FUZZY LOGIC FOR SPEED CONTROLLING OF STEPPER MOTOR

ANCHAL KATIYAR

M.Tech (VLSI) Scholar, Electronics and communication Engineering, Jayoti Vidyapeeth Women’s University, Jaipur, Rajasthan, India, E-mail: anchalkatiyar9@gmail.com

ABSTRACT

Stepping motors are widely used in robotics and in the numerical control of machine tools where they have to perform high-precision positioning operations. Nevertheless, the variations of the mechanical configuration of the drive, which are common to these two applications, can lead to a loss of synchronism for high stepping rates. Moreover, the classical open-loop speed control is weak and a closed-loop control becomes necessary. In this paper, the fuzzy logic principle is applied to control the speed of a stepping motor drive with feedback. Speed control of DC Motor is vital in many applications. In this paper, an effort has been made to control the speed of the DC motor using fuzzy logic control (FLC) based on Lab VIEW (Laboratory Virtual Instrument Engineering Workbench) program. Lab VIEW provides a graphical programming environment suited for high-level or system-level design. The fuzzy logic controller designed to apply the control voltage that sent to dc motor based on fuzzy rule base of motor speed error (e) and change of speed error (Ce). In this paper results of FLC, PI and PID Controller are compared. The simulation results demonstrate that the response of DC motor with FLC show a satisfactory well damped control performance.

Key words: Conventional controller, FLC, VHDL, FPGA

1. INTRODUCTION

Stepper motors (SM) are widely used in precision positioning applications such as robotic positioning systems, tracking and production lines. SM can be driven by digital and open loop control system. In open loop control, the speed response of the SM suffers from large overshoot, oscillatory response and long settling time. Additionally, the motor must respond to each excitation change. If the excitation changes are made too quickly, the stepper motor may lose some steps and therefore it will be unable to move the rotor to the new demanded position. Therefore, a permanent error can be introduced between the load position and that expected by the controller. Due to these limitations, the stepper motor cannot be used without feedback sensor and closed loop control system with high performance applications where the exact position or rotor speed is required. In the present paper, the PID control system is used to enhance the properties of the open loop control system of the hybrid stepper motor speed for a given reference input. Also, instead of using pulse train to drive the stepper motor, a microstepping technique, which consists of sine and cosine signals, is used. The performance of the proposed control system is tested by applying a sudden load disturbance and compared with that obtained from the open loop control system. Stepper motors provide a means for precise positioning and speed control without the use of feedback sensors. The basic operation of a stepper motor allows the shaft to move a precise number of degrees each time a pulse of electricity is sent to the motor. Since the shaft of the motor moves only the number of degrees that it was designed for when each pulse is delivered, you can control the pulses that are sent and control the positioning and speed. The rotor of the motor produces torque from the interaction between the magnetic field in the stator and rotor. The strength of the magnetic fields is proportional to the amount of current sent to the stator and the number of turns in the windings.

1.1 FLC FOR BRUSHLESS SERVO DRIVE

Fuzzy control structure that was implemented for trajectory tracking control of a brushless servo drive system. The control structure employs a fuzzy logic controller incorporating an H∞ tracking controller via an acceleration feedback signal. The fuzzy logic controller is equipped with an adaptive-law-based Lyapunov synthesis approach to compensate for system uncertainty and random changes in the external load acting on the drive system. The proposed control structure is experimentally verified on a state-of-the-art d SPACE DS1104 digital signal processor (DSP)-based data acquisition and control system in a laboratory 1-hp brushless drive system. The controllers are first designed in Simulink. Then, the Real-Time Workshop is used to automatically generate optimized C code for real-time applications. Afterward, the interface between MATLAB/Simulink and the dSPACE DS 1104 allows the control algorithm to run on the hardware processor of the DSP. The result is a powerful testbed for the rapid design and implementation of the hybrid tracking controllers for a wide variety of operating conditions. The implementation of the fuzzy logic controller is very straightforward by coding each component of the fuzzy inference system in VHDL according to the design specifications. The design of the FLC is highly flexible as the membership functions and rule base can be easily changed. Moreover the performance of FLC is compared with PI controller for set point change and load change. The performance FLC was much superior to conventional PI controller and FLC is able to compensate load changes better than PI controller. By simply changing some parameters in the codes and design constraint on the specific synthesis tool, one can experiment with different design circuitry.
to get the best result in order to satisfy the system requirements. The FLC can also be used for control purposes in other applications.

1.2 FLC FOR CURRENT MOTORS

The speed of the motor has to be controlled, which in turn controls the vehicle dynamics to run the vehicle. So, the main aim is to determine the motor speed, which drives the vehicle. In this regard, parameters such as acceleration, braking, energy status, gear and terrain are considered. This system, which functions as a closed loop system, also takes the motor speed as a reference along with the above-mentioned parameters to estimate the variation of the motor speed. They found that the real-time implementation of the fuzzy logic controller for the various driving conditions and terrains has been achieved on a Xilinx Spartan 2E FPGA using VHDL. This controller has been implemented for four different types of terrains. The implementation of this controller for the fuzzy module along with other blocks such as ADC and LCD in the closed-loop control system of a stepper motor is performed. Further progress in the controller is, in the development of controller for battery monitoring that could alarm the state of charge (SOC) and time of recharge, that work presented for a novel fuzzy logic controller (FLC)-based wide-speed-range operation of interior permanent magnet synchronous motor (IPMSM) drives. The proposed FLC is designed in such a way that it can simultaneously control both torque and flux of the motor while maintaining current and voltage constraints. Thus, a stand-alone FLC is utilized; whose outputs are d- and q-axis currents. The proposed FLC is designed based on conventional maximum torque per ampere operation below the rated speed and the field-weakening operation above the rated speed. The complete IPMSM drive is experimentally implemented, utilizing dSPACE DSP board DS1104 for a prototype 5-hp motor. The performance of the proposed drive is tested both in simulation and experiment at different operating conditions.

1.3 FUZZY VS. CONVENTIONAL CONTROL

In order to design a conventional controller for controlling a physical system, the mathematical model of the system is needed. A common form of the system model is differential equations for continuous-time systems or difference equations for discrete-time systems. Unless physical insight and the laws of physics can be applied, establishing an accurate nonlinear model using measurement data and system identification methods is difficult in practice. Even if a relatively accurate model of a dynamic system can be developed, it is often too complex to use in controller development, especially for many conventional control design procedures that require restrictive assumptions for the plant. As an alternative, fuzzy control provides a formal methodology for representing, and implementing a human’s heuristic knowledge about how to control a system, which may provide a new paradigm for nonlinear systems. Fuzzy controller is unique in its ability to utilize both qualitative and quantitative information. Qualitative information is gathered not only from the expert operator strategy, but also from the common knowledge. Fuzzy control should not be employed if the system to be controlled is linear, regardless of the availability of its model. PID control and various other types of linear controllers can effectively solve the control problem with significantly less effort, time, and cost.

The benefits of fuzzy controllers could be summarized as follows:

1. Fuzzy controllers are more robust than PID controllers because they can cover a much wider range of Operating conditions than PID can, and can operate with noise and disturbances of different nature.

2. Developing a fuzzy controller is cheaper than developing a model-based or other controller to do the same.

1.4 FLC APPLICATIONS

Large numbers of fuzzy control applications with the physical systems require a real-time operation to interface high speed constraints; higher density programmable logic devices such as field programmable gate array (FPGA) can be used to integrate large amounts of logic in a single IC. This work reviews the state of the art of FPGA with the focus on FPGA-based fuzzy logic controller. The work starts with an overview of FPGA in order to get an idea about FPGA architecture, and followed by an explanation on the hardware implementation with both type analogue and digital implementation, a comparison between fuzzy and conventional controller also provided in this work. A survey on fuzzy logic controller structure is highlighted in this article with the focus on FPGA-based design of fuzzy logic controller with different applications. Finally, they provided the simulation and experimental results form the literature and concluded the main differences between software-based systems with respect to FPGA-based systems, and the main features for FPGA technology and its real-time applications.

2. METHODOLOGY

2.1 INTRODUCTION TO FUZZY LOGIC CONTROLLER

Fuzzy logic has rapidly become one of the most successful of today’s technology for developing sophisticated control system. With it aid complex requirement so may be implemented in amazingly simple, easily minted and inexpensive controllers. The past few years have witnessed a rapid growth in number and variety of application of fuzzy logic. The application range from consumer products such as cameras, camcorder, washing machines and microwave ovens to industrial process control, medical instrumentation, and decision support system. Many decision-making and problem solving tasks are too complex to be understand quantitatively however, people succeed by using knowledge that is imprecise rather than precise. Fuzzy logic is all about the relative importance of precision .fuzzy logic has two different meanings in narrow senses, fuzzy logic is a logical system which is an extension of multi valued logic .but in wider sense fuzzy logic is synonymous with the theory of fuzzy sets. Fuzzy set theory is originally introduced by Lotfi Zadeh in the
1960's resembles approximate reasoning in its use of approximate information and uncertainty to generate decisions.

Several studies show, both in simulations and experimental results, that Fuzzy Logic control yields superior results with respect to those obtained by conventional control algorithms. Thus, in industrial electronics the FLC control has become an attractive solution in controlling the electrical motor drives with large parameter variations like machine tools and robots. However, the FLC controllers design and tuning process is often complex because several quantities, such as membership functions, control rules, input and output gains, etc, must be adjusted. The design process of a FLC can be simplified if some of the mentioned quantities are obtained from the parameters of a given Proportional-Integral controller (PIC) for the same application. Fig 2.1 shows the block diagram of Fuzzy Logic Controller.

Fig 2.1: Block diagram of Fuzzy Logic Controller

2.2 FUZZY LOGIC CONTROLLER

Fuzzy logic expressed operational laws in linguistic terms instead of mathematical equations. Many systems are too complex to model accurately, even with complex mathematical equations; therefore traditional methods become infeasible in these systems. However, fuzzy logic's linguistic terms provide a feasible method for defining the operational characteristics of such systems.

Fuzzy logic controller can be considered as a special class of symbolic controller. The configuration of fuzzy logic controller block diagram is shown in Fig.2.2

![Fig 2.2: Structure of Fuzzy logic controller]

The fuzzy logic controller has three main components

1. Fuzzification
2. Fuzzy inference
3. Defuzzification

2.1.1. 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 following functions:

1. Multiple measured crisp inputs first must be mapped into fuzzy membership function this process is called fuzzification.
2. Performs a scale mapping that transfers the range of values of input variables into corresponding universes of discourse.
3. Performs the function of fuzzification that converts input data into suitable linguistic values which may be viewed as labels of fuzzy sets.

Fuzzy logic linguistic terms are often expressed in the form of logical implication, such as if then rules. These rules define a range of values known as fuzzy membership functions. Fuzzy membership function may be in the form of a triangular, a trapezoidal, a bell or another appropriate form.

The inputs of the fuzzy controller are expressed in several linguist levels. As shown in Fig.2.3 these levels can be described as Positive big (PB), Positive medium (PM), Positive small (PS) Negative small (NS), Negative medium (NM), Negative big (NB) or in other levels. Each level is described by fuzzy set.

![Fig 2.3: Seven levels of fuzzy membership function]

2.1.2. Fuzzy inference

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. There are two types of fuzzy inference systems that can be implemented in the Fuzzy Logic Toolbox: Mamdani-type and Sugeno-type. These two types of inference systems vary somewhat in the way outputs are determined. Fuzzy inference systems have been successfully applied...
in fields such as automatic control, data classification, decision analysis, expert systems, and computer vision. Because of its multidisciplinary nature, fuzzy inference systems are associated with a number of names, such as fuzzy-rule-based systems, fuzzy expert systems, fuzzy modeling, fuzzy associative memory, fuzzy logic controllers, and simply (and ambiguously) fuzzy Mamdani’s fuzzy inference method is the most commonly seen fuzzy methodology.

Mamdani’s method was among the first control systems built using fuzzy set theory. It was proposed in 1975 by Ebrahim Mamdani as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators. Mamdani’s effort was based on Lotfi Zadeh’s 1973 paper on fuzzy algorithms for complex systems and decision processes.

The second phase of the fuzzy logic controller is its fuzzy inference where the knowledge base and decision making logic reside. The rule base and data base from the knowledge base. The data base contains the description of the input and output variables. The decision making logic evaluates the control rules. The control-rule base can be developed to relate the output action of the controller to the obtained inputs.

2.1.3. Defuzzification

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. The output of the inference mechanism is fuzzy output variables. The fuzzy logic controller must convert its internal fuzzy output variables into crisp values so that the actual system can use these variables. This conversion is called defuzzification. One may perform this operation in several ways.

$$x = \frac{\int \mu(x)dx}{\int \mu(x)dx}$$

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.

3. FUZZY LOGIC CONTROL OF THE DC MOTOR

The fuzzy logic controller was applied to the speed loop by replacing the classical polarization index (PI) controller. The fuzzy logic controlled BDCM drive system block diagram is shown in Fig 3.1.

![Overall block diagram of speed control of DC motor](image-url)
There are a number of reasons for using fuzzy logic in the speed control of DC motor, the primary advantage being the flexibility offered by fuzzy logic. The backbone of any FLC is embodied in a set of fuzzy rules, with two implications:

The fact that the control strategy is represented by a set of rules and not an elaborated set of equations. This allows the designer to change the basic characteristics of the Controller with minimal fuss, simply by redefining the rules.

The fuzzy aspect of the rules, deals with the imprecise definition of the system. This allows vagueness in the design of the control system to be tolerated to a certain degree and eliminates the need for a well-defined mathematical model of the plant. The following sections describe the development of FLC for speed control of DC motor. The present design utilizes three types of membership functions – Γ-function, L-function and Λ-function. These functions have been proven to produce good results for control applications and can be easily implemented into hardware. The universe of discourse of the input variables is partitioned into five fuzzy sets or linguistic values (B1 to B5), while the output variable can take any of the nine linguistic values (D1 to D9). Graphical representations of the membership functions are shown in Fig 3.2. and Fig 3.3.

The crisp values of the input variables are mapped onto the fuzzy plane using the equations above. It gives each input variable a membership function relating to the fuzzy sets, (B1 i to B5 i) . It has to be pointed out that in these equations, B i j is used to denote the linguistic value as well as membership function, while membership function using μB i j (x i ) . The universe of discourse of the output variable is divided into nine linguistic Values. The membership functions of the output values are intentionally made to be symmetrical, as this will simplify the defuzzification computation. E1 to E9 are the mean of each function and act as the weightings to

the weighted average method of defuzzification. Each input variable can take any of the five linguistic values, therefore 25 (= 5 X 5) rules are formulated. The rules have the typical fuzzy rule structure, using linguistic variables in both the antecedent and consequent, and are expressed in IF-THEN manner. They map the input states onto 25 output conditions (C1 to C25). The fuzzy rules have the general form,

\[ R^K : \text{IF} \ x_1 \text{is} A^{k_1}_{i_1} \text{AND} x_2 \text{is} A^{k_2}_{i_2} \text{THEN} \ y \text{is} C^K \]

Then the rule base can be represented by a fuzzy associative memory (FAM) table (Table 3.1). The FLC design in this work incorporates Mamdani’s implication method of inference, which is one of the most popular methods in fuzzy control applications. In essence, Mamdani’s implication for the fuzzy rule of (1) is given by

\[ \mu_C(y) = \max_k [\min(\mu_{A^k_{i_1}}(x_1), \mu_{A^k_{i_2}}(x_2))] \quad k = 1,2,\ldots,25 \]

The implication has a simple min–max structure which makes it easy to incorporate into hardware. The block diagram in Fig.3.3 provides an overview of the controller’s internal structure. Two input variables are fuzzyfied, producing the corresponding linguistic values and membership functions (Bi j). The first phase of Mamdani’s implication involves min-operation since the antecedent pairs in the rule structure are connected by a logical ‘AND’. All the rules are then aggregated using a max-operation.

The dominant criteria in the selection of defuzzification technique lie in the implementation stage. The weighted average method is viewed to be an appropriate technique for systems involving hardware implementation. Due to the fact that the output membership functions are symmetrical in nature, the mean of the fuzzy sets can be used as weightings for the defuzzification process. This technique requires several multiplyby- a-constant operations and only one division process. Rule base moulds the functionality of an FLC. The rules are most likely to be formulated based on some level of human understanding of the plant. The
design of the FLC in this paper is based on PI controllers. The rule base is constructed from the control law of a PI system.

3.1 PI CONTROL

The proportional–integral (PI) controller is a well-known system in control engineering. It is, in essence, a lag compensator characterized by the transfer function

\[ G(s) = K \left( 1 + \frac{1}{Ts} \right) \]

Where, \( G(s) \) is the gain; \( K \) is the control parameter; \( T \) is the time constant.

The control law is given by the equation

\[ u_{\text{PI}} = K_p e + K_i \int_0^t e \, dt \]

(3)

Where, \( u \) is the control signal; \( e \) is the error, given by \( e = \) (input value) – (reference value).

Differentiating (3) gives

\[ \frac{du}{dt} = K_i \frac{de}{dt} + K_p e \]

(4)

In discrete-time systems, (4) can be written as

\[ u(kt) - u((kt-T)) = K_p \cdot (e(kt) - e((kt-T))) + K_i \cdot e(kt) \]

\[ \Delta u = K_p \Delta e + K_i e \]

(5)

Where,

\( \Delta u \) is the change in \( u \) over one sampling period

\( \Delta e \) is the change in \( e \) over one sampling period.

The values of controller parameters are found from Z-H tuning method. From the values found are \( K_p=0.12 \) and \( K_i = 0.0201 \). The characteristic of a PI controller can be represented by the phase plane diagram shown in Fig. 3.8. A diagonal line where \( u = 0 \) divides the area where \( u \) is positive and \( u \) is negative.

![Fig 3.4: Characteristic of PI controller](image)

At this stage, the control law in (8) is not in fuzzy terms. In order to design a fuzzy controller based on the PI control structure, the following definitions are made:

Let \( E \) be the linguistic variable for the error \( e \)

\( \Delta E \) be the linguistic variable for the change of error \( \Delta e \)

\( U \) be the linguistic variable for the control output \( u \).

The maximum range of motor is +/- 1500 rpm. The possible error range is -750 to 750. The universe of Discourse of the change in error is based on experiment data from conventional controller which gives the range of error is +/- 15. Output of the controller ranges from +/- 8. Linguistic variables for \( E, \Delta E, U \) are defined as

\[ LE = \{ \text{Negative Big}(-750), \text{Negative}(-500), \text{Zero}(0), \text{Positive}(500), \text{Positive Big}(750) \} \]

\[ L\Delta E = \{ \text{Negative Big}(-150), \text{Negative}(-75), \text{Zero}(0), \text{Positive}(75), \text{Positive Big}(150) \} \]

\[ LU = \{ \text{Negative Very Big}(-8), \text{Negative Big}(-6), \text{Negative Small}(-4), \text{Negative Small}(-2), \text{Zero}(0), \text{Positive Small}(2), \text{Positive Big}(6), \text{Positive Very Big}(8) \} \]

The Corresponding PI control law in IF-THEN rules has form:

\[ RK : \text{IF} \ X_1 \text{ IS} A_{k1} \text{ AND} X_2 \text{ IS} A_{k2}, \text{THEN} Y \text{ IS} C_k \]

Where,

\( A_{k1} \) can take any linguistic value in the set \( LE \)

\( A_{k2} \) can take any linguistic value in set \( L\Delta E \)

\( C_k \) can take any linguistic value in the set \( LU \)

To implement this design into the FLC, let:

- \( x_1 = E \)
- \( x_2 = \Delta E \)
- \( \{ B_1, B_2, B_3, B_4, B_5 \} = \{ \text{Negative Big}(NB), \text{Negative}(N), \text{Zero}(Z), \text{Positive}(P), \text{Positive Big}(PB) \} \) for \( i = 1,2 \)
- \( \{ D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8 \} = \{ \text{Negative Very Big}(NVB), \text{Negative Big}(NB), \text{Negative}(N), \text{Negative Small}(NS), \text{Zero}(Z), \text{Positive Small}(PS), \text{Positive}(P), \text{Positive Big}(PB), \text{Positive Very Big}(PVB) \} \)

Table 3.1: FAM table for FLC design (Rule set)

![Table 3.1: FAM table for FLC design (Rule set)](image)

Figure 3.5 shows a block diagram demonstrating the implementation of the FLC in a Speed control of DC motor. In this application, the input interface converts...
the output of the speed sensor into error and change of error which are used as the two inputs to the FLC.

![Overall block diagram of FLC based control system for DC motor](image)

**Fig 3.5: Overall block diagram of FLC based control system for DC motor**

Another interface converts the output into the required value for the plant. The characteristics of the interfacing blocks can be described by the following equations:

**Input interface:**

\[ e = V_{REF} - V_c; \quad x_1 = e \quad ; \quad x_2 = x_1 - x_1z_1 \]

**Output interface:**

\[ \Delta u = y; \quad u = \Delta u + uz_1 \]

4. SIMULATION AND SYNTHESIS

4.1 SIMULATION RESULTS

In this work we have implemented fuzzy logic controller for speed control of DC motors. Here we check input range of voltage reference to get the desired voltage output. There is some another inputs reset and clock. Our project will work on positive edge of clock and low reset. If reset is high all output will be zero. Here we take input range (-8 to -6) volt we will get output voltage -8 volt fig 4.1 shows the simulation result for -8 to -6 volts.

![Simulation result for -8 to -6 volts](image)

**Fig 4.1: simulation Result for -8 to -6 volts.**

For input range (-6 to -4) volt we will get output voltage -6 volt fig 4.2 shows the simulation result for -6 to -4 volts.

![Simulation result for -6 to -4 volts](image)

**Fig 4.2: simulation Result for -6 to -4 volts.**
For input range (-4 to -2) volt we will get output voltage -4 volt fig 4.3 shows the simulation result for -4 to -2 volts.

Fig 4.3: simulation Result for -6 to -4 volts.

For input range (-2 to 0) volt we will get output voltage -2 volt fig 4.4 shows the simulation result for -2 to 0 volts.

Fig 4.4: simulation Result for -2 to 0 volts

For input range (0 to 2) volt we will get output voltage 2 volt fig 4.5 shows the simulation result for 0 to 2 volts.

Fig 4.5: simulation Result for 0 to 2 volts
For input range (2 to 4) volt we will get output voltage 4 volt fig 4.6 shows the simulation result for 2 to 4 volts.

Fig 4.6: simulation Result for 2 to 4 volts

For input range (4 to 6) volt we will get output voltage 6 volt fig 4.7 shows the simulation result for 4 to 6 volts.

Fig 4.7: simulation Result for 4 to 6 volts

For input range (6 to 8) volt we will get output voltage 8 volt fig 4.8 shows the simulation result for 6 to 8 volts.

Fig 4.8: simulation Result for 6 to 8 volts
4.2 SYNTHESIS RESULT

Fig 4.9 shows the RTL of our code, fig 4.10 shows the internal RTL and fig 4.11 shows the devices utilized in our work. Here we attach the synthesis report of our code.

5. FUTURE EXPECTS

The dc motor is used to drive the load and stepper motor change the loads according to the speed of dc motor. Fuzzy logic controller has two inference system: one is mamdani and another one is sugeno. Sugeno type inference system used when linear relationship exists between input and output. Thus in future we can Simulink another model for stepper motor control using fuzzy logic by using sugeno inference system. There is a feature in stepper motor Simulink model that we can initially set the rotor position. If we do so for specific gear angle of rotation of stepper motor may change.

6. CONCLUSION

This paper named “FUZZY LOGIC FOR SPEED CONTROLLING OF STEPPER MOTOR” undertook by the student of M.Tech (VERI LARGE SCALE INTEREGERATION) FOURTH SEMESTER under the guidance and support of our teacher. The simulation results show that the proposed control strategy operates robustly under modeling uncertainty, with a good dynamic performance. The results of experiment demonstrate that the proposed fuzzy logic controller is sensitive to variation of the reference speed. Moreover the performance of FLC is compared with PI controller for set point change and load change. The performance of FLC was much superior to conventional PI controller and FLC is able to compensate load changes better than PI controller.

REFERENCE


4. Introduction to fuzzy logic using matlab by Sivanandam, S. Sumathi.