Sudan University of Science and Technology

College of Graduate Studies

DC Motor Speed Control Using Fuzzy Logic and Proportional-Integral-Derivative Controller

التحكم في سرعة محرك تيار مستمر باستخدام المنطق الغامض والمتحكم التناسبي التكاملي التفاضلي

Thesis Submitted in Partial Fulfillment of The Requirements for The Degree of Master of Computer and Network Engineering

By:

Alfatih Yassin Awad Babiker

Supervisor:

Dr. Mohammed Alnour Abdalla

March 2019

Sudan University of Science & Technology

College of Graduate Studies

بسم الله الرحمن الرحيم

كلية الدراسات العليا

جامعة السودان للعلوم والتكنولوجيا

كلية الدراسات العليا

Ref: SUST/ CGS/A11

Approval Page

(To be completed after the college council approval)

Name of Candidate: Alfatih Yassin Awad Babiker

Thesistitle: Dc Motor Speed Control Using Fuzzy Logic and Proportional – Integral – Derivative Controller

Degree Examined for: Master

Approved by:

1. External Examiner

Name: Dr. Othman Hassan Abdalla

Signature: .... Date: 25/5/2019

2. Internal Examiner

Name: Dr. Ala Eldin Abdallah Awada

Signature: .... Date: ....

3. Supervisor

Name: Dr. Mohammed Elnour AbdAlla

Signature: .... Date: 25/5/2019

cgs @ sustech edu. البريد الالكتروني 83 769363 /فاكس 407 ص.ب

الآية

قال تعالى:

(شَهِدَ اللّهُ أَنَّهُ لاَ إِلَـهَ إِلاَّ هُوَ وَالْمَلاَئِكَةُ وَأُوْلُواْ الْعِلْمِ قَآئِمَاً بِالْقِسْطِ لاَ إِلَـهَ إِلاَّ هُوَ الْعَزِيزُ الْحَكِيمُ).

صدق الله العظيم

سورة آل عمران الآية (18)

Dedication

To My Beloved Mother and Father

Acknowledgement

First and Foremost, I have to thank my research supervisor ***Dr. Mohammed Alnour Abdallah****. Without his assistance and dedicated involvement in every step throughout the process, this research would have never been accomplished. I would like to thank you very much for your support and understanding.*

ABSTRACT

DC Motors are widely used in industries for various purposes. Many situations demand changes in the speed of the DC Motor, and this makes necessity to develop a method to effectively control the speed of DC motor. There are conventional and digital controller types, Proportional Integral (PI), Proportional Derivative (PD), Proportional Integral Derivative (PID) and Fuzzy Logic Controller (FLC). The Fuzzy Logic Controller is used to improve the performance of each controller (PI, PD and PID) used to control speed of the DC motor. In this project PI, PD and PID (with and without Fuzzy Logic Controller) was modelled and simulated in MATLAB SIMULINK environment to control speed of the DC Motor. Aforementioned Controllers were analyzed and evaluated on four terms overshoot, rise time, steady state error and settling time, the purpose of the evaluation to find the best performance to control the speed of DC motor. As a result of this, the Fuzzy PID is the best controller because its improved performance when compared with others controllers.

المستخلص

تستخدم محركات التيار المستمر على نطاق واسع في الصناعات لأغراض متعددة. تتطلب العديد من الحالات التغير في سرعة محرك التيار المستمر، وهذا يتطلب ضرورة لإيجاد طريقة للتحكم الفعال في سرعة محرك التيار المستمر. حيث أن هناك أنواع من أجهزة التحكم التقليدية والرقمية مثل المتحكم التناسبي التكاملي (PI)، التناسبي التفاضلي (PD)، التناسبي التكاملي التفاضلي (PID) ومتحكم المنطق الغامض (FLC). متحكم المنطق الغامض تستخدم في تحسين الأداء لكل من المتحكم التناسبي التكاملي، التناسبي التفاضلي و التناسبي التكاملي التفاضلي للتحكم في سرعة محرك التيار المستمر. وفي هذا المشروع تم استخدام كل من طرق التحكم السابقة لتحسين أداء التحكم في سرعة محرك التيار المستمر. في هذا المشروع تم تصميم نموذج PI، PD وPID (مع وبدون متحكم المنطقي غامض) ومحاكاته في بيئة MATLAB SIMULINK للتحكم في سرعة التيار المستمر. وبعد تحليل وتقييم المتحكمات في أربعة خصائص هي التجاوز الأقصى، وزمن الصعود، خطأ الحالة المستقرة وزمن السكون بغرض الحصول على أفضل أداء. وكنتيجة لذلك، فإن المتحكم التفاضلي التكاملي التناسبي الغامض (FPID) هو أفضل متحكم بسبب تحسن الأداء مقارنة بالمتحكمات الأخرى.

List of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Meaning
AC	Alternating Current
Ce	Change in Error
DC	Direct Current
E	Error
FLC	Fuzzy Logic Controller
FPD	Fuzzy Proportional Derivation
FPI	Fuzzy Proportional Integral
FPID	Fuzzy Proportional Derivation Integral
GA	Genetic Algorithm
Kd	Differential Term
Ki	Integral Term
Kp	Proportional Term
PD	Proportional Derivation
PI	Proportional Integral
PID	Proportional Derivation Integral
PWM	Pulse Width Modulation

CHAPTER ONE

INTRODUCTION

CHAPTER ONE

INTRODUCTION

1.1 Preface

The development and applications of power electronics in industry has directly increased the use of Direct Current (DC) machine, because they have many good characteristics, high start torque characteristic, high response performance, easier to be linear control [1]. Nowadays, their uses are not limited to the car application (electrics vehicle), but also find applications in weak power using battery system (motor of toy) and for the electric traction in the multi-machine systems. The speed of DC motor can be adjusted or controlled easily to a great extend to provide easy controllability and high performance [2].

Mostly, the feedback control loop is indispensable for the desired and better performance of the system. However, this feedback system has slow response. For the fast-dynamic response of the system, many control strategies have been deployed in various feedback control systems. The vital role of controls in a drive system include precise and quick tracking for reference speed with minimum overshoot or undershoot and having little or eliminated steady state error[3].

To achieve optimal efficiency of induction motors, several control techniques have been developed to control the induction motor such as scalar control, vector or field-oriented control, direct torque control. Scalar control is one of the first control techniques of induction motors. In this method the ratio of both the amplitude and frequency of the supply voltage is kept constant in order to maintain a constant air gap flux and hence provide maximum torque. Scalar control drives are easy to implement but does not yield satisfactory results for high performance applications because of inherent coupling effects between torque and flux give sluggish response and system is easily prone to instability. This problem can be solved by field-oriented control or direct torque control. In most of industrial drive control applications, the standard method to control induction motor is based on the field oriented or vector control principle in order to achieve the best dynamic behavior. In this method the decoupling between the flux and torque allows the induction motor to be controlled in a similar method to that in the control of separately exited dc motors. Therefore, it can be used for high performance applications [4].

There are many techniques used to control DC motor, for example proportional integral derivation (PID) and Fuzzy Logic controller (FLC). Today most famous and most frequently used type of controller in industry is PID controller, but PID controllers don't offer satisfactory results when adaptive algorithm is required. The PID controller can be reduced the rise time since the proportional controller has effect on this. It can also eliminate the steady-state error by using the integral controller and lastly it can increase the stability of the system by using the derivative control [5].

Fuzzy Logic Controller offers some solutions. Basic advantages of Fuzzy Logic Controller, it does not require knowing complete mathematical model of system. Popularity of FLC is explained with fact that it puts clear and simple implementation of human thinking into controlling algorithm. Fuzzy controllers are robust regarding dynamic changes and have wide stability range. FLC only based on approximate and linguistic information [6].

Fuzzy Logic Controllers (FLCs) save many advantages, it is very easy to comprehend because the concept behind the control is very simple.

Fuzzy control possesses the intuition like a human which gives it the strength to adapt to the difficulties in the control and work well even with noisy input.

Finally, it does not consist of complex mathematical analysis, hence very easily designed and easy calculation. Extensively existing toolboxes and dedicated integrated circuits [7].

1.2 Problem Statement

DC motors are most widely used in industries. DC Motors speed may be changed, this sudden change is unacceptable in such systems, because this change lead to great harms in performance. Therefore, there is need for more efficient and reliable DC Motor controller for current and future applications.

1.3 Objectives

The main objectives of this research are to:

1. Modelling the DC motor for speed control purposes.
2. Design and simulate a conventional PID controller that can be used to control speed of the DC motor using MATLAB.
3. Design and simulate a Fuzzy-Logic PID controller that can be used to control speed of the DC motor using MATLAB.
4. Analyze and compare the performance of the DC motor (minimum overshoot, rise time, steady state error and settling time) using PID and Fuzzy Logic Controllers.

1.4 Methodology

This project evaluates the performance of two techniques used in DC motor control. In this project, based mathematical model controllers (PI, PD and PID) and Fuzzy Logic controller (FLC) as non-mathematical model (FPI, FPD and FPID), will simulated in MATLAB/SIMULINK. The system performance of simulate on is evaluated in terms of Rising Time, Overshoot, Settling Time and Steady State Error.

The system (plant) under control is a continuous-time system. The 'heart' of the controller is a digital computer. The system in this project is DC motor which controlled by based mathematical model controllers. Then FL controllers will be used to control motor to improve step response parameters. These parameters are Rising Time, Overshoot, Steady State Error and Settling time.

1.5 Thesis Layout

This research consists of five chapters.

Chapter One introduce the project, problem statement, solution and objectives of the project. The last section summarizes the methodology of the project.

Chapter Two introduces the concept of DC motor techniques. Also, it introduces the concept of PID and FL controllers and previous studies.

Chapter Three includes an overview of the simulation tools used in the research and the simulation process.

Chapter Four summarizes the results obtained after a whole simulation process and discussing it.

Chapter Five outlines the main conclusions and gives recommendations for future work.

CHAPTER TWO

LITERATURE REVIEW

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

A direct current (DC) motor is an electrical machine which converts direct current (DC) electrical power into mechanical power. DC motors have been used in almost every aspect of our daily life like in toys, fans, automobile drives. Small DC motors are used in toys, tools and appliances. Larger DC motors are used in electrical automobiles, propulsion systems and elevators. Industrially a good performing DC motor is required of high-speed controllability, steady and transient state stability and good Torque-Speed characteristics.

The speed of a DC motor is very easily controlled compared to AC motors. The making of highly controlled motors is critical for Industrial purposes. For a satisfactory operation, a DC motor must have an excellent speed tracking and regulation of load. DC motors are easily constructed when compared to the AC motors which are bulky. DC motors are very economical when the requirement of horse power is high [1].

2.2 Electrical Motors

There are two main types of electrical motors. There are direct current or DC and alternating current or AC motors. The reference of DC or AC refers to how the electrical current is transferred through and from the motor. Both types of motors have different functions and uses. DC motors come in two general types. They can have brushes or be brushless (synchronous motor). Then, AC motors come in two different types which are they can be single phase and three phases. In this project only, DC motors was covered.

There are several types of DC motor that commonly used in the industry such as brushed DC motor and the synchronous DC motor. The brushed DC motor generates torque directly from DC power supplied to the motor by using internal commutation, stationary permanent magnets, and rotating electrical magnets. Advantages of a brushed DC motor include low initial cost, high reliability, and simple control of motor speed. Although that, the brushed DC motor need maintenance regularly by replacing the brushes and springs which carry the electric current, as well as cleaning or replacing the commutated. These components are necessary for transferring electrical power from outside the motor to the spinning wire windings of the rotor inside the motor. The Figure 2.1 below shows the brushed DC motor [2].

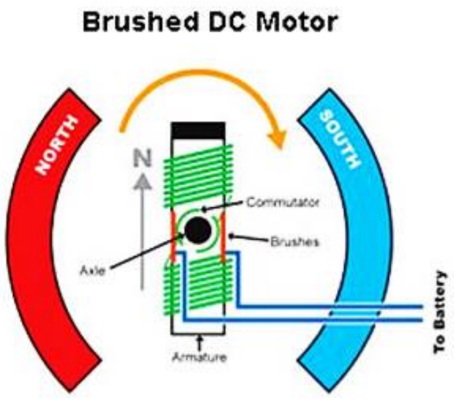

Figure2.1: Brushed DC Motor.

The second type of DC motors is synchronous, there are two types of synchronous DC motor which are the brushless DC motor and the stepper motor. Both require external commutation to generate torque. The motor is lockup if driven by DC power. Brushless DC motor use a rotating permanent magnet in the rotor and stationary electrical magnets on the motor housing. Brushless motor consists a controller that used to converts from DC to AC. This design is simpler than brushed motor because it eliminates the complication of transferring power from outside the motor to the spinning rotor. This type of motor needs no maintenance and more efficient compared to the brushed motor that discussed. Figure 2.2 shows that the brushless DC motor using three poles to operates [2].

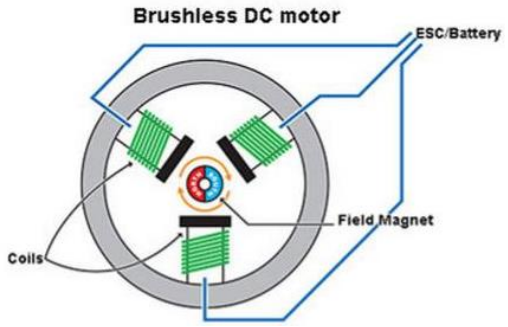

Figure2.2: Brushless DC Motor (synchronous).

The second type of synchronous motors is stepper motor which is the electric motor that can divide a full rotation into a large number of steps. The motor's position can be controlled precisely without any feedback mechanism. Stepper motors are similar to switched reluctance motors which are very large stepping motors with a reduced pole count, and generally are closed loop commutated. Figure 2.3 shows the operation of the stepper motor [2].

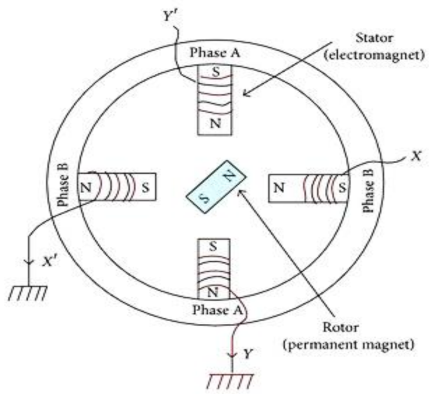

Figure2.3: Stepper DC Motor (synchronous).

2.3 DC Motor Operation

The DC motor operation is depending on supply and the flux which result from the supply, first the motor is excited by a separate supply to the field which creates a flux in the field. The supply for armature sets up a current in the armature and also a flux in the rotor. The interaction between these two fluxes creates a torque which starts to rotate the rotor. The field current is independent of the armature current. Thus, this motor can be controlled without disturbing the filed flux using the armature control [1].

2.4 Control Theory

Control theory is an interdisciplinary branch of engineering and mathematics that deals with the behavior of dynamical systems. The desired output of a system is called the reference. When one or more output variables of a system need to follow a certain reference over time, a controller manipulates the inputs to a system to obtain the desired effect on the output of the system. Figure 2.4 shows the closed loop control system [3].

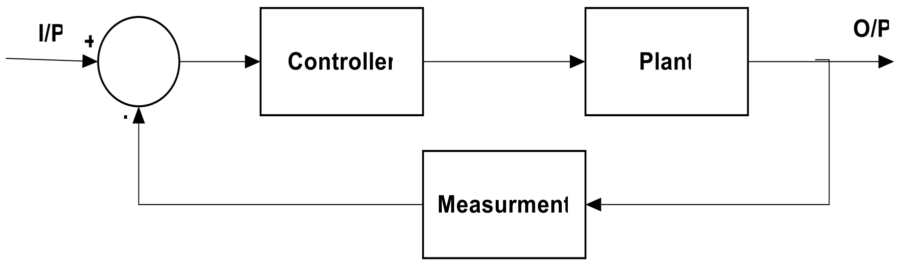

Figure 2.4: The Closed Loop Control System [3].

2.5 Speed Control Method of DC Motor

The speed of DC motor can be varied by controlling the field flux, the armature resistance or the terminal voltage that applied to the armature circuit (armature voltage). The three most common speed control methods are field resistance control, armature voltage control, and armature resistance control.

2.5.1 Field Resistance Control Method

In the field resistance control method, a series resistance is inserted in the shunt-field circuit of the motor in order to change the flux by controlling the field current. It is theoretically expected that an increase in the field resistance will result in an increase in the load speed of the motor and in the slope of torque speed curve [2].

2.5.2 Armature Voltage Control Method

In the armature voltage control method, the voltage applied to the armature circuit, is varied without changing the voltage applied to the field circuit of the motor. Therefore, the motor must be separately excited to use armature voltage control. When the armature voltage is increased, the no-load speed of the motor increases while the slope of torque speed curve remains unchanged since the flux is kept constant [2].

2.5.3 Armature Resistance Control Method

The armature resistance control is the less commonly used method for speed control in which an external resistance is inserted in series with the armature circuit. An increase in the armature resistance results in a significant increase in the slope of the torque speed characteristic of the motor while the no-load speed remains constant [2].

2.5.4 Pulse Width Modulation (PWM)

Pulse-width modulation (PWM), is a modulation technique that conforms the width of the pulse, formally the pulse duration, based on modulator signal information. Although this modulation technique can be used to encode information for transmission, its main use is to allow the control of the power supplied to electrical devices, especially to inertial loads such as motors. The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast pace. The longer the switch is on compared to the off periods, the higher the power supplied to the load is. The PWM switching frequency has to be much faster than what would affect the load, which is to say the device that uses the power. Typically switching have to be done several times a minute in an electric stove, 120 Hz in a lamp dimmer, from few kilohertz (kHz) to tens of kHz for a motor drive and well into the tens or hundreds of kHz in audio amplifiers and computer power supplies. The term duty cycle describes the proportion of 'on' time to the regular interval or 'period' of time. A low duty cycle corresponds to low power, because the power is off for most of the time. Duty cycle is expressed in percent, 100% being fully on [2].

2.6 Conventional controller

Conventional controllers were used widely in industry due to their simplicity and ease of implementation, furthermore it has given reasonable response in numerous applications. The most famous conventional feedback controller is PID controller.

2.6.1 PI Controller

P-I controller is mainly used to eliminate the steady state error resulting from P controller. However, in terms of the speed of the response and overall stability of the system, it has a negative impact. This controller is mostly used in areas where speed of the system is not an issue. Since P-I controller has no ability to predict the future errors of the system it cannot decrease the rise time and eliminate the oscillations. If applied, any amount of I guarantees set point overshoot [8].

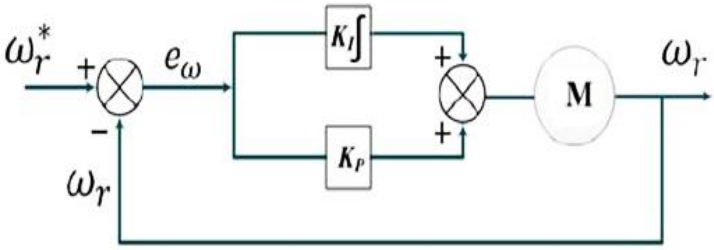

Figure 2.5 shows the block diagram of PI controller.

2.6.2 PD Controller

The aim of using P-D controller is to increase the stability of the system by improving control since it has an ability to predict the future error of the system response. In order to avoid effects of the sudden change in the value of the error signal, the derivative is taken from the output response of the system variable instead of the error signal. Therefore, D mode is designed to be proportional to the change of the output variable to prevent the sudden changes occurring in the control output resulting from sudden changes in the error signal. In addition, D directly amplifies process noise therefore D-only control is not used [8].

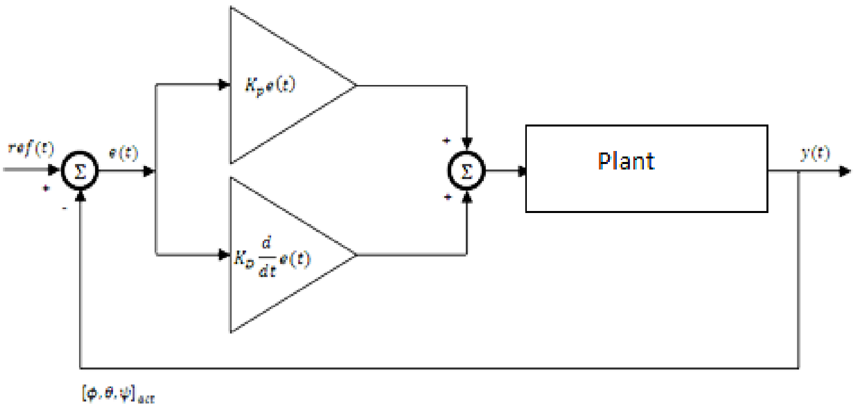

Figure 2.6: lock Diagram of PD controller.

2.6.3 PID Controller

A proportional–integral–derivative controller (PID controller) is a generic control loop feedback mechanism (controller) widely used in industrial control systems a PID is the most commonly used feedback controller. A PID controller calculates an error value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process control inputs. In the absence of knowledge of the underlying process, PID controllers are the best controllers. However, for best performance, the PID parameters used in the calculation must be tuned according to the nature of the system, while the design is generic, the parameters depend on the specific system. Figure 2.7 shows the block diagram of PID controller [5].

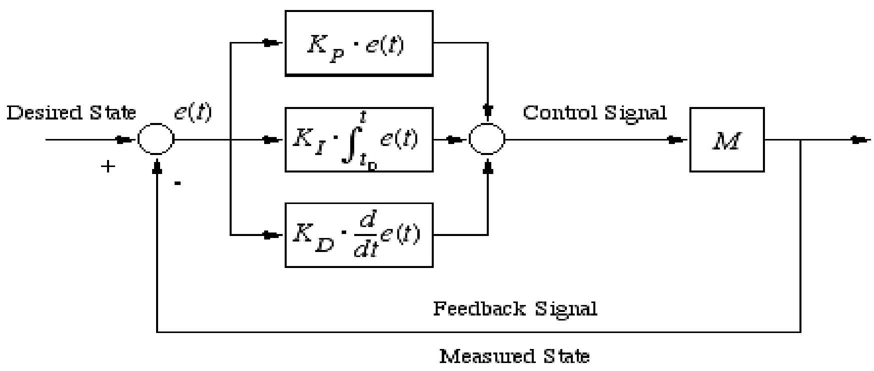

Figure 2.7: Block Diagram of PID controller.

2.7 PID Controller Theory

The PID controller is probably the most-used feedback control design. PID is an acronym for Proportional-Integral-Derivative, referring to the three terms operating on the error signal to produce a control signal. If u(t) is the control signal sent to the system, y(t) is the measured output and r(t) is the desired output, and tracking error:

$$e(t) = r(t) - y(t) \tag{2.1}$$

a PID controller has the general form:

$$u(t) = k_p e(t) + k_i \int e(t) + k_d \frac{d}{dt} e(t) \tag{2.2}$$

The desired closed loop dynamics is obtained by adjusting the three parameters kp , Ki and kd often iteratively by tuning and without specific knowledge of a plant model [4].

The proportional term **Kp** (sometimes called gain) makes a change to the output that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain. Larger values typically mean faster response since the larger the error, the larger the Proportional term compensation. An excessively large proportional gain will lead to process instability and oscillation [2][4].

Integral term **Ki** the contribution from the integral term (sometimes called reset) is proportional to both the magnitude of the error and the duration of the error. Summing the instantaneous error over time (integrating the error) gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain and added to the controller output. The magnitude of the contribution of the integral term to the overall control action is determined by the integral gain, k_i. Larger values imply steady state errors are eliminated more quickly. The trade-off is larger overshoot: any negative error integrated during transient response must be integrated away by positive error before we reach steady state [2][4].

Derivative term **Kd** the rate of change of the process error is calculated by determining the slope of the error over time (i.e., its first derivative with respect to time) and multiplying this rate of change by the derivative gain k_d. The magnitude of the contribution of the derivative term (sometimes called rate) to the overall control action is termed the derivative gain, k_d. Larger values decrease overshoot, but slows down transient response and may lead to bad performance due to signal noise amplification in the differentiation of the error [2][4].

PID Tuning a control loop is the adjustment of its control parameters (gain/proportional band, integral gain/reset, derivative gain/rate) to the optimum values for the desired control response. Good performance (bounded oscillation) is a basic requirement, but beyond that, different systems have different behavior, different applications have different requirements, and some desiderata conflict. Further, some processes have a degree of non-linearity and so parameters that work well at full-load conditions don't work when the process is starting up from no-load; this can be corrected by gain scheduling (using different parameters in different operating regions). PID controllers often provide acceptable control even in the absence of tuning, but performance can generally be improved by careful tuning, and performance may be unacceptable with poor tuning [4].

PID tuning is a difficult problem, even though there are only three parameters and in principle is simple to describe, because it must satisfy complex criteria within the limitations of PID control. There are several methods for tuning a PID loop. The most effective methods generally involve the development of some form of process model, then choosing P, I, and D based on the dynamic model parameters. Manual tuning methods can be relatively inefficient, particularly if the loops have response times on the order of minutes or longer. The choice of method will depend largely on whether or not the loop can be taken "offline" for tuning, and the response time of the system. If the system can be taken offline, the best tuning method often involves subjecting he system to a step change in input, measuring the output as a function of time, and using this response to determine the control parameters [4].

Manual Tuning is if the system must remain online, one tuning method is to first set Ki and Kd values to zero. Increase the Kp until the output of the loop oscillates, then the Kp should be set to approximately half of that value for a quarter amplitude decay type response. Then increase Ki until any offset is correct in sufficient time for the process. However, too much Ki will cause problem. Finally, increase Kd, if required, until the loop is acceptably quick to reach its reference after a load disturbance. However, too much Kd will cause excessive response and overshoot. A fast PID loop tuning usually overshoots slightly to reach the setpoint more quickly; however, some systems cannot accept overshoot, in which case an overdamped closed-loop system is required, which will require a Kp setting significantly less than half that of the Kp setting causing oscillation. Table 2-1 explain the Effects of increasing a parameter independently [4].

Ziegler-Nichols Tuning is another tuning method formally known as the Ziegler–Nichols method, introduced by John G. Ziegler and Nathaniel B. Nichols. As in the method above, the Ki and Kd gains are first set to zero. The P gain is increased until it reaches the ultimate gain, Ku, at which the output of the loop starts to oscillate. Ku and the oscillation period Pu are used to set the gains as shown in table 2-2 [5].

Table 2-1: Effects of increasing a (PID) parameter independently [4].

Parameter	Rise Time	Overshoot	Settling Time	Steady State Error
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Decrease significantly
Kd	Minor Decrease	Minor Decrease	Minor Decrease	No effect in Theory

Table 2-2: Ziegler-Nichols Tuning Method [5].

Control Type	Kp	Ki	Kd
P	0.5 Ku	-	-
PI	0.45 Ku	1.2 Kp/ku	-
PID	0.6 ku	2Kp/ku	0.6Kpku/8

2.8 Fuzzy Logic Theory

The Fuzzy theory was first put forward by L.A. Zadeh in 1965. He felt that the classical theory concentrates much on precision rather than easy and efficient controlling mechanism. Unlike classical sets, the Fuzzy sets have a certain degree of membership for each element.

Fuzzy sets depend on certain rules. The rule base is the most important requirement for the fuzzy logic. The rule base generally consists of various cases of If-Then rules. First the fuzzy sets and the membership functions are declared. Then the If-Then rules for the membership functions are decided for the particular control. The output is controlled by these rules on input. A typical If-Then rule consists of two parts. They are Antecedent and Consequence or Conclusion. The If statement is the

Antecedent and the Then statement is the Consequence {If - (Antecedent) & Then - (Consequence)} [1].

One of the main features of fuzzy logic is its ability to operate with vague or ambiguous concepts typical of qualitative reasoning, based on a mathematical support quantitative conclusion can be drawn from a set of observations and qualitative rules. Fuzzy logic control is the application of fuzzy inference process automation. A typical fuzzy controller infers the consequent of more or less large simple rules, this process of reasoning can be performed in parallel, yielding the result with a simple logical sum. This parallel processing capability allows even relatively complex controllers to perform the fuzzy inference in a minimum computation time.

In a fuzzy logic system, the inference mechanism decides what rules to apply for the corresponding inputs by matching the fuzzified inputs to the premises of the rules in the rule base. The inference mechanism provides a fuzzy set that indicates the certainty that the plant input should take the various values. The defuzzification is used to convert the fuzzy set produced by the inference mechanism into a crisp output to be used by the plant. The most important specifications of fuzzy logic control method are their fuzzy logical ability in the quality perception of system dynamics and the application of these quality ideas simultaneously for control system. A simple block diagram of a fuzzy logic system is shown in Figure 2.8 [2].

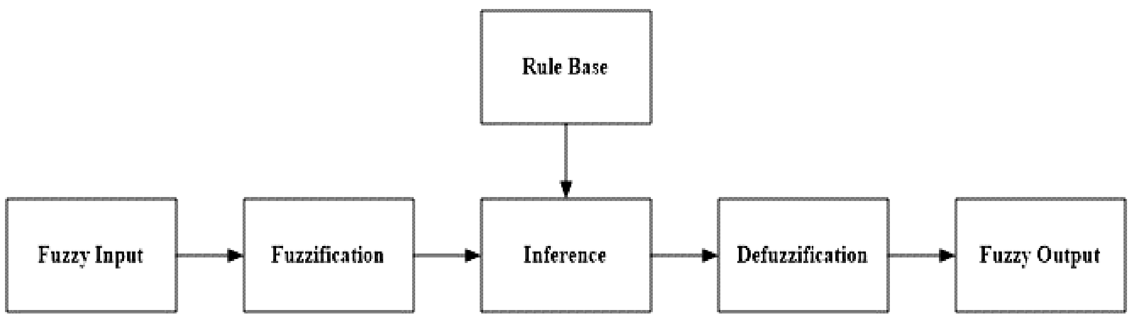

Figure2.8: Block Diagram of Fuzzy Logic System.

Fuzzy Input: the inputs are most often hard or crisp measurement from some measuring equipment is converted into fuzzification block.

Fuzzification: the fuzzification block performs the following tasks:

1. Measures the value of input variables.
2. Performs a scale mapping that transfers the range of values of input variables into the corresponding universes of discourse.
3. Performs the function of fuzzification, which converts input data into suitable linguistic values that may be viewed as labels of fuzzy sets [4].

Knowledge Based / Rule Based: the collection of rules is called a rule base. The rules are in “If Then” format and formally the If side is called the conditions and the Then side is called the conclusion. The computer is able to execute the rules and compute a control signal depending on the measured inputs error (e) and change in error (ce). In a rule-based controller the control strategy is stored in a more or less natural language. A rule base controller is easy to understand and easy to maintain for a non-specialist end user and an equivalent controller could be implemented using conventional techniques [4].

Fuzzy Rule Inference: fuzzy inference is of two methods. They are Mamdani and Sugeno. Mamdani’s methods of the Fuzzy interface is the most commonly used method. It was among the first control systems built using fuzzy set theory. It was first put forward by Ebrahim Mamdani as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators. This inference method expects the output variable to be fuzzy sets. It is possible and also efficient to use a single spike in the output as membership function rather than a distributed fuzzy set. This is known as singleton output membership function. It enhances the Defuzzification process because it greatly simplifies the computation required by the more general Mamdani method which finds the centroid of the two-dimensional function. But in the Sugeno type of inference can be used to model any inference system in which the output membership function is either linear or constant [1].

Defuzzification: is when all the actions that have been activated are combined and converted into a single non-fuzzy output signal which is the control signal of the system. The output levels are depending on the rules that the systems have and the positions depending on the non-linearities existing to the systems. To achieve the result, develop the control curve of the system representing the I/O relation of the systems and based on the information, define the output degree of the membership function with the aim to minimize the effect of the non-linearity [4].

Fuzzy Output: is output gain that can be tuned and also become as an integrator. The output crisp value can be calculated by the center of gravity or the weighted average [4].

2.9 Related Work

This section introduces the some of related and previous studied and works in this topic then it shows the final results of their work. Firstly, Purushotam Kumar, Prabhakar Kumar Prabhat, Mithun Kumar, Dr. S.D. Choudhary controlled DC motor speed with PID controller and first system is checked without controller on loaded and unloaded condition then add PID controller and system is tuned using its existing tuning methods. After its system is further tuned in order to get desired value with less steady state error.

They use Fuzzy controller to improve the motor speed control, they found that the Fuzzy controller provides better control strategies than other controllers. They used MATLAB\SIMULINK to simulate the required model.

Moreover, A. Fatah presented a hybrid PID-Fuzzy control system for the speed control of a three-phase squirrel cage induction motor. The proposed method incorporates fuzzy logic and conventional controllers with utilization of vector control technique. This method combines the advantages of fuzzy logic controller and conventional controllers to improve the speed response of the induction motor. The design of fuzzy system consists of 9 fuzzy variables and 49 IF-THEN rules that define the behavior of the system. The FLC observes the loop error signal and correspondingly control the PID input error signal so that the actual speed signal matches the reference speed signal with reduced rise time, settling time, and peak over shoot. Implementation and simulation results using MATLAB/SIMULINK of various hybrid system controllers such as (PI-, PD-, and PID fuzzy) are compared along with conventional PI controller in terms of several performance measurements such as rise time, maximum percent overshoot, settling time, and steady state error at various load conditions. The results verified the effectiveness of the proposed hybrid speed controller under different operating conditions and demonstrated improvements in performance in speed tracking and system’s stability [7].

KHAIRUL in [8], aimed to control speed of the DC Series Wound Motor. Four controllers (P, PI, PID and Fuzzy controller) were used in this project. Initially all the controllers are developed by using MATLAB Simulink model. In this project, PID, PI, and P controller are developed and tuned in order to get faster step response and the Fuzzy Logic Controller (FLC) is design based on the membership function and the rule base. Finally, The FLC is the best controller according to the final result of this project.

According to A.M.B. YUSOF, PID controller and FL controller were compared by simulating in MATLAB/SIMULINK in terms of Rise time, Settling Time, Percent Overshoot and Integral Absolute error. Fuzzy Logic was used to tune each parameter of the proportional, integral and derivative (Kp, Ki, Kd) gains of the PID controller. The FLC had two inputs. One is the motor speed error between the reference and actual speed and the second is changed in speed error (rate of change error). The output of the FLC i.e. The parameter of PID controller are used to control the speed of the DC motor. Finally the FUZZY-PID controller had tuned by trial and error and performance parameters are Rise time, Settling Time, Percent Overshoot and Integral Absolute error [4].

Finally, in [2] the performance of a selected DC motor controlled by a Proportional Integral Derivative (PID) controller and Fuzzy Logic Controller were investigated. An overshoot speed was observed with an accompanied settling time thereby confirming the behavior of a PID controller and Fuzzy Logic Controller. It was therefore a matter of necessity to tune the Fuzzy Logic Controller in order to obtain the desired performance. On the other hand, a Fuzzy Logic Controller was applied to the DC motor was investigated. With the application of appropriate expert rules, there was minimum overshoot and the settling time is within the desired value. With the Fuzzy Logic Controller, manual tuning was eliminated and intelligent tuning takes the center stage with satisfactory performance. The speed control of the DC motor was performed using PID and FLC in MATLAB environment. The results showed that the FLC approach had minimum overshoot, minimum transient and steady state error, which showed more effectiveness and efficiency of FLC than conventional PID controller.

In this project the PID controller and Fuzzy logic controller will be simulated in MATLAB environment to control of DC motor speed. Then many techniques will be compared in terms overshoot, settling Time, Stelling time and steady state error.

CHAPTER THREE

SYSTEM MODELING AND SIMULATION

CHAPTER THREE

SYSTEM MODELING AND SIMULATION

3.1 Introduction

This chapter will start with the method and alternatives that have been used from the beginning until the end of this project. Beginning from mathematical modelling of DC motor until the PD, PI, and PID controller have been designed. In the last part, FPD, FPI, FPID were designed to control DC motor speed. MATLAB environment used to simulate the previous controllers, then compare in terms of overshoot, settling time, rise time and steady state error.

3.2 DC Motor Model

Figure 3.1 shows Schematic representation of the considered DC motor. The rotor and the shaft are assumed to be rigid [6].

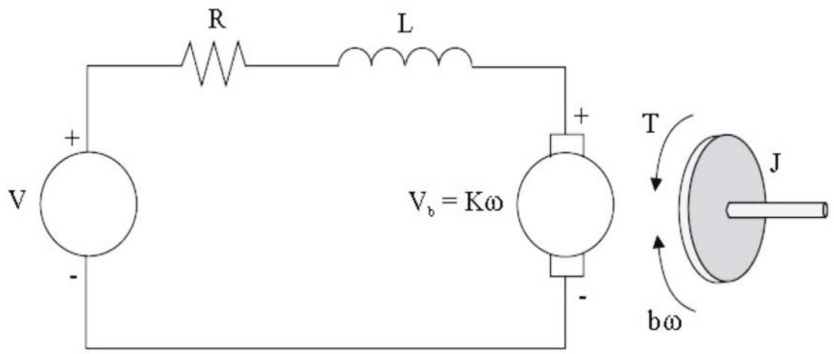

Figure 3.1: Schematic representation of the considered DC motor.

For this project, the following values for the physical parameters of the DC motor have been used in this project [9].

Table 3-1: DC Motor Parameters Values [12].

Parameter	Value
Armature resistance (Ra)	5 Ω
Armature inductance (La)	0.2 H
Electromagnetic torque constant (Kt)	0.5
Back electromotive force constant (Kb)	1.25 Nm/A
Moment of inertia of the rotor (J)	0.1 kg.m^2
Damping friction of the mechanical system (B)	0.008 Nms

The motor torque, T, is related to the armature current, i, by a constant factor K:

$$T = K_t I_a \tag{3.1}$$

The back electromotive force (EMF), Vb, is related to the angular velocity by:

$$V_b = K_b \omega \tag{3.2}$$

From Figure 3.1, the following equations based on the Newton's Law combined with the Kirchhoff's Law can be written as:

$$J \frac{d\omega}{dt} + B\omega = T \tag{3.3}$$

$$L_a \frac{dI_a}{dt} + \mathrm{R}_a I_a = V_i - V_b \tag{3.4}$$

Using the Laplace transform, equations (3.3) and (3.4) can be written as:

$$Js\omega(s) + B\omega(s) = T(s) \tag{3.5}$$

$$L_a s I_a(\mathrm{s}) + \mathrm{R}_a I_a(s) = V_i(\mathrm{s}) - V_b(s) \tag{3.6}$$

where (s) denotes the Laplace operator. substitute (3.6) in (3.5) to obtain:

$$Js\omega(s) + B\omega(s) = K_t I_a(s) = K_t\left[\frac{V_i(s)-V_b(s)}{L_a s+R_a}\right] \tag{3.7}$$

From equation (3.7), the transfer function from the input voltage, V(s), to the output angle, θ , directly follows:

$$\omega(s) = K_t \frac{V_i(s)-V_b(s)}{(L_a s+R_a)(Js+B)} \tag{3.8}$$

From the block diagram in Figure 3.2, after applying feedback for the internal block, it is easy to see that the transfer function from the input voltage, V(s), to the angular velocity, ω, is:

$$\frac{\omega(s)}{V(s)} = \frac{K_t}{(L_a s+R_a)(Js+B)+K_t K_b} \tag{3.9}$$

The final transfer function of the DC motor is going to be as following:

$$\frac{\omega(s)}{V(s)} = \frac{K_t}{JL_a s^2+(JR_a+BL_a)s+(BR_a+K_t K_b)} \tag{3.10}$$

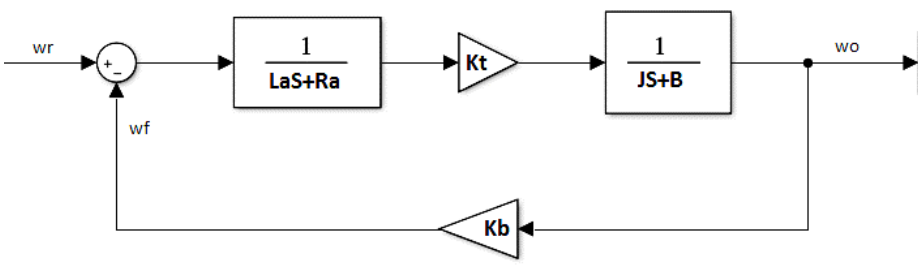

Figure 3.2: DC Motor Block Diagram.

After substitution of system parameters that are shown in table 3-1, the dc motor transfer function is given by the following equation.

$$\boldsymbol{G(s)} = \frac{\mathbf{25}}{\boldsymbol{s}^2+\mathbf{25.08}\boldsymbol{s}+\mathbf{33.25}} \tag{3.11}$$

This system is second order over-damped system with un-damped natural frequency of 5.7663 rad/s, damping ratio of 2.1747 and two poles -1.4044, -23.6756. figure 3.3 shows model with parameters. All simulation results of different controllers will be shown in chapter four.

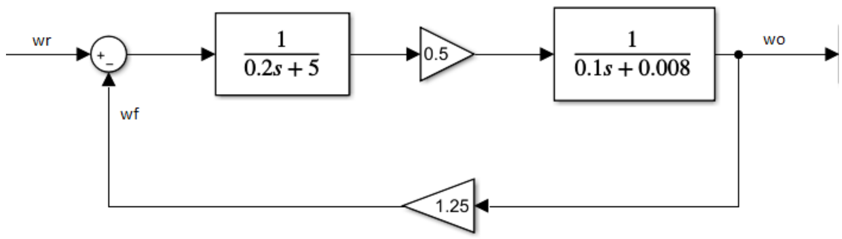

Figure 3.3: DC Motor without Controller.

3.3 Conventional Controller Modelling

3.3.1 PI Controller

Firstly, DC motor was modelled and simulated using PI controller (as shown in figure3.4) to control speed. PI controller is tuned using Zeigler-Nicholas Tuning method with tuning values are Kp= 3.359 and Ki= 4.717.

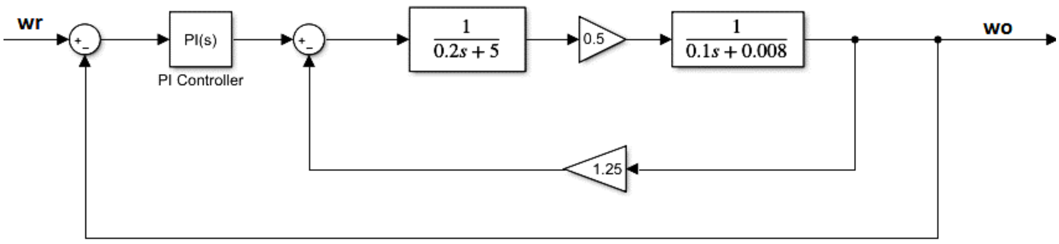

Figure 3.4: DC Motor speed control using PI Controller.

3.3.2 PD Controller

As shown in figure 3.5, DC motor was modelled and simulated using PD controller to control speed. PD was tuned by Zeigler-Nicholas Tuning method with tuning values are Kp= 104.6 and Kd= 1.978.

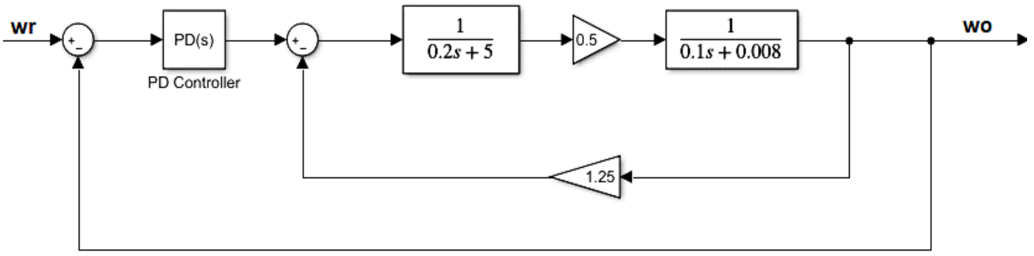

Figure 3.5: DC Motor speed control using PD Controller.

3.3.3 PID Controller

PID controller is mainly to adjust an appropriate proportional gain (*KP*), integral gain *(KI)*, and differential gain (*KD*) to achieve the optimal control performance. Ziegler- Nichols is a type of continuous cycling method for controller tuning, *kp* is increased from small value till the point at which the system goes too unstable.

The gain at which system starts oscillating is noted as the ultimate gain *(Ku)* and period of oscillations is the ultimate time period *(Pu),* these two parameters are used to find the loop-tuning constants of the PID controller using Zeighler-Nicholas method for tuning. PID controller system figure 3.6 shows the Simulink model for PID controller and tuning values are Kp= 1.895, Ki= 4.717 and Kd= -0.0067.

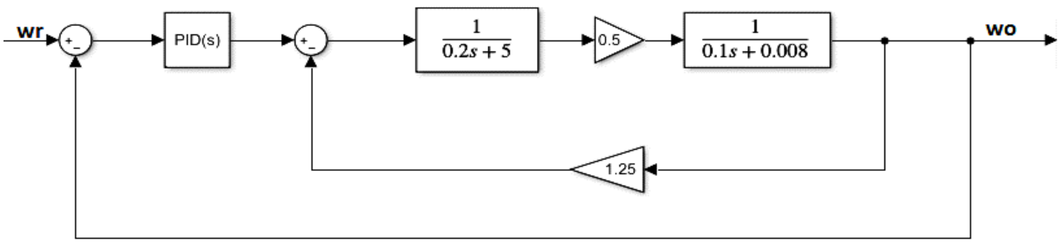

Figure 3.6: Simulink Model for System using PID controller.

3.4 Fuzzy Logic Controller Modelling

Fuzzy logic is express by means of the human language. Based on fuzzy logic, a fuzzy controller converts a linguistic control strategy into automatic control strategy, and fuzzy rules are constructed by expert experience or knowledge database.

The error *e(t)* and the change of error *ce(t)* of the angular velocity is the variable inputs of the fuzzy logic controller. The control voltage u(t) is the variable output of the fuzzy logic controller.

$$e(t) = r(t) - u(t) \quad (3.12)$$

$$ce(t) = e(t) - e(t-1) \quad (3.13)$$

Fuzzy controller is used to improve the performance of PI, PD and PID controllers. The universe of discourse of fuzzy variable sets including speed control error *(e)*, change of error *(ce)* are defined as :NL means negative large, NS means negative small, ZE means zero, PS means positive small, PL means positive large, and *kp* (the change of *kp*), *ki* (the change of *ki*) and *kd* (the change of *kd*) are defined as :PVS means positive very small, PS means positive small, PMS means positive medium small, PM means positive medium, PML means positive medium large, PL means positive large, PVL means positive very large.

Figure 3.7 and figure 3.8 shows the input values (error and change of error) which applied in all controllers when using fuzzy controller.

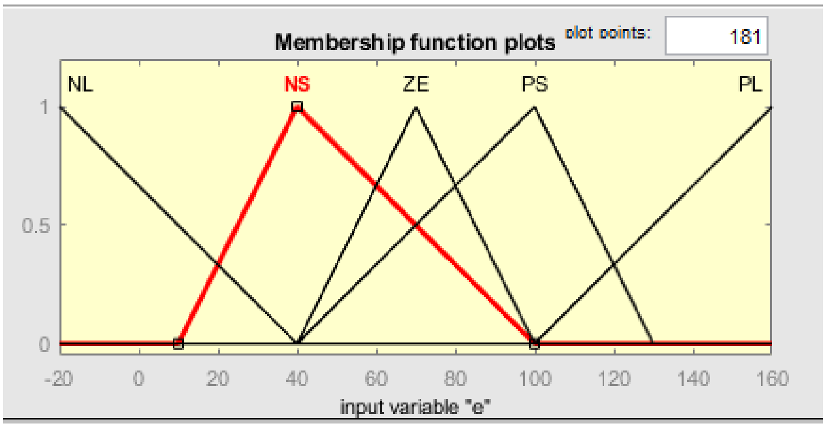

Figure 3.7: speed control error *(e)*.

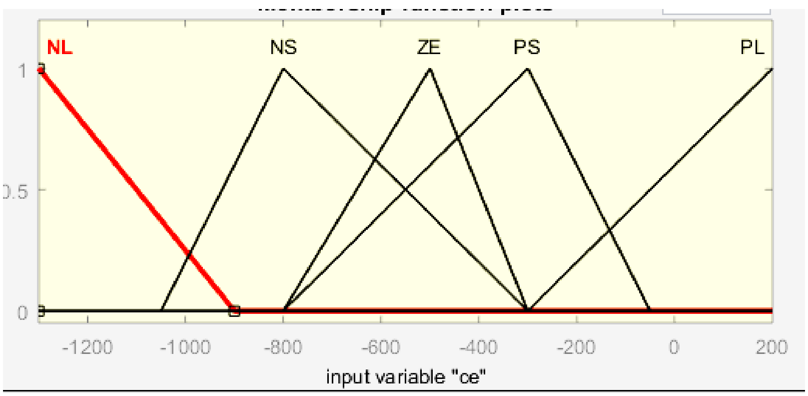

Figure 3.8: change speed error *(ce)*.

3.4.1 Fuzzy PI Controller

Figure 3.9 shows the Simulink models which is used to simulate Fuzzy PI controller. In Fuzzy PI controller, Fuzzy controller used for tuning PI controller, there are two output control signals (Kp and Ki) as shown in figure 3.9 and 3.10. The rules applied described in tables 3-2 ,3-3.

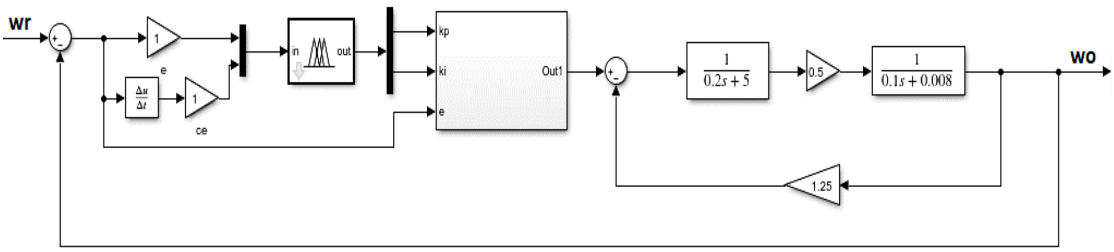

Figure 3.9: Simulink Model of Fuzzy PI Controller.

Figure 3.10 shows the PI controller subsystem which used in Fuzzy PI controller.

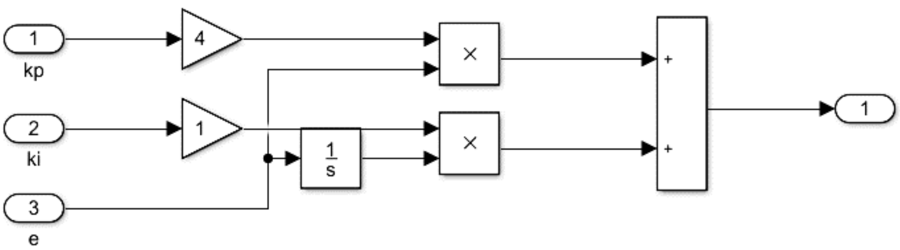

Figure 3.10: Simulink Model of Fuzzy PI Controller subsystem.

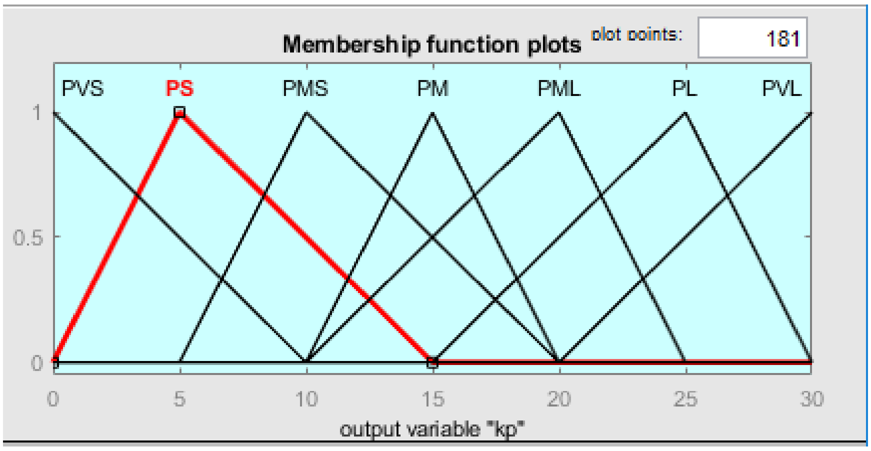

Figure 3.11: the degree of membership of change *kp*.

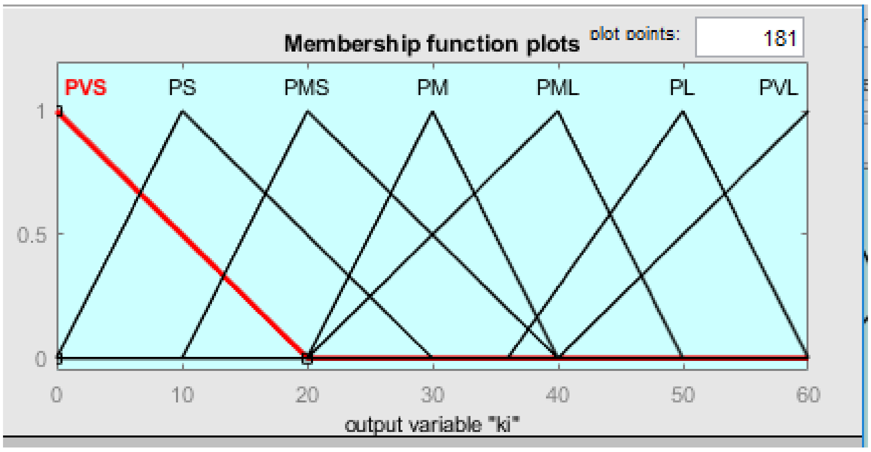

Figure 3.12: the degree of membership of change *Ki*.

Table 3-2: The rules applied in PI controller for Kp.

e\ce	**NL**	**NS**	**ZE**	**PS**	**PL**
NL	PVL	PVL	PVL	PVL	PVL
NS	PML	PML	PML	PL	PVL
ZE	PVS	PVS	PS	PMS	PMS
PS	PML	PML	PML	PL	PVL
PL	PVL	PVL	PVL	PVL	PVL

Table 3-3: The rules applied in PI controller for Ki.

e\ce	NL	NS	ZE	PS	PL
NL	PM	PM	PM	PM	PM
NS	PMS	PMS	PMS	PMS	PMS
ZE	PS	PS	PVS	PS	PS
PS	PMS	PMS	PMS	PMS	PMS
PL	PM	PM	PM	PM	PM

3.4.2 Fuzzy PD Controller

Figure 3.13 shows the Simulink models which is used to simulate Fuzzy PD controller. In Fuzzy PD controller, Fuzzy controller used for tuning PD controller, there are two output control signals (Kp and Kd) as shown in figure 3.14 and 3.15. The rules applied described in tables 3-4 (for Kp) and 3-5 (for Kd).

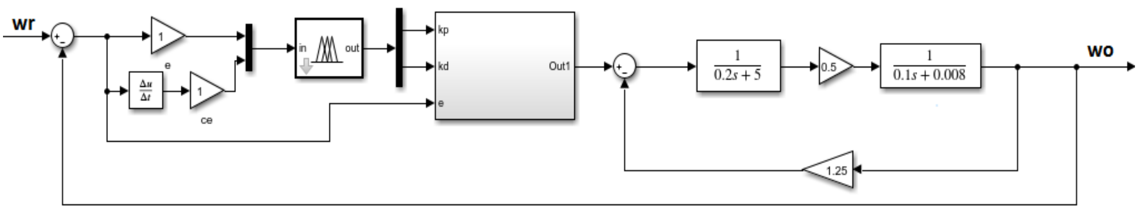

Figure 3.13: Simulink Model of Fuzzy PD Controller.

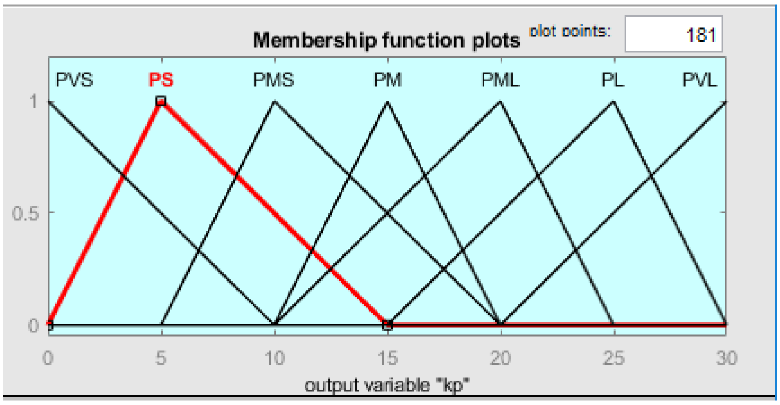

Figure 3.14: the degree of membership of change *kp*.

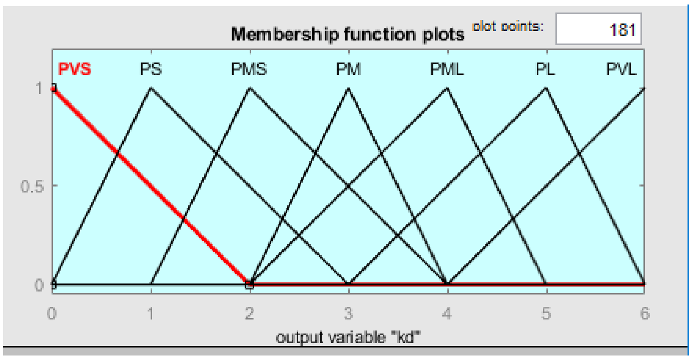

Figure 3.15: the degree of membership of change *Kd.*

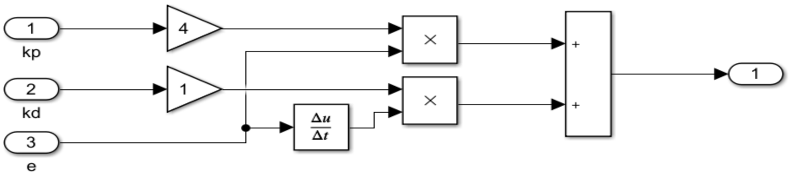

Figure 3.16: Simulink Model of Fuzzy PD Controller subsystem.

Table 3-4: The rules applied in PD controller for Kp.

e\ce	**NL**	**NS**	**ZE**	**PS**	**PL**
NL	PVL	PVL	PVL	PVL	PVL
NS	PML	PML	PML	PL	PVL
ZE	PVS	PVS	PS	PMS	PMS
PS	PML	PML	PML	PL	PVL
PL	PVL	PVL	PVL	PVL	PVL

Table 3-5: The rules applied in PD controller for Kd.

e\ce	**NL**	**NS**	**ZE**	**PS**	**PL**
NL	PVS	PMS	PM	PL	PVL
NS	PMS	PML	PL	PVL	PVL
ZE	PM	PL	PL	PVL	PVL
PS	PML	PVL	PVL	PVL	PVL
PL	PVL	PVL	PVL	PVL	PVL

3.4.3 Fuzzy PID Controller

The structure of the fuzzy auto tuning PID controller designed for control speed of dc motor is shown in figure 3.17. its inputs are control error *(e)* and the change of control error *(ce)*. the fuzzy auto tuner block adjusts the parameter of the incremental PID controller, and the incremental PID controller calculates the control output.

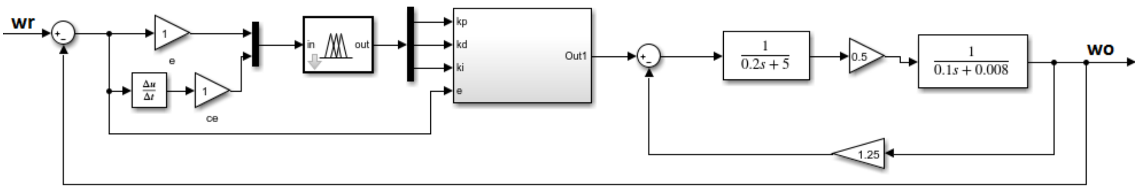

Figure 3.17: Simulink Model of Fuzzy PID Controller.

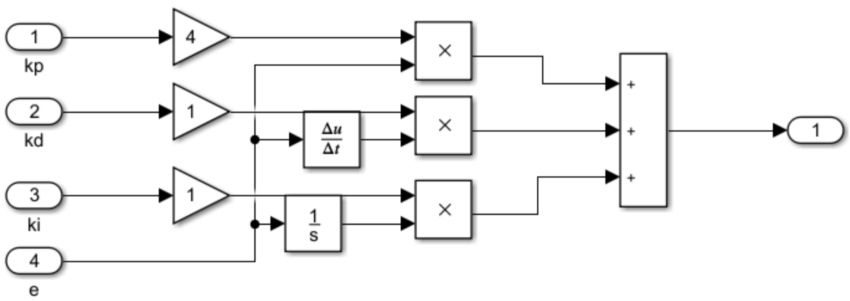

Figure 3.18: Simulink Model of fuzzy PID subsystem.

The fuzzy auto-tuning of PID controller is to find the fuzzy logic relationship between three parameters of PID with error *(e)* and change of error *(ce)*, calculate *(e)*and *(ce)*in cycle in the operation of control system and adjust *(kp)*,*(ki)* and *(kd)* on-line according to the fuzzy logic control principle .

In this way, the fuzzy tuner can satisfy the different requirements of PID controller parameters with different *(e)* and *(ce)*, and make the controlled object possess good dynamic and static performance. when tuning, the fuzzy tuner should adjust the value of *(kp)*, *(ki)* and *(kd)* in the

PID controller comprehensively taking into consideration of all control specifications including system response speed, overshoot, steady-state control error etc.

The role of proportional control action is to speed up the control response of controlled system, and to improve control accuracy. the integral control action is taken to eliminate the steady state control error, and the role of differential control action is to improve the dynamic property of control system.

Set the membership function of fuzzy variables, the degree of membership of change *kp (kp)* is shown in Figure (3.19), the degree of membership of change *Ki (Ki)* is shown in Figure (3.20), the degree of membership of change *kd (kd)* is shown in Figure (3.21).

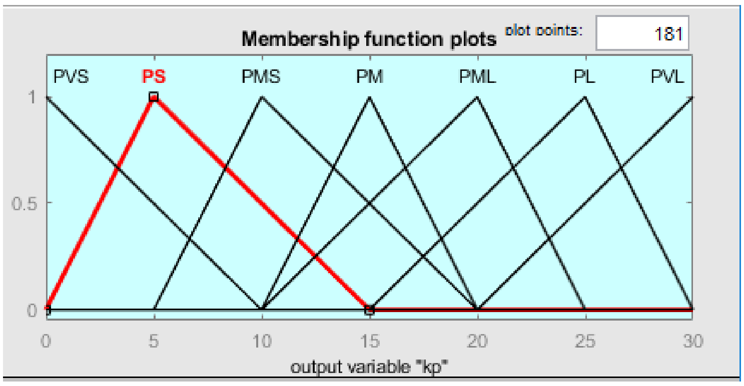

Figure 3.19: the degree of membership of change *kp.*

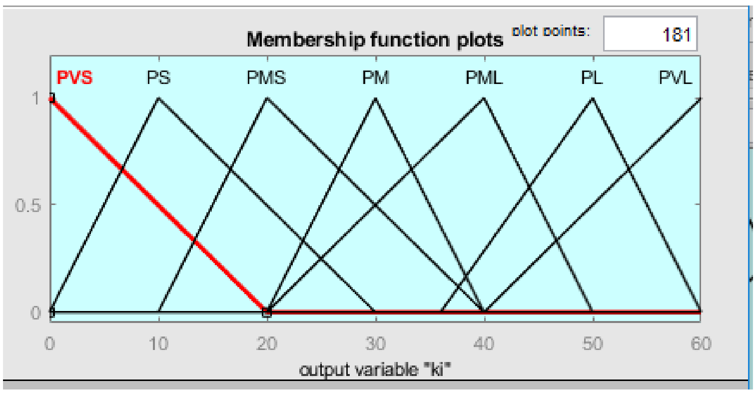

Figure 3.20: the degree of membership of change *Ki.*

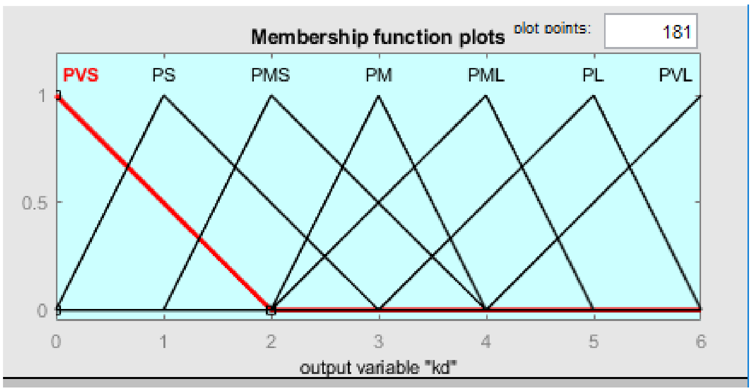

Figure 3.21: the degree of membership of change *kd.*

The fuzzy tuning rule sets of *(kp), (ki)* and *(kd)* are designed as illustrated in table 3-6, 3-7 and 3-8.

Table 3-6: The rules applied in PID controller for Kp.

e\ce	**NL**	**NS**	**ZE**	**PS**	**PL**
NL	PVL	PVL	PVL	PVL	PVL
NS	PML	PML	PML	PL	PVL
ZE	PVS	PVS	PS	PMS	PMS
PS	PML	PML	PML	PL	PVL
PL	PVL	PVL	PVL	PVL	PVL

Table 3-7: The rules applied in PID controller for Ki.

e\ce	**NL**	**NS**	**ZE**	**PS**	**PL**
NL	PM	PM	PM	PM	PM
NS	PMS	PMS	PMS	PMS	PMS
ZE	PS	PS	PVS	PS	PS
PS	PMS	PMS	PMS	PMS	PMS
PL	PM	PM	PM	PM	PM

Table 3-8: The rules applied in PID controller for Kd.

e\ce	NL	NS	ZE	PS	PL
NL	PVS	PMS	PM	PL	PVL
NS	PMS	PML	PL	PVL	PVL
ZE	PM	PL	PL	PVL	PVL
PS	PML	PVL	PVL	PVL	PVL
PL	PVL	PVL	PVL	PVL	PVL

For example, the rules applied in Fuzz PID as show in figure 3.22, if e is equal to zero (NL) and change of error equal to zero (NL), then kp equal to 26.4 (PVL), ki equal to 30 (PM) and kd equal to 5.27 (PVS).

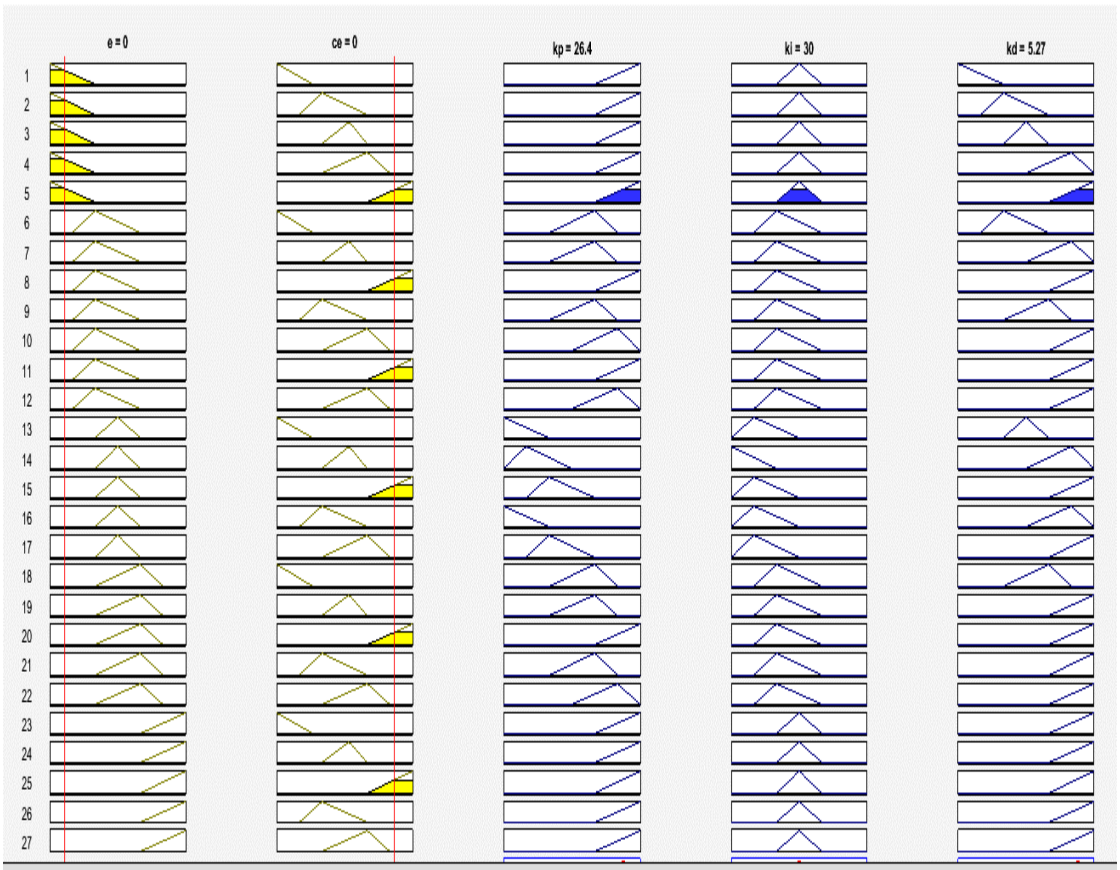

Figure 3.22: The rule viewer of Fuzzy PID.

CHAPTER FOUR

RESULTS AND DISCUSSION

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, the simulation results are presented. PD, PI, PID, FPD, FPI and Fuzzy PID are evaluated in terms of overshoot(Mp), rise time (Tr), settling time (Ts) and steady state error. The previous parameters can be calculated as following:

$$Mp = e^{-(\sigma/\omega_d)\pi} \tag{4.1}$$

$$Tr = \frac{\pi - \tan^{-1}\frac{\omega_d}{-\sigma}}{\omega_d} \tag{4.2}$$

$$Ts = \frac{4}{\vartheta\omega_n} \tag{4.3}$$

4.2 System without Controller

According to figure 4.1, the system was simulated without using any controller. In this case the step response does not reach the desired value, rise time is too high and the steady state error is very high.

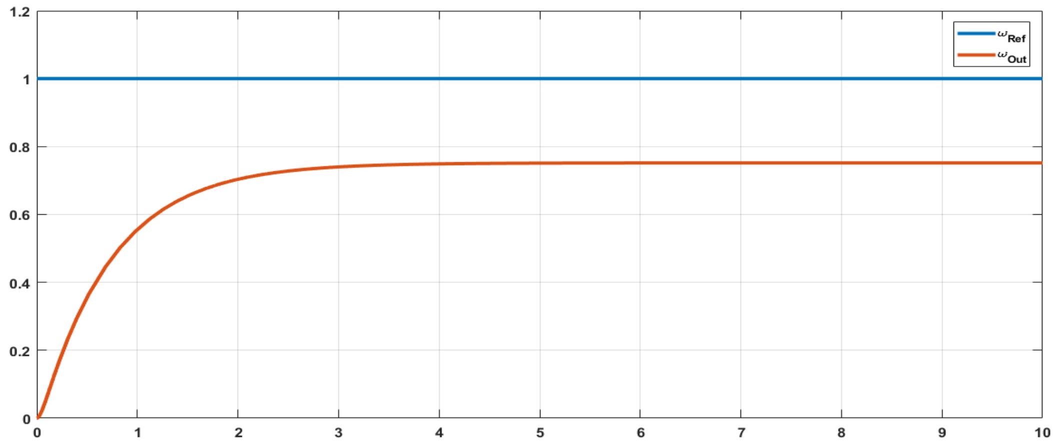

Figure 4.1: Step Response for the System without Controller.

4.3 Control System using PD & PI controller

Figure 4.2 shows the response of PD controller. In PD controller the step response rise time is very low, minimum over shoot. But there is steady state error.

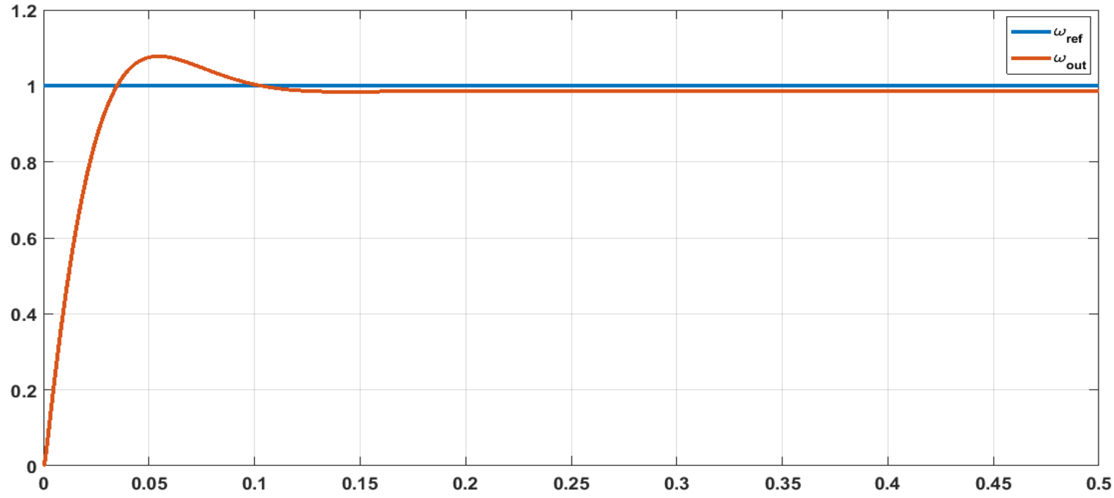

Figure 4.2: Step Response for the System using PD Controller.

Figure 4.3 shows the step response of PI controller. In PI controller the step response rise time is very low, minimum steady state error. But there is high overshoot (just over 10%).

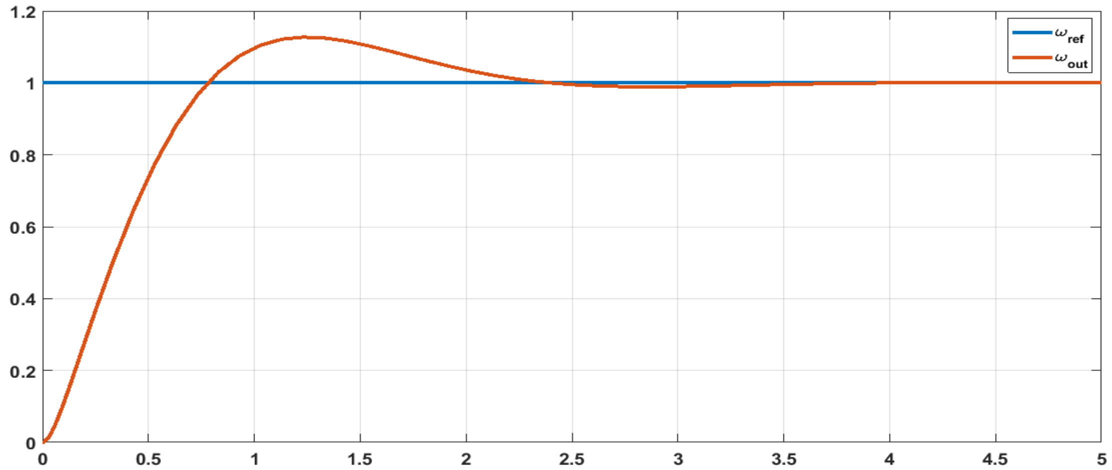

Figure 4.3: Step Response for the System using PI Controller.

4.4 Control System using PID controller

According to figure 4.4, in this system, PID was simulated. In this case the step response reaches the desired value, rise time is suitable and the steady state error is very low as show in table 4.1. But in PID controller overshoot occurred.

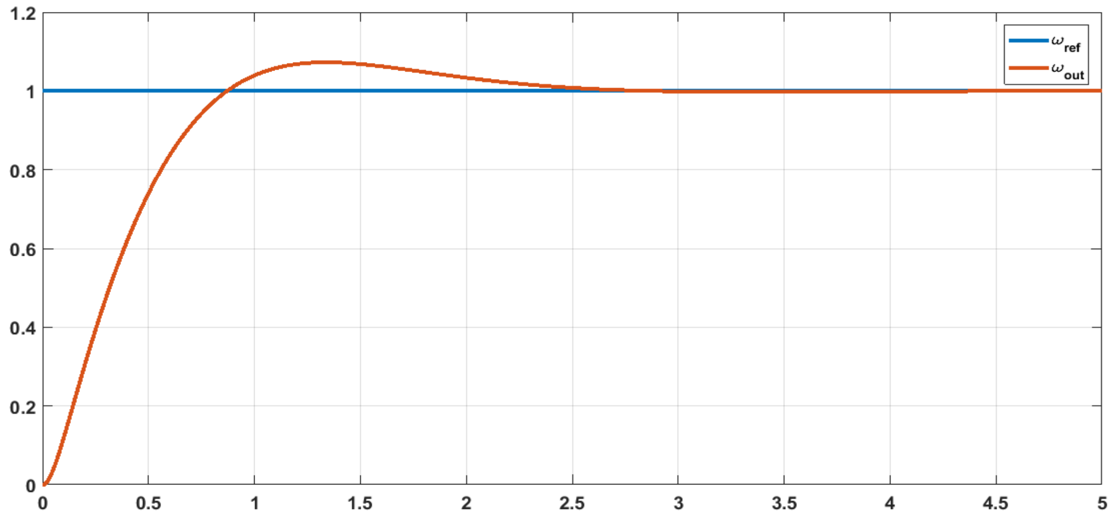

Figure 4.4: Step Response for the System using PID Controller.

In figure 4.5 shows the comparisons between PI, PD and PID controller. PD has minimum overshoot, minimum rise time, minimum settling time but there is steady state error (it did not reach the desired value). PI controller has overshoot, rise time and settling time more than PD controller, but the steady state error was minimized. In PID controller more rise time occurred, overshoot was decreased, settling time was increased but the system approximately has not steady state error. Table 4.1 shows the comparison between PD, PI and PID.

Table 4-1: Comparison Between the Step Response for each Controller (PI, PD, PID).

Parameter	PI	PD	PID
Overshoot (%)	12	8	9
Rise Time (sec)	0.8	0.033	0.9
Steady State Error	0	0.035	0
Settling Time (msec)	2.8	0.14	3.03

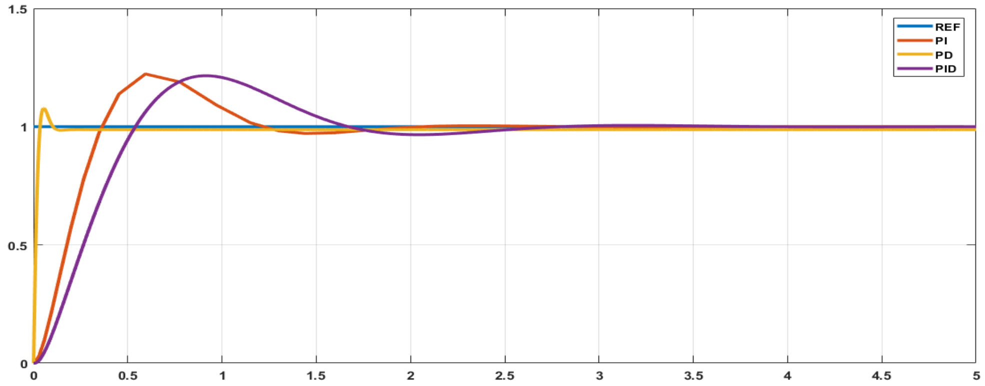

Figure 4.5: Comparison between PI, PD and PID.

4.5 Control System using Fuzzy PD and Fuzzy PI controller

Figure 4.6 shows the response of fuzzy PD controller. In fuzzy PD controller the step response rise time is increased, over shoot decreased according to PD controller. But there is steady state error.

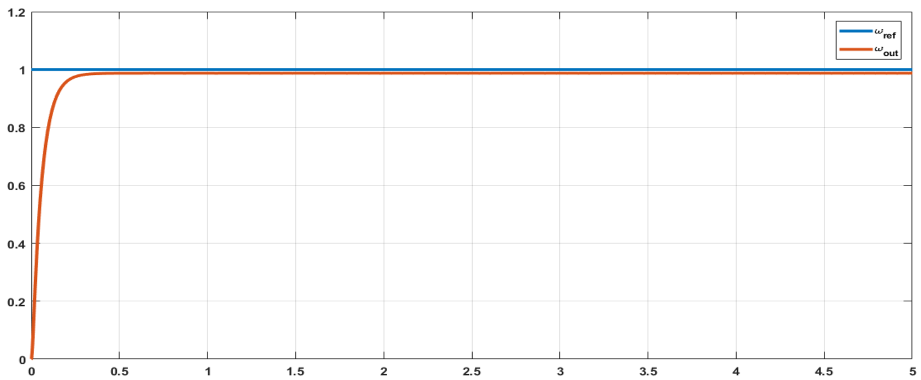

Figure 4.6: Step Response for the System using FPD Controller.

Figure 4.7 shows the step response of fuzzy PI controller. In fuzzy PI controller the step response rise time is very low, minimum steady state error. But there is high overshoot (over 40%).

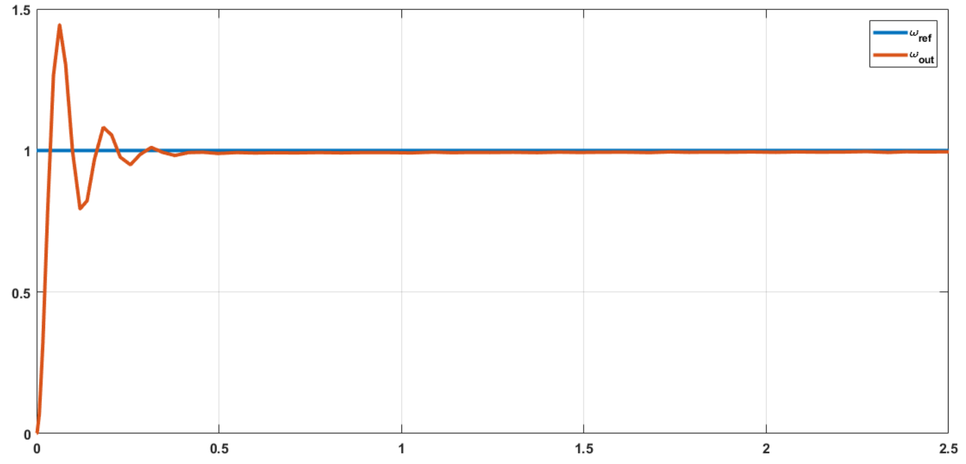

Figure 4.7: Step Response for the System using FPI Controller.

4.6 Control System using Fuzzy PID controller

According to figure 4.8, in this system, fuzzy used for PID parameters tuning. In this case the response reaches the desired value, no steady state error, minimize the overshoot and rise time. Table 4.2 shows the response parameters for each controller.

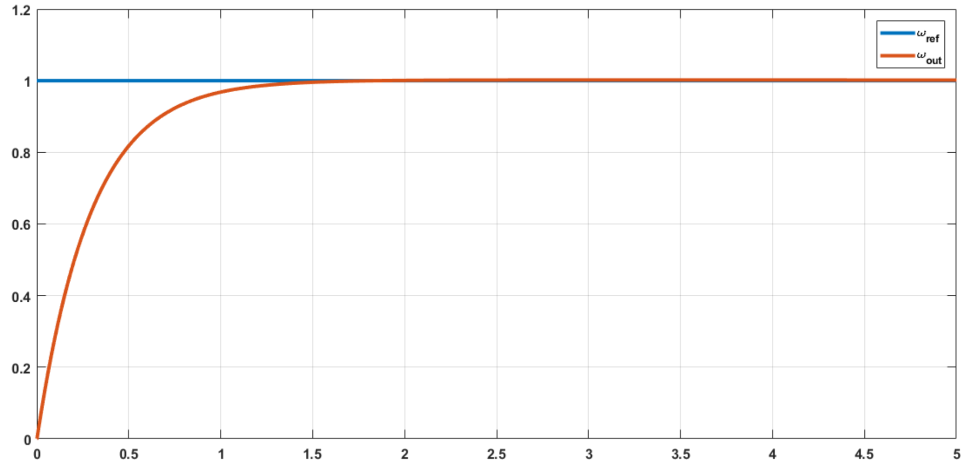

Figure 4.8: Step Response for the System using FPID Controller.

Figure 4.9 shows the error response (decrease through time) when FPID controller is used.

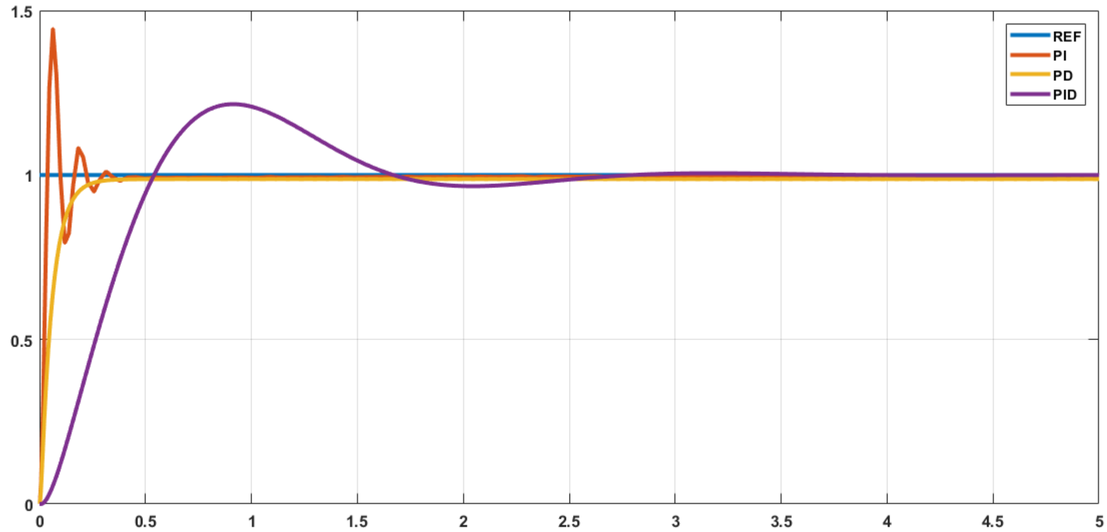

Figure 4.9: Comparison between FPI, FPD and FPID.

Table 4-2: Comparison Between the Output Responses for Controller (FPI, FPD and FPID).

Parameter	FPI	FPD	Fuzzy PID
Overshoot (%)	44.20	0	0
Rise Time (sec)	0.0625	0.22	0.575
Steady State Error	0.002	0.01	0
Settling Time (msec)	0.375	0.43	1.5

From the simulation results it is concluded that, compared with the conventional PID controller, Fuzzy controller and self-tuning PID controller. Fuzzy-PID controller has a better performance in both transient and steady state response. The self-tuning Fuzzy-PID has better dynamic response curve, short response time, smaller overshoot, less peak amplitude, minimum settling time, small steady state error, high steady precision compared to the conventional PID controller and Fuzzy controller.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In this project designed a DC motor whose speed can be controlled using mathematical model controllers (PD, PI and PID). The proportional, integral and derivate *(Kp, Ki, Kd)* gains of the mathematical model controllers are adjusted according to fuzzy logic. First, the fuzzy logic controllers were designed according to fuzzy rules so that the systems are fundamentally robust.

There are rules for self-tuning of each parameter of mathematical model controllers. The FLC has two inputs. One is the motor speed error between the reference and actual speed and the second is change in speed error (speed error derivative). Secondly, the output of the FLC i.e. the parameters of PID controller are used to control the speed of DC Motor.

Comparison between the mathematical model controllers and the fuzzy logic controller was done on the basis of the simulation result obtained by MATLAB. The simulation results demonstrate that the designed FL controller realize a good dynamic behavior of the DC motor, a perfect speed tracking with less rise time and minimum overshoot, minimum steady state error, with less settling time and give better performance compared to conventional mathematical model controllers.

5.2 Recommendations

1. This technique can be extended to other types of motors such as AC motor.

2. The parameters of PID controller can also be tuned by using genetic algorithm (GA).

3. Intelligent based fuzzy logic controller is better than fuzzy logic controller because of simulation of human intelligence.

REFERENCES

REFERENCES

[1]A. P. Singh, U. Narayan, And A. Verma, "Speed Control Of Dc Motor Using Pid Controller Based On Matlab " 2013.

[2]G. A. B. A. A. Sadiq, E. C. Anene, H. B. Mamman, "A Fuzzy-Based Speed Control Of Dc Motor Using Combined Armature Voltage And Field Current," 2013.

[3]M. R. K. A. A. K. U. Ghazali, "Speed Control Of Dc Motor Under Varying Load Using Pid Controller," 2015.

[4]A. Fatah, "Design And Analysis Of Speed Control Using Hybrid Pid-Fuzzy Controller For Induction Motors," 2015.

[5]N. I. B. P. Jabo, "Speed Control Of Dc Motor Using Pid Controller Implementation With Visual Basic," 2008.

[6]J. N. Rai, "Speed Control Of Dc Motor Using Fuzzy Logic Technique," 2012.

[7]T. Y. R. Krishna, "Speed Control Of Separately Excited Dc Motor Using Fuzzy Logic Controller," 2015.

[8] C. Copot, C.I. Muresan, R. De Keyser, "Speed and Position Control of a Dc Motor Using Fractional Order Pi-Pd Control ", 2013.

[9] S. B. Saat, "Dc Motor Speed Control Using Fuzzy Logic Controller," 2014.