CHAPTER ONE

INTRODUCTION

1.1 Introduction:

PID controllers date to 1890s governor design. PID controllers were subsequently developed in automatic ship steering. One of the earliest examples of a PID-type controller was developed by Elmer Sperry in 1911, while the first published theoretical analysis of a PID controller was by Russian American engineer Nicolas Minorsky, (Minorsky 1922). Minorsky was designing automatic steering systems for the US Navy, and based his analysis on observations of a helmsman, noting the helmsman controlled the ship based not only on the current error, but also on past error as well as the current rate of change; this was then made mathematical by Minorsky. His goal was stability, not general control, which simplified the problem significantly. While proportional control provides stability against small disturbances, it was insufficient for dealing with a steady disturbance, notably a stiff gale (due to steady-state error), which required adding the integral term. Finally, the derivative term was added to improve stability and control.

Trials were carried out on the USS New Mexico, with the controller controlling the angular velocity (not angle) of the rudder. PI control yielded sustained yaw (angular error) of ±2°. Adding the D element yielded a yaw error of ±1/6°, better than most helmsmen could achieve. The Navy ultimately did not adopt the system, due to resistance by personnel. Similar work was carried out and published by several others in the 1930s. [1]

In the early history of automatic process control the PID controller was implemented as a mechanical device. These mechanical controllers used a lever, spring and a mass and were often energized by compressed air. These pneumatic controllers were once the industry standard.

Electronic analog controllers can be made from a solid-state or tube amplifier, a capacitor and a resistor. Electronic analog PID control loops were often found within more complex electronic systems, for example, the head positioning of a disk drive, the power conditioning of a power supply, or even the movement-detection circuit of a modern seismometer. Nowadays, electronic controllers have largely been replaced by digital controllers implemented with microcontrollers or FPGAs. However, analog PID controllers are still used in niche applications requiring high-bandwidth and low noise performance, such as laser diode controllers.

Most modern PID controllers in industry are implemented in programmable logic controllers (PLCs) or as a panel-mounted digital controller. Software implementations have the advantages that they are relatively cheap and are flexible with respect to the implementation of the PID algorithm. PID temperature controllers are applied in industrial ovens, plastics injection machinery, hot stamping machines and packing industry.

A proportional–integral–derivative controller (PID controller) is a control loop feedback mechanism (controller) commonly used are used in industrial production for controlling equipment or machines (Industrial control systems).

Control system is a device, or set of devices, that manages, commands, directs or regulates the behavior of other devices or systems. There are two common classes of control systems, open loop control systems and closed loop control systems. In open loop control systems output is generated based on inputs. In closed loop control systems current output is taken into consideration and corrections are made based on feedback. A closed loop system is also called a feedback control system.

In the case of linear feedback systems, a control loop, including sensors, control algorithms and actuators, is arranged in such a fashion as to try to regulate a variable at a setpoint or reference value. An example of this may increase the fuel supply to a furnace when a measured temperature drops. PID controllers are common and effective in cases such as this. Control systems that include some sensing of the results they are trying to achieve are making use of feedback and so can, to some extent, adapt to varying circumstances.

Linear control systems use linear negative feedback to produce a control signal mathematically based on other variables, with a view to maintain the controlled process within an acceptable operating range. The output from a linear control system into the controlled process may be in the form of a directly variable signal, such as a valve that may be 0 or 100% open or anywhere in between. Sometimes this is not feasible and so, after calculating the current required corrective signal, a linear control system may repeatedly switch an actuator, such as a pump, motor or heater, fully on and then fully off again, regulating the duty cycle using pulse-width modulation. [2]

The PID control scheme is named after its three correcting terms, whose sum constitutes the manipulated variable (MV). The proportional, integral, and derivative terms are summed to calculate the output of the PID controller. The proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant K_p, called the proportional gain constant, the contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain (K_i) and added to the controller output, and last term the derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain (K_d). The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain (K_d).

1.2 Objective:

Study the PID controller theory, design and tuning using five methods with certain transfer function and compare the result.

1.3 Methodology:

We extract the open loop response with step function to get some points that helping to calculate the parameters of PID function (K_p, T_i, T_d) any method has table explain used to find his own value, simulate these values to get the transient response of the closed loop system (settling time (t_s), the rise time (t_r) and the peak overshoot (MP%)) finally discussed the results, compare it and choose the best method.

1.4 Thesis Structure:

This thesis will be divided into five chapters. Chapter one deals an introduction, while chapter tow makes a review of the previous studies, Chapter three view a PID controller theory, Chapter four will cover the proposed tuning methods to be evaluated and their responses will be compared using percent overshoot, settling time, and rise time, and finally chapter 5 will draw conclusions and the recommendations.

CHAPTER TOW:
LITERATURE REVIEW

2.1 Background:

PID (proportional integral derivative) control is one of the earlier control strategies. Its early implementation was in pneumatic devices, followed by vacuum and solid state analog electronics, before arriving at today's digital implementation of microprocessors. It has a simple control structure which was understood by plant operators and which they found relatively easy to tune. Since many control systems using PID control have proved satisfactory, it still has a wide range of applications in industrial control. According to a survey for process control systems conducted in 1989, more than 90 of the control loops were of the PID type. PID control has been an active research topic for many years. Since many process plants controlled by PID controller shave similar dynamics it has been found possible to set satisfactory controller parameters from less plant information than a complete mathematical model These techniques came about because of the desire to adjust controller parameters in situ with a minimum of effort, and also because of the possible difficulty and poor cost benefit of obtaining mathematical models. The two most popular PID techniques were the step reaction curve experiment, and a closed-loop experiment under proportional control around the nominal operating point. [3]

By most accounts PID control was introduced in 1910, by Elmer Sperry's ship autopilot. The Fulscope pneumatic controller, which was introduced by Taylor Instrument Companies, was completely redesigned in 1939. This new improved version provided in addition to proportional and reset control, an action dubbed "Pre-act" by the Taylor Instrument Company. In the same year

“Hyper-reset” was introduced in the Stabilog pneumatic controller, which was a product designed by the Foxboro Instrument Company which also previously only had proportional and reset control. The Pre-act and Hyper-reset terms provided a control action proportional to the derivative of the error signal. The reset provided a control action proportional to the integral of the error signal therefore both controllers offered PID control. Only the Taylor Instrument Fulscope offered full field adjustment of the controller parameters. The Stabilog had to be set at the factory to one of the four available derivative-plus-integral terms. The proportional gain of the controller was field adjustable. With the availability of adjustments for the three terms came the problems. There were no established rules or methods for choosing the appropriate settings for each of the three terms in the controller. The Taylor Instrument Companies realized that this was a weakness and carried out extensive studies in an attempt to devise a set or rules for choosing the proper controller settings for the process being controlled. [4]

The end results of these studies were two papers, by J.G. Ziegler and N.B. Nichols, which were published in 1942 and 1943. Their work presented two ways of determining controller settings. One was based on open-loop tests the other on closed-looped tests. Both were based on empirical data. Their contribution was a quantum leap forward in the science of tuning industrial controllers. It was about ten years or more after that before other authors started to improve and refine their recommendations, but the essence of their approach has remained unchanged to this day. With advances in technology over the years and the advent of digital computing, automatic control now offers a wide range of choices for control schemes. [5, 6]

PID control algorithms remain the most popular control scheme applied in industry. They are utilized in more than 90% of control applications. The PID controller use has been recommended for the control of processes with low to medium order plant transfer functions that have relatively small time delays. The

PID control scheme is also well suited when parameter setting must be made using tuning rules and when controller synthesis is performed either once or more often due to its ability to allow for easy parameter changes. The success of PID control in the process and manufacturing industry is based on the ability to stabilize and control around 90% of existing processes. [7]

This success is overshadowed, however, by a lack of performance in many applications. It has been reported that a large percentage of the installed PID controllers are operated in a manual mode, and that about 65% of the loops operating in the automatic mode generate a greater variance in closed-loop operation than in open-loop operation (i.e. the automatic controllers are poorly tuned). [7, 8]

This deficiency in controller performance is usually the result of a poorly chosen set of operating parameters due to:

- Lack of knowledge among commissioning personnel and operators.
- Generic tuning methods based on criteria that do not match the specific needs.
- The large variety of PID structures, which leads to errors during the application of standard tuning rules.

These and other surveys show that the selection of PID controller tuning parameters is a common problem in many applications. The most straight-forward way to set up controller parameters is through the use of tuning rules. Currently there is a plethora of literature on the subject of PID tuning techniques and standards. The problem is that this information is disseminated among a large variety of sources and therefore is not conveniently communicated to the engineering and industrial community. The topic has been covered and discussed in media such as journal papers, conference papers, websites and books for the last sixty to seventy years. [7]

A. O'Dwyer, author of the Handbook of PI and PID Controller Tuning Rules, has recorded 408 separate sources of tuning rules. Another issue is the fact that current undergraduate courses in control theory only minimally cover the ideal independent or parallel version of the PID control algorithm. There is no single PID algorithm. Different fields of engineering using feedback control have used different algorithms ever since feedback controls systems began to be mathematically analyzed. [9]

It is often forgotten or simply not known that different manufacturers implement different versions of the PID controller algorithm. The engineer responsible for tuning a control loop must be aware of the form of the algorithm used for the PID controller. Controller tuning rules that work reasonably well on one PID architecture may not work well on another. Another issue is that many engineers prefer one method of tuning over another due to familiarity or ease of use. The question is which method gives the lowest percent overshoot and settling time consistently for a variety of plants. This is motivation behind the work in this thesis on the evaluation of tuning techniques used in industry. [7]

CHAPTER THREE

PID THEORY

3.1 Overview:

This chapter will give an introduction to PID controller design. In section 3.2.1 the proportional controller will be reviewed. The definition of steady-state error will be reviewed as well as the rules for determining steady-state error. Examples of proportional design for a type zero and type one system will be demonstrated using the root locus method, section 3.2.2 there will be a review of the ideal integral compensator. This section will use root locus techniques to add a PI (proportional plus integral) controller to a system to improve its steady-state error without appreciably changing its transient response, section 3.2.3 will cover the design of a PD (proportional plus derivative) controller. It will be shown that the PD controller can be used to improve transient response as well as offer a slight improvement in steady-state error, and section 3.2.4 the realization and design of a PID (proportional plus integral plus derivative) controller will be reviewed. Using root locus techniques, a PID controller will be designed and tested to offer an improvement of steady state error as well as transient response.

3.2 The PID controller types:

Design of different form of the controller:

3.2.1 The proportional controller:

The proportional controller or P controller is the most basic controller. It is simple to implement and easy to tune. Figure (3.1) is a block diagram of a proportional controller. In this system **R(s)** is the reference input and **U(s)** is the output of the controller. **G(s)** is the plant transfer function, and **C(s)** is the variable being controlled. The error **E(s)** equals **R(s) – C(s)**. [8]

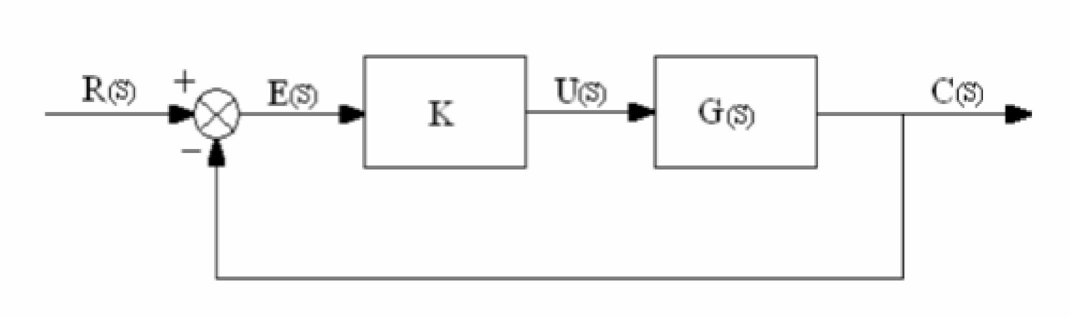


Figure (3.1): Block diagram of a proportional controller

If we consider a step input to the system and make the assumption that **U(t)** must be a finite non-zero value, in order to evoke a non-zero output **C(t)**, an error **E(t)** must exist. Letting **Uss** be the steady-state output of the controller and **Ess** be the steady-state error we have **Uss = K Ess**.

Rearranging we have:

$$Ess = \frac{1}{K}\,Uss \qquad (1)$$

As **K** is increased the steady-state error can be made smaller. This example assumes that there is no integration in the forward path of the system, i.e. the plant **G(s)** does not have a pure integration in its transfer function. If the plant, G(s) were to be approximated as the simplified transfer function we would have the following system shown in Figure (3.2).

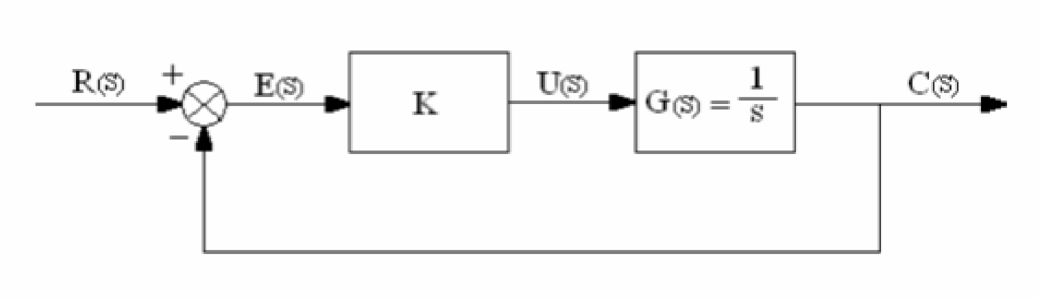


Figure (3.2): Proportional controller acting on a motor

In this case there will be zero steady-state error. For the same step input **R(t)**, as **C(t)** increases **E(t)** will decrease until it reaches zero since **E(t) = R(t) –C(t)**. Since an integrator can have a constant output without any input there will always be a non-zero value for **C(t)**.

Depending on the type of system, a proportional controller, or any controller for that matter, may or may not have a non-zero steady-state error. The following rules apply to negative unity feedback systems. It can be shown that the number of pure integrations in the forward path transfer function **G(s)** of a closed loop negative feedback system will determine the steady-state error, **e(∞)**, for step input **R(s)** :

$$e(\infty) = \lim_{s\to 0} \frac{sR(s)}{1+KG(s)} \qquad (2)$$

For a unit step input, substituting R(s) = 1/s into Equation (3.2) gives:

$$e(\infty) = \lim_{s\to 0} \frac{s/s}{1+KG(s)} = \lim_{s\to 0} \frac{1}{1+KG(s)}$$

The term, $\lim_{s\to 0} KG(s)$ is given the symbol $\mathbf{K_p}$ and is called the position error coefficient.

$$e(\infty) = \frac{1}{1+\mathrm{Kp}} \qquad (3)$$

To get a small steady-state error to a step input, K_p must be made high. This can be achieved by increasing the proportional gain K. Therefore, the higher the gain, the smaller the error will be. [10]

The error coefficient can be increased, and the result is a reduction in error simply by increasing K, the proportional gain of the system. However, increasing K may lead to instability. Since this is a review of proportional control, the system type, i.e. how many pure integrations in the forward transfer function G(s), will determine the value of the steady-state error. In most cases, it is required that the steady-state error of the closed loop system due to a step input be zero. For this to be so, $\mathbf{K_p}$ must be infinite. The open loop transfer function, KG(s) can be expressed in factored form as,

$$KG(s) = \frac{K(s + a1) + (s + a2) \ldots}{s^n(s + b1) + (s + b2) \ldots}$$

If the power n of the factor s^n, is zero, then it is clear that $\mathbf{K_p}$ will not be infinite. However, if n is greater than or equal to one, $\mathbf{K_p}$ will always be infinite. Therefore the value of n determines the value of the error coefficients, which in turn determine whether the steady-state error equals zero. A system is called type zero if n=0, type one if n=1, type tow if n=2, and so on. Table (3.1) is a summary of system type and steady state errors.

Table (3.1): Relationships between input, system type, static error constants, and steady-state errors

		Type 0		Type 1		Type 2	
Input	**Steady-state error formula**	**Static error constant**	**Error**	**Static error constant**	**Error**	**Static error constant**	**Error**
Step, $1/s$	$\frac{1}{1+K_p}$	$K_p =$ Constant	$\frac{1}{1+K_p}$	$K_p = \infty$	0	$K_p = \infty$	0

For Figure (3.2), if we let $G(s) = \frac{1}{(s+3)(s+5)}$, type one system which is a second order it can be seen by the step response plot in Figure (3.3) that the steady state error for a proportional gain K of one that $e_{ss} = \frac{1}{1+K_p}$=0.9333 . [7]

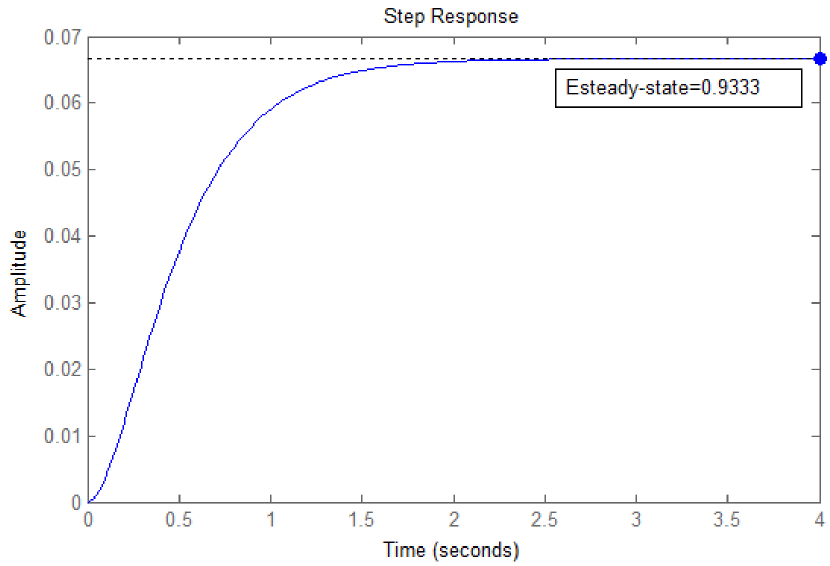


Figure (3.3): Second order type one system step response with gain of 1

A plot of the root locus in Figure (3.4) of the system shows that it will remain stable as the gain K is increased.

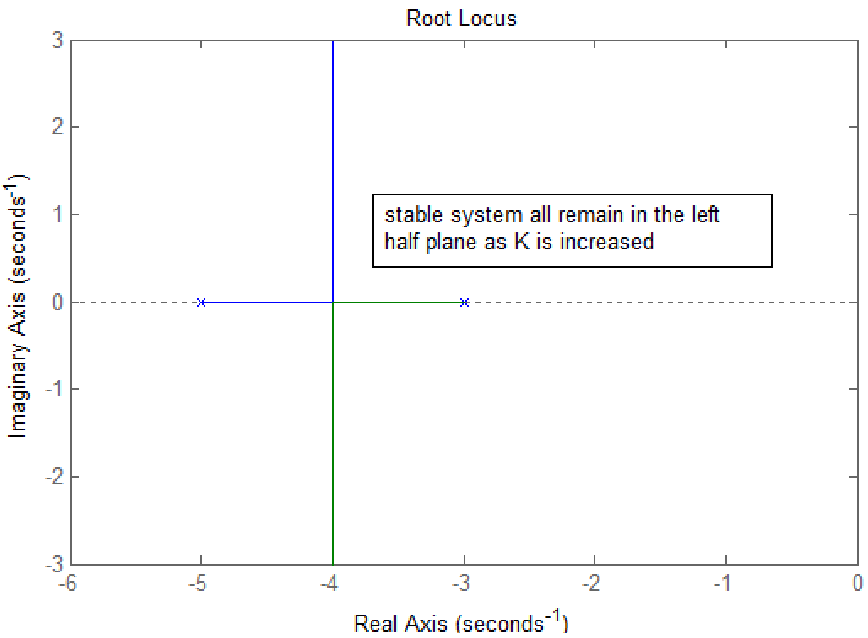


Figure (3.4): Root locus of second order system

By raising the controller gain to 400 we achieve a steady-state error of 0361. The system remains stable and the settling time decreases, however we have introduced a certain amount of overshoot and ringing into the system. Figure (3.5) depicts the results of a step response to the system with the gain increased to 400.

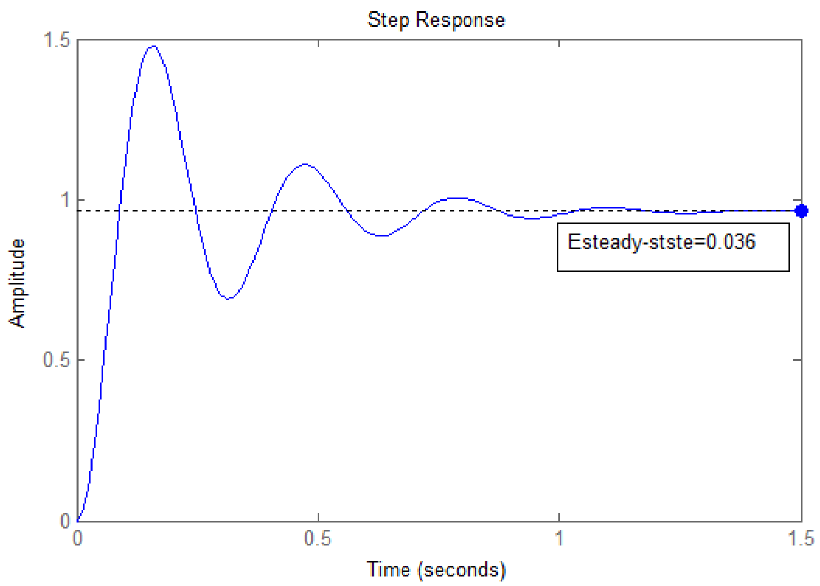


Figure (3.5): Second order plant response with gain of 400

If the plant were to be represented as a type one third order system with $G(s) = \frac{1}{s(s+3)(s+5)}$, type tow system the steady state error for a step input will now be zero. Increasing the gain beyond a certain point will cause instability. By reviewing the root locus plot in Figure (3.6) we see that when the system gain is increased to a value greater than 120, the poles of the system will move into the right half plane and the system will become unstable. [9]

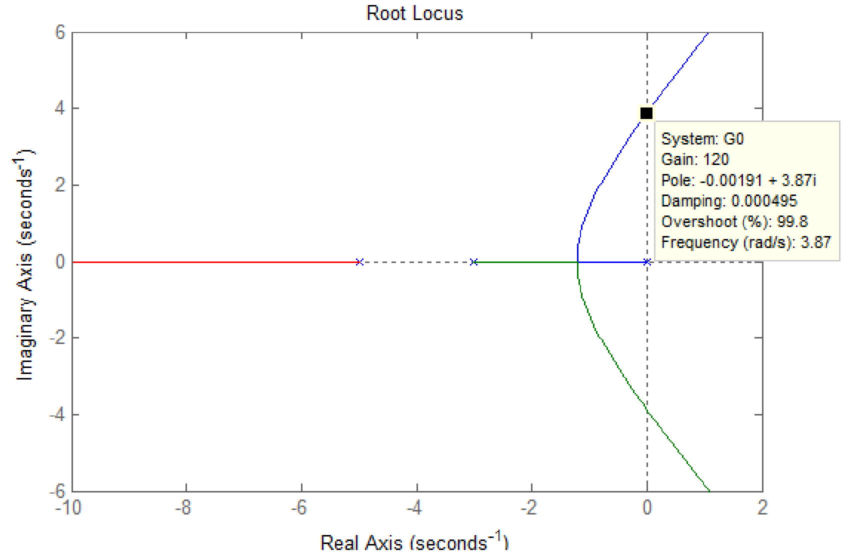


Figure (3.6): Root locus of a third order system

Figure 7 shows step responses for K = 1 and 100. It should be noted that both systems have zero steady state error, however as the gain is increased to 100 the system starts to ring. The tradeoff is that with K=100 the response has a shorter settling time. If the gain is increased to 120 and beyond the system will become unstable. Note that as shown in Figure 8, with a gain of 120 the system oscillates at its natural frequency of 3.87 radians/sec.

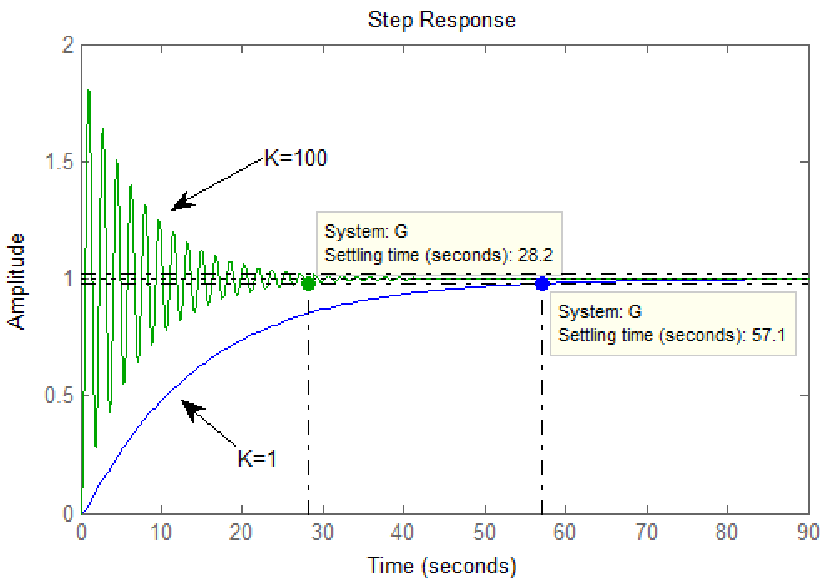


Figure (3.7): Step response for type 1 third order system with K=1, K= 100

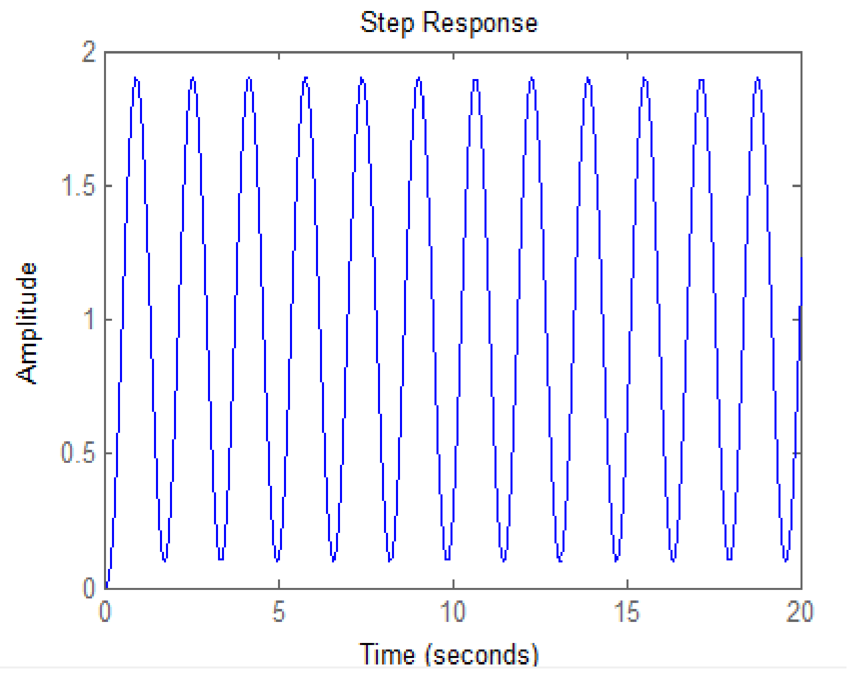


Figure (3.8): Step response for type one third order system with K=120

Tuning a proportional controller is fairly straightforward. The gain is simply raised until instability appears, then it is decreased until the desired performance is achieved. In industry when tuning a loop, if it is possible to apply a square wave to the system the following procedure is used

1. Set K low

2. Apply a square wave having a fundamental frequency that is about 10% of the system bandwidth (point where gain has fallen to -3db) to insure that there is no roll off of the output due to bandwidth limitations.

3. Raise K for little or no overshoot.

4. If the system response does not meet operation criteria, continue lowering K until satisfactory results are obtained. Otherwise the process complete.

A square wave is a rather difficult command to follow perfectly like that of a step response, therefore a small amount of overshoot to a square wave is acceptable in most cases.

Often times other factors, primarily noise, will ultimately limit the proportional gain to a value below what the stability criterion demands. [10]

3.2.2 Integral controller:

The major shortcoming of the proportional controller for a type zero system is that the steady state error is not exactly zero. This is readily corrected by using an ideal integral compensator. Because the integral output will grow ever larger with even small DC error, any integral gain will eliminate steady-state error. This single advantage is why PI (proportional plus integral) control is often preferred over P only control. A compensator that uses pure integration to improve steady-state error is referred to as an ideal integral compensator. The ideal compensator has to be constructed with active components, which in the case of electric networks requires the use of active amplifiers and sometimes additional power sources. A passive compensator is less expensive to implement, however in this case the steady-state error is not driven to zero, where as it is in cases where ideal compensation is used. [10]

It has been shown in section 3.2.1 that steady-state error can be removed simply by adding a pure integration to the controller or plant in a cascaded system. This of course will change the system type from a type zero to a type one. The problem that may arise is that adding this pure integration will also change the transient response characteristics of the system. Figure (3.9) shows a type zero, third order, plant using a proportional controller.

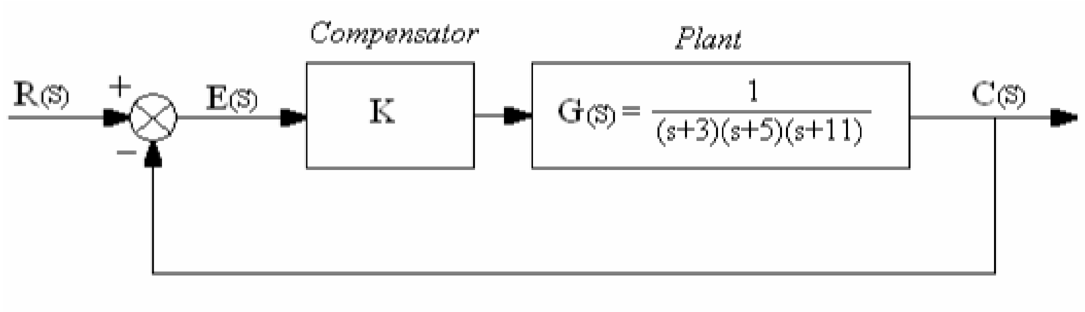


Figure (3.9): type zero proportional controlled system

If this system were operating with the desired transient response, corresponding to a damping ratio ζ = 0.2, we would require a gain $K = 722$ as can be seen by the plot of the root locus for the system in Figure (3.10). However, this system gives us a steady-state error of 0.186. This can be seen in Figure (3.11). It should also be noted that this system can be approximated as a second order system since the third pole is much farther to the left real component σ = -16 than the two dominant poles for which σ= -1.49.

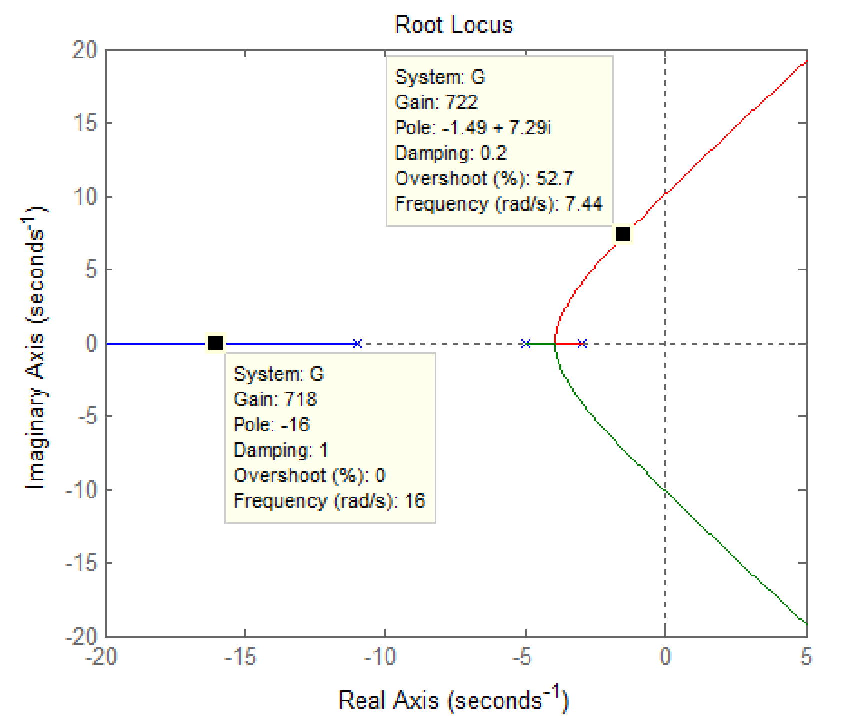


Figure (3.10): P only system operating at 0.2 damping ratio

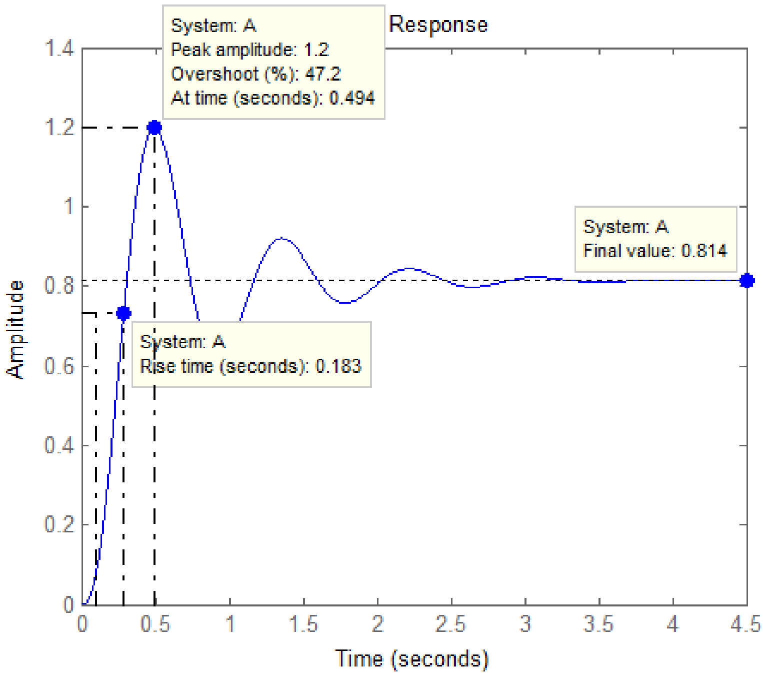


Figure (3.11): Step response for P only compensated system

If we were to add an integrator to the proportional controller, the system type becomes one، therefore eliminating any steady-state error to the step input. The problem here is that the original pole location for ζ =0.2 is no longer on the root locus for the system, as can be seen in figure (3.10).

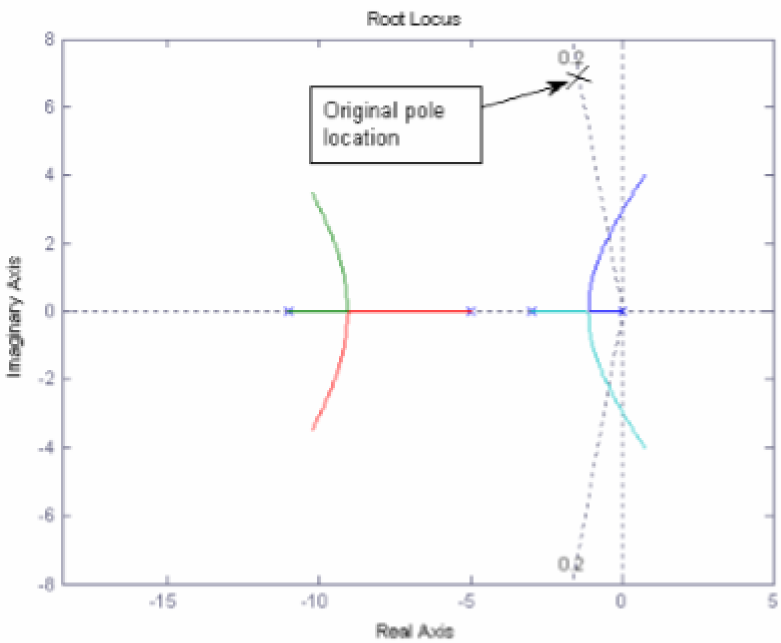


Figure (3.12): Root locus for PI control with 0.2 damping ratio and no zeros

Analyzing the root locus in Figure (3.12) it is found that a system gain of 412 will result in a damping ration ζ=0.2. This will give the same percent overshoot as the original system but with zero steady-state error. However the transient response will be considerably slower i.e. longer rise time and longer settling time, as seen in Figure (3.13). [9]

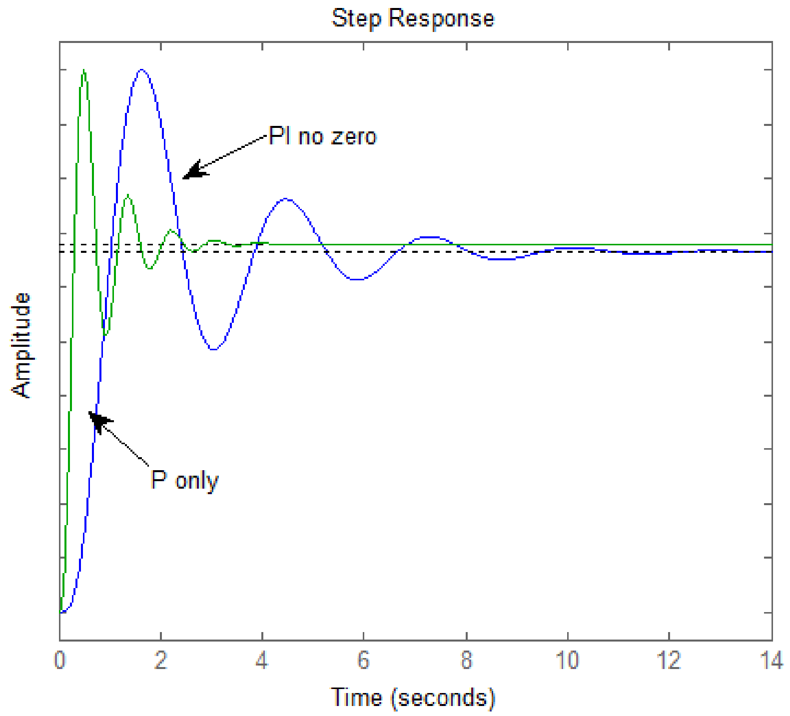


Figure (3.13): Normalized step response for P and PI controller with pure integration and no zero

The system can be made more like the original P only system shown in Figure (3.11) and still eliminate steady state error by adding a zero to the controller near the origin. The effect of the zero will help cancel out the angular contribution of the added pole at the origin. This is the final implementation of an ideal PI controller; one realization of the system is depicted in Figure (3.14). [8]

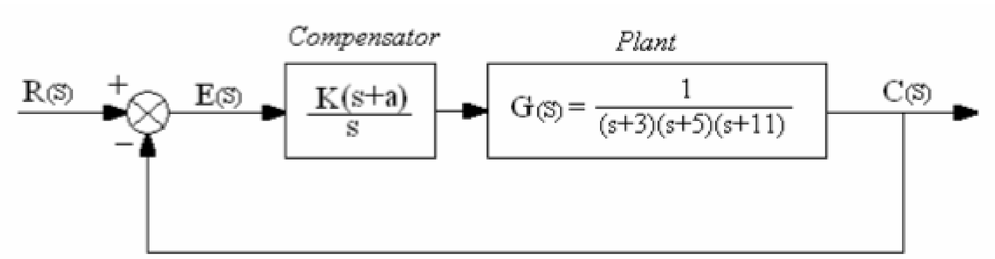


Figure (3.14): Full PI compensator with zero added

If parameter **a** in Figure (3.14) is chosen to be equal to 0.2 we have the root locus plot shown in Figure (3.15).

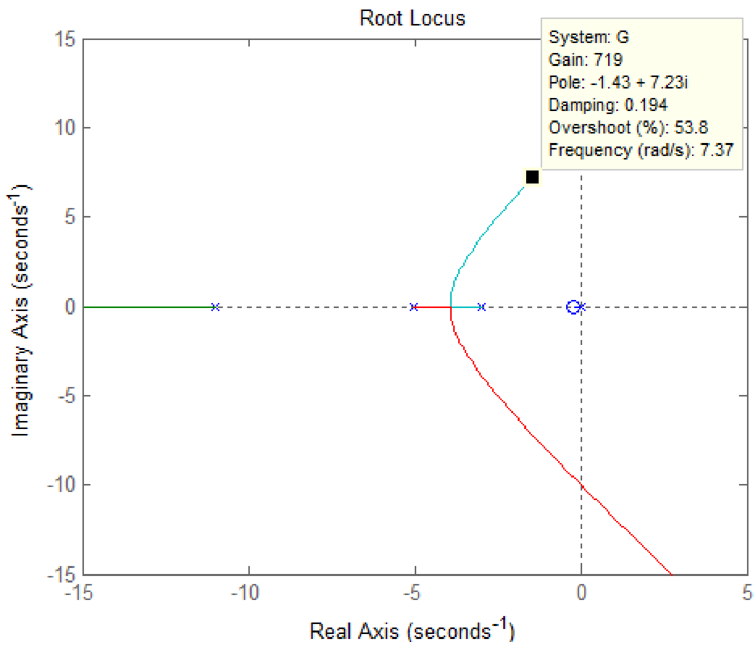


Figure (3.15): Root locus of PI system with zero added

Note that this root locus is extremely close to the original root locus of the proportional only system. The result is a system with the desired transient response and zero steady-state error to a step input. This can be seen in the step response plot of Figure (3.16).

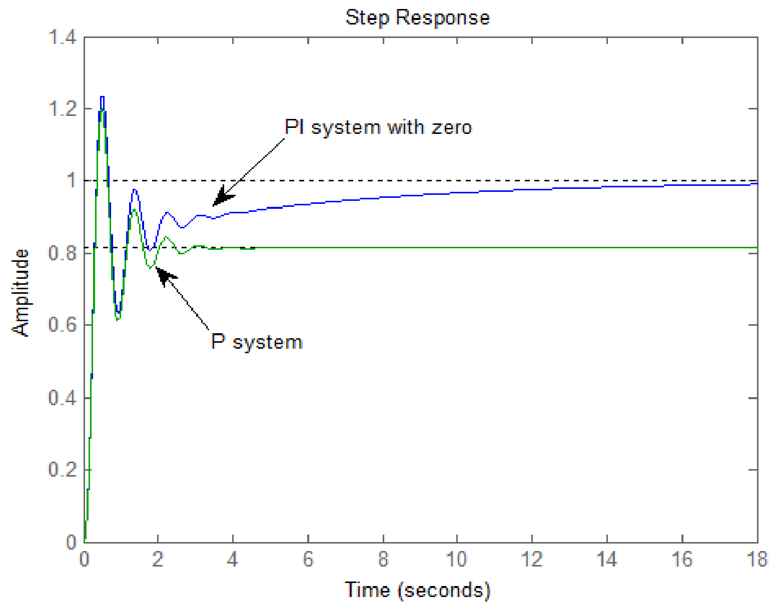


Figure (3.16): Step response for P and PI system with zero added

It should be noted the transient response of both systems are rather similar, however the settling time of the PI system is approximately 14 seconds while the settling time of the P only system is time of the PI system is approximately 14 seconds while the settling time of the P only system is fact is that the compensated system reaches the uncompensated system’s final value in less time. The remaining time is used to improve the steady-state error over that of the uncompensated system. The typical textbook realization of the ideal PI controller is in what is called the parallel form shown in Figure (3.17). [10]

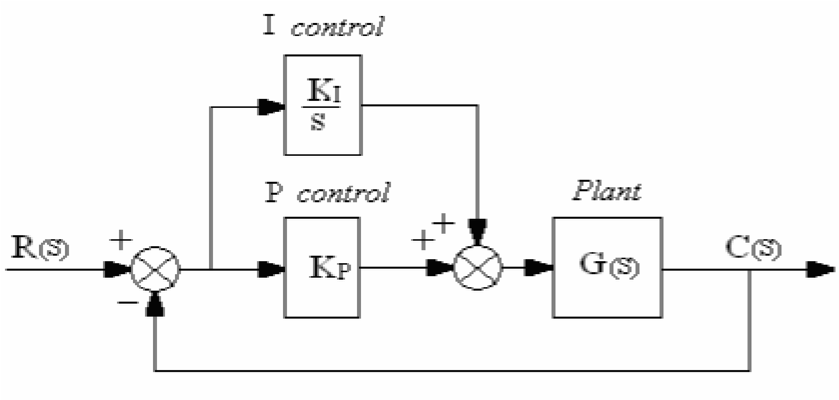


Figure (3.17): Parallel form of PI controller

The controller transfer function is given by:

$$G_c(s) = K_p + \frac{K_i}{s} = \frac{K_p(s+\frac{K_i}{K_p})}{s} \quad (4)$$

Where, comparing with the compensator of Figure (3.14), it can be seen that $a = \frac{K_i}{K_p}$ and $K_p = K$ the process for tuning a PI controller is much the same as tuning a P controller. The following method may be used in industry, provided that a square wave can be applied to the system:

i. Zero Ki and set K_p low.
ii. Apply a square wave at about 10% of the desired loop bandwidth to insure there is no roll off.
iii. Raise Kp for little or no overshoot.
iv. If the system response is too noisy, lower K_p until it is not.
v. Raise Ki for 15% overshoot.

3.2.3 The Derivative Controller:

If a system were to already have zero steady-state error, i.e. type one or greater, or an acceptable level of steady-state error, the designer may want to improve the transient response of the system. The design objective here may be to reduce settling time and achieve a desirable percent overshoot. This can be accomplished by the use of ideal derivative compensation. The term ideal refers to the fact that a pure differentiation is applied to the forward path. The ideal proportional plus derivative PD controller uses active components in its realization, and the pros and cons of design and manufacturing the system are similar to those of the previous active PI network.

The transient response of a system can be chosen by selecting the required closed-loop pole locations on the s-plane. If these pole locations are not already on the root locus of the system, then the system root locus must be reshaped in order to include these poles. One way to accomplish this is to add a zero to the forward path transfer function.

$$G_c(s) = s + a_0 \tag{5}$$

This is the ideal derivative or PD controller and is the sum of a differentiator and a pure gain. In the next example the effects of adding zeros at -3, -4 and -6 will be examined on the following uncompensated plant. [9]

$$G(s) = \frac{1}{(s+2)(s+3)(s+8)}$$

Figure (3.18) shows the root locus of the uncompensated system with a damping ratio $\zeta = 0.6$.

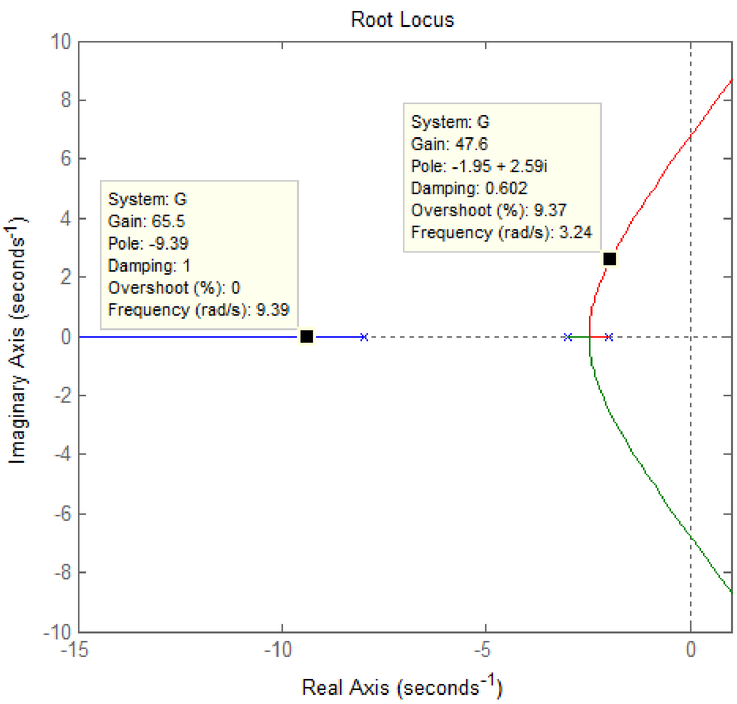


Figure (3.18): Type zero system before PD compensation

Note that the real component of the original system's third closed loop pole for achieving the damping ratio of 0.6 that is at least 5 times that of the dominant closed loop poles. Thus, the original system can be approximated as a second order system. Now, adding a zero to the original system at -3 gives the corresponding transfer function, and the root locus shown in Figure (3.19)

$$G(s) = \frac{K(s+3)}{(s+2)(s+3)(s+8)}$$

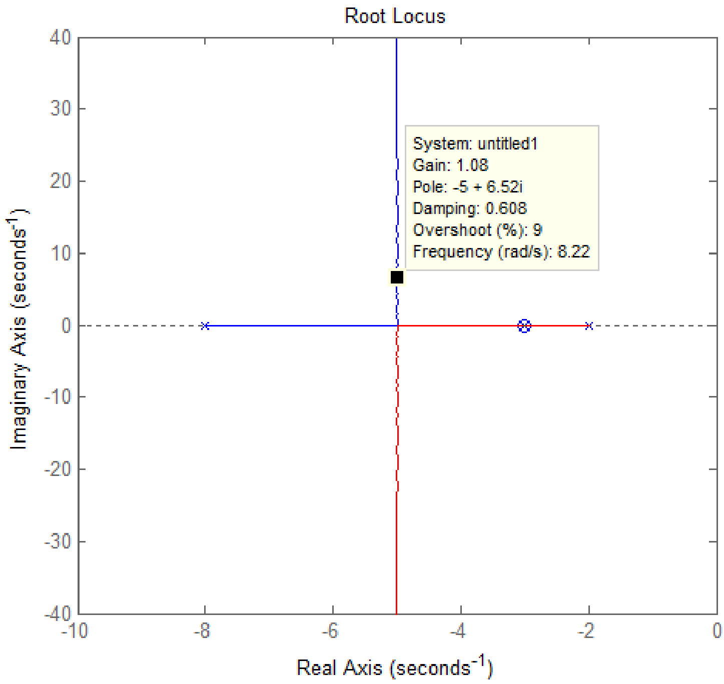


Figure (3.19): Type zero compensated system with zero at -3

Notice that the zero at -3 cancels out the open loop pole at -3, thus turning the system into a pure second order system. It can also be seen for the same damping ratio, the pole and gain values have changed. Figure (3.20) is the root locus for the original system with a zero added at -4 thus,

$$G(s) = \frac{K(s+4)}{(s+2)(s+3)(s+8)}$$

It can be seen on this plot that for the same damping ratio, the pole, and gain values have changed from those of the previous two systems. Also note that the third closed loop pole is not far removed from the two dominant closed loop pole locations, however the third pole is in close enough proximity to the added zero to approximate a pole zero cancellation. Therefore this system can be approximated as a second order system. [9]

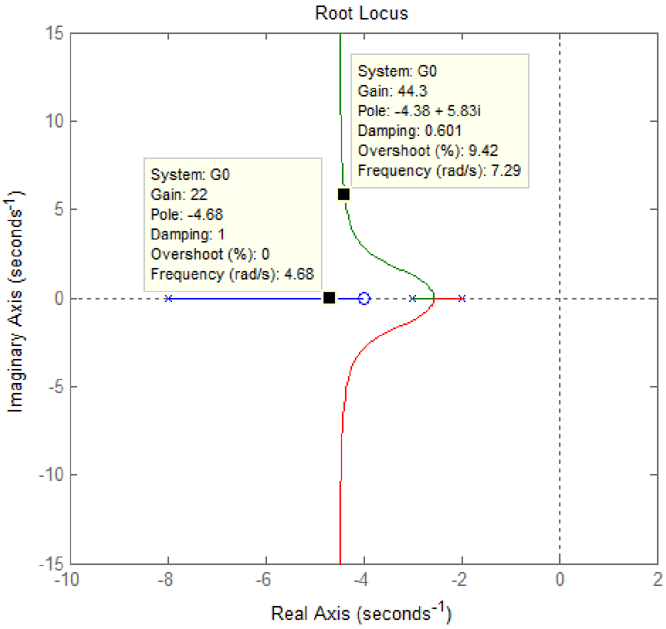


Figure (3.20): Type zero compensated system with zero at -4

In Figure (3.21) the zero is now moved to -7 giving the transfer function

$$G(s) = \frac{K(s+7)}{(s+2)(s+3)(s+8)}$$

Again it is observed that this system has different pole and gain values for a $\zeta = 0.6$. This system can also be approximated as a second order system because of the fact that the zero is fairly far removed from the dominate pole pair and it is also in close proximity to the third pole, offering a rough approximation of pole zero cancellation.

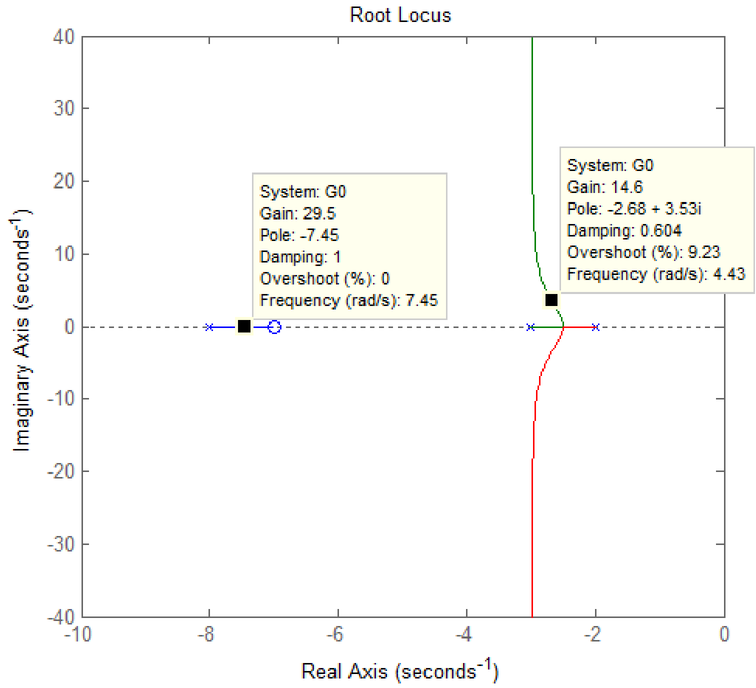


Figure (3.21): Type zero compensated system with zero at -7

By examining the normalized step responses to the four systems in Figure (3.22) it can be seen that the percent overshoot in each case is the same, corresponding to the choice of $\zeta = 0.6$. It can also be observed that the peak time and settling time have decreased from those of the original uncompensated system. From Figure (3.23) which shows the actual step responses of the systems, it can be observed that as the added zeros traverse farther to the left from the dominant pole pair on the real axis, (as seen on the root locus) there is a point where the effect of the zero is lessoned and the system response starts reverting back to that of the original uncompensated system. [8]

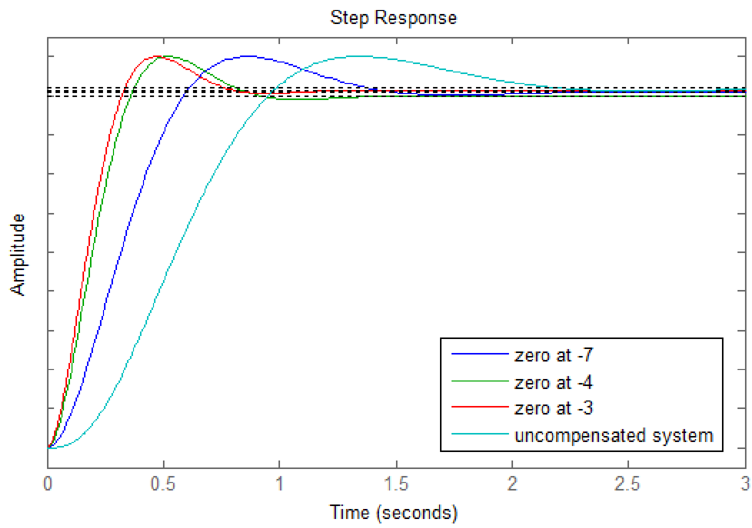


Figure (3.22): Normalized step responses for uncompensated and derivative compensated systems

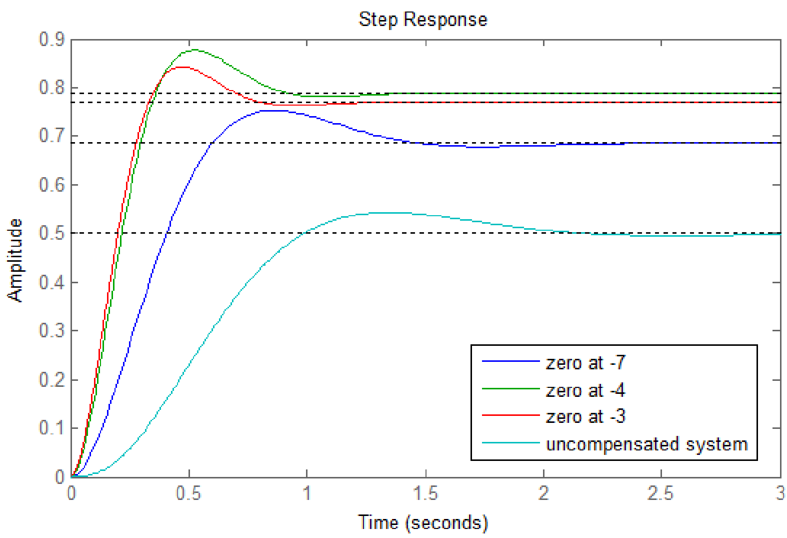


Figure (3.23): Step responses for uncompensated and derivative compensated systems

It can be seen that the step responses for the systems with a zero at -3 and -4 give the most improvement to transient response and steady-state error therefore, it is important to make a judicious choice when selecting the zero location. The common textbook realization of a PD control scheme is shown in Figure (3.24)

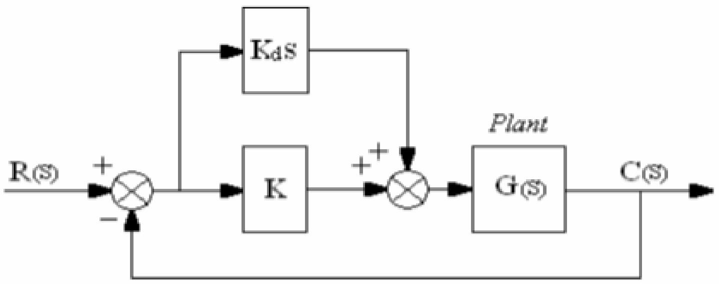


Figure (3.24): Implementation of PD controller

The transfer function of the controller itself can be represented as

$$G_c(s) = K + K_d s = K_d(s + \frac{K}{K_d}) \quad (6)$$

With this representation K / K_d can be chosen to equal the negative value of the required controller zero, while K_d can be chosen to meet the required loop gain. Some things to consider about pure derivative gain are the fact that differentiation is a noisy process. Derivatives by nature have high gain at high frequencies. The level of noise is usually low, but the frequency of noise is high compared to the signal. Differentiation at high frequencies can lead to large unwanted signals. [9, 10]

3.2.4 The Proportional Integral Derivative Controller:

A system that can be used to improve steady-state error as well as transient response is known as the proportional integral derivative controller or PID controller. The mathematical or ideal textbook configuration of the system can be seen in Figure (3.25).

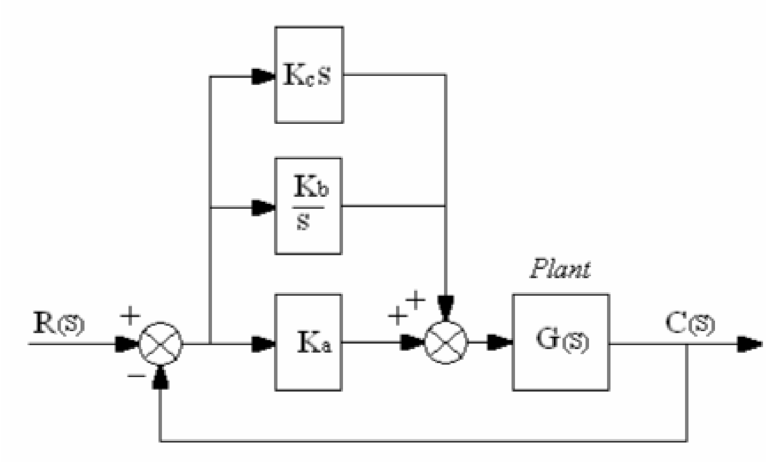


Figure (3.25): Ideal PID representation

The controller transfer function can be represented as

$$G_c(s) = K_a + \frac{K_b}{s} + K_c s = \frac{K_a s + K_b + K_c s^2}{s} = \frac{K_c(s^2 + \frac{K_a}{K_c}s + \frac{k_b}{k_c})}{s} \quad (7)$$

Notice that this controller has one pole at the origin and two zeros. From the review for PI and PD controllers it can be seen that one of the zeros and the pole at the origin will pertain to the ideal integral compensator, and the remaining zero will be used to design in the ideal derivative compensator [10].

The following process can be used to design a PID system. Choosing the example plant transfer

$$G_p(s) = \frac{(s+7.8)}{(s+2)(s+4)(s+9)}$$

and the operating criteria that the uncompensated system operating at 25% overshoot is to be improved to have a 30% reduction in settling time and zero steady-state error, while maintaining 25% overshoot. The root locus of the system is plotted and the closed loop pole for 25% over shoot is determined. As shown in Figure 26, the third pole and zero are found to be a little closer than we would like from the dominant poles to evaluate the system as if it were second order. A simulation was performed and it was found that the second order approximation is still sufficiently valid for us to proceed. Next we find the compensator pole that will yield the 30% reduction in settling time and still maintain 25% over shoot. Using Equation (8) the settling time (Ts) of the uncompensated system is found to be 1.166 seconds; through simulation, the settling time was actual.986 seconds, which is sufficiently close to this demonstration. The new required settling time is.816 seconds, using the calculated value of settling time. The real part of the new pole location is σ = -4.902.

$$T_s = \frac{4}{\xi\omega_n} = \frac{4}{\sigma} \qquad (8)$$

∵ Where σ is the real part of the closed loop dominant pole obtained from the root locus

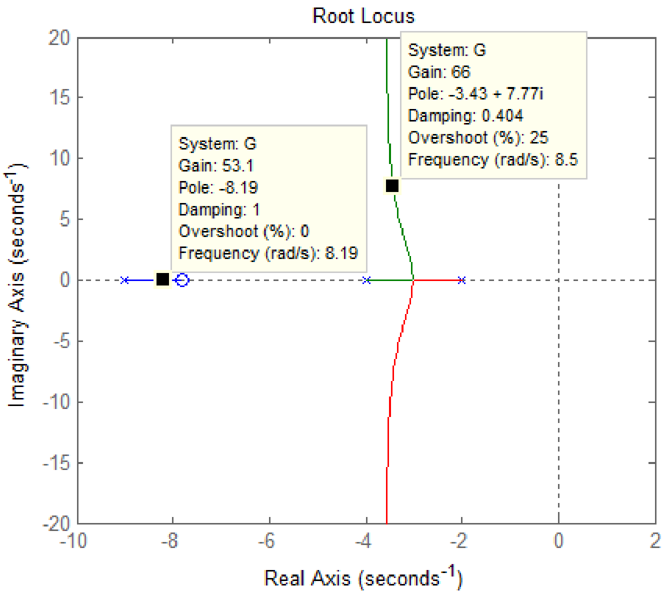


Figure (3.26): System before PID implementation

Again, using Equation (3.10), ω_d is found to equal 11.133. The new closed loop poles for 25% overshoot and a reduced settling time of 0.816 seconds are

$$-4.902 \pm j11.133$$

Finding the pole and zero angles and using Equation (9) to solve for θ_z We have $\theta_z = 13.62°$.

Using Equation (3.10) the compensator zeroes location is found to be-50.833.

$$\sum \theta_z - \sum \theta_p = (2n+1)180°, \tag{9}$$

Where $n = 0, 1, 2 \ldots \theta_p$

$$\omega_d = \sigma \tan(\cos^{-1}(\varsigma)) \tag{10}$$

The root locus of the compensated system,

$$G_p(s) = \frac{K(s+7.8)(s+50.833)}{(s+2)(s+4)(s+9)}$$

Is plotted and it can be seen that a second order approximation is still questionable. A simulated step response is applied to the system and it can be seen that the first part of the design goal has been accomplished. The percent over shoot remains at 25% while the settling time has decreased from 0.986 second to 0.671 seconds, which is a 31% reduction in settling time. Figure (3.27) is the new root locus, while Figure (3.28) gives step responses for the original and compensated systems.

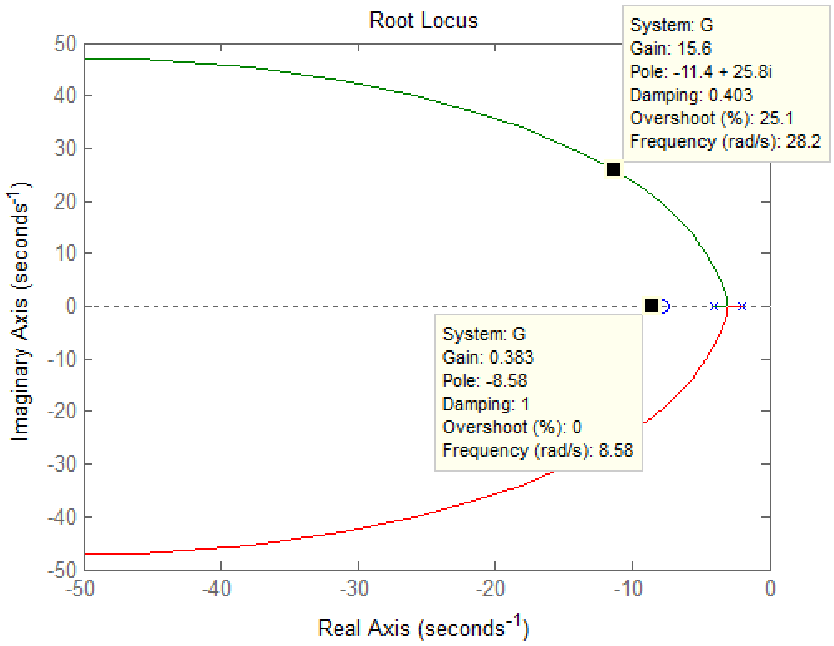


Figure (3.27) Root locus of derivative compensated system

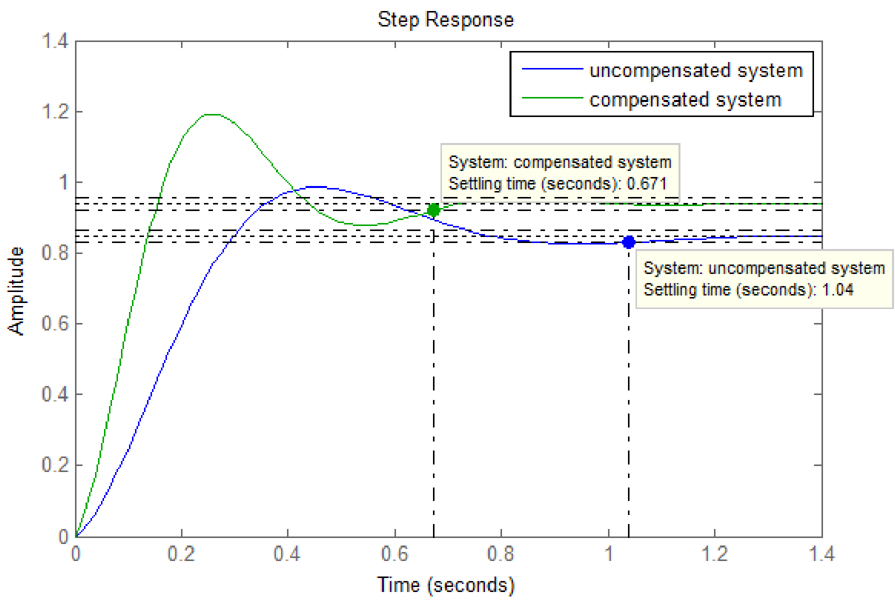


Figure (3.28) Step responses for uncompensated and derivative compensated systems

It can be seen from the step response plots that the system steady-state error has already improved. Next we will add an ideal integral compensator to complete the design of the PID controller. The ideal integral compensator will be added to reduce the remaining steady-state error to zero. The key here is to place the integral compensator zero close to the origin. The choice for the zero of the ideal integral compensator will be -0.9 which gives us the PID controller and plant transfer function as

$$G(s) = \frac{K(s + 7.8)(s + 50.833)(s + 0.9)}{s(s + 2)(s + 4)(s + 9)}$$

Where K is equal to 1.9. Figures (3.29) and (3.30) show the root locus plot for the PID system and the corresponding step responses. Notice that the root locus for the PID system now has four closed loop poles. The step response shows that while the ideal derivative compensator decreased the settling time by the desired amount and also lowered the steady state error, the PID compensator brought the steady-state error two zero, however the settling time increased from that of the derivative compensation, yet was still an improvement from that of the uncompensated system. [9]

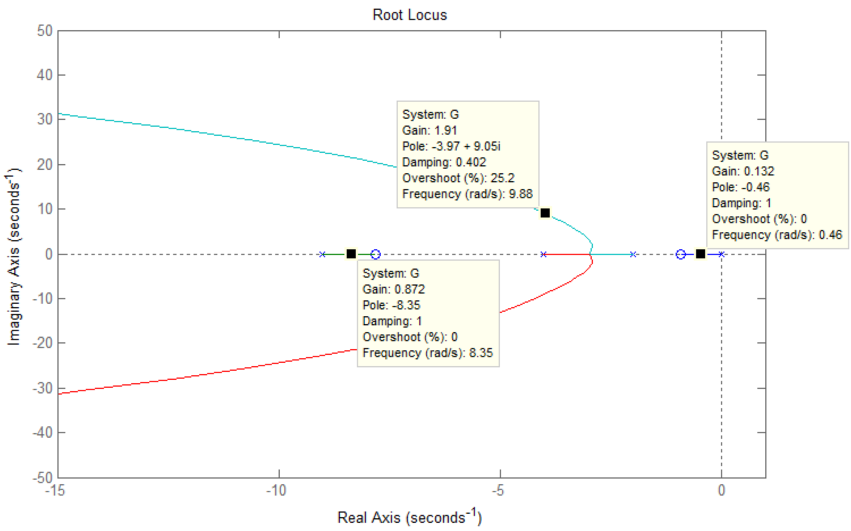


Figure (3.29): PID compensated root locus

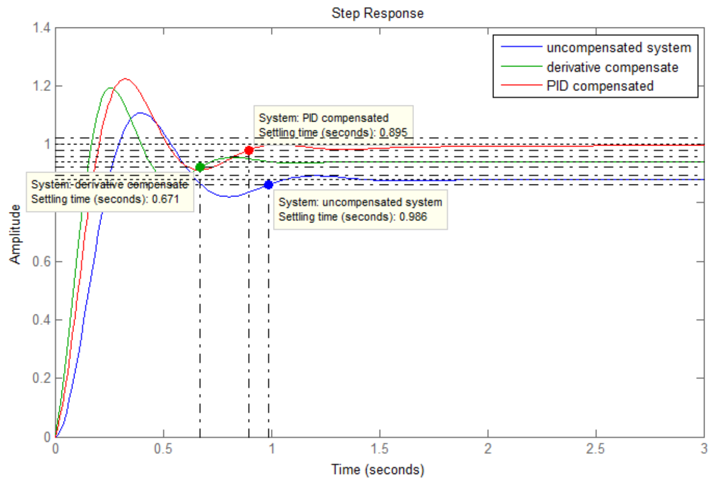


Figure (3.30): Step responses for uncompensated, derivative, and PID

The PID controller transfer function is

$$G_{pid}(s) = \frac{k(s+50.833)(s+0.9)}{s} = \frac{1.9(s^2+51.7323s+45.750)}{s}$$

Comparing this to Equation (7) and solving for the gains $\mathbf{K_a}$, $\mathbf{K_b}$, and $\mathbf{K_c}$, we obtain $K_a = 98.297$, $K_b = 86.925$, $K_c = 1.9$.

CHAPTER FOUR

SIMULATION AND RESULTS

4.1 Introduction:

In this chapter will explain the tuning methods, the way to use it, the system that able to work with it, applied it to example transfer function and calculate the result using MATLAB and compare it.

The transfer function:

$$G(s) = \frac{10}{(s+2)(s+4)(s+8)}$$

Firstly, for the open loop methods need to determine the open loop response and the close loop method to frequency response in tests of system section and compensate it in table of method and extract the transient response parameters (t_r, t_s and MP%) using MATLAB code in appendix A, compare the result and choose the ideal method.

4.2 System tests:

4.2.1 Open loop step response:

Figure (4.1) shows the open loop step response (source value=1), and extract the open loop parameters (a=0.0289, K=0.156, L=0. 197, T=1.063).

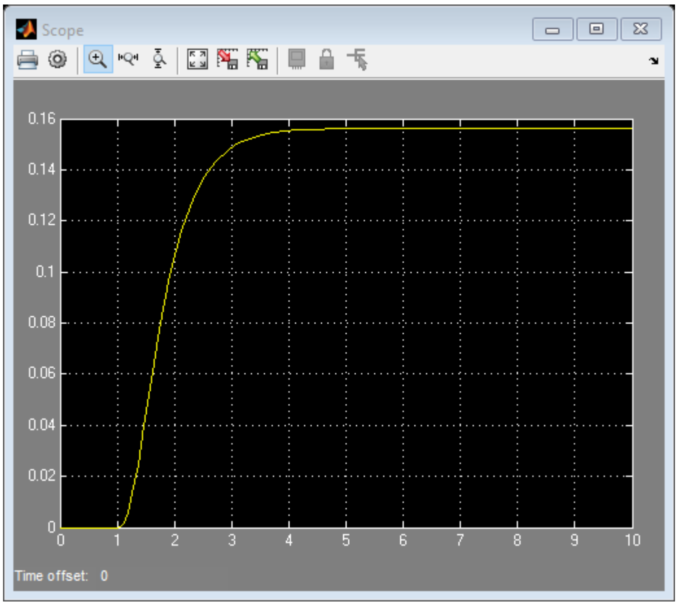


Figure (4.1) open loop response of the system

4.2.2 Close loop response (Frequency response):

Figure (4.2) display to the root locus diagram of the system. In figure (4.3) a, b and c the closed loop response shown with various gain values.

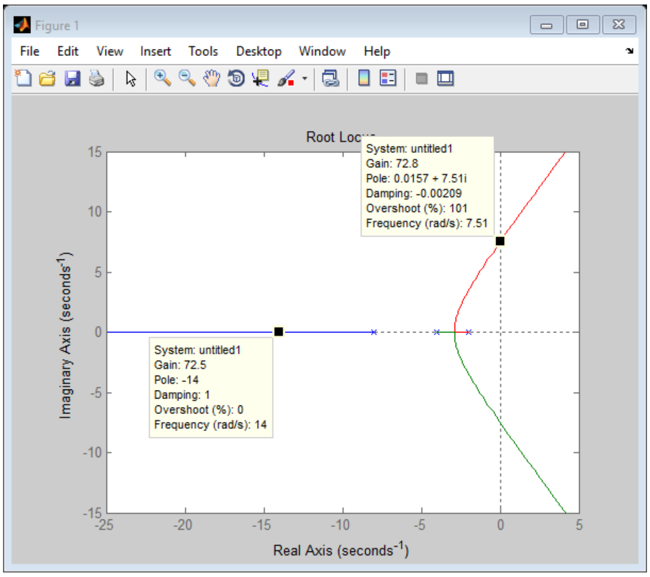


Figure (4.2): Root locus of the system

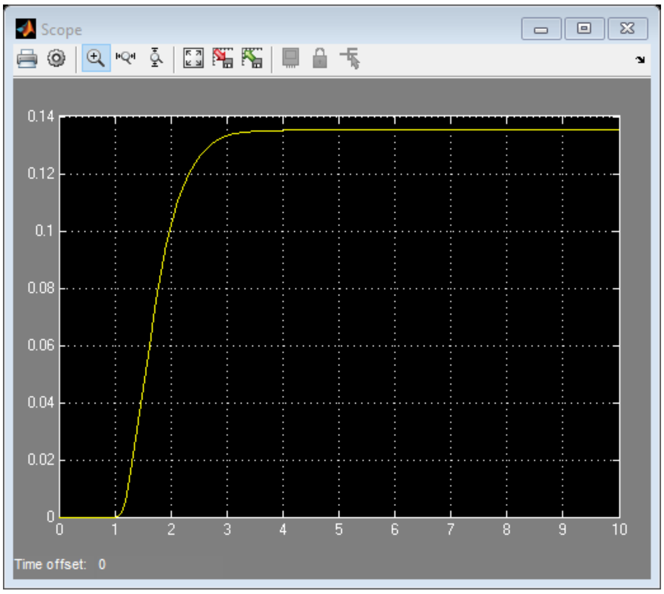


Figure (4.3.a) closed loop response without gain

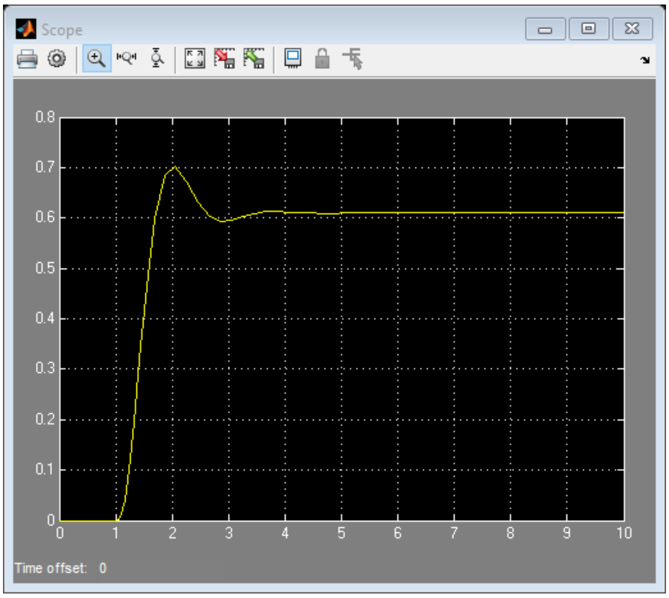


Figure (4.3.b) closed loop response with K=10

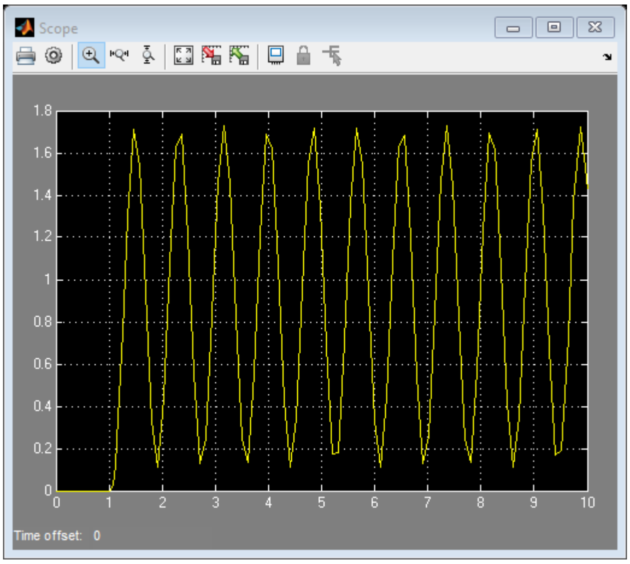


Figure (4.3.c) closed loop response with K=72.

4.3 Tuning methods:

4.3.1 Ziegler Nichols Open loop tuning method:

In 1942 J.G Ziegler and N.B. Nichols derived their first method of PID tuning through empirical testing. This method was based on the plant reaction to a step input and characterized by two parameters. The method is often referred to as the Open Loop, or Step Response tuning method. The parameters T, a and L is determined by applying a unit step function of the process. [11]

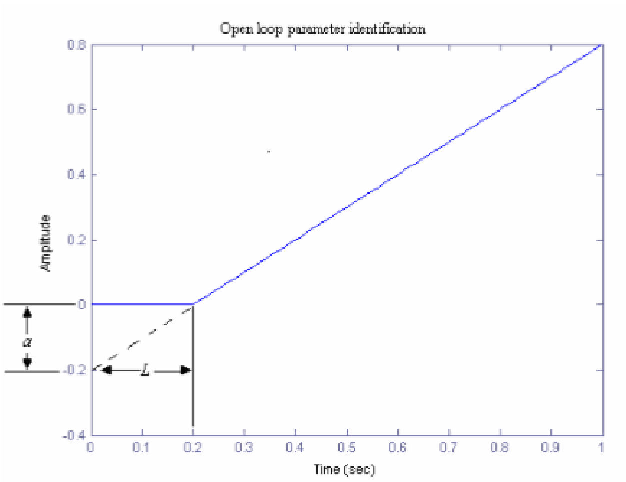


Figure (4.4) Open loop parameter identification

Ziegler and Nichols suggested that set the values of the parameters K_p T_i and T_d according to the formula shown in Table (4.1)

Table (4.1) opens loop tuning method parameters

Controller	K_p	T_i	T_d
P	***1/a***	∞	***0***
PI	***0.9/a***	***3L***	
PID	***1.2/a***	***2L***	***L/2***

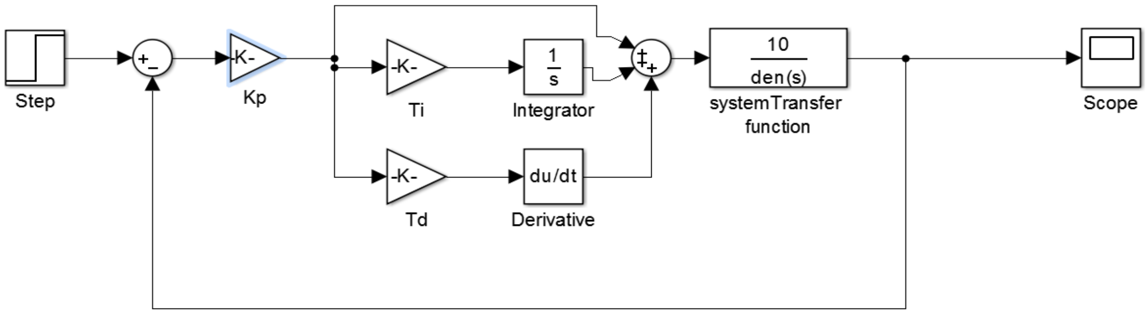
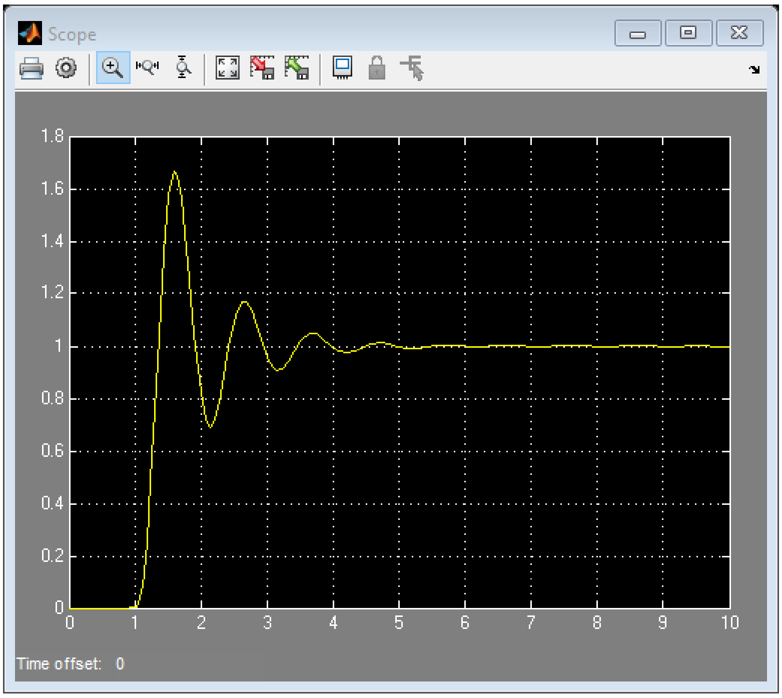


Figure (4.5): Block diagram of the system using simulink

Applying this method on the system give the value



K_p	T_i	T_d
41.522	0.394	0.085

Figure (4.6) system response using a PID controller tuned by the open loop method

4.3.2 Ziegler Nichols Closed loop tuning method:

In this method, firstly set T, = ∞ and Td = 0. Using the proportional control action only (see Figure 4.7), increase K, from 0 to a critical value K_{cr} at which the output first exhibits sustained oscillations. (If the output does not exhibit sustained oscillations for whatever value K, may take, then this method does not apply.)

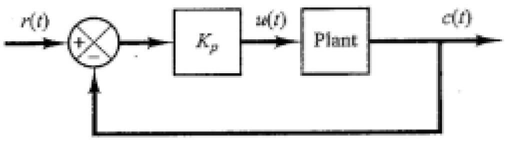


Figure (4.7) Closed-loop system with a proportional controller.

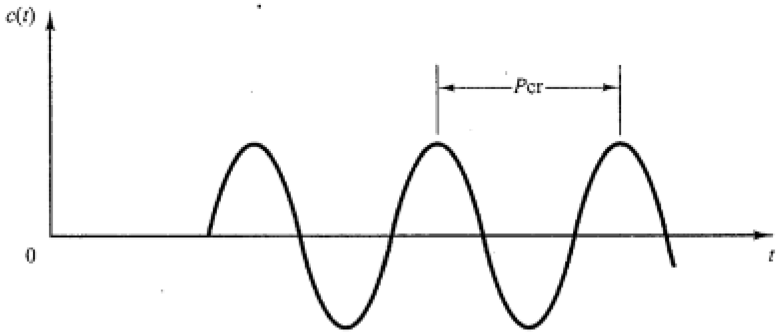


Figure (4.8) Sustained oscillation with period P_{cr}.

Thus, the critical gain K_{cr}, and the corresponding period P_{cr}, are experimentally determined (see Figure 4.8). Ziegler and Nichols suggested that set the values of the parameters K_p T_i and T_d according to the formula shown in Table (4.2). [11]

Table (4.2): Ziegler-Nichols Tuning Rule Based on Critical Gain Kc, and Critical Period P_{cr}

Controller	K_p	T_i	T_d
P	***0.5Kcr***	∞	***0***
PI	$\mathbf{0.45K_{cr}}$	$\mathbf{P_{cr}/1.2}$	***0***
PID	$\mathbf{0.6K_{cr}}$	$\mathbf{0.5P_{cr}}$	$\mathbf{0.125P_{cr}}$

As shown in the closed loop response from the root locus diagram can we extract that K_{cr}=72 and P_{cr}=0.8

K_p	T_i	T_d
43.2	0.4	0.1

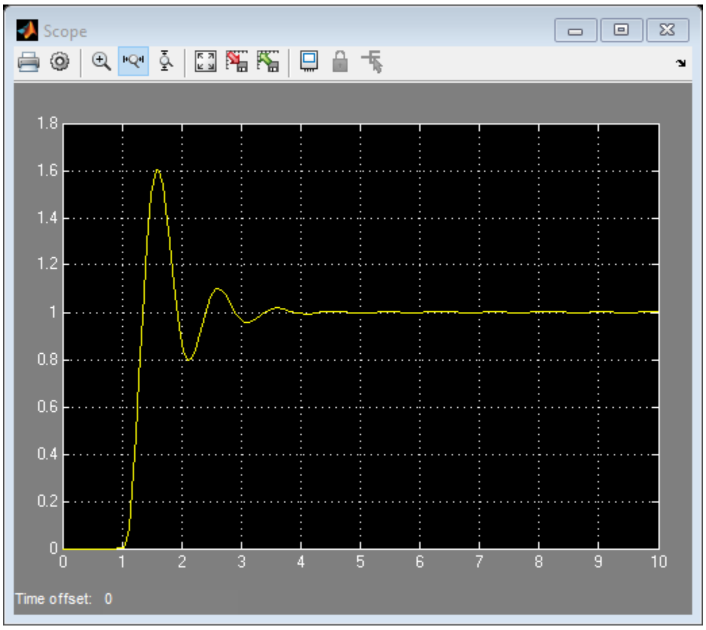


Figure (4.9): system response using a PID controller tuned by the closed loop method

4.3.3 Cohen-Coon tuning method:

This method also likes Ziegler Nichols open loop method also referring to the open loop response (FOPTD) in figure (4.1).

The different controllers can be designed with the direct use of Table (4.3). [12]

$$a = \frac{KL}{T} \qquad \tau = \frac{L}{L+T}$$

Table (4.3) controller parameters of Cohen–Coon method.

Controller	K_p	T_i	T_d
P	$\frac{1}{a}(1+\frac{0.35\tau}{1-\tau})$	∞	0
PI	$\frac{0.9}{a}(1+\frac{0.92\tau}{1-\tau})$	$\frac{3.3-3\tau}{1+1.2\tau}L$	0
PD	$\frac{1.24}{a}(1+\frac{0.13\tau}{1-\tau})$	∞	$\frac{0.27-3.6\tau}{1-0.87\tau}L$
PID	$\frac{1.25}{a}(1+\frac{0.18\tau}{1-\tau})$	$\frac{3.3-\tau}{1+1.2\tau}L$	$\frac{0.37-0.37\tau}{1-0.81\tau}L$

The PID controller parameters of the system

K_p	T_i	T_d
44.69	0.522	0.0704

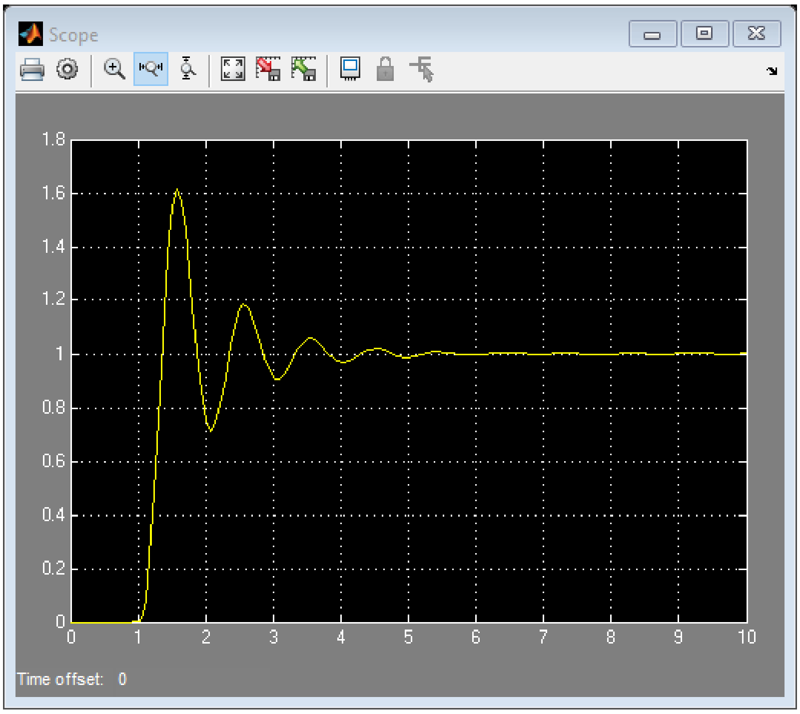


Figure (4.10) system response using a PID controller tuned by the Cohen coon method

4.3.4 Chien–Hrones–Reswick PID Tuning:

This method that has proposed by Chien, Hrones and Reswich, it's a modification of open loop Ziegler and Nichols method. They proposed to use “quickest response without overshoot” or “quickest response with 20% overshoot” as a design criterion. They also made the important observation that tuning for set point responses and load disturbance responses are different. To tune the controller according to the C- H-R method the parameters of first order plus dead time model are determined in the same manner of the Z-N method. The controller parameters can then be determined from the Tables (4.4) and (4.5). The tuning rules based on the 20% overshoot design criterion are quite similar to the Z-N method. However, when the 0% overshoot criteria is used, the gain and the derivative time are smaller and the integration time is larger. This means that the proportional action and the integral action, as well as the derivative action, are smaller. [12]

Table (4.4) CHR is tuning for set-point regulation

Controller	With 0% overshoot			With 20% overshoot		
	K_p	T_i	T_d	K_p	T_i	T_d
P	0.3/a	∞	0	0.7/a	∞	0
PI	0.35/a	1.2T	0	0.6/a	T	0
PID	0.6/a	T	0.5L	0.95/a	1.4T	0.47L

Table (4.5) CHR is tuning for disturbance rejection

Controller	With 0% overshoot			With 20% overshoot		
	K_p	T_i	T_d	K_p	T_i	T_d
P	0.3/a	∞	0	0.7/a	∞	0
PI	0.6/a	4L	0	0.7/a	2.3L	0
PID	0.95/a	2.4L	0.42L	1.2/a	2L	0.42L

As shown the second table give values almost equal ZN open loop method We can neglect it and design the controller from the first table. Set point regulation 0% overshoot:

K_p	T_i	T_d
20.76	1.063	0.0985

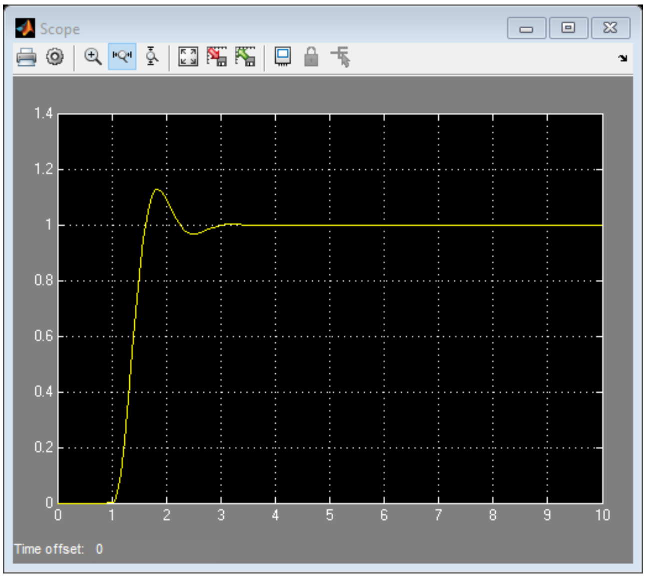


Figure (4.11) system with PID controller tuned using CHR 0% overshoot method

And 20% overshoot:

K_p	T_i	T_d
32.87	1.4882	0.09259

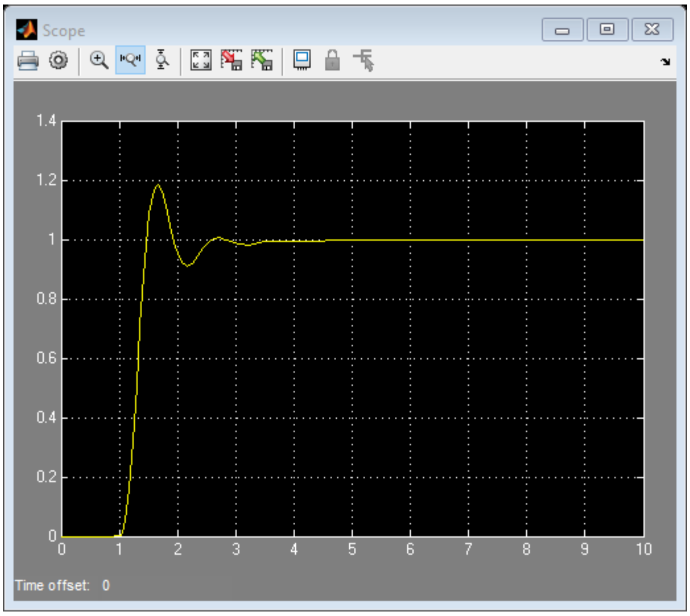


Figure (4.12) system with PID controller tuned using CHR 20% overshoot method

4.3.5 TheWang–Juang–Chan Tuning method:

Based on the optimum ITAE criterion, the tuning algorithm proposed by Wang, Juang, and Chan is a simple and efficient method for selecting the PID parameters. [12]

If the k, L, T parameters of the plant model are known, the controller parameters are given by

$$K_p = \frac{\left(0.7303+\frac{0.5307T}{L}\right)(T+0.5L)}{K(T+L)}, \quad T_i = T + 0.5L\ , \quad T_d = \frac{0.5LT}{T+0.5L} \qquad (11)$$

Then;

K_p	T_i	T_d
19.61	1.073	0.1051

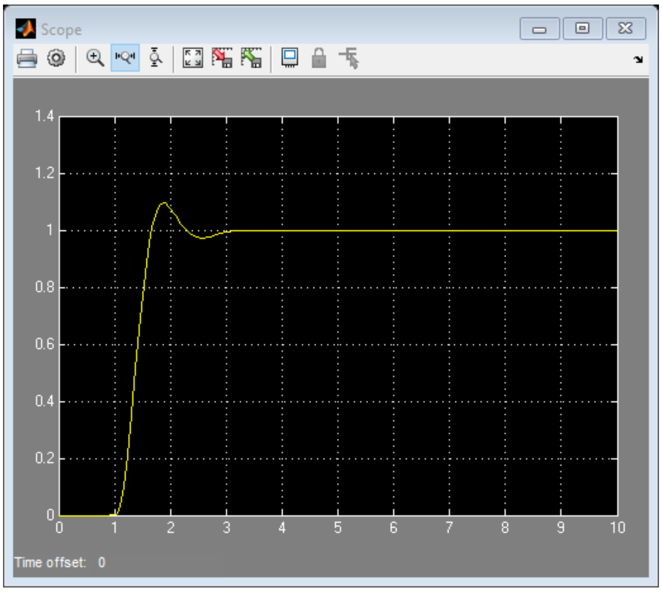


Figure (4.13) system with PID controller tuned using TheWang–gang–chain method

4.4 The results and compare:

The parameters of the PID controllers designed by the various methods shown in table (4.6)

Table (4.6)
PID

Tuning method	K_p	T_i	T_d
ZN (Open loop)	41.522	0.394	0.085
ZN (closed loop)	43.2	0.4	0.1
Cohen-Coon	44.69	0.522	0.0704
CHR 0% overshoot	20.76	1.063	0.0985
CHR 20%overshoot	32.87	1.4882	0.09259
Wang Juang Chan	19.61	1.07258	0.1051

parameters for each tuning method.

The response of the system with the each method shown in table (4.7)

Table (4.7)

Tuning method	Time rise (t_r)	Time settling (t_s)	Overshoot (MP%)
ZN (Open loop)	**0.202**	**3.38**	**55.5**
ZN (closed loop)	**0.194**	**2.67**	**49.4**
Cohen-Coon	**0.199**	**3.22**	**53.3**
CHR 0% overshoot	**0.368**	**1.97**	**8.67**
CHR 20%overshoot	**0.262**	**2.59**	**17.3**
Wang Juang Chan	**0.39**	**2.1**	**5.98**

parameter of the system response

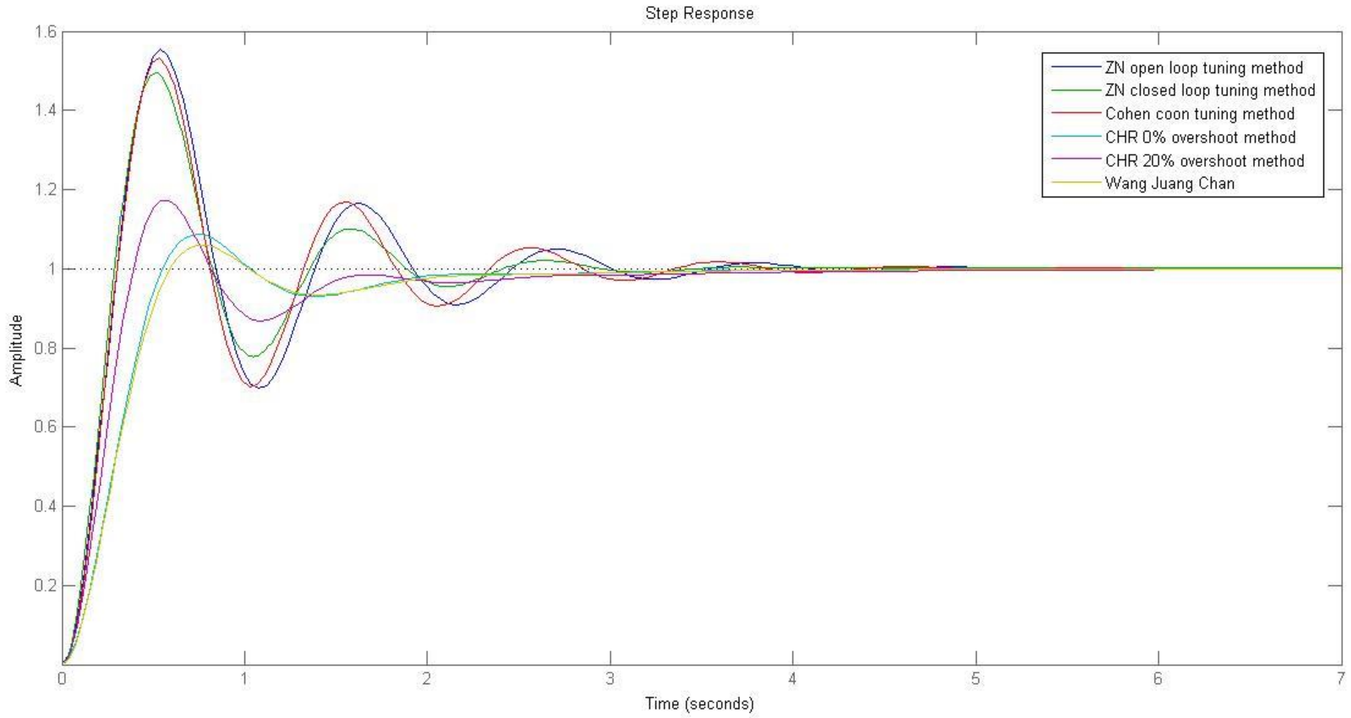


Figure (4.13) system responses all curves

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion:

In this thesis, studying the PID theory, design of different form of the controller and evaluated five different methods used in industry to tune PID controllers. The five methods were chosen because they are the more popular ones, also it wanted to evaluate open loop methods as well as closed loop methods of tuning, this tuning PID controller rules just work with the ideal structure of PID.

After using simulation and get the result it notice that some method gives a big value in overshoot and great time rise so to choose the suitable method of the system most determine the desired specification may be the result from the simulation is not equal the real system result because the simulation environment is perfectly opposite the reality.

It was concluded that PID controller is still great and important. It's deserve more focus and investigations to get most accuracy tuning methods that will help in all fields that need it.

5.2 Recommendations:

- Tuning the PID controller for a plant that unknown transfer function.