Fuzzy Logic based design of Network control system

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology.

In

Electronics and Communication Engineering

Under the Supervision of

Dr Rajiv Kumar

By

Komal Agarwal ***(091027)***
Meghna Lahiri ***(091064)***
Kushmeen Kambow ***(091070)***

to

JAYPEE UNIVERSITY OF
INFORMATION TECHNOLOGY

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled **"Fuzzy logic based design of Network Control System"**, submitted by Komal Agarwal (091027), Meghna Lahiri (091064), Kushmeen Kambow (091070) in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

10/05/13

Date: 29th May 2013

Supervisor's Name: Dr Rajiv Kumar
Designation: Assistant Professor

Acknowledgement

We are deeply indebted to Professor Dr Rajiv Kumar for his able guidance, expert advice and showing me the right way of carrying out the research project. We are also thankful to him for giving us the special attention and cooperation in between his very busy hours. We are also grateful to our entire electronics faculty for their efforts to strengthen the fundamental concepts so that one can take up any assignment of serious gravity with confidence. We would like to take this opportunity to express our heart-felt indebtedness for his support and enlightenment. At this moment we would also like to express our gratitude for the technical staff of our laboratories. They have always helped us in every-way they can during our project work. We also duly acknowledge the work of the people listed in references.

Date:29th May 2013 **Name of the student: Komal Agarwal (091027)** Komalagarwal
Meghna Lahiri (091064) Meghna
Kushmeen Kambow Kushmeen

(091070)

Table of Content

TABLE OF FIGURES

Figure no. : Title **Page No.**

ABBREVIATIONS AND SYMBOLS

- NCS - Network Control System
- QoC-Quality of Control
- PID – Proportional Integral derivative
- MF – Membership functions
- FLC – Fuzzy Logic controller
- I_f – Field Current
- Ia – Armature Current
- Eg – Back Emf
- Td -Developed torque
- Pd -Required power
- Va -voltage control
- Vf -field control
- B -viscous friction constant,(N.m/rad/s)
- TL -load torque (N.m)
- J -inertia of the motor (Kg.m2) 15
- Kv- motor voltage constant (in V/A-rad/s)
- ω - motor speed (in rad/sec)
- IP -Internet Protocol
- TCP- Transmission Control Protocol
- UDP- User Datagram Protocol
- IS - Intermediate System
- VoIP - Voice over IP
- DNS -Domain Name System
- SNMP- Simple Network Management Protocol
- DHCP- Dynamic Host Configuration Protocol
- RIP -Routing Information Protocol
- e -speed error
- ce - change in speed error
- du - control output
- DOB -degree of belongingness

XI. ABSTRACT

This report talks about Network Control Systems and use of Fuzzy logic in its implementation. Fuzzy logic is a convenient way to map input space to an output space. Fuzzy logic offers several unique features that make it a particularly good choice for many control problems. It is conceptually easy to understand. The mathematical concepts behind fuzzy reasoning are very simple. Fuzzy logic control is a control algorithm based on a linguistic control strategy, which is derived from expert knowledge into an automatic control strategy. It doesn't need any difficult mathematical calculation like the others control system. While the others control system use difficult mathematical calculation to provide a model of the controlled plant, it only uses simple mathematical calculation to simulate the expert knowledge. The use of fuzzy logic for the speed control of a DC motor has been studied and explained. To improve the performance we can also use the adaptive fuzzy controller and neuro-fuzzy controllers. The speed is altered by changing the voltage. Such a fuzzy control scheme consists of two parts. The fuzzy logic controller which is used to generate a control signal for a chopper & speed control of separately excited DC Motor. Triangular membership functions have been used for input variable speed error, change in speed error and control output i.e. change in firing angle. The simulation using Instrument Control Toolbox in Matlab have been used to study the UDP/TCP transmission over the network. In our network , the controller is at one place and the dc motor plant at another which are connected through a network.

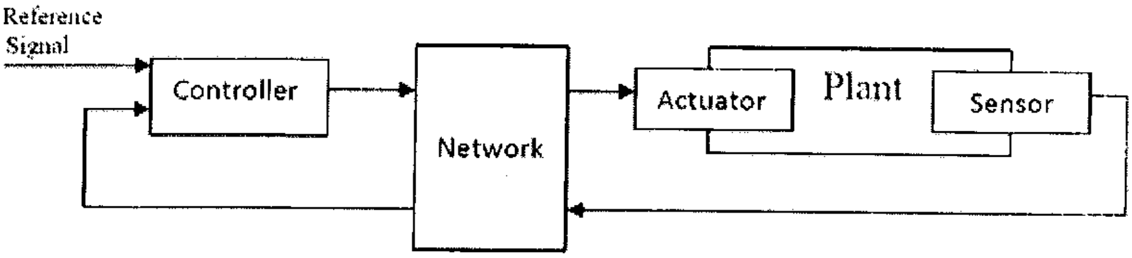
The hardware implementation of the above system can also be achieved which can be taken up as the future scope for the project.

CHAPTER 1: INTRODUCTION

1.1 NETWORK CONTROL SYSTEMS

Network control systems (NCSs) combine two engineering fields, control and computer networks. Computer networks can be wired or wireless. It is a control system wherein the control loops are closed through a real-time network. The defining feature of an NCS is that control and feedback signals are exchanged among the system's components in the form of information packages through a network. The most important feature of NCS is that it connects cyberspace to physical space thus, enabling execution of several tasks from long distance.

A networked control system (NCS) is a feedback control system where the feedback loops are closed by means of an electronic network. An NCS benefits its implementer by reduced cost, wiring, and centralized system maintenance.



A block diagram of an NCS

Figure 1

The functionality of a typical NCS is established by the use of four basic elements:
Sensors - to acquire information,
Controllers - to provide decision and commands,
Actuators -to perform the control commands and
Communication network - to enable exchange of information.

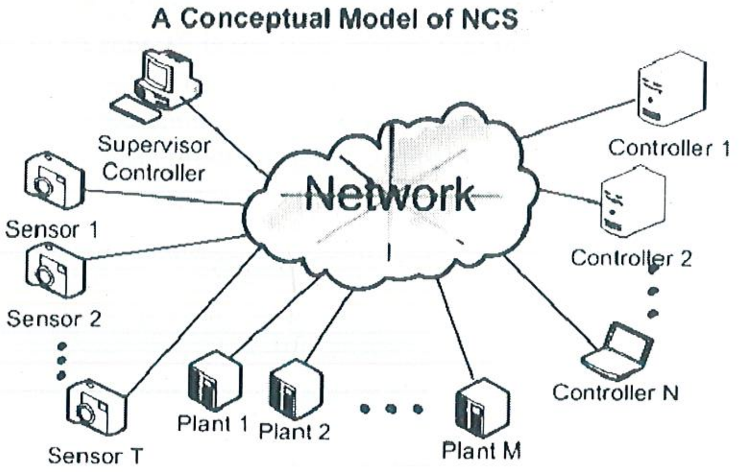


Figure 1: NCS Model

A networked control system with plant, sensor, actuator, controller and network is shown in Fig. The centralized controller communicates with the actuator and sensor over the network .The signal takes some time to transmit between the controller and actuator and between sensor and controller. The transmission time taken by the signal depends upon the topology, routing scheme and the scheduling policy used in the system

.

There are three kinds of delays in Networked Control Systems:
1. Data transfer delay between the sensor and the controller. .
2. Computational delay in the controller,
3. Data transfer delay between controller and actuator,

1.2 MAJOR TOPICS OF ANALYSIS

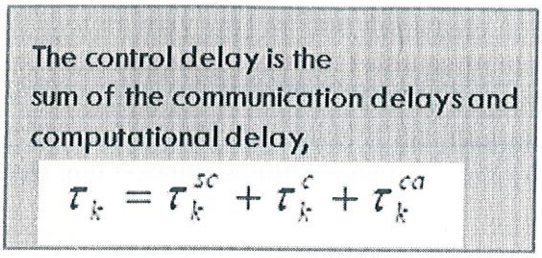
1.2.1 TIME DELAY COMPENSATION

The time delays in the NCS may deteriorate the system performance and cause the system instability. Therefore, it is necessary to design a controller which can compensate for the time delays and improve the control performance of the NCS. At the time of control calculation, no exact correction can be made. An estimator can be used to predict an undelayed plant state and make it available for control calculation.
Network delays in a NCS can be categorized from the direction of data transfers as the sensor-to-controller delay τsc and the controller-to-actuator delay τca. The delays are computed as

$\tau^{sc} = t^{cs} - t^{se}$ ....(1)

$\tau^{ca} = t^{rs} - t^{ce}$..............(2)

where $t^{se}$is the time instant that the remote system encapsulates the measurement to a frame or a packet to be sent, t^{cs} is the time instant that the controller starts processing the measurement in the delivered frame or packet, t^{ce} is the time instant that the main controller encapsulates the control signal to a packet to be sent, and t^{rs} is the time instant that the remote system starts processing the control signal.



The control delay is the sum of the communication delays and computational delay,

$$\tau_k = \tau_k^{sc} + \tau_k^{c} + \tau_k^{ca}$$

Figure 2 : Time Delay Formula

1.2.2 PACKET LOSS

The network can be viewed as a web of unreliable transmission Paths. Some packets not only suffer transmission delay but, even worse, can be lost during transmission. Network packet drops occasionally happen on an NCS when there are node failures or message collisions. In Internet Protocol (IP) networks, Transport layer protocols: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are used for the transmission of data from one process to another. Depending upon the protocol used, the packets are either retransmitted or not if they are lost or received with error at the receiving end. TCP is connection-oriented reliable transport layer protocol. It uses an acknowledgment mechanism to check the error-free arrival of data. In TCP, messages are retransmitted in case of loss of data or erroneous arrival of packets. The packets arrive in the correct packet order at the receiver side. On the other hand, UDP is connection-less and unreliable transport layer protocol. There are no retransmissions in UDP and the packets may arrive in out-of-sequence manner at the receiver end. The retransmissions and acknowledgments can lead to congestion as the network gets loaded. Therefore, UDP is much faster than TCP, but it is also unreliable.

1.2.3 QUALITY OF CONTROL

From the control viewpoint, Quality of Control (QoC) is the most important aim of NCSs. In classical feedback control theory, several properties are used to evaluate the performance of closed-loop systems. The primary evaluation is mainly concerned with meeting the closed loop system response characteristics (such as transient response and steady-state accuracy) and stability. Beyond these requirements, controller designs attempt to minimize the system error for certain anticipated inputs or perturbations. The closed-loop system error is defined as the difference between the desired response and the actual response of the controlled system.

1.2.4 JITTER

Jitter can be defined as time-variations in actual start times of actions, as opposed to stipulated start times. Current control systems are almost universally based on synchronous, clocked systems, so they require communications networks that guarantee delivery of sensor, actuator, and other signals. Jitter depends on clock accuracy, scheduling algorithms, and computer hardware architecture. It is important to derive robust controllers that can guarantee performance despite jitter. The sampling interval of system may vary so that the sensor and at the actuator. Typically, the jitter changes at each sampling interval. For a particular system load condition, and if the system is reasonably predictable, it should be possible to determine the values that the sampling interval actually can take. If the jitter is known prior to run-time its effects can be analyzed and possibly compensated for.

1.2.5 AREAS OF NETWORK CONTROL SYSTEM

Telepresence

Telepresence refers to a set of technologies which allow a person to feel as if they were present, to give the appearance of being present, or to have an effect, via telerobotics, at a place other than their true location.

Telepresence requires that the users' senses be provided with such stimuli as to give the feeling of being in that other location. Additionally, users may be given the ability to affect the remote location. In this case, the user's position, movements, actions, voice, etc. may be sensed, transmitted and duplicated in the remote location to bring about this effect. Therefore information may be traveling in both directions between the user and the remote location.

A popular application is found in telepresence videoconferencing, the highest possible level of video telephony. Telepresence via videodeploys greater technical sophistication and improved fidelity of both sight and sound than in traditional videoconferencing. Technical advancements in mobile collaboration have also extended the capabilities of videoconferencing beyond the boardroom for use with hand-held mobile devices, enabling collaboration independent of location.

Remote surgery

The possibility of being able to project the knowledge and the physical skill of a surgeon over long distances has many attractions. Thus, again there is considerable research underway in the subject. (Locally controlled robots are currently being used for joint replacement surgery as they are more precise in milling bone to receive the joints.) The armed forces have an obvious interest since the combination of telepresence, teleoperation, and telerobotics can potentially save the lives of battle casualties by allowing them prompt attention in mobile operating theatres by remote surgeons.

Recently, teleconferencing has been used in medicine (telemedicine or telematics), mainly employing audio-visual exchange, for the performance of real time remote surgical operations - as demonstrated in Regensburg, Germany in 2002. In addition to audio-visual data, the transfer of haptic (tactile) information has also been demonstrated in telemedicine.

Remote laboratory

Remote Laboratory (also known as online laboratory, remote workbench) is the use of telecommunications to remotely conduct real (as opposed to virtual) experiments, at the physical location of the operating technology, whilst the scientist is utilizing technology from a separate geographical location.

1.3 CONTROLLERS

In everyday speech, the term control and its many variations is frequently used. We can control a situation, such as a policeman controlling the traffic, or a fireman bringing the fire under control. Or an argument may get out of control, or something might happen to us because of circumstances beyond our control. The term control obviously implies the restoration of a desirable state which has been disturbed by external or internal influences. Control processes exist in the most diverse areas. In nature, for instance, control processes serve to protect plants and animals against varying environmental conditions. In economics, supply and demand control the price and delivery time of a product. In any of these cases, disturbances may occur that would change the originally established state. It is the function of the control system to recognize the disturbed state and correct it by the appropriate means. In a similar way as in nature and economics, many variables must be controlled in technology so that equipment and systems serve their intended purpose. With heating systems, for example, the room temperature must be kept constant while external influences have a disturbing effect, such as fluctuating outside temperatures or the habits of the residents as to ventilation, etc. In technology, the term 'control' is not only applied to the control process, but also to the controlled system. People, too, can participate in a closed loop control process. Closed loop control is a process whereby one variable, namely the variable to be controlled (controlled variable) is continuously monitored, compared with another variable, namely the reference variable and influenced in such a manner as to bring about adaptation to the reference variable. The sequence of action resulting in this way takes place in a closed loop in which the controlled variable continuously influences itself.

1.3.1 TYPES OF CONTROLLER

1.3.1.1 PID CONTROLLER

- P controllers are employed in easy-to-control systems where steady-state error is acceptable. A stable and dynamic control response is reached at minimum effort.
- PD controllers:-It makes sense to employ PD controllers in systems with great lag where offset is tolerable. The D component increases the speed of response so that control dynamics improve compared to P controllers.
- I controllers are suitable for use in applications with low requirements as to the control dynamics and where the system does not exhibit great lag. It is an advantage that errors are completely eliminated.
- PI controllers combine the advantages of both P and I controllers. This type o controller produces a dynamic control response without exhibiting steady-state error. Most control tasks can be solved with this type of controller. However, if it is required that the speed of response be as high as possible regardless of the great lag, a PID controller will be the proper choice.
- PID controllers are suitable for systems with great lag that must be eliminated as quick as possible. Compared to the PI controller, the added D component results in better control dynamics. Compared to the PD controller, the added I component prevents error in steady state

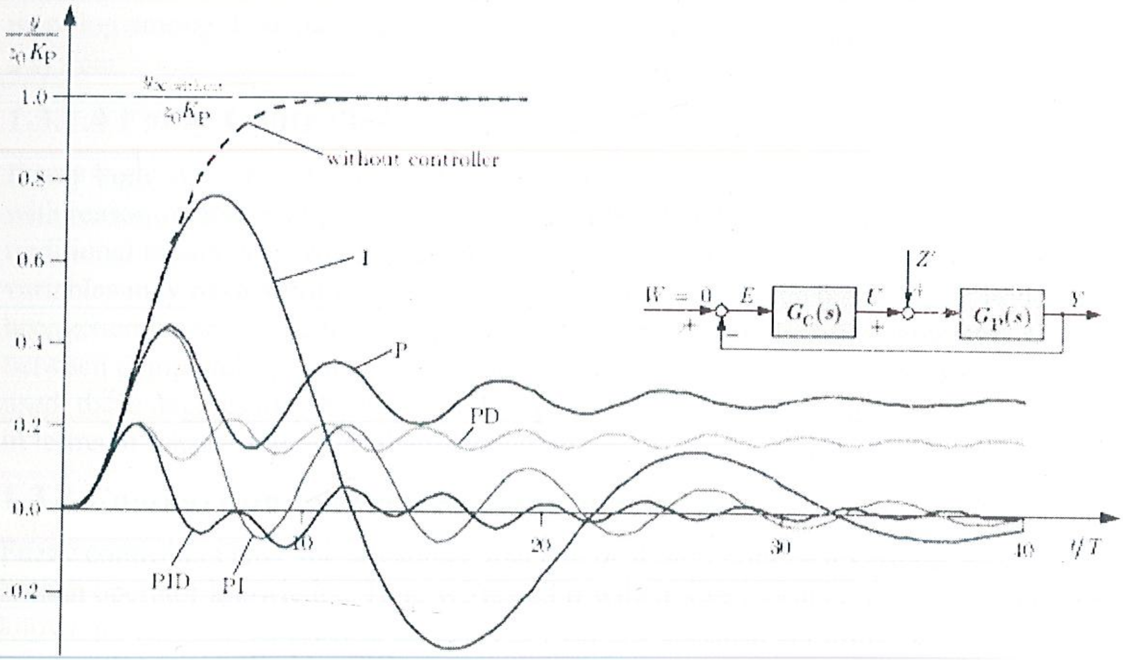


Figure 3: comparison of P,I,PI,PD,PID controllers

1.3.1.2 Bang Bang Controllers

Despite the low-brow sounding name of the Bang-Bang controller, it is a very useful tool that is only really available using digital methods. A better name perhaps for a bang-bang controller is an on/off controller, where a digital system makes decisions based on target and threshold values, and decides whether to turn the controller on and off. Bang-bang controllers are a non-linear style of control .Consider the example of a household furnace. The oil in a furnace burns at a specific temperature -- it can't burn hotter or cooler. To control the temperature in your house then, the thermostat control unit decides when to turn the furnace on, and when to turn the furnace off. This on/off control scheme is a bang-bang controller.

1.3.1.3 Neural network controllers

Neural networks have been used to solve problems in almost all spheres of science and technology. Neural network control basically involves two steps:

- System identification
- Control

It has been shown that a feed forward network with nonlinear, continuous and differentiable activation functions have universal approximation capability. Recurrent networks have also been used for system identification. Given, a set of input-output data pairs, system identification aims to form a mapping among these data pairs. Such a network is supposed to capture the dynamics of a system.

1.3.1.4 Fuzzy Controller

Fuzzy logic is a form of many-valued logic or probabilistic logic; it deals with reasoning that is approximate rather than fixed and exact. Compared to traditional binary sets (where variables may take on true or false values) fuzzy logic variables may have a truth value that ranges in degree between 0 and Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. Furthermore, when linguistic variables are used, these degrees may be managed by specific functions. Irrationality can be described in terms of what is known as the fuzzjective.

1.3.2 Comparison between PID and fuzzy controller

Fuzzy controllers have the advantage that can deal with nonlinear systems and use the human operator knowledge. Here we tested it with a linear system of second order with known parameters. In order to compare it with one classical controller we simulated the same system controlled by PID.

PID controller has only three parameters to adjust. Controlled system shows good results in terms of response time and precision when these parameters are well adjusted.

Fuzzy controller has a lot of parameters. The most important is to make a good choice of rule base and parameters of membership functions. Once a fuzzy controller is given, the whole system can actually be considered as a deterministic system. When the parameters are well chosen, the response of the system has very good time domain characteristics. The fuzzy controlled system is very sensitive to the distribution of membership functions but not to the shape of membership functions.

Fuzzy controlled system doesn't have much better characteristics in time domain that PID controlled system, but its advantage is that it can deal with nonlinear systems.

One of the most important problems with fuzzy controller is that the computing time is much more long that for PID, because of the complex operations as fuzzification and particularly defuzzification.

Some optimization can be done if the defuzzification method is simplified. It means that it is recommended to avoid center of gravity method.

PID controller cannot be applied with the systems which have a fast change of parameters, because it would require the change of PID constants in the time. It is necessary to further study the possible combination of PID and fuzzy controller. It means that the system can be well controlled by PID which is supervised by a fuzzy system.

CHAPTER 2: FOUNDATION OF FUZZY LOGIC

2.1 FUZZY SETS

Fuzzy sets are sets whose elements have degrees of membership. Fuzzy sets were introduced by Lotfi A. Zadeh (1965) as an extension of the classical notion of set. In classical set theory, the membership of elements in a set is assessed in binary terms according to a bivalent condition — an element either belongs or does not belong to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval [0, 1]. Fuzzy sets generalize classical sets, since the indicator functions of classical sets are special cases of the membership functions of fuzzy sets, if the latter only take values 0 or 1. A classical set is a set with a crisp boundary, a classical set "A" of real numbers greater than 6 can be expressed as

$A=\{x / x>6\}$

Where there is a clear unambiguous boundary 6 such that if x is greater than this number, the x belongs to the set A, otherwise x descent belongs to the set.

2.2 BASIC DEFINITIONS AND TERMINOLOGY

A fuzzy set is a pair (A,m) where A is a set and For each , m(x) is the grade of membership of x. If A = {x1,...,xn} the fuzzy set (A,m) can be denoted {m(x1) / x1,...,m(xn) / xn}.

An element mapping to the value 0 means that the member is not included in the fuzzy set, 1 describes a fully included member. Values strictly between 0 and 1 characterize the fuzzy members. The set is called the support of the fuzzy set (A,m) and the set is called the kernel of the fuzzy set (A,m).

Sometimes, a more general definition is used, where membership functions take values in an arbitrary fixed algebra or structure L; usually it is required that L be at least a poset or lattice. The usual membership functions with values in [0, 1] are then called [0, 1]-valued membership functions. This generalization was first considered in 1967 by Joseph Goguen, who was a student of Zadeh.

If x is a collection of objects denoted generally by x then a fuzzy set A in x is defined as a set of ordered pairs.

$A=\{x, \mu A(x)/ x \in x\}$ Where, $\mu A(x)$ =MF's for the fuzzy set A.

The MF's maps each elements of x to a membership grade (or membership value) between 0 and 1.

Usually x is refer to as the universe of discourse, or simply the universe and it may consist of discrete (ordered or unordered) objects or continuous space.

2.2.1FUZZY SETS WITH A CONTINUOUS UNIVERSE

Let x=R+ be the set of possible ages for human being. The fuzzy set B=" about 50 years old" may be expressed as

B={x, μA(x)/ x є x}

Where, μB(x) =1/ (1+ ((x-50/100)4))

So, the constructions of a fuzzy set depend on two things-:

(1)The identification of a suitable universe of discourse.

(2)The specification of an approximate MF's. As MF's are subjective, which means MF's are specified for the some concept.

The subjectivity and no-randomness of fuzzy set is the primary difference between the study of fuzzy sets and probability theory.

2.2.2 LINGUISTIC VARIABLE AND LINGUISTIC VALUES

X="AGE" THEN WE CAN DEFINE FUZZY SETS "YOUNG" , "MIDDLE AGE" , AND "OLD" THAT ARE CHARACTERIZED BY MF'S μyoung(x), μmiddle age(x) and μold(x), respectively.

A linguistic variable "age" can assume different linguistic values such as young, middle age and old.

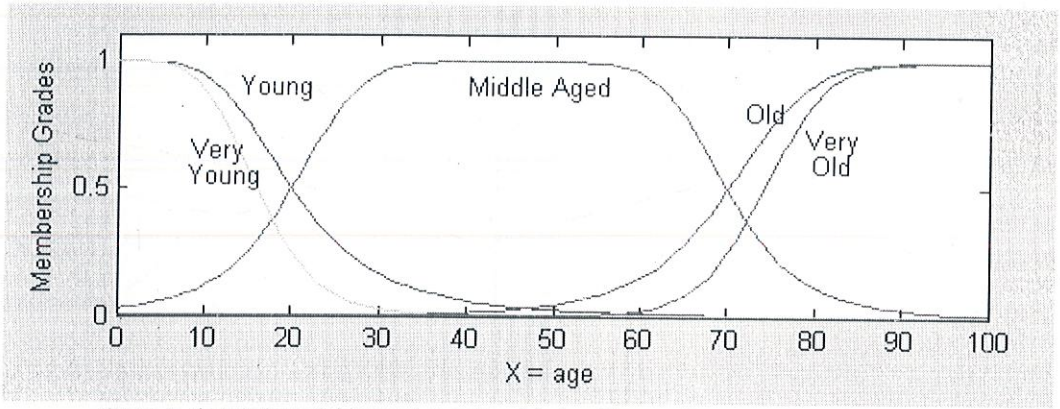


Figure 4: Linguistic variables and membership functions

2.3 SET THEORETIC OPERATIONS

Union, intersection and compliment are the most basic operation on classical sets on the basic of these three operations, a number of identities can be established. Corresponding to the ordinary set operations of union, intersection and compliment, fuzzy sets have similar operation.

2.3.1 CONTAINMENT OR SUBSET-:

Fuzzy set A is contained in fuzzy set B (or equivalently A is subset of B, or A is similar than or equal to B) if and only if $\mu A(x) \leq \mu B(x)$ for all x.

$A \leq B \leftrightarrow \mu A(x) \leq \mu B(x)$.

2.3.2 UNION (DISJUNCTION)-:

The union of two fuzzy sets A and B is a fuzzy set C, written as C=AUB or C=A or B, where MF's is defined to those A and b by

$\mu C(x) = \text{Max}\ (\mu A(x), \mu B(x)) = \mu A(x)\ V\ \mu B(x)$

2.3.3 INTERSECTION (CONJUCTION)-:

Then intersection of two fuzzy sets A and B are the fuzzy set C, written as C=AB or C=A AND B, where MF's related to those of A and B by

$\mu C(x) = \text{Max}\ (\mu A(x), \mu B(x)) = \mu A(x)\ V\ \mu B(x)$

2.3.4 COMPLIMENT (negation)-:

The compliment of fuzzy set a, denoted by A (7 A, NOT A), is defined as

$\mu A(x) = 1 - \mu A(x)$

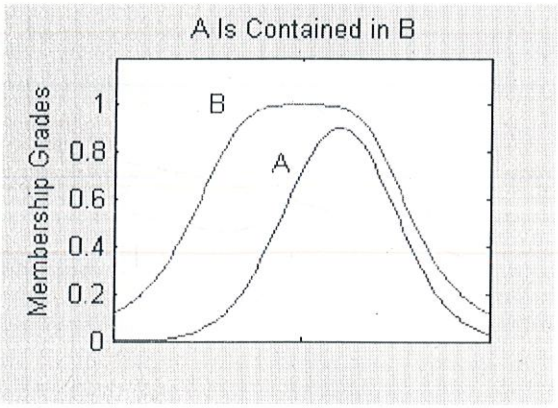


Figure 5: membership functions

2.4 FUZZY IF-THEN RULES

A fuzzy if-then rule (also known as fuzzy rule, fuzzy implication or fuzzy conditional statements) assume the form.

If x is A then y is B.

Where, A and B are linguistic values defined by fuzzy sets on universe of discourse X and y, respectively after "x is A" is called the antecedent or precise will "y is B" is called the consequence or conclusion.

- If pressure is high, then volume is small

- If the road is slippery, then driving is dangerous
- If tomato is red, then it is ripe.
- If the speed is high, then apply the brake a little.

A relation between two variables x & y; this suggest that a fuzzy if then rule be defined as a binary fuzzy relation R on the product space X*Y.

CHAPTER 3 : FUZZY LOGIC

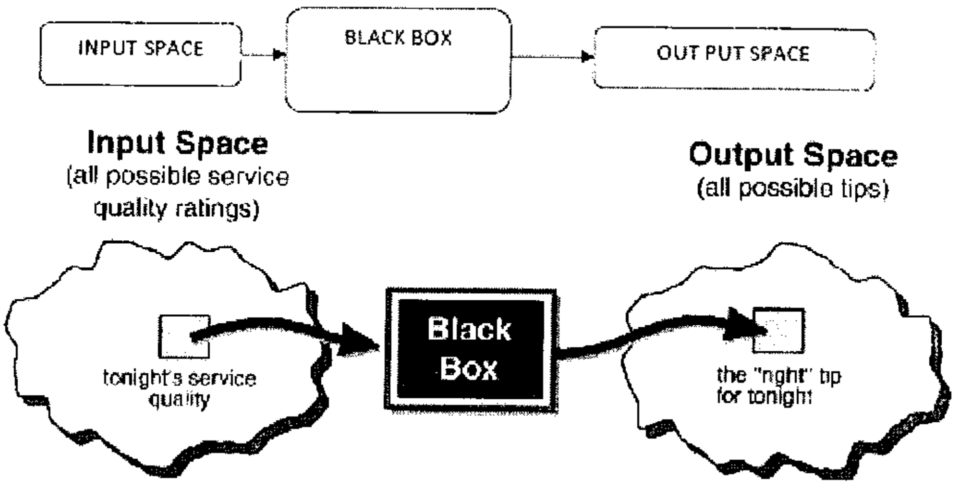
3.1 FUZZY LOGIC

Fuzzy logic has two different meanings, in a narrow sense, fuzzy logic is a logical system, which is an extension of multivolume logic, and however, in a wider sense fuzzy logic is almost synonymous with the theory of fuzzy sets, a theory which relates to classes of objects with un-sharp boundaries in which membership is a matter of degree. In this perspective fuzzy logic in its more narrow definition, fuzzy logic differs both in concept and substance from traditional multi-valued logical system.

Fuzzy logic is a convenient way to map input space to an output space. Mapping input to output is the starting point for everything.

The main objectives of this study are:-

- To understand the fuzzy logic and how fuzzy rule base system can be applied in derivation of rules for power system planning, operation and control.
- To study the MATLAB fuzzy toolbox and understand the concept of member function & rule base.
- To derive optimal power system operation rules at the macro level for better performance and control of the power system.
- To develop a fuzzy logic tool for deriving optimal operational rules at the macro level for better performance and control of power system.



An input-output map for the tipping problem:
"Given the quality of service, how much should I tip?"

Figure 6: Fuzzy logic concept

3.2 WHY USE FUZZY LOGIC?

Fuzzy logic offers several unique features that make it a particularly good choice for many control problems.

Fuzzy logic is conceptually easy to understand. The mathematical concepts behind fuzzy reasoning are very simple. Fuzzy logic is a more intuitive approach without the for-reach complexity. Fuzzy logic is flexible. Fuzzy logic is tolerant of imprecise data. Fuzzy logic can model non-linear function of arbitrary complexity. Fuzzy logic is based on natural language.

It is inherently robust since it does not require precise, noise-free inputs and can be programmed to fail safely if a feedback sensor quits or is destroyed. The output control is a smooth control function despite a wide range of input variations.

Since the Fuzzy logic controller processes user-defined rules governing the target control system, it can be modified and tweaked easily to improve or drastically alter system performance. New sensors can easily be incorporated into the system simply by generating appropriate governing rules.

Fuzzy logic is not limited to a few feedback inputs and one or two control outputs, nor is it necessary to measure or compute rate-of-change parameters in order for it to be implemented. Any sensor data that provides some indication of a system's actions and reactions is sufficient. This allows the sensors to be inexpensive and imprecise thus keeping the overall system cost and complexity low.

Because of the rule-based operation, any reasonable number of inputs can be processed (1-8 or more) and numerous outputs (1-4 or more) generated, although defining the rule base quickly becomes complex if too many inputs and outputs are chosen for a single implementation since rules defining their interrelations must also be defined.

It would be better to break the control system into smaller chunks and use several smaller Fuzzy logic controllers distributed on the system, each with more limited responsibilities.

Fuzzy logic can control nonlinear systems that would be difficult or impossible to model mathematically. This opens doors for control systems that would normally be deemed unfeasible for automation.

3.3 HOW FUZZY LOGIC IS USED?

- Define the control objectives and criteria: What am I trying to control? What do I have to do to control the system? What kind of response do I need? What are the possible (probable) system failure modes?
- Determine the input and output relationships and choose a minimum number of variables for input to the Fuzzy logic engine (typically error and rate-of-change-of-error).
- Using the rule-based structure of Fuzzy logic, break the control problem down into a series of IF X AND Y THEN Z rules that define the desired system output response for given system input conditions.
- Create Fuzzy logic membership functions that define the meaning (values) of Input /Output terms used in the rules.
- Create the necessary pre- and post-processing Fuzzy logic routines if implementing in software, otherwise program the rules into the Fuzzy logic hardware engine.
- Test the system, evaluate the results, tune the rules and membership functions, and retest until satisfactory results are obtained.

3.4 DESIGN USING FUZZY LOGIC TOOLBOX KIT IN MATLAB

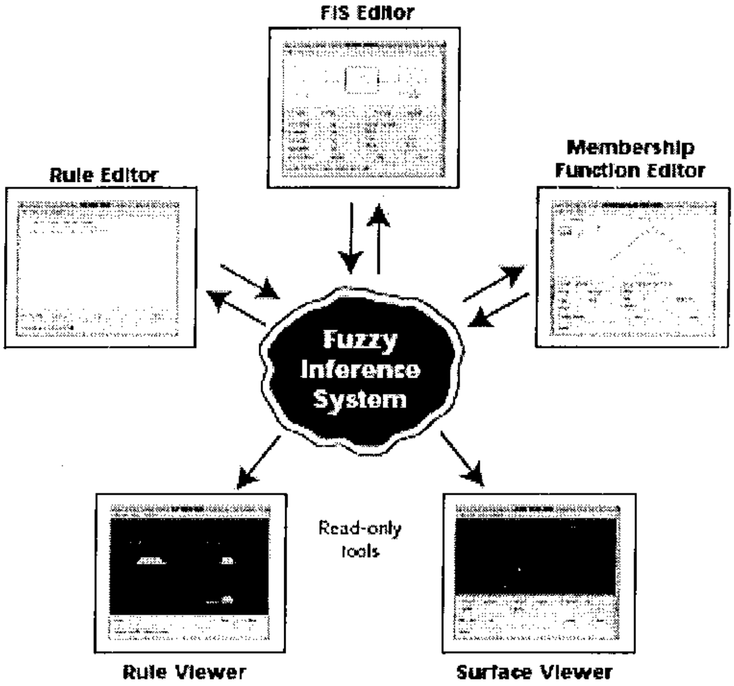
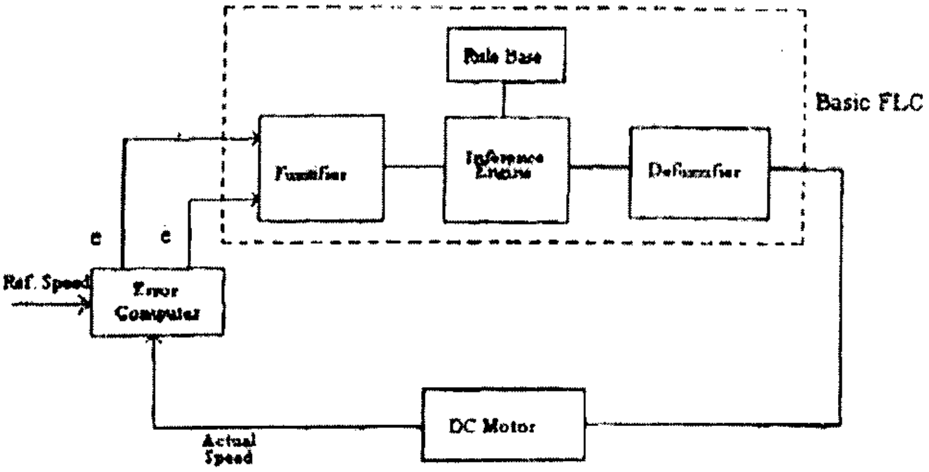


Figure 7: Fuzzy Toolkit

CHAPTER 4 : FUZZY LOGIC CONTROLLER

4.1 INTRODUCTION

Fuzzy logic control is a control algorithm based on a linguistic control strategy, which is derived from expert knowledge into an automatic control strategy. Fuzzy logic control doesn't need any difficult mathematical calculation like the others control system. While the others control system use difficult mathematical calculation to provide a model of the controlled plant, it only uses simple mathematical calculation to simulate the expert knowledge. Although it doesn't need any difficult mathematical calculation, but it can give good performance in a control system. Thus, it can be one of the best available answers today for a broad class of challenging controls problems.



Block diagram of Fuzzy Logic Controller

Figure 8

4.2 BASIC CONFIGURATION OF FUZZY LOGIC CONTROLLER (FLC)

Fuzzy logic control is derived from fuzzy set theory. In fuzzy set theory, the transition between membership and non-membership can be graded. Therefore, boundaries of fuzzy sets can be vague and ambiguous, making it useful for approximate systems. Fuzzy Logic Controller (FLC) is an attractive choice when precise mathematical formulations are not possible. Other advantages are

- It can work with less precise inputs.
- It doesn't need fast processors.
- It needs less data storage in the form of membership functions and rules than conventional look up table for nonlinear controllers.
- It is more robust than other non-linear controllers.

There are three principal elements to a fuzzy logic controller.

a. Fuzzification module (Fuzzifier)
b. Rule base and Inference engine
c. Defuzzification module (Defuzzifier)

4.3 STEPS IN FUZZY LOGIC CONTROLLER

4.3.1 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(ACE)/dt) are chosen to be the input signals of fuzzy AGC.

The membership function is a graphical representation of the magnitude of participation of each input. There are different memberships functions associated with each input and output response. In this study, we use the trapezoidal membership function for input and output variables. The number of membership function determines the quality of control which can be achieved using fuzzy controller. As the number of membership function increases, the quality of control improves. As the number of linguistic variables increases, the computational time and required memory increases. Therefore, a compromise between the quality of control and computational time is needed to choose the number of linguistic variables.

For the speed control of DC motor study, five linguistic variables for each of the input and output variables are used to describe them, as in the following Table.

4.3.2 RULE BASE

As discussed earlier IF-THEN rules are applied in a fuzzy logic controller.

4.3.3DEFUZZIFICATION

The reverse of Fuzzification is called Defuzzification. The use of Fuzzy Logic Controller (FLC) produces required output in a linguistic variable (fuzzy number). According to real world requirements, the linguistic variables have to be transformed to crisp output. Centre of gravity method is the best well-known defuzzification method and used in this research work. Sugeno type of defuzzification method is adopted in this work. It obtains the center of area occupied by the fuzzy set. It is given by the expression.

Defuzzification is the process of producing a quantifiable result in fuzzy logic. Typically, a fuzzy system will have a number of rules that transform a number of variables into a "fuzzy" result, that is, the result is described in terms of membership in fuzzy sets. For example, rules 33

designed to decide how much pressure to apply might result in "Decrease Pressure (15%), Maintain Pressure (34%), Increase Pressure (72%)". Defuzzification would transform this result into a single number indicating the change in pressure. The simplest but least useful defuzzification method is to choose the set with the highest membership, in this case, "Increase Pressure" since it has a 72% membership, and ignore the others, and convert this 72% to some number. The problem with this approach is that it loses information. The rules that called for decreasing or maintaining pressure might as well have not been there in this case.

A useful defuzzification technique must first add the results of the rules together in some way. The most typical fuzzy set membership function has the graph of triangle. Now, if this triangle were to be cut in a straight horizontal line somewhere between the top and the bottom, and the top portion were to be removed, the remaining portion forms a trapezoidal. The first step of defuzzification typically "chops off" parts of the graphs to form trapezoids (or other shapes if the initial shapes were not triangles). For example, if the output has "Decrease Pressure (15%)", then this triangle will be cut 15% the way up from the bottom. In the most common technique, all of these trapezoids are then superimposed one upon another, forming a single geometric. Then, the centroid of this shape, called the fuzzy centroid, is calculated. The x coordinate of the centroid is the defuzzified value.

4.4 FUZZY CONTROL IN DETAIL:

Fuzzy controllers are very simple conceptually. They consist of an input stage, a processing stage, and an output stage. The input stage maps sensor or other inputs, such as switches, thumbwheels, and so on, to the appropriate membership functions and truth values. The processing stage invokes each appropriate rule and generates a result for each, then combines the results of the rules. Finally, the output stage converts the combined result back into a specific control output value.

The most common shape of membership functions is triangular, although trapezoidal and bell curves are also used, but the shape is generally less important than the number of curves and their placement. From three to seven curves are generally appropriate to cover the required range of an input value, or the "universe of discourse" in fuzzy jargon.

As discussed earlier, the processing stage is based on a collection of logic rules in the form of IF-THEN statements, where the IF part is called the "antecedent" and the THEN part is called the "consequent". Typical fuzzy control systems have dozens of rules.

Consider a rule for a thermostat:

"IF temperature is COLD then heater is HIGH" 34

This rule uses the truth value of the "temperature" input, which is some truth value of "cold", to generate a result in the fuzzy set for the "heater" output, which is some value of "high".

This result is used with the results of other rules to finally generate the crisp composite output. Obviously, the greater the truth value of "cold", the higher the truth value of "high", though this does not necessarily mean that the output itself will be set to "high", since this is only one rule among many. In some cases, the membership functions can be modified by "hedges" that are equivalent to adjectives. Common hedges include "about", "near", "close to", "approximately", "very", "slightly", "too", "extremely", and "somewhat". These operations may have precise definitions, though the definitions can vary considerably between different implementations.

In practice, the fuzzy rule sets usually have several antecedents that are combined using fuzzy operators, such as AND, OR, and NOT, though again the definitions tend to vary: AND, in one popular definition, simply uses the minimum weight of all the antecedents, while OR uses the maximum value. There is also a NOT operator that subtracts a membership function from 1 to give the "complementary" function.

There are several different ways to define the result of a rule, but one of the most common and simplest is the "max-min" inference method, in which the output membership function is given the truth value generated by the premise.

Rules can be solved in parallel in hardware, or sequentially in software. The results of all the rules that have fired are "defuzzified" to a crisp value by one of several methods. There are dozens in theory, each with various advantages and drawbacks.

The "centroid" method is very popular, in which the "centre of mass" of the result provides the crisp value. Another approach is the "height" method, which takes the value of the biggest contributor. The centroid method favors the rule with the output of greatest area, while the height method obviously favors the rule with the greatest output value.

The diagram below demonstrates max-min inferencing and centroid defuzzification for a system with input variables "x", "y", and "z" and an output variable "n". Note that "mu" is standard fuzzy-logic nomenclature for "truth value":

CHAPTER 5 : DC MOTOR

5.1 DC MOTOR AND ITS OPERATION

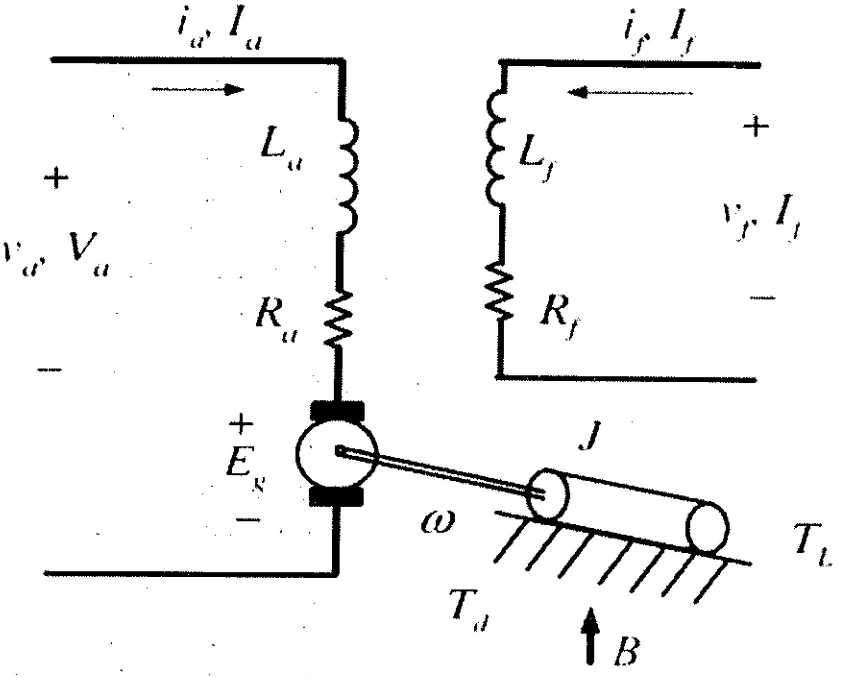


Figure 9 : DC Motor

- The field windings are used to excite the field flux.
- Armature current is supplied to the rotor via brush and commutator for the mechanical work.
- Interaction of field flux and armature current in the rotor produces torque.

5.2 OPERATION:

• When a separately excited motor is excited by a field current of if and an armature current of ia flows in the circuit, the motor develops a back emf and a torque to balance the load torque at a particular speed.

• The if is independent of the ia. Each windings are supplied separately. Any change in the armature current has no effect on the field current.

• The if is normally much less than the ia.

5.3 FIELD AND ARMATURE EQUATIONS:

5.3.1 INSTANTANEOUS FIELD CURRENT:

vf= Rfif+Lf*dif/dt
Where Rf and Lf are the field resistance and inductor, respectively.

5.3.2 INSTANTANEOUS ARMATURE CURRENT :

Va= Raia+La*dia/dt+eg
Where Ra and La are the armature resistance and inductor, respectively.
The motor back emf, which is also known as speed voltage, is expressed as:
eb= Kv ω if
Kv is the motor voltage constant (in V/A-rad/s) and ω is the motor speed (in rad/sec)

5.4 BASIC TORQUE EQUATION:

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

Td = J*dω/dt+Bω+TL

The torque developed by the motor is:

Td=Kt*if*ia

Where (Kt=Kv) is torque constant in V/A-rad/sec.
Sometimes it is written as:

Td=Kt*φ*ia

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

B : viscous friction constant,(N.m/rad/s)
TL :load torque (N.m)
J : inertia of the motor (Kg.m2) 15

5.5 STEADY STATE OPERATION:

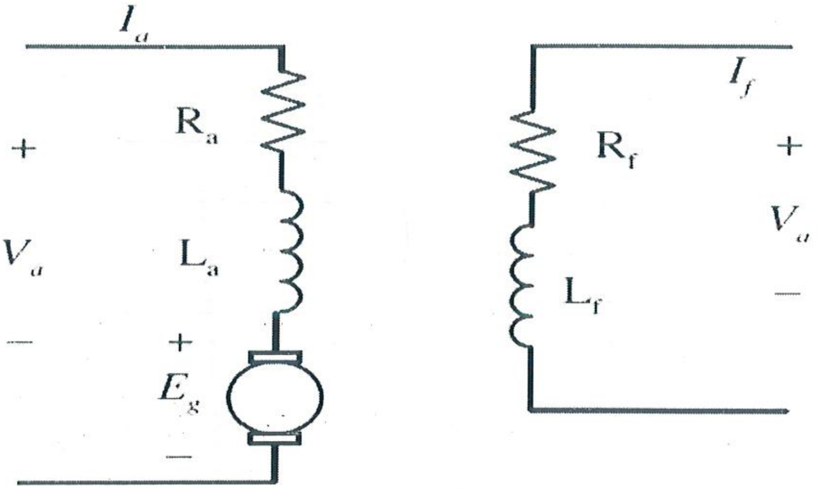


Figure 10

Separately Excited DC Motor, in Steady State under study state operation, time derivatives is zero. Assuming the motor is not saturated.
For field circuit,
Vf=If*Rf
The back emf is given by:
Eg=Kv*ω*If
The armature circuit:Va = Ia*Ra+Eg = Ia*Ra+Kv*ω*If

5.5.1 STEADY-STATE TORQUE AND SPEED:

The motor speed can be easily derived:
if Ra is a small value (which is usual), or when the motor is lightly loaded, i.e. Ia is small, that is if the field current is kept constant, the motor speed depends only on the supply voltage.
The developed torque is: Td=Kt*If*Ia=Bω+TL 16
The required power is: Pd = Tdω

5.5.2 TORQUE AND SPEED CONTROL:

From the derivation, several important facts can be deduced for steady-state operation of DC motor.

- For a fixed field current, or flux (If), the torque demand can be satisfied by varying the armature current (Ia).
- The motor speed can be varied by controlling Va (voltage control) and controlling Vf(field control).
- These observations lead to the application of variable DC voltage for controlling the speed and torque of DC motor.

5.6 VARIABLE SPEED OPERATION:

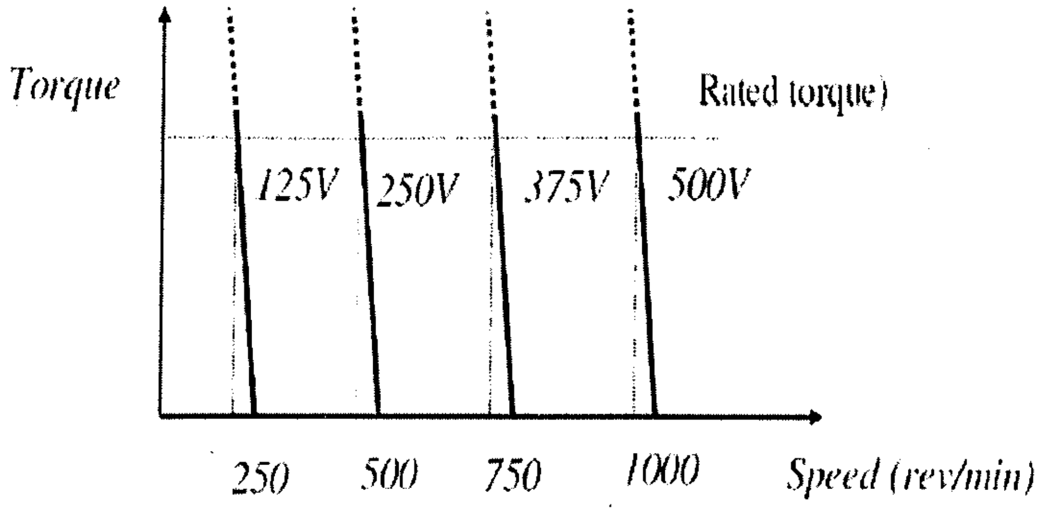


Figure 11:Torque Vs Speed Characteristic For Different Armature Voltages

Family of steady-state torque speed curves for a range of armature voltage can be drawn as above:

- The speed of DC motor can simply be set by applying the correct voltage.
- Note that speed variation from no-load to full load (rated) can be quite small. It depends on the armature resistance.

5.7 BASE SPEED AND FIELD-WEAKENING:

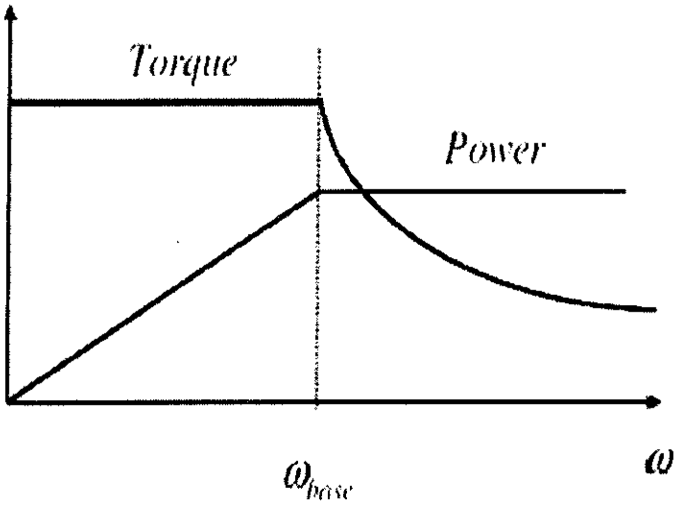


Figure 12: Torque Vs Speed And Power Vs Speed Characteristic Of Separately Excited DC Motor

Base speed: (ωbase)
-- the speed which correspond to the rated Va, rated Ia and rated If.
20

• Constant Torque region (ω >wbase)
– Ia and If are maintained constant to met torque demand. Va is varied to control the speed.
Power increases with speed.

• Constant Power region (ω >wbase)
– Va is maintained at the rated value and If is reduced to increase speed. However. the power developed by the motor (= torque x speed) remains constant. This phenomenon is known as field weakening.

CHAPTER 6: HARDWARE-IN LOOP SIMULATION

In Internet Protocol (IP) networks, Transport layer protocols:
Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are used for the transmission of data from one process to another. Depending upon the protocol used, the packets are either retransmitted or not if they are lost or received with error at the receiving end. TCP is connection-oriented reliable transport layer protocol. It uses an acknowledgment mechanism to check the error-free arrival of data. In TCP, messages are retransmitted in case of loss of data or erroneous arrival of packets. The packets arrive in the correct packet order at the receiver side. On the other hand, UDP is connection-less and unreliable transport layer protocol. There are no retransmissions in UDP and the packets may arrive.in out-of-sequence manner at the receiver end. The retransmissions and acknowledgments can lead to congestion as the network gets loaded. Therefore, UDP is much faster than TCP, but it is also unreliable. The simulations of Networked Control System have been performed using instrument control toolbox of MATLAB/ Simulink. The simulink models with network are shown. The simulink blocks are simulated in host PC and remote PC. The sending Simulink model with fuzzy controller is kept in host PC and receiving simulink block with the dc motor plant is kept in remote PC. This configuration forms an environment of closed loop networked
Control system. The Transport layer protocol User Datagram Protocol (UDP) is used for sending data over the network.UDP has been preferred over TCP since UDP provides increased speed of communication and TCP introduces more delays and can lead to undesirable performance of theNCS.UDP Send and UDP Receive blocks have been used in the simulink models.

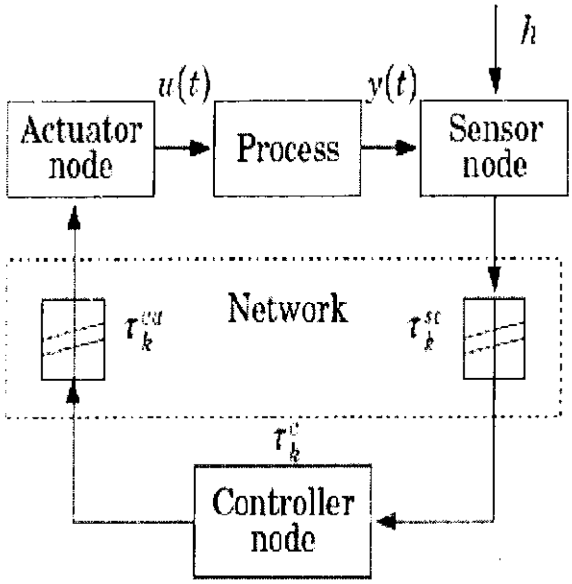


Figure 13:hardware in loop

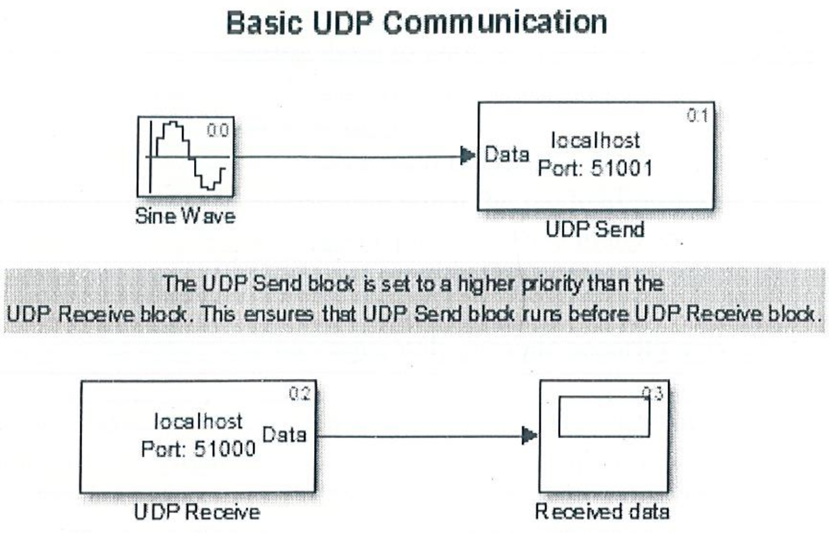


Figure 14

The User Datagram Protocol (UDP) is one of the core members of the Internet Protocol Suite, the set of network protocols used for the Internet. With UDP, computer applications can send messages, in this case referred to as *datagrams*, to other hosts on an Internet Protocol (IP) network without requiring prior communications to set up special transmission channels or data paths. UDP is sometimes called the Universal Datagram Protocol. The protocol was designed by David P. Reed in 1980.
The service provided by UDP is an unreliable service that provides no guarantees for delivery and no protection from duplication (e.g. if this arises due to software errors within an Intermediate System (IS)). The simplicity of UDP reduces the overhead from using the protocol and the services may be adequate in many cases. UDP uses a simple transmission model without implicit hand-shaking dialogues for guaranteeing reliability, ordering, or data integrity. Thus, UDP provides an unreliable service and datagrams may arrive out of order, appear duplicated, or go missing without notice. UDP assumes that error checking and correction is either not necessary or performed in the application, avoiding the overhead of such processing at the network interface level. Time-sensitive applications often use UDP because dropping packets is preferable to using delayed packets. If error correction facilities are needed at the network interface level.
UDP provides a minimal, unreliable, best-effort, message-passing transport to applications and upper-layer protocols. Compared to other transport protocols, UDP and its UDP-Lite variant are unique in that they do not establish end-to-end connections between communicating end systems. UDP communication consequently does not incur connection. Establishment and teardown overheads and there is minimal associated end system state. Because of these characteristics, UDP can offer a very efficient communication transport to some applications, but has no inherent congestion control or reliability. A second unique characteristic of UDP is that it provides no inherent On many platforms, applications can send UDP datagram at the line rate of the link interface, which is often much greater than the available path capacity, and doing so would contribute to congestion along the path, applications therefore need to be designed responsibly (RFC 4505). One increasingly popular use of UDP is as a tunneling protocol,

where a tunnel endpoint encapsulates the packets of another protocol inside UDP datagrams and transmits them to another tunnel endpoint, which decapsulates the UDP datagrams and forwards the original packets contained in the payload. Tunnels establish virtual links that appear to directly connect locations that are distant in the physical Internet topology, and can be used to create virtual (private) networks. Using UDP as a tunneling protocol is attractive when the payload protocol is not supported by middle boxes that may exist along the path, because many middle boxes support UDP transmissions. UDP does not provide any communications security. Applications that need to protect their communications against eavesdropping, tampering, or message forgery therefore need to separately provide security services using additional protocol mechanisms.

UDP's stateless nature is also useful for servers that answer small queries from huge numbers of clients. Unlike TCP, UDP is compatible with packet broadcast (sending to all on local network) and multicasting (send to all subscribers).

UDP applications use datagram sockets to establish host-to-host communications. Sockets bind the application to service ports that function as the endpoints of data transmission. A port is a software structure that is identified by the port number, a 16 bit integer value, allowing for port numbers between 0 and 65,535. Port 0 is reserved, but is a permissible source port value if the sending process does not expect messages in response.

6.1RELIABILITY AND CONGESTION CONTROL SOLUTIONS

Lacking reliability, UDP applications must generally be willing to accept some loss, errors or duplication. Some applications such as TFTP may add rudimentary reliability mechanisms into the application layer as needed. Most often, UDP applications do not require reliability mechanisms and may even be hindered by them. Streaming media, real-time multiplayer games and voice over IP (VoIP) are examples of applications that often use UDP. If an application requires a high degree of reliability, a protocol such as the Transmission Control Protocol or erasure codes may be used instead.

Lacking any congestion avoidance and control mechanisms, network-based mechanisms are required to minimize potential congestion collapse effects of uncontrolled, high rate UDP traffic loads. In other words, since UDP senders cannot detect congestion, network-based elements such as routers using packet queuing and dropping techniques will often be the only tool available to slow down excessive UDP traffic. The Datagram Congestion Control Protocol (DCCP) is being designed as a partial solution to this potential problem by adding end host TCP-friendly congestion control behavior to high-rate UDP streams such as streaming media.

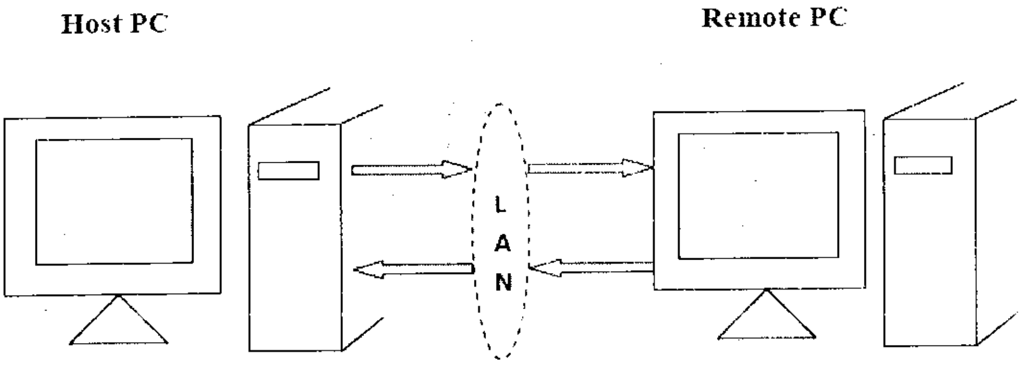
6.2 APPLICATIONS

While the total amount of UDP traffic found on a typical network is often in the order of only a few percent numerous key Internet applications use UDP, including: the Domain Name System (DNS), where queries must be fast and only consist of a single request followed by a single reply packet, the Simple Network Management Protocol (SNMP), the Dynamic Host Configuration Protocol (DHCP) and the Routing Information Protocol (RIP).

Voice and video traffic is generally transmitted using UDP. Real-time video and audio streaming protocols are designed to handle occasional lost packets, so only slight degradation in quality occurs rather than large delays if lost packets are retransmitted. Because both TCP and UDP run over the same network, many businesses are finding that a recent increase in UDP traffic from these real-time applications is hindering the performance of applications using TCP, such as point of sale, accounting, and database systems. When TCP detects packet loss, it will throttle back its data rate usage. Since both real-time and business applications are important to businesses, developing quality of service solutions is crucial.

6.3 MEASUREMENT OF DELAY IN FEEDBACK LOOP

In this section we are find the process delay in feedback loop, this delay measurement will perform using two computer PC“s. One is acts a host pc and anther is acts as remote pc, these both PC“s are connected through a Local Area Network (LAN) and communicating each other as shown fig.5.1.



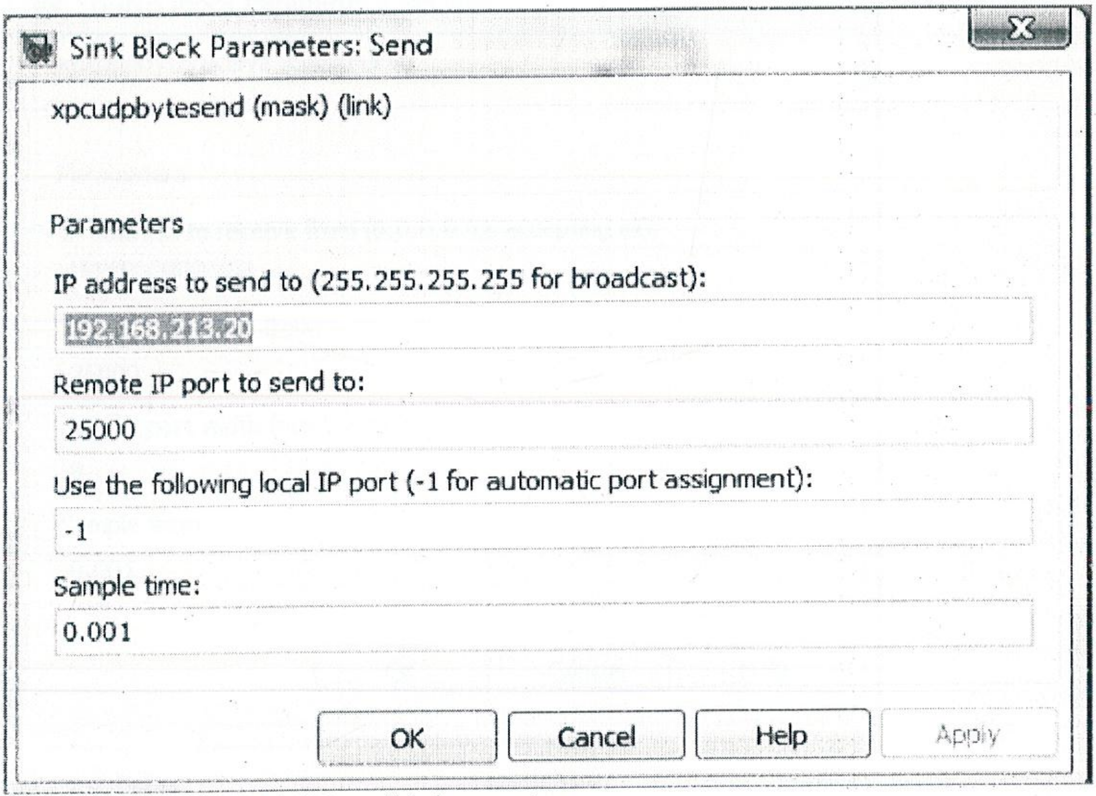
Communicating two PC's through LAN

Figure 15

Host PC Remote PC

Communicating two PC‟s through LAN Figure shows the transmitting simulink block, this block put in host PC. For transmitting the signal from one system to another we are using the UDP protocol. These blocks will be available in MATLAB/simulink.UDP blocks mostly used for transmitting a signal or any data from one system to another through a LAN. These blocks works based on IP address of the systems. Transmitting simulink blocks in host PC. The sinusoidal signal of amplitude 1v and frequency of 1 Hz is applied as input, shown in fig. The simulink block of pack is used to convert one or more Simulink signals of varying data types to a single vector of uint8 as required by the Send block. This block is the exact analog of the Pack block. It receives a vector of uint8 and outputs various Simulink data types in different sizes. The Pack block is on the sending side and the Unpack block is on the receiving side in different models.

UDP send: The Send block has only one input port, which receives the uint8 vector that is sent as a UDP packet.



Sink Block Parameters: Send

xpcudpbytesend (mask) (link)

Parameters

IP address to send to (255.255.255.255 for broadcast): 192.168.213.20

Remote IP port to send to: 25000

Use the following local IP port (-1 for automatic port assignment): -1

Sample time: 0.001

Figure 16:UDP send block parameters

Block Parameters:

☐ *IP address to send to:* Specify the IP address to send the packet.

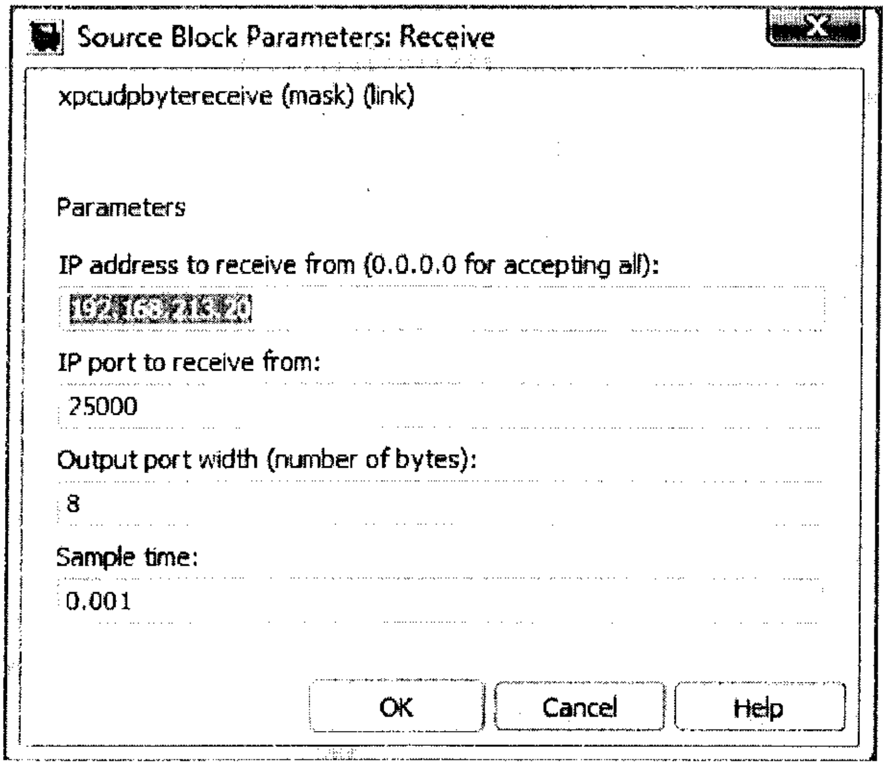
☐ *IP port to send to:* Specify the port to which to send the packet.

☐ *Use the following local IP port:* Set this parameter to -1 (default) to allow the networking stack to automatically determine the local IP port that is used for sending. Otherwise, specify a particular port to send a packet from that port.

☐ *Sample time:* You can set this parameter to -1 for an inheritable sample time, but it is recommended that this be set to some specific (large) value to eliminate chances of dropped packets. This is especially true when you are using a small base sample time.

UDP receive:

The Receive block has two output ports. The first port is the output of the received data as a vector of uint8 while the second one is a flag indicating whether any new data has been received. This port outputs a value of 1 for the sample when there is new data and a 0 otherwise. The default behavior of the Receive block is to keep the previous output when there is no new data. You can modify this behavior by using the second port to flag when there is new data. as shown in fig



xpcudpbytereceive (mask) (link)

Parameters

IP address to receive from (0.0.0.0 for accepting all):
192.168.213.20

IP port to receive from:
25000

Output port width (number of bytes):
8

Sample time:
0.001

Figure 17:UDP receive block parameters

Block Parameters:

☐ *IP address to receive from:* If set to a specific IP address, only packets arriving from that IP address are received.

☐ *IP port to receive from:* Port that the block accepts data from. The other end of the communication sends data to the port specified here.

☐ *Output port width:* Width of the acceptable packets. You can obtain this when designing the other side (send side) of the communication.

CHAPTER 7: WORK DONE

7.1 SIMULATION RESULTS FOR DC MOTOR WITH PID CONTROLLER

7.1.1 SIMULINK MODEL

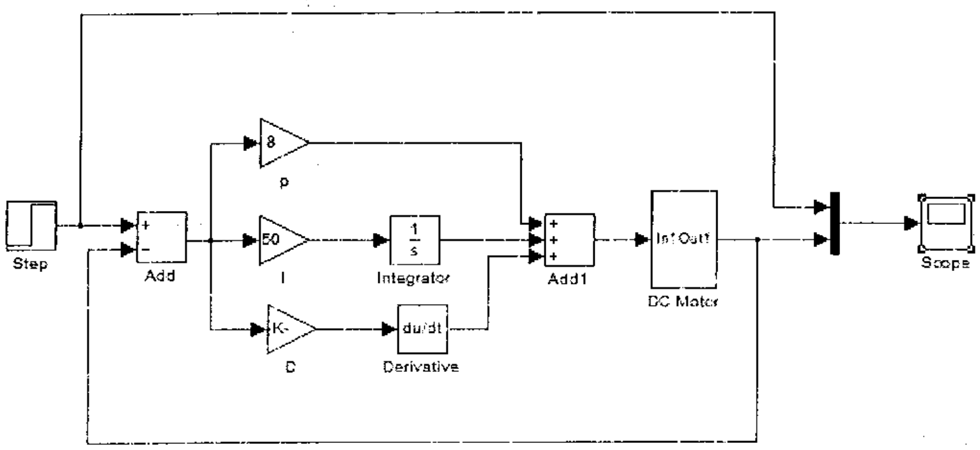


Figure 18

7.1.2 SIMULATION RESULTS

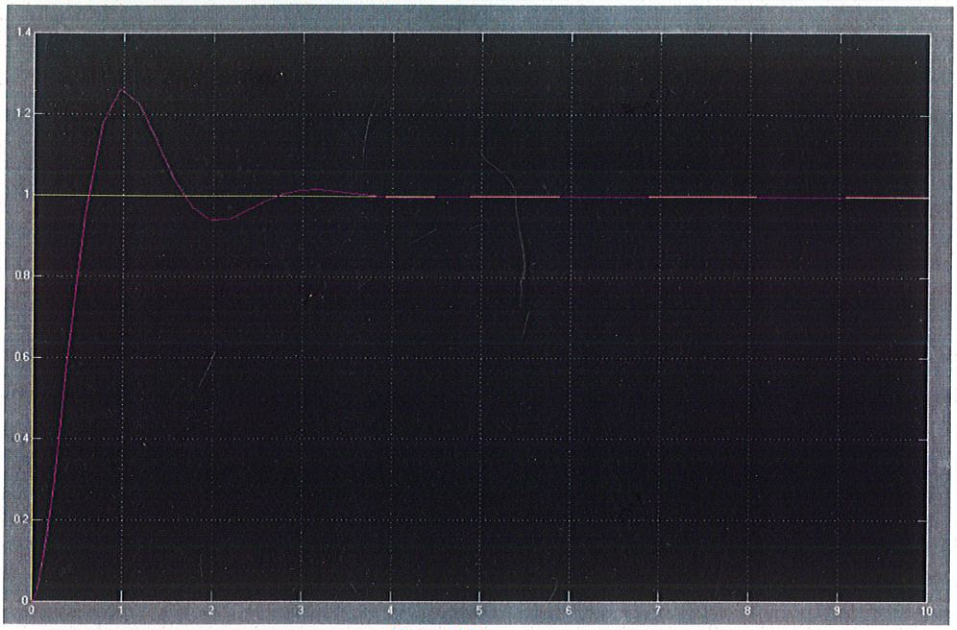


Figure 19

7.2FUZZY RULE BASE FOR SPEED CONTROL OF DC MOTOR:

- The algorithm for fuzzy speed control in details can be summarized as follows. A numerical example is included in each step for clarity of understanding the principle.

- The triangular membership functions for input variable speed error (e), change in speed error (ce) and control output (du) i.e. change in firing angle are shown in normalized units.

The general considerations in the design of the controller are:

- If both e and ce are zero, then maintain the present control setting i.e. du=0
- If e is not zero but is approaching this value at a satisfactory rate, then maintain the present control setting.
- If e is growing then change the control signal du depending on the magnitude and sign of e and ce to force e towards zero.
- e and ce are speed error and change of speed error respectively (normalized).

- du is the change in firing angle (normalized).
- Linguistic fuzzy sets are defined as below.

Z = zero
PS = Positive medium
PM = Positive Big
NS = Negative small
NM = Negative medium
NB = Negative Big

The table below shows the corresponding rule table for the speed controller.
The top row and left column of the matrix indicate the fuzzy sets of the variable e and ce respectively and the membership function of the body of the matrix.
There may be 7x7=49 possible rules in the matrix, where a typical rule reads as
IF e = PS and ce = NM then du = NS

du e ce	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

Rule Matrix for Fuzzy Speed Control

Figure 20

Identify the four valid rules from the rule matrix table for Z and PS values of e and ce. These are:

R1: IF e=Z AND ce= Z THEN du = Z
R2: IF e=Z AND ce= PS THEN du = PS
R3: IF e=PS AND ce= Z THEN du= PS
R4: IF e=PS AND ce= PS THEN du= PM

7.2.1 MEMBERSHIP FUNCTIONS:

7.2.1.1MEMBERSHIP FUNCTION FORe :

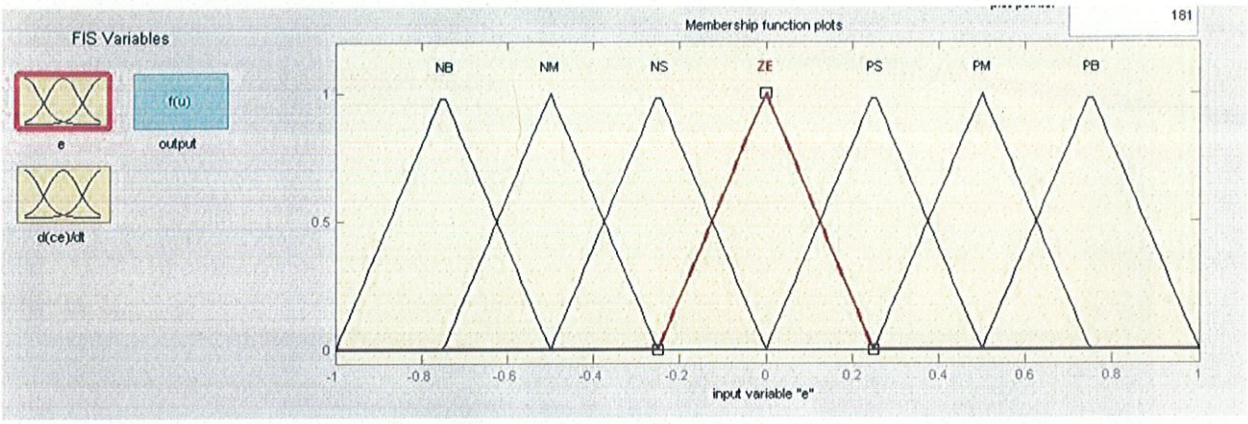


Figure 21

7.2.1.2 MEMBERSHIP FUNCTION FOR ce:

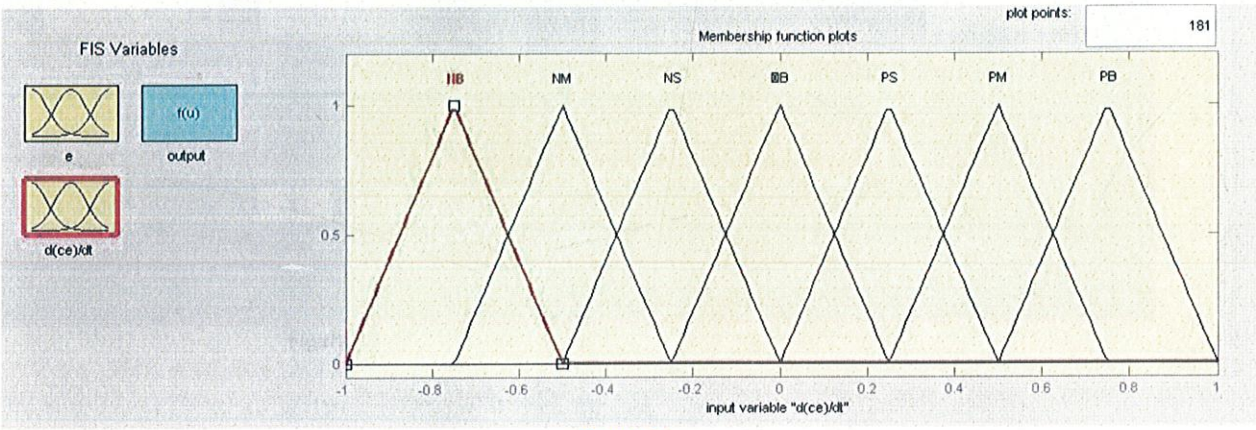
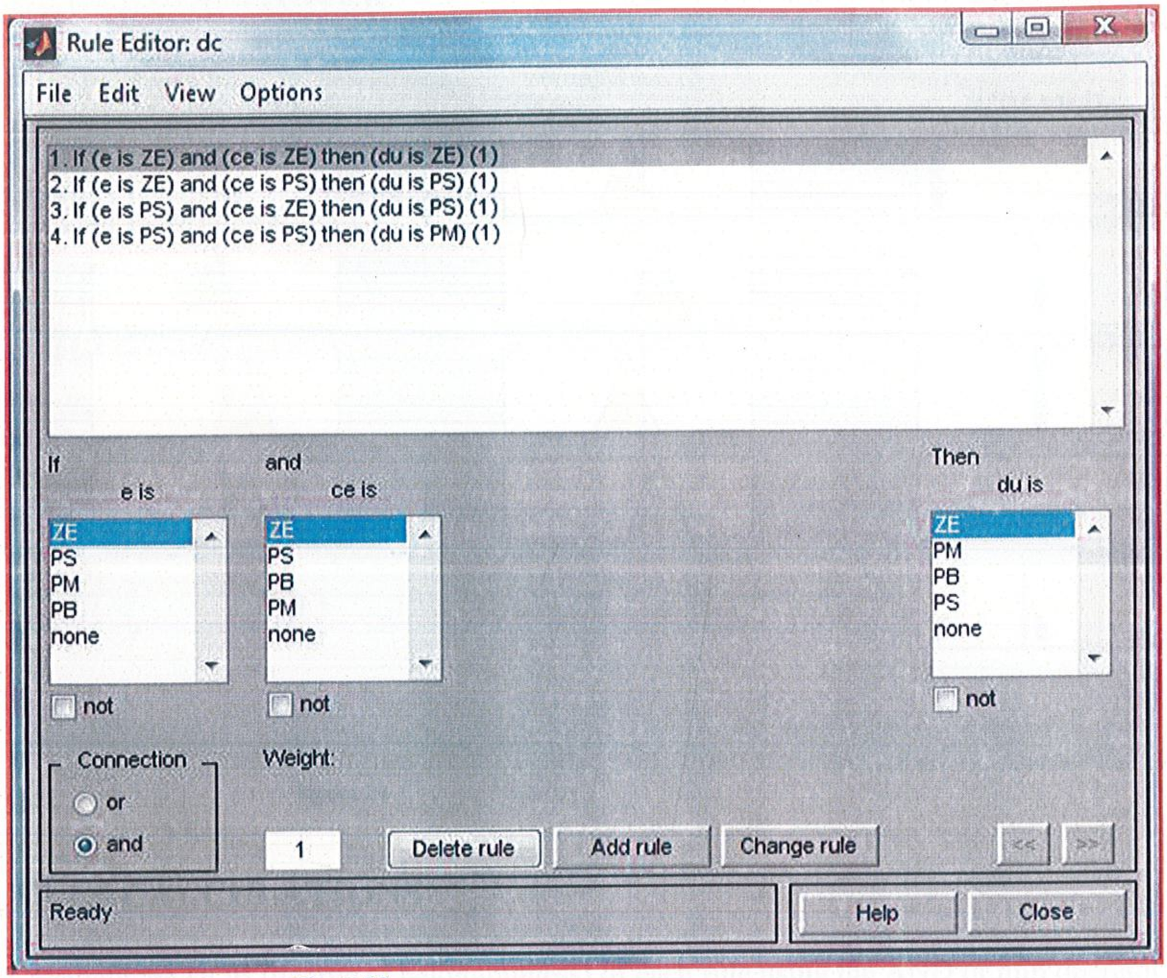


Figure 22

7.2.1.3 RULE EDITOR



1. If (e is ZE) and (ce is ZE) then (du is ZE) (1)
2. If (e is ZE) and (ce is PS) then (du is PS) (1)
3. If (e is PS) and (ce is ZE) then (du is PS) (1)
4. If (e is PS) and (ce is PS) then (du is PM) (1)

If e is ZE
and ce is ZE
Then du is ZE
Connection: ● and
Weight: 1
Ready

Figure 23

7.2.1.4 RULE VIEWER

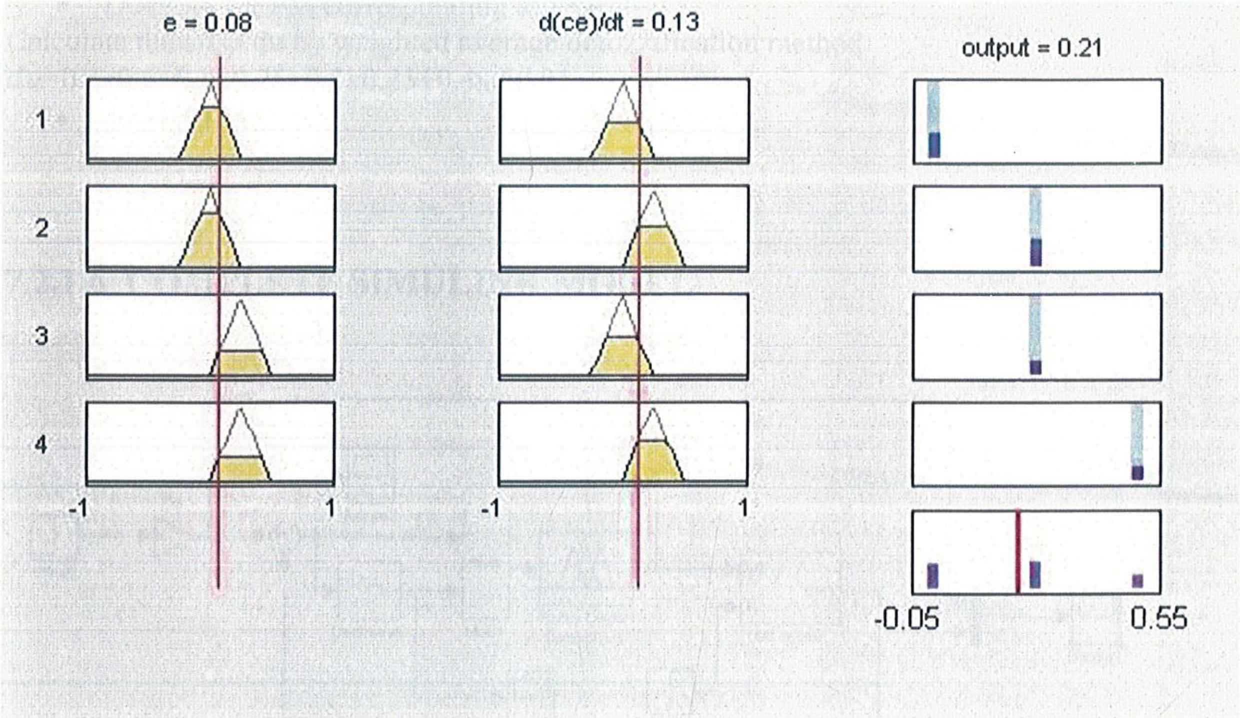


Figure 24

7.2.1.5 CALCULATIONS

Calculate the DOB (degree of belongingness) of each rule using the AND or min operator

- DOB1=min { μz(e), μz(ce)} = min {0.6,0.3}=0.3
- DOB2=min { μz(e), μPS(ce)} = min {0.6,0.7}=0.6
- DOB3=min { μPS(e), μz(ce)} = min {0.4,0.3}=0.3
- DOB4=min { μPS(e), μPS(ce)} = min {0.4,0.7}=0.4

Sample speed ωr* and ωr

Compute error e and change in error ce

- e(k) = ωr* – ωr
- ce(k) = e(k) – e(k-1)

From the figure e=0.008 and ce=0.13
Calculate the degree of membership of e and ce for the relevant fuzzy sets

- μz(e)=0.6 and μPS(e) =0.4
- μz(ce)=0.3 and μPS(ce) =0.7

Retrieve the amount of correction dui (i=1,2,3,4) corresponding to each rule in the table

- Du1=0 for Z corresponding to DOB1=0.3
- Du2=0.25 for PS corresponding to DOB2=0.6
- Du3=0.3 for PS corresponding to DOB3=0.3
- Du4=0.4 for PM corresponding to DOB4=0.4

Calculate the crisp du by weighted average defuzzification method
du=0.3x0.6+0.6x0.25+0.3x0.25+0.4x.50.21

- du= 0.21

7.2.1.6 COMPLETE SIMULINK MODEL

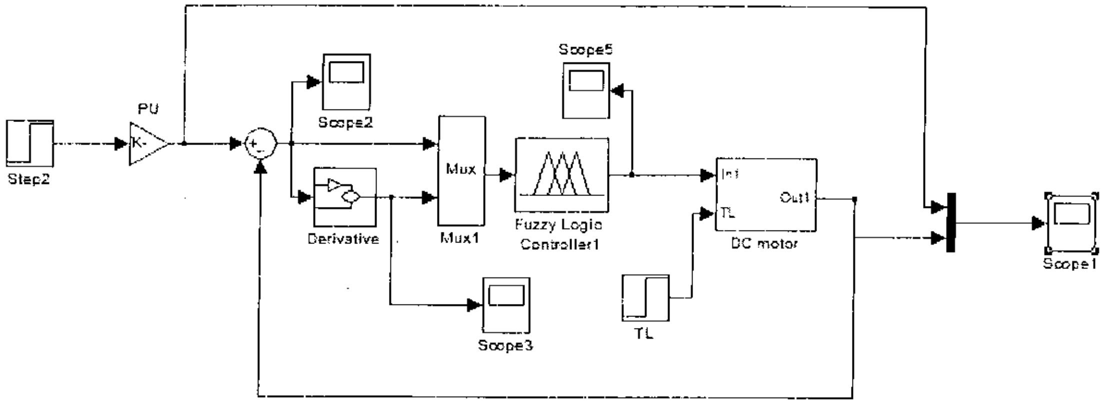


Figure 25

7.2.1.7 SIMULINK MODEL OF DC MOTOR

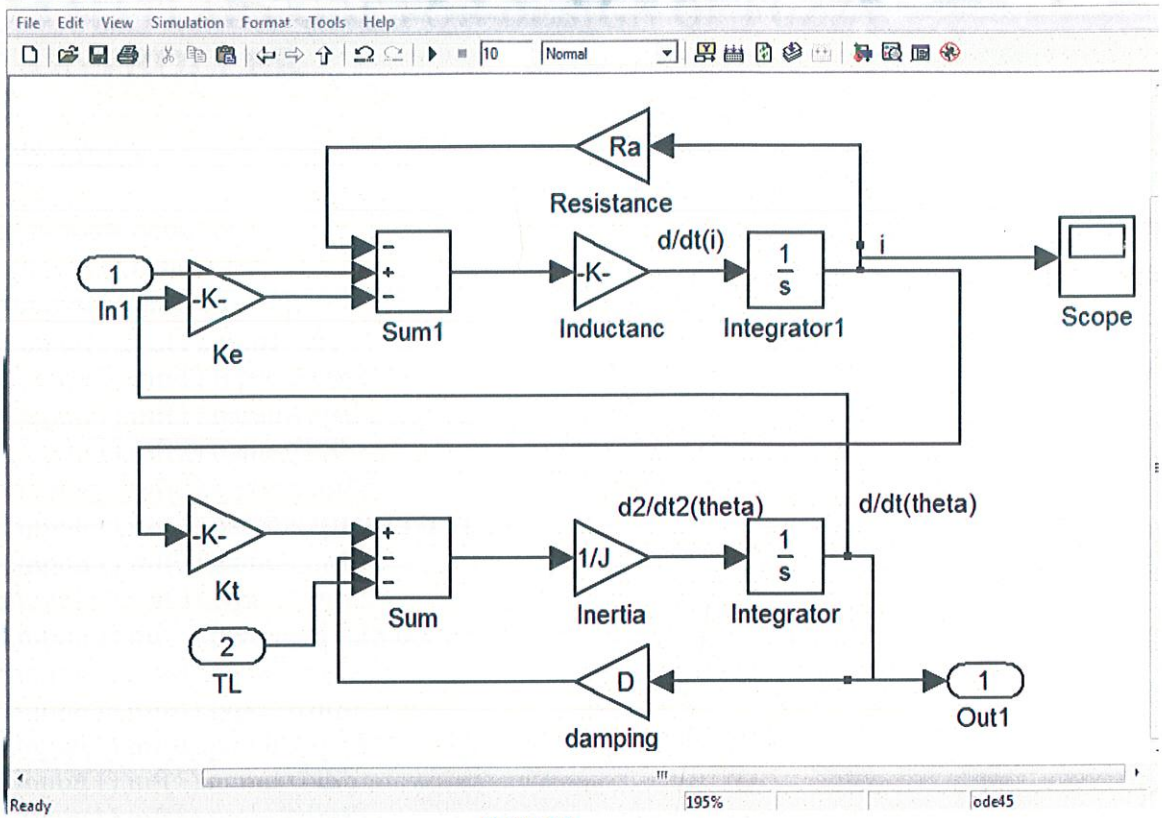


Figure 26

7.2.2.8 SIMULATION RESULTS

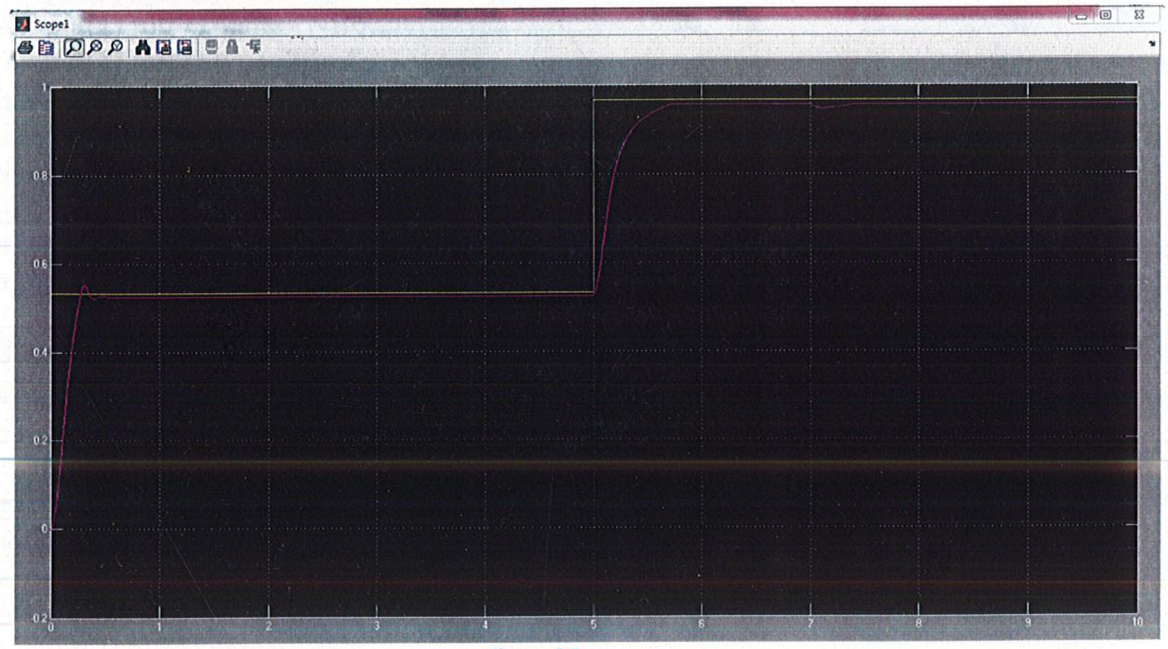


Figure 27

7.3 MATLAB CODE FOR DESIGN OF FUZZY CONTROLLER

FIS Editor

```
a=newfis('dcmotor');
a.input(1).name='e';
a.input(1).range=[-1 1];
a.input(1).mf(1).name='Z';
a.input(1).mf(1).type='trimf';
a.input(1).mf(1).params=[-0.25 0 -0.25];
a.input(1).mf(2).name='PS';
a.input(1).mf(2).type='trimf';
a.input(1).mf(2).params=[0 0.25 0.5];
a.input(1).mf(3).name='PM';
a.input(1).mf(3).type='trimf';
a.input(1).mf(3).params=[0.25 0.5 0.75];
a.input(1).mf(4).name='PB';
a.input(1).mf(4).type='trimf';
a.input(1).mf(4).params=[0.5 0.75 1];
a.input(1).mf(5).name='NS';
a.input(1).mf(5).type='trimf';
a.input(1).mf(5).params=[-0.5 -.25 0];
a.input(1).mf(6).name='NM';
a.input(1).mf(6).type='trimf';
a.input(1).mf(6).params=[-.75 -.5 -.25];
a.input(1).mf(7).name='NB';
a.input(1).mf(7).type='trimf';
a.input(1).mf(7).params=[-1 -.75 -.5];
a.input(2).name='ce';
a.input(2).range=[-1 1];
a.input(2).mf(1).name='Z';
a.input(2).mf(1).type='trimf';
a.input(2).mf(1).params=[-0.25 0 -0.25];
a.input(2).mf(2).name='PS';
a.input(2).mf(2).type='trimf';
a.input(2).mf(2).params=[0 0.25 0.5];
a.input(2).mf(3).name='PM';
a.input(2).mf(3).typc='trimf';
a.input(2).mf(3).params=[0.25 0.5 0.75];
a.input(2).mf(4).name='PB';
a.input(2).mf(4).type='trimf';
a.input(2).mf(4).params=[0.5 0.75 1];
a.input(2).mf(5).name='NS';
```

```
a.input(2).mf(5).type='trimf';
a.input(2).mf(5).params=[-0.5 -.25 0];
a.input(2).mf(6).name='NM';
a.input(2).mf(6).type='trimf';
a.input(2).mf(6).params=[-.75 -.5 -.25];
a.input(2).mf(7).name='NB';
a.input(2).mf(7).type='trimf';
a.input(2).mf(7).params=[-1 -.75 -.5];
a.output(1).name='du';
a.output(1).range=[-1 1];
a.output(1).mf(1).name='Z';
a.output(1).mf(1).type='trimf';
a.output(1).mf(1).params=[-0.25 0 -0.25];
a.output(1).mf(2).name='PS';
a.output(1).mf(2).type='trimf';
a.output(1).mf(2).params=[0 0.25 0.5];
a.output(1).mf(3).name='PM';
a.output(1).mf(3).type='trimf';
a.output(1).mf(3).params=[0.25 0.5 0.75];
a.output(1).mf(4).name='PB';
a.output(1).mf(4).type='trimf';
a.output(1).mf(4).params=[0.5 0.75 1];
a.output(1).mf(5).name='NS';
a.output(1).mf(5).type='trimf';
a.output(1).mf(5).params=[-0.5 -.25 0];
a.output(1).mf(6).name='NM';
a.output(1).mf(6).type='trimf';
a.output(1).mf(6).params=[-.75 -.5 -.25];
a.output(1).mf(7).name='NB';
a.output(1).mf(7).type='trimf';
a.output(1).mf(7).params=[-1 -.75 -.5];
writefis(fismat,'motor');
```

RULE Editor:

```
a.rule(1).antecedent=[1 1];
a.rule(1).consequent=[1];
a.rule(2).antecedent=[1 2];
a.rule(2).consequent=[2];
a.rule(3).antecedent=[2 1];
a.rule(3).consequent=[2];
a.rule(4).antecedent=[2 2];
a.rule(4).consequent=[3];
ruleview(a);
```

OUTPUT :

```
a = readfis('motor.fis');force=evalfis([0.08 0.13], a);
```

7.4 HARDWARE-IN LOOP SIMULATION USING UDP PROTOCOL

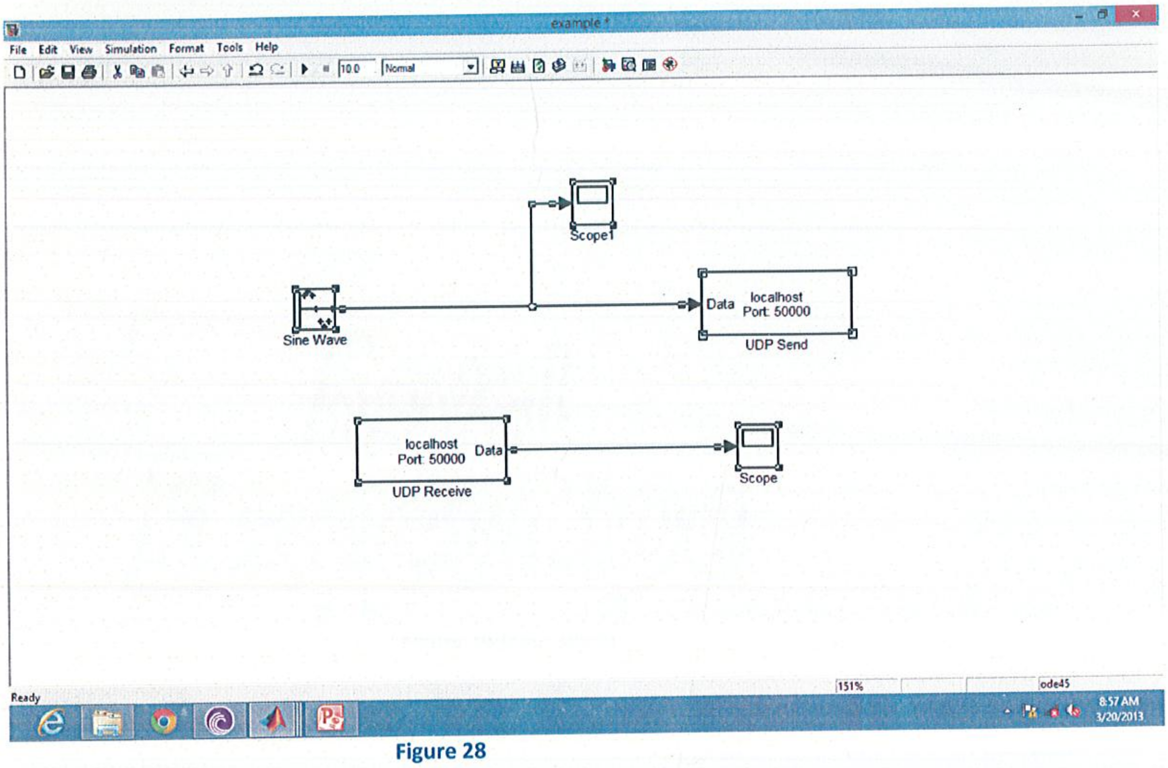


Figure 28

7.4.1 SIMULATION RESULTS

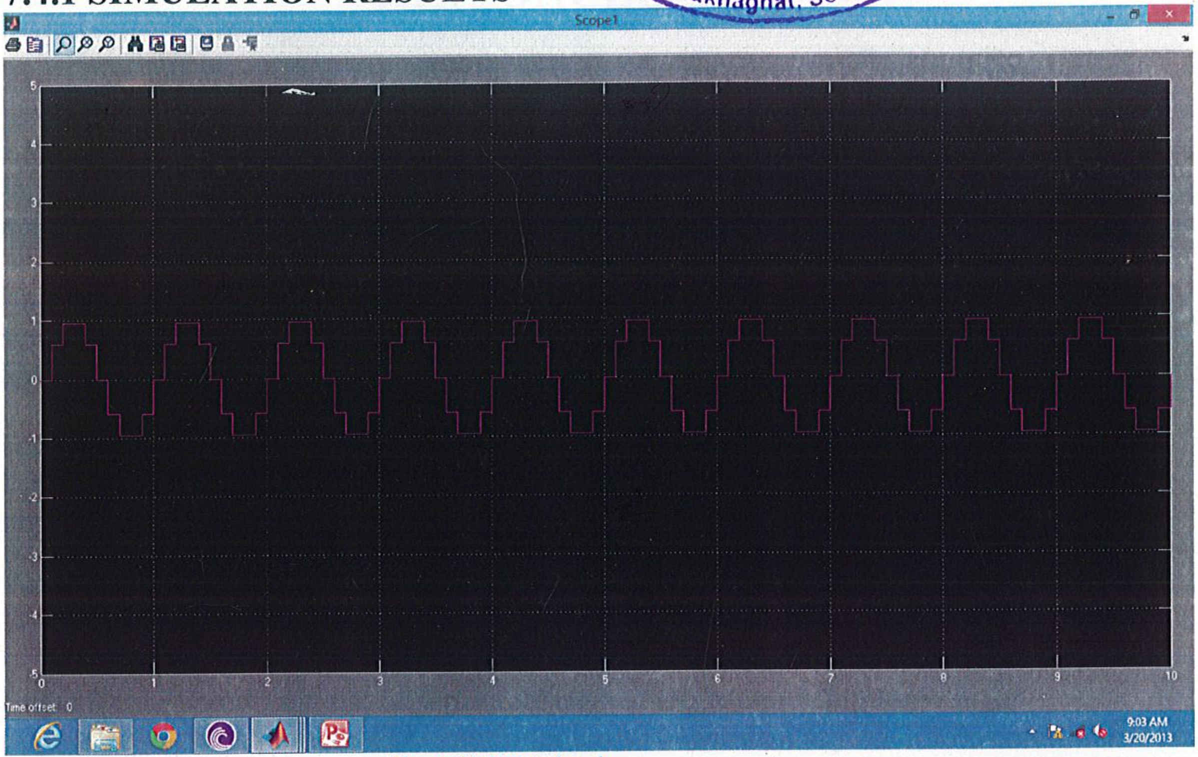


Figure 29:input signal

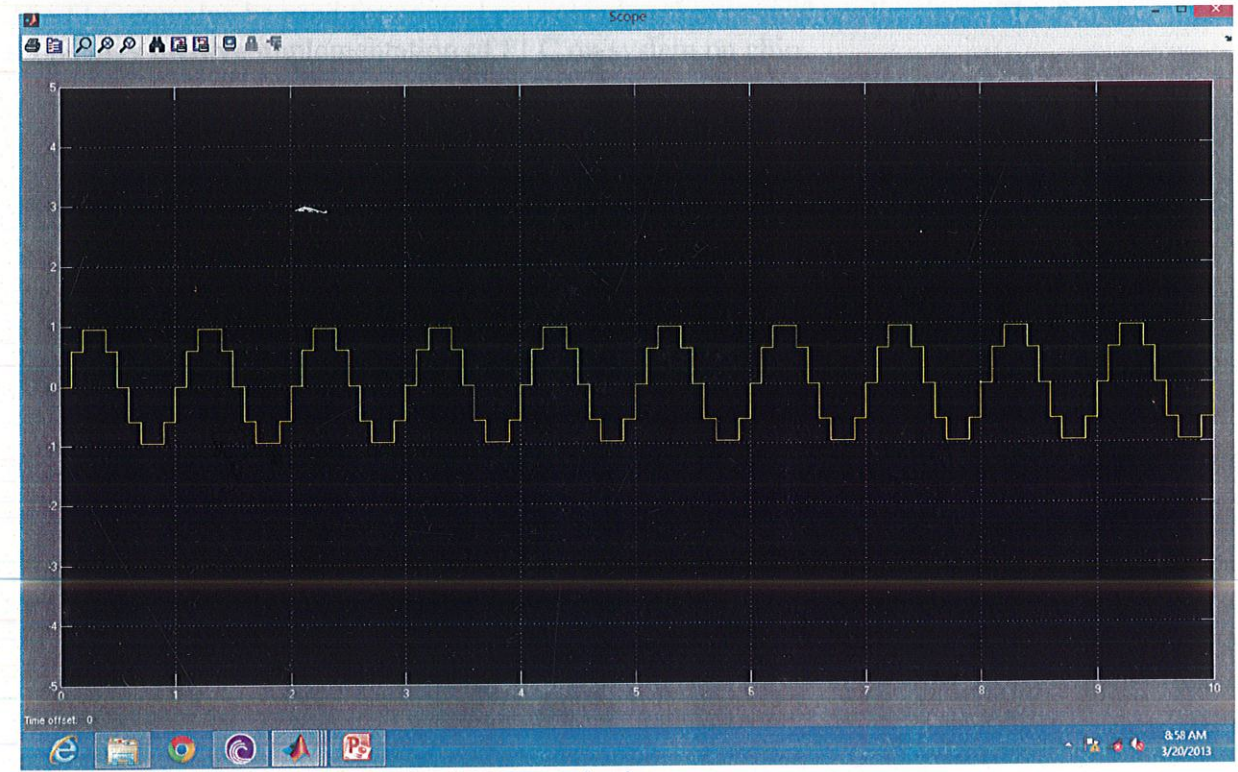


Figure 30:output signal

XII.CONCLUSION

In the simulation scenario, the direct structure of the networked DC motor control system is simulated using MATLAB/ SIMULINK under fully controlled environments for Fuzzy Logic Controller. We have studied basic definition and terminology of fuzzy logic with the help of Matlab. The results show that MATLAB paired with Simulink is a good simulation tool for modeling and analyze fuzzy logic controlled DC motor drives.

The background of DC Motors has been studied. The study of Characteristics of separately excited DC motor is done. The steady state operation and its various torque-speeds, torque-current characteristics of DC motor are studied.

Due to simple formulas and computational efficiency, both triangular MFs has been used to design fuzzy industrial controllers especially in real-time implementation.

The speed of a separately excited DC Motor has been successfully controlled by using fuzzy logic controller technique.

Graph for the speed response of separately excited DC Motor using fuzzy logic Controller has been plotted. FLC controller works better than any other PID controller. It is seen that the desired real speed and torque values could be reached in a short time by FLC controller.

The FLC can also be used for control purposes in other control applications. The future scope may involve implementation of FLC on system on chip.

XIII.FUTURE WORK

The further work will be concentrated on the improvement of fuzzy logic used to compensate the time delay, jitter and packet loss. Also the fuzzy control system can be extended by employing the adaptive fuzzy control which will be more independent of the variation in the motor parameter. The fuzzy system can also be tested for other types of existing membership functions that could give more smooth control.

In future scope Hardware-In loop simulation can be further developed in which we control the plant through a controller at distant places. This focuses on telepresence advantage of the NCS.

Also hardware implementation of this network can also be done.

XIV.REFERENCES:-

1. SIMULATIONS FOR TIME DELAY COMPENSATION IN NETWORKED CONTROL SYSTEM
   Sumit Vardhan, Student Member, IEEE, and Dr Rajiv Kumar, Member, IEEE
2. NETWORKED CONTROL SYSTEM: SURVEY AND DIRECTIONS
   Sachi Mittal, Anwar S Siddiqui
3. DESIGN AND SIMULATION OF AN INTELLIGENT TRAF CONTROL SYSTEM
4. PID CONTROLLER BASED CHOPPER-FED DC MOTOR DRIVE USING FUZZY LOGIC :Atul Kumar Dewangan*, Durga Sharma**, Shikha Mishra***
5. Neuro-Fuzzy Controller for Networked DC Motor Control
   B. Sharmila , N. Devarajan
6. PERFORMANCE OF NETWORKED DC MOTOR WITH FUZZY LOGIC CONTROLLER: B Sharmila1 and N Devarajan2
7. DC MOTOR CONTROL USING FUZZY LOGIC CONTROLLER: Rahul Malhotra and Tejbeer Kaur
8. PID Controller Based Chopper-Fed DC Motor Drive Using Fuzzy Logic Atul Kumar Dewangan, Durga Sharma, Shikha Mishra
9. Design and implementation of Fuzzy logic controller for a DC motor : Salma Khan, Prof. Uma Kulkarni
10. SPEED CONTROL OF DC MOTOR BY FUZZY CONTROLLER: Nagendra singh
11. SPEED CONTROL OF A SEPARATELY EXCITED DC MOTOR USING FUZZY LOGIC CONTROL BASED ON MATLAB SIMULATION PROGRAM: Atul Kumar Dewangan, Sashai Shukla, Vinod Yadu
12. REALIZATION OF FUZZY LOGIC CONTROLLED BRUSHLESS DC MOTOR DRIVES USING MATLAB/SIMULINK:Mehmet Cunkas, Omer Aydoğdu
13. TIME DELAY COMPENSATION SCHEMES WITH APPLICATION TO NETWORKED CONTROL SYSTEM: BOLLEPALLY RAJU
14. www.wikipedia.com
15. www.mathworks.in
16. www.google.com