

CHAPTER ONE

INTRODUCTION

1.1 Introduction

With the development of power electronics resources, the direct current machine has become more and more useful. The speed of DC motor can be adjusted to a great extent as to provide easy controllability and high performance. DC motors used in many applications such as steel rolling mills, electric trains, electric vehicles, electric cranes and robotic manipulators require speed controllers to perform their tasks. The most flexible control is obtained by means of separately excited DC Motor in which the armature and field circuits are provided with separate sources. For the armature source a controlled rectifier or chopper is required. Armature voltage control method is used to vary the speed up to the rated speed and the motor operates in the constant torque region.

There are several Conventional as well as intelligent controllers to control the speed of DC motor such as: PID Controller, Fuzzy Logic Controller, Neuro-Fuzzy Controller etc...

Fuzzy logic applications to the control of drive systems have been increasing exponentially in the past few years. Drive systems possess inherent characteristics, such as nonlinearities, unavailability of a precise model or its excessive complexity, that make them well suited for FL control. Fuzzy control can be applied to virtually any task that relied on human experience and intuition. The fuzzy approach is advantageous for speed control of the motor because of intuitiveness, simplicity, easy implementation, and minimum knowledge of system dynamics to acquire a robust, fast, and precise control of motor drive.

1.2 Statement of the Problem

There have been several conventional control techniques in DC motor drives. The conventional control strategies are a fixed structure, fixed parameter design. Hence the tuning and optimization of these controllers are challenging and difficult task,

particularly, under varying load conditions, parameter changes, abnormal modes of operation, etc. Attempts to overcome such limitations using adaptive and variable structure control have had limited success due to complexity, requiring of estimation stages, model structure changes due to discontinuous drive mode of operation, parameter variations, load excursions and noisy feedback speed and current signals.

1.3 Objective

The main objectives of this study are:-

- Design a control system for a separately excited DC Motor using fuzzy logic controller.
- Simulation of control system for a separately excited DC Motor using MATLAB
- Compare fuzzy logic controller with PID controller.

1.4 Methodology/Approach

Literature review of related study to understand the separately excited DC Motor and fuzzy logic. Mathematical and computer models of a DC motor and fuzzy logic is developed. MATLAB is used for system simulation.

1.5 Layout

This thesis consists of five chapters: chapter one contain an introduction to the principles of this work, statement of problem, objective and Methodology. Chapter two discusses a theoretical background of dc machines, DC motors and Separately Excited DC Motors; it is also contain information about PID control, fuzzy control and loads. Chapter three presents the system control design of speed control for the separately excited DC motor. Chapter four presents the simulation results .chapter five provides the conclusion and recommends.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The electrical machine is the essential element in the field of power generation and electrical drives. Duty of the electrical machine is a save, economic and ecological generation of electrical energy as well as its low-loss transformation for distribution purposes and its accordant utilization in electrical drive applications. The electrical machine is utilized in centralized as well as in distributed energy transducer systems. Electrical machines appear as alternators in power plants and in solitary operation, also as transformers and transducers in electrical installations. They are also used as drive motors in industrial, trade, agricultural and medical applications, machine tools, buildings and household appliances. Railway, automotive, naval, aviation and aeronautic systems are equipped with electrical machines as well. Special machine models are used in magnetic-levitation technology and induction heating. The power range leads from mW to GW. Requirements of the entire system determine the design conditions of the electrical machine. Essential factors like functionality, costs, availability and influence on the environment need to be taken into account.

2.2 DC Machines

Electrical machines appear as different types of construction. Most common types are DC machines as well as rotating field machines such as induction or synchronous machines. Due to its name, the DC machine is fed by DC current. Rotating field machines are to be supplied by a three-phase alternating current, called three-phase AC (Alternating Current). In case of a single-phase AC current availability, universal motors (AC-DC motors) and single phase induction machines are applied. Basically electrical machines can be operated in both motor and generator mode [1]. Figure 2-1 shows DC machine and its parts.

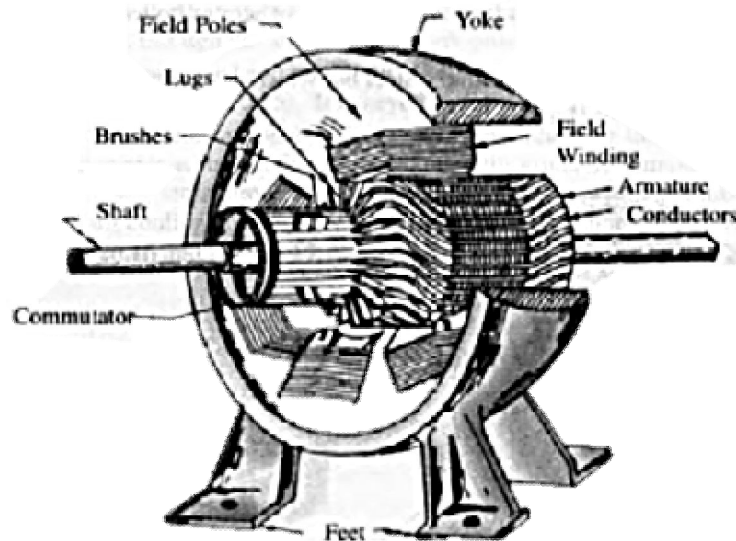


Figure 2-1: DC machine

2.2.1 Main part of DC machine

❖ Stator

It consists of of yoke that is lower grade steel; supports poles; return path for flux; encloses machine, and poles which are core and shoes; high grade steel. Pole core may or may not be laminated. Pole shoes are usually laminated (flux fluctuation due to rotating armature slots). Pole shoes (i) reduce reluctance of the air gap (by increasing its area); (ii) improve flux density distribution in the air gap; and (iii) provide mechanical support for the field coils. Stator also contain field coils which provide the MMF for the main working flux. There may be more than one set of coils. The coils of one set are identical to each other, and are connected together electrically.

❖ Rotor (Armature)

The armature is made of high-grade steel laminations punched and stacked together to form a cylinder or drum. It is mounted on the shaft (directly for small machines, and by means of a spider for larger machines). It may have radial and axial ventilation ducts for cooling. The armature winding are placed in slots around the armature periphery.

❖ Air gap

The air gap is the space between stator and rotor needed to allow relative motion between them. Magnetically, it introduces a high reluctance in the path of the

working flux (most of the field MMF is consumed in the air gap); it is therefore made as short as possible, typically 0.5-5.0 mm.

❖ Commutator

The function of the commutator is to interface between the alternating currents and EMF's in the rotating armature coils on the one side, and the direct current and voltage at the machine terminals.

Brushes are carbon or graphite blocks mounted in stationary holders with spring pressure to maintain good electrical contact with the rotating commutator segments [2].

2.2.2 DC machine operation

The operation of dc machines is based on the interaction between the armature conductors and the air gap field, which results in induced electro motive force (EMF) and developed torque. The main field is the field produced by the field coils. DC machine can be used either as a motor or as a generator [3]. Figure 2-2 shows motor and generator action.

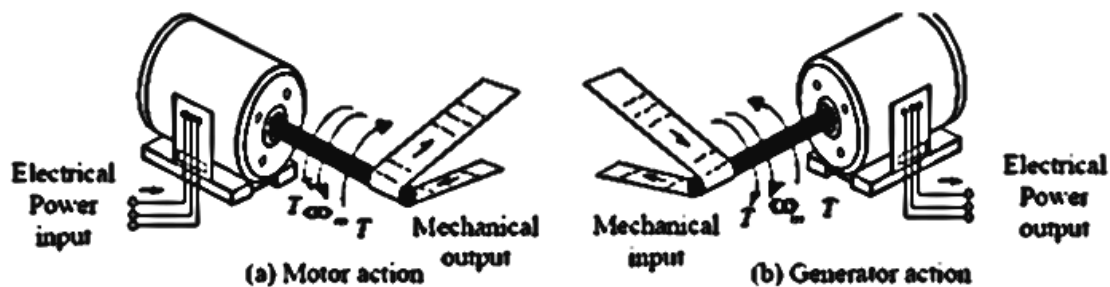


Figure 2-2: Motor and generator action

❖ Motor operation

The input power is electrical, and the output power is mechanical. Part of the input power is lost as electrical (copper) losses in the windings, and the remainder is available for electromechanical energy conversion; part of the converted power is lost supplying the losses due to rotation, and the remainder is available as a mechanical output power to drive the load. That is, the shaft torque available at the load, T_L , is less than the developed torque T_d .

❖ Generator operation

The input power is mechanical, and the output power is electrical. Part of the input is lost as rotational losses, and the remainder is available for electromechanical energy conversion; part of the converted power P_c is lost as electrical (copper) losses in the windings, and the remainder is available as electrical output power to supply the load [2].

2.3 DC Motor

Motors come in many sizes and types, but their basic function is the same. Motors of all types serve to convert electrical energy into mechanical energy. They can be found in elevators, CD players, toys, robots, automobiles, and many other places. D.C. motors are motors that run on Direct Current from a battery or DC power supply. Direct Current is the term used to describe electricity at a constant voltage. When a battery or DC power supply is connected the motor converts electrical energy to mechanical work as the output shaft turns [4].

2.3.1 Applications

There are two main types of application for which DC motors are more suitable than AC motors: battery operated equipment, and applications requiring accurate or flexible control of speed or torque. Battery-operated equipment includes small portable apparatus, cordless tools, and toys, as well as electric drives in road vehicles. The accurate control of dc motors makes them suitable for servomotor duty in automatic control systems. The flexibility of control of dc motors makes them suitable for certain heavy power applications such as lifts, cranes, hoists, and electric traction (electric trains), as well as certain drives in heavy industry. These applications can involve frequent changes in speed, stops and starts, and possibly reversals [2].

2.3.2 Principle of operation

There are two conditions which are necessary to produce a force on the conductor. The conductor must be carrying current, and must be within a magnetic field. When these two conditions exist, a force will be applied to the conductor, which will attempt to move the conductor in a direction perpendicular to the magnetic field. This is the basic theory by which all DC motors operate. In an actual DC motor, several such coils are wound on the rotor, all of which experience force,

resulting in rotation. The greater the current in the wire, or the greater the magnetic field, the faster the wire moves because of the greater force created. At the same time this torque is being produced, the conductors are moving in a magnetic field. At different positions, the flux linked with it changes, which causes an EMF to be induced. This voltage is in opposition to the voltage that causes current flow through the conductor and is referred to as a counter-voltage or back EMF.

The value of current flowing through the armature is dependent upon the difference between the applied voltage and this counter-voltage. The current due to this counter-voltage tends to oppose the very cause for its production. it results in the rotor slowing down[3].

2.3.3 DC motor method

DC motors are described by the method used to excite the field. The four most common methods are: separately-excited, shunt-connected, series-connected, and compound. The separately-excited machine has no physical connection between the field and armature windings. Each circuit is excited from its own power supply. A shunt-connected machine has the field circuit connected in parallel with the armature circuit. Both circuits have the same total voltage drop across them. The series-connected machine has the field circuit in electrical series with the armature circuit. Both circuits share the same current. A compound machine contains two independent field circuits. One field circuit is connected in series with the armature circuit, and the other field circuit is connected to shunt either the armature circuit or the series combination of series field and armature circuits. The former is called a “short” shunt, and the latter is called a “long” shunt. Figure 2-3 shows the various connections [5].

2.4 Separately Excited DC Motors

Separately excited DC motors are very often used as actuators in industrial applications. These actuators have low friction, small size, high speed, low construction cost, no gear backlash, operate safely without the use of limit switches and generate moderate torque at a high torque to weight ratio [6]. Figure 2-4 shows separately excited DC motor.

2.4.1 Operation

When a separately excited motor is excited by a field current (I_f) and an armature current (I_a) flows in the circuit, the motor develops a back EMF and a torque to balance the load torque at a particular speed. The (I_f) is independent of the I_a , each windings are supplied separately. Any change in the armature current has no effect on the field current [7].

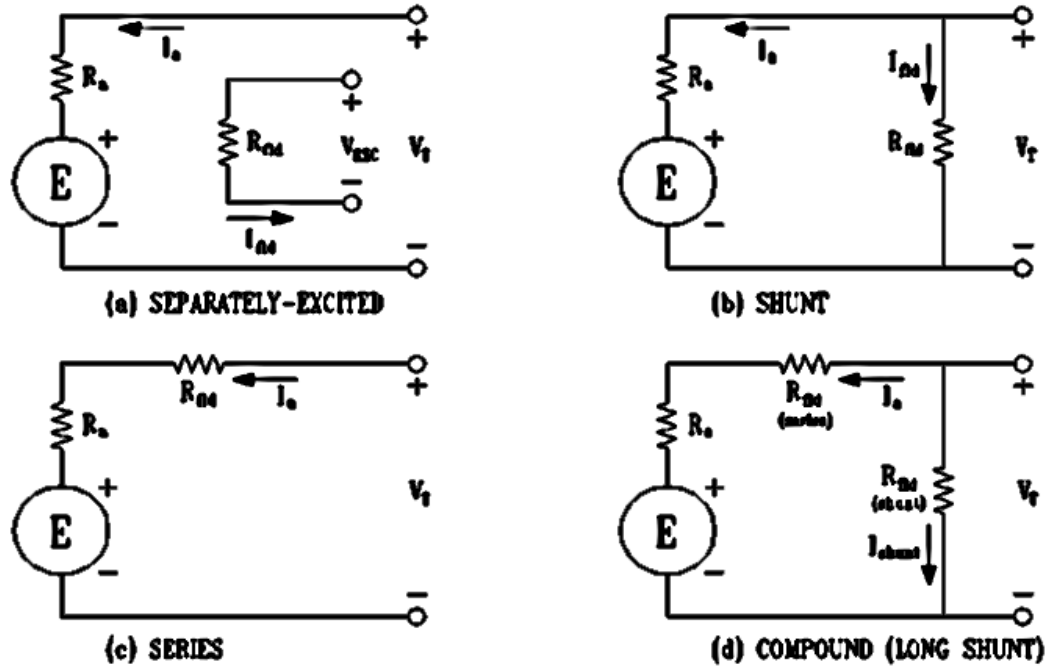


Figure 2-3: DC motor connections

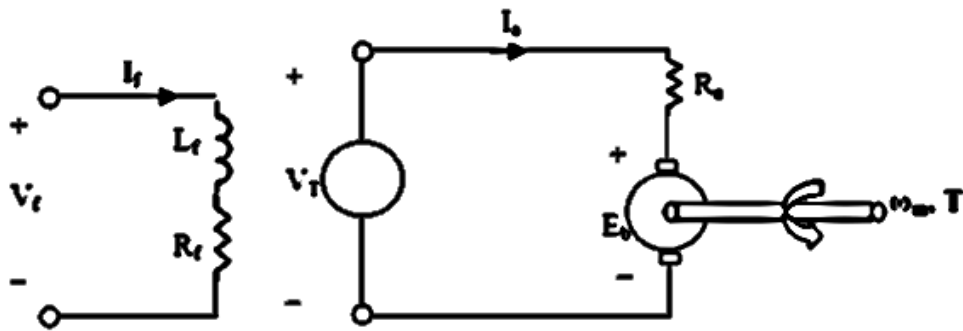


Figure 2-4: Separately excited dc motor

2.4.2 Represent equations

The dynamic of the separately excited DC motor may be expressed by the equations [8].

$$v_a(t) = R_a \cdot i_a(t) + L_a \cdot \frac{di_a(t)}{dt} + K_b \cdot w(t) \quad (2-1)$$

$$K_T \cdot i_a(t) = J_m \cdot \frac{dw(t)}{dt} + B_m \cdot w(t) + T_L \quad (2-2)$$

Where v_a = armature voltage (v)

R_a = armature resistance (Ω)

i_a = armature current (A)

L_a = armature inductance (H)

W = angular speed (rad/s)

J_m = rotor inertia (kgm^2)

B_m = viscous friction coefficient (Nms/rad)

K_T = torque constant (Nm/A)

K_b = back emf constant (Vs/rad)

T_L = load torque (Nm)

2.4.3 Torque Characteristic

If the demagnetizing effect of armature reaction is neglected, the developed torque T_d will be directly proportional to the armature current i_a , so that the two variables are related by the straight line shown dotted in Figure 2-5. Armature reaction may reduce the flux ϕ , and hence reduce T_d , so that the actual relationship between T_d and i_a is curved slightly below the straight line. The load torque T_L is less than the developed torque due to rotational losses, so that the T_L curve is slightly below the T_d curve, Figure 2-5 shows the relationship between torque and current is sometimes called the torque characteristic of the motor [2] .

2.4.4 Control of motor speed

Control of speed can be done by use of different methods.

❖ Armature voltage control

In separately-excited motors, the voltage applied to the motor can be varied with the field remaining constant. Different voltages then give different intercepts (different no load speeds), and result in a family of parallel (i.e. same slope) mechanical c/s as in Figure 2-6.

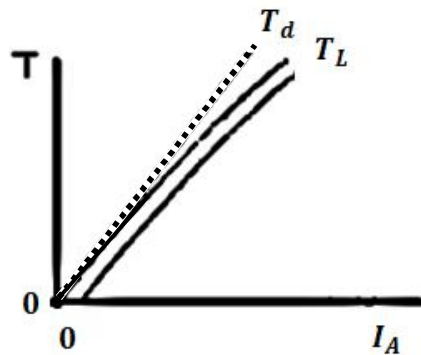


Figure 2-5: Torque characteristic

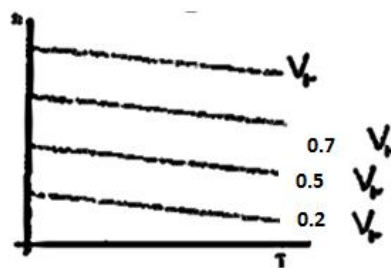


Figure 2-6: Different speeds for different voltage

The simplest method of obtaining variable DC voltage is to use a voltage divider, but this method is impractical and uneconomical; it is used only for testing. In modern applications, variable DC voltage for the armature is often obtained from a solid-state controlled rectifier, with the field fed from an uncontrolled rectifier, Figure 2-7.

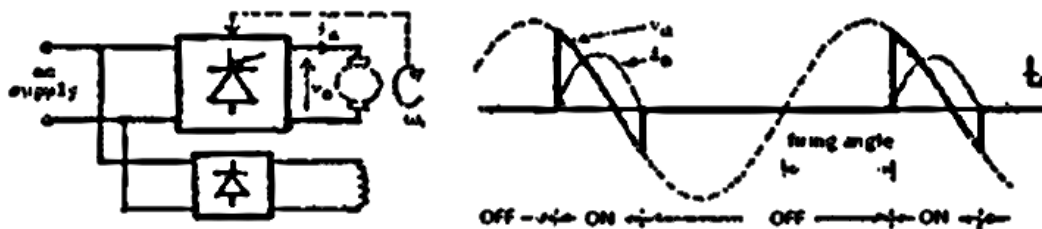


Figure 2-7: Solid-state controlled rectifier with the field fed from an uncontrolled rectifier

The firing angle of the controlled rectifier may be changed manually, but in practice it is adjusted automatically using a speed signal or armature current signal (i.e. load), or both for optimum control. In road vehicles, the supply is itself dc,

and hence needs no rectification. Voltage control is often obtained by an electronic chopper circuit. Choppers may use Pulse-Width Modulation (PWM) at constant frequency or Pulse-Frequency Modulation (PFM) with constant pulse width. Another effective method for obtaining smooth voltage control is the Ward-Leonard system. The dc motor is fed from a dc generator driven by some prime-mover (e.g. ac motor or diesel engine). By varying the field excitation of the generator, the armature voltage of the motor varied (and can be even reserved). The motor field is fed from an exciter (small DC generator) or rectifier at constant voltage. The Ward-Leonard system is generally more expensive than a solid-state drive, but has compensating advantages for certain applications.

❖ **Field control**

If the field circuit resistance is increased, the field current, and hence the main field, will be reduced, and the speed will increase. The higher the field resistance, the higher the intercept and the greater the slope (i.e. the c/s becomes softer). The flux cannot be reduced indefinitely because the speed becomes too high and may damage the motor. Moreover, if the main field becomes too weak, the demagnetizing effect of armature reaction becomes prominent (relatively large) which may lead to instability [2].

2.5 PID Control

A Proportional–Integral–Derivative (PID) controller is a three-term controller that has a long history in the automatic control field, starting from the beginning of the last century. Owing to its intuitiveness and its relative simplicity, in addition to satisfactory performance which it is able to provide with a wide range of processes, it has become in practice the standard controller in industrial settings. It has been evolving along with the progress of the technology and nowadays it is very often implemented in digital form rather than with pneumatic or electrical components. It can be found in virtually all kinds of control equipment's, either as a stand-alone (single-station) controller or as a functional block in Programmable Logic Controllers (PLCs) and Distributed Control Systems (DCSs). Actually, the new potentialities offered by the development of the digital technology and of the software packages have led to a significant growth of the research in the PID control field: new effective tools have been devised for the improvement of the

analysis and design methods of the basic algorithm as well as for the improvement of the additional functionalities that are implemented with the basic algorithm in order to increase its performance and its ease of use.

The success of the PID controllers is also enhanced by the fact that they often represent the fundamental component for more sophisticated control schemes that can be implemented when the basic control law is not sufficient to obtain the required performance or a more complicated control task is of concern. [9]

2.5.1 The PID controller algorithm

In analog control system, PID controller is used commonly. The conventional PID controller is a linear control method. It compounds the outputs of proportional, integral and derivative parts linearly to control the system. Figure 2-8 shows the block diagram of the PID controller. The algorithm of PID controller can be given as follows:

$$e(t) = r(t) - y(t) \quad (2-3)$$

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t) dt + T_D \frac{de(t)}{dt} \right] \quad (2-4)$$

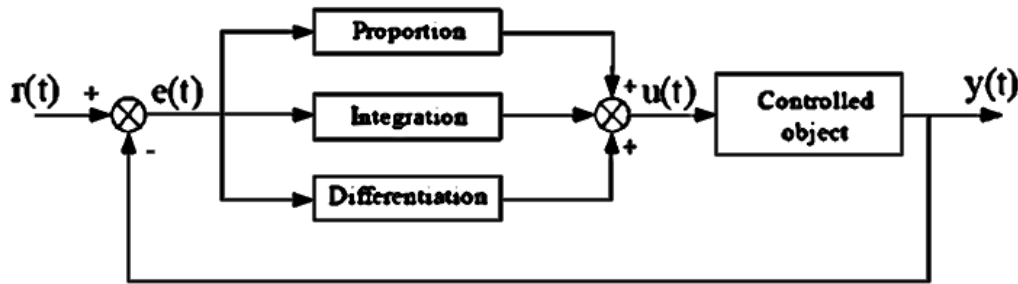


Figure2-8: The block diagram of the PID controller.

Where $y(t)$ is the output of the system, $r(t)$ is the reference input of the system, $e(t)$ is the error signal between $y(t)$ and $r(t)$, $u(t)$ is the output of the PID controller, K_p is proportional gain, T_i is integral time constant and T_D is derivative time constant. Equation also can be rewritten as below:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (2-5)$$

Where K_i is integral gain, K_d is derivative gain, and $K_i = K_p / T_i$, $K_d = K_p T_d$.

In PID controller, the relation between PID parameters and the system response specifications is clear. Each part has its certain function where: [10]

- ❖ **Proportion** can increase the response speed and control accuracy of the system. Bigger K_p can lead to faster response speed and higher control accuracy. But if K_p is too big, the overshoot will be large and the system will tend to be instable.
- ❖ **Integration** is used to eliminate the steady-state error of the system. With bigger K_i , the steady-state error can be eliminated faster. But if K_i is too big, there will be integral saturation at the beginning of the control process and the overshoot will be large.
- ❖ **Differentiation** can improve the dynamic performance of the system. But if K_d is too big, the response process will brake early, the regulating time will be prolonged.

2.5.2 Tuning of PID parameters

The tuning parameters of the PID controller should be set with in-depth consideration of the process dynamics. There are various tuning methods for tuning parameters of the PID controller on the basis of the process model.

i. Trial-and-error tuning

Trial-and-error tuning is used to determine the tuning parameters of a PID controller by inspecting the dynamic behavior of the controlled process output.

ii. Ziegler–Nichols tuning rule

The Ziegler–Nichols (ZN) tuning rule uses the ultimate gain and the ultimate period of the process. Table 1 provides the tuning parameters of the PID controller for the given ultimate data set of the process.

Procedure to tuning:

- (a) Increase the proportional gain until the system oscillates; that gain is the ultimate gain K_u .
- (b) Read the time between peaks T_u at this setting.

(c) Table 2-1 gives approximate values for the controller gains.

Table 2-1: the Ziegler-Nichols tuning parameters

Controller	K_P	K_i	K_D
P	$0.5K_u$	–	–
PI	$0.45K_u$	$T_u/1.2$	–
PID	$0.6K_u$	$T_u/2$	$T_u/8$

iii. Internal model control tuning rule

The objective of the Internal Model Control (IMC) tuning rule is to match the control performance of the PID controller with that of the IMC controller.

It needs the following First Order plus Time Delay (FOPTD) model:

$$G_m(s) = \frac{k \exp(-\theta s)}{\tau s + 1} \quad (2-6)$$

Where k , τ and θ denote the static gain, the time constant and the time delay respectively. The IMC tuning rule determines the tuning parameters using the formulas in Table 2-2 [11].

Table 2-2: Internal model control (IMC) tuning parameters

controller	Tuning parameters		
	kK_P	K_i	K_D
PI	$(2\tau + \theta)/2\lambda$	$\tau + \theta/2$	–
PID	$(2\tau + \theta)/2(\lambda + \theta)$	$\tau + \theta/2$	$\tau\theta/(2\tau + \theta)$

2.6 Fuzzy Control

Almost every industry and household has motors used in their equipment or appliances. Motors that are often controlled by computers have also become an essential part of many motion control systems. Major problems in applying a conventional control algorithm in a speed controller are the effects of nonlinearity in a DC motor. The nonlinear characteristics of a DC motor such as dead zone, saturation and friction could degrade the performance of conventional controllers. Many advanced model-based control methods such as variable-structure control

and model reference adaptive control have been developed to reduce these effects. However, the performance of these methods depends on the accuracy of system models and parameters.

Generally, an accurate non-linear model of an actual DC motor is difficult to find, and parameter values obtained from system identification may be only approximated values. Emerging intelligent techniques have been developed and extensively used to improve or to replace conventional control techniques because these techniques do not require a precise model. One of intelligent techniques, fuzzy logic developed by Zadeh is applied for controller design in many applications. A Fuzzy Logic Controller (FLC) was proved analytically to be equivalent to a nonlinear controller when a nonlinear defuzzification method is used. Also, the results from the comparisons of conventional and fuzzy logic control techniques in the form of the FLC and fuzzy compensator showed that fuzzy logic can reduce the effects of nonlinearity in a DC motor and improve the performance of a controller [12].

2.6.1 General features

- Fuzzy logic is conceptually easy to understand. The mathematical concepts behind fuzzy reasoning are very simple. What makes fuzzy nice is the “naturalness” of its approach and not its far-reaching complexity.
- Fuzzy logic is flexible. With any given system, it’s easy to layer more functionality on top of it without starting again from scratch.
- Fuzzy logic is tolerant of imprecise data. Everything is imprecise if you look closely enough, but more than that, most things are imprecise even on careful inspection. Fuzzy reasoning builds this understanding into the process rather than tacking it onto the end.
- Fuzzy logic can model nonlinear functions of arbitrary complexity. It can create a fuzzy system to match any set of input-output data. This process is made particularly easy by adaptive techniques like Adaptive Neuro Fuzzy Inference Systems (ANFIS), which are available in the Fuzzy Logic Toolbox.
- Fuzzy logic can be built on top of the experience of experts. In direct contrast to neural networks, which take training data and generate opaque, impenetrable

models, fuzzy logic helps reliance on the experience of people who already understand the system.

- Fuzzy logic can be blended with conventional control techniques. Fuzzy systems don't necessarily replace conventional control methods. In many cases fuzzy systems augment them and simplify their implementation.
- Fuzzy logic is based on natural language. The basis for fuzzy logic is the basis for human communication. This observation underpins many of the other statements about fuzzy logic [13].

2.6.2 Fuzzy inference systems

Fuzzy Inference System (FIS) is the process of formulating the mapping from a given input to an output using fuzzy logic. Fuzzy inference systems have been successfully applied in fields such as automatic control, data classification, decision analysis, expert systems, and computer vision. There are two types of fuzzy inference systems that can be implemented Mamdani-type and Sugeno-type. These two types of inference systems vary somewhat in the way outputs are determined.

Mamdani-style inference requires finding the centroid of a two-dimensional shape by integrating across a continuously varying function.

Michio Sugeno suggested to use a single spike, a singleton, as the membership function of the rule consequent. A singleton, or more precisely a fuzzy singleton, is a fuzzy set with a membership function that is unity at a single particular point on the universe of discourse and zero everywhere else[13].

2.6.3 Fuzzy controller

Fuzzy logic control is a control algorithm based on a linguistic control strategy, which is derived from expert knowledge into an automatic control strategy. [14]

A block diagram for a fuzzy control system is given in Figure 2-9. The fuzzy controller consists of the following four components: [14]

- i. Rule base: set of fuzzy rules of the type “if-then” which use fuzzy logic to quantify the expert's linguistic descriptions regarding how to control the plant.

- ii. Inference mechanism: emulates the expert's decision-making process by interpreting and applying existing knowledge to determine the best control to apply in a given situation.
- iii. Fuzzification interface: converts the controller inputs into fuzzy information that the inference process can easily use to activate and trigger the corresponding rules.
- iv. Defuzzification interface: converts the inference mechanism's conclusions into exact inputs for the system to be controlled.

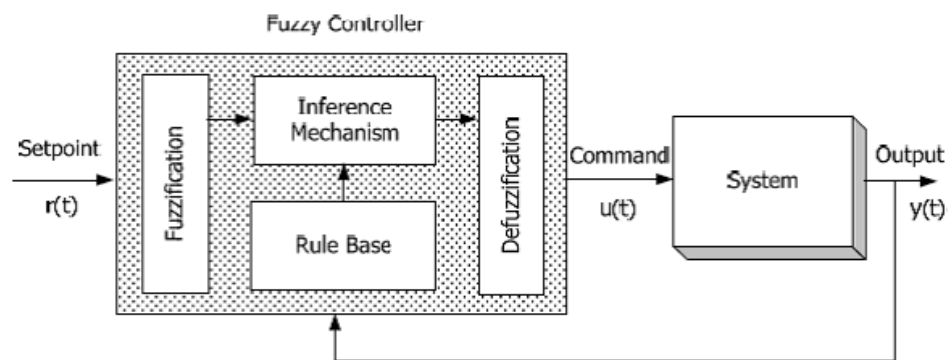


Figure 2-9: Fuzzy controller

Fuzzy logic operation principle

The operation principle of a FL controller is similar to a human operator. It performs the same actions as a human operator does by adjusting the input signal looking at only the system output as in Figure 2-10, two input signals [the main signal and its change] to the FL controller are converted to fuzzy numbers first in fuzzifier. Then they are used in the rule table to determine the fuzzy number of the compensated output signal. Finally, the resultant united fuzzy sets representing the controller output are converted to the crisp values.

The FL based controller is designed to act as an integrator controller, such that the resultant incremental output $\Delta u(k)$ is added to the previous value $u(k-1)$ to yield the current output $u(k)$. Recalling the digital solution of an integrator using Euler's integration as,

$$u(k) = u(k-1) + \Delta u(k) \quad (2-7)$$

In a digital integration, the term $\Delta u(k)$ is expressed as

$$\Delta u(k) = K_I T_s e(k) \quad (2-8)$$

Where K_I is integral constant, T_s is sampling period, and $e(k)$ is the integrated signal. The change $\Delta u(k)$ on the output of an integrator becomes zero when the input $e(k)$ is zero. Therefore output of an integrator retains the previous value. Hence, (1) can be used for both an integrator and FL controller. The difference between an integrator and FL controller is the method that is used to obtain $\Delta u(k)$, which is obtained using (2) for an integrator, and using fuzzy inference system shown in Figure 2-10 for FL controller [15].

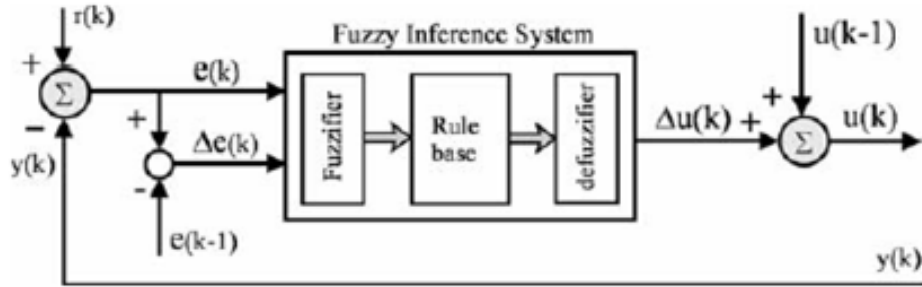


Figure 2-10: The basic structure of fuzzy logic based controller.

- **Fuzzification**

The first step in designing a fuzzy controller is to decide which state variables represent the system dynamic performance must be taken as the input signal to the controller. Fuzzy logic uses linguistic variables instead of numerical variables. The process of converting a numerical variable (real number or crisp variables) into a linguistic variable (fuzzy number) is called Fuzzification. System variables, which are usually used as the fuzzy controller inputs includes states error, state error derivative, state error integral or etc. In power system, based on previous experience, Area Control Error and its derivative ($d(e)/dt$) are chosen to be the input signals of fuzzy controller [7].

Membership Function

A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the universe of discourse, a fancy name for a simple concept [13]. The definition of the membership functions is a very delicate point in the design of the fuzzy controller (FC), because the only restriction that a membership function has to satisfy is that its values must be in the [0,1] range. A fuzzy set can therefore, unlike a crisp one, be represented by an infinite number of membership functions.

A main issue in the choice of the membership functions involved in a fuzzy controller is their number that is the cardinality of each universe of discourse of the variables. Each element of the universe of discourse must belong to a positive degree to at least one of these fuzzy sets, so that every real input in the range under study will be taken into account. A greater resolution is achieved with a high number of membership functions, on the other side a lower computational complexity is achieved when this number decreases, as with it the number of rules decreases, too.

Another point to consider is the normalization of the membership functions involved in the FC. Normalizing each membership function, that is having at least one input in which its value is one, is not a general rule, it is more a practical rule. The membership functions can be functional or numerical. A numerical assignment of the membership functions can be easily made by means of a truth table and is useful when the universe of discourse is discrete. If the membership function is functionally defined, then we have many possibilities, the most commonly used can be either piece wise linear functions or triangular (three parameters) or trapezoidal (Four parameters), or smoother functions such as the symmetric Gaussian function. Figure 2-11 shows some of membership functions.

$$\text{Gaussian } (x, [\sigma, c]) = e^{\frac{-(x-c)^2}{2\sigma^2}} \quad (\text{two parameters}) \quad (2-9)$$

The generalized Bell curve

$$\text{Bell } (x, [a, b, c]) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (\text{three parameters}) \quad (2-10)$$

Or the non-symmetric sigmoidal function

$$\text{Sigmoid}(x,[a,c]) = \frac{1}{1+e^{-a(x-c)}} \text{(two parameters)} \quad (2-11)$$

Other curves may be obtained by the combination of the above functions, by using some splines curves or some inverse tangent curves [16].

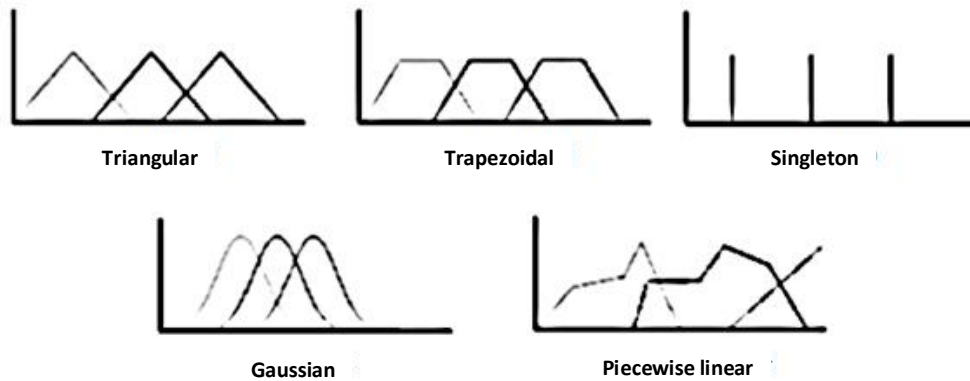


Figure 2-11: Membership functions

- **Fuzzy rule**

The fuzzy rules represent the knowledge and abilities of a human operator who makes necessary adjustments to operate the system with minimum error and fast response. In order to model the actions that a human operator would decide whether the change, Δu , in the controller output is to be increased or decreased according to the error e and its change Δe , it is necessary to observe the behaviors of the error signal e and its change Δe on different operating regions, as shown in Figure 2-12. The output Δu from the fuzzy logic (FL) controller is the change that is required to increase or decrease the overall control action to the controlled system. Therefore, the signs of e and Δe are used to determine the signs of Δu , which determines whether the overall control signal is to be increased. The sign of Δu should be positive if u is required to be increased and it should be negative otherwise [15].

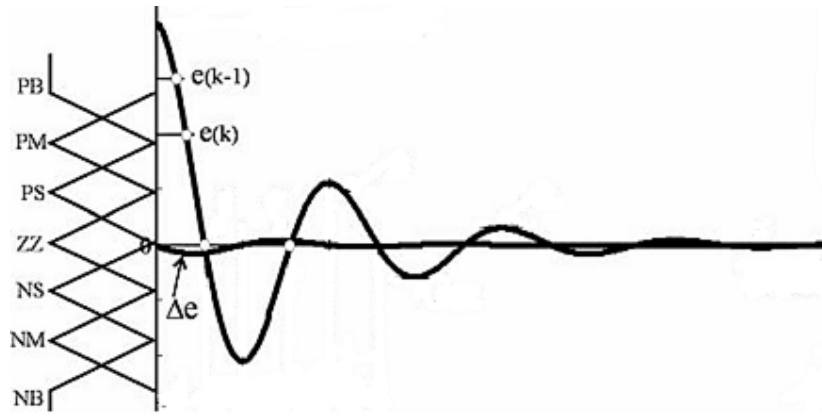


Figure 2-12: The time responses of error and error change for a generalized second order system.

Inference Engine

For each rule, the inference engine looks up the membership values in the condition of the rule.

- ❖ **Aggregation** the aggregation operation is used when calculating the degree of fulfillment or firing strength α_k of the condition of a rule k . A rule, say rule 1, will generate a fuzzy membership value μ_{e1} coming from the error and a membership value μ_{ce1} coming from the change in error measurement. The aggregation is their combination,

$$\mu_{e1} \text{ and } \mu_{ce1} \quad (2-12)$$

Aggregation is equivalent to fuzzification, when there is only one input to the controller, Aggregation is sometimes also called fulfillment of the rule or firing strength.

- ❖ **Activation** the activation of a rule is the deduction of the conclusion, possibly reduced by its firing strength. Min or product (*) is used as the activation operator. Both methods work well in general, although the multiplication results in a slightly smoother control signal.
- ❖ **Accumulation** all activated conclusions are accumulated, using the max operation; alternatively, sum accumulation counts overlapping areas more than once [17].

Defuzzification

The resultant membership values of the active rules determine the weights of the fuzzy sets in the universe of Δu as shown in Figure 2-13 with the shaded parts of the fuzzy sets. Then the averaged value of the union of these shaded fuzzy sets is used to obtain final crisp output as $\Delta u(k)$. This final process is called defuzzification of the fuzzy output[15]. There is several defuzzification methods as:[17]

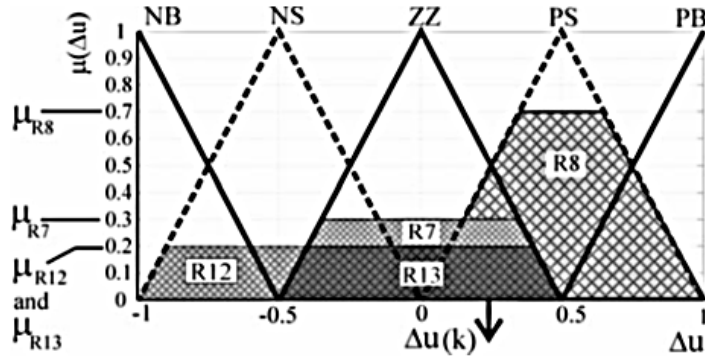


Figure 2-13: Membership functions used to represent fuzzy partitioning

❖ Centre of Gravity

The crisp output value u is the abscissa under the Centre of gravity of the fuzzy set,

$$u = \frac{\sum_i \mu(x_i)x_i}{\sum_i \mu(x_i)} \quad (2-13)$$

Here x_i is a running point in a discrete universe, and $\mu(x_i)$ is its membership value in the membership function. The expression can be interpreted as the weighted average of the elements in the support set. For the continuous case, replace the summations by integrals.

It is a much used method although its computational complexity is relatively high. This method is also called centroid of area.

❖ Centre Of Gravity method for Singletons

If the membership functions of the conclusions are singletons the output value is

$$u = \frac{\sum_i \mu(s_i)s_i}{\sum_i \mu(s_i)} \quad (2-14)$$

Here s_i is the position of singleton i in the universe, and $\mu(s_i)$ is equal to the firing strength

α_i of rule i . This method has a relatively good computational complexity, and u is differentiable with respect to the singletons s_i , which is useful in neurofuzzy systems.

❖ Bisector Of Area

This method picks the abscissa of the vertical line that divides the area under the curve in two equal halves. In the continuous case,

$$u = \left\{ \int x \mid \int_{Min}^x \mu(x) dx = \int_x^{Max} \mu(x) dx \right\} \quad (2-15)$$

Here x is the running point in the universe, $\mu(x)$ is its membership, Min is the leftmost value of the universe, and Max is the rightmost value. Its computational complexity is relatively high, and it can be ambiguous. For example, if the fuzzy set consists of two singletons any point between the two would divide the area in two halves; consequently it is safer to say that in the discrete case, BOA is not defined.

❖ Mean of Maxima

An intuitive approach is to choose the point with the strongest possibility, i.e. maximal membership. It may happen, though, that several such points exist, and a common practice is to take the Mean Of Maxima (MOM). This method disregards the shape of the fuzzy set, but the computational complexity is relatively good.

❖ Leftmost Maximum and Rightmost Maximum

Another possibility is to choose the Leftmost Maximum (LM), or the Rightmost Maximum (RM). In the case of a robot, for instance, it must choose between left or right to avoid an obstacle in front of it. The defuzzifier must then choose one or the other, not something in between. These methods are indifferent to the shape of the fuzzy set, but the computational complexity is relatively small.

2.6.4 Process of developing a fuzzy expert system

- i. Specify the problem and define linguistic variables.
- ii. Determine fuzzy sets.
- iii. Elicit and construct fuzzy rules.
- iv. Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference into the expert system, one may choose one of two options:

- Build the system using a programming language such as C/C++ or Pascal, or
- Apply a fuzzy logic development tool such as MATLAB Fuzzy Logic Toolbox
 - v. Evaluate and tune the system, the last, and the most laborious, task is to evaluate and tune the system. We want to see whether the fuzzy system meets the requirements [18].

2.6.5 Tuning fuzzy systems

- i. Review model input and output variables, and if required redefine their ranges.
- ii. Review the fuzzy sets, and if required define additional sets on the universe of discourse. The use of wide fuzzy sets may cause the fuzzy system to perform roughly.
- iii. Provide sufficient overlap between neighboring sets. It is suggested that triangle-to-triangle and trapezoid-to-triangle fuzzy sets should overlap between 25% to 50% of their bases.
- iv. Review the existing rules, and if required add new rules to the rule base.
- v. Examine the rule base for opportunities to write hedge rules to capture the pathological behavior of the system.
- vi. Adjust the rule execution weights. Most fuzzy logic tools allow control of the importance of rules by changing a weight multiplier.
- vii. Revise shapes of the fuzzy sets. In most cases, fuzzy systems are highly tolerant of a shape approximation [18].

2.7 Loads

Direct-current motors transform electrical energy into mechanical energy. They drive devices such as hoists, fans, pumps, calendars, punch-presses, and cars. The torque-speed characteristic of the motor must be adapted to the type of the load it has to drive [19].

2.7.1 Load types

There are different types of load as:

i. Constant torque load

The torque demanded by the load is constant throughout the speed range. Loads of these types are essentially friction loads. The figure (2-14) below

shows the constant torque and its effect on horsepower demanded by the load.

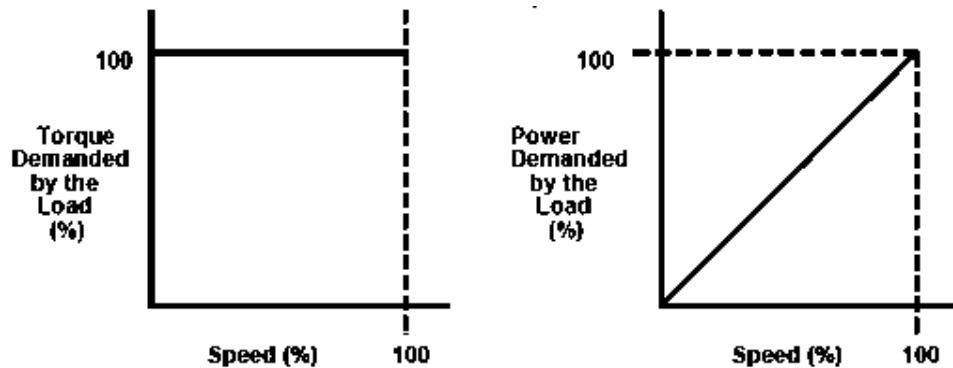


Figure (2-14): Constant Torque Load

Since Horse Power HP is a product of torque times speed, and torque remains constant in this type of load, horsepower is a function of speed.

$$HP = \frac{Torque * Speed}{5252} \text{ OR watts} = \frac{Torque * Speed}{9.55} \quad (2-16)$$

Where:

Torque = lb-ft.

Torque = Nm

Speed = RPM

Speed = RPM

5252 = a proportionality constant

9.55 = a proportionality constant

Examples of this type of load are conveyors and extruders. Constant torque is also used when shock loads, overloads or high inertia loads are encountered.

❖ Variable torque load

With this type of load, the torque demand increases with speed as shown in Figure 2.15, usually speed squared (Speed²), Horsepower is typically proportional to speed cubed (Speed³).

Examples of loads that exhibit variable load torque characteristics are centrifugal fans, pumps and blowers. This type of load requires much lower torque at low speeds than at high speeds.

❖ Constant horsepower operation

This is a function of the motor being operated above base motor speed. The horsepower demanded by the load is constant within the speed range. The speed and torque are inversely proportional to each other as shown in figure 2-16

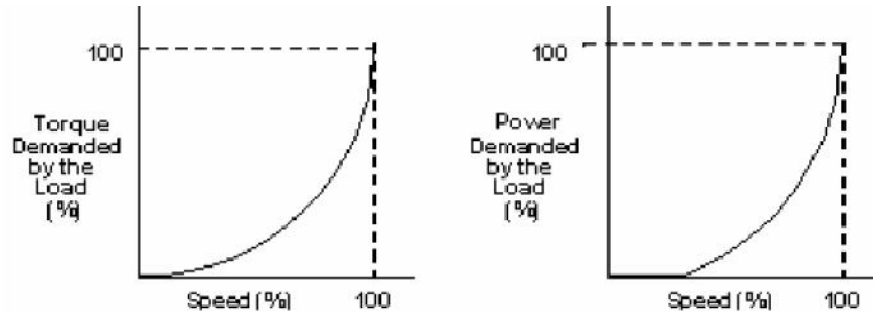


Figure (2-15): Variable Torque Load

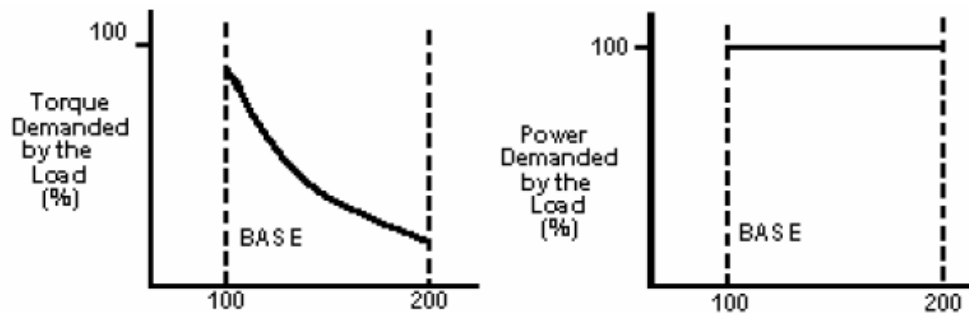


Figure (2-16): Constant Horsepower Operation

2-7-2 Overload Operation

Many constant torque applications require intermittent operation in overload. Acceleration and deceleration torque requirements may be severe to meet a machine “cycle time” [20].

CHAPTER THREE

SYSTEM CONTROLLER DESIGN

3.1 Introduction

The development of high performance motor drives is very important for industrial as well as other purpose applications. Generally, a high performance motor drive system must have good dynamic speed command tracking and load regulating response. Many varieties of control schemes such as Proportional (P), Proportional Integral (PI), Proportional Derivative Integral (PID), and Fuzzy Logic Controller (FLCs), have been developed for speed control of DC motors [7].

DC Motor control has been used for variable speed and position applications for many decades and historically were the first choices for speed control applications requiring accurate speed control, controllable torque, reliability and simplicity. The basic principle of a DC variable speed drive is that the speed of a separately excited DC motor is directly proportional to the voltage applied to the armature of the DC motor.

To optimize the performance of DC Motor it often requires to design and engineering effort. The solution to this challenge is greatly simplified by using practical modeling methods. Modeling is a finite element analysis has been successfully applied to model rotating electrical machines with complicated geometry and magnetic saturation. In control systems, modeling is a method to convert the system into the appropriate diagram that can be used for analysis.

3.2 Separately Excited DC Motor Modeling

In modeling, the aim is to find the governing differential equations that relate the applied voltage to the produced torque or speed of the rotor [21]. Figure 3-1 shows the equivalent circuit with armature voltage control and the model of a general mechanical system that incorporates the mechanical parameters of the motor and the mechanism coupled to it.

Armature reactions effects are ignored in the description of the motor. The fixed voltage V_f is applied to the field and the field current settles down to a constant

value. A linear model of a simple DC motor consists of a mechanical equation and electrical equation [22].

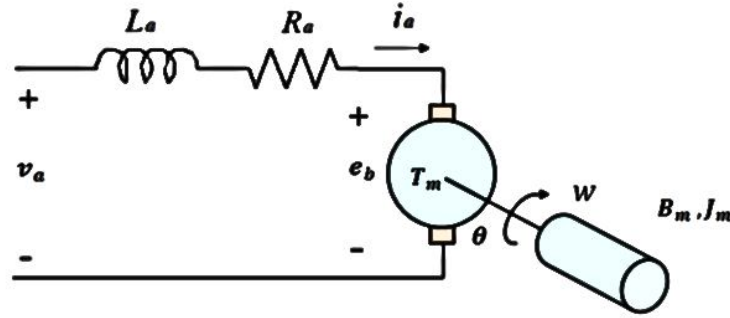


Figure (3- 1): Equivalent circuit of a separately excited DC Motor

From Figure 3-1, Kirchhoff's Voltage Law (KVL) is applied to the electrical circuit. These can be written

$$v_a(t) = R_a \cdot i_a(t) + L_a \cdot \frac{di_a(t)}{dt} + e_b \quad (3-1)$$

Where: v_a is armature voltage (V), i_a is armature current (A), R_a is armature resistance (Ω), L_a is armature inductance (H), e_b is back EMF (V). Setting $e_b(t)$ in (3-1) equals to $K_b \cdot w(t)$ the electrical equation in (3-1) becomes

$$v_a(t) = R_a \cdot i_a(t) + L_a \cdot \frac{di_a(t)}{dt} + K_b \cdot w(t) \quad (2-1)$$

For normal operation, the developed torque must be equal to the load torque plus the friction and inertia.

$$T_m(t) = J_m \cdot \frac{dw(t)}{dt} + B_m \cdot w(t) + T_L \quad (3-2)$$

Where: T_m is motor torque (Nm), J_m is rotor inertia (kgm^2), W is angular speed (rad/s), B_m is viscous friction coefficient (Nms/rad), T_L is load torque (Nm). Setting $T_m(t)$ equal to $K_T \cdot i_a(t)$ and $T_L = 0$ yields

$$K_T \cdot i_a(t) = J_m \cdot \frac{dw(t)}{dt} + B_m \cdot w(t) \quad (2-2)$$

Taking the Laplace transforms for (2-1) and (2-2) yields

$$V_a(s) = R_a \cdot I_a(s) + L_a \cdot I_a(s) \cdot s + K_b \cdot W(s) \quad (3-3)$$

$$K_T \cdot I_a(s) = J_m \cdot W(s) \cdot s + B_m \cdot W(s) \quad (3-4)$$

Current obtained from (3-4) as

$$I_a(s) = \frac{J_m \cdot W(s) \cdot s + B_m \cdot W(s)}{K_T} \quad (3-5)$$

And then substituted in (3-3) get

$$V_a(s) = \frac{W(s)}{K_T} [(R_a \cdot J_m \cdot s + R_a \cdot B_m) + (L_a \cdot J_m \cdot s^2 + L_a \cdot B_m \cdot s) + K_b \cdot K_T] \quad (3-6)$$

So the relationship between rotor shaft speed and applied armature voltage is represented by the transfer function:

$$\frac{w(s)}{V_a(s)} = \frac{K_T}{L_a \cdot J_m \cdot s^2 + (R_a \cdot J_m + L_a \cdot B_m) \cdot s + (R_a \cdot B_m + K_b \cdot K_T)} \quad (3-7)$$

The relation between the position and speed is:

$$\theta(s) = \frac{1}{s} W(s) \quad (3-8)$$

Then the transfers function between shaft position and armature voltage at no-load is:

$$\frac{\theta(s)}{V_a(s)} = \frac{K_T}{L_a \cdot J_m \cdot s^3 + (R_a \cdot J_m + L_a \cdot B_m) \cdot s^2 + (K_T \cdot K_b + R_a \cdot B_m) \cdot s} \quad (3-9)$$

By breaking down the transfer function between speed and armature voltage will get a simple transfer functions that can be connected with each other as shown in Figure (3-2)[8].

The effect of load on the separated excited DC motor is appears in the equation (3-10).

$$T_m(t) = J_m \cdot \frac{dw(t)}{dt} + B_m \cdot w(t) + T_L \quad (3-10)$$

From this can find that

$$w(s) = (T_m - T_L) \cdot \frac{1}{(J_m s + B_m)} \quad (3-11)$$

So it can be enter the load to the block diagram of separately excited DC motor as shown in figure (3-3).

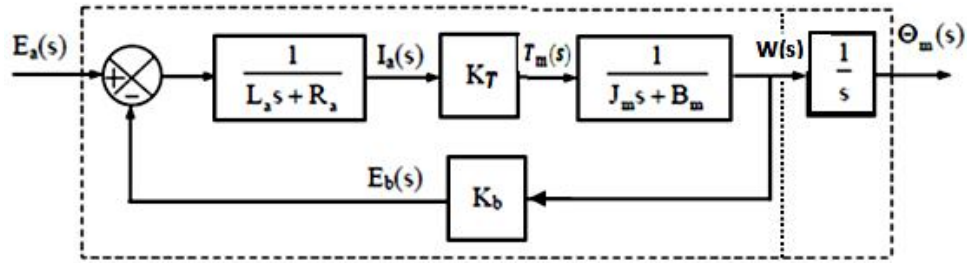


Figure (3-2): Block diagram of separately excited DC motor

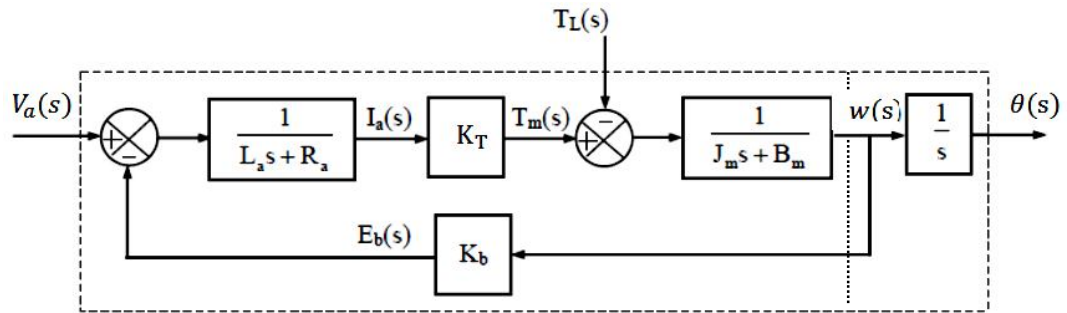


Figure (3-3): Block diagram of separately excited DC motor with load

❖ Specification of the separately excited DC motor

Separately excited DC motor with the below parameters was used[8].

$$R_a = 11.2 \, \Omega$$

$$L_a = 0.1215 \, \text{H}$$

$$J_m = 0.02215 \, \text{kgm}^2$$

$$B_m = 0.002953 \, \text{Nms/rad}$$

$$K_i = 1.28 \, \text{Nm/A}$$

$$K_b = 1.28 \, \text{Vs/rad}$$

Load torque is opposed by a load that assumed in this experiment to be proportional to speed as:

$$\text{No-load} = 0$$

Half of load (50%) = $0.5 \times \text{speed}$

Full load = speed

3.3 Controller Design for Motor Speed Control System

For system controller design, it is important to use MATLAB which is a technical computing environment (program) designed for numerical computation and visualization. Over the years, it has become a standard tool in many universities and research organizations for performing mathematical calculations; MATLAB has developed into a powerful tool to simplify mathematical analysis [23].

3.3.1 The input and output for the system

First, it must identify the variables to be controlled. For the speed control of separated excited DC motor it can control the amount of energy (v) supplied by the armature voltage as it will be the input for the separately excited DC motor, as load is input also, and speed will be the output.

3.3.2 PID controller design

PID controllers are widely used in industrial control applications due to their simple structures, comprehensible control algorithms and low costs[8].

The block used in the modeling of the PID controller is related to the transfer function for speed of the separated excited DC motor is. As shown in Figure (3-4), the difference between the desired speed as reference input and the output feedback which is the speed of the motor is passed as input into the PID controller. PID controller can be considered as subsystem as shown in Figure (3-5) which contains the proportional gain scaling factor (K_P), the derivative gain scaling factor (K_D) and the integral gain scaling factor (K_I).

The derivative gain factor and the integral gain factor are both passed through a derivative block and an integral block, respectively before being summed up with the proportional gain factor. The output of the PID controller subsystem serves as an input to the DC motor block which is simply the transfer function of the DC motor. Also should be taken into consideration the load change which is have an obvious effect on the output signal.

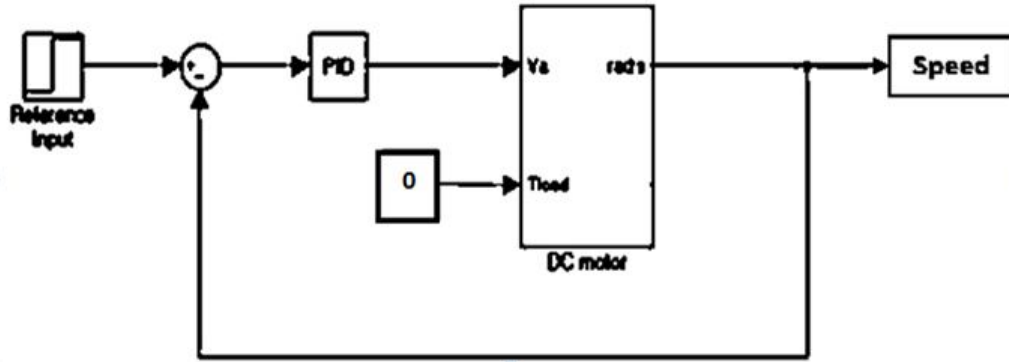


Figure (3-4): Model of the PID controller

The PID controller model in this topic was hand-tuned by first increasing the value of the proportional gain K_P until the desirable response is obtained. The derivative gain K_D and the integral gain K_I are then adjusted to improve and optimize the response of the system with consider to the load change by trying for different level of the load.

Performance criteria were specified for a full load as:

Rise time (t_r) < 2s

Settling time (t_s) < 11s

Maximum overshoot (M_p) < 12%

Steady state error (e) < 1%

A fairly optimal response is achieved for a proportional gain value of 19, a derivative gain value of 2.3 and an integral gain value of 11.

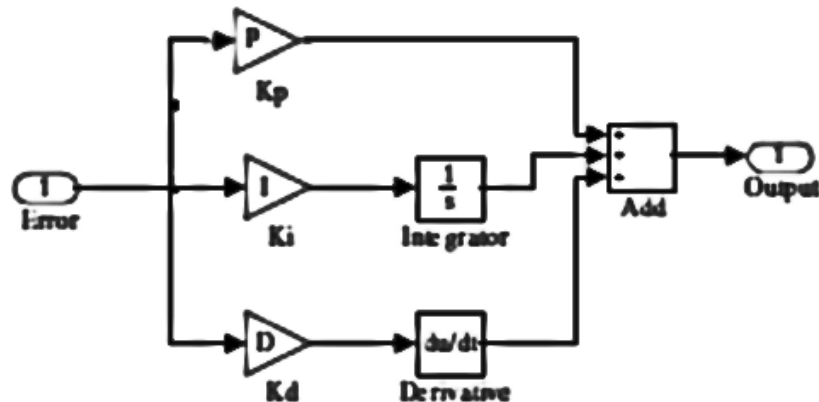


Figure (3-5): PID controller

3.3.3 Design of fuzzy controller

In the design of fuzzy controller the fuzzy logic toolbox is highly impressive in all respect. It makes fuzzy logic an effective tool for the conception and design of intelligent systems. The fuzzy logic toolbox is easy to master and convenient to use. And last, but not least important, it provides a reader-friendly and up-to-date introduction to the methodology of fuzzy logic and its wide-ranging applications [13].

i. Input / Output for Fuzzy Controller

The control function (which is the output from fuzzy controller) will carry out from the error and the variation in this error (which are the inputs for fuzzy controller). Figure (3-6) show fuzzy controller for a DC motor system.

$$e(t) = r(t) - y(t) \quad (2-3)$$

Once the fuzzy controller's inputs and outputs are selected, the next step is to determine the reference input desired, which in this case will be $r=10$ (unit step input of ten).

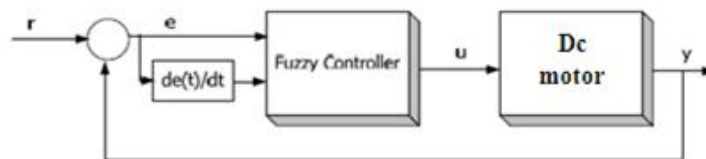


Figure (3-6): Fuzzy controller for a DC motor system

ii. Fuzzy logic algorithm explained

The process of fuzzy logic is explained in algorithm 1: Firstly, a crisp set of input data is gathered and converted to a fuzzy set using fuzzy linguistic variables, fuzzy linguistic terms and membership functions. This step is known as fuzzification. Afterwards, an inference is made based on a set of rules. Lastly, the resulting fuzzy output is mapped to a crisp output using the membership functions, in the defuzzification step [8].

iii. Fuzzification

The process of generating membership values for a fuzzy variable using membership functions [13].

❖ Linguistic Variables, Linguistic Terms

The linguistic variables use to describe the time-varying inputs and outputs of the fuzzy controller. A linguistic variable is generally decomposed into a set of linguistic terms such as “large positive and medium negative”, each member of this decomposition is called a linguistic term and can cover a portion of the overall values of the linguistic variable [25].

In this topic of separated excited DC motor speed control the linguistic variables will be the error “difference between the output speed and the set point” and error variation as inputs to the fuzzy controller and control function as the output which it will be the armature voltage, each of them decomposed to the following linguistic Terms:

1. LN Large Negative
2. MN Medium Negative
3. SN Small Negative
4. ZE Zero
5. SP Small Positive
6. MP Medium Positive
7. LP Large Positive

The Graphical User Interface (GUI) is tools provided by the Fuzzy Logic Toolbox[13]. It uses to build a controller system. The window shown in Figure 3-7 is the Fuzzy Inference System (FIS) Editor, which used to create the inputs and outputs for the fuzzy controller. In this Thesis It two inputs (error and change-error) /one output system (control action)are built, these were specified by the membership function editor.

❖ Membership function

Membership functions are used in the fuzzification and defuzzification steps of an FLS, to map the non-fuzzy input values to fuzzy linguistic terms and vice versa. A membership function is used to quantify a linguistic term [25]. There are different

membership functions associated with each input and output response. The most common shape of membership functions is triangular, although trapezoidal and bell curves are also used [7].

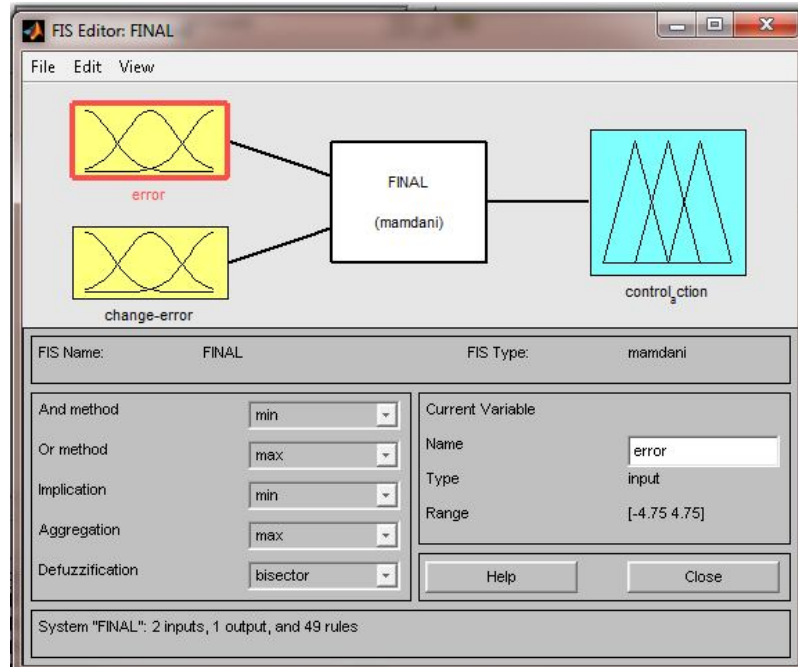


Figure (3-7): The Fuzzy Inference System Editor in the Fuzzy Logic Toolbox

The Membership Function Editor as shown in Figure (3-8), Figure (3-9) and Figure (3-10) are used to select the appropriate membership functions for the inputs and output variables. The main issue in the choice of the membership functions involved in a fuzzy controller is their number that is each element of the universe of discourse must belong to a positive degree to at least one of the fuzzy sets, so that every real input in the range under study will be taken into account. A greater resolution is achieved with a high number of membership functions, on the other side a lower computational complexity is achieved when this number decreases, as with it the number of rules decreases too[26].

Actually from three to seven curves are generally appropriate to cover the required range of an input value [7]. Here seven membership functions were chosen with gaussmf shape, in the range of -4.75 to 4.75 for input (error), and -1.65 to 1.65 for (change in error) as shown in Figure (3-8), (3-9). And seven membership functions

with trimf shape in the range of 470.5 to 470.5 for the output as shown in Figure (3-10).

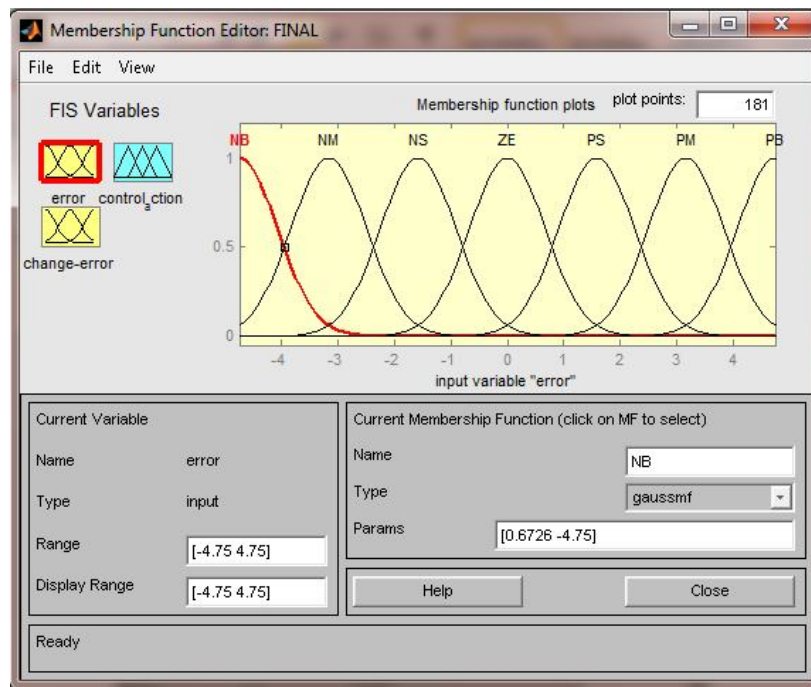


Figure (3-8): Membership function editor (error)

iv. Fuzzy rules

In an FLS, a rule base is constructed to control the output variable. A fuzzy rule is a simple IF-THEN rule with a condition and a conclusion[25]. The linguistic quantifiers defined earlier were used to craft rules that capture the expert's knowledge regarding how to control the system. , it can define every possible motor speed control situation. Since it used a finite number of linguistic variables and values, there are a finite number of possible rules. For the motor speed control in this topic, there are two input linguistic variables and seven linguistic values, there are $7^2 = 49$ possible rules (every possible combination of the values of the linguistic variables).

A convenient way of representing the set of rules when the number of inputs to the fuzzy controller is low (three or fewer) is by using a table. For the separated excited motor speed control table (3-1) shows the rule base. Each square represents the linguistic value of the consequent of a rule, with the left column and the top row containing the linguistic values of the antecedent variables.

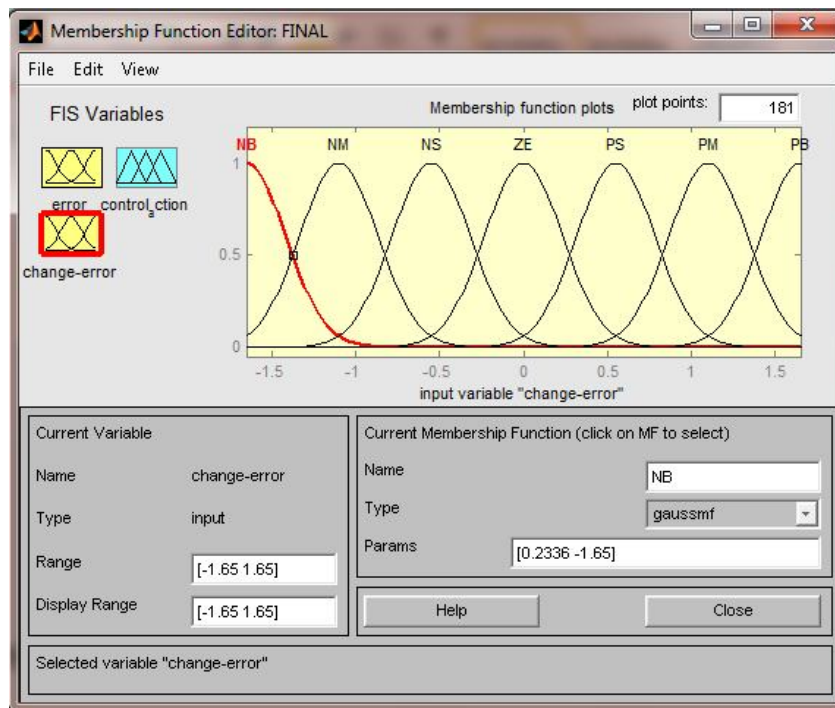


Figure (3-9): Membership function editor (change in error)

In fuzzy controller design, constructing rules using the graphical rule editor Interface is fairly Self-evident, based on the descriptions of the input and output variables defined by the FIS Editor [13]. Figure 3-11 shows the graphical rule editor for the fuzzy controller that used to control the system.

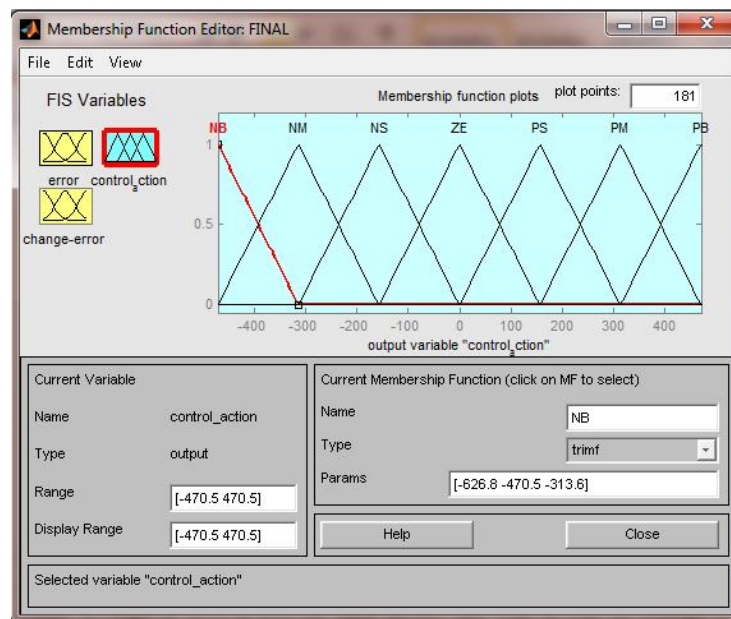


Figure (3-10): Membership function editor for output

Table (3-1): Rule base for separated excited DC motor speed control

GE E	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

v. Fuzzy set operations

The evaluations of the fuzzy rules and the combination of the results of the individual rules are performed using fuzzy set operations, there are different Fuzzy set operations, and the mostly-used operations are max and min.

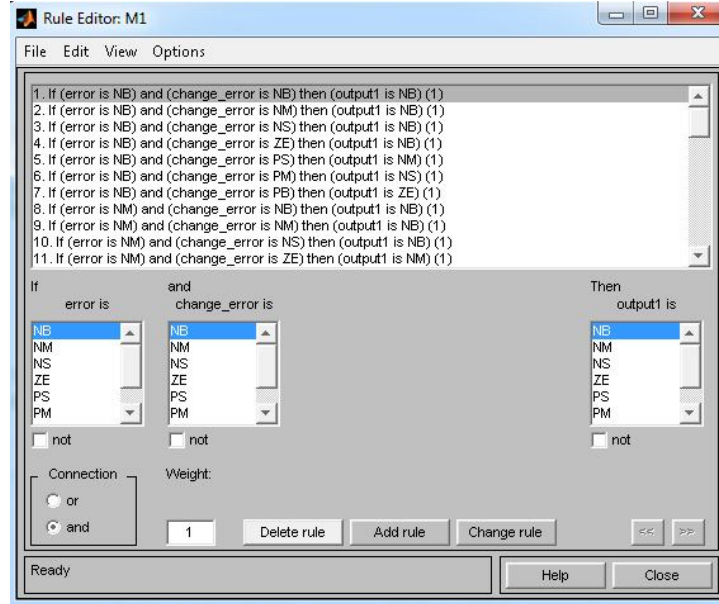


Figure (3-11): Graphical rule editor in the fuzzy logic toolbox

i. Inference

After evaluating the result of each rule, these results should be combined to obtain a final result. This process is called inference. The results of individual rules can be combined in different ways.

The maximum algorithm is generally used for this purpose [25].

In fuzzy controller design The Rule Viewer displays a roadmap of the whole fuzzy inference process [13].

Figure (3-12) shows the rule viewer for the fuzzy controller of the motor speed control system.

ii. Defuzzification

After the inference step, the input to the defuzzification process is a fuzzy set. The aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set [13].

This is the purpose of the defuzzifier component of an FLS. Defuzzification is performed according to the membership function of the output variable. There are different algorithms for defuzzification. For this topic as shown in figure 6, was worked on Mamdani-style inference and chose bisector as defuzzification algorithms. The Surface Viewer is a GUI tool that lets you examine the output surface of a FIS, for any one or two inputs, since it does not alter the fuzzy system [13]. Figure 3-13 shows the surface viewer for the fuzzy controller that builds to control the speed of the motor.

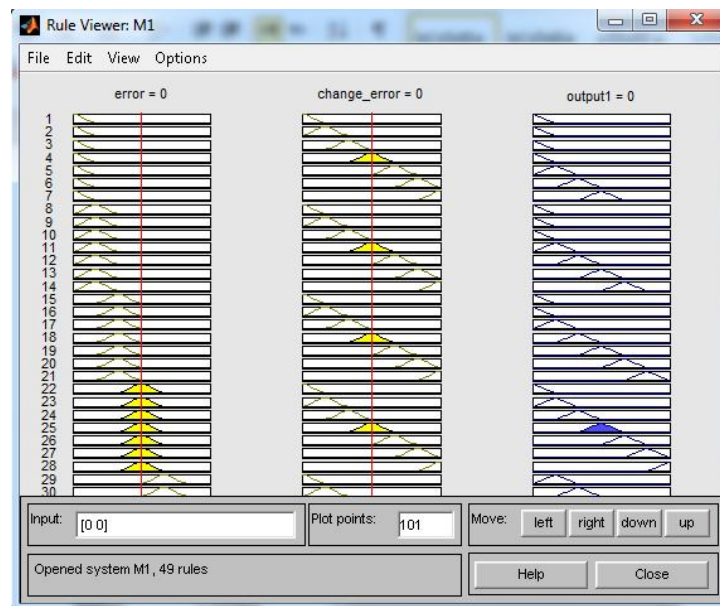


Figure (3-12): Rule viewer in the fuzzy logic toolbox

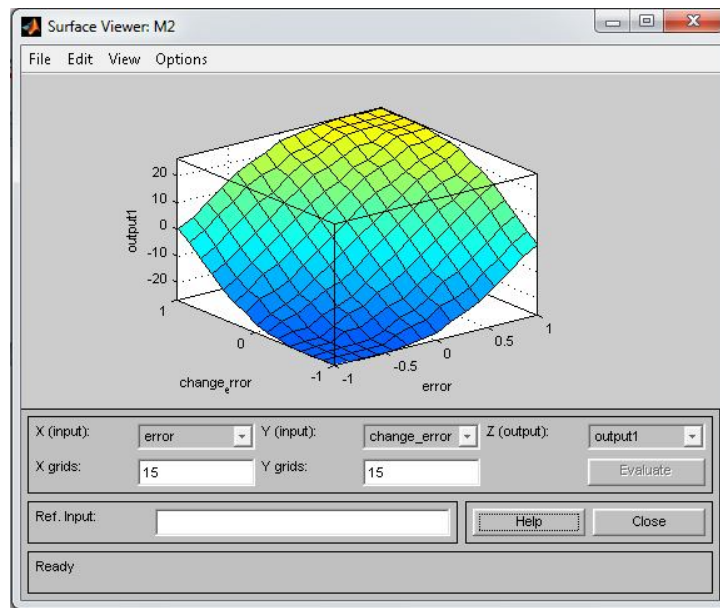


Figure (3-13): Surface viewer in the Fuzzy Logic Toolbox

CHAPTER FOUR

SIMULATION RESULTS AND DISCUSSIONS

4.1 Introduction

Simulation used to represent the dynamic equations of a system may show the actual physical variables that appear in the system, or it may show variables that are used purely for mathematical convenience. In either case the overall response of the system is the same [27].

4.2 Motor Simulation Results without Controller

To get output required or good response it is helpful to use MATLAB capabilities. Here MATLAB simulation was used to get the output time response of the separately excited DC motor in different cases. In this research the unit step was used as input signal to the system with value equal to 10, and scope to illustrate the output time response. First, system is simulated in MATLAB software without control and examine the output signal of the system in different values of load. Figure (4-1) shows the simulink diagram of separately excited DC motor without control.

To simplify the simulation it is better to build this diagram in subsystem as shown in Figure (4-2).

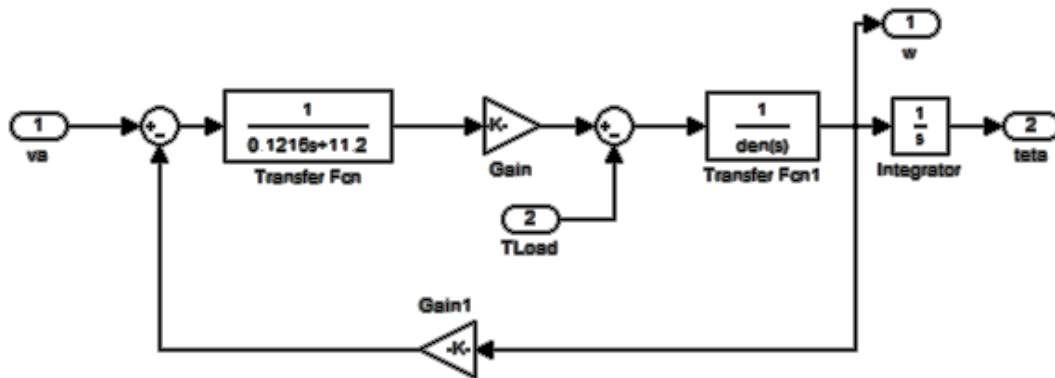


Figure (4-1): Separately excited DC motor simulink diagram

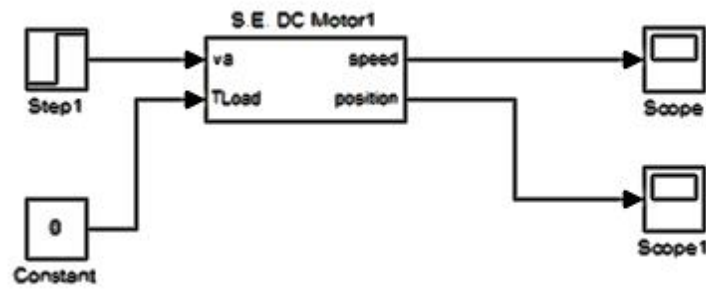


Figure (4-2): Simulink diagram of the system at no-load without controller

4.2.1 Result for motor response at no-load

The output response of the system to step input (of 10) without control in case of (no-load) is illustrated in Figure (4-3).

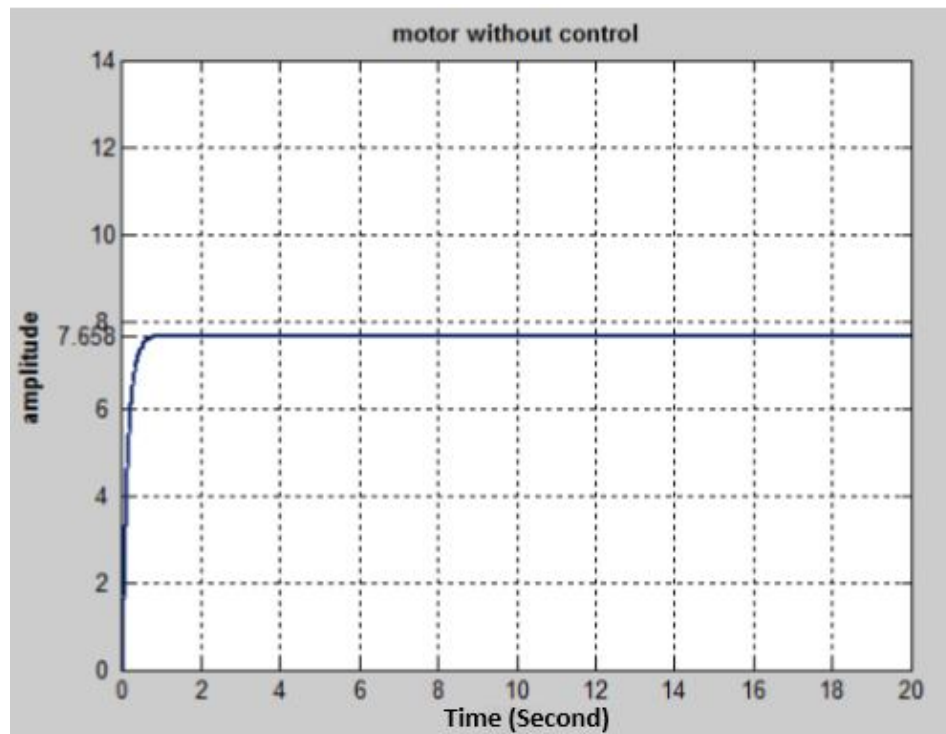


Figure (4-3): Unit step response of the system at no-load

4.4.2 Result of motor response at 75% of load

For 75% of load the Simulink diagram of the separately excited DC motor is illustrated in Figure (4-4), and the output response is shown in Figure (4-5).

4.4.3 Result of motor response at Full Load

In case of full load the output illustrated in Figure (4-6).

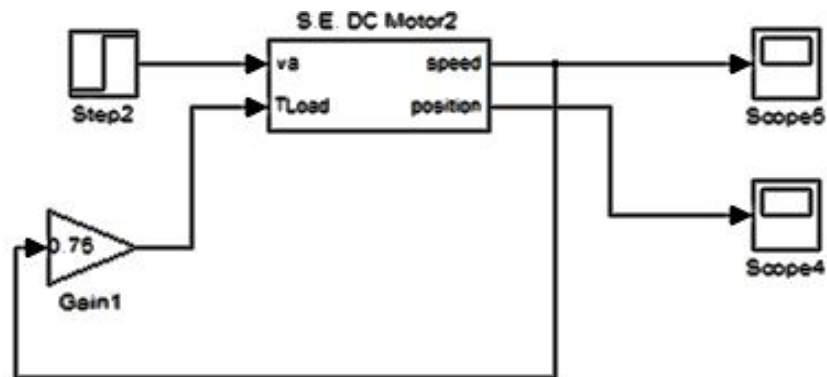


Figure (4-4): Simulink diagram of the system at 75% of load without controller

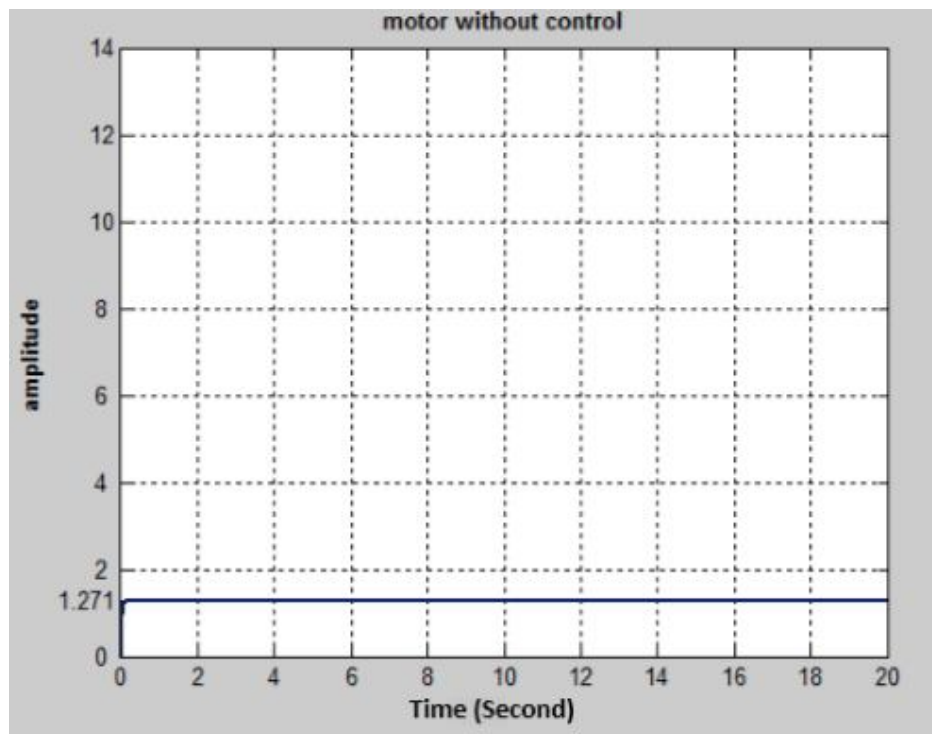


Figure (4-5): Unit step response of the system at 75% of load without controller

4.3 System Simulation Results with PID Controller

The PID controller parameters were tuned by using trial and error method to control the DC motor, the parameters values are: $k_p=19$ $k_i=11$ $k_d=2.3$.

4.3.1 System result with PID controller at no-Load

The system simulink diagram with PID controller (in case of no-load) is shown in the Figure (4-7), and the output response is shown in Figure (4-8).

4.3.2 System result with PID controller at 75% of Load

In case of 75% of load the system output response is shown in Figure (4-9).

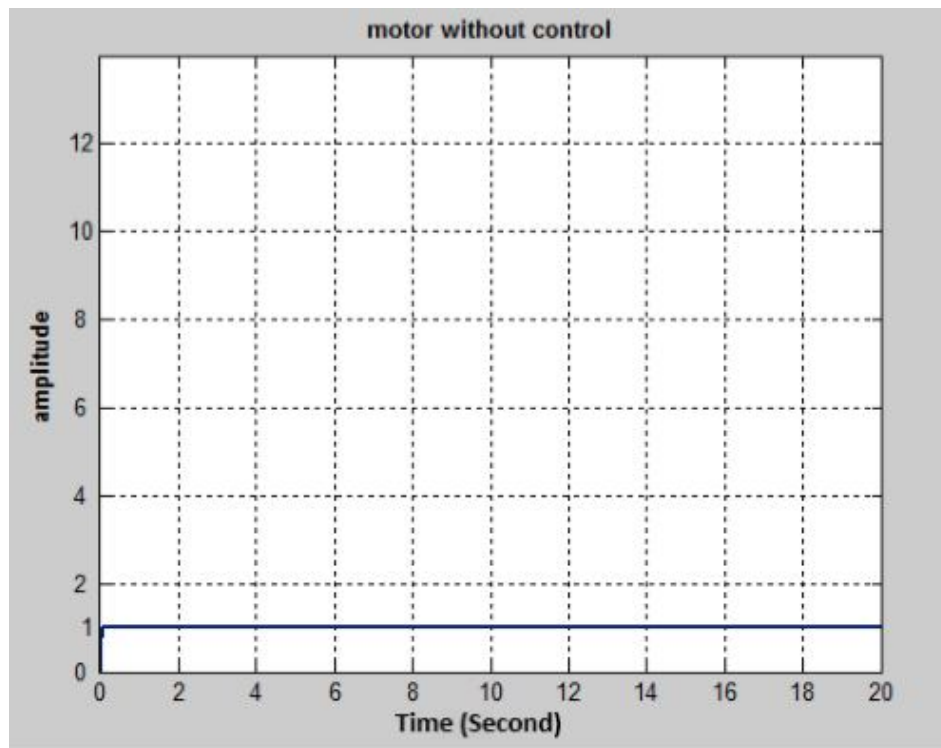


Figure (4-6): Unit step response of the system at full load without controller

4.3.3 System result with PID controller at full load

In case of full load the system output response is shown in Figure (4-10).

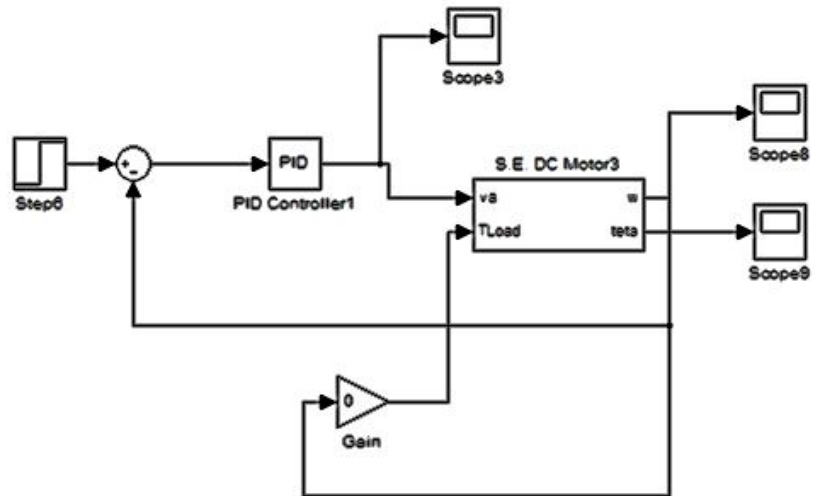


Figure (4-7): Simulink diagram of the system at no-load with PID controller

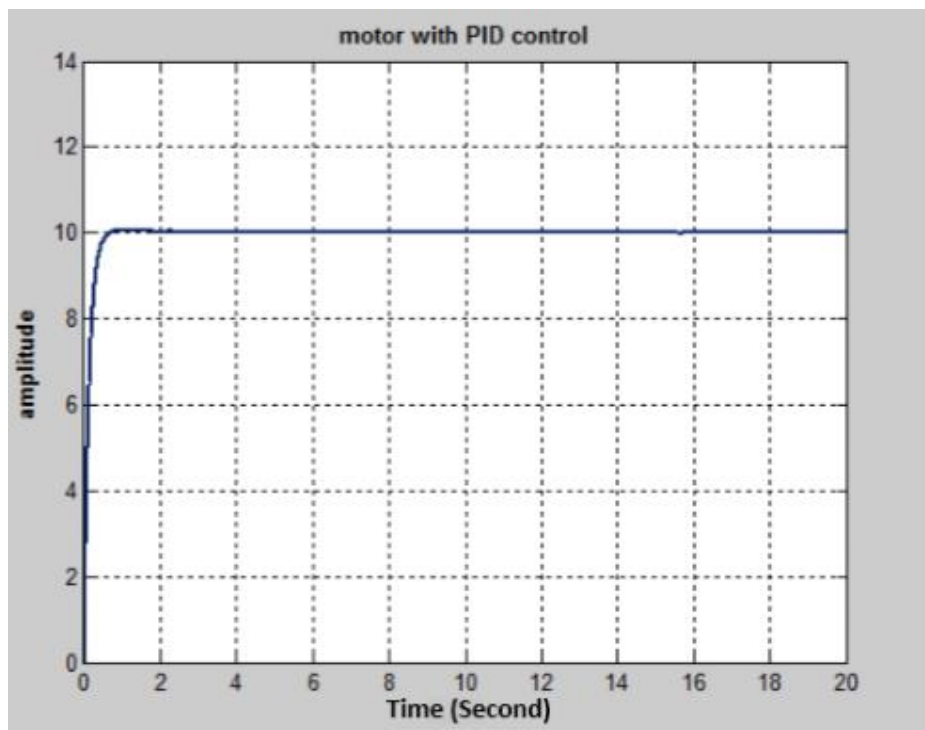


Figure (4-8): Unit step response of the system at no-load with PID controller

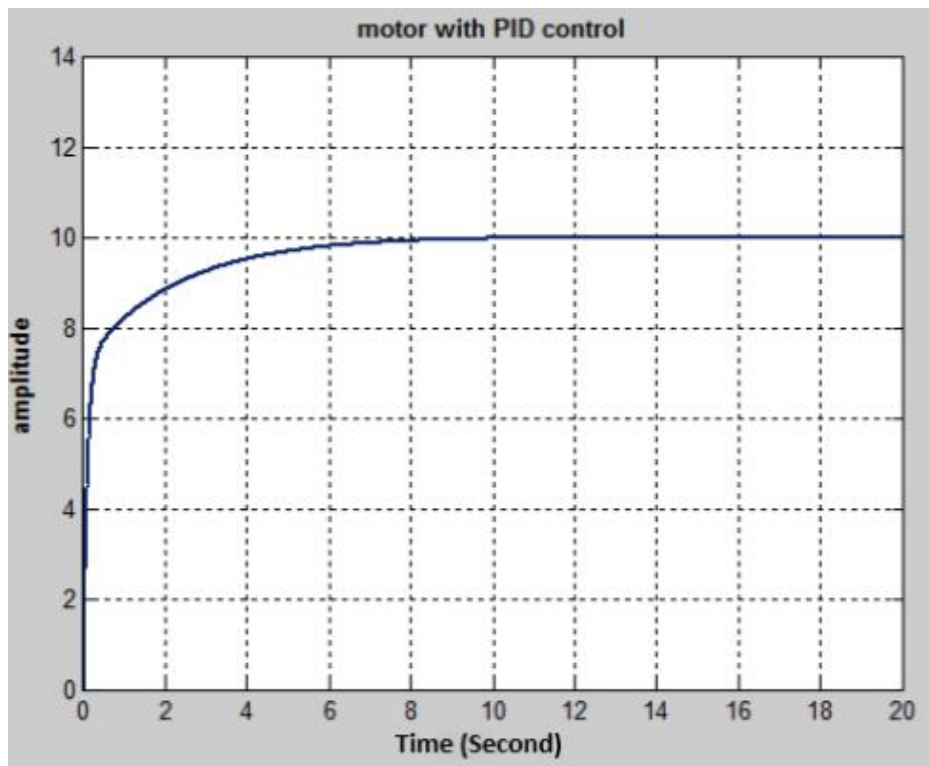


Figure (4-9): Unit step response of the system at 75% of load with PID controller

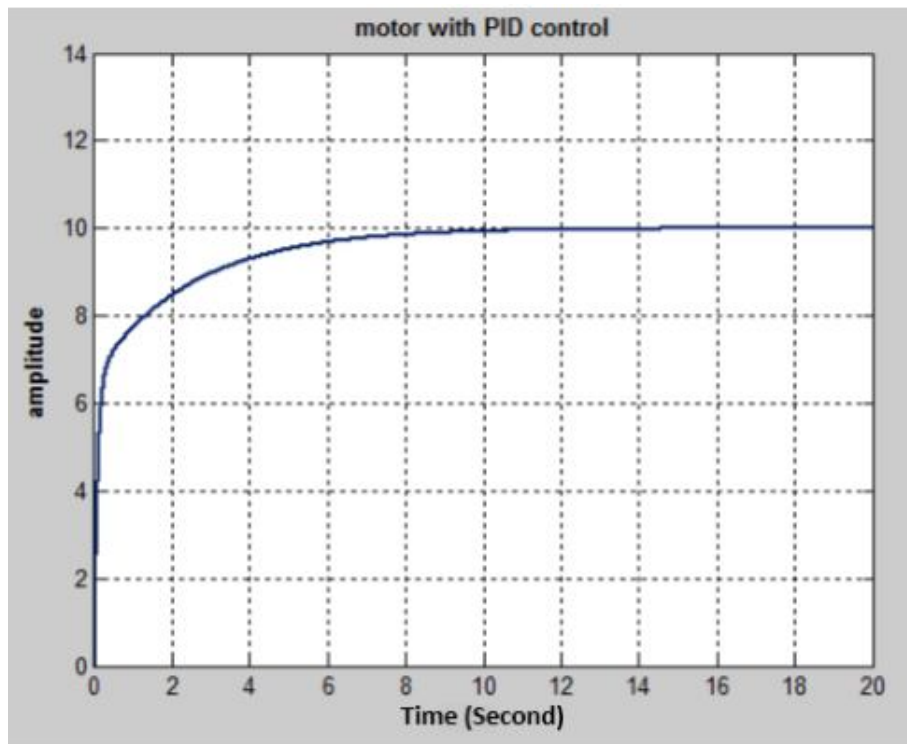


Figure (4-10): Unit step response of the system at full load with PID controller

4.4 System Simulation Result with Fuzzy Controller

To control separately excited DC motor with fuzzy controller the parameters are tuned using trial and error method. The parameter values are: $k_1=0.32$, $k_2=10$, $k_3=0.08$, $k_4=10$, $k_5=1.5$. Figure (4-11) shows simulink diagram for system with fuzzy controller.

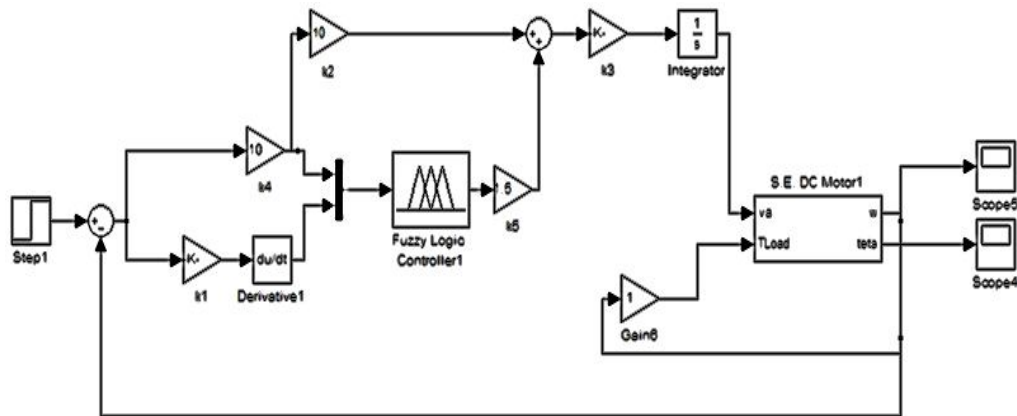


Figure (4-11): Simulink diagram of the system at no-load with fuzzy controller

4.4.1 System result with PID controller at no-load

In case of no-load (gain6 set to 0) the unit step response of motor with fuzzy control is shown in Figure (4-12).

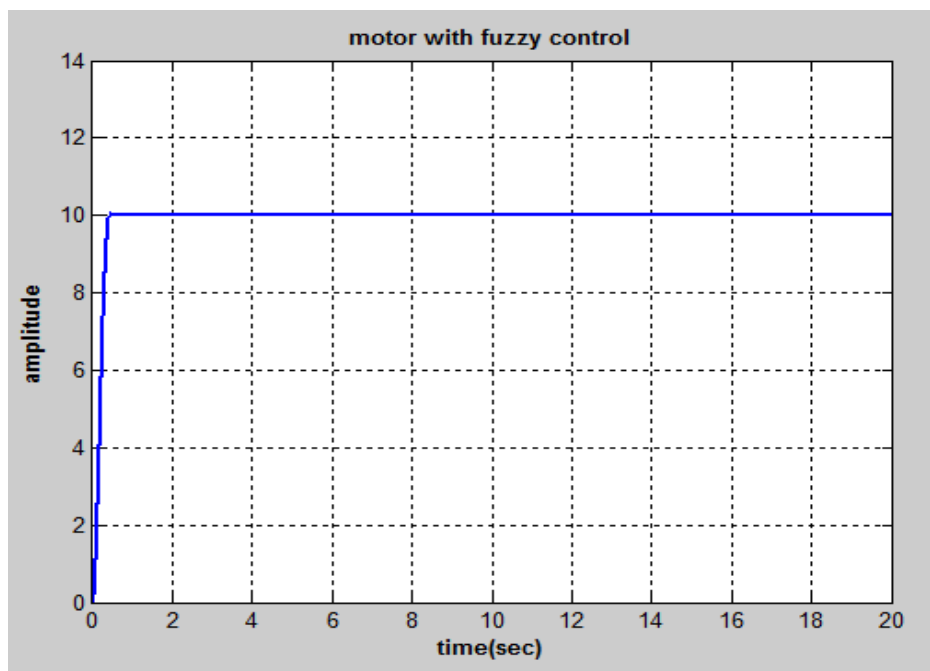


Figure (4-12): Unit step response of the system at no-load with fuzzy controller

4.4.2 System result with PID controller at 75% of load

For 75% of load (gain6=0.75) the output response is shown in Figure (4-13).

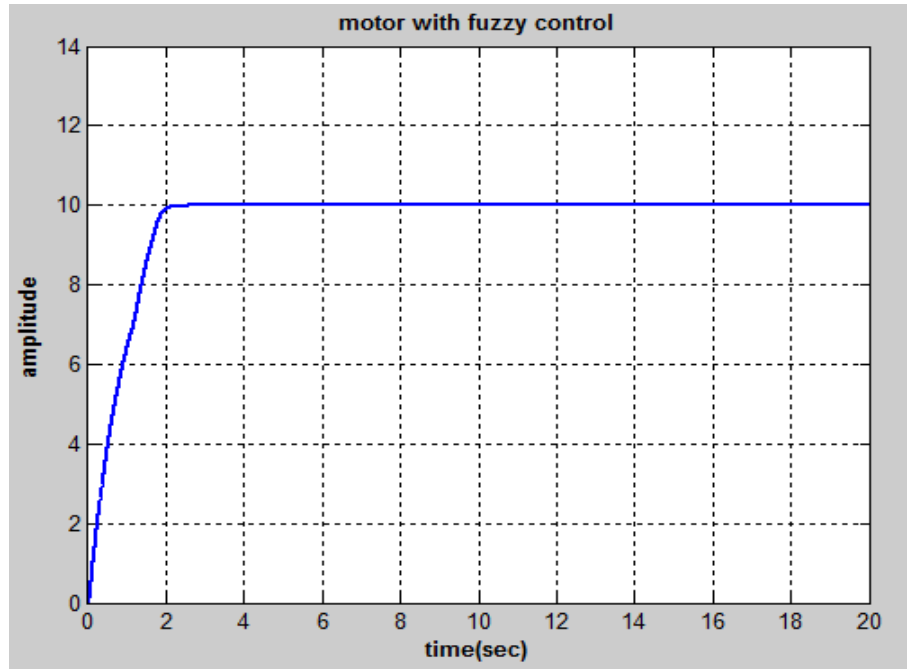


Figure (4-13): Unit step response of the system at 75% of load with fuzzy controller

4.4.3 System Result with PID controller at full load

The output response for full load (gain6=1) illustrate in figure (4-14).Figure (4-15) shows control signal of the fuzzy controller. Figure (4-16) shows control signal of PID controller (in case of no load).

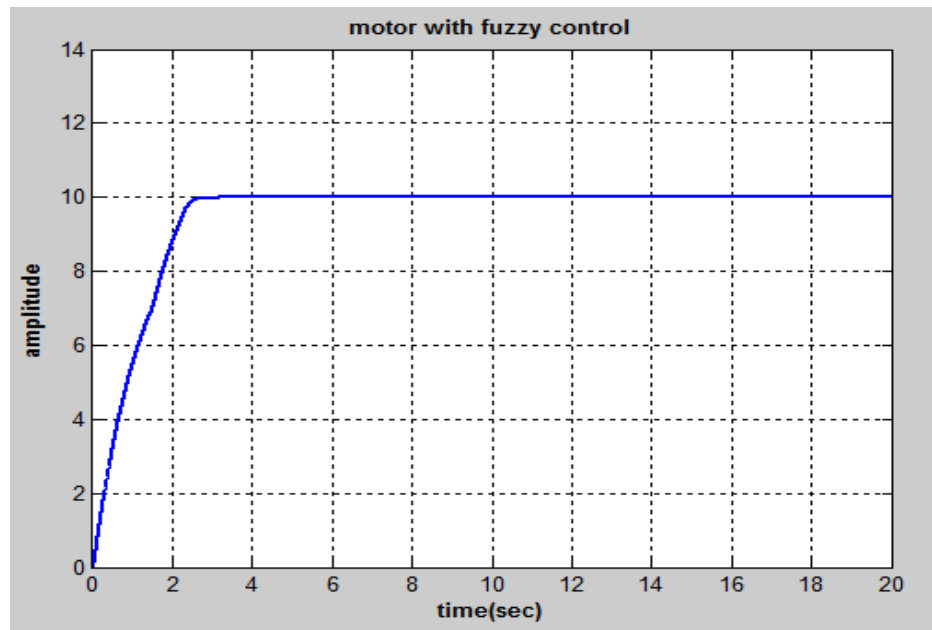


Figure (4-14): Unit step response of the motor in full load with fuzzy control system

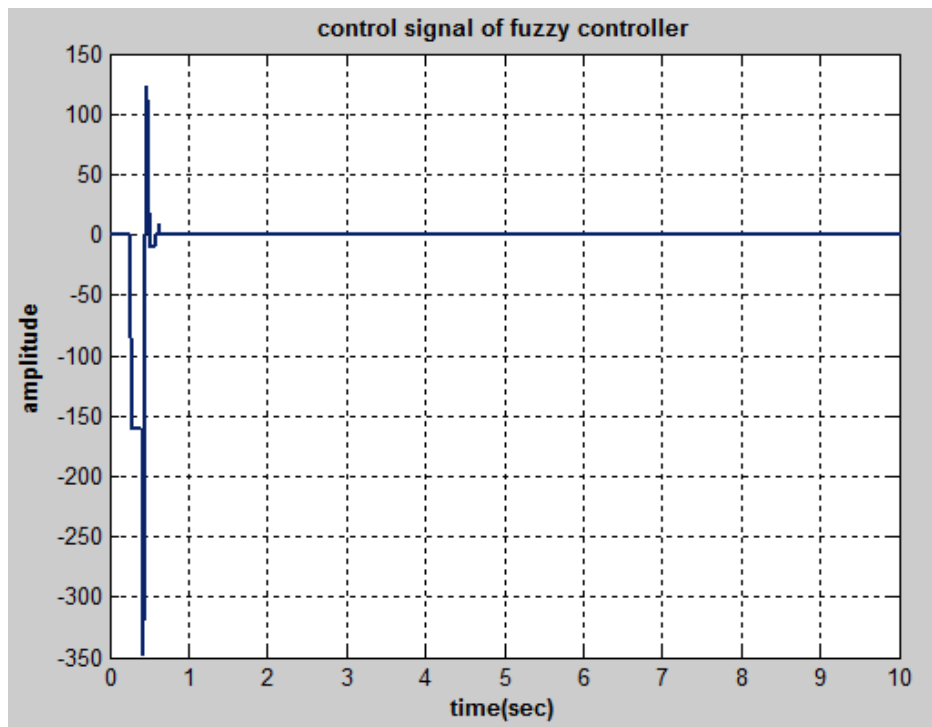


Figure (4-15): Control signal of the fuzzy controller

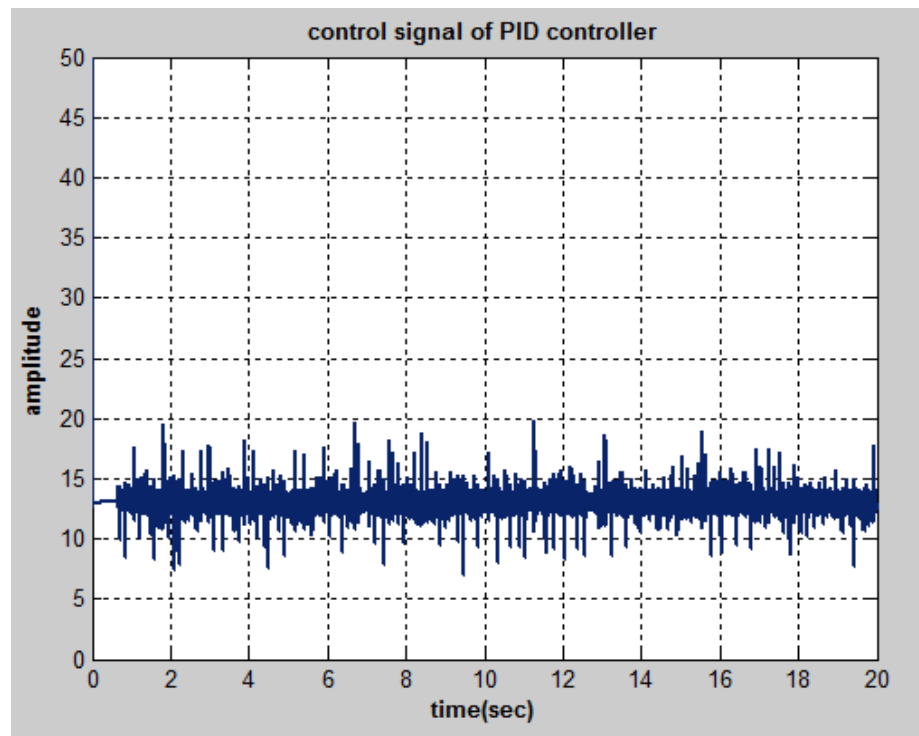


Figure (4-16): Control signal of PID controller

4.5 Results Comparison and Discussions

To achieve the desired goal of this study which is the control of DC motor, system was converted to its equivalent mathematical model and applied control system to it through the MATLAB program. Figure (4-17) shows comparison of system responses for no-load using PID and fuzzy controllers, figure (4-18) shows comparison for 75% of load and figure (4-19) shows comparison for full-load.

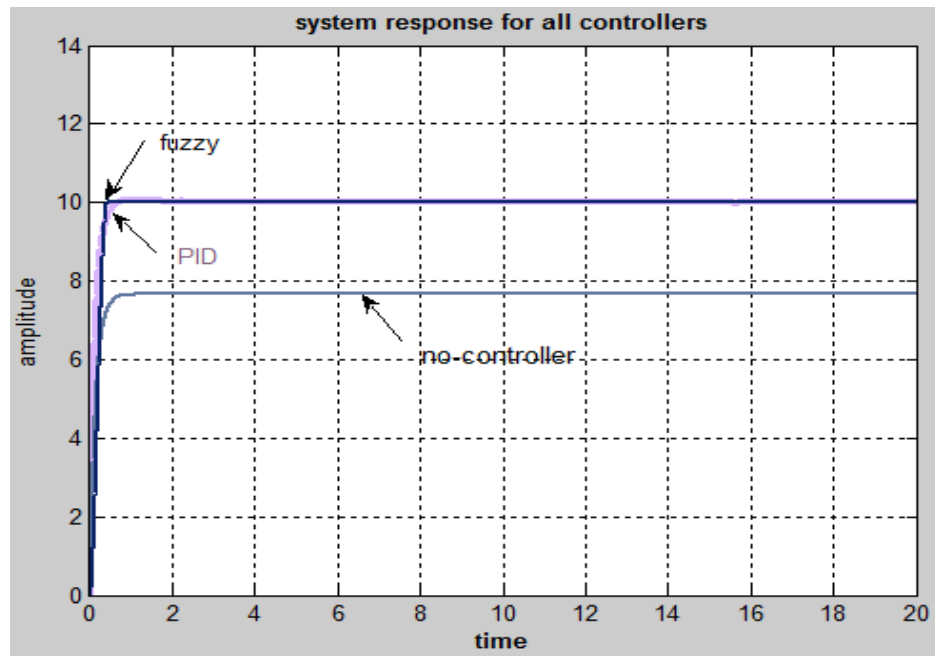


Figure (4-17): Outputs signal of the system in no-load

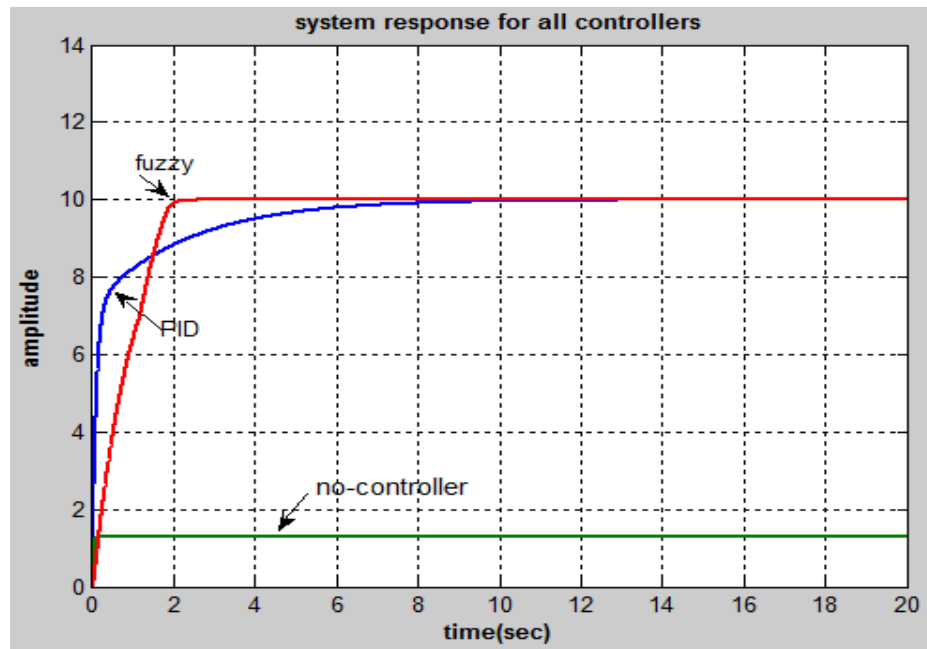


Figure (4-18): Outputs signal of the system in 75% of load

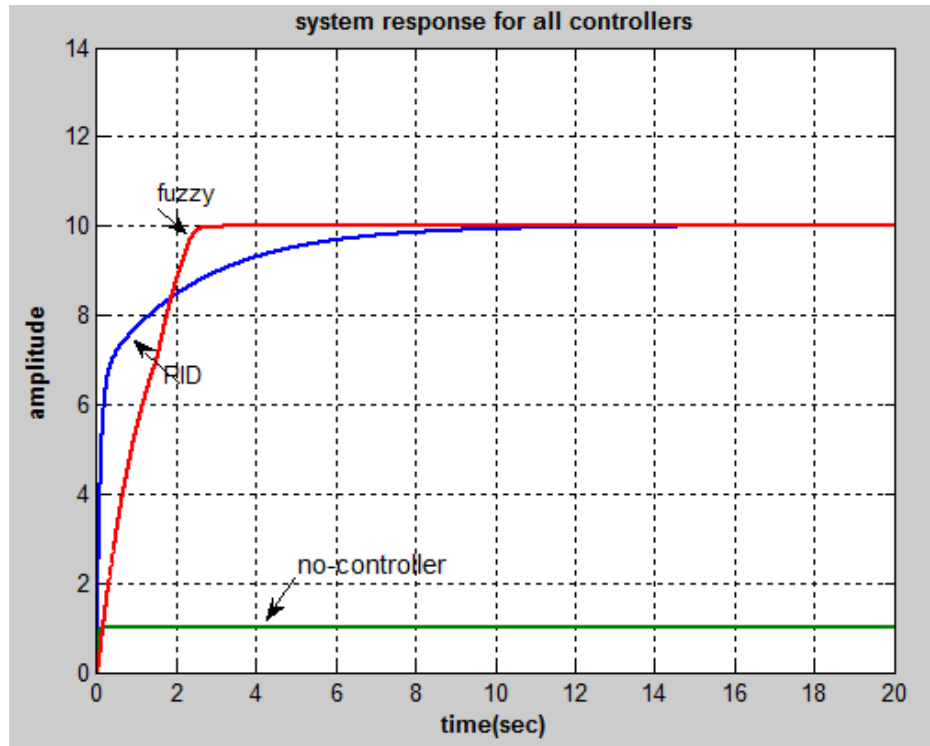


Figure (4-19): Outputs signal of the system in full-load

After analysis the above graphs, the comparative tables are made and result are analyzed and discussed. Tables (4-1), (4-2), (4-3), (4-4) shows the comparative analysis of the controllers.

Table (4-1): rise time Comparison

controller	Rise time (second)		
	No-load	75% of load	Full-load
without	0.32	0.065	0.056
PID	0.296	2.366	3.075
Fuzzy	0.339	1.66	2.08

Table (4-2): Settling time Comparison

controller	Settling time (second)		
	No-load	75% of load	Full-load
without	0.852	0.107	0.134
PID	0.641	11.54	13
Fuzzy	0.464	2.268	2.84

Table (4-3): Steady state error Comparison

controller	Steady state value		
	No-load	75% of load	Full-load
without	2.34	8.729	9.006
PID	0	0	0
Fuzzy	0	0	0

Table (4-4): Peak Overshoot Comparison

controller	Overshoot MP		
	No-load	75% of load	Full-load
without	0%	0%	0%
PID	0.5%	0%	0%
Fuzzy	0.6%	0%	0%

The performance specifications of the controlled system using FLC are better than that of the controlled system using classical PID controller.

In not few number of thesis the load is applied to the motor as constant value, here in this study the load was applied to separately excited DC motor as proportional to speed of motor where its practical way of connect load to separately excited DC motor. the Comparisons between the system outputs responses for the Dynamic performance analysis (Rise Time, Settling Time, Steady state error, Peak Overshoot) showed in tables below (4-1),(4-2),(4-3),(4-4) Respectively. From these tables it found that the motor without control system has good Dynamic performance analysis only in steady state error has bad output, which makes the system unacceptable. Use trial and error to find PID parameters which give a good control for the system it is not bad and often take no long time but still the use of PID controller Is not the best choice for this study according to its bad settling time and not good rise time.

For this system under study the fuzzy controller is the best system controller but the use of trial and error as method to find its system controller parameters is not a practical tool because there is multiple choices in the way of build a good control circuit when use it as controller for the circuit.

Maybe the use of another methods or approach to find control parameters of fuzzy system control make the use of it more easy and Quicker to find its suitable control system parameters and give more good result.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The PID controller and fuzzy controller for separately excited DC motor speed controller have been designed using MATLAB software. Simulation results for system without controller showed that motor need control systems and were obtained. System under PID control have bad settling time. when fuzzy controller was applied , good response obtained after time and effort to find suitable parameters. Increasing load of the system under control result in increasing rise and settling time.

Recommendations:

- System parameters can be tuned using genetic algorithms (GA).
- Design Neuro-Fuzzy controller for DC separately excited motor.

REFERENCES

- [1] Gerhard Henneberger, *Electrical Machines I (Basics, Design, Function, Operation)*, Busch, Schulte, 2002.
- [2] Dr. AF BATI, Electrical Machines Notes.
- [3] Allan R. Hambley - Giorgio Rizzoni, DC MOTORS.
- [4] Webmaster: Matt Page, Intro to D.C. Motors,
<http://lancet.mit.edu/motors/motors1.htm>, 1999.
- [5] , *DC Motors and Generators*, experiment, 2012.
- [6] Mounir HADEF - Mohamed Rachid MEKIDECHE, Parameter identification of a separately excited dc motor via inverse problem methodology, Turk J Elec Eng & Comp Sci, Vol.17, No.2. ,2009.
- [7] SANJEEV KUMAR-VIVEK BARANWAL, *SPEED CONTROL OF SEPARATELY EXCITED DC MOTOR USING NEURO FUZZY TECHNIQUE*, PROJECT THESIS , Rourkela, National Institute of Technology, Orissa. 2009-2010.
- [8] Manafeddin Namazov-Onur Basturk, DC motor position control using fuzzy proportional-derivative controllers with different defuzzification methods, Turkish Journal of Fuzzy Systems, Vol.1, No.1, pp. 36-54. , 2010.
- [9] Antonio Visioli, *Practical PID Control*, Springer, Verlag London, 2006.
- [10] Tamer Mansour, *PID Control, Implementation and Tuning*, first edition , InTech, Croatia, 2011.
- [11] Su Whan Sung Jietae Lee- In-Beum Lee, *PROCESS IDENTIFICATION AND PID CONTROL*, John Wiley & Sons (Asia) Pte Ltd, Singapore, 2009.
- [12] Dr. Maan M. Shaker- Yaareb M.B. Ismeal Al khashab, Design and Implementation of Fuzzy Logic system for DC motor Speed Control, Iraq J. Electrical and Electronic Engineering, Vol.6 No.2. ,2010.
- [13] The MathWorks, *Fuzzy Logic Toolbox User's Guide*, Natick, 1995 – 1999.
- [14] Dr. Mohammad Fazle Azeem, *Fuzzy Inference System- Theory and Applications*, inTech, Croatia, 2012.
- [15] Ismail H. ALTAS - Adel M. SHARAF, *A GENERALIZED DIRECT APPROACH FOR DESIGNING FUZZY LOGIC CONTROLLERS IN MATLAB/SIMULINK GUI ENVIRONMENT*, International Journal of Information Technology and Intelligent Computing, vol.1, no.4, 2007.
- [16] Bernadette Bouchon Meunier-Mariagrazia Dotoli-Bruno Maione, ON THE CHOICE OF MEMBERSHIP FUNCTIONS IN A MAMDANI-TYPE FUZZY CONTROLLER.
- [17] Jan jantzen, *design of fuzzy controllers*, report, lyngby, technical university of Denmark, Denmark, 1998.
- [18] Alan Holland, Fuzzy Systems Case Study, Intelligent Systems: Lectures, Universitat de Girona, 2010.
- [19] Theodore Wildi, *Electrical Machines, Drives, and Power Systems*, Sixth Edition ,Prentice Hall, New Jersey, 2006.

- [20] Allen bradley, *drives engineering handbook*, Rockwell International Corporation. USA, 2004.
- [21] Waleed I. Hameed- Khearia A. Mohamad, SPEED CONTROL OF SEPARATELY EXCITED DC MOTOR USING FUZZY NEURAL MODEL REFERENCE CONTROLLER, International Journal of Instrumentation and Control Systems (IJICS) Vol.2, No.4., 2012.
- [22] Ravinder Kumar- Vineet Girdhar, High Performance Fuzzy Adaptive Control For D.C. Motor, International Journal of Engineering Research & Technology (IJERT), Vol. 1, Issue 8., 2012.
- [23] John J. D'Azzo - Constantine H. Houppis- Stuart N. Sheldon, , *LINEAR CONTROL SYSTEM ANALYSIS AND DESIGN WITH MATLAB*, Fifth Edition, Marcel Dekker, NEW YORK BASEL, 2003.
- [24] Guanrong Chen- Trung Tat Pham, *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*, CRC Press, New York Washington, 2001.
- [25] A Short Fuzzy Logic Tutorial, 2010.