“Design and Simulation of DC Motor Drive Controller by using PID and Fuzzy logic”

A MASTER’S THESIS

in

Electrical and Electronic Engineering

Atılım University

BY

ZAID MUSAAB GHANIM

JANUARY 2015

“Design and Simulation of DC Motor Drive Controller by using PID and Fuzzy logic”

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

ATILIM UNIVERSITY

BY

ZAID MUSAAB GHANIM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF ELECTRICAL ENGINEERING

JANUARY 2015

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

(Title and Name)

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

(Title and Name)

Head of Department

This is to certify that we have read the thesis “Thesis Name” submitted by “Candidates Name” and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

(Title and Name)	(Title and Name)
Co-Supervisor	Supervisor

Examining Committee Members

....

....

....

....

....

Date: (Thesis defence date)

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name:Zaid Musaab Ghanim

Signature:

ABSTRACT

“Design and Simulation of DC Motor Drive Controller by using PID and Fuzzy logic”

GHANIM, Zaid Musaab

M.S., Electrical Engineering Department

Supervisor: Asst.Prof.Dr. Alparslan Çağrı Yapıcı

January 2015, 44pages

This thesis presents PID controller implemented by fuzzy logic and FPGA. Controller model is carried out by using MATLAB-SIMULINK. FPGA is emulated for the generated HDL code. Simulation and emulation results are compared between Fuzzy logic and PID implementation on FPGA. The results indicates improvement in step response characteristics such as reducing the steady states error, rise time, settling time and maximum overshoot in control lf a DC motor.

Keywords: Fyzzy Logic, FPGA, PID, Dc motor.

ÖZ

“PID ve Bulanık mantık kullanarak DC Motor Sürücü Devresinin Tasarım ve Benzetimi”

GHANIM, Zaid Musaab
Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Yar. Doç.Dr. Alparslan ÇağrıYapıcı

Ocak 2015, 44sayfa

Bu tez çalışması bulanık mantık ve FPGA Kullanarak gerçekleştirilen PID Kontrol devresini sunmaktadır.Kontrol devresi modeli MATLAB SIMULINK kullanılarak gerçekleştirlmistir.FPGA üretilen HDL kodu için emüle edilmiştir. Simülasyon ve emülasyon sonuçları FPGA üzerinde gerçeleştirilen PID ve bulanık mantık ile karşılaştılmıştır .Sonuçlar durgun durum hatası yükseleme zamanı yatışma süresi ve maksimum aşma miktarı gibi DC motor basamak cevabı kavramlarındaki iyileşmeyi göstermektedir.

AnahtarKelimeler: Fyzzy Mantık, FPGA, PID, Dc motoru.

To My Parents

ACKNOWLEDGMENTS

I express sincere appreciation to my supervisor Asst.Prof.Dr. Alparslan Çağrı Yapıcı for his guidance and insight throughout the research. To my parents, I offer sincere thanks for her continuous support and patience during this period.

TABLE OF CONTENTS

CHAPTER ONE

CHAPTER TWO

CHAPTER THREE

CHAPTER FOUR

CHAPTER FIVE

LIST OF FIGURES

ABBREVIATIONS

PWM: Pulse Wide Modulation

LED: Light-Emitting Diode

AMP: Amplificatory

IC: Integrated Circuit

PCB: Printed Circuit Board

GND: Ground

PWR: Power

FWD: Forward

REV: Reverse

NB: Negative Big

NS: Negative Small

ZE: Zero

PS: Positive Small

PB: Positive Big

FIS: Fuzzy Inference System

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

DC motors are nonlinear systems; they have convened a very important role within the improvement of the economic revolution, working as a heart in various applications. DC motors are a major part in automatic systems; every automatic system must contain a mechanism module that creates the system to really perform its operation.

Nowadays, the speed control of a DC Motor is accomplished by terminal voltage control. A conventional DC motor can operate in four different quadrants by changing the polarity of voltage and direction of current. These four modes are: forward motoring (positive voltage and current), forward regeneration (positive voltage, negative current), reverse motoring (negative voltage and current) and reverse regeneration (negative voltage and positive current). The term regeneration (also known as regenerative braking) means operating the motor as a generator. This breaks the motor by converting its mechanical energy into electrical energy and sending it back to the batteries.

Most of the industrial controllers are PID in nature. The major reasons behind the popularity of PID controllers are their simplicity in structure and their applicability to a variety of processes. Moreover the controller can be tuned for a process, even without a detailed mathematical model of the process [1]. Fuzzy logic provides a certain level of artificial intelligence to the conventional controllers, leading to the effective Fuzzy controllers. Process loops that can benefit from a non-linear control response are excellent candidates for Fuzzy control, since Fuzzy logic provides a fast response times with virtually no overshoot [2].

In this thesis we use Fuzzy to control on speed of the DC motor. Fuzzy logic can produce a system to match a set of input & output data. Fuzzy logic will model nonlinear functions of impulsive complexness. Fuzzy logic is made particularly simple by adjustive techniques like Adaptive Neuro-Fuzzy logical thinking Systems (ANFIS) that are available in a toolbox.

In addition, the PID system controllers are implemented by using SIMULINK and MODELSIMSE-64 10.1C software in digital domain in order to be used in field programmable gate array (FPGA). This design has low complexity in controlling.

In the previous study [3], a tachogenerator is used as a speed sensor which generates a back EMF corresponding to the speed attained by a DC Motor. This immediate value of output voltage provided by the tachogenerator is then compared between the desired voltage corresponding and the desired speed. The result of was used by the microcontroller to control the firing angle of the silicon-controlled rectifier(SCR) for controlling the voltage applied to the DC Motor which in turn adjusts directly the motor speed to attain the desired value. So a continuous closed loop speed control system has been achieved. Proportional (P) Control Algorithm has been used in that paper. Another study [4], there was a refer to improving adaptive Fuzzy logic speed controller for a DC motor, based on FPGA hardware implementation. The developed controller includes an adaptive Fuzzy logic control (AFLC) algorithm, which is designed and verified with a nonlinear model of the DC motor. Then, it would be synthesized, functionally verified and implemented using Xilinx Integrated Software Environment (ISE) and Spartan-3E FPGA. The performance of this controller has been achieved with success valid with smart trailing results beneath completely different in operation conditions. Also another study [5], the PWM speed regulation of a DC motor based on intelligent control is discussed. The simulation is done with the SIMULINK after that the mathematical model of the controlled object is built. This article introduces the PWM bipolar drive of the DC motor, designs a Fuzzy controller and a neutral network controller and then discusses the application of artificial intelligence in the speed regulation of DC motor. Another study [6] the interested with the development of improved cascaded speed control systems for Thyristor driven DC-motors. The best options of the work may be summarized within the following four points: (1) Development of a digital dual-mode adaptive controller for the inner current control loop and of a model reference adaptive controller for the outer speed control loop, thus making the entire system adaptive. (2) Development of robust controllers both for the inner current loop and also for the outer speed loop, thus making the entire system robust. (3) Implementation of the above control strategies, adaptive and robust, in a 16-bit single board computer with floating-point coprocessor. (4) Comparison of the results of both robust and adaptive improved cascaded schemes with a commercially available controller. The obtained results showed that the model reference adaptive control concept and the robust control strategy can be applied with success for the speed control of a DC motor. Another study [7] the proposal was to a design and built a control system for position control for DC motors. This is a closed loop control system. To control the speed of a DC motor, pulse width modulation (PWM) technique was used. Very high speed integrated circuit hardware description language (VHDL) was used to program the software module, and implemented in field-programmable gate array FPGAs. And in the last study [8], Proposed an approach to subdivide the step to micro steps with required resolution for stepper motor to improve positioning accuracy, in order to achieve both high speed and high precision for motion control. In addition, the designs of the hardware and software are also proposed in detail to control the currents of the motor precisely. A controller with flexible algorithm and programmable logic based on Field Programmable Gate Array (FPGA) was introduced.

This research will study DC motor control because the DC motors are a major part in automatic systems; every automatic system must contain a mechanism module that creates the system to really perform its operation.

This research will show some methods that can be used in a DC motor control. The designing will show PID controllers, Fuzzy logic control and PID implementation for FPGA using SIMULINK MATLAB and MODELSIM.

CHAPTER TWO

PID CONTROLLER AND FUZZY LOGIC FOR DC MOTORS

The major reasons behind the popularity of PID controllers are their simplicity in structure and their applicability to a variety of processes. Moreover the controller can be tuned for a process, even without detailed mathematical model of the process [1]. Fuzzy logic provides a certain level of artificial intelligence to the conventional controllers, leading to the improving the effectiveness of Fuzzy controllers. Process loops that can benefit from a non-linear control response are excellent candidates for Fuzzy control. Since Fuzzy logic provides a fast response times with virtually no overshoot [2].

2.2 Modeling of DC motor

DC motors convert the electrical power to the mechanical power. The DC motor's electrical circuit is defined in the Figure 2.1 [9].

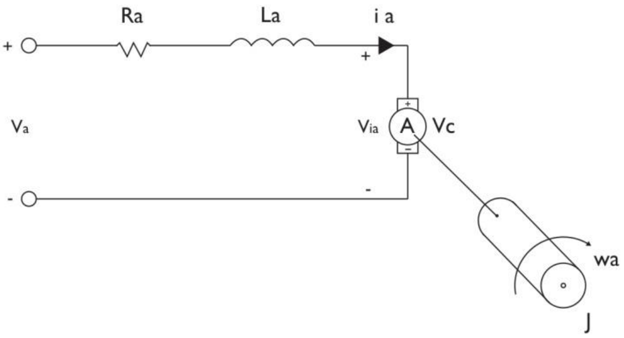


Figure 2.1 Electrical Circuit of a DC motor

Here V_a is the voltage source of DC Motor, J is the moment of inertia, $R_a$is the resistance of armature, $L_a$is the inductance of armature ω_a is the speed of DC motor.

Electrical characteristics of the circuit shown in Figure 2.1 can be expressed by Kirchhoff's law which when applied to an the electrical loop can derive an differential equation for the circuit

$$V_a - V_{ra} - V_{ia} - V_c = 0 \tag{2.1}$$

Voltage is obtained from Ohm's law across the armature resistor as $V_{ra} = i_a R_a$

The inductance voltage is symmetrical to the variation of current passing through the coil regard to time, and may be written as:

$$V_{La} = L_a \frac{di_a}{d_t} \tag{2.2}$$

Electromotive force (EMF) can be written as

$$E_a = K_e \omega_a \tag{2.3}$$

Where Ke is back EMF constant

Substituting all (2.1), (2.2) and (2.3) in (2.4) the result is

$$V_a - i_a R_a - L_a \frac{di_a}{dt} - K_e \omega_a = 0 \tag{2.4}$$

Mechanical Characteristics; can also describe on the system, performing an energy balance has that score of torques whom must be equal to zero, therefor

$$T_e - T_\omega - T_{\omega\prime} - T_1 = 0 \tag{2.5}$$

Where T_ω is torque due to friction, $T_{\omega\prime}$ is torque due to inertia, T_1 is torque load and T_e is electromagnetic torque.

From the above equations the current during the armature in the electromagnetic torque can be obtained as follows:

$$T_e = K_t i_a \tag{2.6}$$

$$T_{w\prime} = J \frac{d\omega_a}{dt} \tag{2.7}$$

$$T_\omega = B.\omega \tag{2.8}$$

Substituting equations (2.6), (2.7) and (2.8) gives the following differential equation

$$K_t i_a - J \frac{dw_a}{dt} - B\omega_a - T_l = 0 \tag{2.9}$$

State space representation of the equations (2.5) and (2.9) can be written as

$$\frac{dia}{dt} = -\frac{Ra}{La}\, i_a - \frac{Ke}{La}\omega_a + \frac{Va}{La} \tag{2.10}$$

$$\frac{d\omega_a}{dt} = \frac{k}{J} I_a - \frac{B}{J}\omega_a - T_l \tag{2.11}$$

By representing equations in the matrix form, the result is

$$\frac{d}{dt}\begin{bmatrix} I_a \\ \omega_a \end{bmatrix} = \begin{bmatrix} \frac{-R_a}{L_a} & -\frac{K_e}{L_a} \\ \frac{K_t}{J} & -\frac{B}{J} \end{bmatrix} \begin{bmatrix} I_a \\ \omega_a \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix} \begin{bmatrix} V_a \\ T_l \end{bmatrix} \tag{2.12}$$

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} ia \\ \omega a \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_a \\ T_1 \end{bmatrix} \tag{2.13}$$

By applying Laplace transforms to equations (2.11) and (2.12), the resulting mathematical form is:

$$I_a(s) = \frac{V_a(s) - E_a(s)}{R_a + sL_a} \qquad E_a(s) = K_e.\omega_a(s) \tag{2.14}$$

$$\omega_a(s) = \frac{T_e(s) - T_1(s)}{sJ + B} \qquad T_e(s) = L_a(s) \tag{2.15}$$

By using the equation 2.14, 2.15 which results from DC motor equation, the block diagram in Figure 2.2 can be obtained

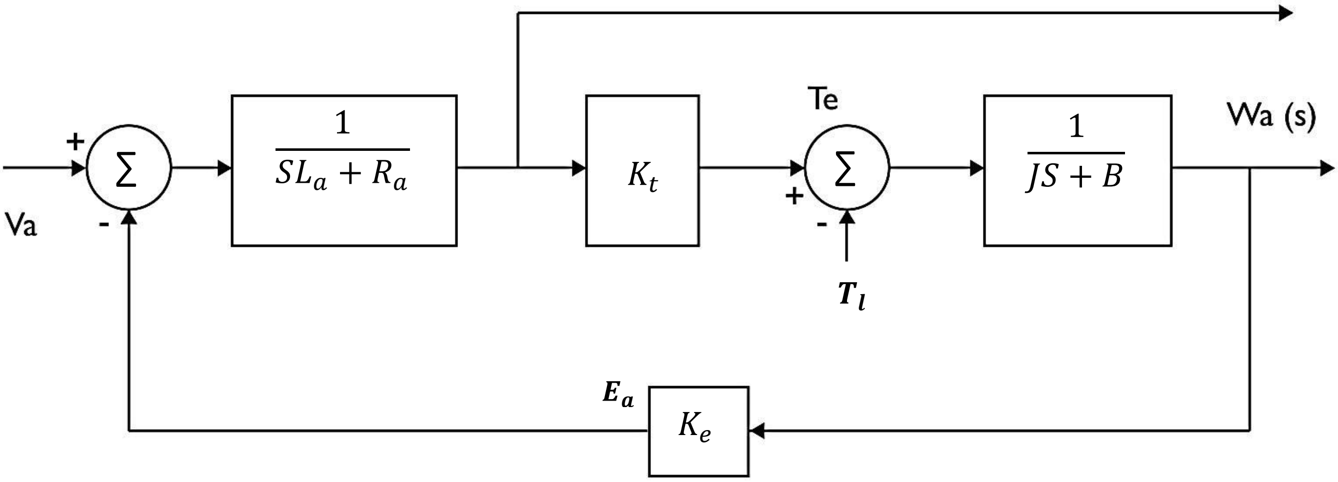


Figure 2.2 Block diagram of DC motor

2.3 PID Controller Design

These days, PID controllers are widely used in industry because of their simplicity in structure and their applicability to a variety of processes. The input to PID is error signal which can be described by the difference between reference signal and the output signal. The output of PID controllers is the sum of three terms: proportional term, integral term and the derivative term [10]. The transfer function of PID is represented as:

$$u(t) = K_p e(t) + K_i \int e(t)dt + k\frac{de(t)}{dt} \tag{2.16}$$

where Kp is the Proportional gain, Ki is the Integral gain, Kd is the Derivative gain, e(t) is the error signal. Figure 2.3 shows a standard PID controller block diagram [10],

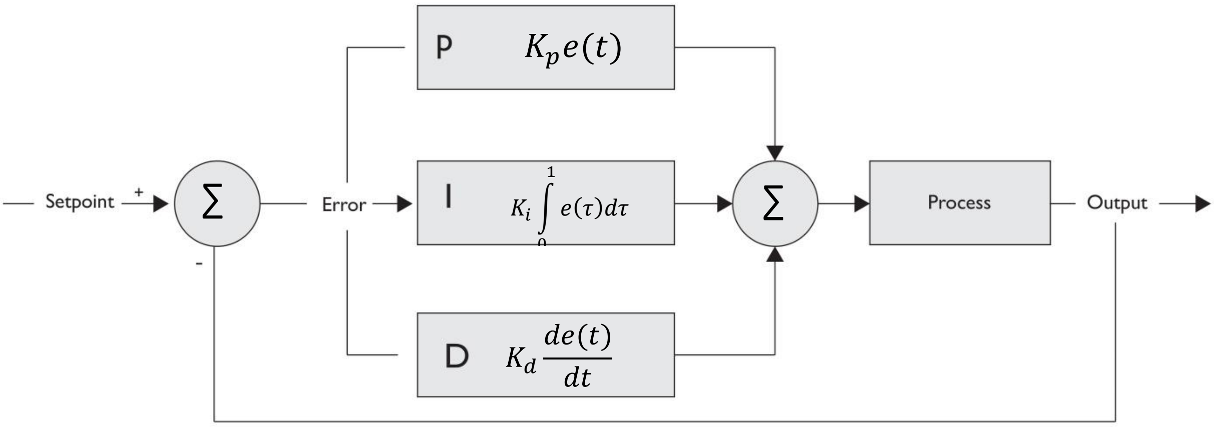


Figure 2.3 PID controller

In order to minimize rise time, proportional controller Kp is used. Introducing an integral controller can eliminate steady-state error, which is responsible for increasing the overshoot. Introducing a derivative controller increases the stability of the system, reduces the overshoot and improves the transient response. Effects of each controller and the characteristics of parameters are summarized in the Table 1[10].

Table 1 Characteristics of parameters PID controllers

Parameter	Rise time	Overshoot	Settling time	Steady-state error
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Increase
Kd	Small Change	Decrease	Decrease	Small Change

A proportional controller reduces error but does not eliminate it (unless the process has naturally integrating properties), i.e. an offset between the actual and desired value will normally exist [11].

2.4. FUZZY LOGIC:

In recent years, with the rapid development of technology. Fuzzy logic has one of the most present technologies for improving sophisticated control systems. Since it is an intelligent controller. Mathematical logic or (Fuzzy logic) addresses such applications utterly because it resembles human deciding with an ability to get precise solutions from bound. It fills an important gap in engineering style strategies left vacant by strictly mathematical approaches (e.g. linear management, design), and strictly logic-based approaches (e.g. skilled systems) in system style. Whereas alternative approaches need correct equations to model real-world behaviors. Thereby, Fuzzy style will accommodate the ambiguities of real-world human language and logic, it provides every Associate in Nursing intuitive methodology for describing systems in human terms and automates the conversion of those system specifications into effective models.

The Rules contribute to inferences even when facts do not exactly match antecedent and Rule-based systems may be analyzed and improved. The Technology is easy to convert from product to product. The ability to control unstable systems [12].

2.4.1 Fuzzy sets

A collection of definitions, can be described is distinguishable objects of our intuition which may be treated as an entire. The objects are the members of the idea give us a great freedom of choice. Objects must be ’definite’: given an object and a set, it must be possible to determine whether the object is, or is not, a member of the set. Objects must also be ’distinguishable’: given a set and its members, it must be possible to determine whether any two members are different, or the same. The members fully outline a group to see membership. It is necessary that the sentence x may be a member of X, wherever x is replaced by associate object and X by the name of a group, is either true or false, to use the symbol ϵ and write $x \epsilon X$ if object x may be a member of the set X. The assumption that the numbers confirm a group is like saying: Two sets X and Y square measure equal,$X = Y$, if (if and solely if) they need an equivalent members. The set whose members square measure the objects,$x1$,$x2$ . . . ,xn is written [12].

2.4.4 Operations on Fuzzy sets

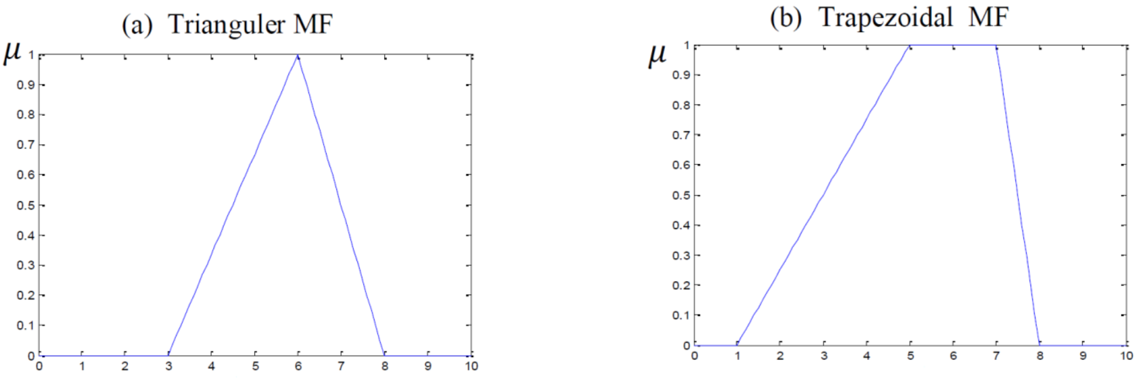
We extend the classical set hypothetical operations from normal pure mathematics to Fuzzy sets. We have a tendency to note that everyone those operations that square measure extensions of crisp ideas scale back to their usual, which means once the Fuzzy subsets have membership degrees that square measure drawn from(0 ,1). Therefore, once extending operations to Fuzzy sets we have a tendency to use an identical image as in pure mathematics [13].

2.4.8 Membership Functions

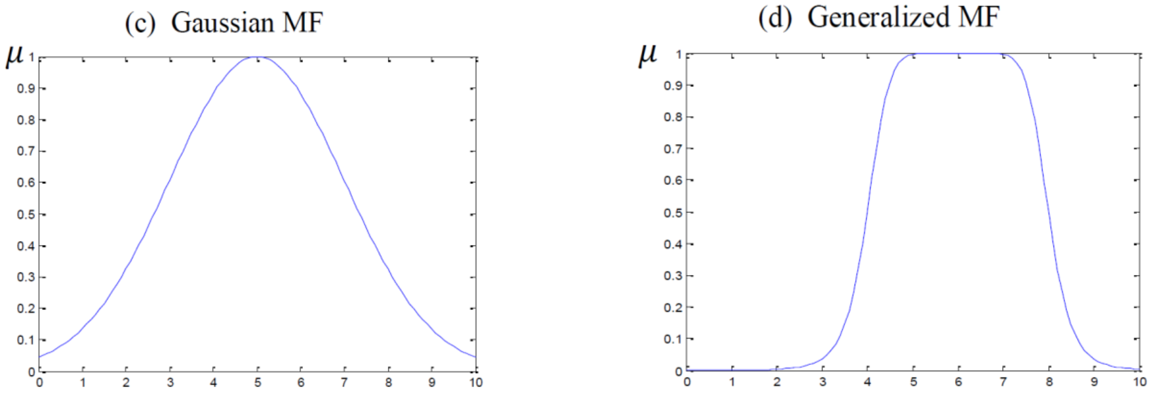
Usually one of the following membership functions can be considered:

- Triangular:tri(x;a,b,c)= $\max\{\min\left\{\frac{x-a}{b-a}\ ,\frac{c-x}{c-b}\right\}, 0\}$ (2.21)
- Trapezoidal:trap(x; a,b ,c ,d)=$\max\left\{\min\left\{\frac{x-a}{b-a}\ ,\frac{d-x}{d-b}, 1\right\}, 0\right\}$ (2.21)
- Gaussian:gauss(x;c,σ)=$\exp[-\frac{1}{2}(\frac{x-c}{\sigma})^2]$ (2.23)
- GeneralisedBell:gbell(x;a ,b)=$\frac{1}{1+\left|\frac{x-b}{a}\right|^{2a}}$ (2.24)

In the Figure 2.4 show the type and the different between the Membership Function


(a) Trianguler MF

(b) Trapezoidal MF


(c) Gaussian MF

(d) Generalized MF

Figure 2.4 Shape of different Membership Function

2.4.9 Fuzzy Systems

A Fuzzy System will contrasts with a traditional (crisp) system in 3 main ways. A Linguistic Variable is defined as a variable whose values are sentences in a natural state. If tall, not tall, very tall, etc. are the values of height, which is a linguistic variable. A Fuzzy Conditional Statements are expressions of the form If A THEN B, where A and B have a Fuzzy meaning, e.g. If x is small THEN y is large, where small and large are viewed as labels of Fuzzy sets. A Fuzzy Algorithm is an ordered sequence of instructions which may contain a Fuzzy assignment and a conditional state

2.4.9 Fuzzy Logic Control System

Fuzzy logic control is considered to be the best used in complex, all-in addition, Fuzzy logic deals with several processes that shall be controlled by the human operator without much knowledge of their underlying dynamics. The basic plan behind FLC is to include the "expert experience" of an operator within the style of the controls in dominant method whose input – output relationship is represented by an assortment of Fuzzy control rules (e.g., IF-THEN rules) involving linguistic variables instead of an advanced dynamic model. The utilization of linguistic variables, will introduce the essential design and functions of a symbolic logic controller.

A typical design of FLC is shown below, that includes a 4 principals comprises: a falsifier, a Fuzzy rule base, logical thinking engine, and a defuzzifier. Figure 2.5 shows the structure of the Fuzzy logic controller.

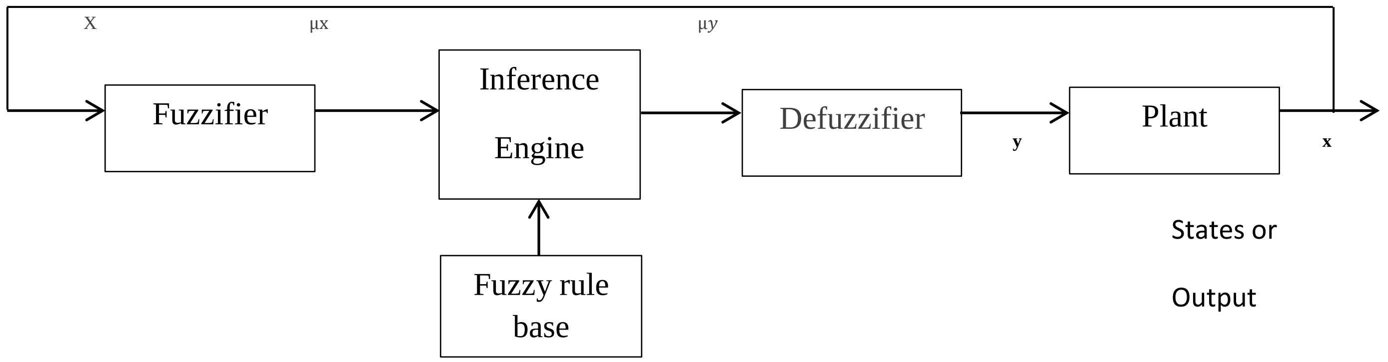


Figure 2.5 Structure of Fuzzy logic controllers

The system will make a Fuzzy logic decision if the output from the defuzzifier isn't a bearing action for a plant. The fuzzier decides how the crisp input will be converted into a fuzzy input to be used by the inference engine. Fuzzy inference is the actual process of mapping from a given input to an output using fuzzy logic. The defuzzifier decides how to convert the fuzzy result from the inference engine back into a crisp value. The control action should be within the kind of a crisp value. Defuzzification is that the method of translating the Fuzzy set allotted to an effect output variable into such a crisp value [14].

There are numerous strategies for defuzzification. The subsequent three are the foremost outstanding in Fuzzy control. There are three types, Center of Gravity (COG), Bisector of Area (BOA) and Mean of Maximum (MOM)

a) Center of Gravity (COG)

For separate sets COG is termed center of gravity for singletons wherever the brittle control value is that the Cartesian coordinate of the middle of gravity of the Fuzzy set, which is calculated as follows:

$$Ucogs = \frac{\sum \mu c(Xi)Xi}{\sum \mu c(Xi)} \qquad (2.24)$$

Where Xi is point in the universe of conclusion (i=1,2,3,…) and µc(Xi) is the membership value of the resulting conclusion set. For continuous sets summations are changed by integrals.

b) Bisector of Area (BOA)

The Bisector of Area defuzzification methodology calculates the Cartesian coordinate of the vertical line that divides the world of the ensuing membership perform into 2 equal areas. For distinct sets, $Uboa$ is the abscissa Xj that minimizes

$$\sum_{i=1}^{J} \mu c(Xi) - \sum_{i=j+1}^{imax} \mu c(Xi)\,, i < j < i\,max \tag{2. 25}$$

Here is $i\,max$ the index of the largest abscissa $Ximax$. BOA is a computationally complex method.

c) Mean of Maximum (MOM)

In this technique the crisp value is to settle on the purpose with the best membership. There could also be many points within the overall implicit Fuzzy set that have most membership value. so it is common to observe and calculate the average of these points. This technique is termed Mean of Most (MOM) and therefore the crisp value is calculated as follows:

$$Umom = \frac{\sum_{i\in I} Xi}{|I|}, I = \{\frac{i}{\mu c(xi)} = \mu max\} \tag{2.26}$$

Here I is the (crisp) set of indices i where $\mu c(Xi)$ reaches its maximum $\mu\,max$, and $|I|$is its cardinality (the number of members) [15].

CHAPTER THREE

SIMULINK MODEL OF DC MOTOR SYSTEM

Modeling, analysis and design of controllers are done in MATLAB/SIMULINK. Simulation of mathematical model of a physical system is done using SIMULINK. It helps in solving a problem in a graphical form by interconnecting various blocks and figuring the solution to differential equations through the use of graphs and plots. SIMULINK imitates a digital computer emulation of an analog computer [16].

3.1 DC Motor Modeling

The rotor and the shaft are assumed to be rigid. The motor torque, T, is related to the armature current i, by a constant factor K:

$$T = K_i . i \tag{3.1}$$

The back electromotive force (emf), V_b, is related to the angular velocity by:

$$E_a = k\omega_a = k\frac{d\theta}{dt} \tag{3.2}$$

From Figure 2.1, the following equations based on the Newton's Law combined with the Kirchhoff's Law can be written as:

$$L\frac{di}{dt} = -K_e\omega_a - Ri_a + V_a \tag{3.3}$$

$$J\omega(t) = K_i \tag{3.4}$$

$$V = R_i + L_i(t) + K_\omega \tag{3.5}$$

using Laplace transformation, equation (3.5) can be written as:

$$\frac{1}{s+1} \times \frac{1}{s^2+s+1} \tag{3.6}$$

From the above equations, the modeling DC motor can be built and can be represented in Figure 3.2. The numerator coefficient and the denominator coefficient can be represented in equation (3.7) the input to DC motor represent the output voltage from PID and the output of DC motor is represent velocity(ω).

$$\frac{1}{s+1} \times \frac{1}{s^2+5s+6} \tag{3.7}$$

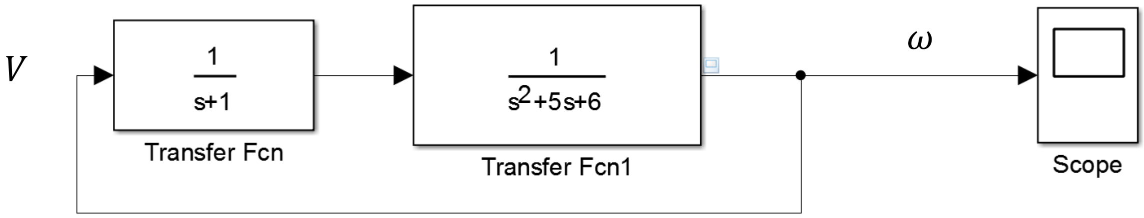


Figure 3.2 DC motor modeling

3.2. **PID Controller Modeling System**

Figure 3.3 show below the overall simulation circuit for the PID controller. The PID controller will control input to the Pulse Width Modulation (PWM) terminal in the back converter. At several set points, output from the PID controller will generate different values. Building on that, average voltage supply to the voltage terminal of the DC motor and speed of the DC motor can be varied.

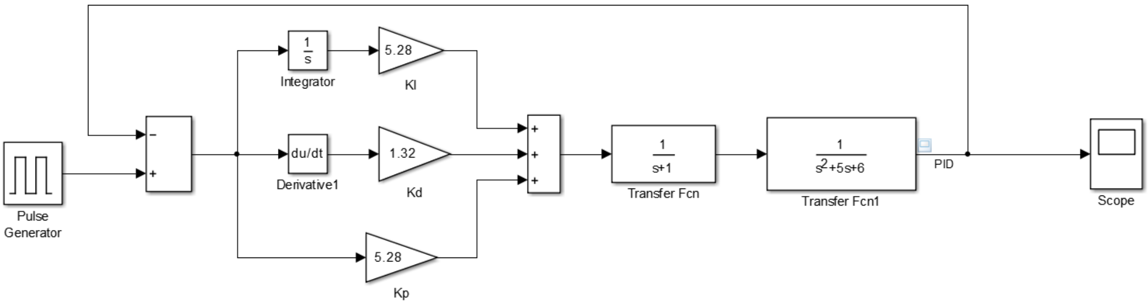


Figure 3.3 PID controller modeling system

3.2.1 PID Controller Design in SIMULINK

To obtain the optimum gain value for Kp, Ki and Kd, several gain value have been tested. The best combinations of gain value are chosen with respect to minimum overshoot, minimum steady state error and minimum steady state time. Figure 3.4 shows the PID controller design represented in this thesis.

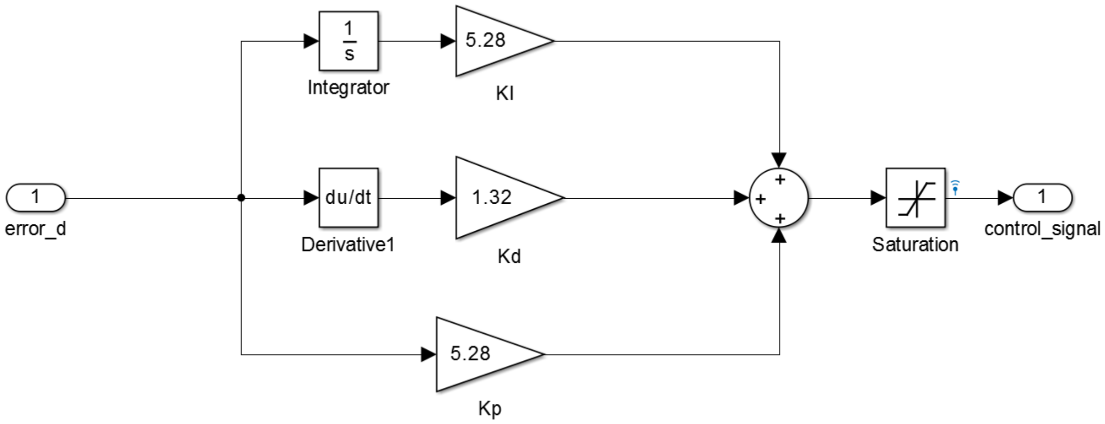


Figure 3.4 PID controller design in SIMULINK

PID parameters has used in this SIMULINK was Kp is 5.28, Ki is 5.28 and Kd is 1.32.

Figure 3.5 shows the simulation results of using speed PID controllers with the DC motor and explains the effect of using the PID for controlling the DC motor. Y axis represented voltage and X axis represented time.

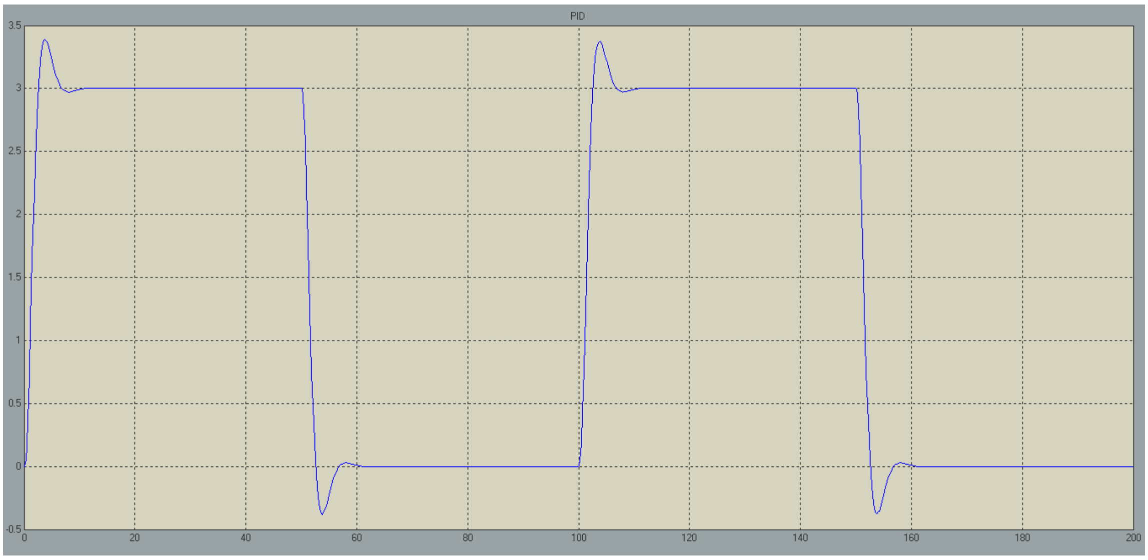


Figure 3.5 Simulation results of using speed PID

Figure 3.5 shows that the SIMULINK of PID has an overshoot, the value of the overshoot is start from 3 to 3.4 and the maximum overshoot is 0.4v and the duration is near to 5ms from the time of signal.

3.3 Fuzzy Logic Controller Modeling System

The speed control system under consideration is shown in Figure 3.6. The speed control loop has an inner current loop to provide fast transient response as well as to limit the armature current. The speed controller is designed in such a way to produce a desired reference signal for the current controller. The output of the current controller is fed to a gain or a pulse width modulator which controls the terminal voltage. Figure 3.6 shows Fuzzy implementation that are designed in this thesis with two inputs and three outputs, the outputs represent Kp, Ki, Kd. The DC motor designed and connected with the outputs from the Fuzzy logic controller.

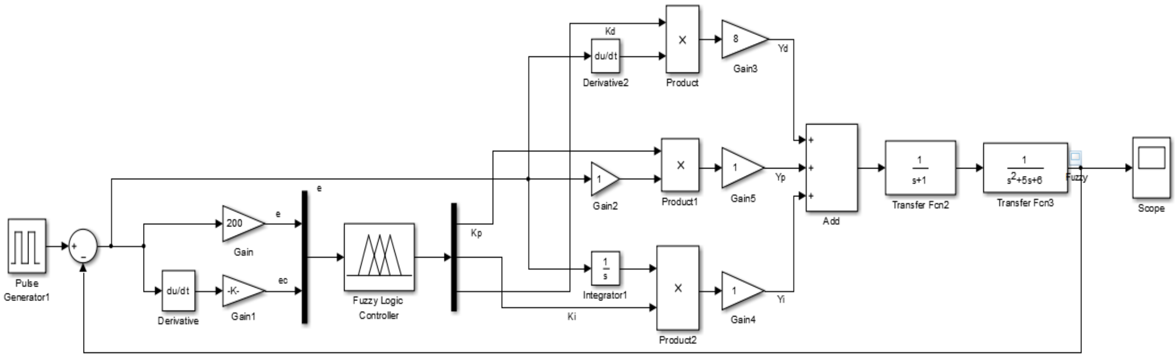


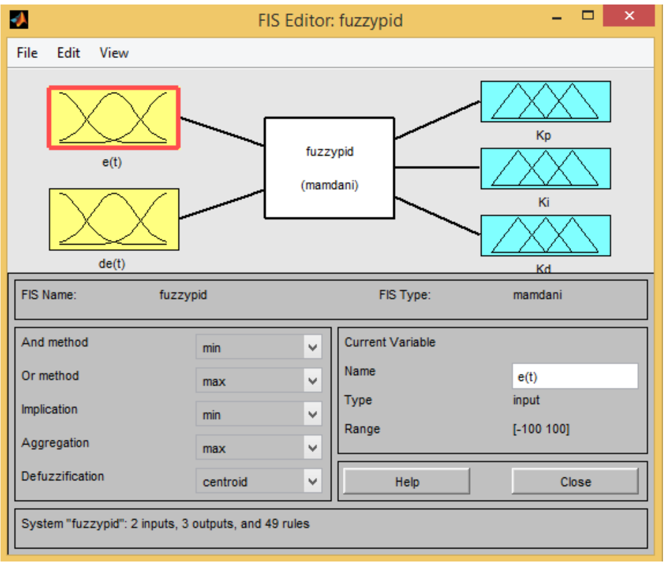
Figure 3.6 SIMULINK Model for Control of the DC motor using the Fuzzy PID controller

The value of Fuzzy parameters obtained in this SIMULINK schematic is Kp is 1, KI is 1, and Kd is 8

Figure 3.7 shows that the Fuzzy inference system got the parameters from the SIMULINK model for control of the DC motor using Fuzzy PID controller.

FLC controller design and the performance of the controller has been analyzed and evaluated in term of percentage of maximum overshoot (OS%), rising time, peak time, settling time and steady state error. In this thesis Fuzzy Logic Controller (7×7) matrix was used. And the design of the FLC has been tested.
FLC toolbox in SIMULINK has been used. Below are general setting parameter and method that had been used for the FLC controller Fuzzy inputs are composed of seven

membership functions of error and seven membership functions of error derivative. Fuzzy outputs have been control output (7 membership). Fuzzy Inference system was Mamdani and the number of Rules base was 49. Figure 3.8 show the Fuzzy logic controller:



FIS Name: fuzzypid
FIS Type: mamdani

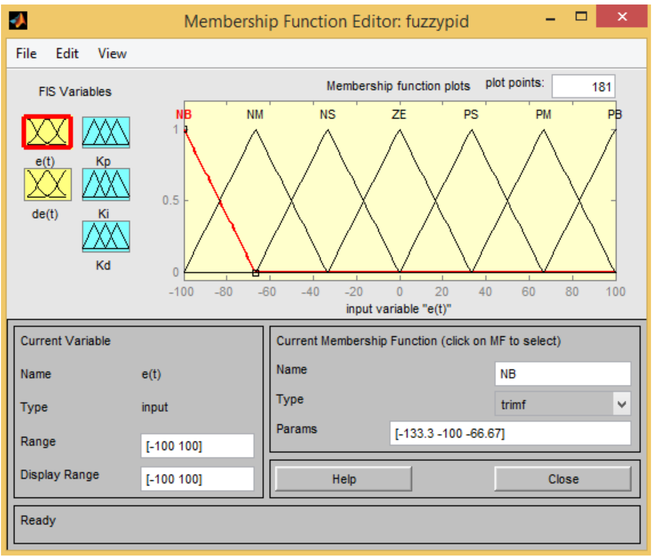
And method: min
Or method: max
Implication: min
Aggregation: max
Defuzzification: centroid

Current Variable
Name: e(t)
Type: input
Range: [-100 100]

System "fuzzypid": 2 inputs, 3 outputs, and 49 rules

Figure 3.7 FLC controllers FIS editor

The membership functions for the error are shown in Figures 3.8 and the membership functions for error derivative are shown in Figure 3.9. These functions are assumed to be same for both the input variables.

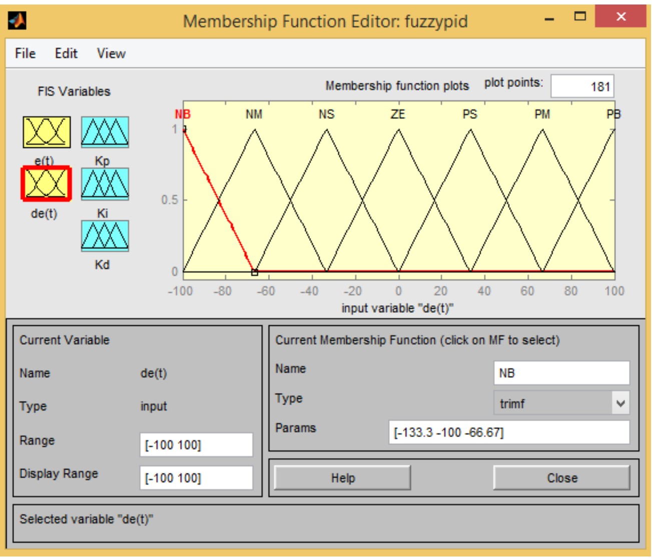


Current Variable
Name: e(t)
Type: input
Range: [-100 100]
Display Range: [-100 100]

Current Membership Function (click on MF to select)
Name: NB
Type: trimf
Params: [-133.3 -100 -66.67]

Ready

Figure 3.8 Membership functions of error.



plot points: 181

Current Variable

Name de(t)

Type input

Range [-100 100]

Display Range [-100 100]

Current Membership Function (click on MF to select)

Name NB

Type trimf

Params [-133.3 -100 -66.67]

Selected variable "de(t)"

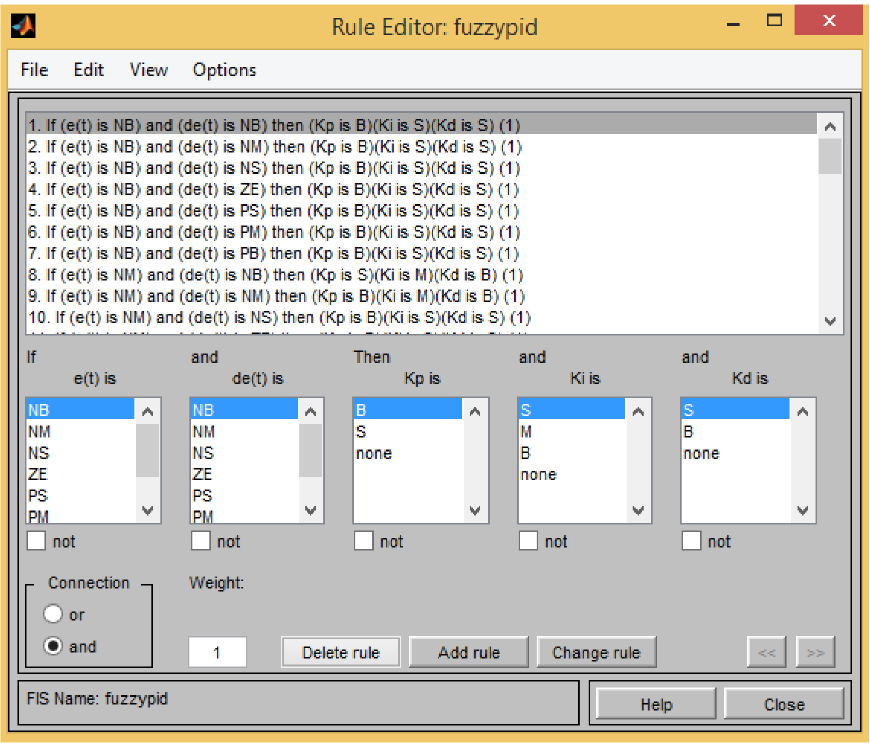
Figure 3.9 Membership functions error derivative.

Refering to Figure 3.9 above, Table 2 shows detail of setting parameter for the input error membership function. Range of error: -100 to 100

Error	Value
NB	[-133.3 -100 -66.67]
NM	[-100 -66.67 -33.33]
NS	[-66.67 -33.33 7.105e-15]
Z	[-33.33 4.441e-16 33.33]
PS	[7.105e-15 33.33 66.67]
PM	[33.33 66.67 100]
PL	[66.67 100 133.3]

Table 2 Detail of setting parameter for input error membership function.

The Fuzzy logic controller has two input variables error and rate of change in error. And three output variables Kp, Ki and Kd. There are 49 rules are defined using the linguistic variables, shown in Figure 3.10.



1. If (e(t) is NB) and (de(t) is NB) then (Kp is B)(Ki is S)(Kd is S) (1)
2. If (e(t) is NB) and (de(t) is NM) then (Kp is B)(Ki is S)(Kd is S) (1)
3. If (e(t) is NB) and (de(t) is NS) then (Kp is B)(Ki is S)(Kd is S) (1)
4. If (e(t) is NB) and (de(t) is ZE) then (Kp is B)(Ki is S)(Kd is S) (1)
5. If (e(t) is NB) and (de(t) is PS) then (Kp is B)(Ki is S)(Kd is S) (1)
6. If (e(t) is NB) and (de(t) is PM) then (Kp is B)(Ki is S)(Kd is S) (1)
7. If (e(t) is NB) and (de(t) is PB) then (Kp is B)(Ki is S)(Kd is S) (1)
8. If (e(t) is NM) and (de(t) is NB) then (Kp is S)(Ki is M)(Kd is B) (1)
9. If (e(t) is NM) and (de(t) is NM) then (Kp is B)(Ki is M)(Kd is B) (1)
10. If (e(t) is NM) and (de(t) is NS) then (Kp is B)(Ki is S)(Kd is S) (1)

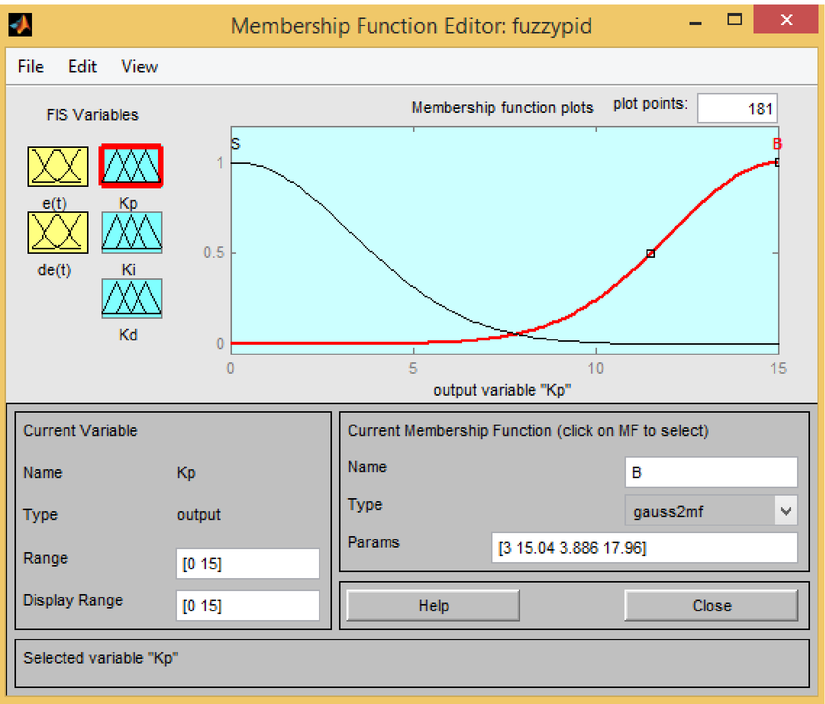
Connection
○ or
● and

Weight: 1

FIS Name: fuzzypid

Figure 3.10 Rule Editor

The membership functions for Fuzzy output variables (Kp, Ki, Kd) can be shown in Figures 3.11, 3.12 and 3.13:



plot points: 181

Current Variable

Name Kp

Type output

Range [0 15]

Display Range [0 15]

Current Membership Function (click on MF to select)

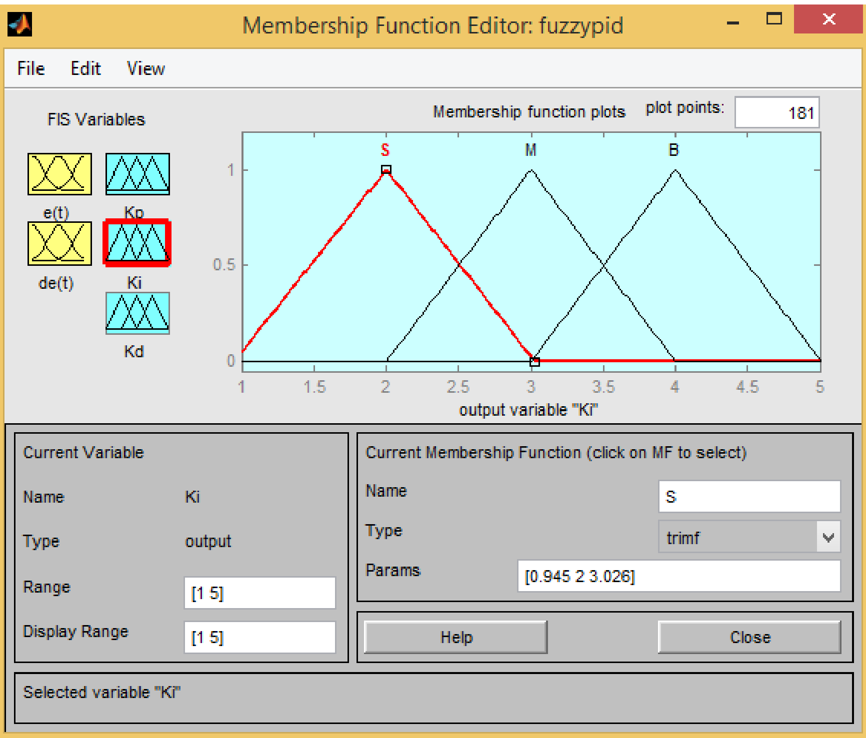
Name B

Type gauss2mf

Params [3 15.04 3.886 17.96]

Selected variable "Kp"

Figure 3.11 Membership functions for Kp



Current Variable

Name Ki

Type output

Range [1 5]

Display Range [1 5]

Current Membership Function (click on MF to select)

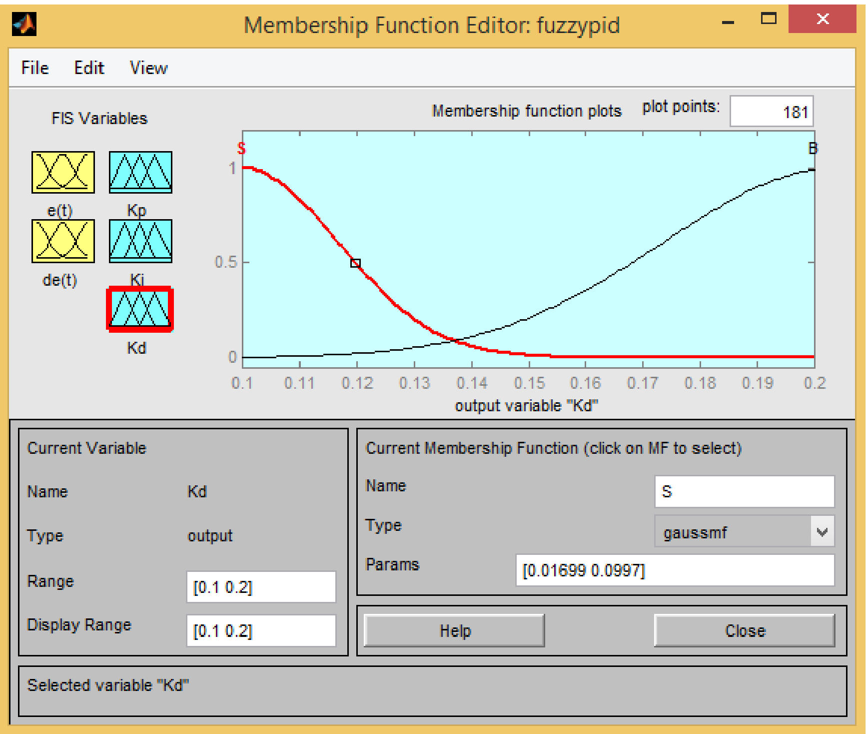
Name S

Type trimf

Params [0.945 2 3.026]

Selected variable "Ki"

Figure 3.12 Membership functions for Ki



Current Variable

Name Kd

Type output

Range [0.1 0.2]

Display Range [0.1 0.2]

Current Membership Function (click on MF to select)

Name S

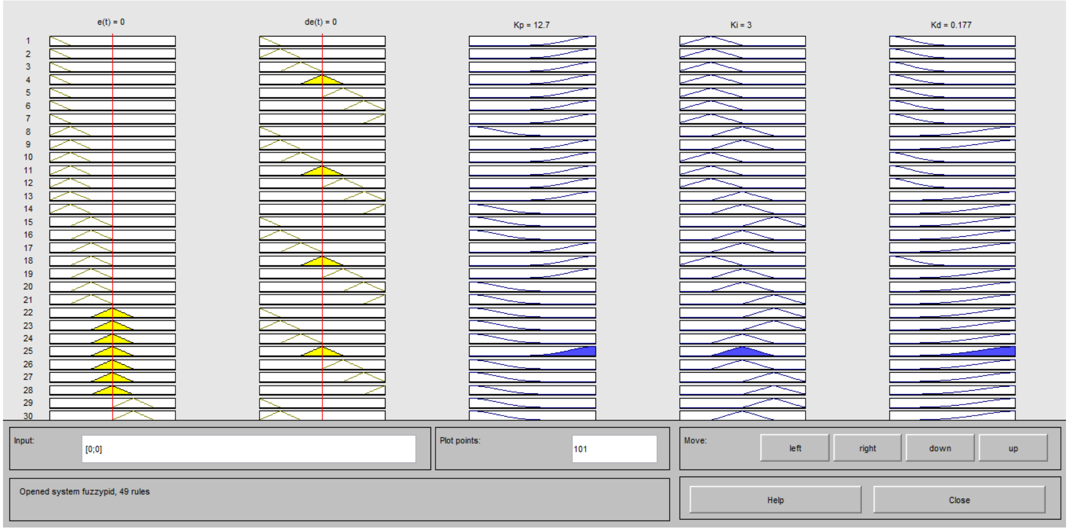
Type gaussmf

Params [0.01699 0.0997]

Selected variable "Kd"

Figure 3.13 Membership functions for Kd

Figure 3.14 shows the Rule viewer got the parameters from the SIMULINK Model for Control of the DC motor using Fuzzy PID controller.



e(t) = 0

de(t) = 0

Kp = 12.7

Ki = 3

Kd = 0.177

Input: [0;0]

Plot points: 101

Opened system fuzzypid, 49 rules

Figure 3.14 Rule viewer

In previous Figures SIMULINK Model for Control of DC motor using Fuzzy PID controller resulted in Rule surface viewers of Kp, Ki, Kd shown in the Figure s 3.15, 3.16, 3.17.

Figure 3.18 shown the simulation results for using speed Fuzzy controllers with the DC motor, and explains the effect of using the Fuzzy controller for controlling the speeding DC motor. Y axis represented voltage and X axis represented time.

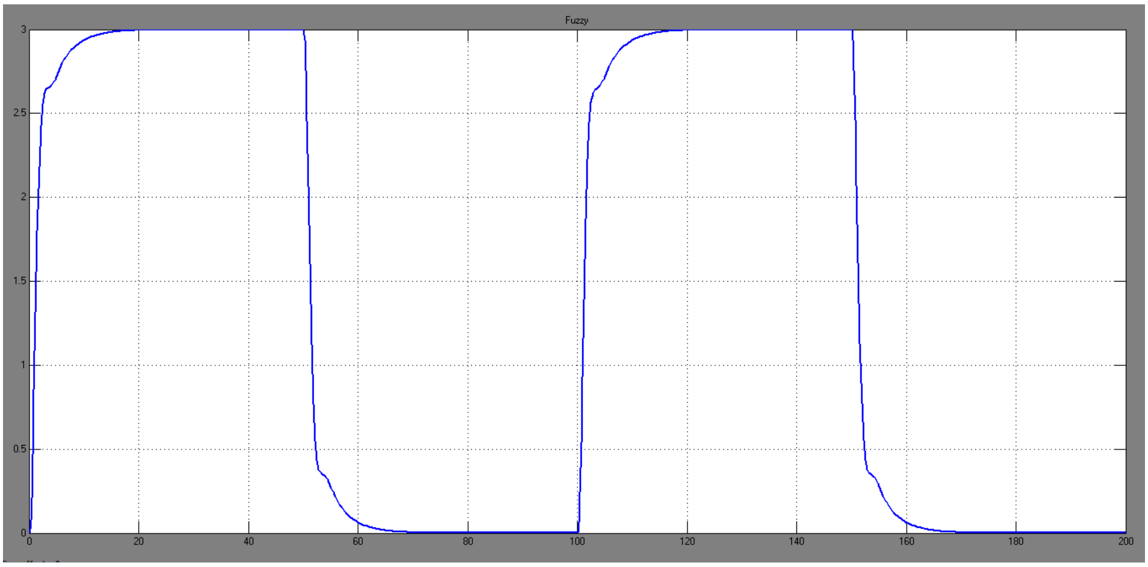


Figure 3.18 Simulation results of using speed PID controllers.

Figure 3.18 shown the effect of using the Fuzzy logic controller, Figure 3.18 has rise time with DC motor used in this simulation, settling time starts from 2.6v to 3 and with a duration of 9ms.

Figure 3.19 shows the effect of using Fuzzy logic and PID controllers and comparing between them. The red signal represented the output of PID and the blue one represented the output of Fuzzy logic. Y axis represented voltage and X axis represented time.

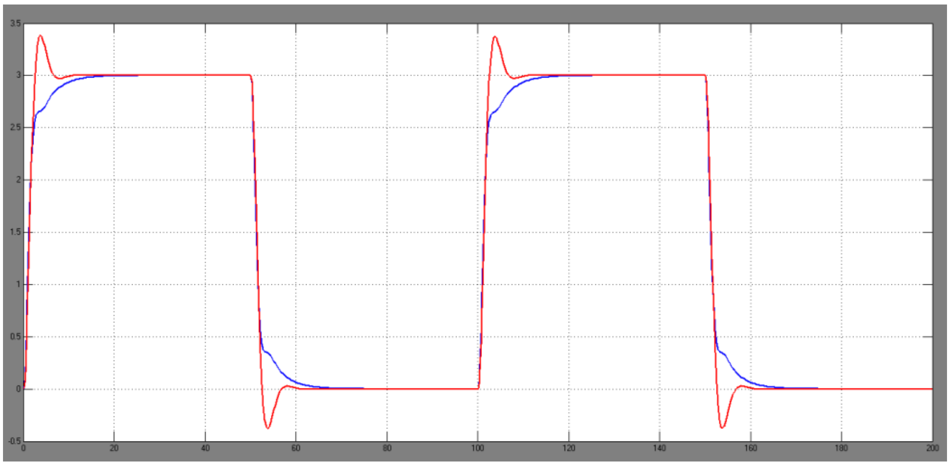


Figure 3.19 compare between Fuzzy logic and PID controllers

CHAPTER FOUR

IMPLEMENTATION OF DC MOTOR SYSTEM

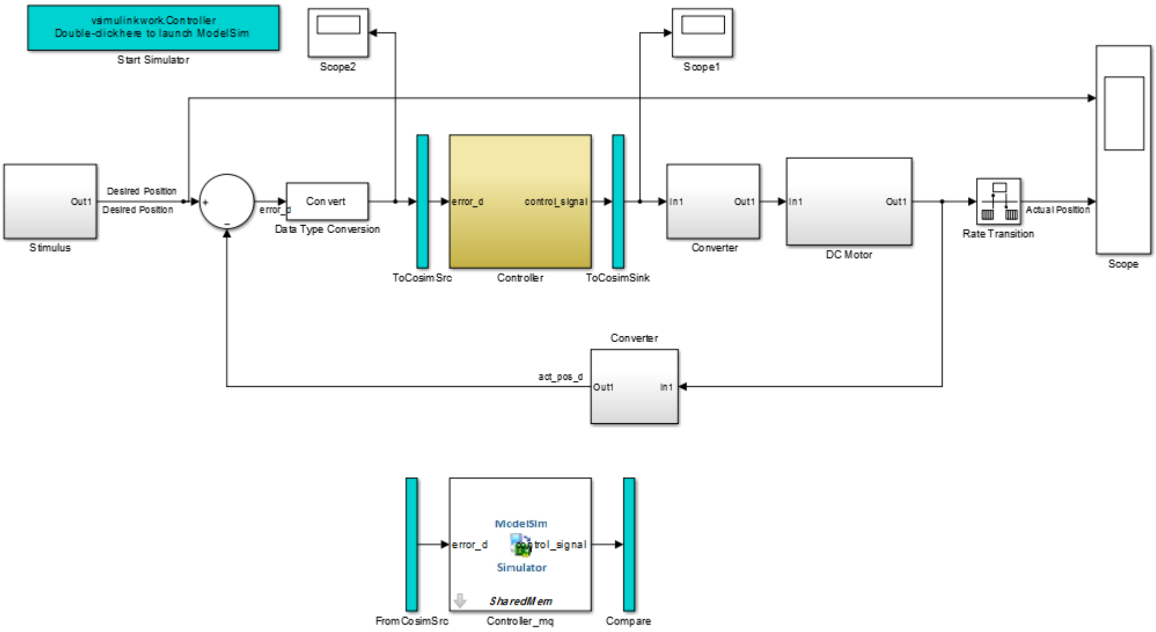


Figure 4.1 MATLAB SIMULINK model for real time implementation on the DC Motor using PID algorithm.

In Figure 4.1 a PID algorithm was implemented to observe a result with zero error, in this process SIMULINK converted the input to PID from analog to digital. And after that, translate the output from PID and input to DC Motor and from digital to analog. The PID algorithm built is shown in Figure 4.2:

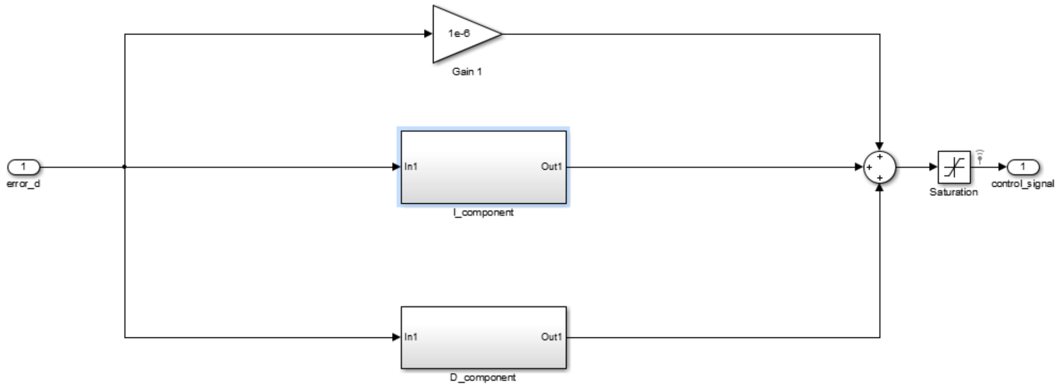


Figure 4.2 PID controller design

Inside the I_component, component represent the Integral phase is built as shown in Figure 4.3 and inside the D_component, another component represents the Derivative is built as shown in Figure 4.4

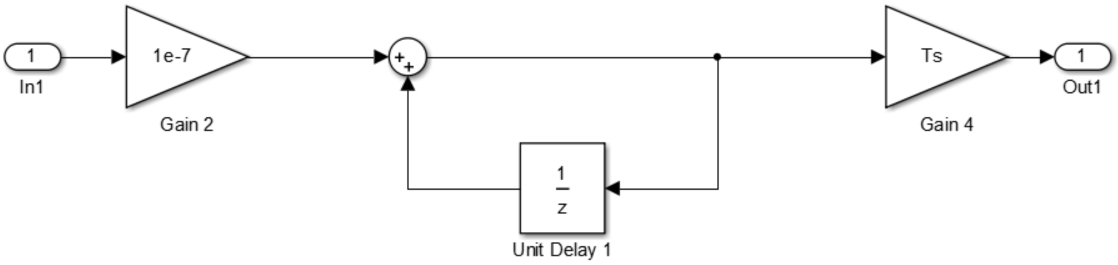


Figure 4.3 PID controller design Integral component

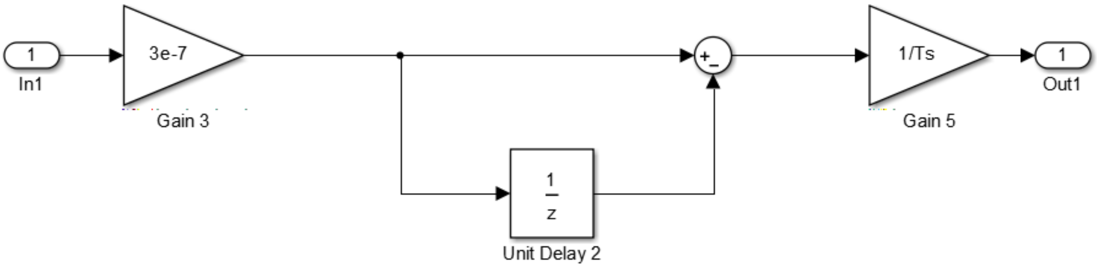


Figure 4.4 PID controller design Derivative component

Figure 4.5 shows the input to the PID algorithm after converting it from analog to digital. Y axis here represented the input voltage to PID and X axis represented the time.

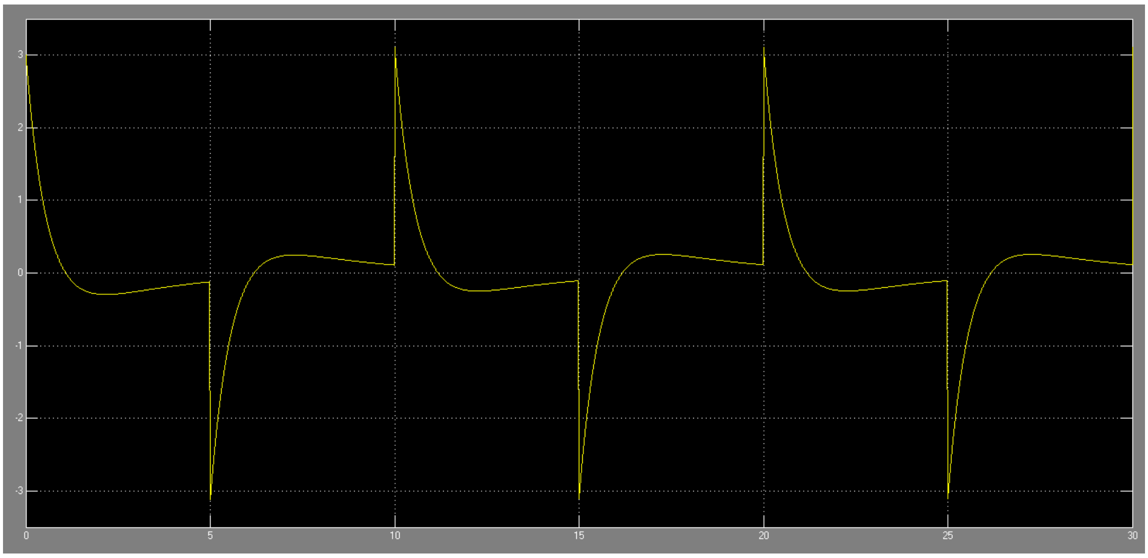


Figure 4.5 the input to the PID algorithm

The output of MATLAB SIMULINK model for real time when implemented on the DC motor using the PID algorithm can be shown in Figure 4.6 and the output without overshoot and without error can also be seen in Figure 4.6

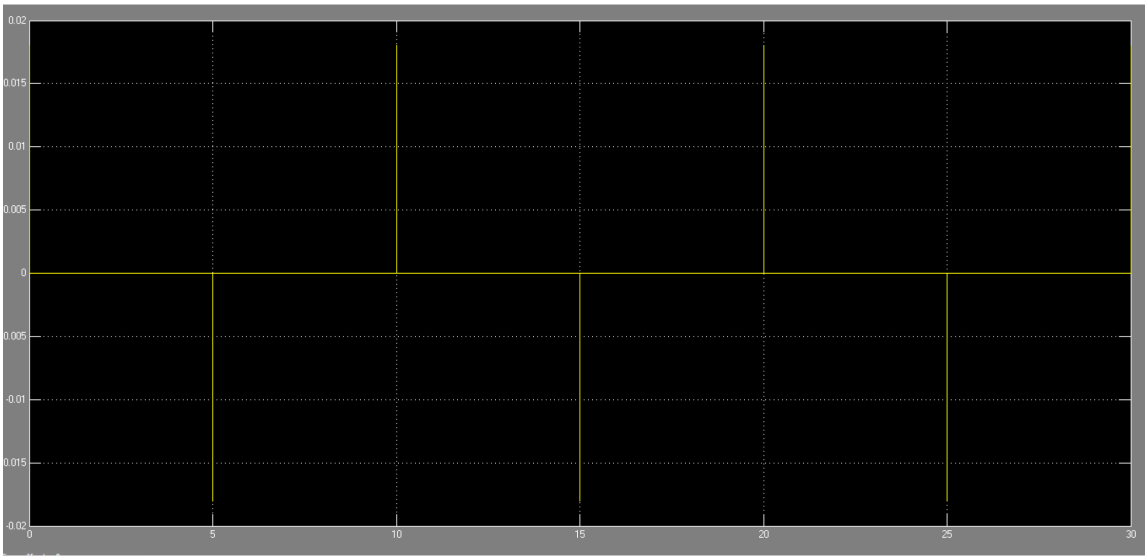
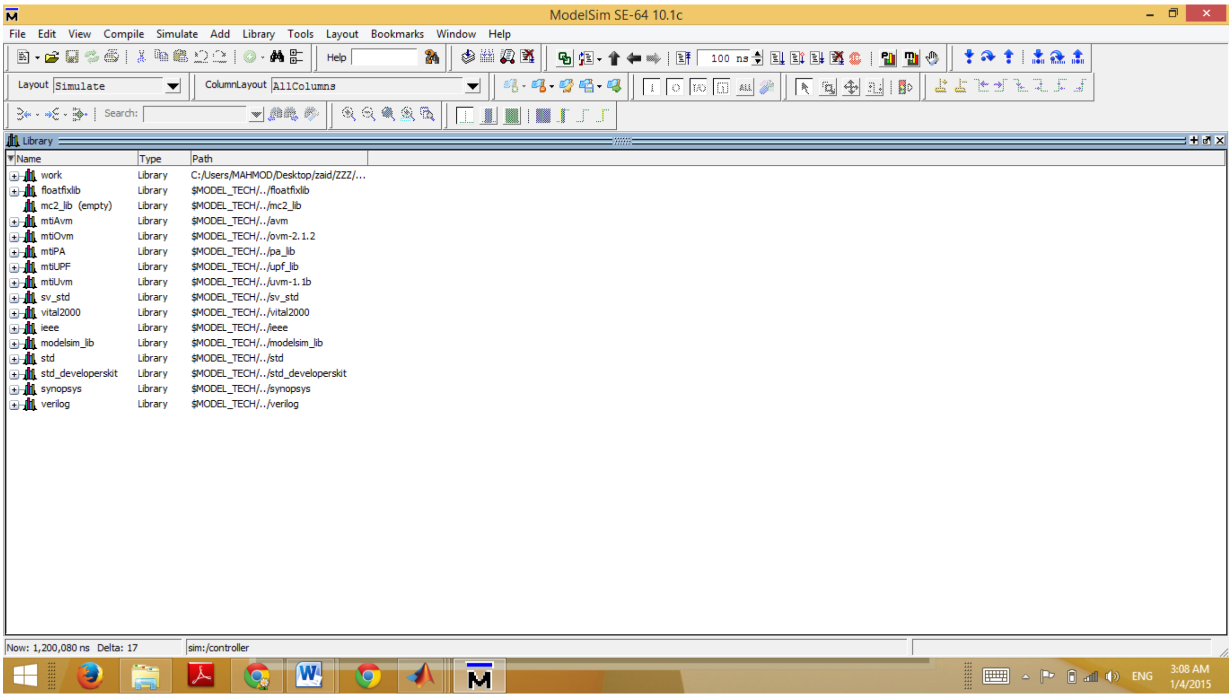


Figure 4.6 Output of PID algorithms

Generate VHDL code for FPGA, HDL Coder generates portable, synthesizable VHDL and Verilog code from MATLAB functions, SIMULINK models, and State flow charts. The generated HDL code can be used for FPGA programming or ASIC prototyping and design.

HDL Coder provides a workflow advisor that automates the programming of Xilinx and Altera FPGAs. We can control HDL architecture and implementation, highlight critical paths, and generate hardware resource utilization estimates. HDL Coder provides traceability between the SIMULINK model and the generated Verilog and VHDL code, enabling code verification for high-integrity applications adhering to DO-254 and other standards.

We implemented and simulated the DC motor in the FPGA using SIMULINK MATLAB and MODELSIM simulator, Figure 4.7 shows the implementation using MODELSIM SE 10.1c used here.



Layout Simulate

ColumnLayout AllColumns

Name	Type	Path
work	Library	C:/Users/MAHMOD/Desktop/zaid/ZZZ/...
floatfixlib	Library	$MODEL_TECH/../floatfixlib
mc2_lib (empty)	Library	$MODEL_TECH/../mc2_lib
mtiAvm	Library	$MODEL_TECH/../avm
mtiOvm	Library	$MODEL_TECH/../ovm-2.1.2
mtiPA	Library	$MODEL_TECH/../pa_lib
mtiUPF	Library	$MODEL_TECH/../upf_lib
mtiUvm	Library	$MODEL_TECH/../uvm-1.1b
sv_std	Library	$MODEL_TECH/../sv_std
vital2000	Library	$MODEL_TECH/../vital2000
ieee	Library	$MODEL_TECH/../ieee
modelsim_lib	Library	$MODEL_TECH/../modelsim_lib
std	Library	$MODEL_TECH/../std
std_developerskit	Library	$MODEL_TECH/../std_developerskit
synopsys	Library	$MODEL_TECH/../synopsys
verilog	Library	$MODEL_TECH/../verilog

Now: 1,200,080 ns Delta: 17

sim:/controller

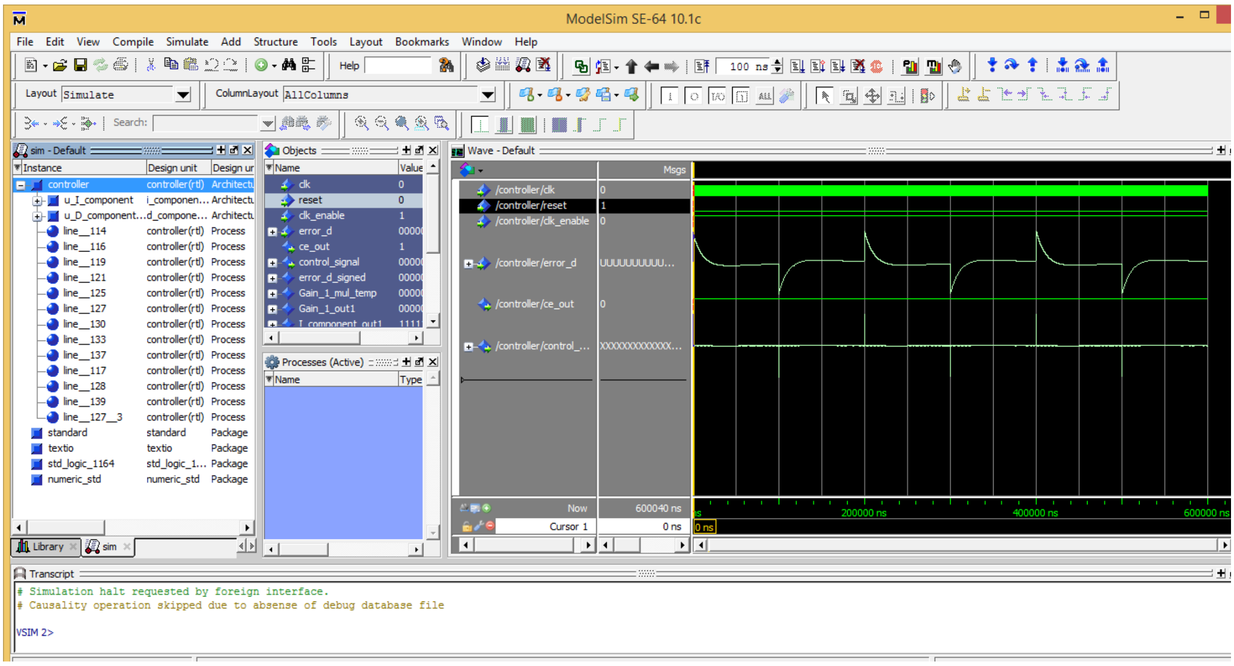
Figure 4.7 Modelism SE 10.1c

4.1. Generating VHDL code for FPGA

The FPGA industry sprouted from programmable read-only memory (PROM) and programmable logic devices (PLDs). PROMs and PLDs both had the option of being programmed in batches in a factory or in the field (field-programmable). However, programmable logic was hard-wired between logic gates [17]. In the late 1980s, the Naval Surface Warfare Department funded an experiment proposed by Steve Casselman to develop a computer that would implement 600,000 reprogrammable gates. Casselman was successful and a patent related to the system was issued in 1992 [17].

HDL Coder generates portable, synthesizable VHDL and Verilog code from MATLAB functions, Simulink models, and Stateflow charts. The generated HDL code can be used for FPGA programming or ASIC prototyping and design.

HDL Coder provides a workflow advisor that automates the programming of Xilinx and Altera FPGAs. We can control HDL architecture and implementation, highlight critical paths, and generate hardware resource utilization estimates. HDL Coder provides traceability between Simulink model and the generated Verilog and VHDL code, enabling code verification for high-integrity applications adhering to DO-254 and other standards. We implemented and simulated the DC motor in the FPGA using SIMULINK MATLAB and MODELSIM simulator, Figure 4.7 shows the implementation and results obtained on FPGA which are exactly similar to the result that were obtained from MATLAB shown in Figure 4.6



```
# Simulation halt requested by foreign interface.
# Causality operation skipped due to absense of debug database file

VSIM 2>
```

Figure 4.8 result for system on FPGA

CHAPTER FIVE

CONCLUSION

By reviewing results obtained from a number of experimental methods which includes classical PID controllers, Fuzzy logic control and PID implemented in FPGA using SIMULINK MATLAB and MODELSIM. It’s clear that the overall performance of PID implemented in the FPGA using SIMULINK MATLAB and MODELSIM was better than the classical methods for tuning PID controllers. PID implemented in FPGA and has shown acceptable results and improvements of the system behavior.

When comparing between the three methods that were used in this research we can recommend that the engineers can depend on PID implemented in the FPGA using SIMULINK MATLAB and MODELSIM because of the steady state results obtained from this method.

The application software, MATLAB-SIMULINK, provided extreme help in this research project and affected the results and the credibility in a positive meaning.

REFERENCES

[1] B. Liptak, “Process Control: Instrument Engineers Handbook”, Fourth Edition, Volume Two: Process Control and optimization, pages 114-120, 2005

[2] Abdullah I. Al-Odienat, Ayman A. Al-Lawama, “The Advantages of PID Fuzzy Controllers Over The Conventional Types”, Mutah University, American Journal of Applied Sciences 5 (6): 653-658, 2008

[3] Reetam Mondal, Arumay Mukhopadhyay, Debdoot Basak, “Embedded System of DC Motor Closed Loop Speed Control based on 8051 Microcontroller”, Procedia Technology, Volume 10, 2013, Pages 840-848.

[4] E.A. Ramadan, M. El-bardini, M.A. Fkirin, “Design and FPGA-implementation of an improved adaptive Fuzzy logic controller for DC motor speed control”, Ain Shams Engineering Journal, Volume 5, Issue 3, September 2014, Pages 803-816.

[5] Wenbin Yan, Dada Wang, Pengfei Jia, Weiguo Li, “The PWM speed regulation of DC motor based on intelligent control”, Systems Engineering Procedia, Volume 3, 2012, Pages 259-267.

[6] S.A.KH. Mozaffari Niapour, M. Tabarraie, M.R. Feyzi, “A new robust speed-sensorless control strategy for high-performance brushless DC motor drives with reduced torque ripple”, Control Engineering Practice, Volume 24, March 2014, Pages 42-54.

[7] R.M. Stephan, V. Hahn, J. Dastych, H. Unbehauen, “Adaptive and robust cascade schemes for thyristor driven DC-motor speed control”, Automatica, Volume 27, Issue 3, May 1991, Pages 449-461.

[8] Urmila Meshram, Pankaj Bande and R. R. Harkare, " Hardware and Software Co-design for Robot Arm Position Control Using VHDL & FPGA ". International Conference on Advances in Recent Technologies in Communication and Computing. 2009, pp. 780-782.

[9] N. C. PEMMARAJU,” Intelligent Speed Control Of DC Motor Using ANFIS”, Texas A&M University, Master thesis, December 2006

[10] F. M. DARBI, “A Comparison Between Soft Computing Motor System” Atilim University, May 2014, thesis pages 10-15.

[11] M.J. Willis, “Proportional-Integral-Derivative Control”, University of Newcastle, October, 1999, pp.

[12] J. Jantzen, K. Lyngby, “Tutorial On Fuzzy Logic”, Technical University of Denmark, Oersted-DTU, Automation, Bldg 326, 2800, DENMARK. Tech. report no 98-E 868 (logic), revised 17 Jun 2008

[13] G. J. Klir, B. Yuan, “Fuzzy Sets and Fuzzy Logic”, Theory and application, 1995

[14] M. S. Stachowicz, “Introduction to Intelligent Control”, University of Minnesota Duluth, Feb 2010.

[15] Pandey, Manoj, “Analysis of Self Tuning Fuzzy PID Internal Model Control”, Thapar University(TU), Master Thesis, July 2010.

[16] Control Theory 1 Laboratory 1, “Introduction to simulink and dSPACE”, http://routh.ee.adfa.edu.au/~irp/T eaching/Control_Theory_l /Lab 1 /lab 1 .pdf, accessed on October 29, 2006.

[17]web.archive.org/web/20070412183416/http://filebox.vt.edu/users/tmagin/history.htm