IMPLEMENTATION OF PROPORTIONAL, INTEGRAL AND DERIVATIVE(PID) TERMS FOR THERMAL MANAGEMENT THROUGH SIMULINK AND ARDUINO PLATFORM

By:

MITCHELLE LAW JYY JINN

(Matrix no.: 128944)

Supervisor:

Dr. Yu Kok Hwa

May 2019

This dissertation is submitted to

Universiti Sains Malaysia

As partial fulfilment of the requirement to graduate with honors degree in

BACHELOR OF ENGINEERING (MECHANICAL ENGINEERING)

School of Mechanical Engineering

Engineering Campus

Universiti Sains Malaysia

DECLARATION

I hereby declare that I have conducted, completed the research work and written the dissertation entitled “Implementation of Proportional, Integral and Derivative (PID) Terms for Thermal Management through Simulink and Arduino Platform”. I also declare that it has not been previously submitted for the award of any degree or diploma or other similar title of this for any other examining body or University.

Name of student : Mitchelle Law Jyy Jinn Signature:

Date :

Witnessed by

Supervisor : Dr. Yu Kok Hwa Signature:

Date :

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest appreciation to my supervisor, Dr. Yu Kok Hwa for his continuous supervision, assistance and support from the initial to the final stage of my final year project. This dissertation would not be possible without his patient guidance and enthusiastic encouragement throughout the whole period. The valuable lessons learnt not only helped me in enhancing my knowledge, but the interpersonal skill and the positive attitude as well.

Besides, I would also like to thank my friends for sharing their opinions and knowledge about the control system as well as MATLAB Simulink with me. Thanks for their encouragements too so that I can continuously be motivated during the development of this thesis. Next, I would like to express my sincere gratitude to my family who constantly support and encourage me spiritually and mentally.

In addition, I also need to thank all the academic staffs at School of Mechanical Engineering, University Science Malaysia (USM) for their assistances and services to help me in the completion of report. Not to forget the involved library staffs who are willing to carry out a few workshops about writing and citation of the thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

AC	Alternating Current
CV	Control Variable
D	Derivative controller
e	Error
I	Integral controller
P	Proportional controller
PI	Proportional and Integral controller
PID	Proportional, Integral and Derivative controller
PV	Process variable
PWM	Pulse Width Modulation
s	Complex Laplace variable
SP	Set point
SSR	Solid State Relay
ω_n	Natural frequency
ζ	Damping ratio
G(s)	Open-loop transfer function
R(s)	System input
C(s)	System output
y_{final}	Steady state output of response graph
y(t)	Process output
u(t)	Control signal
Ku	Ultimate gain
Tu	Ultimate period
τ	Time constant
T_{ref}	Reference temperature

ABSTRAK

Tujuan tesis ini dijalankan adalah untuk mengimplementasikan sistem kontrol PID dalam pengurusan terma dan menganalisiskan prestasi bagi sistem PID tersebut. Ini dilakukan melalui kawalan suhu mentol lampu berpunca arus bolak-balik (AC) dengan bantuan MATLAB Simulink, Arduino, Relay Solid-State (SSR) dan sensor suhu TMP36. Parameter Proporsional, Integral dan Diferensial (K_p, K_i dan K_d) masing-masing diperiksa dalam pelbagai nilai untuk mencapai titik set 60°C. Dari penyiasatan, nilai-nilai yang sesuai bagi K_p, K_i and K_d dipilih berdasarkan kecenderungan mereka untuk memberikan ralat keadaan mantap dan lajakan yang boleh diabaikan, masa kenaikan dan masa penyelesaian yang lebih kurang, di samping kestabilan yang bagus. Melalui pemerhatian, nilai proporsional yang lebih tinggi dapat mengurangkan masa kenaikan, masa penyelesaian dan ralat keadaan mantap pada respon graph. Manakala, kontroler integrator dapat menghapuskan ralat keadaan mantap. Nilai Integral yang tinggi akan menyebabkan lajakan selain memanjangkan masa penyelesaian. Seterusnya, kontroler diferensial ditambah untuk menambahbaikan kestabilan sistem. Nilai diferensial yang lebih tinggi menunjukkan kestabilan yang lebih baik. Untuk eksperimen ini, K_p = 10, K_i = 0.02 dan K_d = 5 dianggap sesuai dan nilai-nilai ini digunakan untuk mensimulasikan pengurusan haba bagi persediaan ini pada titik set suhu yang berbeza iaitu, 50 ° C, 70 ° C dan 80 ° C pada lampu mentol AC. Semua titik set boleh dicapai dengan tiadanya ralat keadaan mantap, masa kenaikan dan penyelesaian yang bagus, tiada lajakan, dan kestabilan yang bagus. Cara penalaan percubaan dan kesilapan PID yang digunakan ini adalah kurang rumit dan analitikal berbanding dengan cara penalaan yang sedia ada.

ABSTRACT

The purpose of the thesis is to employ PID control system for thermal management and to evaluate the performance of the PID control system. This is carried out through the control of temperature of the Alternating Current (AC) lightbulb with the aid of MATLAB Simulink, Arduino, Solid-State Relay (SSR) and TMP36 temperature sensor. The proportional, integral and derivative parameters are examined in a range of values to achieve temperature set point of 60°C. From the investigations, the suitable values for K_p, K_i and K_d are, respectively, chosen based on their tendency to give negligible steady state error, less rise time, less settling time, good stability and no overshoot or undershoot. Through the observation, higher proportional gain contributes to less rise time, settling time and steady state error. Meanwhile, integral controller can eliminate the steady state error. High integral value will cause overshoot and undershoot as well as increases the settling time. Next, derivative controller is added to improve stability of the system. Higher derivative value shows better stability. For this experiment, K_p=10, K_i=0.02 and K_d=5 is deemed suitable and these values are employed to simulate thermal management for this setup for different temperature set points at 50°C, 70°C and 80°C on the AC lightbulb. All the set points can be achieved with no steady state error, good rise time, good settling time, no overshoot, no undershoot and good stability. The, trial and error PID tuning method is considered as less complicated and less analytical way compared to the existing tuning methods.

CHAPTER 1: INTRODUCTION

1.1 Research Background

Control is needed in many industries such as control the speed of motors, temperature, pressure and the liquid level to obtain a desired and an accurate output. There are two types of control systems which are the open-loop control and closed-loop control. In an open-loop control system as shown in Figure 1.1, the system output is directly controlled without utilising any information about the physical output. In other words, the actuator signal to the system is unaffected by the actual output of the system. Oppositely, closed-loop control system as shown in Figure 1.2 has the feedback system that measures the actual output and then compare it with the desired output. With such feedback, the actuator input signal can therefore be changed to achieve the desired output [1].

Figure 1.1: Open-loop Control System [2]

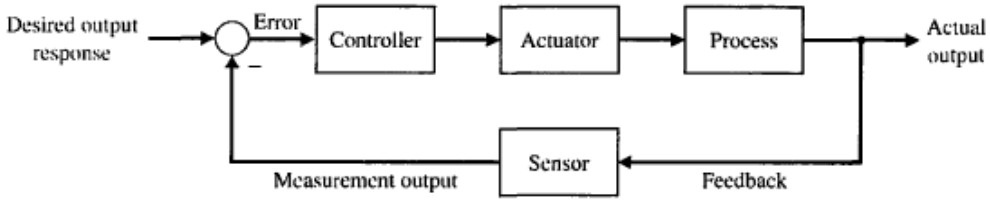

Figure 1.2: Closed-loop Control System [2]

According to Ernest O Doebelin [3], among many control systems, on-off switching is the simplest form of control and thus it is the most commonly used one. In a plant, when the output is below the set point, positive error is obtained which will fully turn on the plant and if the error is negative, the plant is being turned off fully as illustrated in Figure 1.3 [4].

Figure 1.3: Amplitude vs Time Graph for On-off Controller [4]

In the textbook entitled “Feedback Control for Computer System”, Philipp K. Janert [4] states that the principle issue of on-off controller is that the system never settles down to a steady state but instead it oscillates constantly and rapidly between its two extreme states. Quickly turning on and off around a set point will harm the actuator. Consequently, PID (Proportional, Integral and Derivative) controller which apply the closed-loop feedback control system and pulse width modulation (PWM) is more preferred. Besides, on-off controllers are generally less utilized than PID controllers, because they are not as powerful or as flexible [5].

The primary advancement of the PID controller was created in 1911 by Elmer Sperry. In 1933, Taylor Instrumental Company (TIC) presented the first pneumatic controller with a fully tuneable proportional controller. A few years later, proportional-integral controller was created to eliminate the steady state error found in proportional controllers. In 1940, TIC built up the first PID pneumatic controller with a derivative action, which reduced overshooting issues. J.G. Ziegler and N.B. Nichols published two papers in 1942 and 1943 that indicated how ideal controller parameters could be picked based first on open-loop tests on the plant and second on closed-loop tests on the plant [6]. The Ziegler and Nichols tuning methods have kept on being utilized, although later G.H. Cohen and G.A. Coon also invented an alternative tuning method in 1950 [6].

Fundamentally, for temperature control, a PID controller uses the sensor such as thermocouple, and changes the measurement to understandable engineering term that is Degrees Celsius (°C). Error is then determined by subtracting the actual output from a desired SP. Next, the error is acted upon by the P, I and D terms simultaneously. It is visible that every procedure control application would profit from the PID control. For example, it is used in curing rubber. Exact temperature control guarantees complete cure is accomplished without unfavourably influencing material properties. PID also

applied in baking such that commercial ovens must follow firmly endorsed heating and cooling sequences to ensure the necessary reactions take place.

PID control has achieve great improvement from conventional tuning methods to optimization techniques. The examples of conventional methods are Ziegler-Nichols method and Cohen-Coon method. Meanwhile, optimization tuning methods consist of Artificial Neural Network, Genetic Algorithms and Fuzzy logic PID which are explained further in literature review at Chapter Two. In this project, an Arduino platform and MATLAB Simulink are used to simulate the trial and error PID tuning method.

1.2 Problem Statement

There are existing methods for PID tuning that are done through a series of formulas such as Ziegler-Nichols method and Cohen-Coon method which are considered as very analytical ways and they are not applicable to all plants. Moreover, there are also intelligence ways for PID tuning which needs good supporting computer hardware and one needs to be expert in this specific topic. It is undeniable that such methods generate the optimize, precise and accurate P, I and D value for a system. Nevertheless, it is good to find more alternative ways so that one has more choices in selecting the PID tuning methods for certain system based on the knowledge, software, hardware and cost available as well as based on the desired degree of accuracy and precision of the output. In this project, a less analytical trial and error method for PID tuning is applied. MATLAB Simulink with Arduino I/O package are utilized so that real time response of the PID controller and the output temperature can be observed.

1.3 Objective of the Research

The objectives of this project are: -

- to employ PID control system for thermal management through MATLAB Simulink and Arduino I/O package.
- to evaluate the performance of PID control system for different working conditions.

1.4 Scope of Research

PID is a very common instrument used in industrial control applications such that it can be used for regulation of speed, flow, pressure and other process variables. In this project, the area of application of PID controller only focus on thermal management or temperature control of AC system as good temperature control can avoid major safety, quality, and productivity problems.

The main limitation of the tuning method in this project is that auto tuning cannot be done such that the suitable P, I and D values cannot be determined by the system itself. Manual tuning is needed by choosing the values that fit the system. Moreover, although the method used can achieve the desired outcome, the chosen values may not in the optimize stage.

1.5 Organization of Thesis

In this thesis, the writing is separated into five main chapters. These include introduction (chapter 1), literature review (chapter 2), methodology (chapter 3), results and discussions (chapter 4) as well as the conclusion (chapter 5).

Chapter 1 begins with a general background relating to on-off controller, the history and the evolvement of the use of PID controller.

Chapter 2 reviews the concepts related to the PID controller and the introduction of existing conventional and intelligent methods for PID tuning.

Chapter 3 discusses the adopted methods and procedures in the project. A methodology flowchart is presented to illustrate the project flow. Moreover, the software and hardware setup for PID tuning are discussed in detail.

Chapter 4 reflects the results of the PID tuning. Suitable parameters for P, I and D are chosen and the characteristics as well as effects of applying each of the controllers are discussed.

In chapter 5, the outcome of the research is concluded by answering the research objectives. In addition, future works are recommended to provide alternative to improve the system in the future.

CHAPTER 2: LITERATURE REVIEW

2.1 Process Control Basic Terminology

Essentials terms in the process control are the object of control, process variable (PV), control variable (CV), set point (SP), error (e), disturbances and feedback. Object of control can be characterized as dynamic process which noteworthy parameter, such as temperature, is subjected to control. In the meantime, the PV implies the current measured value of a process which is being controlled. The desired outcome is called the set point (SP). CV is the variable that can be changed to realize a desired outcome. The distinction between the PV and SP is the error (e). Disturbances are variables that cause undesired effects to the value of PV [7]. In a feedback system as illustrated in Figure 2.1, there is a process being controlled where a sensor will send signal to the controller and then some adjustments is made at the controller to send desired signals.

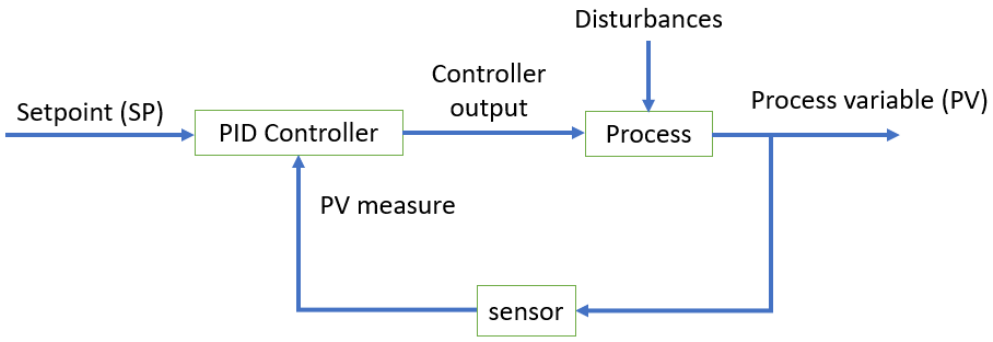

Figure 2.1: Feedback System [8]

2.2 Background of Pulse Width Modulation (PWM)

Pulse width modulation (PWM) is an essential way for controlling analogue circuits with a microprocessor's digital outputs. PWM can be utilized in numerous applications, such as measurement, communications and power control. As stated by Michael Barr [9], the system expenses and power utilization can be reduced by controlling analogue circuits digitally.

PWM signal comprises of continuous pulses of a certain period which implies the time taken by a point to travel the distance equal to one wavelength. These pulses

can only have binary values, that is high or low. Pulse width and duty cycle are the quantities that are closely related to the PWM. Pulse width is the time for which the output of the PWM is high. The duty cycle is the percentage of the pulse width to the period taken such that:

$$\text{Duty Cycle} = \frac{t_{on}}{T} \times 100\% \tag{2.1}$$

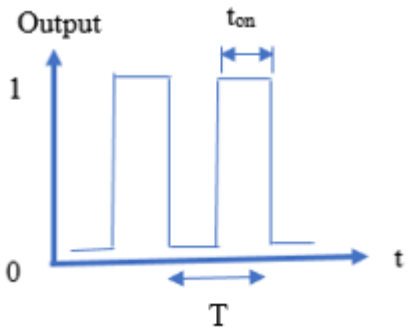

Figure 2.2: PWM

In the project, the duty cycle controls the temperature of the lightbulb such that Solid State Relay (SSR) receives PWM signal through PWM pin of Arduino. The function of PWM is to convert the control signal from the PID into a switching signal. Thus, by varying the pulse width through PID signal, the duty cycle of the PWM can be varied. By varying the duty cycle of the lightbulb, the temperature of the lightbulb can be varied.

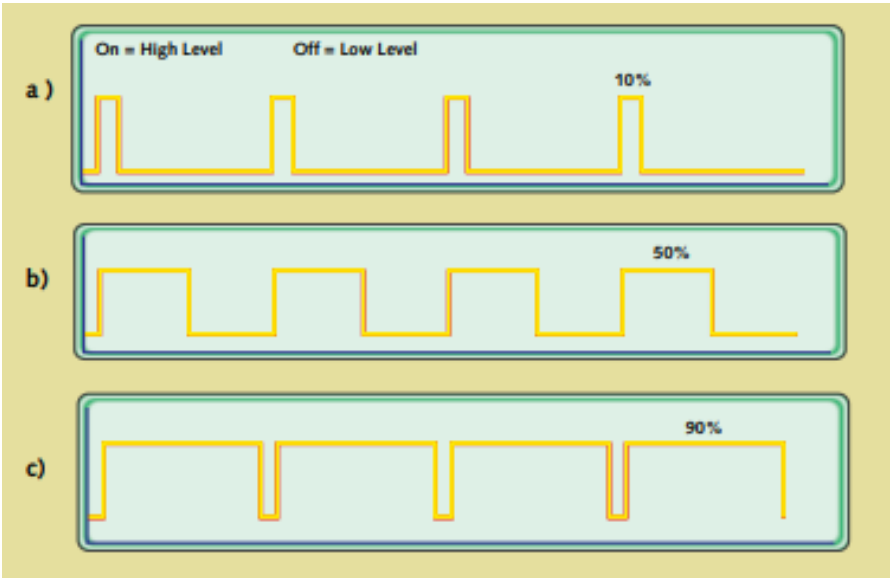

Figure 2.3: Graphs of Different PWM Signals [9]

Figure 2.3 shows three different PWM signals. Figure 2.3a demonstrates a PWM yield at a 10% duty cycle. That is, the signal is on for 10% of the period and off the other 90%. Figures 2.3b and 2.3c show PWM yields at half and 90% duty cycles, individually. These three PWM outputs encode three different analogue signal values, at 10%, 50%, and 90% of the full strength. If, for instance, the supply is 5V and the duty cycle is 10%, a 0.5V analogue signal results.

2.3 Background of Step Response Graph and its Characteristics

Generally, there are two sorts of step response, the first-order and the second-order. The order of the differential equation is the highest degree of derivative present in an equation. The system whose input-output equation is a first order differential equation is called first-order system. Most of the practical models are first-order systems. Allude to the block diagram of the closed-loop control system as shown in Figure 2.4, an open-loop transfer function, G(s)= $\frac{1}{sT}$ is associated with a unity negative feedback.

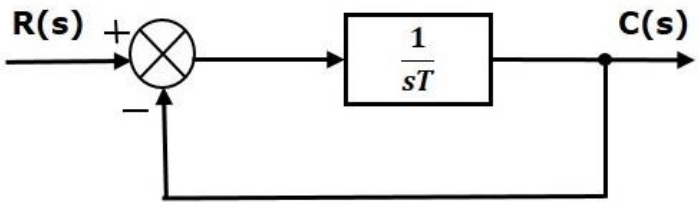

Figure 2.4: Block Diagram for First-Order Response [10]

The formula of the transfer function for a close loop feedback system is as shown in Equation (2.2):

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)} \tag{2.2}$$

Then substitute G(s)= $\frac{1}{sT}$ into Equation (2.2) and become Equation (2.3):

$$\frac{C(s)}{R(s)} = \frac{\frac{1}{sT}}{1 + \frac{1}{sT}} = \frac{1}{sT + 1} \tag{2.3}$$

From the Equation (2.3), the power of 's' is one in the denominator term. Hence, the above transfer function is of the first-order and the system is said to be the first-order system [10]. The step response graph of the first-order system is as shown in Figure 2.5.

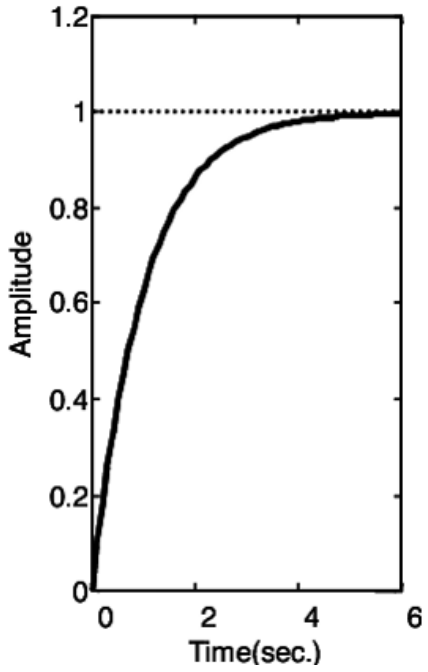

Figure 2.5: Step Response Graph of the First-Order System [11]

Meanwhile, a system whose input-output equation is a second-order differential equation is called second-order system. Refer to the block diagram of closed-loop control system as shown in Figure 2.6, an open-loop transfer function, $G(s) = \frac{\omega_n^2}{s(s+2\zeta\omega_n)}$ is connected with a unity negative feedback.

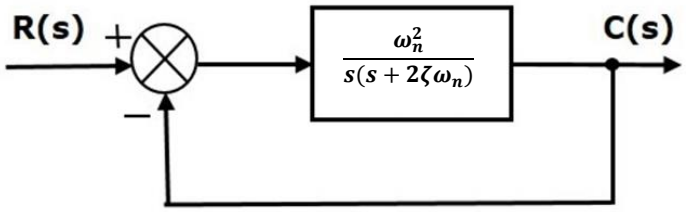

Figure 2.6: Block Diagram for Second-Order Response [12]

Then, substitute $G(s) = \frac{\omega_n^2}{s(s+2\zeta\omega_n)}$ into Equation (2.2) to become Equation (2.4):

$$\frac{C(s)}{R(s)} = \frac{\left(\frac{\omega_n^2}{s(s+2\zeta\omega_n)}\right)}{1+\left(\frac{\omega_n^2}{s(s+2\zeta\omega_n)}\right)} = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2} \quad (2.4)$$

The power of 's' is two in the denominator term. Hence, the transfer function as shown in Equation (2.4) is of the second-order and the system is said to be the second-order system. The characteristic equation is $s^2 + 2\zeta\omega_n s + \omega_n^2$=0, where $\boldsymbol{\omega}_n$ is the natural frequency and ζ is the damping ratio. If 0< ζ <1, system is named as under damped system. If ζ =1, system is named as critically damped system. Finally, if ζ >1, system is named as over damped system [12]. The effects of different types of damping are represented in Figure 2.7.

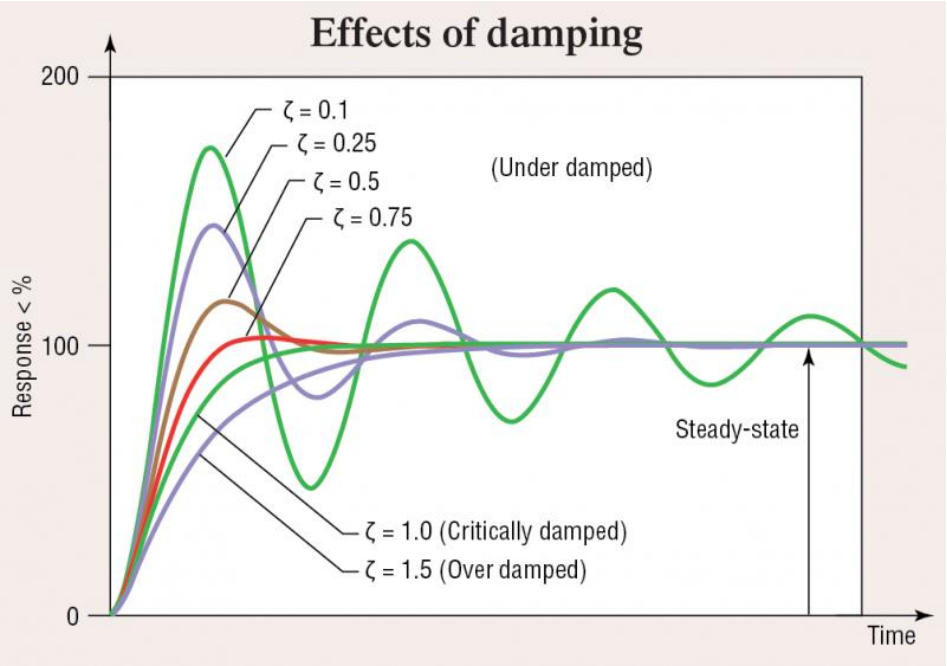

Figure 2.7: Effects of Damping [13]

The step response graph of the second-order system is as shown in Figure 2.8.

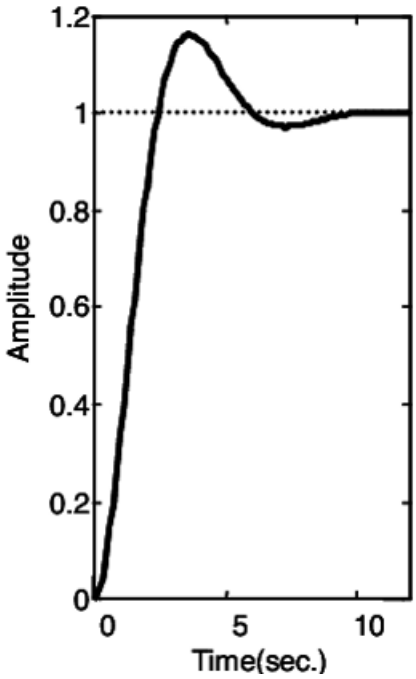

Figure 2.8: Step Response Graph of the Second-Order System [11]

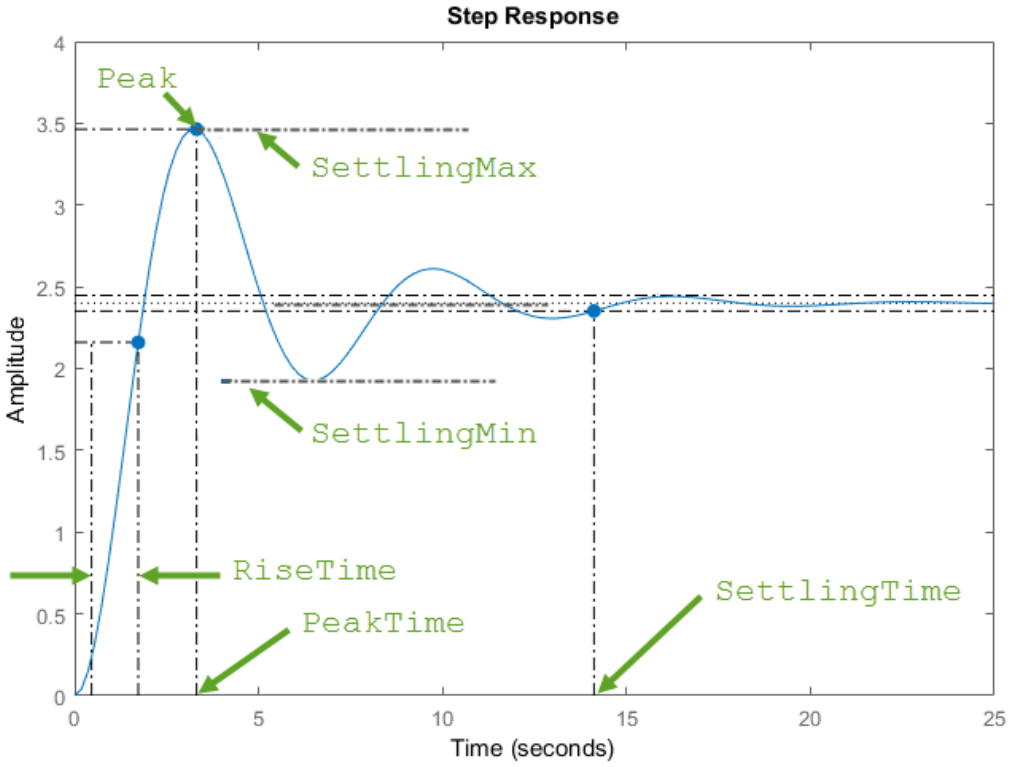

Figure 2.9: Quantities on a Typical Second-Order Response [14]

Figure 2.9 shows the characteristic that possessed by a second-order response. Rise time is the time taken for the response to rise from 10% to 90% of the steady-state

response. Meanwhile, settling time is the time taken for the error $|y(t) - y_{final}|$, which is the difference between the response y(t) and the steady-state response y_{final}, to fall to within 2% of y_{final}. Next, SettlingMin refer to the minimum value of y(t) once the response has risen and SettlingMax refer to the maximum value of y(t) once the response has risen. Besides, peak means peak absolute value of y(t) and peak time is the time at which the peak value occurs. Percentage overshoot is calculated using the Equation (2.5):

$$\text{Percentage of overshoot} = \frac{Peak - y\ final}{y\ final} \times 100\% \quad (2.5)$$

2.4 Background of PID

2.4.1 Proportional Action

The proportional control action is proportional to the error, e(t), and thus the expression is as shown in Equation (2.6).

$$u(t)=K_p\ e(t) = Kp\ [r(t)-y(t)] \quad (2.6)$$

where Kp is the proportional gain, r(t) is the SP and y(t) is the process output. The transfer function of a proportional controller can be derived trivially as:

$$C(s)= K_p \quad (2.7)$$

The primary downside of utilizing a pure proportional controller is that it cannot reach the exact SP and subsequently creates a steady-state error. However, a larger proportional gain can result in smaller steady state error besides less rise time taken and larger overshoot occurred.

2.4.2 Integral Action

The integral action is proportional to the integral of the error. The expression is as shown in Equation (2.8):

$$u(t)= K_i \int_0^t e(\tau)d\tau \quad (2.8)$$

where Ki is the integral gain. The integral action is related to the past values of the error and can return the process back to the exact SP. The corresponding transfer function is as shown in Equation (2.9):

$$C(s)=\frac{K_i}{s} \quad (2.9)$$

The steady state error will be wiped out, and SP will eventually be reached by having integral gain in controller. Nonetheless, an integral controller may introduce oscillations. If a positive error persists, then the integral term in the controller will increase. The result will be a positive input to the plant that will persist even after there is no more error. If so, the plant output will overshoot which introduces negative error [4]. Integral control also responds relatively slowly to an error signal and it causes a large deviation initially at the instant the error is produced which lead to system instability. Thus, the integral control mode is not typically utilized alone, but is combined with another control mode.

2.4.3 Derivative Action

While the proportional action is based on the current value of the error and the integral action is based on the past values of the error, the derivative action is based on the predicted future values of the error. An ideal derivative control law can be expressed as shown in Equation (2.10):

$$u(t)=K_d \frac{de(t)}{dt} \quad (2.10)$$

where K_d is the derivative gain. The corresponding controller transfer function is as shown in Equation (2.11):

$$C(s)= K_d s \quad (2.11)$$

Derivative action responses to abrupt changes in the process value, due to factors such as an external disturbance, so that control will quickly return to the original status.

2.4.4 PID Controller

Applying a PID control law comprises of applying appropriately the sum of three types of control actions: a proportional action, an integral action and a derivative one. This error signal (e) will be sent to the PID controller, and the controller computes the proportional, the derivative and the integral of this error signal. Refer to the Figure 2.10, the signal (u) past the controller is equivalent to the proportional gain (K_p) times the magnitude of the error plus the integral gain (K_i) times the integral of the error plus the derivative gain (K_d) times the derivative of the error.

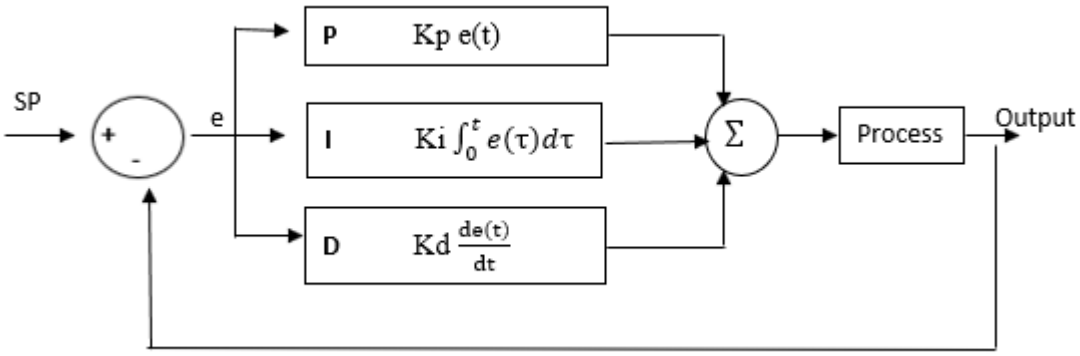

Figure 2.10: PID Control System

The formula for PID controller is as shown in Equation (2.12).

$$u(t) = K_p\, e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \qquad (2.12)$$

The function u will be used to determine new output which will be sent back to the sensor again to find the new e [15]. In a nutshell, by using PID action, the temperature is controlled smoothly by proportional action, automatic offset adjustment is made by integral control action and quick response to an external disturbance is made possible by derivative control action.

2.5 Background of PID tuning

2.5.1 Conventional Techniques

Conventional techniques are also considered as manual tuning. There are two types of manual tuning methods that are trial and error method and manual tuning in an analytical way such as Ziegler-Nichols method and Cohen-Coon method.

2.5.1.1 Trial and Error Method

The initial step for trial and error method is by setting the integral and derivative values to zero and then proportional gain is increased to reach the point where the output of the control loop equivocates at a constant rate. In the event that the increments of the K_p are not influencing the stability of the system, the response will accelerate. When proportional response reaches the appropriate speed, the integral value is then set to reduce the steady state error. However, this process will also increase the possibility of overshoot. After insignificant steady state error is accomplished by PI controller, the derivative value is added to diminish the overshoot of the system response.

The disadvantages of using trial and error are that the method is iterative and time-consuming, and if used on hardware, it can cause damage. However, for controlling an unstable system, trial and error method provides efficient and optimized control over the process when compared with the other approaches such as Ziegler-Nichols method. It is noticed that the use of a trial and error method can result in increased stability in the system with less overshoots and noticeable acceptable settling time [16].

2.5.1.2 Ziegler-Nichols Tuning Method

A more analytic method for tuning compared to the trial and error method is known as the Ziegler-Nichols tuning method. This method published by John G. Ziegler and Nathaniel B. Nichols is expected to accomplish a quick closed-loop step response without excessive oscillations and excellent disturbance rejection. There are two ways to carry out Ziegler-Nichols tuning method. First way requires the computation of the

ultimate gain, K_u and ultimate period, T_u based on closed-loop concepts [2]. The second way depends on open-loop concepts that depending on reaction curves [2].

The closed-loop Ziegler-Nichols tuning method considers the closed-loop system response to a step input with the PID controller in the loop. Initially the K_d and K_i respectively, are set to zero. The K_p is increased until the closed-loop system reaches the boundary of instability. The gain on the border of instability, denoted by K_u, is called the ultimate gain. The period of the sustained oscillations, denoted by T_u, is called the ultimate period. Once K_u and T_u are resolved, the PID gains are processed using the relationships in Table 2.1 according to the Ziegler-Nichols tuning method.

Table 2.1: Ziegler-Nichols PID Tuning Using K_u and T_u [2]

Ziegler-Nichols PID Controller Gain Tuning Using Closed-loop Concepts

Controller Type	K_P	K_I	K_D
Proportional (P) $G_c(s) = K_P$	$0.5K_U$	–	–
Proportional-plus-integral (PI) $G_c(s) = K_P + \frac{K_I}{s}$	$0.45K_U$	$\frac{0.54K_U}{T_U}$	–
Proportional-plus-integral-plus-derivative (PID) $G_c(s) = K_P + \frac{K_I}{s} + K_D s$	$0.6K_U$	$\frac{1.2K_U}{T_U}$	$\frac{0.6K_U T_U}{8}$

2.5.1.3 Cohen-Coon Method

A decade after Ziegler-Nichols published a paper based upon PID tuning, Cohen-Coon developed another tuning method. The Cohen-Coon method is a more complex version of the Ziegler-Nichols method. According to Amit Kumar and K.K Garg [17], the Cohen-Coon method is almost same as the Ziegler-Nichols method, but the difference comes with the fact that Cohen-Coon provides the faster rise time.

Cohen-Coon method of tuning is based on the open-loop testing. The response of the system is modelled as first-order response having dead time as:

$$G(s) = \frac{Ke^{-t_d s}}{(\tau s + 1)} \tag{2.13}$$

$$\text{Process gain, } K = \frac{\Delta \text{PV (in \%)}}{\Delta CV(in\ \%)} \tag{2.14}$$

where τ is the time constant of the system and t_d is the dead time

The above three parameters (K, τ and t_d) are calculated from Figure 2.11, and then these parameters are utilized to calculate the tuning parameters.

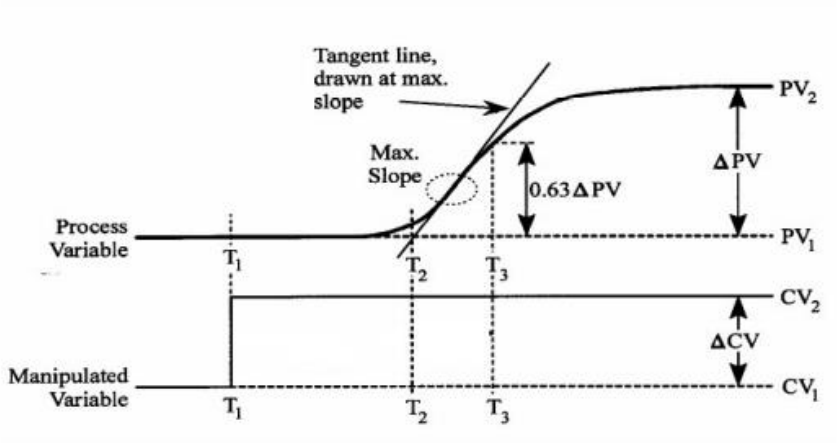

Figure 2.11: Cohen-Coon Method [18]

where t_d=T_2-T_1 and τ=T_3-T_2

Table 2.2: Cohen-Coon Controller Settings [18]

Parameters / Types of controller	Controller gain, K_c	Integral time, τ_i	Derivative time, τ_d
P	$\frac{1}{K}\frac{\tau}{t_d}(1+\frac{t_d}{3\tau})$	-	-
PI	$\frac{1}{K}\frac{\tau}{t_d}(0.9+\frac{t_d}{12\tau})$	$t_d\frac{30+3\frac{t_d}{\tau}}{9+20\frac{t_d}{\tau}}$	-
PID	$\frac{1}{K}\frac{\tau}{t_d}(\frac{4}{3}+\frac{t_d}{4\tau})$	$t_d\frac{32+6\frac{t_d}{\tau}}{13+8\frac{t_d}{\tau}}$	$t_d\frac{4}{11+2\frac{t_d}{\tau}}$

2.5.2 Computational and Intelligent Optimization Techniques

Conventional PID controller does not give acceptable performance for systems with uncertain dynamics, time delays and non-linearity [19]. Thus, intelligent methods are invented to automatically tune the PID parameters for obtaining satisfactory response. Computational models are used for auto tuning of PID controllers by originally set the PID parameters. Then, the process is modelled by using computational model and the outputs are compared to see if there are any process variations, in which case the PID parameters are reset to give the desired response. The examples of such techniques are Fuzzy Logic PID, Artificial Neural Network and Genetic Algorithm.

2.5.2.1 Fuzzy Logic (FL) PID

Fuzzy controllers execute a control strategy derived from linguistic rules, which are converted into mathematical terms through the concepts of fuzzy sets and fuzzy logic [20].

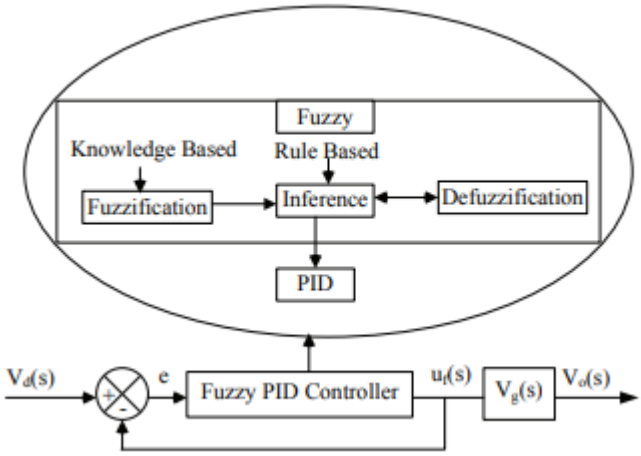

Figure 2.12: Block Diagram for Illustration of FL PID System [21]

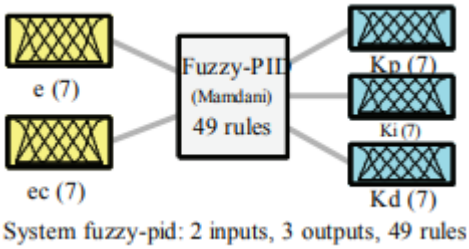

Figure 2.13: Overall View for FL PID System [21]

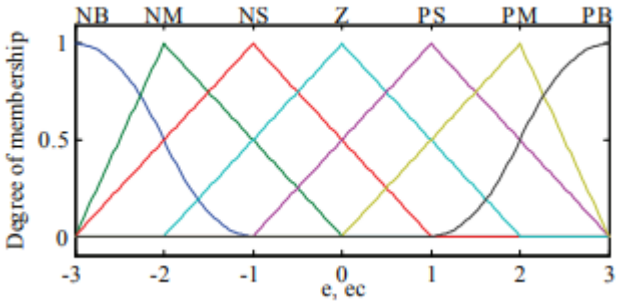

Figure 2.14: Inputs' Subsets [21]

The fuzzy system can separate into three sets as shown in Figure 2.12. The fuzzy controller has fuzzy and PID block. The fuzzy block is divided up into three blocks which the first block is the fuzzification that converts the crisp input to the linguistic variable. The fuzzification is a knowledge-based task. The input variables are to be mapped on fuzzy sets. The next block is the inference which is a rule-based task created by experts' knowledge. Defuzzification converts the linguistic variable back to a crisp value. As shown in the block diagram in Figure 2.12, V_o is the output voltage, u_f is the output obtained from the FL-PID which is input to the plant and e is the error between V_o and V_g.

In the work presented by Chauhan at al. [21], membership functions used for the input and output variables are as shown in Figure 2.13. There are two inputs, that are error (e) and change in error (ec) as shown in Figure 2.14. Each input has seven fuzzy subsets that are expressed as linguistic variables which represents the degree of direct current (DC) voltage. The seven subsets include NB (Negative Big), NM (Negative Medium), NS (Negative Small), ZE (Zero), PB seven fuzzy NB (Negative Large), NM (Negative Medium), NS (Negative Small), Z (Zero), PS (Positive Small), PM (Positive Medium) and PB (Positive Big). The number values are to be chosen as the linguistic variables as -3 for NB, -2 for NM, -1 for NS, 0 for Z, 1 for PS, 2 for PM, and 3 for PB. In this simple way, the numeric values are just for representation of designing the fuzzy logic but do not represent the true values of the DC voltage.

Next, “rule base” is created based on the knowledge of the conditions for controlling the DC microgrid voltage. Here the PID parameters are adjusted for the best control of DC microgrid voltage by using the feedback obtained from the voltage error itself. Membership function obtained for output variables K_p, K_i, and K_d are as shown in Figure 2.15. As there are two inputs and seven linguistic values, so there are $7^2 = 49$ rules for adjusting the every PID parameter.

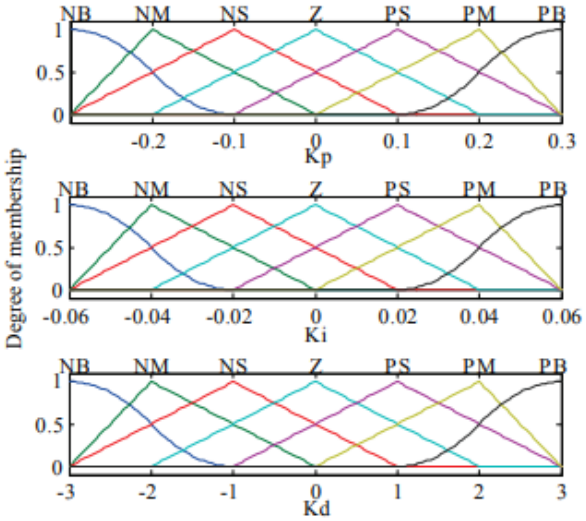

Figure 2.15: Subsets for Three Outputs (Kp, Ki and Kd) [21]

For all the problems with the conventional PID controller, Fuzzy PID controller method is better method of controlling to the complex and unclear model systems [20].

2.5.2.2 Artificial Neural Networks

An Artificial Neural Network (ANN) is a mathematical model or computational model that tries to simulate the structure of biological neural networks. It comprises of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. Much of the time ANN is an adaptive system that changes its structure dependent on external or internal data that flows through the network amid the learning phase.

Though ANN can model even highly non-linear systems, it is not used in control due to limited applicability in PID controllers, partially because the neural network

control design has some drawbacks such as the number of layers and the numbers of neurons per layer are often hard to be determined [22]. The flow chart for neural network based PID controller is as shown in Figure 2.16.

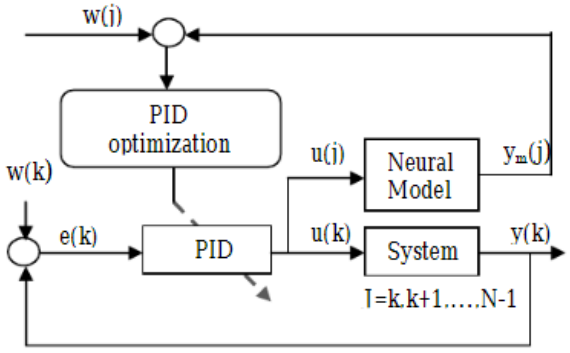

Figure 2.16: Flowchart of Neural Network-Based Controller [22]

2.5.2.3 Genetic Algorithm PID

A Genetic Algorithm is an optimization technique which is based on evolution theory. It represents an iteration process which each iteration is called the generation. The individuals in a population are represented by chromosomes and each of them is associated to a fitness value. Basic operations are selection, reproduction, crossover and mutation [23]. Parent selection gives more reproductive chances to the fittest individuals. During crossover some reproduced individuals cross and exchange their genetic characteristics. Mutation represents a change in the gene that flips a randomly selected gene in a chromosome.

The implementation of the tuning procedure through genetic algorithms starts with the definition of the chromosome representation. As illustrated in Figure 2.17, the chromosome is formed by three values that correspond to the three gains (K_p, K_i and K_d) that will be adjusted to achieve a satisfactory behaviour.

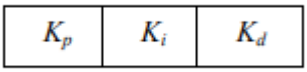

Figure 2.17: Chromosome Definition [23]

Since the objective is to minimize the error between the desired output and actual output, the fitness function has been defined as:

$$\text{Fitness} = 3^{n} [\text{desired output} - \text{actual output}]^{2} \qquad (2.15)$$

where n is the number of samples. The minimization of (2.15), performed by the Genetic Algorithm by adjusting the PID gains, will ensure that the actual output is as close as possible to the desired one. One of the advantages of using genetic algorithms is that it does not require a precise mathematical formulation for the problem. Besides, genetic algorithms are robust and optimal solution may be found within reasonable time.

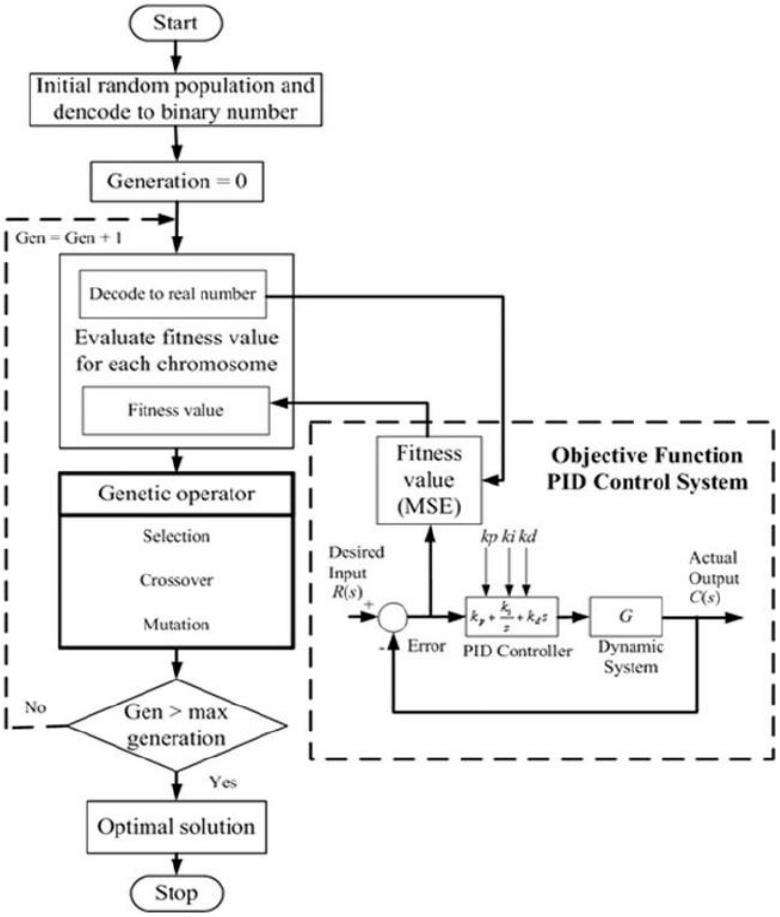

Figure 2.18: Flowchart of Genetic Algorithm-Based Tuning [24]

2.5.3 PID Tuning Software

PID tuning and loop optimization software are used to ensure consistent results. These software packages will gather the data, develop process models, and suggest optimal tuning.

2.5.3.1 LabVIEW

NI Company provides PID control toolkit used in LabVIEW, which can help engineers quickly and efficiently build a digital PID controller by combining with the NI data acquisition device, thus a complete system produces an accurate and reliable result [25]. The PID controller algorithm is simulated by using LabVIEW (G language) software. Installing LabVIEW PID Control Toolkit in NI CD Toolkit software can generate the toolkit in LabVIEW as shown in Figure 2.19.

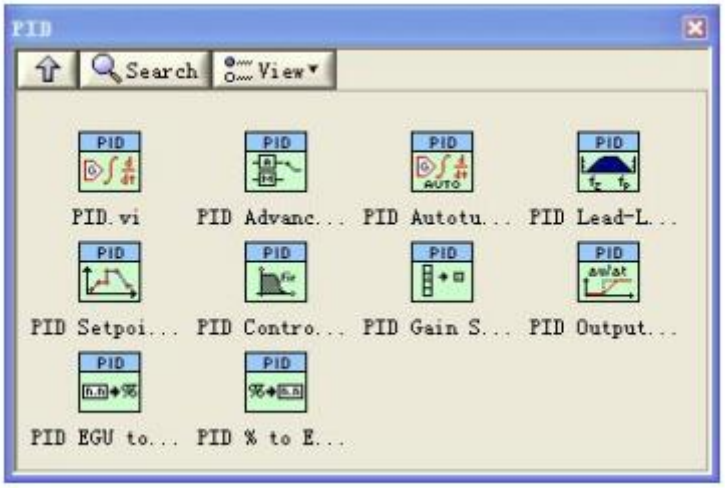

Figure 2.19: LabVIEW PID [25]

LabVIEW can do auto-tuning by using the features such as PID Advanced Auto-tuning, PID Auto-tuning, PID Auto-tuning (Temperature) and PID Auto-tuning Design as shown in Figure 2.19. The flowchart as shown in Figure 2.20 shows the processes of tuning for control the temperature in the LabVIEW.

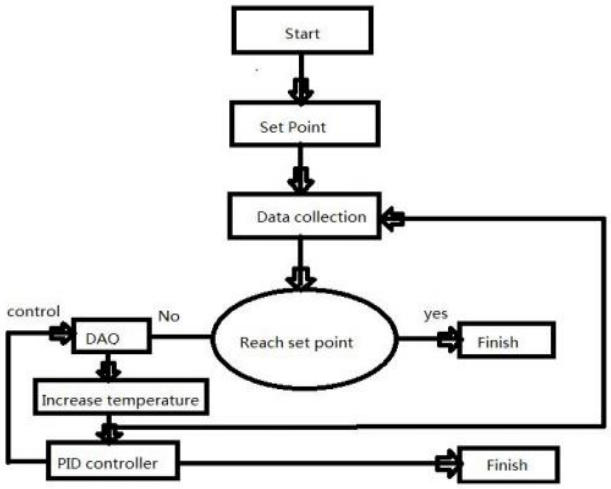

Figure 2.20: Temperature Control Flowchart for LabVIEW [25]

When the temperature reaches the SP, the system will finish the task. If not, the data acquisition board will give a feedback value to the system. By adjusting of the PID controller, the system will keep the heat device increase the temperature till it reaches the SP and finishes the task.

2.6 Temperature Control

Temperature controllers, such as on-off or PID controllers, control the temperature so that the PV will be the same as the SP. Generally, there are three forms of response desired from the temperature controllers. Most of the time as well as in this project, SP is required to be reached as quick as possible without overshooting as shown in Figure 2.22 [26]. Another case is that a response quickly increases the temperature even if it overshoots is required as shown in Figure 2.21 and there is also case which requires response slowly increases the temperature as shown in Figure 2.23 [26].

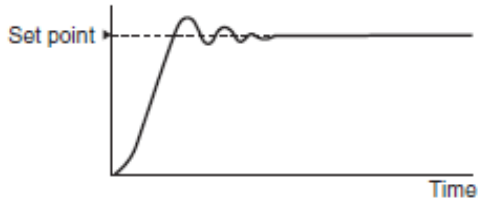

Figure 2.21: Response where Overshooting and Undershooting Occur while Reaching SP [26]