# CHAPTER ONE

# INTRODUCTION

## 1.1 General Concepts

The PID controller is the most common form of feedback in use today. It was an essential element of early governors and it became the standard tool when process control emerged in the 1940. In process control today, more than 95% of the control loops are of PID type, most loops are actually PI control. PID controllers are today found in all areas where control is used. The controllers come in many different forms. There are stand-alone systems in boxes for one or a few loops, which are manufactured by the hundred thousand yearly. PID control is an important ingredient of distributed control system. The controllers are also embedded in many special-purpose control systems. The family of PID controllers is rightly known as the building blocks of control theory owing to their simplicity and ease of implementation. A PID controller attempts to correct the error between a measured process variable and a desired set point by calculating and then outputting a corrective action that can adjust the process accordingly. PID controller has the optimum control dynamics including zero steady state error, fast response (short rise time), no oscillations and higher stability. The adjustment of the controller parameters is called the controller design or the controller tuning. A conventionally tuned PID controller with fixed parameters may usually derive lesser control performance when it comes to system demands. [1]

## 1.2 Problem Statement

Due to the PID applications in industries it will be very important topic to adjust the PID controller to operate in proper way without any losses in production**.**

## 1.3 Objectives

-To study the PID controller as a hardware tool.

-To study the tuning methods used in the PID adjustment.

-To apply these methods using MATLAB simulation.

-To select the best method used in PID adjustment.

## 1.4 Methodology

A MATLAB simulation had been carried out in this research to covert the tuning methods.

## 1.5 Project Layout

This research consists of an abstract and five chapters. Chapter one deals with an introduction illustrate the general concepts, problem statement, objectives and methodology. Chapter two deals with the theory of the PID controller illustrate an introduction, PID representation, proportional control, integral control, derivative control, proportional and derivative control, PID control. Chapter three discusses the methods used to tune the PID controller, chapter four deals with the investigation of these tuning methods through simulation results, and comparison was done between these results. Finally, chapter five includes the conclusion, recommendation and suggestions for future work.

# CHAPTER TWO

# PROPORTIONAL INTEGRAL DERIVATIVE CONTROLLER

## 2.1 Introduction

PID 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 controller have proved satisfactory, it still has a wide range of applications in industrial control. PID control is often combined with logic, sequential functions, selectors, and simple function blocks to build the complicated automation systems used for energy production, transportation, and manufacturing .PID control has been an active research topic for many years. Since many process plants controlled by PID controller has been found possible to set satisfactory controller, parameters from less plant. Information than a complete mathematical model. The concept of a control system is to sense deviation of the output from the desired value and correct it, till the desired output is achieved. The deviation of the actual output from its desired value called an error. The measurement of error is possible because of feedback. The feedback allows comparing the actual output with its desired value to generate the error. The error is denoted as e (t).

The desired value of the output is also called reference input or a set point .The error obtained is required to be analysis to take proper corrective action. The two most popular PID techniques were the step reaction curve experiment and a closed-loop cycling experiment under proportional control around the nominal operating point. These techniques come about because of the desire to adjust controller parameters in situ with a minimum effort, and also because of obtaining mathematical models. Atypical structure of a PID control system is shown in Figure 2.1, where it can be seen that in a PID controller, the error signal e (t) is used to generate the proportional integral, and derivative actions, with the resulting signals weighted and summed to form the control signal u (t) applied to the plant model.

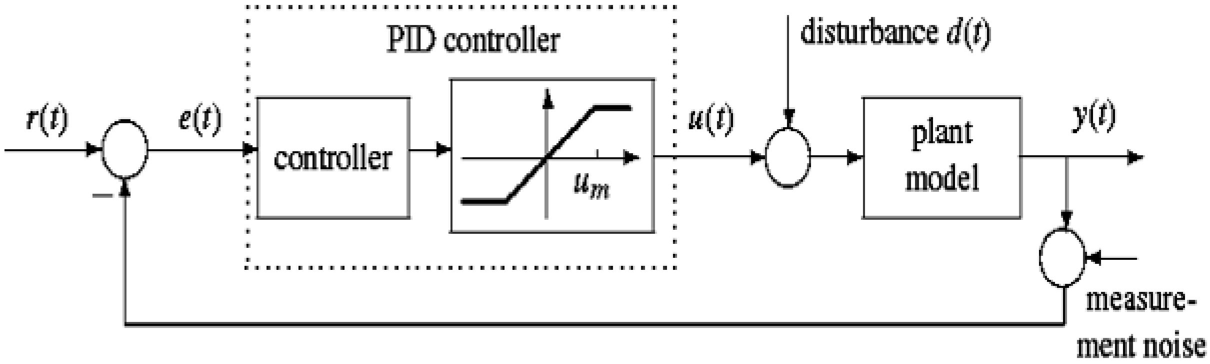


Figure 2.1: Atypical PID control structure

PID controller remains an important control tool for three reasons past: record of success, wide availability and simplicity in use these reasons reinforce one another thereby ensuring that the more general frame work of digital control with higher order controllers has not really been able to displace PID control. It is really only when the process situation demands a more sophisticated controller or a more involved controller solution to control a complexity of the process demands a multi-loop or multivariable control solution, a network based on PID control building blocks is often used.[1]

## 2.2 PID Representation

A PID controller involves three terms: the Proportional term designated as Kp, the Integral term designated as Ki/s, and the Derivative term designated as sKd.Applying a PID control law consists of applying properly the sum of three types of control actions: a proportional action, an integral action and a derivative one.

A mathematical description of the PID controller in the time-domain is given by the following equation:

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t)dt + T_d \frac{de(t)}{dt} \right] \quad (2.1)$$

Where u(t) is the controller output (input signal to the plant model), the error signal e(t) is defined as e(t) = r(t) – y(t) ,and r(t) is the reference input signally(t) is the output .The control variable is thus a sum of three terms: the P-term (which is proportional to the error), the I-term (which is proportional to the derivative of the error). The controller parameters are proportional gain$K_p$, integral time Ti and derivative time Td. The integral gain (Ki), and the derivative gain (Kd) can be expressed as:

$$Ki = \frac{Kp}{Ti} \quad (2.2)$$

$$Kd = Kp*Td \quad (2.3)$$

The PID controller is quite sophisticated and three different representations can be given. First, there is a symbolic representation Figure 2.2(a), where each of the three terms can be selected to achieve different control actions. Secondly, there is a time domain operator form Figure 2.2(b), and finally, there is a Laplace transform version of the PID controller Figure 2.2(c). This gives the controller an *s*-domain operator interpretation and allows the link between the time domain and the frequency domain to enter the discussion of PID controller performance.

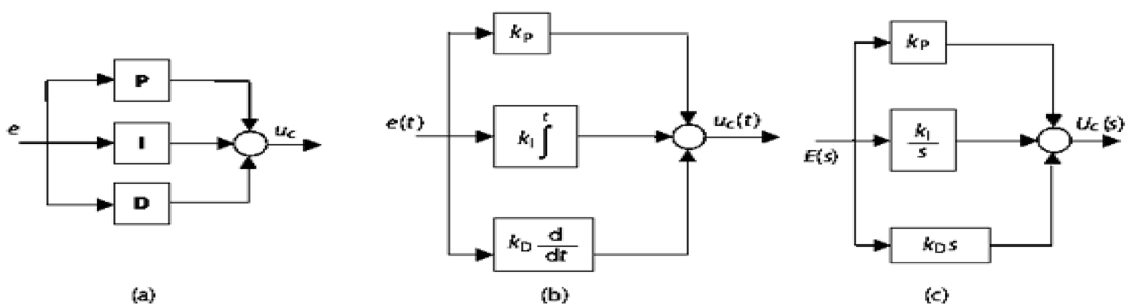


Figure 2.2: PID controller representation

## 2.3 Proportional Control

Proportional control is denoted by the P-term in the PID controller. It is used when the controller action is to be proportional to the size of the process error signal

$$e(t) = r(t) - y(t) \qquad (2.4)$$

The time and Laplace domain representations for proportional control are given as:

$$\text{time domain} \quad u_c(t) = K_p * e(t) \qquad (2.5)$$

$$\text{Laplace domain} \quad U_c(s) = K_p * E(s) \qquad (2.6)$$

Where the proportional gain is denoted Kp. Figure 2.3 shows the block diagrams for proportional control.

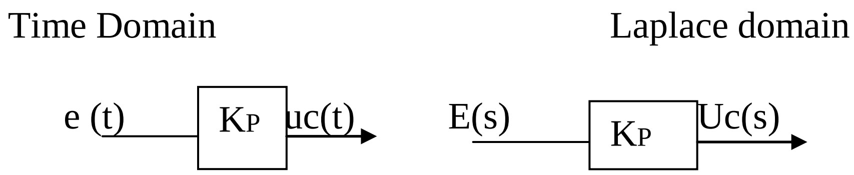


Figure 2.3: Block diagrams of proportional control term.

The simplest idea is that the compensation signal (actual controller output) is proportional to the error e (t). It is obvious that the value of the Kp determines the controller “sensitivity” how much compensation to enact for a given change in error. For all commercial devices, the proportional gain is a positive quantity .Because the negative feedback as shown in Figure 2.1, the controller output moves in the reverse direction of the controller variable. In the liquid level control for example, if the inlet flow is disturbed such that the level rises above the set point, then e<o, and that leads the controller output to decrease. In this case, the selecting or purchasing a valve such that a lowered signal means opening the valve (decreasing flow resistance). Mathematically, this valve has a negative steady state gain. For a positive gain increased signal means opening the valve. In this case, a negative proportional gain is needed. Commercial devices provide such a “switch” on the controller box to invert the signal mathematically, the sign of the compensation term is changed. By

the definition of control problem, there should be no error at t=0, and the deviation variable of the error is simply the error itself. Generally, the proportional gain is dimensionless (i.e., $u_c(t)$ and e(t) have the same units). Many controller manufacturers use the percent Proportional Band, which is defined as:

$$PB = \frac{100}{Kp} \tag{2.7}$$

A high proportional gain is equivalent to narrow PB, and a low gain is wide PB. We can interpret PB as the range over which the error must change to drive the controller output over its full range. The general qualitative features of proportional control are:

- The proportional controller will improve or accelerate the response of a process. the larger Kp is, the faster and more sensitive is the change in the compensation with respect to given error. However, if the Kp is too large, the control compensation to overreact, leading to oscillatory response. In the worst case scenario, the system may become unstable.
- There are physical limits to a control mechanism. A controller (like an amplifier) can deliver only so much voltage or current; a valve can deliver only so much fluid when fully opened. At these limits, the control system is saturated.
- The system with only a proportional controller to have a steady-state error (or an offset). This is one simplistic way to see why. Let's say we change the system to a new set point. The proportional controller output is required to shift away from the previous bias and move the system to a new steady-state.
- To tackle a problem considers a simple proportional controller first. This may be all we need (lucky break) if the offset is small enough (for us to bear with) and the response is adequately fast. Even if this is not the case, the analysis should help us plan the next step.

## 2.4 Integral Control

Integral control is denoted by the I-term in the PID control is used when it is required that the controller correct for any steady offset from a constant reference signal value. Integral control overcomes the shortcoming of proportional control by eliminating offset without the use of excessively large controller gain.The time and Laplace domain representations for integral control are given as:

Time domain $u_c(t)$ = Ki $\int_0^T e(t)dt$ (2.8)

Laplace domain $U_c(s)$ = $[\frac{Ki}{S}]$ E(s) (2.9)

Where the integral controller gain is denoted Ki. The time and Laplace block diagrams are shown in Figure 2.4

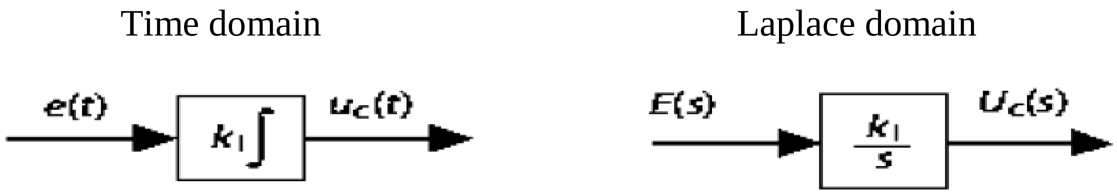


Figure 2.4: Block diagrams of integral controller

The integral action accumulated the error from t=0 to the present value. Thus the integral is not necessarily zero even if the current error is zero Moreover; the value of the integral will not decrease unless the integrand e'(t) changes its sign. As a result, integral action forces the system to overcompensate and leads tooscillatory behavior, i.e., the closed –loop system will exhibit an under damped response. If there is too much integral action, the system may become unstable. If the error cannot be eliminated within a reasonable period, the integral term can become so large that the controller is saturated, this may happen during start-up or large set point changes. It may also happen during start-up or large set point changes. It may also happen if the proportional gain is too small. The main function of the integral action is to make sure that the process output agrees with the set point in steady state. With proportional control, there is normally a control error in steady state. With integral action, a small positive error will always lead to an increasing control signal, and a negative error will give a decreasing control signal no matter how small the error is. Integral action can also be visualized as a device that automatically resets the bias term of a proportional controller. This is illustrated in the block diagram in Figure 2.5, which shows a proportional controller with a reset that is adjusted automatically. The adjustment is made by feeding bock a signal, which is a filtered value of the output, to the summing point of the controller. This was actually one of the early inventions of integral action, or "automaticreset" as it was also called.

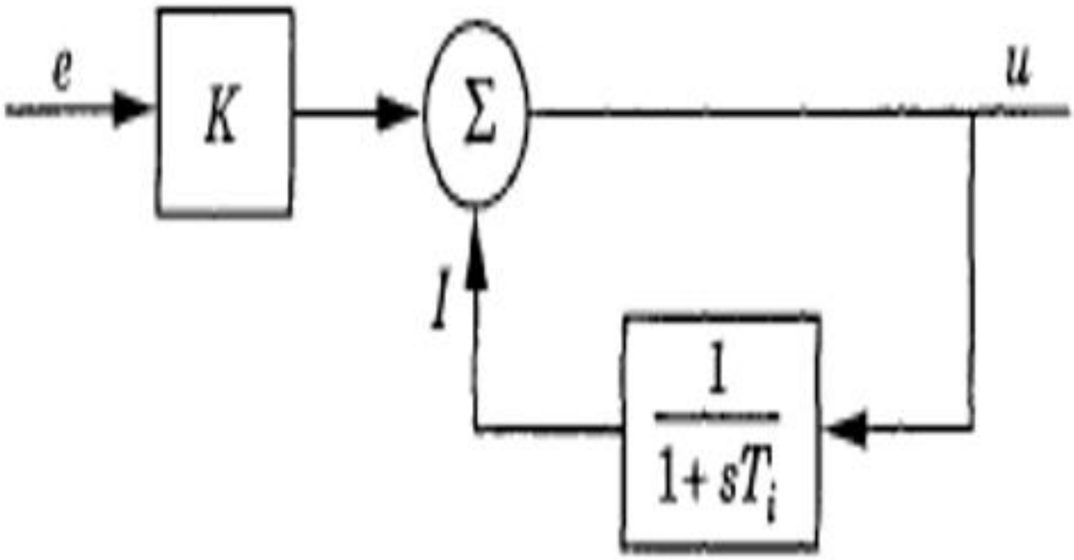


Figure 2.5: Implementation of integral action as Positive feedback around a lag.

In practice, integral action is never used by itself. The norm is a proportional-integral (PI) controller. The time –domain equation and the Laplace transform of (PI) are given by:

Time domain $u_c(t)$ =Kp [e (t) +$\frac{1}{Ti}\int_0^t e(t)dt$] (2.10)

Laplace domain $U_c(s)$ = Kp [1+$\frac{1}{Tis}$ ] (2.11)

The General qualitative features of (PI) control are:

- PI control can eliminate offset. By using a PI controller in the design if the offset is unacceptably large.
- The elimination of the offset is usually at the expense of a more under damped system response. The oscillatory response may have a short rise time, but is penalized by excessive overshoot or exceedingly long settling time.
- Because of the inherent under damped behavior, be careful with the choice of the proportional gain. In fact, usually lower the proportional gain (or detune the controller) when adding integral control.
- Derivative action is frequently not used. It is an interesting observation that many industrial controllers only have PI action and that in others the derivative action can be switched off. It can be shown that PI control is adequate for all processes where the dynamics are essentially of the first order. It is fairly easy to find out if this is the case by measuring the step response or the frequency response of the process. If the step response looks like that of a first order system or, more precisely, it the nyquist curve lies in the first and the fourth quadrants only, then PI control is sufficient. Another reason is that the process has been designed so that its operation does not require tight control. Then, even if the process has higher-order dynamics, what it needs is an integral action to provide zero steady-state offset and an adequate transient response by proportional action.

## 2.5 Derivative Control

The purpose of the derivative action is to improve the closed-loop stability. Derivative control uses the rate of change of an error signal and is the D term in the PID controller. The time and Laplace domain representation for derivative control are given as:

Time domain $u_c(t) = \mathrm{Kd}\frac{de(t)}{dt}$ (2.12)

Laplace domain $U_c(s)$ = Kd*s*E(s) (2.13)

Where the derivative control gain is denoted Kd. This particular form is termed pure derivative control, for which the block diagram representations are shown in Figure 2.6

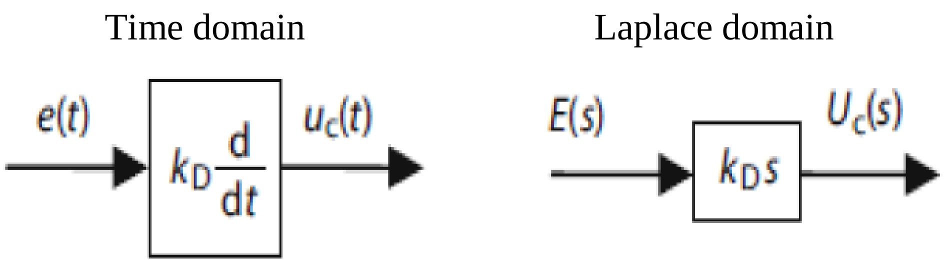


Figure 2.6: Block diagrams of derivative control term.

The derivative term D is proportional to the time derivative of the control error. This term allows prediction of the future error. To use derivative control more care is needed than when using proportional or integral control. For example, in most real applications a pure derivative control terms cannot be implemented due to possible measurement noise amplification and a modified term has to be used instead. However, derivative control has useful design features and is an essential element of some real-world control applications: for example, tachogenerator feedback in DC motor control is a form of derivative control.

## 2.6 Proportional and Derivative Control

A property of derivative control that should be noted arises when the controller input error signal becomes constant but not necessarily zero, as might occur in steady-state process conditions. In these circumstances, the derivative of the constant error signal is zero and the derivative controller produces no control signal. Consequently the controller is taking no action and is unable to correct for steady-state offsets. The derivative control term is always used in combination with a proportional term. This combination is called (Proportional Derivative), or PD control. The formula for simple PD controller is given as:

$$u_c(t) = \text{Kp*e (t)} + \text{Kd}\,\frac{de(t)}{dt} \qquad (2.14)$$

$$U_c(s) = [\text{Kp+Kd*S}]\ \text{E(s)} \qquad (2.15)$$

Where the proportional gain is Kp and the derivative gain is Kd. The Block diagrams for simple PD controllers are given in Figure 2.7

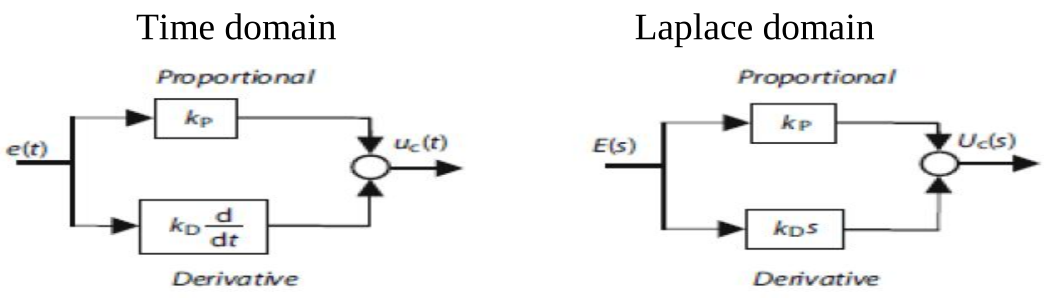


Figure 2.7: Block diagrams of PD control

The General qualitative features of PD control are:

- PD control is not useful for systems with large dead time or noisy signals.
- The sign of the rate of change in the error could be opposite that of the proportional or integral terms. Thus adding derivative action to control PI may counteract the overcompensation of the integration action. PD control may improve system response while reducing oscillations and the overshoot. (Formal analysis later will show that the problem is more complex than this simple statement).
- If simple proportional control works fine (in the sense of acceptable offset), we may try PD control. Similarly, we may try PID on top of PI control.The additional stabilizing actions allow us to a larger proportional gain and obtain a faster system response.

## 2.7 Proportional-Integral-Derivative Control

Finally, all the components together make a three term PID controller. The family of PID controllers is constructed from various combinations of the Proportional, Integral and Derivative terms as required to meet specific performance requirements. the formula for the basic parallel PID controller is

$$U_c(s) = [Kp + \frac{Ki}{s} + Kd*s]\ E(s) \qquad (2.16)$$

This controller formula is often called the textbook PID controller form because it does not incorporate any of the modifications that are usually implemented to give a working PID controller. For example, the derivative term is not usually implemented in the pure form due to adverse noise amplification properties. Other modifications that are introduced into the PID control include those used to deal with behavior that arises because the PID controller operates directly on the reference error signal. This formula is also known as parallel or decoupled PID form this is because the PID controller has three decoupled parallel paths. A numerical change in any individual coefficient, Kp, Ki or Kd changes only the size of the contribution in the path of the term. For example if the value of Kd is changed (decoupled) and independent from the size of the proportional and integral terms. not all manufacturers produce PID's that conform to the ideal structure. So before commencing tuning it is important to know the configuration of the PID algorithm. The majority of tuning rules are only valid for the ideal architecture. If the algorithm is different than the controller parameters suggested by a particular tuning methodology will have to be altered. The Ideal PID controller mathematical representation of algorithm is:

$$\frac{U(s)}{E(s)} = Kp\ [1 + \frac{1}{Ti*s} + Td*s] \qquad (2.17)$$

Where U(s) is the controller output. One disadvantage of this ideal configuration is that a sudden change in set point (and hence e) will cause the derivative term to become very large and thus provide a "derivative" to the final control Element which is undesirable.

An alternative implementation is:

$$U(s) = Kp\ [1+\frac{1}{Ti*s}]\ E\ (s) + Td*s\ Y\ (s) \qquad (2.18)$$

Where Y(s) is the system output. The derivative mode acts on the measurement and not the error .After a change is set point. The outputs will more slowly avoiding after set point changes. This is therefore a standard feature of the most commercial controllers. Finally, PID controller has been widely used to control industrial process loops because of its implementation advantages. However, it is very difficult to achieve an optimal PID gain with no experience since the parameters of the PID controller has to be manually tuned by trial and error. This research focuses on some methods used to determine or to set the parameters of the PID (Kp, Ki, and Kd) to set optimal controller**.**

# CHAPTER THREE

# PID CONTROLLER TUNING METHODS

## 3.1 Introduction

It is interesting to note that more than half of the industrial controllers used today utilize PID or modified PID control schemes. Analog PID controllers are mostly hydraulic, pneumatic, electric, and electronic types or their combinations. Currently, many of these controllers are transformed into digital forms through the use of microprocessors. Because most of PID controllers are adjusted in field, by using these tuning methods delicate and fine tuning of PID controllers can be made in field industry. Also automatic tuning methods of the PID controller have been developed and some of the PID controllers may possess on line automatic tuning capabilities. Modified forms of PID control, such as I-PD control and two degrees of freedom PID control, are currently in use in industry. Many practical methods for switching (from manual to automatic operation) and gain scheduling are commercially available. The usefulness of PID controllers lies in their general applicability to most control systems, in the field of process controls systems, it is a well-known fact that the basic and modified PID control schemes have proved their usefulness in providing satisfactory control, although they may not provide optimal control in many given situations. All general methods for control design can be applied to PID control. A number of special methods that are tailor made for PID control have also been developed, these methods are often called tuning methods. Irrespective of the method used, it is essential to always consider the key elements of control, load disturbances, sensor noise, process uncertainty and reference signals. To obtain rational methods for designing controllers it is necessary to define the main purpose of the control system, and the design methods differ with respect to the knowledge of the process dynamics they require. A PI controller is described by two parameters (Kp and Ti) and a PID controller by three or four parameters (Kp, $T_i$, $T_d$, and

N). The classical Ziegler-Nichols methods are characterized by two parameters. One parameter is related to the process gain and the other describes how fast the process is. In the step response method, the parameters are simply obtained from the step response. In the frequency response method, the parameters are: the ultimate gain and the ultimate frequency. An obvious extension of the frequency response method is to develop methods that are based on more knowledge of the open-loop transfer function, e.g., the slope of the transfer function or its values at two or more frequencies. Also we discuss various methods that are based on attempts to shape the loop transfer function.

## 3.2 Tuning Methods for PID Controller

In control designs it is often convenient to have a few parameters that can be changed to influence the performance of the system. The parameters should be chosen in such a way that their influence on the performance of the system is transparent. Figure 3.1 shows a PID control of a plant.

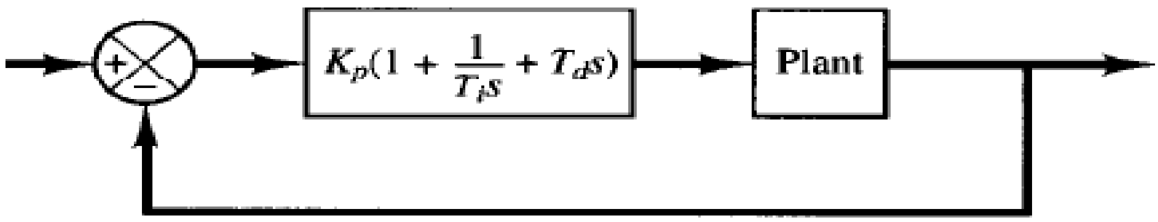


Figure 3.1: PID control of a pant

If a mathematical model of the plant can be derived, then it is possible to apply various design techniques for determining parameters of the controller that will meet the transient and steady-state specifications of the closed-loop system. However, if the plant is so complex that its mathematical model cannot be easily obtained, then analytical approach to design PID controller is not possible. Then the experimental approaches to tuning PID controllers must be restored.

### 3.2.1 Ziegler-Nichols tuning methods

Ziegler and Nichols proposed methods for determining values of the proportional gain $K_P$, integral time $T_i$, and derivative time $T_d$ based on the transient response characteristics of a given plant. Such determination of the parameters of PID controllers or tuning of PID controllers can be made by engineers on site by experiments on the plant. There are two methods known as Ziegler-Nichols tuning methods. In both methods, they aimed at obtaining 25% maximum overshoot in step response, as shown in Figure 3.2.

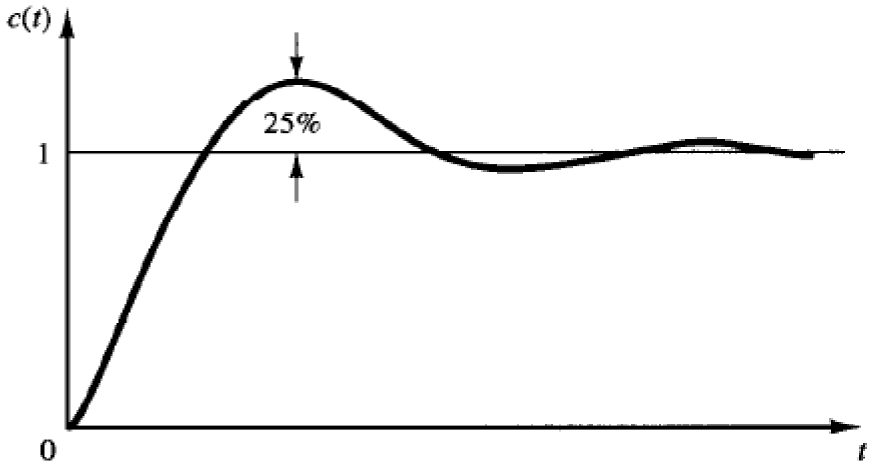


Figure 3.2: Unit step response curve showing 25% maximum overshoot

First method: In this method, obtain experimentally the response of the plant to a unit step input, as shown in Figure 3.3. If it involves neither integrator nor dominant complex-conjugate poles, then such a unit step response curve may look like an S-shaped curve, as shown in Figure 3.4. If the response does not exhibit an S-shaped curve, this method does not apply. Such step-response curves may be generated experimentally or from a dynamic simulation of the plant. The S-shaped curve may be characterized by two constants, delay timeL and time constant T.

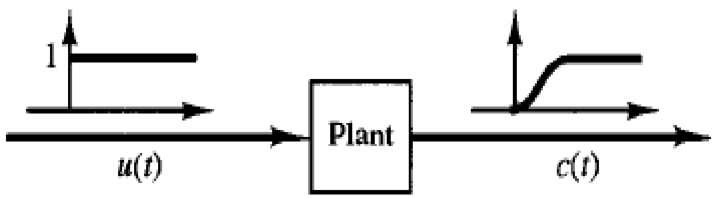


Figure 3.3: Unit step response of the plant

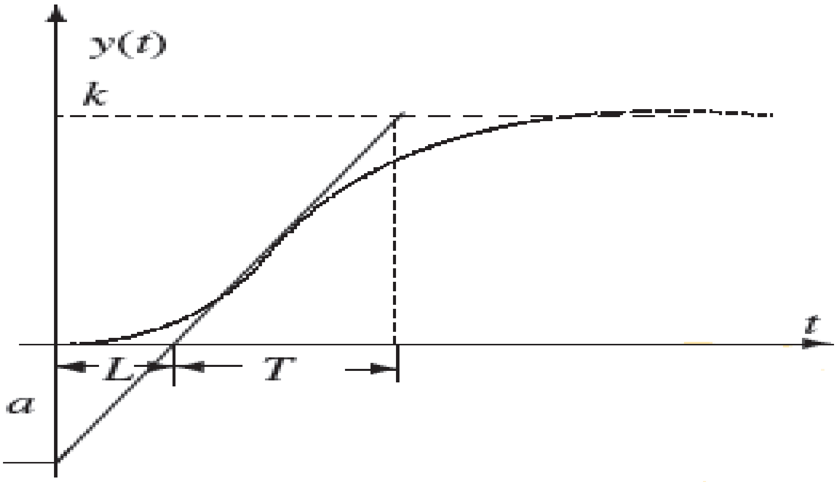


Figure 3.4: S-shaped response curve

The delay time and time constant are determined by drawing a tangent line at the inflection point of the S-shaped curve and determined the intersections of the tangent line with the time axis and line c(t)=k, as shown in Figure 3.4. The transfer function C(s)/U(s) may then be approximated by a first-order with a transport lag as follows:

$$\frac{C(s)}{U(s)} = \frac{ke^{-Ls}}{Ts+1} \qquad (3.1)$$

Ziegler-Nichols suggests setting the values of $K_P$, $T_i$, and $T_d$ according to the formula shown in table 3.1. Notice that the PID controller tuned by the first method of Ziegler-Nichols methods gives:

$$G_c(s) = k_p(1 + \frac{1}{T_i s} + T_d s) \qquad (3.2)$$

By using the values of ($K_P$, $T_i$, and $T_d$) in table (3.1):

$$G_c(s) = 1.2\frac{T}{L}(1 + \frac{1}{2Ls} + 0.5Ls)$$

$$G_c(s) = 0.6T\frac{(s + \frac{1}{L})^2}{s} \qquad (3.3)$$

Table 3.1: Ziegler-Nichols tuning method (First method)

| Type of controller | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| P | $\frac{T}{L}$ | $\infty$ | 0 |

| | | | |
|---|---|---|---|
| PI | $0.9\frac{T}{L}$ | $3L$ | 0 |
| PID | $1.2\frac{T}{L}$ | $2L$ | $0.5L$ |

Thus, the PID controller has a pole at the origin and double zero at s = -1/ L.

Second method: In this method, set Ti = ∞ and Td = 0. Using the proportional control action only as shown in Figure 3.5.

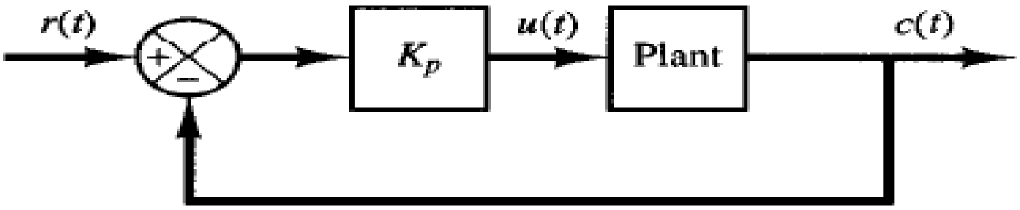


Figure 3.5: Closed-loop system with a proportional controller

Increase $K_p$ from 0 to a critical value $K_{cr}$ where the output first exhibits sustained oscillations. If the output does not exhibit sustained oscillations for whatever the value of $K_p$ may take, then this method does not apply; Thus the critical gain $K_{cr}$ and the corresponding period $p_{cr}$ are experimentally determined as shown in Figure 3.6.

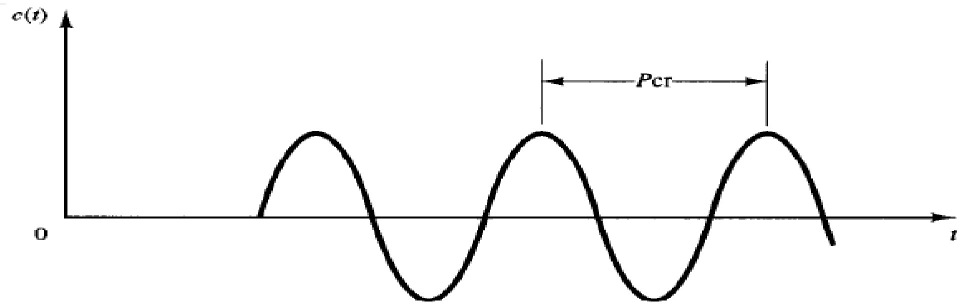


Figure 3.6: Sustained oscillations with period $p_{cr}$

Ziegler-Nichols suggests that the values of the parameters $K_p$, $T_i$, and $T_d$ must be set according to the formula shown in table 3.2.

Table 3.2: Ziegler-Nichols tuning method (Second method)

| Type of controller | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| P | $0.5k_{cr}$ | ∞ | 0 |

| | | | |
|---|---|---|---|
| PI | $0.45k_{cr}$ | $\frac{1}{1.2}p_{cr}$ | 0 |
| PID | $0.6k_{cr}$ | $0.5p_{cr}$ | $0.125p_{cr}$ |

Notice that the PID controller tuned by the second method of Ziegler-Nichols methods gives:

$$G_c(s) = k_p(1 + \frac{1}{T_i s} + T_d s) \tag{3.4}$$

By using the values of ($K_P$, $T_i$, and $T_d$) in table 3.2:

$$G_c(s) = 0.6k_{cr}(1 + \frac{1}{0.5p_{cr}s} + 0.125p_{cr}) \tag{3.5}$$

$$G_c(s) = 0.075p_{cr}k_{cr}\frac{(s + \frac{4}{p_{cr}})^2}{s} \tag{3.6}$$

Thus, the PID controller has a pole at the origin and double zero at $s = -4/p_{cr}$ is written to design PI/PID controllers by using the Ziegler–Nichols tuning methods; Ziegler-Nichols tuning methods have been widely used to tune PID controllers in process control systems where the plant dynamics are not precisely known. Over many years, such tuning methods proved to be very useful. Ziegler-Nichols tuning methods can be applied to plants whose dynamics are known and many analytical and graphical approaches to the design of PID controllers are available, in addition to Ziegler-Nichols tuning methods. If the transfer function of the plant is known, a unit step response may be calculated or critical gain Kcr and critical period Pcr may be calculated; then using those calculated values; it is possible to determine the parameters Kp, Ti, Td from table 3.1 or table 3.2. However the usefulness of Ziegler-Nichols tuning methods and other tuning methods becomes apparent when the plant dynamics are not known so that no analytical or graphical approaches to design of controllers are available. Generally, for plants with complicated dynamics but no integrators, Ziegler-Nichols tuning methods can be applied. However, if the plant has an integrator, these methods may not be applied in some cases. [2]

### 3.2.2 Chien-Hrones-Reswick PID tuning method

There have been many suggestions of modifications of the Ziegler- Nichols methods. Chien, Hrones and Reswick (CHR) changed the step response method to give better damped closed-loop systems. They proposed to use "quickest response without overshoot" or "quickest response with 20% overshoot" as design criteria ; They also made the important observation that tuning for set point response or load disturbance response are different. To tune the controller according to the CHR method, the parameters a and L of the process model are first determined in the same way as for the Ziegler-Nichols step response method. The controller parameters for the load disturbance response method are then given as functions of these two parameters. They are summarized in Table 3.3.

Table 3.3: CHR tuning method for set-point regulation

| Type of controller | With 0% overshoot | | | With 20% overshoot | | |
|---|---|---|---|---|---|---|
| | Kp | Ti | Td | Kp | Ti | Td |
| P | $\frac{0.3}{a}$ | ∞ | 0 | $\frac{0.7}{a}$ | ∞ | 0 |
| PI | $\frac{0.35}{a}$ | $1.2T$ | 0 | $\frac{0.6}{a}$ | $T$ | 0 |
| PID | $\frac{0.6}{a}$ | $T$ | $0.5L$ | $\frac{0.95}{a}$ | $1.4T$ | $0.47L$ |

The tuning method based on the 20% overshoot design criteria in Table 3.3 are quite similar to the Ziegler-Nichols step response method presented in Table 3.1. However, when the 0% overshoot design criteria is used, the gain and the derivative time are smaller and the integral time is larger ; This means that the proportional action, the integral action, as well as the derivative action, are smaller. In the set point response method, the controller parameters are not only based on a and L, but also on the time constant T. The tuning rules for set point response are summarized in Table 3.4.

Table 3.4: CHR tuning method for disturbance rejection

| Type of controller | With 0% overshoot | | | With 20% overshoot | | |
|---|---|---|---|---|---|---|
| | Kp | Ti | Td | Kp | Ti | Td |
| P | $\frac{0.3}{a}$ | $\infty$ | 0 | $\frac{0.7}{a}$ | $\infty$ | 0 |
| PI | $\frac{0.6}{a}$ | $4L$ | 0 | $\frac{0.7}{a}$ | $2.3L$ | 0 |
| PID | $\frac{0.95}{a}$ | $2.4L$ | $0.42L$ | $\frac{1.2}{a}$ | $2L$ | $0.42L$ |

### 3.2.3 The Cohen-Coon tuning method

The Cohen-Coon method is based on the process model:

$$G_p(s) = \frac{k_p}{1+sT}e^{-sL} \tag{3.7}$$

The main design criterion is rejection of load disturbances. It attempts to position dominant poles that give a quarter amplitude decay ratio. For P and PD controllers the poles are adjusted to give maximum gain, subject to the constraint on the decay ratio. This minimizes the steady state error due to load disturbances. For PI and PID control the integral gain $k_i$ = $Kp/T_i$ is maximized. This corresponds to minimization of integral error (IE) due to a unit step load disturbance. For PID controllers three closed-loop poles are assigned; two poles are complex, and the third real pole is positioned at the same distance from the origin as the other poles. The pole pattern is adjusted to give quarter amplitude decay ratio, and the distance of the poles to the origin are adjusted to minimize $I_E$. Since the process is characterized by three parameters (K, L, and T), it is possible to give tuning methods where controller parameters are expressed in terms of these parameters. Such method was derived by Cohen and Coon based on analytical and numerical computations. The formulas are given in Table 3.5

Table 3.5: Controller parameters of Cohen–Coon method

| Controller | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|

| | | | |
|---|---|---|---|
| P | $\frac{1}{a}(1+\frac{0.35\tau}{1-\tau})$ | $\infty$ | 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 | $\frac{1.24}{a}(1+\frac{0.13\tau}{1-\tau})$ | $\infty$ | $\frac{0.27-0.36\tau}{1-0.87\tau}L$ |
| 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 parameters a=K*L/T and $\tau$ = L/ (L + T) are used in the table. A comparison with Table 3.1 shows that the controller parameters are close to those obtained by the Ziegler-Nichols step response method for small $\tau$ . The method does suffer, however, from the decay ratio being too small, which means that the closed-loop systems obtained have low damping and high sensitivity.

### 3.2.4 The Wang-Juang-Chan tuning method

This tuning algorithm proposed by Wang, Juang, and Chan is a simple and efficient method for selecting the PID parameters. If the k, L, and T parameters of the plant model are known, the controller parameters are given by:

$$k_p = \frac{(0.7303+0.5307T/L+0.5L)}{k(T+L)}, \; T_i = T+0.5L \;, \; Td = \frac{0.5LT}{T+0.5L} \qquad (3.8)$$

Where the values of K, T, and L are obtained for the step response curve.

### 3.2.5 Optimum PID controller design

Optimum setting algorithms for a PID controller were proposed by Zhuang and Atherton for various criteria. Consider the general form of the optimum criterion:

$$j_n(\theta) = \int_0^\infty [t^n e(\theta,t)]^2 dt \qquad (3.9)$$

Where e (θ, t) is the error signal which enters the PID controller, with θ the PID controller parameters. Two setting strategies are proposed: one for the set-point input and the other for the disturbance signal d (t). In particular, three values of n are discussed, i.e., for n = 0, 1, 2. These three cases correspond, respectively, to three different optimum criteria: the integral squared error (ISE) criterion, integral squared time weighted error (ISTE) criterion, and the integral squared time-squared weighted error ($IST^2E$) criterion. The expressions given were obtained by fitting curves to the optimum theoretical results.

Set-point optimum PID tuning : if the plant can be represented by the First order plus dead time (FOPDT) model in equation 3.10, the typical PI controller can be empirically represented as:

$$G(s) = \frac{k}{Ts+1} e^{-Ls} \tag{3.10}$$

$$k_p = \frac{a_1}{k}\left(\frac{L}{T}\right)^{b1} \quad , \quad T_i = \frac{T}{a_2 + b_2(L/T)} \tag{3.11}$$

Where the (a, b) pairs can be obtained from Table 3.6. When the first-order approximation to the plant model can be obtained.

Table 3.6: Set-point PI controller parameters

| Range of L/T | 0.1-1 | | | 1.1-2 | | |
|---|---|---|---|---|---|---|
| Criterion | ISE | ISTE | $IST^2E$ | ISE | ISTE | $IST^2E$ |
| a1 | 0.980 | 0.712 | 0.569 | 1.072 | 0.786 | 0.628 |
| b1 | −0.892 | −0.921 | −0.951 | −0.560 | −0.559 | −0.583 |
| a2 | 0.690 | 0.968 | 1.023 | 0.648 | 0.883 | 1.007 |
| b2 | −0.155 | −0.247 | −0.179 | −0.114 | −0.158 | −0.167 |

The PI controller can be designed easily by the direct use of equation 3.12.

For the PID controller, its gains can be set as follows:

$$k_p = \frac{a_1}{k}\left(\frac{L}{T}\right)^{b1}, \ T_i = \frac{T}{a_2 + b_2(L/T)} \ , \ T_d = a_3 T\left(\frac{L}{T}\right)^{b3} \tag{3.12}$$

Where the parameters (a, b) should be determined according to Table 3.7.

Table 3.7: Set-point PID controller parameters

| Range of L/T | 0.1-1 | | | 1.1-2 | | |
|---|---|---|---|---|---|---|
| Criterion | ISE | ISTE | $IST^2E$ | ISE | ISTE | $IST^2E$ |
| a1 | 1.048 | 1.042 | 0.968 | 1.154 | 1.142 | 1.061 |
| b1 | −0.897 | −0.897 | −0.904 | −0.567 | −0.579 | −0.583 |
| a2 | 1.195 | 0.987 | 0.977 | 1.047 | 0.919 | 0.892 |
| b2 | −0.368 | −0.238 | −0.253 | −0.220 | −0.172 | −0.165 |
| a3 | 0.489 | 0.385 | 0.316 | 0.490 | 0.384 | 0.315 |
| b3 | 0.888 | 0.906 | 0.892 | 0.708 | 0.839 | 0.832 |

Disturbance rejection PID tuning : sometimes one may want to design disturbance rejection PID controllers, i.e., to design a controller having a good rejection performances on the disturbance signal d (t). The parameters of the PI controller should be set as:

$$k_p = \frac{a_1}{k}\left(\frac{L}{T}\right)^{b1} \quad (3.13)$$

$$T_i = \frac{T}{a_2}\left(\frac{L}{T}\right)^{b2} \quad (3.14)$$

Where the parameters (a, b) are obtained directly from Table 3.8.Furthermore, for the PID controller:

$$k_p = \frac{a_1}{k}\left(\frac{L}{T}\right)^{b1} \quad (3.15)$$

$$T_i = \frac{T}{a_2}\left(\frac{L}{T}\right)^{b2} \quad (3.16)$$

$$T_d = a_3 T\left(\frac{L}{T}\right)^{b3} \quad (3.17)$$

Table 3.8: Disturbance rejection PI controller parameters

| Range of L/T | 0.1-1 | | | 1.1-2 | | |
|---|---|---|---|---|---|---|
| Criterion | ISE | ISTE | $IST^2E$ | ISE | ISTE | $IST^2E$ |

| | | | | | | |
|---|---|---|---|---|---|---|
| a1 | 1.279 | 1.015 | 1.021 | 1.346 | 1.065 | 1.076 |
| b1 | −0.945 | −0.957 | −0.953 | −0.675 | −0.673 | −0.648 |
| a2 | 0.535 | 0.667 | 0.629 | 0.552 | 0.687 | 0.650 |
| b2 | 0.586 | 0.552 | 0.546 | 0.438 | 0.427 | 0.442 |

And the (a, b) parameters are determined from Table 3.9.

Table 3.9: Disturbance rejection PID controller parameters

| Range of L/T | 0.1-1 | | | 1.1-2 | | |
|---|---|---|---|---|---|---|
| Criterion | ISE | ISTE | $IST^2E$ | ISE | ISTE | $IST^2E$ |
| a1 | 1.473 | 1.468 | 1.531 | 1.524 | 1.515 | 1.592 |
| b1 | −0.970 | −0.970 | −0.960 | −0.735 | −0.730 | −0.705 |
| a2 | 1.115 | 0.942 | 0.971 | 1.130 | 0.957 | 0.957 |
| b2 | 0.753 | 0.725 | 0.746 | 0.641 | 0.598 | 0.597 |
| a3 | 0.550 | 0.443 | 0.413 | 0.552 | 0.444 | 0.414 |
| b3 | 0.948 | 0.939 | 0.933 | 0.851 | 0.847 | 0.850 |

### 3.2.6 Analytical tuning methods

There are several analytical tuning methods where the controller transfer function is obtained from the specifications by a direct calculation. Let $G_p$ and $G_c$ be the transfer functions of the process and the controller. The closed-loop transfer function obtained with error feedback is then:

$$G_o = \frac{G_p G_c}{1 + G_p G_c} \tag{3.18}$$

Solving this equation for $G_c$ we get:

$$G_c = \frac{1}{G_p} \cdot \frac{G_o}{1 - G_o} \tag{3.19}$$

If the closed-loop transfer function $G_o$ is specified and $G_p$ is known, it is thus easy to compute $G_c$. The key problem is to find reasonable ways to determine $G_o$ based on engineering specifications of the system.It follows from equation

(3.16) that all process poles and zeros are canceled by the controller. This means that the method cannot be applied when the process has poorly damped poles and zeros. The method will also give a poor load disturbance response when slow process poles are canceled. Haalman method is one of these methods which discussed in the following section.

The Haalman method:

This method used to determine an ideal loop transfer function $G_e$ that gives the desired performance and to choose the controller transfer function as:

$$G_c = \frac{G_e}{G_p} \tag{3.20}$$

Where $G_p$ is the process transfer function. Such an approach can give PI and PID controllers provided that $G_e$ and $G_p$ are sufficiently simple. There are many ways to obtain a suitable $G_e$- For systems with a time delay L, Haalman has suggested choosing:

$$G_e(s) = \frac{2}{3Ls} e^{-sL} \tag{3.21}$$

The value 2/3 was found by minimizing the mean square error for a step change in the set point. Notice that it is only the dead time of the process that influences the loop transfer function. All other process poles and zeros are canceled, which may lead to difficulties.

Applying Haalman's method to processes with the transfer function:

$$G_p(s) = \frac{1}{1+sT} e^{-sL} \tag{3.22}$$

Gives the controller:

$$G_c(s) = \frac{2(1+sT)}{3Ls} = \frac{2T}{3L}(1+\frac{1}{sT}) \tag{3.23}$$

Which is a PI controller with K = 2T/3L and $T_i$ = T. These parameters can be compared with the values K = 0.9T/L and $T_i$ = 3L obtained by the Ziegler-Nichols (first method). A PID controller is obtained if the method is applied to a process with the transfer function:

$$G_p(s) = \frac{1}{(1+sT_1)(1+sT_2)} e^{-sL} \tag{3.24}$$

The parameters of the controller are:

$$k = \frac{2(T_1 + T_2)}{3L} \tag{3.25}$$

$$T_i = T_1 + T_2 \tag{3.26}$$

$$T_d = \frac{T_1 T_2}{(T_1 + T_2)} \tag{3.27}$$

# CHAPTER FOUR

# SIMULATION AND RESULTS

## 4.1 Introduction

There are many methods to tune and obtain the parameters of the PID controller (KP, Ti, and Td). These methods to analyze the performance of the system controlled by PID controller. Figure 4.1 illustrates the overall controlled system that indicates the plant and PID controller. A given transfer function of the plant in Figure 4.1 is used, and then we apply these method, and some comparisons between them will be mode .then determining which method is the best for the given transfer function according to the results obtained from the simulation.

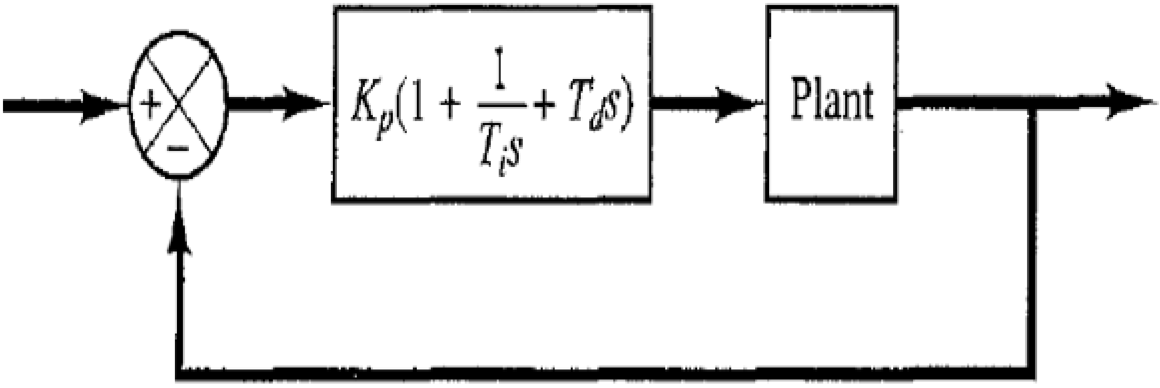


Figure 4.1: PID control of plant

The transfer function of PID controller is:

$$Gc(s)= \frac{(Kp*Ti*Td)s^2+(Kp*Ti)s+Kp}{Ti*s} \quad (4.1)$$

## 4.2MATLAB Simulation

The name MATLAB stands for Matrix Laboratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects. MATLAB is a high-performance language for technical computing.

It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for teaching and research. MATLAB has many advantages compared to conventional computer languages for solving technical problems. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide. It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering.

## 4.3 The Plant

Assumed that the transfer function of the plant on Figure 4.1 is given as:

$$G(s) = \frac{100}{(S+1)(S+3)(S+5)(S+7)}$$

(4.2)

The open loop step response before PID controller of the plant is shown in Figure 4.2. By using MATLAB program which mentioned in APPENDIX A and with steady state value 0.9524 form the step response obtained the parameters (K, L, and T) as (K=0.9524,L=0.5177 ,T=2.0454), The specification of the open loop step is shown in Table 4.1

Table 4.1: The open loop step response of the plant without PID

| The specifications |
|---|

| Rise time | Settling time | Peak over shoot | Peak time |
|---|---|---|---|
| 2.48 sec | 4.69 sec | 0 % | >9 sec |

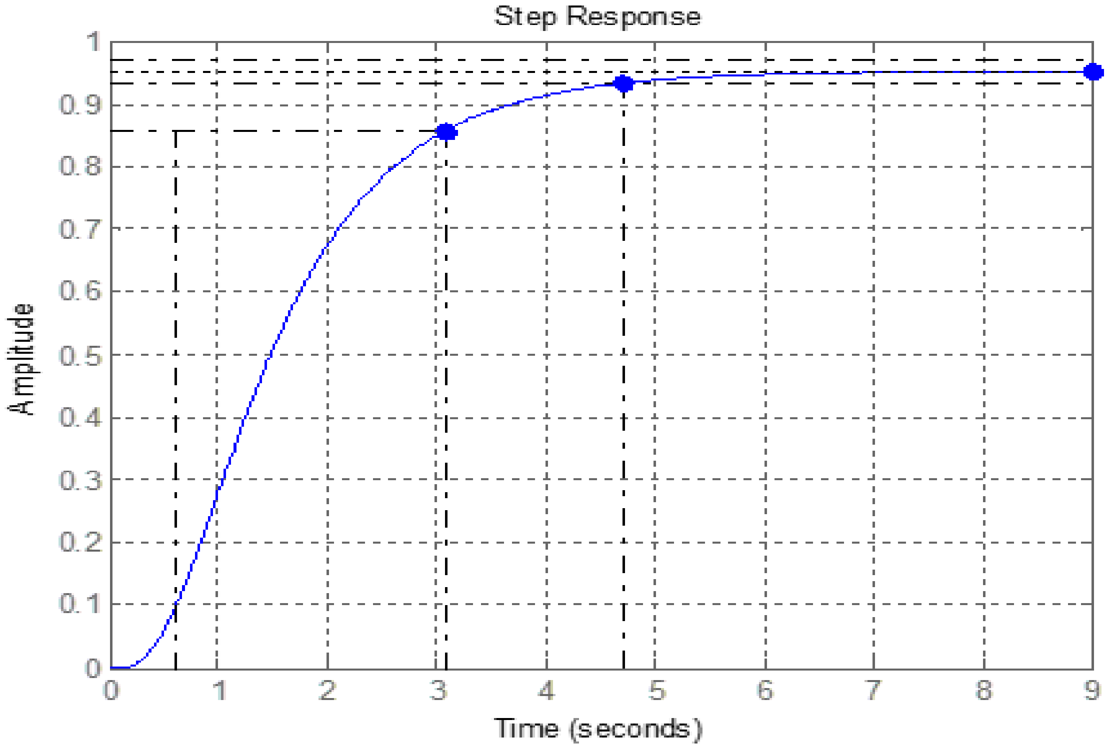

Figure 4.2: The open loop step response of the plant

## 4.4 The PID controller tuning methods

The closed loop step response specification can be obtained using the following methods**.**

### 4.4.1 Zeigler-Nichols Tuning Method

According to Zeigler- Nichols tuning method (first method) the parameters of the controller can be obtained:

Kp=1.2/a, Ti=2*L, Td=0.5*L (4.3)

Where a=$\frac{K*L}{T}$

Therefore a =(0.9524*0.5177)/2.0454= 0.2411 , Kp=(1.2/0.2411)=4.9781, Ti=(2*0.5177)=1.0354, and Td=(0.5*0.5177)=0.25885. By substituting these values the controller transfer function can be written as:

$$Gc(s) = \frac{1.3342\,s^2 + 5.15432\,s + 4.9781}{1.0354\,s} \qquad (4.4)$$

The close loop step response using Z-N (first method) and specification shown in Figure 4.3 and Table 4.2 respectively

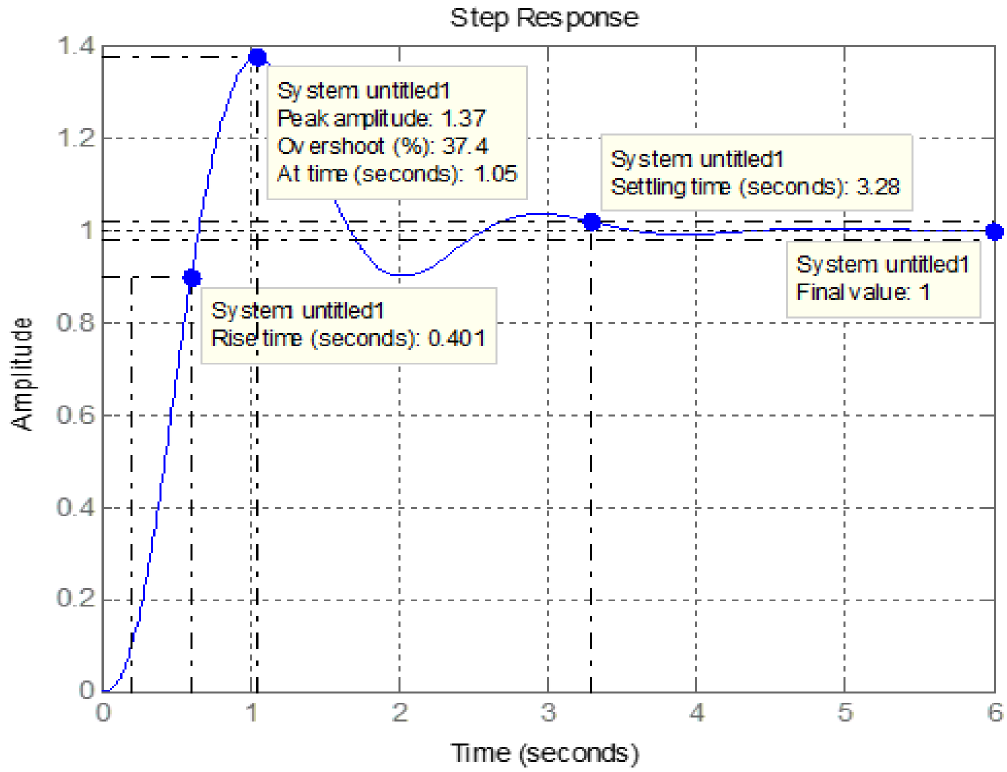


Figure 4.3: closed loop step response of the plant using Zeigler-Nichols method (first method)

Table 4.2: The close loop step response of the plant using Zeigler-Nichols method (first method)

| The specifications | | | |
|---|---|---|---|
| Rise time | Settling time | Peak over shoot | Peak time |
| 0.401 sec | 3.28 sec | 37.4% | 1.05 sec |

According to Zeigler-Nichols tuning method (second method). The critical gain (which gives the oscillations in Figure 4.4) is equal to $K_{cr}$ =7.2 and the critical period is equal to $P_{cr}$ =1.895.

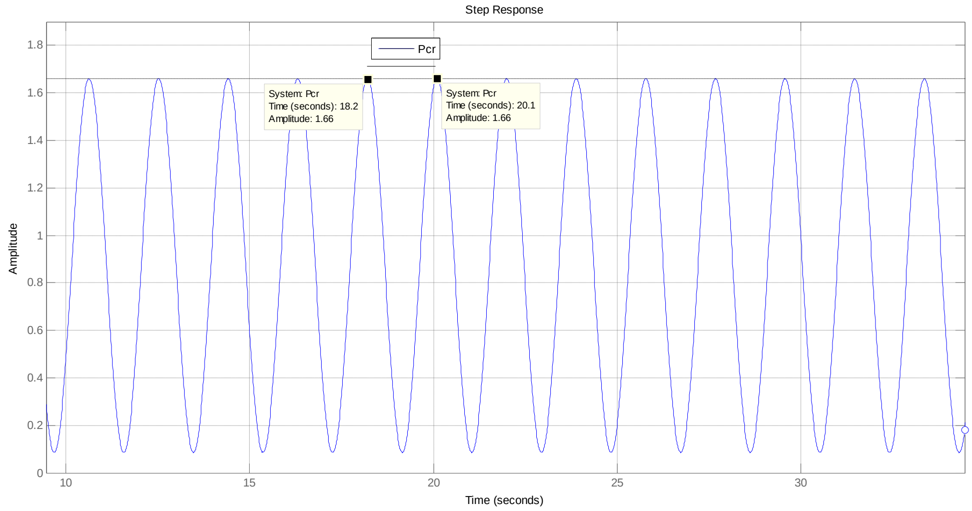


Figure 4.4: The plant sustained oscillations with period $P_{cr}$

The controller parameters can be obtained as:

$K_p = 0.6 * K_{cr}$ =0.6*7.2= 4.3200, $T_i = 0.5 * P_{cr}$ = 0.5*1.895=0.9475

$T_d = 0.125 * P_{cr}$ =.125*1.895=0.2369.

Therefor the controller transfer function is:

$$Gc(s) = \frac{0.9697s^2 + 4.093s + 4.3200}{0.9475s} \quad (4.5)$$

The close loop step response using Z-N (second method) and specification shown in Figure 4.5 and Table 4.3 respectively

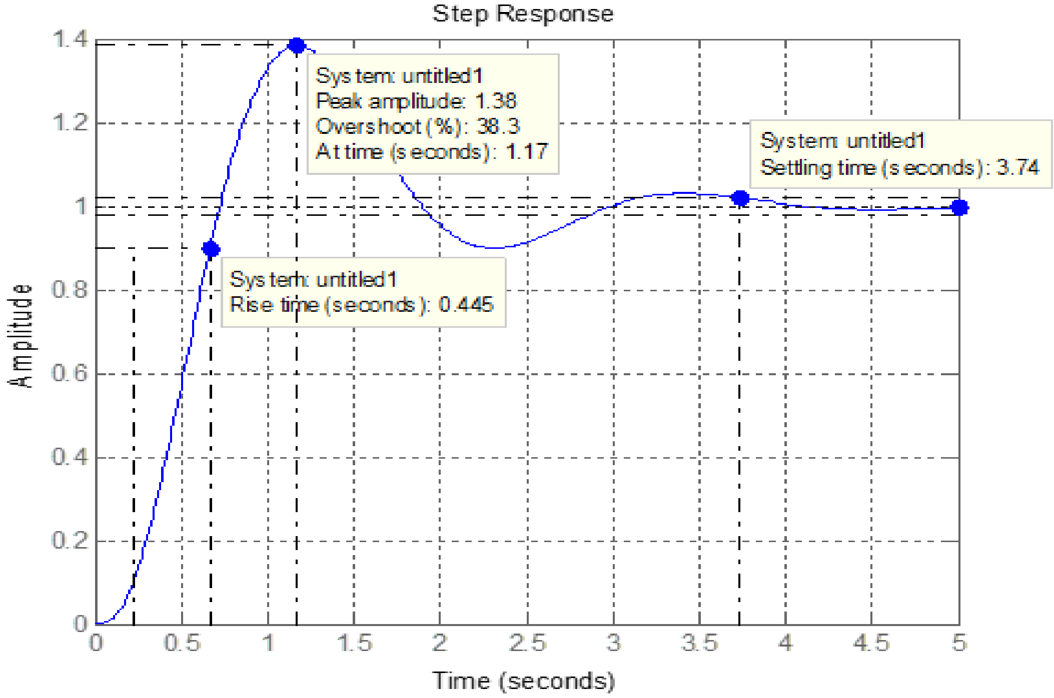


Figure 4.5: The close loop step response using Zeigler-Nichols method (second method)

Table 4.3: The closed loop step response using Zeigler-Nichols method (second method)

| The specifications | | | |
|---|---|---|---|
| Rise time | Settling time | Peak over shoot | Peak time |
| 0.445 sec | 3.74 sec | 38.3% | 1.17 sec |

### 4.4.2 Chien-Hornes-Reswick (CHR) Tuning Method

According (CHR) using method (first method (0% overshoot)) the controller parameters are obtained as the follows:

Kp=0.6/a=0.6/ (0.9524*0.5177/2.0454)=2.4890, Ti=T=2.0454

And Td=0.5*L= 0.5*0.5177=0.25885

There for the controller transfer function is:

$$Gc(s) = \frac{1.3178s^2+5.091s+2.4890}{2.0454s} \qquad (4.6)$$

And the close loop step response is shown in Figure 4.6

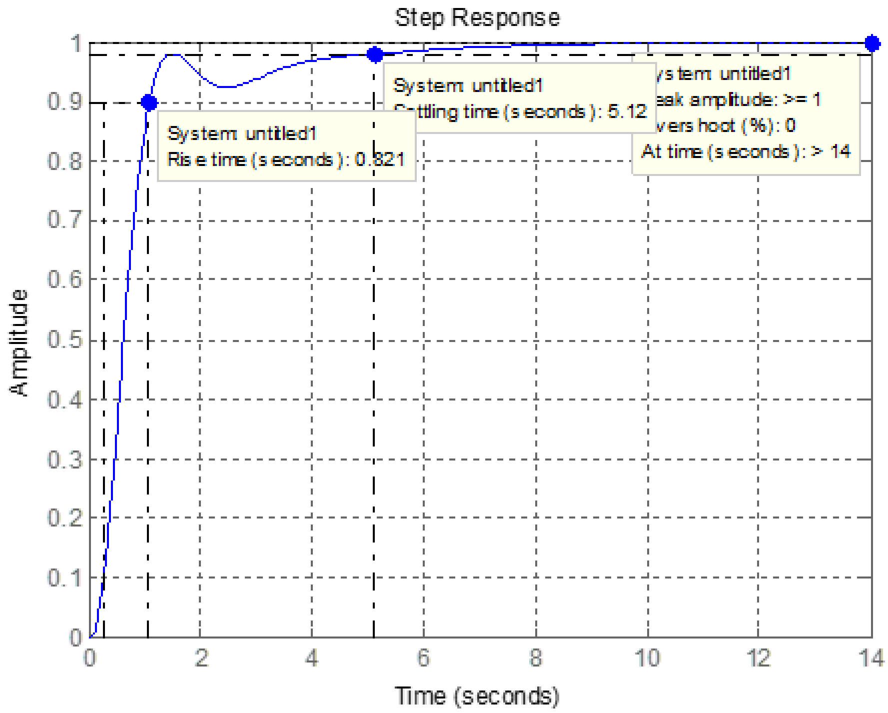


Figure 4.6: The close loop step response of the plant using CHR tuning method (first method)

The close loop step response specification shown in Table 4.4

Table 4.4: The closed loop step response with a controller of plant

| The specifications | | | |
|---|---|---|---|
| Rise time | Settling time | Peak over shoot | Peak time |
| 0.821 sec | 5.12 sec | 0 % | >14 sec |

The controller parameters obtained from (CHR) tuning method (second method (20% overshoot)), can be shown as the following:

Kp=0.95/a=0.95/(0.9524*0.5177/2.0454)=3.9411,

Ti=1.4*T=1.4*2.0454=2.8636

And Td=0.47*L=0.47*0.5177=0.2433, and the controller transfer function is:

$$Gc(s) = \frac{2.7458 s^2 + 11.2857 s + 3.9411}{2.8636 s} \quad (4.7)$$

The close loop step response using CHR (second method) and specification shown in Figure 4.7 and Table 4.5 respectively

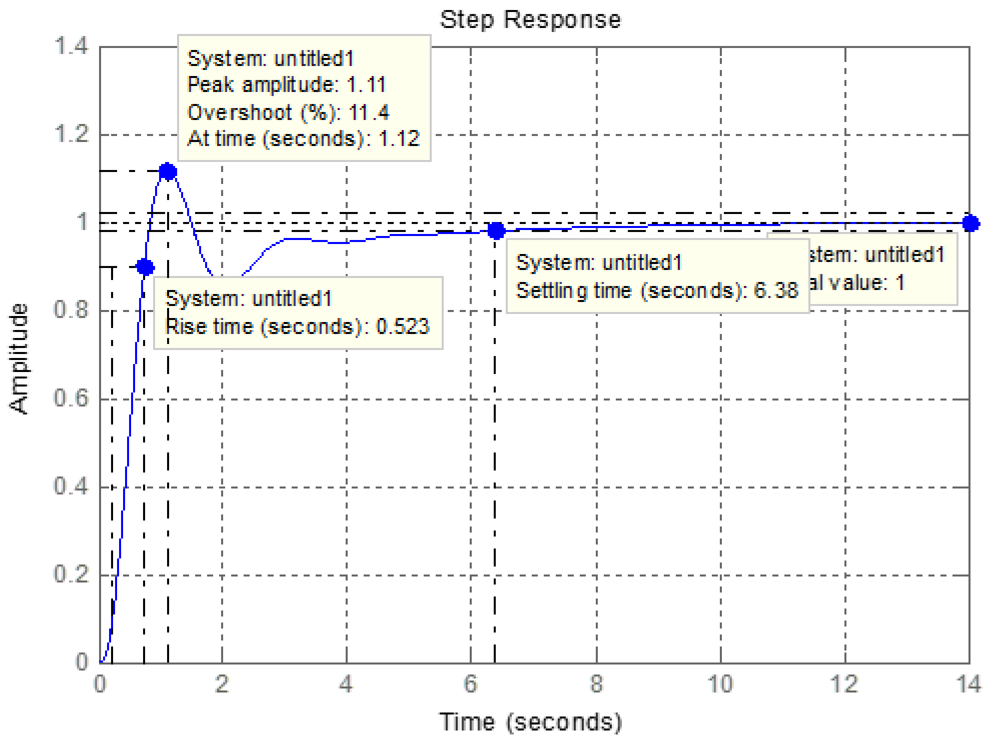


Figure 4.7: The close loop step response using CHR tuning method (second method)

Table 4.5: The close loop step response using CHR tuning method

| the specification | | | |
|---|---|---|---|
| Rise time | Settling time | Peak overshoot | Peak time |
| 0.523 sec | 6.38 sec | 11.4 % | 1.12 sec |

### 4.4.3 Cohen-Coon Tuning Method:

According to this method the controller parameters can be shown as the following: τ=L/ (L+T) =0.5177/(0.5177+2.0454)=0.20198 , a=k*L/T=(0.9524*0.5177)/2.0454=0.24105

Kp= $(\frac{1.25}{a})*(1+\frac{0.18*\tau}{1-\tau})$

=(1.25/0.24105)*(1+((0.18*0.20198)/(1-0.20198)))=5.422

Ti=$\frac{3.3-\tau}{1+1.2\tau}*L$=((3.3-0.20198)/(1+1.2*0.20198))*0.5177=1.2909

Td= $\frac{0.37-0.37\tau}{1-0.81\tau}*L$ = ((0.37-0.37*0.20198)/(1-0.81*0.20198))*0.5177=0.1828

There for the controller transfer function can be shown as below:

$$Gc(s) = \frac{1.2795s^2+6.9993s+5.422}{1.2909s} \quad (4.8)$$

And the close loop step response is shown in Figure 4.8

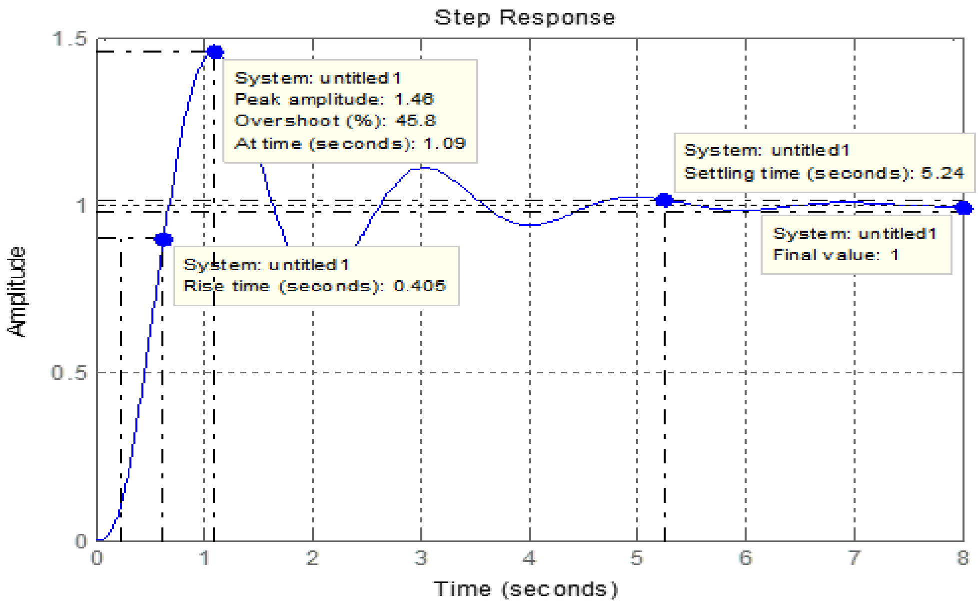


Figure 4.8: The close loop step response using cohn-coon tuning method

The close loop step response specification shown in Table 4.6

Table 4.6: The close loop step response using cohn-coon method

| The specification | | | |
|---|---|---|---|
| Rise time | Settling time | Peak over shoot | Peak time |
| 0.405 sec | 5.24 sec | 45.8 % | 1.09 sec |

### 4.4.4 Wang-Juang-Chan Tuning Method

In this method the controller parameters can be obtained from the following equations:

$$Kp=\frac{(0.7303+0.5307*\frac{T}{L}+0.5L)}{K(T+L)}=\frac{(0.7303+0.5307*\frac{2.0454}{0.5177}+0.5*0.5177)}{0.9524*(2.0454+0.5177)}=1.26415$$

$$T_i=T+0.5*L=2.0454+0.5*0.5177=2.3043$$

$$Td=\frac{0.5*L*T}{T+0.5*l}=\frac{0.5*0.5177*2.0454}{2.0454+0.5*0.5177}=0.2298$$

And the controller transfer function according to these values is:

$$Gc(s)=\frac{0.6694s^2+2.91298s+1.26415}{2.3043s} \quad (4.9)$$

The close loop step response as shown in Figure 4.9:

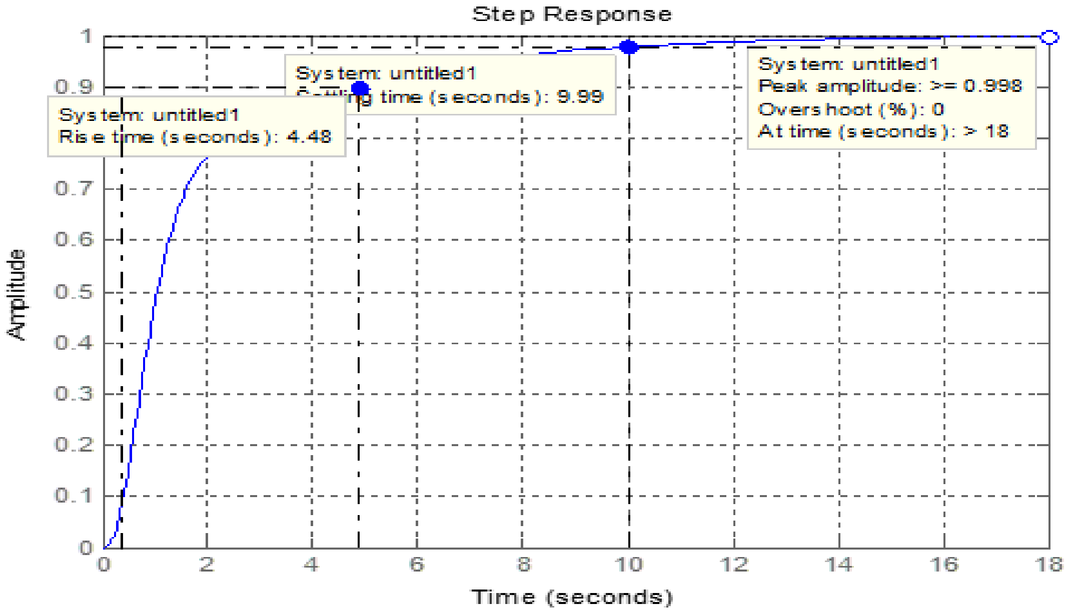


Figure 4.9: The close loop step response using wang –j-c tuning method

The close loop step response specification shown in Table 4.7

Table 4.7: The close loop step response using wang-juang-chan method

| The specification | | | |
|---|---|---|---|
| Rise time | Settling time | Peak over shoot | Peak time |
| 4.48 sec | 9.99 sec | 0 % | >18 sec |

### 4.4.5 Optimum PID Tuning Method

In this method the first step determines the range of (L/T), and then According to this range chooses your tuning equation. The rang of (L/T) from the given transfer function is equal to (0.5177/2.0454)=0.2531 There for the parameters of the controller are obtained as follow:

$$K_p = \frac{a_1}{k}\left(\frac{L}{T}\right)^{b_1}, \quad T_i = \frac{T}{a_2 + b_2(L/T)} \qquad T_d = a_3 T\left(\frac{L}{T}\right)^{b_3} \tag{4.10}$$

The value of the (a, b) depend on which the optimum criteria used (ISE, ISTE, and $IST^2E$)

In (ISE) criteria:

Kp = $\frac{1.048}{0.9524}\left[\frac{0.5177}{2.0454}\right]^{-0.897}$=3.7738, Ti=$\frac{2.0454}{1.195-0.368\left(\frac{.5177}{2.0454}\right)}$=1.8563 and

Td=0.489*2.0454$\left[\frac{0.5177}{2.0454}\right]^{0.888}$=0.2953

The controller transfer function According to these values is:

$$Gc(s)=\frac{2.06866s^2+7.0053s+3.7738}{1.8563s} \tag{4.11}$$

The close loop step response using optimal PID in (ISE) and specification shown in Table 4.8 and Figure 4.10 respectively

Table 4.8: Closed loop step response using ISE optimum tuning method

| The specification |
|---|

| Rise time | Settling time | Peak over shoot | Peak time |
|---|---|---|---|
| 0.509 sec | 2.73 sec | 11.5 % | 1.1 sec |

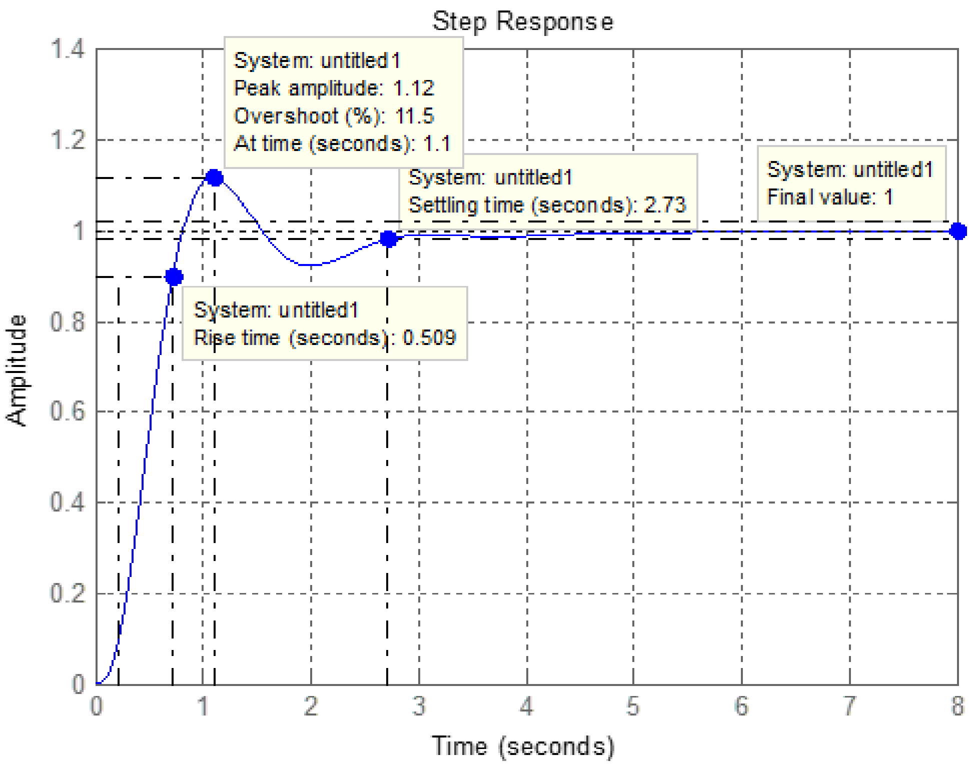


Figure 4.10: The close loop step response using ISE optimum tuning method

In (ISTE) criteria:

$$Kp = \frac{1.042}{0.9524}\left[\frac{0.5177}{2.0454}\right]^{-0.897} = 3.7522,$$

$$Ti = \frac{2.0454}{0.987 - 0.238\left(\frac{.5177}{2.0454}\right)} = 2.2070, \text{ and}$$

$$Td = 0.385 * 2.0454\left[\frac{0.5177}{2.0454}\right]^{0.906} = 0.2268$$

The controller transfer function According to these values is:

$$Gc(s) = \frac{1.878s^2 + 8.2811s + 3.7522}{2.207s} \quad (4.12)$$

The close loop step response using optimal PID in (ISE) and specification shown in Figure 4.11 and Table 4.9 respectively.

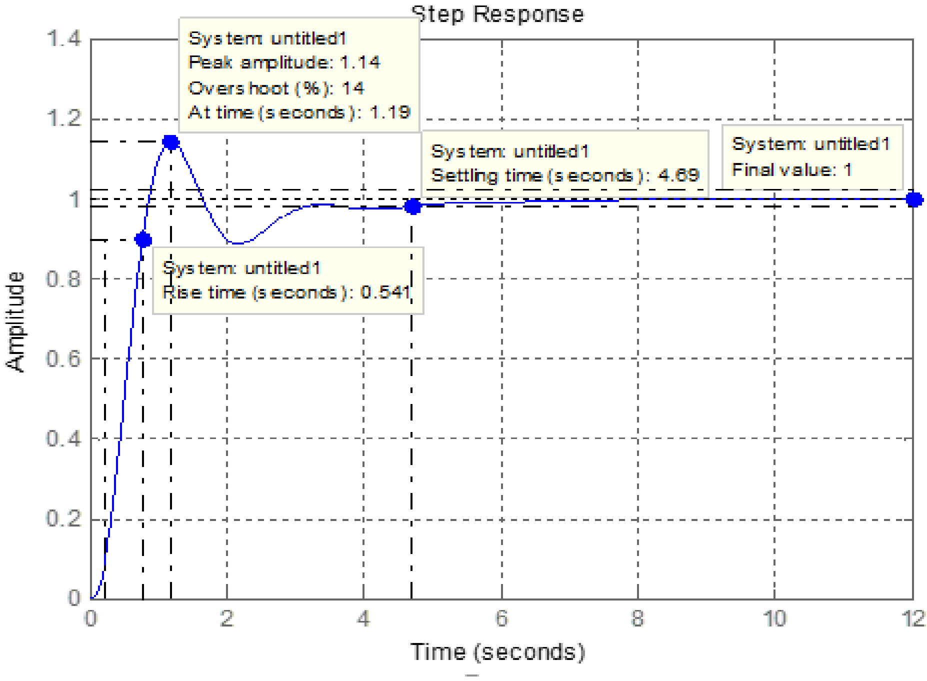


Figure 4.11: The close loop step response using ISTE optimum Tuning method

Table 4.9: the closed loop step response using ISTE optimum Tuning method

| The specification | | | |
|---|---|---|---|
| Rise time | Settling time | Peak over shoot | Peak time |
| 0.541 sec | 4.69 sec | 14% | 1.19 sec |

In (IS$T^2$E) criteria:

Kp = $\frac{0.968}{0.9524}\left[\frac{0.5177}{2.0454}\right]^{-0.904}$=3.5194,

Ti=$\frac{2.0454}{0.977-0.253\left(\frac{0.5177}{2.0454}\right)}$=2.2404, and Td=0.316*2.0454$\left[\frac{0.5177}{2.0454}\right]^{0.892}$=0.18976.

The controller transfer function According to these values is:

$$Gc(s) = \frac{1.4962s^2+7.8849s+3.5194}{2.2404s} \quad (4.13)$$

The close loop step response using optimal PID in (IS$T^2$E) and specification shown in Figure 4.12 and Table 4.10 respectively.

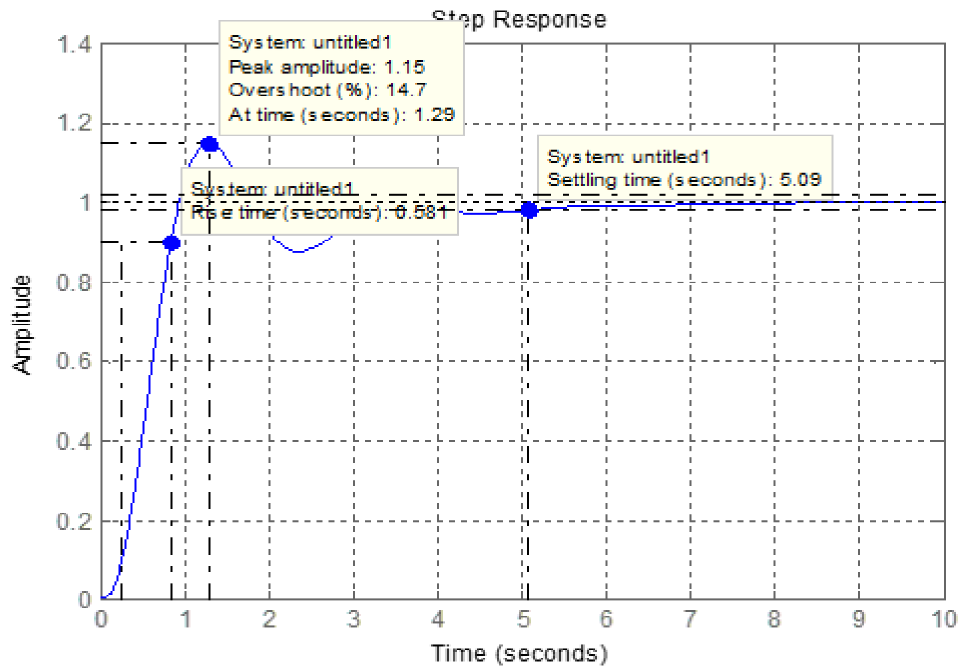


Figure 4.12: Close loop step response using (IS$T^2$E) optimum Tuning method

Table 4.10: The close loop step response using (IS$T^2$E) optimum Tuning method

| The specification | | | |
|---|---|---|---|
| Rise time | Settling time | Peak over shoot | Peak time |
| 0.581 sec | 5.09 sec | 14.7 % | 1.29 sec |

Table 4.11: Comparison between the difference PID tuning methods

| Specifications / The tuning Methods | The rise time tr (sec) | The settling time ts (sec) | The peak overshoot Mp% | The peak time tp (sec) |
|---|---|---|---|---|
| Z-N first method | 0.401 | 3.28 | 37.4 | 1.05 |
| Z-N second method | 0.445 | 3.74 | 38.3 | 1.17 |
| CHR first method | 0.821 | 5.12 | 0.00 | >14 |
| CHR second method | 0.523 | 6.38 | 11.4 | 1.12 |
| Cohen coon method | 0.405 | 5.24 | 45.8 | 1.09 |

| | | | | |
|---|---|---|---|---|
| ISE optimal method | 0.509 | 2.73 | 11.5 | 1.10 |
| ISTE optimal method | 0.541 | 4.69 | 14.0 | 1.19 |
| IS$T^2E$ optimal method | 0.581 | 5.09 | 14.7 | 1.29 |
| Wang –Juang -Chan method | 4.480 | 9.99 | 0.00 | >18 |

The close loop response from all methods is shown in Figure 4.13

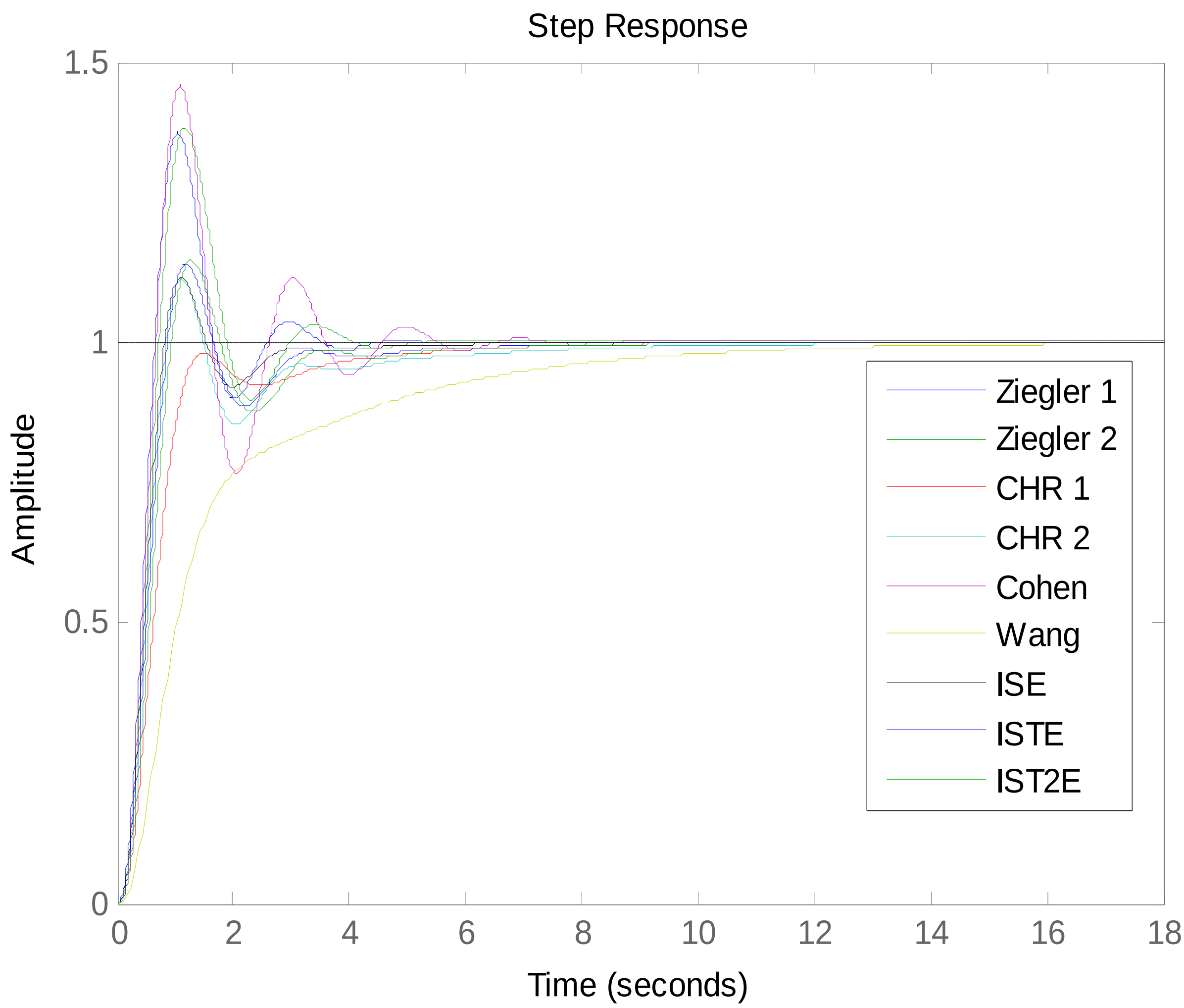


Figure 4.13: Close loop step from all methods

## 4.5 Discussion

Some methods gives critically damped response for the system under consideration .Such method will be ignored for controlling the system , finally observe that from the Table 4.11 the best rise time (tr ) for the given transfer function is obtained when Ziegler- Nichols tuning method (first method ) has applied. While the ISE optimal tuning method give the best settling time(ts) ,and the best overshoot can be obtained when the (CHR)tuning method (first method) applied.

# CHAPTER FIVE

# CONCLUSION AND RECOMMENDATIONS

## 5.1 Conclusion

PID control is still of great interest, and is a promising control strategy that deserves further research and investigation both industry and academia. The tuning methods are only valid for self-regulating processes (i.e. open loop stable processes such as those that may be described by the first order plus dead-time description). The selection of the controller tuning method is the most important thing that obtains the desired specifications. These specifications depend largely upon the control system, the necessary physical modifications and the performance specifications to be achieved. The best method of obtain rise time is Ziegler- Nichols tuning method (first method) and of obtain settling time is the ISE optimal tuning method.

## 5.2 Recommendations

In accordance of the PID developments there are modern technological methods used for adjustment such as:

- Fuzzy PID controller tuning methods.
- Digital PID controller adjustment.
- Genetic algorithms for PID tuning.
- Neuro-fuzzy PID controller tuning methods.

# References

[1] Dingyu xue, yang Quan Chen, and Derek p Atherton, "Linear Feedback Control Analysis and Design with MATLAB". The society for Industrial and Applied Mathematics 2008.

[2] Katsuhiko Ogata," Modern control engineering fifth Edition". Prentice – Hall, New Jersey, 2010.

[3] Karl J.Astrom and Tore Hagglund, "PID controller: theory, design and tuning," .Instrument society of America 1995.

[4] Pao C. Chau, "Chemical process control: A first course with MATLAB". University of California, San Diego 2001.

[5] Cheng-Ching Yu, "Auto Tuning of PID Controllers 2nd Edition"

Spering-Verlage London Limited 2006

[6] Sung. S.WLee,"Modified Proportional – Integral Derivative (PID) Controller and a new tuning method for the PID controller". Ind.Eng.Chem.Res.1995.

[7] Phillips, C. L., and R. D. Harbor, "Feedback Control Systems". Upper Saddle River, NJ: PrenticeHall, 1988.

# APPENDICES

## APPENDIX A

## Delay Time and Time Constant

```
function [K,L,T]=getfod(g,method)
K=dcgain(g);
if nargin==1
[kc,pm,wc,wcp]=margin(g);
ikey=0;
   L=1.6*pi/(3*wc);
   T=0.5*kc*K*L;
   if isfinite(kc)
      x0=[L;T];
      while ikey==0 ,u=wc*x0(1);v=wc*x0(2);
         ff=[K*kc*(cos(u)-v*sin(u))+1+v^2;sin(u)+v*cos(u)];
         j=[-K*kc*wc*sin(u)-K*kc*wc*v*cos(u),-
K*kc*wc*sin(u)+2*wc*v;wc*cos(u)-wc*v*sin(u),wc*cos(u)];
         x1=x0-inv(j)*ff;
         if norm(x1-x0)<1e-8
            ikey=1;
         else
            x0=x1;
         end
      end
      L=x0(1);T=x0(2);
   end
elseif nargin==2 & method==1
   [n1,d1]=tfderv(g.num{1},g.den{1});
   [n2,d2]=tfderv(n1,d1);
   k1=dcgain(n1,d1);k2=dcgain(n2,d2);
```

```
    Tar=-k1/K;T=sqrt(k2/K-Tar^2); L=Tar-T;
end
function [e,f]=tfderv(b,a)
f=conv(a,a); na=length(a);nb=length(b);
e1=conv((nb-1:-1:1).*b(1:end-1),a);
e2=conv((na-1:-1:1).*a(1:end-1),b);
maxL=max(length(e1),length(e2));
e=[zeros(1,maxL-length(e1)) e1]-[zeros(1,maxL-length(e2)) e2];
```

# APPENDIX B
# Zeigler-Nichols Tuning Method Code

```
function [Kp,Ti,Td,H]=ziegler(key,vars)
 Ti=[]; Td=[]; H=1;
 if length(vars)==3,
 K=vars(1); L=vars(2); T=vars(3); a=K*L/T;
 if key==1, Kp=1/a;
 elseif key==2, Kp=0.9/a; Ti=3.33*L;
 elseif key==3 | key==4, Kp=1.2/a; Ti=2*L; Td=L/2; end
 elseif length(vars)==2,
 K=vars(1); Tc=vars(2);
 if key==1, Kp=0.5*K;
 elseif key==2, Kp=0.4*K; Ti=0.8*Tc;
 elseif key==3 | key==4, Kp=0.6*K; Ti=0.5*Tc; Td=0.125*Tc; end
 elseif length(vars)==4,
 K=vars(1); Tc=vars(2); rb=vars(3);
 pb=pi*vars(4)/180; Kp=K*rb*cos(pb);
 if key==2, Ti=-Tc/(2*pi*tan(pb));
 elseif key==3|key==4, Ti=Tc*(1+sin(pb))/(pi*cos(pb)); Td=Ti/4; end
 end
  [Gc,H]=writepid(Kp,Ti,Td,key)
```

# APPENDIX C
# CHR Tuning Method Code

```
function [Gc,Kp,Ti,Td,H]=chrpid(key,tt,vars)
K=vars(1); L=vars(2); T=vars(3); a=K*L/T; Ti=[]; Td=[];
ovshoot=vars(4); if tt==1, TT=T; else TT=L; tt=2; end
if ovshoot==0,
KK=[0.3,0.35,1.2,0.6,1,0.5; 0.3,0.6,4,0.95,2.4,0.42];
else,
KK=[0.7,0.6,1,0.95,1.4,0.47; 0.7,0.7,2.3,1.2,2,0.42];
end
switch key
case 1, Kp=KK(tt,1)/a;
case 2, Kp=KK(tt,2)/a; Ti=KK(tt,3)*TT;
case {3,4}, Kp=KK(tt,4)/a; Ti=KK(tt,5)*TT; Td=KK(tt,6)*L;
end
[Gc,H]=writepid(Kp,Ti,Td,key);
```

# APPENDIX D
# Cohen-Coon Tuning Method

```
function [Gc,Kp,Ti,Td,H]=cohenpid(key,vars)
K=vars(1); L=vars(2); T=vars(3);
a=K*L/T; tau=L/(L+T); Ti=[]; Td=[];
switch key
case 1,Kp=(1+0.35*tau/(1-tau))/a;
case 2,
Kp=0.9*(1+0.92*tau/(1-tau))/a; Ti=(3.3-3*tau)*L/(1+1.2*tau);
case {3,4}, Kp=1.25*(1+0.18*tau/(1-tau))/a;
Ti=(3.3-tau)*L/(1+1.2*tau); Td=0.37*(1-tau)*L/(1-0.81*tau);
 case 5
Kp=1.24*(1+0.13*tau/(1-tau))/a; Td=(0.27-0.36*tau)*L/(1-0.87*tau);
 end
[Gc,H]=writepid(Kp,Ti,Td,key)
```

# APPENDIX E

```
function [Gc,H]=writepid(Kp,Ti,Td,key)
switch key
case 1, Gc=Kp;
case 2, Gc=tf(Kp*[Ti,1],[Ti,0]); H=1;
case 3, nn=[Kp*Td*Ti,Kp*Ti,Kp];
dd=[Ti,0]; Gc=tf(nn,dd); H=1;
end
```