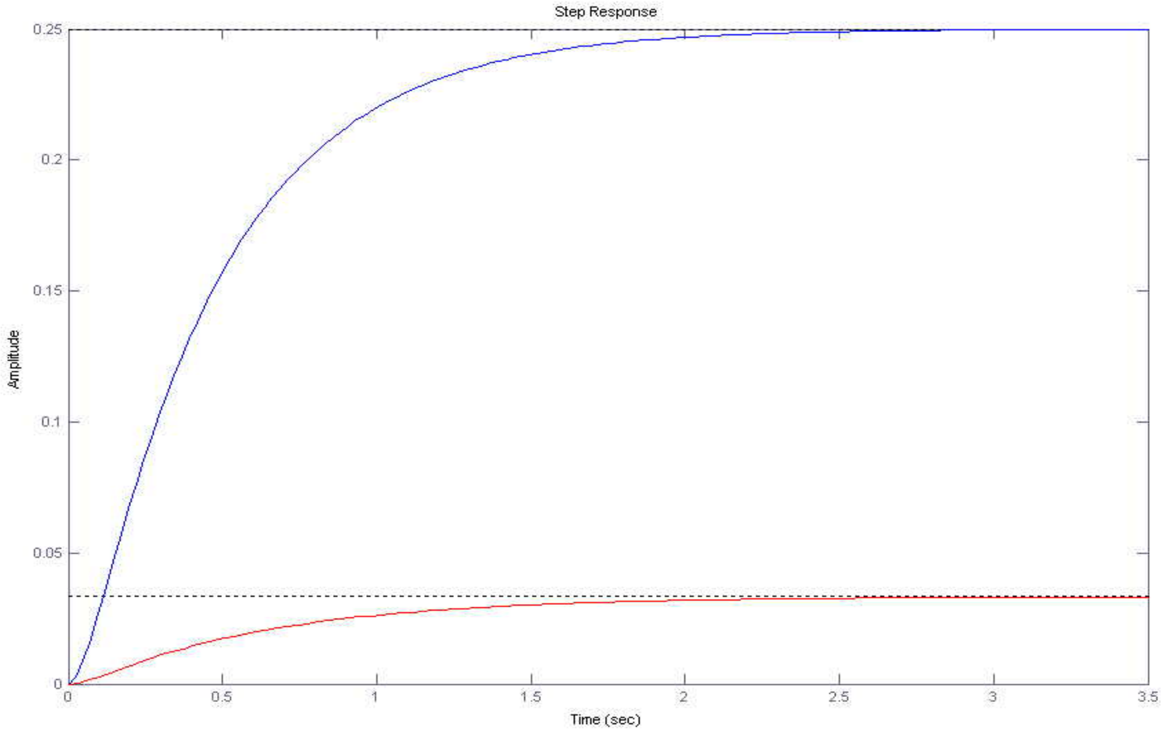

Figure 2: Step response with P controller, Kp = 10, Ki = 0, Kd = 0

Results:

Output Response improved

Rise time decreased

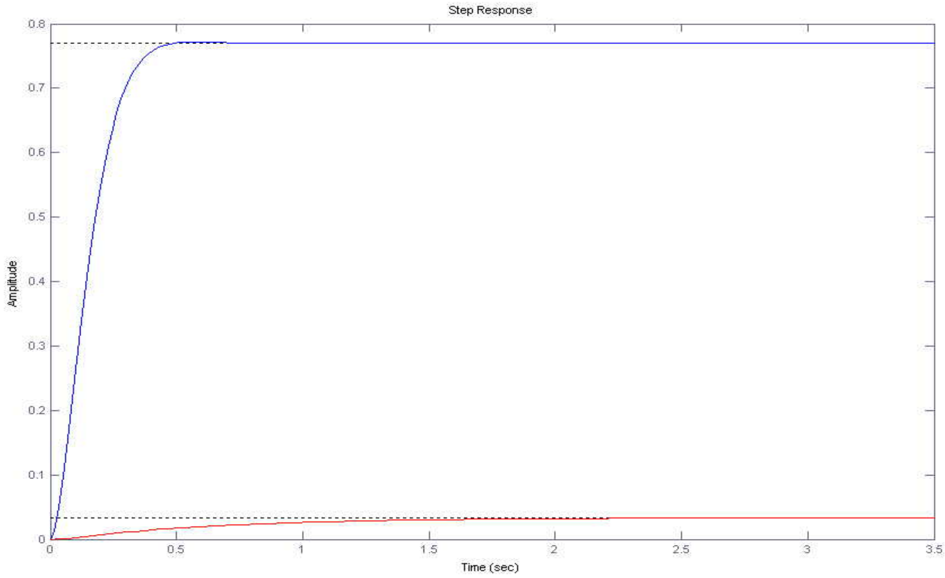

Figure 3: Step response with P controller, Kp = 100, Ki = 0, Kd = 0

Results:

Steady State Error = e_{ss}=0.23(decreased)

Rise Time Decreased

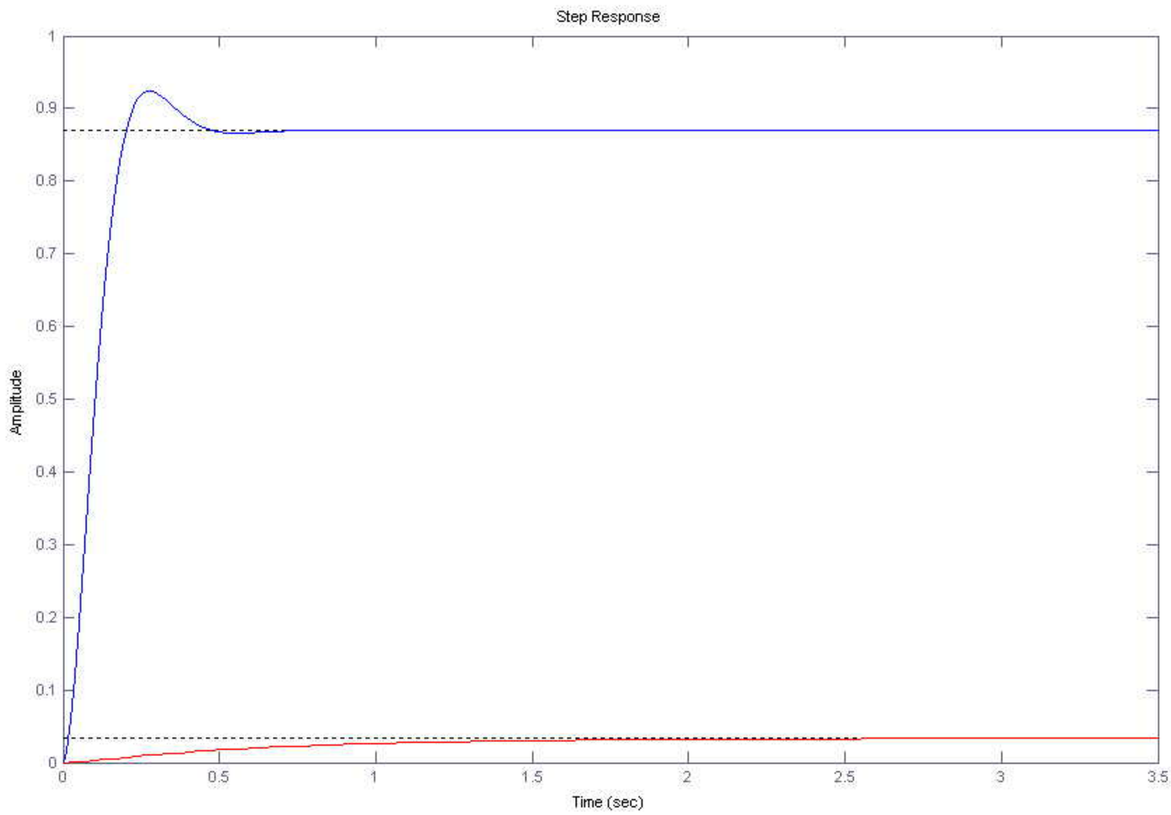

Figure 4: Step response with P controller, Kp = 200, Ki = 0, Kd = 0

Results:

Steady State Error = $\mathbf{e_{ss}}$ = 0.13(decreased)

Rise Time = $\mathbf{t_r}$ = 0.5seconds

Overshoot occurs at the output response

Conclusions:

- Increasing K_p will reduce the steady state error.
- After certain limit increasing K_p only causes overshoot.
- Increasing K_p reduces the rise time.

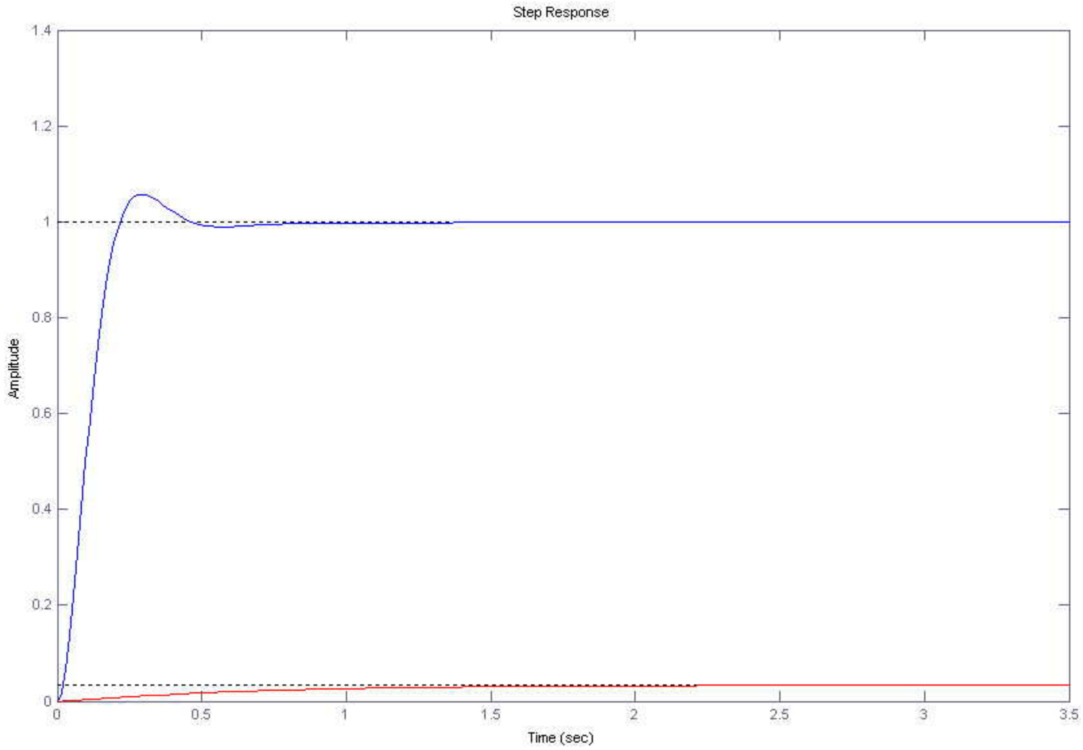

Figure 5: Step response with P-I controller, Kp = 200, Ki = 100, Kd = 0

Results:

Steady State Error = $\mathbf{e_{ss}}$ = 0(eliminated)

Rise Time = $\mathbf{t_r}$ = 0.3 seconds

Setting Time = $\mathbf{t_s}$ = 0.7 seconds

Overshoot Remains

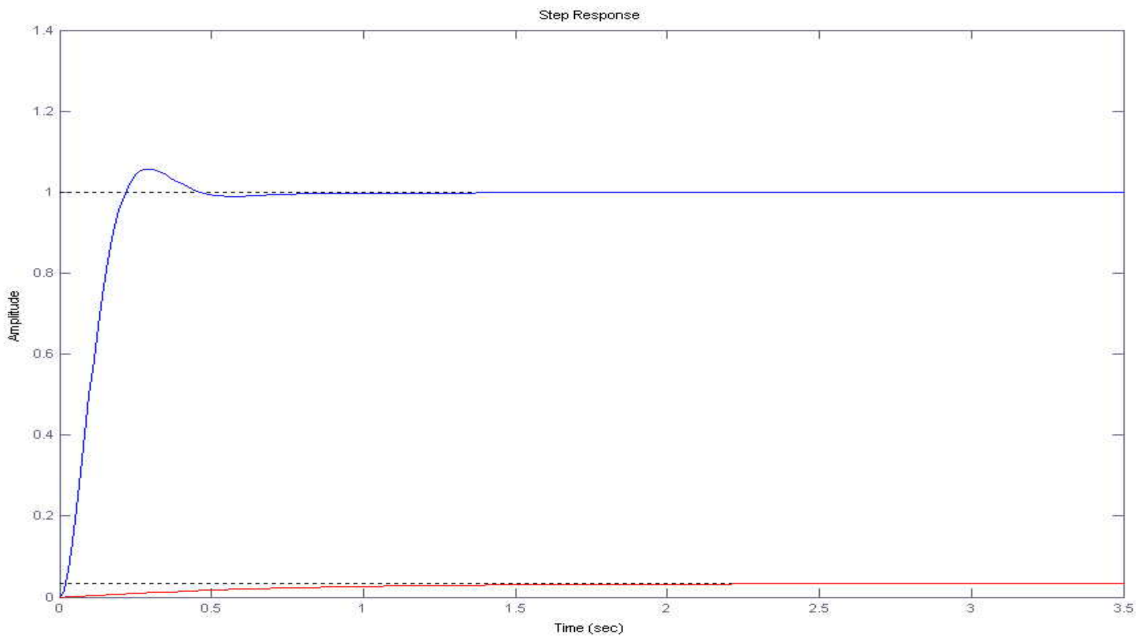

Figure 6: Step response with P-I controller, Kp = 200, Ki = 200, Kd = 0

Results:

Steady State Error = $\mathbf{e_{ss}} = 0$

Rise Time = $\mathbf{t_r}$ = 0.3 seconds (no significant change)

Settling Time = $\mathbf{t_s}$ = 0.7 seconds (no significant change)

Overshoot Remains

Conclusions:

- Integral control eliminates the steady state error.
- After certain limit, increasing Ki will only increase overshoot.
- Increasing Ki reduces the rise time a little.

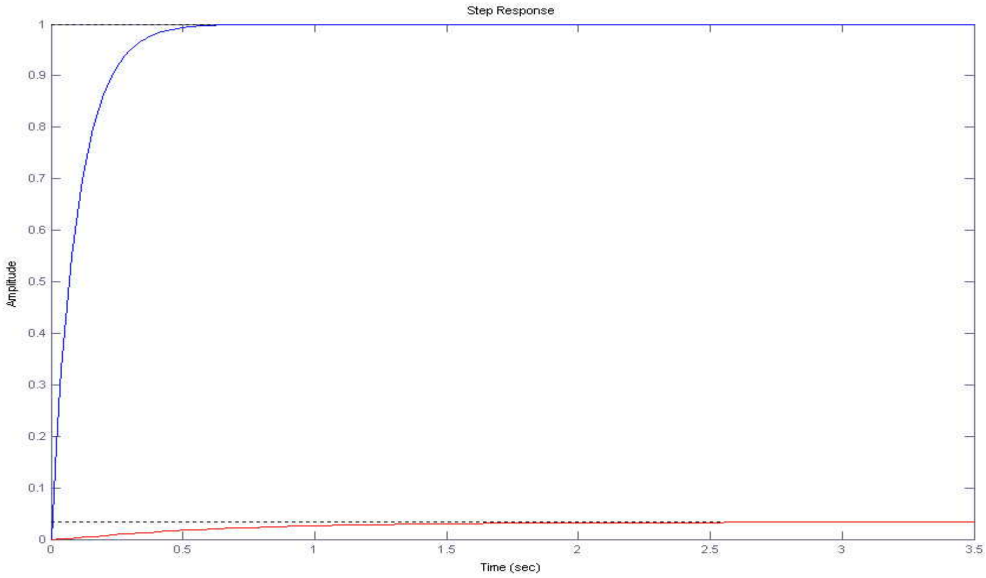

Figure 7: Step response with P-I-D controller, Kp = 200, Ki = 200 and Kd = 10

Results:

Steady State Error = $\mathbf{e_{ss}} = 0$

Rise Time = $\mathbf{t_r}$ = 0.3 seconds (no significant change)

Settling Time = $\mathbf{t_s}$ = 0.7 seconds (no significant change)

Overshoot Remains

Conclusions:

- Increasing K_d decreases the overshoot.
- Increasing K_d reduces the settling time.

Loop Tuning:

Tuning a control loop is arranging the control parameters to their optimum values in order to obtain desired control response. At this point, stability is the main necessity, but beyond that, different systems leads to different behaviors and requirements and these might not be compatible with each other. In principle, P-I-D tuning seems completely easy, consisting of only 3 parameters, however, in practice; it is a difficult problem because the complex criteria at the P-I-D limit should be satisfied. P-I-D tuning is mostly a heuristic concept but existence of many objectives to be met such as short transient, high stability makes this process harder. For example sometimes, systems might have nonlinearity problem which means that while the parameters works properly for full load conditions, they might not work as effective for no load conditions. Also, if the P-I-D parameters are chosen wrong, control process input might be unstable, with or without oscillation; output diverges until it reaches to saturation or mechanical breakage.

For a system to operate properly, the output should be stable, and the process should not oscillate in any condition of set point or disturbance. However, for some cases bounded oscillation condition as a marginal stability can be accepted.

As an optimum behavior, a process should satisfy the regulation and command breaking requirements. These two properties define how accurately a controlled variable reaches the desired values. The most important characteristics for command breaking are rise time and settling time. For some systems where overshoot is not acceptable, to achieve the optimum behavior requires eliminating the overshoot completely and minimizing the dissipated power in order to reach a new set point.

In today's control engineering world, P-I-D is used over %95 of the control loops. Actually if there is control, there is P-I-D, in analog or digital forms. In order to achieve optimum solutions Kp, Ki and Kd gains are arranged according to the system characteristics. There are many tuning methods, but most common methods are as follows:

- Manual Tuning Method
- Ziegler-Nichols Tuning Method
- Cohen-Coon Tuning Method
- PID Tuning Software Methods (ex. MATLAB)

Manual Tuning Method:

Manual tuning is achieved by arranging the parameters according to the system response. Until the desired system response is obtained Ki, Kp and Kd are changed by observing system behavior. Example (for no system oscillation): First lower the derivative and integral value to 0 and raise the proportional value 100. Then increase the integral value to 100 and slowly lower the integral value and observe the system's response. Since the system will be maintained around set point, change set point and verify if system corrects in an acceptable amount of time. If not acceptable or for a quick response, continue lowering the integral value. If the system begins to oscillate again, record the integral value and raise value to 100. After raising the integral value to 100, return to the proportional value and raise this value until oscillation ceases. Finally, lower the proportional value back to 100.0 and then lower the integral value slowly to a value that is 10% to 20% higher than the recorded value when oscillation started (recorded value times 1.1 or 1.2).

Although manual tuning method seems simple it requires a lot of time and experience

Ziegler-Nichols Method:

More than six decades ago, P-I controllers were more widely used than P-I-D controllers. Despite the fact that P-I-D controller is faster and has no oscillation, it tends to be unstable in the condition of even small changes in the input set point or any disturbances to the process than P-I controllers. Ziegler-Nichols Method is one of the most effective methods that increase the usage of P-I-D controllers.

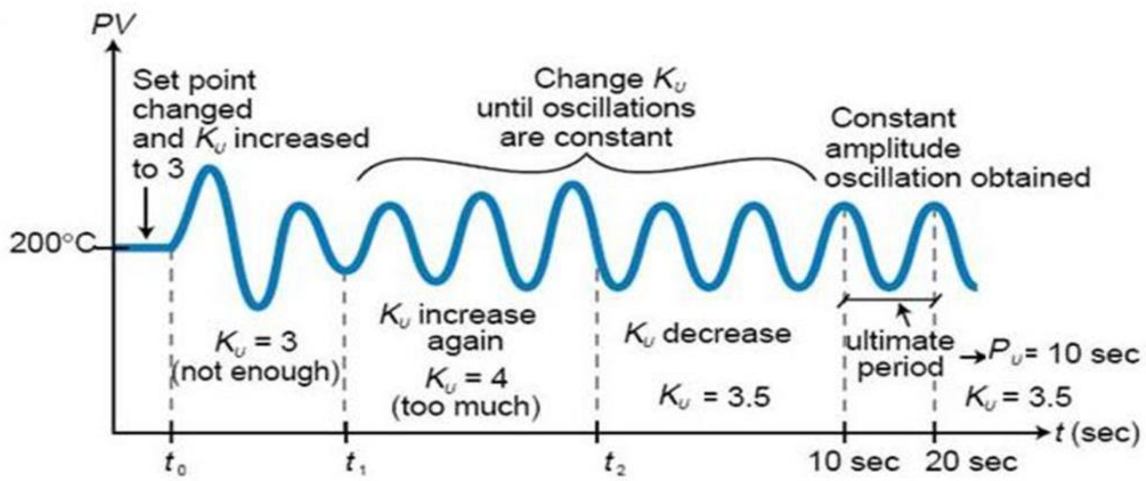

Figure 8: Ziegler-Nichols P-I-D controller tuning method

The logic comes from the neutral heuristic principle. Firstly, it is checked that whether the desired proportional control gain is positive or negative. For this, step input is manually increased a little, if the steady state output increases as well it is positive, otherwise; it is negative. Then, Ki and Kd are set to zero and only Kp value is increased until it creates a periodic oscillation at the output response. This critical Kp value is attained to be "ultimate gain", Kc and the period where the oscillation occurs is named as Pc "ultimate period". As a result, the whole process depends on two variables and the other control parameters are calculated according to the table in the Figure 9.

Ziegler–Nichols method giving *K'* values (loop times considered to be constant and equal to dT)			
Control Type	Kp	Ki'	Kd'
P	$0.50K_c$	0	0
PI	$0.45K_c$	$1.2K_p dT/P_c$	0
PID	$0.60K_c$	$2K_p dT / P_c$	$K_p P_c / (8dT)$

Figure 9: Ziegler-Nichols P-I-D controller tuning method, adjusting Kp, Ki and Kd

Advantages:

- ✓ It is an easy experiment; only need to change the P controller
- ✓ Includes dynamics of whole process, which gives a more accurate picture of how the System is behaving

Disadvantages:

- ➢ Experiment can be time consuming
- ➢ It can venture into unstable regions while testing the P controller, which could cause the System to became out of control
- ➢ For some cases it might result in aggressive gain and overshoot

Cohen-Coon Tuning Method:

This tuning method has been discovered almost after a decade than the Ziegler-Nichols method. Cohen-Coon tuning requires three parameters which are obtained from the reaction curve as in the Figure 10.

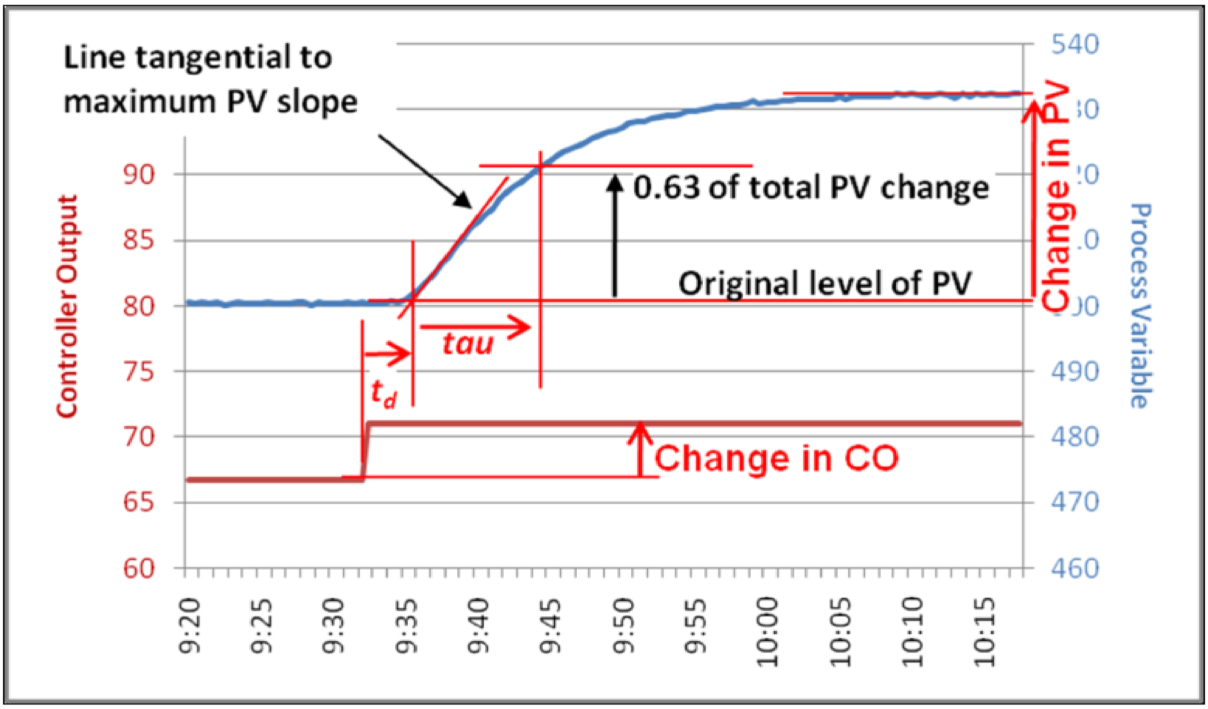

Figure 10: Cohen-Coon P-I-D Tuning Method

The controller is manually placed and after the process settled out a few percent of the change is made in the controller output (CO) and waited for the process variable (PV) to settle out at a new value. As observed from the graph, process gain (gp) is calculated as follow:

gp =pv/co (in %)

The maximum slope at the inflection point on the PV response curve is found and drawn a tangential line. td (dead time) is measured as taking the time difference between the change in CO and the intersection of the tangential line and the original PV level. As a final parameter τ (time constant) as the time difference between intersection at the end of the dead time and the PV reaching 63% of its total change. After converting the time variables into the same units and applying couple of tests until to find similar result, these three variables are used to define new control parameters using the table in the Figure 11 below

	Controller Gain	Integral Time	Derivative Time
P Controller:	$K_C = \frac{1.03}{g_p}\left(\frac{\tau}{t_d}+0.34\right)$		
PI Controller:	$K_C = \frac{0.9}{g_p}\left(\frac{\tau}{t_d}+0.092\right)$	$T_I = 3.33t_d \frac{\tau+0.092t_d}{\tau+2.22t_d}$	
PD Controller:	$K_C = \frac{1.24}{g_p}\left(\frac{\tau}{t_d}+0.129\right)$		$T_D = 0.27t_d \frac{\tau-0.324t_d}{\tau+0.129t_d}$
PID Controller: (Noninteracting)	$K_C = \frac{1.35}{g_p}\left(\frac{\tau}{t_d}+0.185\right)$	$T_I = 2.5t_d \frac{\tau+0.185t_d}{\tau+0.611t_d}$	$T_D = 0.37t_d \frac{\tau}{\tau+0.185t_d}$

Figure 11: Cohen-Coon P-I-D Tuning Method, adjusting Kp, Ki and Kd

Comparison of the two methods:

If we want to compare these two methods, Ziegler-Nichols can be used for any order of the systems, especially for the higher ones, while Cohen-Coon can only be used for first order systems. Therefore, Ziegler-Nichols tuning method is more widely used. However, for the first order systems Cohen-Coon is more flexible since as Ziegler-Nichols is only applicable when the dead time is less than of the time constant, Cohen-Coon is tolerable until of this value and it can be even extended. Therefore for systems having time delay this tuning method is more convenient. All in all, despite the fact that tuning a system seems easy to apply, in practice, it is really hard to analyze and pick a tuning method satisfying all system requirements. Using the logic of arranging the control parameters described above, some PID tuning software methods are developed which are easier to apply and saves time to get an optimum solution.

Transient Responses of P, P-D, P-I and P-I-D controllers:

In this part, transient performances of P, P-D, P-I and P-I-D controllers are explained. Their steady state error performances are also discussed.

- Transient Response of P Controller:

As a general rule, increasing proportional gain decreases the steady state error. However, the actual performance of P controller depends on the order of the plant.

If P controller is used to control a second order plant, it has following properties:

- Increasing gain decreases rise time (Advantage)
- Increasing gain increases percent overshoot and number of oscillations (Disadvantage)
- Increasing gain decreases steady state error (Advantage)
- Steady state is never zero if only-P type controller is used (Disadvantage)
- In order to have zero steady state error gain should be infinity (Physically impossible)

The discussion above shows that only-P control is not enough to control second order plants. In fact, only-P control is usually used to control first order plants, because there are no natural oscillations in first order plants and P control is easy to implement. The following simulations were don MATLAB-Simulink to illustrate the performance of P control on first and second order plants.

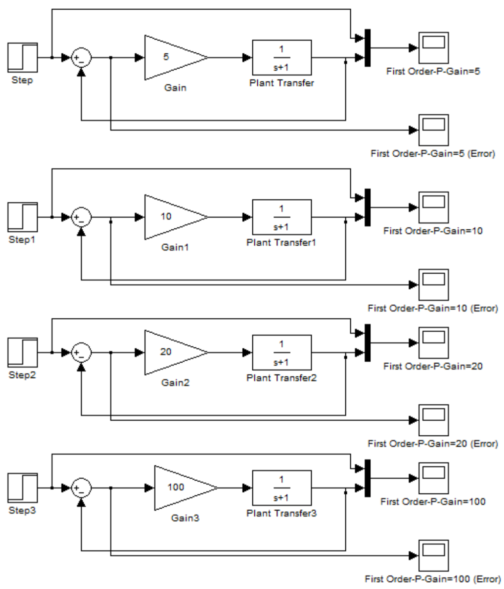

Figure 12: MATLAB-Simulink Diagram to show the effect of P control on first order plant

First order continuous plant transfer function:

$\mathbf{G}_p(s) = 1/s+1$

Input:

$x(t) = 5u(t)$

Controlled system outputs with for various $\mathbf{K_p}$

For $\mathbf{K_p}$=5

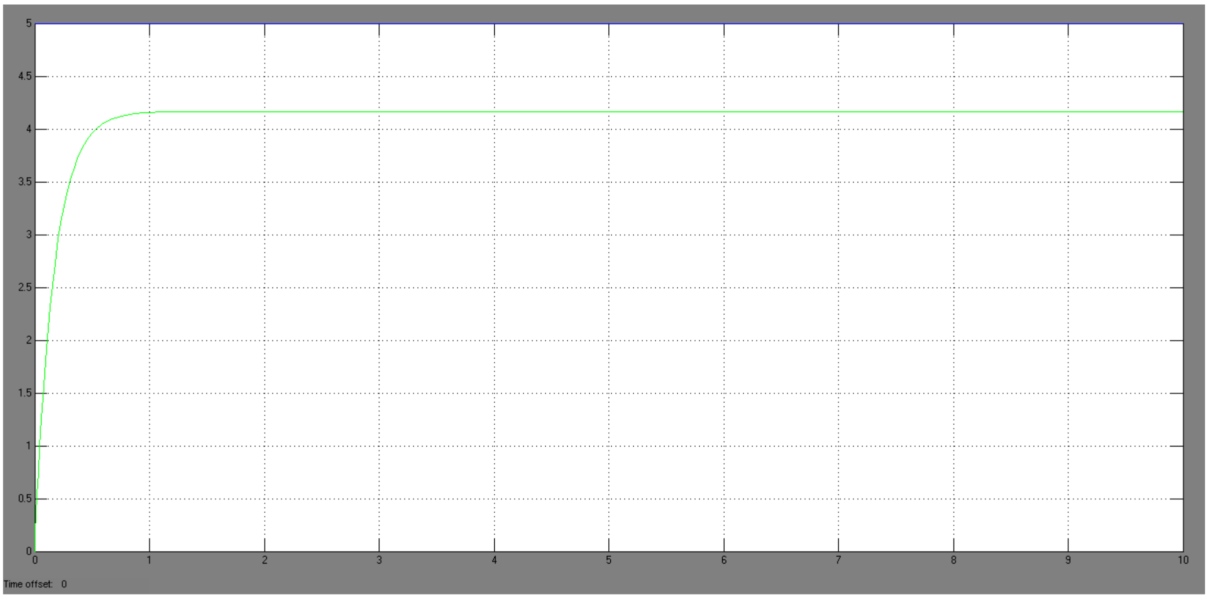

Figure 13: Output of the closed loop system with only P control, Kp = 5

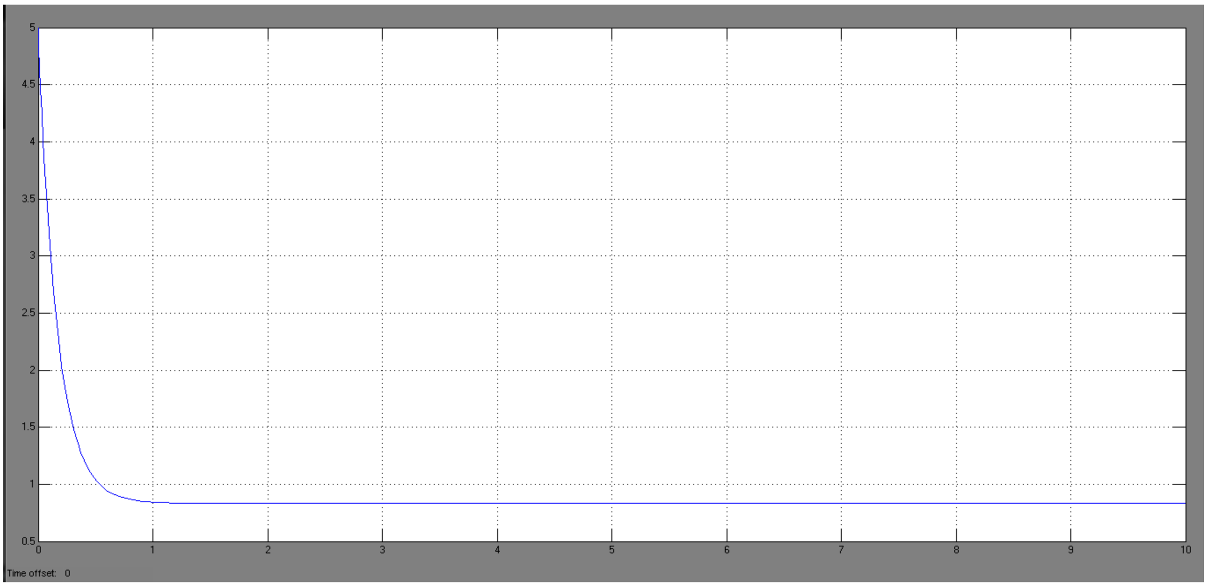

Figure 14: Error of the closed loop system with only P control, Kp=5

Steady State Error = $\mathbf{e_{ss}}$ = 0.84

Rise Time = $\mathbf{t_r}$ = 0.45 seconds

For $\mathbf{K_p}$ = 10

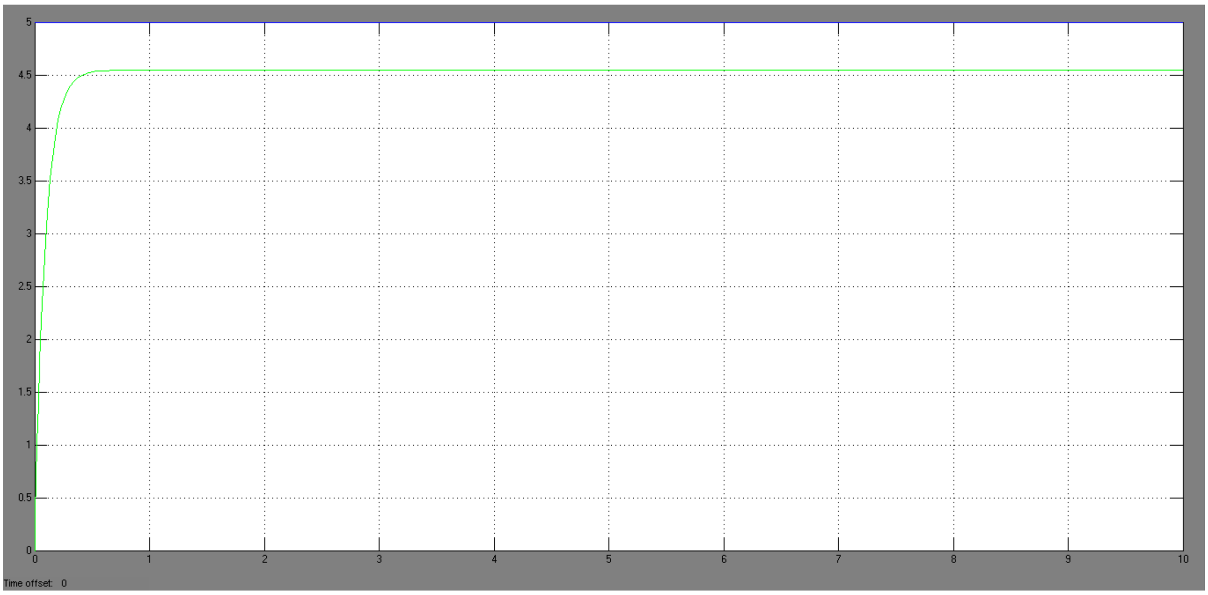

Figure 15: Output of the closed loop system with only P control, Kp=10

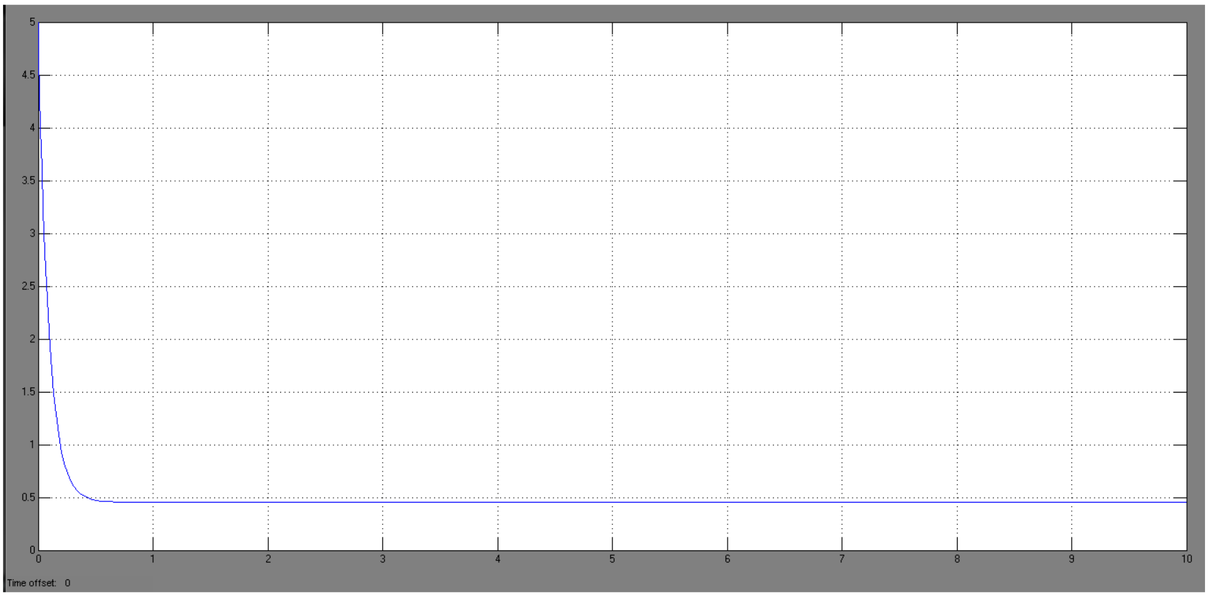

Figure 16: Error of the closed loop system with only P control, Kp=10

Steady State = $\mathbf{e_{ss}}$ =0.45

Rise Time = $\mathbf{t_r}$ = 0.4 seconds

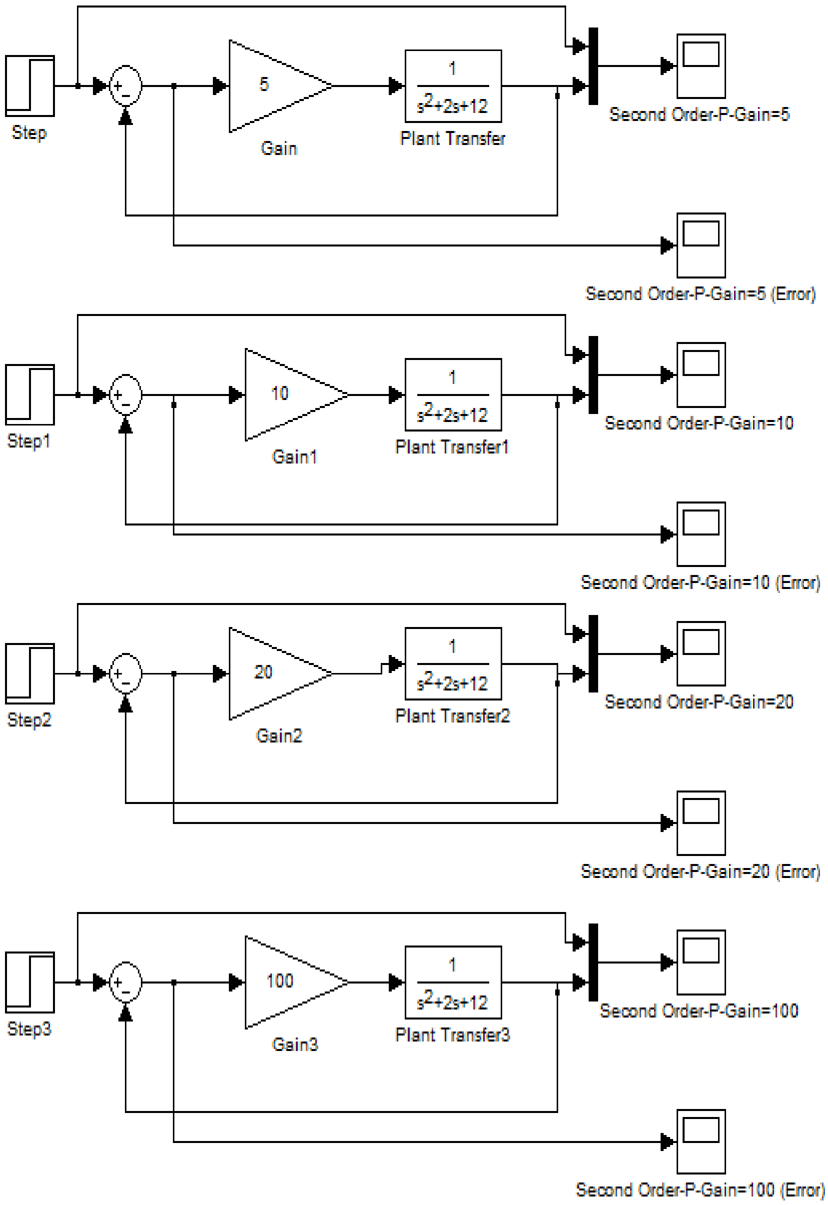

Figure 21: MATLAB-Simulink Diagram to show the effect of P control on second order plant

Second order continuous plant transfer function:

$\mathbf{G_p}(s) = 1/s^2+2s+12$

Input:

$x(t) = 5u(t)$

Controlled system outputs with for various $\mathbf{K_p}$

• Transient Response of P-D Controller:

Derivative action is usually used to improve transient response of the closed loop system. Only D control is not used because it amplifies high frequency noise which is never desired. Derivative action decreases rise time and oscillations. However, it does not have any effect on steady state performance of the closed loop.

The discussion above indicates that with P-D control, steady state error is still non-zero. Derivative control is usually used to decrease oscillations in closed loop system outputs. The following simulations were done on MATLAB-Simulink to illustrate the performance of P-D control on first and second order plants.

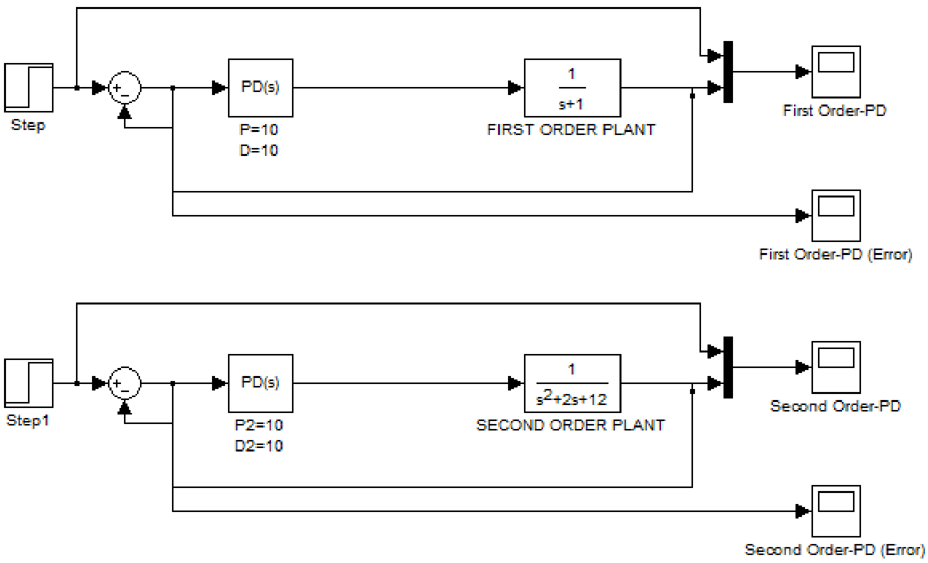

Figure 30: MATLAB-Simulink Diagram to show the effect of P-D control on first and second order plants

First Order Plant transfer function:

$\mathbf{G_p}(s) = 1/s+1$

Second order continuous plant transfer function:

$\mathbf{G_p}(s) = 1/s^2+2s+12$

Input:

$x(t) = 5u(t)$

Controlled system outputs with for various $\mathbf{K_p}$

Controlled first order system outputs:

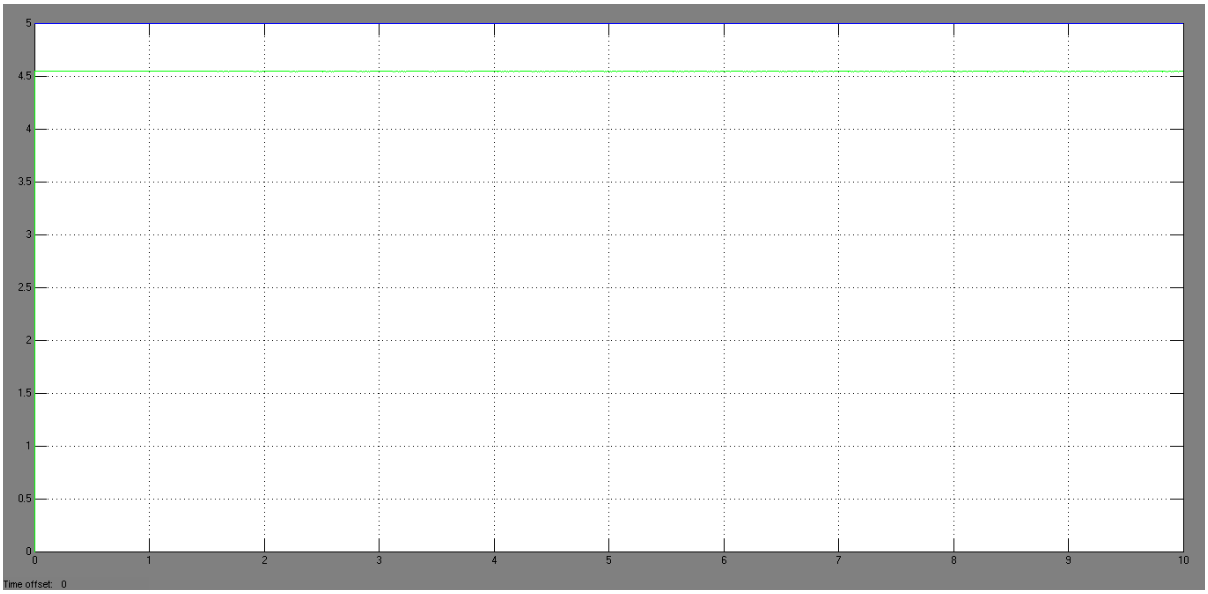

Figure 31: Output of the closed loop system (first order) with P-D control, Kp=10, Kd=10

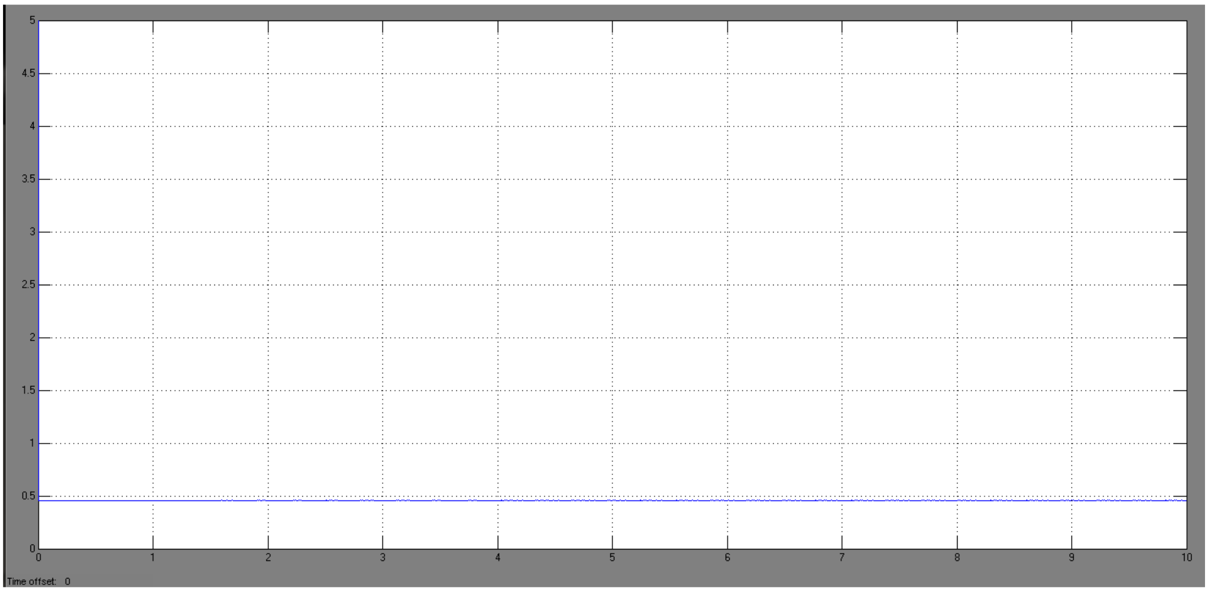

Figure 32: Error of the closed loop system (first order) with P-D control, Kp=10, Kd=10

Steady State = $\mathbf{e_{ss}}$ =0.45

Rise Time = $\mathbf{t_r}$ = 0.005 seconds

Controlled second order system outputs:

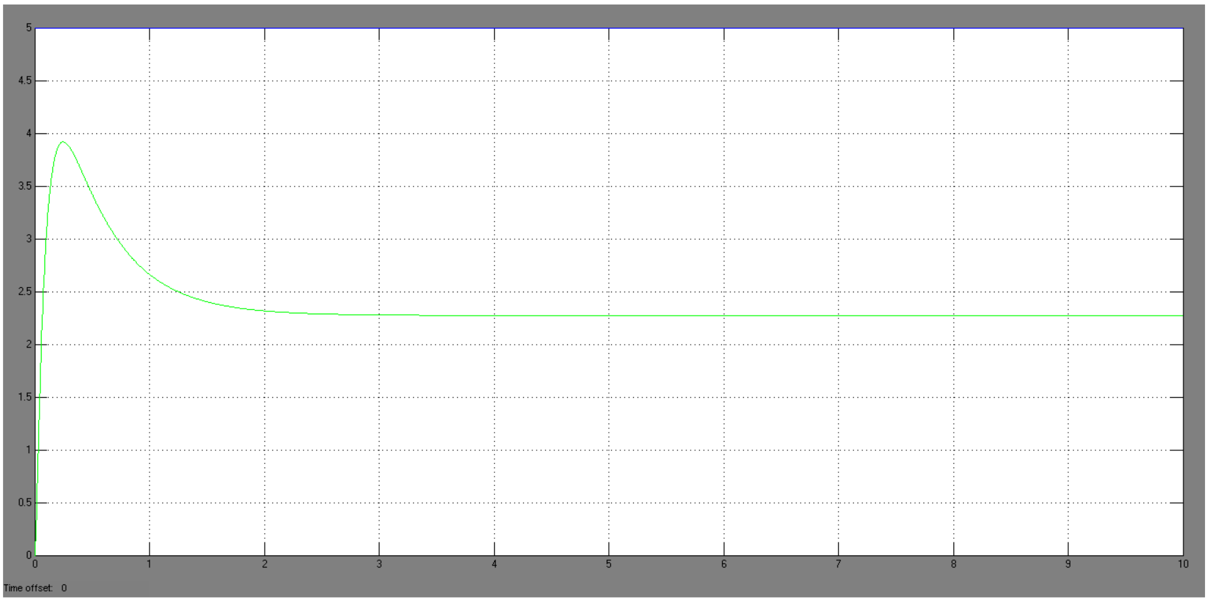

Figure 33: Output of the closed loop system (second order) with P-D control, Kp=10, Kd=10

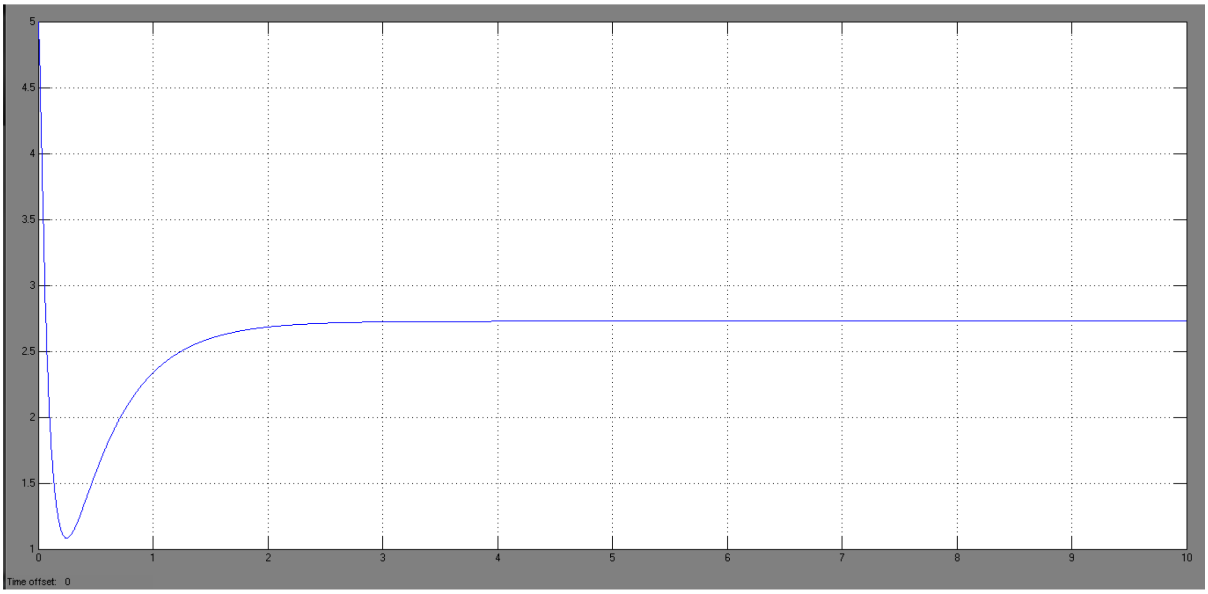

Figure 34: Error of the closed loop system (second order) with P-D control, Kp=10, Kd=10

Steady State = $\mathbf{e_{ss}}$ = 2.7

Rise Time = $\mathbf{t_r}$ = 0.07 seconds

Percentage overshoot = 69.5%

□ Transient Response of P-I Controller:

Integral action eliminates steady state error. However, it has very poor transient response. Using integral action increases the oscillations in the output of the closed loop systems.

The discussion above indicates that with P-I control, steady state error is non-zero. However, Integral control causes too many oscillations in closed loop system outputs. The following simulations were done on MATLAB-Simulink to illustrate the performance of P-I control on first and second order plants.

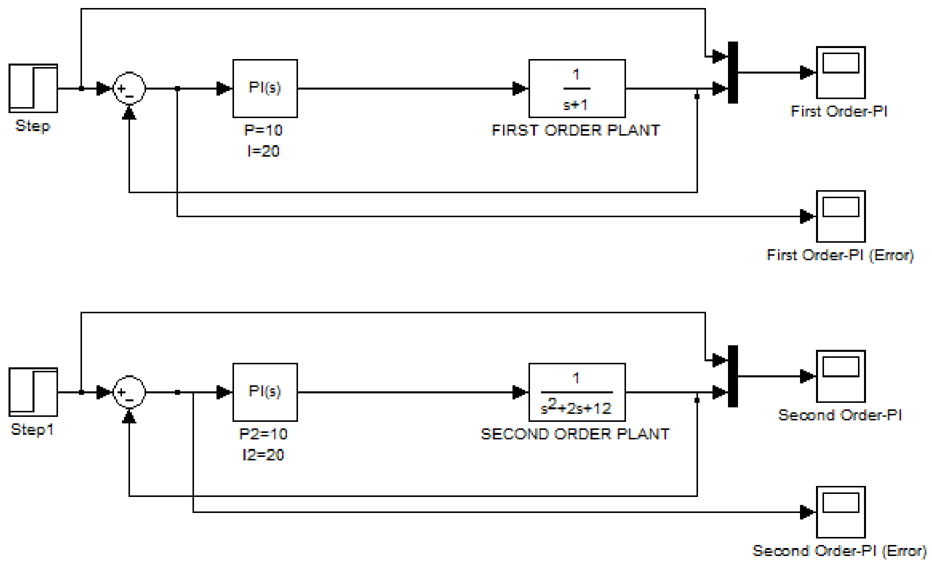

Figure 35: MATLAB-Simulink Diagram to show the effect of P-I control on first and second order plants

First Order Plant transfer function:

$\mathbf{G_p}(s) = 1/s+1$

Second order continuous plant transfer function:

$\mathbf{G_p}(s) = 1/s^2+2s+12$

Input:

$x(t) = 5u(t)$

Controlled system outputs with for various $\mathbf{K_p}$

Controlled first order system outputs:

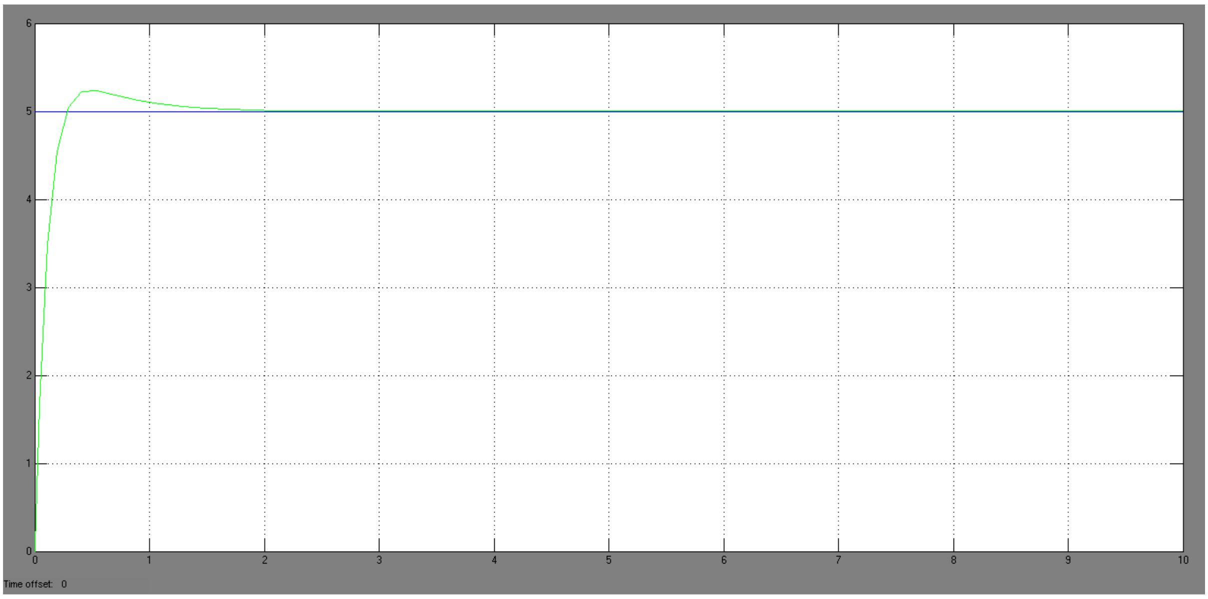

Figure 36: Output of the closed loop system (first order) with P-I control, Kp=10, Ki=20

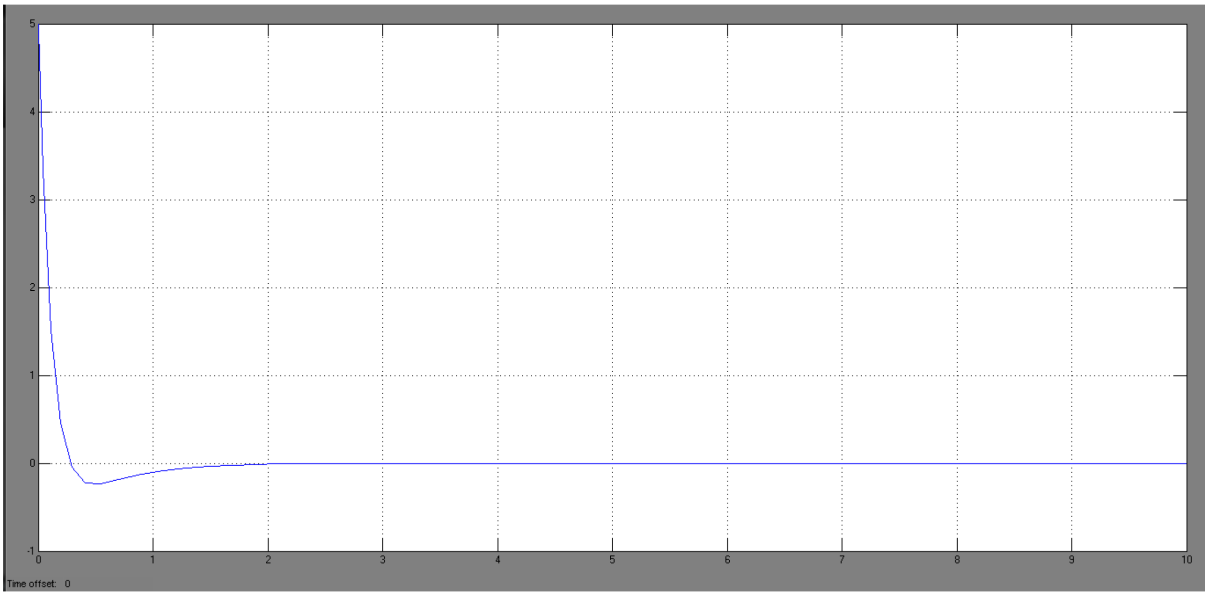

Figure 37: Error of the closed loop system (first order) with P-I control, Kp=10, Ki=20

Steady State = $\mathbf{e_{ss}}$ = 0

Rise Time = $\mathbf{t_r}$ = 0.28 seconds

2% Settling time = $\mathbf{t_s}$ =1.31 seconds

Percentage overshoot = 5%

Controlled second order system outputs:

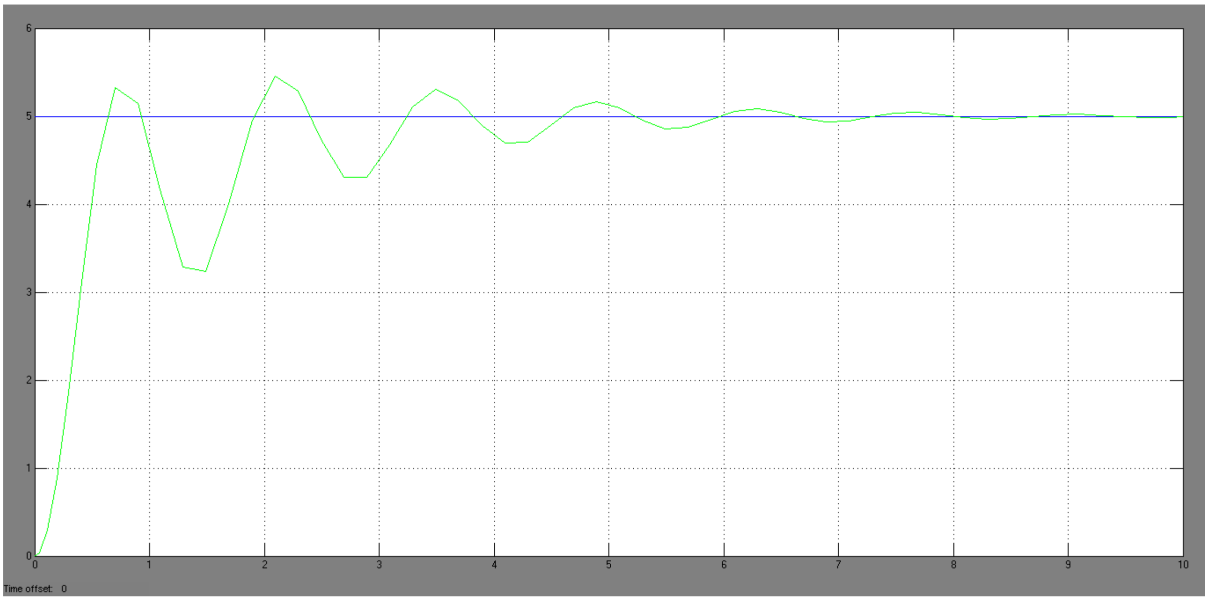

Figure 38: Output of the closed loop system (second order) with P-I control, Kp=10, Ki=20

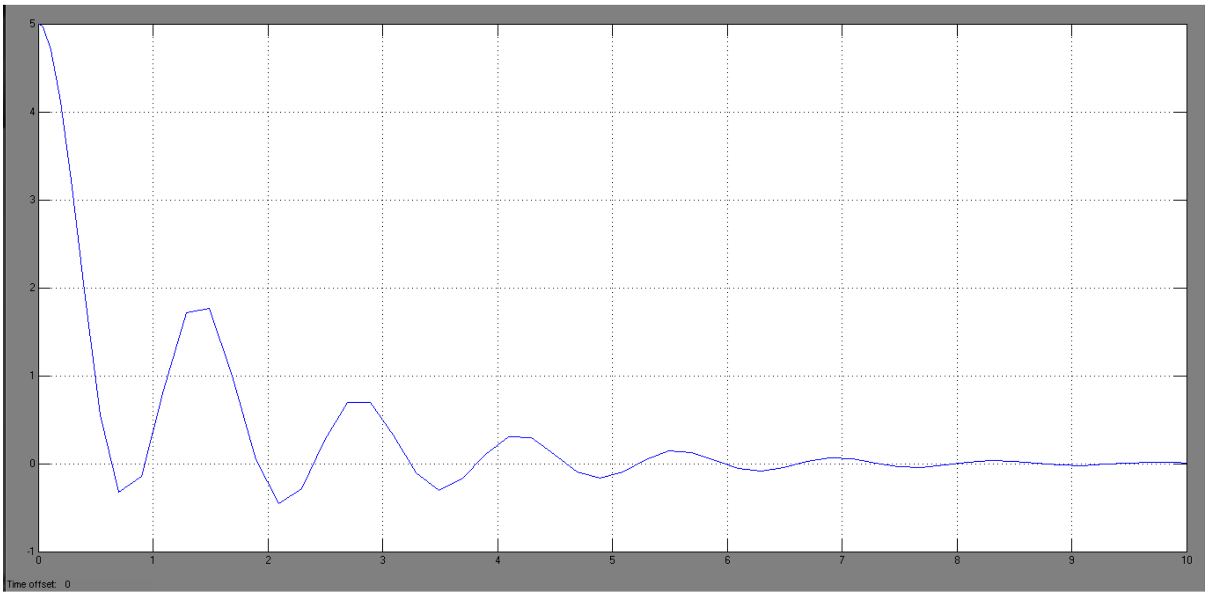

Figure 39: Error of the closed loop system (second order) with P-I control, Kp=10, Ki=20

Steady State = $\mathbf{e_{ss}}$ = 0

Rise Time = $\mathbf{t_r}$ = 0.65 seconds

2% Settling time = $\mathbf{t_s}$ =7.11 seconds

Percentage overshoot = 6%

Transient Response of P-I-D Controller:

P-I-D controller is the optimal controller for high order plants. It has zero steady state error together with acceptable transient response. The only problem with P-I-D control is tuning. Fortunately, MATLAB has automatic tuning option. However, automatic tuning does not usually provide the best results, it only provides optimal results. P-I-D tuning is an engineering art and should be manually done by control engineers.

The following simulations were done on MATLAB-Simulink to illustrate the performance of P-I-D control on first and second order plants.

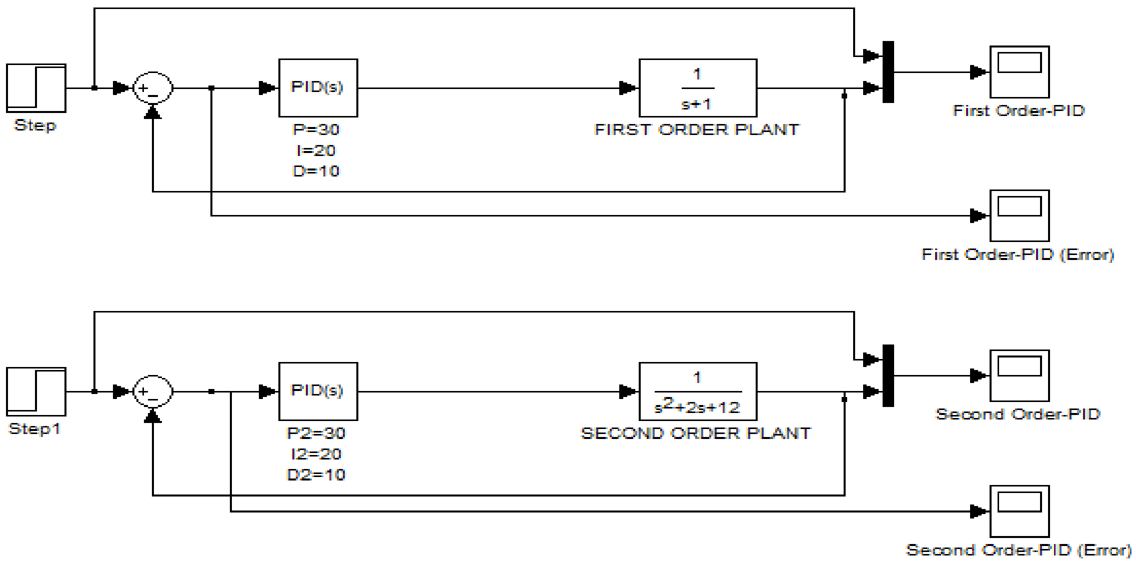

Figure 40: MATLAB-Simulink Diagram to show the effect of P-I-D control on first and second order plants

First Order Plant transfer function:

$$\mathbf{G_p}(s) = 1/s+1$$

Second order continuous plant transfer function:

$$\mathbf{G_p}(s) = 1/s^2+2s+12$$

Input:

$$x(t) = 5u(t)$$

Controlled first order system outputs:

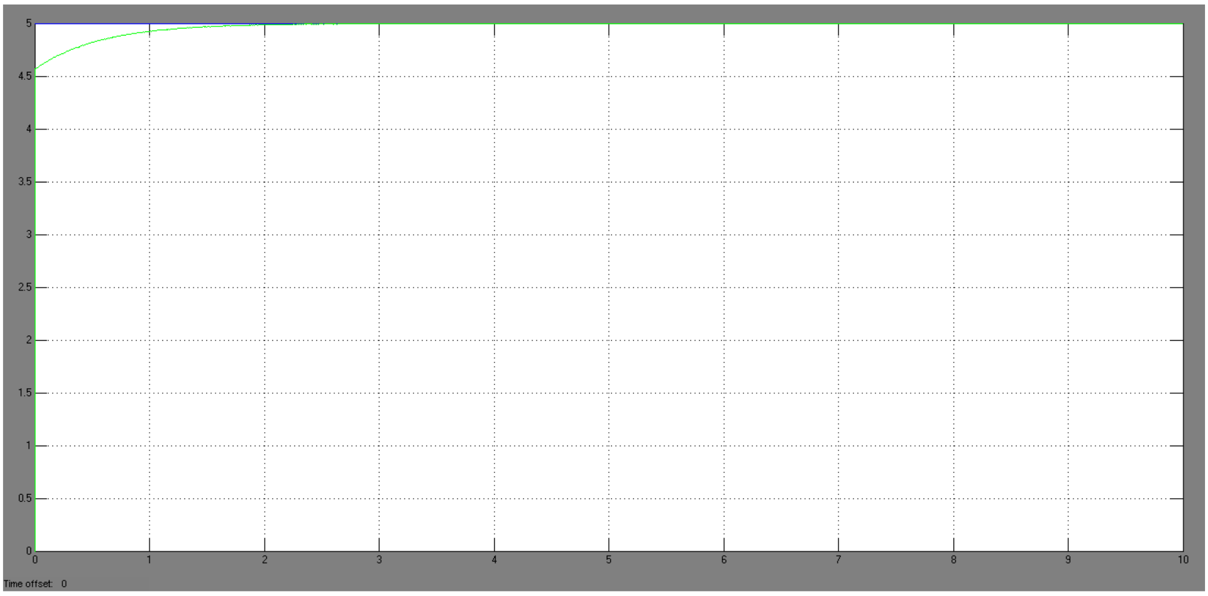

Figure 41: Output of the closed loop system (first order) with P-I-D control, Kp=30, Ki=20, Kd=10

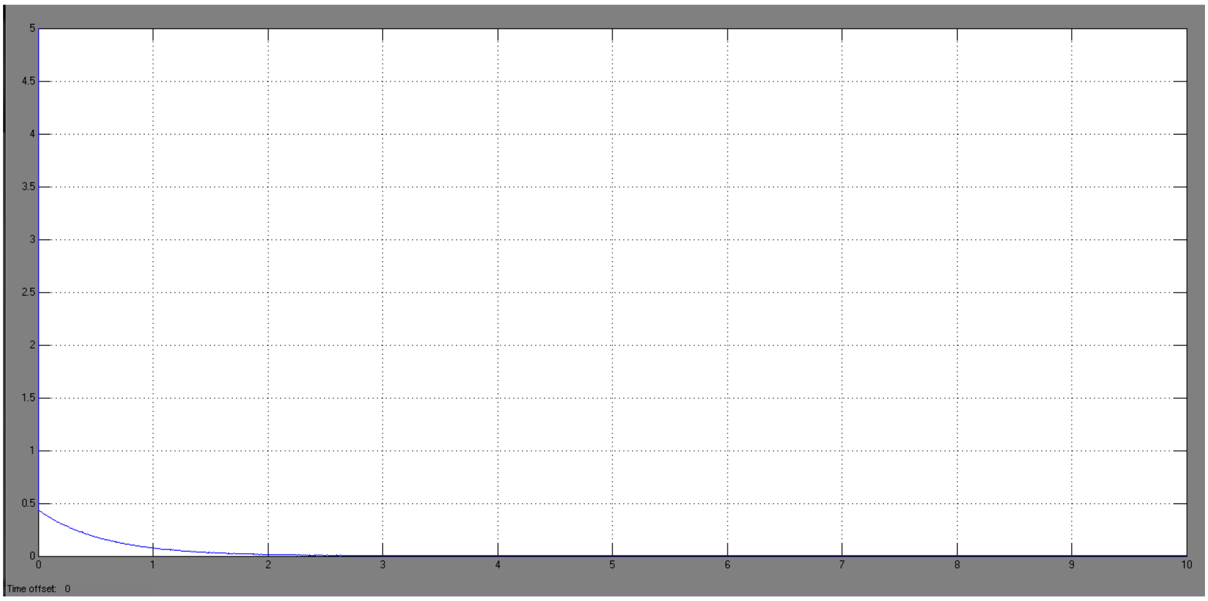

Figure 42: Error of the closed loop system (first order) with P-I-D control, Kp=30, Ki=20, Kd=10

Steady State = $\mathbf{e_{ss}}$ = 0

Rise Time = $\mathbf{t_r}$ = 2.4 seconds

Controlled second order system outputs:

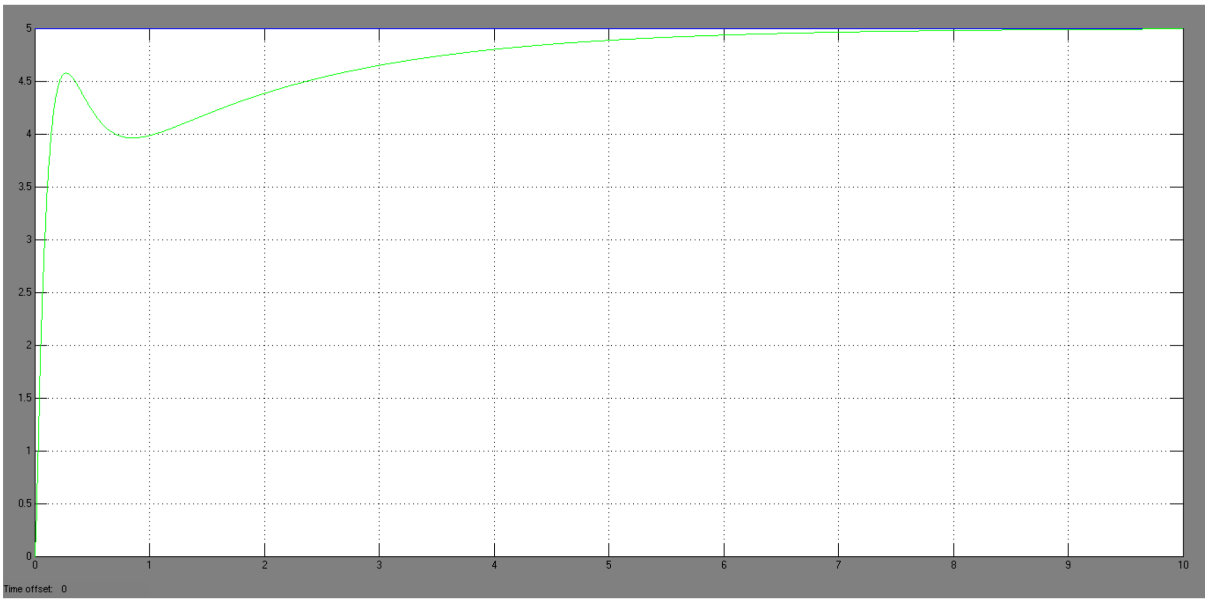

Figure 43: Output of the closed loop system (second order) with P-I-D control, Kp=30, Ki=20, Kd=10

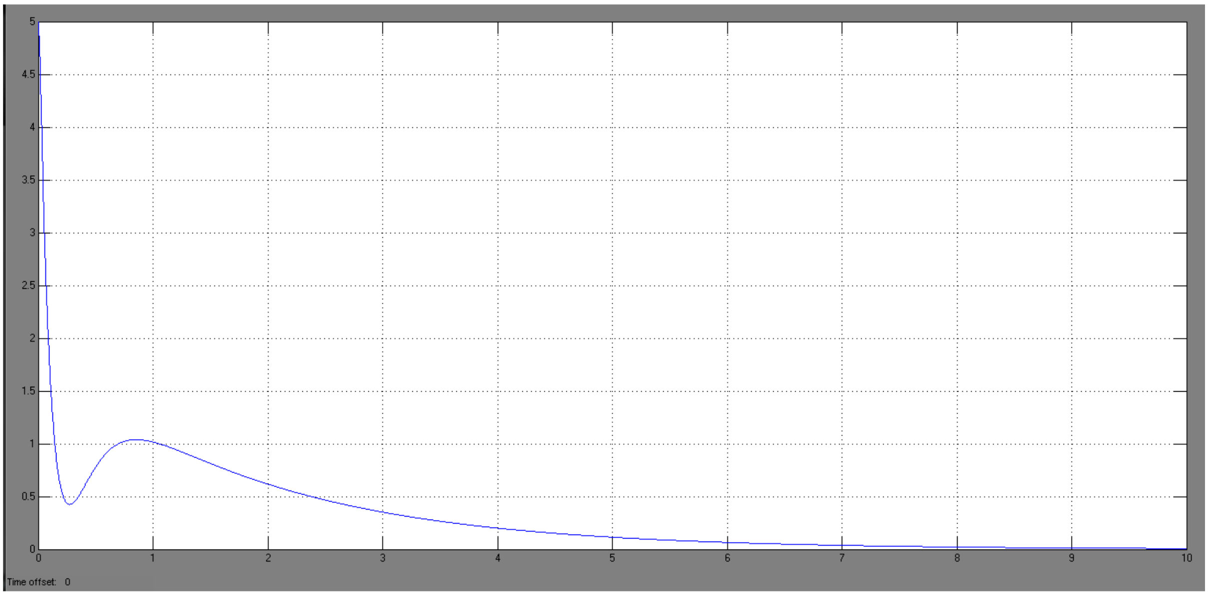

Figure 44: Error of the closed loop system (second order) with P-I-D control, Kp=30, Ki=20, Kd=10

Steady State = $\mathbf{e_{ss}}$ = 0

Rise Time = $\mathbf{t_r}$ = 0.2 seconds

2% Settling time = $\mathbf{t_s}$ =5.24 seconds

Percentage overshoot = 0%

The results above show that P-I-D controller for second order plant requires tuning. Using the automatic tuning option of MATLAB-Simulink one can get the following results:

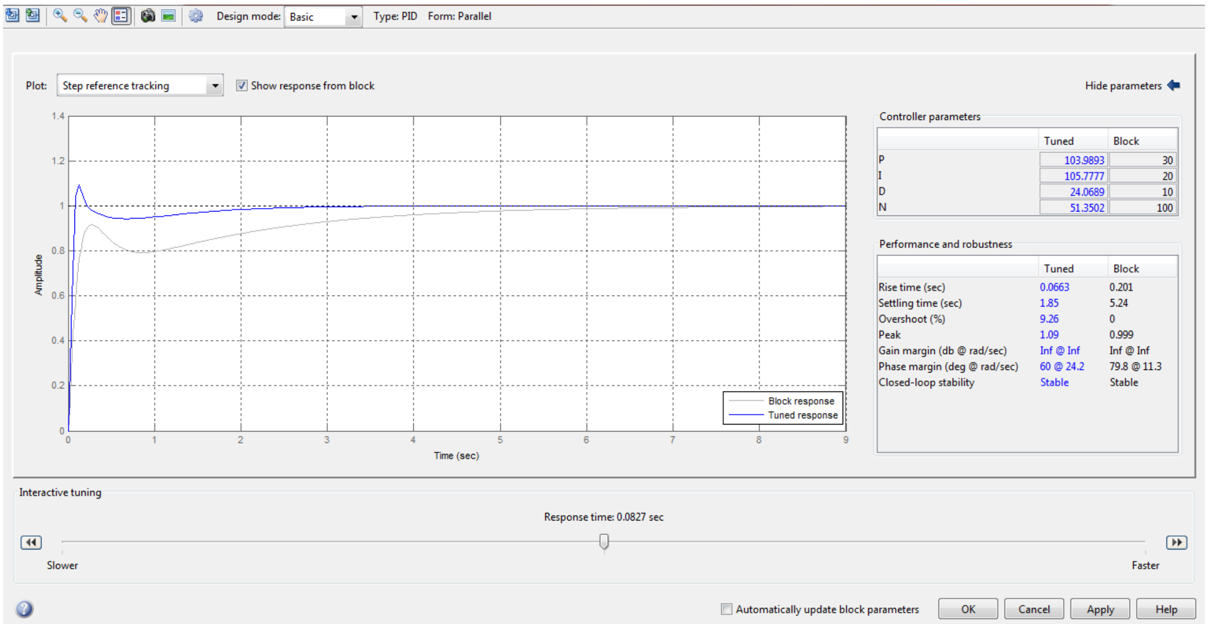

Figure 45: Output of the closed loop system (second order) with P-I-D control after tuning, Kp=104, Ki=106, Kd=24

Steady State = $\mathbf{e_{ss}}$ = 0

Rise Time = $\mathbf{t_r}$ = 0.0663 seconds

2% Settling time = $\mathbf{t_s}$ =1.85 seconds

Percentage overshoot = 9.26%

Phase margin = 60^o (degree)

Conclusions:

P-I-D control and its variations are commonly used in the industry. They have so many applications. Control engineers usually prefer P-I controllers to control first order plants. On the other hand, P-I-D control is vastly used to control two or higher order plants. In almost all cases fast transient response and zero steady state error is desired for a closed loop system. Usually, these two specifications conflict with each other which makes the design harder. The reason why P-I-D is preferred is that it provides both of these features at the same time.

In this recitation, it was aimed to explain how one can successfully use P-I-D controllers in their prospective projects. We tried to focus on almost all aspects of P-I-D control. However, it is almost impossible to fit the explanation of P-I-D controllers within one hour. We suggest for the future Discrete Time Control System students to split the P-I-D controller subject into pieces and explain it more than one recitation hour. Being prospective control engineers, we feel lucky to give a presentation on the P-I-D subject. Finally, we encourage prospective control engineers to use P-I-D controllers wherever necessary, especially, when a great controller is required.