To Observe Stability in the Operation of DC Motor by Incorporating Uncertainty Through Fuzzy Logic

A Thesis report

submitted towards the partial fulfillment of

the requirements for the award of degree of

Master of Engineering

in

Electronic Instrumentation and Control Engineering

Submitted by

Harsimran Jit Singh

Roll. No: 800851008

Under Guidance of

Ms. GAGANDEEP KAUR
Assistant Professor, EIED
Thapar University, Patiala
&
Mr. AJAY KAKKAR
Assistant Professor, ECED
Thapar University, Patiala

Electrical & Instrumentation Engineering Department
THAPAR UNIVERSITY
PATIALA-147004
June- 2010

Certificate

This is to certify that the work presented in this thesis entitled **"To observe stability in the operation of DC motor by incorporating uncertainty through fuzzy logic"** submitted in the partial fulfillment of the requirement for the award of the degree of **Master of Engineering** in **Electronics Instrumentation And Control Engineering** at **Thapar University, Patiala**, is an original record under the supervision and guidance of **Mrs. Gagandeep Kaur, Assistant Professor**. The matter embodied in this report has not been submitted anywhere for the award of any degree.

Date: 05-July-2010
Place: Thapar University, Patiala

(HARSIMRANJIT SINGH)

I CERTIFIED THAT THE ABOVE STATEMENT MADE BY THE STUDENT IS CORRECT TO THE BEST OF OUR KNOWLEDGE AND BELIEF.

Mr. Ajay Kakkar
Assistant Professor, ECED
Thapar University, Patiala

Mrs. Gagandeep Kaur
Assistant Professor, EIED
Thapar University, Patiala

S. Ghosh
Dr. Smarajit Ghosh
Professor& Head, EIED
Thapar University, Patiala

13.7.10
Dr. R.K. Sharma
Dean of Academic Affairs
Thapar University, Patiala

Acknowledgement

I would like to express my deepest thanks to all those people who influenced my work. I would like to add a few heartfelt words for the people who gave their unending support with their unfair humor and warm wishes.

Firstly, I express my sincere thanks to **Dr. Abhijeet Mukherji, Director**, Thapar University, Patiala.

I am feeling greatly obliged by **Dr. Smarajit Ghosh,** Head, Electrical and Instrumentation Engineering Department, Thapar University, Patiala, for providing his encouragement and also full facilities for the execution of this work.

I wish to express my deep gratitude to my guide **Ms. Gagandeep Kaur, Assistant Professor,** Department of Electrical and Instrumentation Engineering, Thapar University, Patiala, for supporting me at the conception of this work and helped me in resolving my problems in many direct and indirect ways.

Her thoughtful and valuable reviews and suggestions have helped me to improve the work. I cannot find words to describe the debt I owe to my guide, for having created a stimulating atmosphere of academic excellence, the basic element of any long lasting endeavor.

I heartily acknowledge the cooperation and moral support of family, my father S. Gian Singh, my mother Ms. Sarabjeet Kaur, and also thankful to my aunt Ms Parkash Kaur and my loving brother Mr. Bhavdeep Singh who always supported me in the entire course of study and thesis work.

(HARSIMRAN JIT SINGH)

Date: 05-July-2010

Place: Thapar University,

Patiala.

Abstract

Growing need of industry for higher productivity is placing new demands on mechanisms connected with electrical motors. They lead to different problems in work operation due to fast dynamics and instability. The stability of the system is essential to work at desired set targets. The non-linear effects caused by a motor frequently reduce stability, which reduces the controller’s ability to maintain speed at set points. Hence the flexibility of a mechanism and its control is very important. The very basic approach in control system engineering is to assume that the mechanism connected to motor is rigid. Hence the variations in set speed targets or any other apparatus connected to the motor are taken into account. This may reduce the ability of a dc machine system to carry out its assignment at set speed. It is more important to know how the mechanism, or in other words the load on the motor behaves. Anon-linear load control method for dc motor is developed and implemented in the thesis. The theoretical results obtained from the control design are validated with the PID model for dc speed control. Generalization of the system allows the methods derived here to be widely implemented in machine automation. The control algorithms are first designed in a specially introduced linear simulation model using simulink and fuzzy controller and then implemented. The results proved that both the controllers are capable of achieving the set speed stable targets but the technology of fuzzy logic provides better controllability at any set speed in lesser time as well as lesser spikes and fluctuations in the entire operation. The superiority of response using fuzzy logic controller over conventional as well as PID based controllers has been proved.

The dc motor is such a drive which is perhaps more often used in all industrial processes including electrical, mechanical, construction, petroleum industry, power sectors, development sites, etc. so the controlled stable operation of this drive attracts the researchers always, still keeping more and more future scope in it.

Chapter 1: Fuzzy Logic

Chapter 3 Simulation and Results

List of Figures

List of Abbreviations

ANN	Artificial Neural Networks
FL	Fuzzy Logic
RBF	Radial Basis Function
T-S	Takagi-Sugen
FNN’s	Fuzzy Neural Networks
NN	Neural Network
AC	Alternating Current
DC	Direct Current
FHP	Fractional Horse Power
FIS	Fuzzy Inference System
FNL	Fuzzy-Neural-Linear
BPTT	Back-propagation-Through-Time

LITERATURE SURVEY

W. Pedrycz, A.F. Rocha [1] introduces different fuzzy-set oriented computational models. The generic topologies emerging are encountered in the theory of fuzzy sets. The logical flavor of the proposed constructs is expressed in terms of operators used in their formalization and a way of their superposition in the neurons.

Kang Hoon [2] provides fuzzy control designers with a generalized design tool for stable fuzzy logic controllers in an optimal sense. Given multiple sets of data disturbed by vagueness uncertainty, the implicative rules that guarantee stability and robustness of closed-loop fuzzy dynamic systems are made. First, the mathematical basis of fuzzy hyper cubes and fuzzy dynamic systems are rigorously studied by considering the membership conditions for perfect recall and the evidential combination for reliable reasoning. Second, the author suggests the cell-state transition method, which utilizes Hsu's cell-to-cell mapping concept.

J. Nie and D.A. Linkens [3] describes an approach to integrating fuzzy reasoning systems with radial basis function (RBF) networks and shows how the integrated network can be employed as a multivariable self-organizing and self-learning fuzzy controller. In particular, by drawing some equivalence between a simplified fuzzy control algorithm (SFCA) and a RBF network, conclude that the RBF network can be interpreted in the context of fuzzy systems and can be naturally fuzzified into a class of more general networks, referred to as FBFN, with a variety of basis functions not necessarily globally radial synthesized from each dimension by fuzzy logical operators. On the other hand, as a result of natural generalization from RBF to SFCA, the fuzzy system like RBF is capable of universal approximation.

Lee Jihong, Korea [4] suggests improving limitations of fuzzy PI controller especially when applied to high order systems, so proposal of two types of fuzzy logic controllers that take out appropriate amounts of accumulated control input according to fuzzily described situations in addition to the incremental control input calculated by conventional fuzzy PI controllers. The structures of the proposed controller were motivated by the problems of fuzzy PI controllers that they generally give inevitable overshoot when one tries to reduce rise time of response especially when a system of order higher than one is under consideration. Since the undesirable characteristics of the fuzzy PI controller are caused by integrating operation of the controller, even though the integrator itself is introduced to overcome steady state error in response, we propose two fuzzy controllers that fuzzily clear out integrated quantities according to situation. The first controller determines the fuzzy resetting rate by situations described fuzzily by error and error rate, and the second one by error and control input. The two structures both give reduced rise time as well as small overshoot.

H.A. Malki, Li Huaidong, Chen Guanrong [5] describes the design principle, tracking performance, and stability analysis of a fuzzy proportional-derivative (PD) controller.

H. Watanabe, W. J. Yakowenko [6] applied the discrimination analysis proposed by Norris, Pilsmorth, and Baldwin to the diagnosis of valvular heart diseases. They proposed the diagnosis method which uses concepts from fuzzy set theory. It consists of two independent parts: discrimination analysis and connectivity analysis. We performed the experiments in order to evaluate the effectiveness of the proposed discrimination analysis part of the method. Also, we extended the original method to handle partial manifestation of symptoms and severity of diseases by using fuzzy sets. In addition, the concept of prototypicalness of patients with a particular disease to improve the performance of the diagnosis is introduced.

G.F. Mauervol [7] presents anti-blocking system (ABS) brake controllers which pose unique challenges to the designer: a) For optimal performance, the controller must operate at an unstable equilibrium point, b) Depending on road conditions, the maximum braking torque may vary over a wide range, c) The tire slippage measurement signal, crucial for controller performance, is both highly uncertain and noisy, d) On rough roads, the tire slip ratio varies widely and rapidly due to tire bouncing, and e) The braking system contains transportation delays which limit the control system bandwidth. A digital controller design was chosen which combines a fuzzy logic element and a decision logic network. The controller identifies the current road condition and generates a command braking pressure signal, based on current and past readings of the slip ratio and brake pressure. The controller detects wheel blockage immediately and avoids excessive slipping. The ABS system performance is examined on a quarter vehicle model with nonlinear elastic suspension. The parallelism of the fuzzy logic evaluation process ensures rapid computation of the controller output signal, requiring less time and fewer computation steps than controllers with adaptive identification.

Tanaka [8] discusses stability analysis of fuzzy-neural-linear (FNL) control systems which consist of combinations of fuzzy models, neural network (NN) models, and linear models. The author considers a relation among the dynamics of NN models, those of fuzzy models and those of linear models. It is pointed out that the dynamics of linear models and NN models can be perfectly represented by Takagi-Sugeno (T-S) fuzzy models whose consequent parts are described by linear equations. In particular, the authors present a procedure for representing the dynamics of NN models via T-S fuzzy models. Next, the authors recall stability conditions for ensuring stability of fuzzy control systems in the sense of Lyapunov. The stability criteria are reduced to the problem of finding a common Lyapunov function for a set of Lyapunov inequalities. The stability conditions are employed to analyze stability of FNL control systems.

F. R. Rubio, M. Berenguel, E. F. Camacho [9] presents an application of fuzzy logic control to the distributed collector field of a solar power plant. The major characteristic of a solar power plant is that the primary energy source, solar radiation, cannot be manipulated. Solar radiation varies throughout the day, causing changes in plant dynamics and strong perturbations in the process. A special subclass of fuzzy inference systems, the TP (triangular partition) and TPE (triangular partition with evenly spaced midpoints) systems is used to obtain adequate control signals in the whole range of possible operating conditions. D. Sinha, P. Sinha, E. R. Dougherty, S. Batman [10] present the general paradigm for lifting binary morphological algorithms to fuzzy algorithms is employed to construct fuzzy versions of classical binary morphological operations. The lifting procedure is based upon an epistemological interpretation of both image and filter fuzzification. Algorithms are designed via the paradigm for various fuzzifications and their performances are analyzed to provide insight into the kind of liftings that produce suitable results.

Lin Yinghua, G. A. Cunningham III, S. V. Coggeshall [11] create a set of fuzzy rules to model a system from input-output data by dividing the input space into a set of subspaces using fuzzy partitions. We create a fuzzy rule for each subspace as the input space is being divided. These rules are combined to produce a fuzzy rule based model from the input-output data. If more accuracy is required, we use the fuzzy rule-based model to determine the structure and set the initial weights in a fuzzy neural network. This network typically trains in a few hundred iterations. Our method is simple, easy, and reliable and it has worked well when modeling large "real world" systems. Xin Wang Li [12] developed a number of stable and optimal fuzzy controllers for linear systems. Based on some classical results in control theory, we design the structure and parameters of fuzzy controllers such that the closed-loop fuzzy control systems are stable, provided that the process under control is linear and satisfies certain conditions. It turns out that if stability is the only requirement, there is much freedom in choosing the fuzzy controller parameters.

M. Figueiredo, F. Gomide, A. Rocha, R. Yager [13] gives the new fuzzy reasoning method for fuzzy control. A comparison with the most useful fuzzy control schemes, for a first-order with time delay process, is carried out. The results obtained show that Yager's method is superior from the point of view of both computational burden and control system behavior. Shieh Kung Ying, Ming Liaw Chang [14] discusses that the dynamic response trajectory of a traditional fuzzy controller cannot be quantitatively controlled, a fuzzy model following controller is proposed here. In the proposed controller, an output feedback linear model following controller (LMFC) is first designed according to the roughly estimated plant model to let its response follow the output generated by a reference model. Then a model following error driven control signal is synthesized such that good model following characteristics can be preserved at various operating conditions. The proposed controller is applied to the speed control of an induction motor drive. Dynamic signal analysis of the model following behavior is made and the procedure for constructing the control algorithms is described in detail. The performance of the drive and the effectiveness of the proposed controller are demonstrated by some simulated and experimental results.

Liang Chen Cheng, Chih Chen Wen [15] investigates the relationship between the piecewise linear fuzzy controller (PLFC), in which the membership functions for fuzzy variables and the associated inference rules are all in piecewise linear forms, and a Gaussian potential function network based controller (GPFNC), in which the network output is a weighted summation of hidden responses from a series of Gaussian potential function units (GPFU's). Systematic procedures are proposed for transformation from a PLFC to its GPFNC counterpart, and vice versa. H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka [16] proposes a genetic-algorithm-based method for selecting a small number of significant fuzzy if-then rules to construct a compact fuzzy classification system with high classification power. The rule selection problem is formulated as a combinatorial optimization problem with two objectives: to maximize the number of correctly classified patterns and to minimize the number of fuzzy if-then rules. Genetic algorithms are applied to this problem. A set of fuzzy if-then rules is coded into a string and treated as an individual in genetic algorithms. The fitness of each individual is specified by the two objectives in the combinatorial optimization problem.

I. Hiraga, T. Furuhashi, Y. Uchikawa, S. Nakayama [17] gives the procedure for acquiring control rules to improve the performance of control systems. A collision avoidance problem is that in which the controlled object is a ship with inertia which must avoid collision with a moving object. It has proven to be difficult to obtain collision avoidance rules, i.e., steering rules and speed control rules, which coincide with the operator's knowledge. This paper shows that rules of this type can be acquired directly from observational data using fuzzy neural networks (FNNs). This paper also shows that the FNN can obtain portions of the fuzzy rules for the inferences of the static and dynamic degrees of danger and the decision table based on the degrees of danger to avoid the moving obstacle.

F. Klawonn, J. Gebhardt, R. Kruse [18] tells about the way engineers use fuzzy control in real world applications is often not coherent with an understanding of the control rules as logical statements or implications. In most cases fuzzy control can be seen as an interpolation of a partially specified control function in a vague environment, which reflects the indistinguishability of measurements or control values. In this paper the authors show that equality relations turn out to be the natural way to represent such vague environments and they develop suitable interpolation methods to obtain a control function. As a special case of our approach the authors obtain Mamdani's model and can justify the inference mechanism in this model and the use of triangular membership functions not only for the reason of simplified computations, and they can explain why typical fuzzy partitions are preferred. The authors also obtain a criterion for reasonable defuzzification strategies.

Jun Zeng Xiao, M. G. Singh [19] presents the decomposition property of fuzzy systems using a simple, constructive, decomposition procedure. That is, by properly dividing the input space into sub-input spaces, a general fuzzy system is decomposed into several sub-fuzzy systems which are the simplest fuzzy systems in the sub-input spaces. Based on the decomposition property of fuzzy systems, the analysis of fuzzy systems can be divided into two steps: first, analyze the properties of the simplest fuzzy systems, and then, use the decomposition property to extend the results to general fuzzy systems. Using this idea, two applications of the decomposition property are given. The first is the application to the representation capability analysis of fuzzy systems. The second is the application to the analysis of a class of nonlinear control systems. Then, based on the piecewise affine fuzzy-system model, the existence condition and the design of a stable control for a class of single-input single-output (SISO) nonlinear systems are presented.

Orlando De Jesús and Martin T. Hagan [20] introduces a general framework for describing dynamic neural networks—the layered digital dynamic network (LDDN). This framework allows the development of two general algorithms for computing the gradients and Jacobians for these dynamic networks: backpropagation-through-time (BPTT) and real-time recurrent learning (RTRL).

Abhijeet V. Nandedkar, and Prabir K. Biswas [21], proposes a fuzzy min-max neural network classifier with compensatory neurons (FMCNs). FMCN uses hyperbox fuzzy sets to represent the pattern classes. It is a supervised classification technique with new compensatory neuron architecture. The concept of compensatory neuron is inspired from the reflex system of human brain which takes over the control in hazardous conditions. Compensatory neurons (CNs) imitate this behavior by getting activated whenever a test sample falls in the overlapped regions amongst different classes.

Jianming Lu, Xue Yuan, and Takashi Yahagi [23] provide a new method of face recognition based on fuzzy clustering and parallel NNs is proposed. The face patterns are divided into several small-scale neural networks based on fuzzy clustering and they are combined to obtain the recognition result.

Yong He, Guoping Liu, and D. Rees [24] proposed a new method for stability analysis of neural networks (NNs) with a time-varying delay. Some less conservative delay-dependent stability criteria are established by considering the additional useful terms, which were ignored in previous methods, when estimating the upper bound of the derivative of Lyapunov functionals and introducing the new free-weighting matrices.

Tomohisa Hayakawa, Wassim M. Haddad, and Naira Hovakimyan [25] provides a neuro-adaptive control framework for continuous- and discrete-time nonlinear uncertain dynamical systems with input-to-state stable internal dynamics is developed. The framework is Lyapunov based and unlike standard neural network (NN) controllers guaranteeing ultimate boundedness, the framework guarantees partial asymptotic stability of the closed-loop system, that is, asymptotic stability with respect to part of the closed-loop system states associated with the system plant states. The neuro-adaptive controllers are constructed without requiring explicit knowledge of the system dynamics other than the assumption that the plant dynamics are continuously differentiable and that the approximation error of uncertain system nonlinearities lie in a small gain-type norm bounded conic sector.

F. Remy, M. Week [27] present an Adaptive Control System, which is applied in the velocity control loop of a dc servo motor The system consists of a neuro fuzzy controller and a neural network model of the motor. the neuro fuzzy controller is trained on-line with an error that is calculated through the neural network model.

Kim Seungwoo, Cho Youngwan, Park Mignon [28] suggests a new fuzzy adaptive controller, which is able to solve the problems of classical adaptive controllers and conventional fuzzy adaptive controllers. It explains the architecture of a fuzzy adaptive controller using the robust property of a fuzzy controller. The basic idea of new adaptive control scheme is that an adaptive controller can be constructed with parallel combination of robust controllers. This new adaptive controller uses a multirule-base architecture which has several independent fuzzy controllers in parallel, each with different robust stability area. Out of several independent fuzzy controllers, the most suited one is selected by a system identifier which observes variations in the controlled system parameter.

A Costa, A. De Gloria, M. Olivieri [29] presents a hardware approach to the design of fuzzy controllers which, by exploiting some peculiar characteristics of fuzzy logic computation, allows one to save power consumption and increase computing speed. We show that the computation involved in a fuzzy controller has some statistic features that can be exploited by asynchronous computation. This paper presents a quantitative study of the statistical properties of fuzzy computation.

Multi-valued logic is fuzzy logic. Fuzzy logic is derived from fuzzy set theory. It deals with reasoning, approximations rather than precise values. The concept of Fuzzy Logic (FL) was conceived by Lotfi Zadeh, a professor at the University of California at Berkley, and presented not as a control methodology, but as a way of processing data by allowing partial set membership rather than crisp set membership or non-membership [1]. Fuzzy logic has rapidly become one of the most successful of today's technologies for developing advanced sophisticated control systems. Fuzzy logic addresses such applications perfectly as it resembles human decision making. It fills the gap in engineering design methods which are left vacant by purely mathematical approaches. In linear control design and also in purely logic-based approaches of expert systems in the system design methodologies. While other approaches require accurate equations to model real-world behaviors, fuzzy design can accommodate the ambiguities of real-world human language and logic. It provides both an intuitive method for describing systems in human terms. It automates the conversion of various system specifications into effective models. Fuzzy systems are an alternative to traditional notions of set membership and logic that has its origins in ancient Greek philosophy. The precision of mathematics owes its success in large part to the efforts of Aristotle and the philosophers who preceded him. In their efforts to devise a concise theory of logic, and later mathematics, the so-called"Laws of Thought" were posited. One of these, the "Law of the Excluded Middle," states that every proposition must either be true or false. There were strong and immediate objections: for example, Heraclitus proposed that things could be simultaneously true and not true. It was Plato who laid the foundation for what would become fuzzy logic, indicating that there was a third region beyond True and False where these opposites tumbled about. Other, more modern philosophers echoed his sentiments, notably Hegel, Marx, and Engels. But it was Lukasiewicz who first proposed a systematic alternative to the bi–valued logic of Aristotle. Even in the present time some Greeks are still outstanding examples for fussiness and fuzziness. Fuzzy Logic [2] has emerged as a profitable tool for the controlling and steering of systems and complex industrial processes, household and entertainment electronics.

Fuzzy logic is a powerful problem-solving methodology with millions of applications in embedded control and information processing. Fuzzy provides extremely simple way to definite conclusions from vague, ambiguous or imprecise information. Actually fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions. Unlike classical logic which requires a deep understanding of a system, exact equations, and precise numeric values, Fuzzy logic incorporates an alternative way of thinking, which allows modeling complex systems using a higher level of abstraction originating from our knowledge and experience. Fuzzy Logic allows expressing this knowledge with subjective concepts of very hot, bright red, and a long time.

On an engineering level, [8] fuzzy logic provides a platform for easily encoding human knowledge into the control of a system. It has been used in an increasing number of applications, especially in Japan. The Sendai railway in Japan is controlled by fuzzy logic controllers. Applications have been developed in tracking problems, tuning, interpolation, classification, handwriting, voice recognition, and image stabilization in video cameras, washing machines, vacuum cleaners, air conditioners, electric fans, hot plates, and Lexus automatic transmissions.

Fuzzy logic and probabilistic logic are mathematically similar. Both of them have truth values ranging between 0 and 1 but their basic concept is different due to different interpretations. Both degrees of truth and probabilities range between 0 and 1.

Probability has nothing in common with fuzziness, these are simply different concepts which superficially seem similar because of using the same unit interval of real numbers they have dual applicability and properties of random variables are analogous to properties of binary logic states. A basic application might characterize sub ranges of a continuous variable. For instance, a temperature measurement for anti-lock brakes might have several separate membership functions defining particular temperature ranges needed to control the brakes properly. Each function maps the same temperature value to a truth value in the 0 to 1 range. These truth values can then be used to determine how the brakes should be controlled.

CHAPTER 1

Fuzzy Logic

1.1 Fuzzy

An inverted pendulum experiment was demonstrated in 1987 that "produced balancing responses nearly 100 times shorter than those of conventional PID controller". PID controllers and variations thereof work well in many control systems. But when the system to be controlled contains uncertainty or is highly complex, poorly understood, or nonlinear, fuzzy control may work better. Fuzzy logic is a method of characterizing knowledge in terms of fuzzy sets and a rule base. A fuzzy system has one or more inputs that are fuzzified, a rule base that is evaluated according to the inputs, and one or more outputs that are defuzzified into "crisp" values. Bringing fuzzy logic to control problems is a way to use a human expert's knowledge about an analog process in a digital computer. Fuzzy logic is not always the best way to solve a control problem, but it offers several advantages.

1.2 Fuzzy Logic

FL requires [2] some numerical parameters in order to operate such as what is considered significant error and significant rate-of-change-of-error, but exact values of these numbers are usually not critical unless very responsive performance is required in which case empirical tuning would determine them. For example, a simple temperature control system could use a single temperature feedback sensor whose data is subtracted from the command signal to compute "error" and then time-differentiated to yield the error slope or rate-of-change-of-error, hereafter called "error-dot". Error might have units of degree and a small error considered to be 2F while a large error is 5F. The "error-dot" might then have units of degree/min with a small error-dot being 5F/min and a large one being 15F/min. These values don't have to be symmetrical and can be "tweaked" once the system is operating in order to optimize performance. Generally, FL is so forgiving that the system will probably work the first time without any tweaking.

1.3 How Does Fuzzy Logic Work

Fuzzy reasoning, approximate reasoning, is an inference procedure whose outcome is conclusion for a set of fuzzy if-then rules. The steps of fuzzy reasoning can be given as follows:

1. Input variables are compared with the MFs on the premise part to obtain the membership values of each linguistic label fuzzification.

2. The membership values on the premise part are combined through specific fuzzy set operations such as: min, max, or multiplication to get firing strength (weight) of each rule.

3. The qualified consequent either fuzzy or crisp is generated depends on the firing strength.

4. The qualified consequents are aggregated to produce crisp output according to the defined methods such as: centroid of area, bisector of area, mean of maximum, smallest of maximum and largest of maximum deffuzification.

1.4 Fuzzy Logic Features

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

1) 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.

2) Since the FL 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.

3) FL 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.

4) Because of the rule-based operation, any reasonable number of inputs can be processed and numerous outputs 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 FL controllers distributed on the system, each with more limited responsibilities.

5) FL 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.

1.5 Fuzzy Sets and Crisp Sets

The very basic notion of fuzzy systems is a fuzzy (sub) set [1]. In classical mathematics we are familiar with what we call crisp sets. For example, the possible interferometric coherence g values are the set X of all real numbers between 0 and 1. From this set X a subset A can be defined, (e.g. all values 0 _ g _ 0.3). The characteristic function of A, (i.e. this function assigns a number 1 or 0 to each element in X, depending on whether the element is in the subset A or not) is shown in Fig.1.1 The elements which have been assigned the number 1 can be interpreted as the elements that are in the set A and the elements which have assigned the number 0 as the elements that are not in the set A.

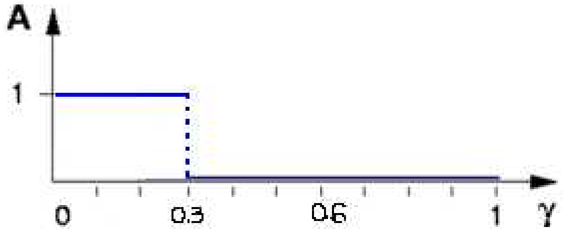


Fig 1.1 Characteristic function of a crisp set

This concept is sufficient for many areas of applications, but it can easily be seen, that it lacks in flexibility for some applications like classification of remotely sensed data analysis.

1.6 Fuzzy Expert System

A fuzzy expert system is an expert system that uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about data. The rules in a fuzzy expert system are usually of a form similar to the following:

if x is low and y is high then z = medium where x and y are input variables, z is an output variable i.e. a name for a data value to be computed, low is a membership function of fuzzy subset defined on x, high is a membership function defined on y, and medium is a membership function defined on z. The antecedents the rule's premise that describes to what degree the rule applies, while the conclusion i.e. the rule's consequent assigns a membership function to each of one or more output variables. Most tools for working with fuzzy expert systems allow more than one conclusion per rule. The set of rules in a fuzzy expert system is known as the rule base or knowledge base.

The general inference process proceeds in three or four basic steps.

1. FUZZIFICATION, the membership functions defined on the input variables are applied to their actual values, to determine the degree of truth for each rule premise.
2. INFERENCE, the truth value for the premise of each rule is computed, and applied to the conclusion part of each rule. This results in one fuzzy subset to be assigned to each output variable for each rule. Usually only min or product is used as inference rules. In MIN inference, the output membership function is clipped off at a height corresponding to the rule premise's computed degree of truth. In product inference, the output membership function is scaled by the rule premise's computed degree of truth.
3. COMPOSITION, all of the fuzzy subsets assigned to each output variable are combined together to form a single fuzzy subset for each output variable. Again, usually MAX or SUM are used. In MAX composition, the combined output fuzzy subset is constructed by taking the point wise maximum over all of the fuzzy subsets assigned to variable by the inference rule (fuzzy logic OR). In SUM composition, the combined output fuzzy subset is constructed by taking the point wise sum over all of the fuzzy subsets assigned to the output variable by the inference rule.
4. Finally is the defuzzification, which is used when it is useful to convert the fuzzy output set to a crisp number. There are many defuzzification methods than you can choose. Two of the more common techniques are the centroid and maximum methods. In the centroid method, the crisp value of the output variable is computed by finding the variable value of the center of gravity of the membership function for the fuzzy value. In the MAXIMUM method, one of the variable values at which the fuzzy subset has its maximum truth value is chosen as the crisp value for the output variable.

The elements which have been assigned the number 1 can be interpreted as the elements that are in the set *A* and the elements which have assigned the number 0 as the elements that are not in the set *A*. This concept is sufficient for many areas of applications, but it can easily be seen, that it lacks in flexibility for some applications like classification of remotely sensed data analysis. For example it is well known that water shows low interferometer coherence κ g in SAR images. Since κg starts at 0, the lower range of this set ought to be clear.

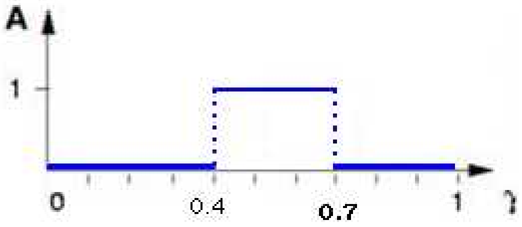


Fig1.2 Characteristic Function of a Crisp Set

The upper range, on the other hand, is rather hard to define. As a first attempt, we set the upper range to 0.2. Therefore we get B as a crisp interval B=[0,0.2]. But this means that a g κκvalue of 0.20 is low but a g κκvalue of 0.21 not. Obviously, this is a structural problem, for if we moved the upper boundary of the range from g κκ=0.20 to an arbitrary point we can pose the same question. A more natural way to construct the set *B* would be to relax the strict separation between *low* and *not low*. This can be done by allowing not only the (*crisp*) decision *Yes/No*, but more flexible rules like “fairly low". A fuzzy set allows us to define such a notion. The aim is to use fuzzy sets in order to make computers more intelligent; therefore, the idea above has to be coded more formally. In the example, all the elements were coded with 0 or 1. A straight way to generalize this concept is to allow more values between 0 and 1. In fact infinitely many alternatives can be allowed between 0 and 1, namely the unit interval I =[0, 1]. The interpretation of the numbers, now assigned to all elements is much more difficult. Of course, again the number 1 assigned to an element means, which the element is in the set *B* and 0 means that the element is definitely not in the set *B*. All other values mean a gradual membership to the set *B*. This is shown in Figure 1.3. The *membership function* is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion.

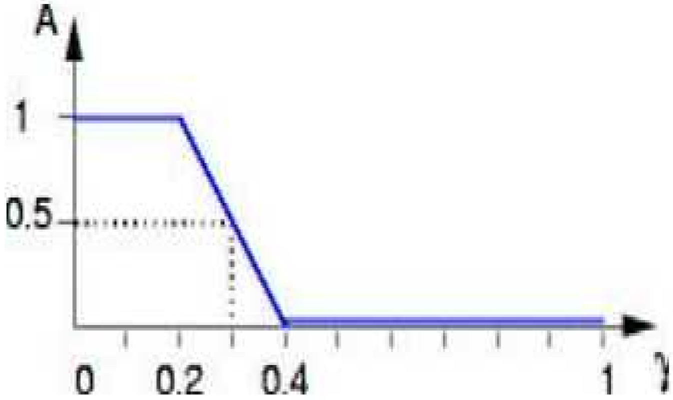


Fig1.3 Characteristic Function of a Fuzzy Set

The membership function, operating in this case on the fuzzy set of interferometric coherence returns a value between 0.0 and 1.0. For example, an interferometric coherence of 0.3 has a membership of 0.5 to the set low coherence (see Figure).It is important to point out the distinction between fuzzy logic and probability. Both operate over the same numeric range, and have similar values: 0.0 representing *False* (or non membership), and 1.0 representing *True* (or full membership). However, there is a distinction to be made between the two statements: The probabilistic approach yields the natural language statement, "There is a 50% chance that is low," while the fuzzy terminology corresponds to "g 's degree of membership within the set of low interferometric coherence is 0.50." The semantic difference is significant: the first view supposes that is or is not low; it is just that we only have a 50% chance of knowing which set it is in. By contrast, fuzzy terminology supposes that is "more or less" low, or in some other term corresponding to the value of 0.50.

1.7 Operations on Fuzzy Sets

We can introduce basic operations on fuzzy sets. Similar to the operations on crisp sets we also want to intersect, unify and negate fuzzy sets. In his very first paper about fuzzy sets, L. A. Zadeh [12] suggested the minimum operator for the intersection and the maximum operator for the union of two fuzzy sets. It can be shown that these operators coincide with the crisp unification and intersection if we only consider the membership degrees 0 and 1. For example, if A is a fuzzy interval between 5 and 8 and B be a fuzzy number about 4 as shown in the Fig.1.4 below

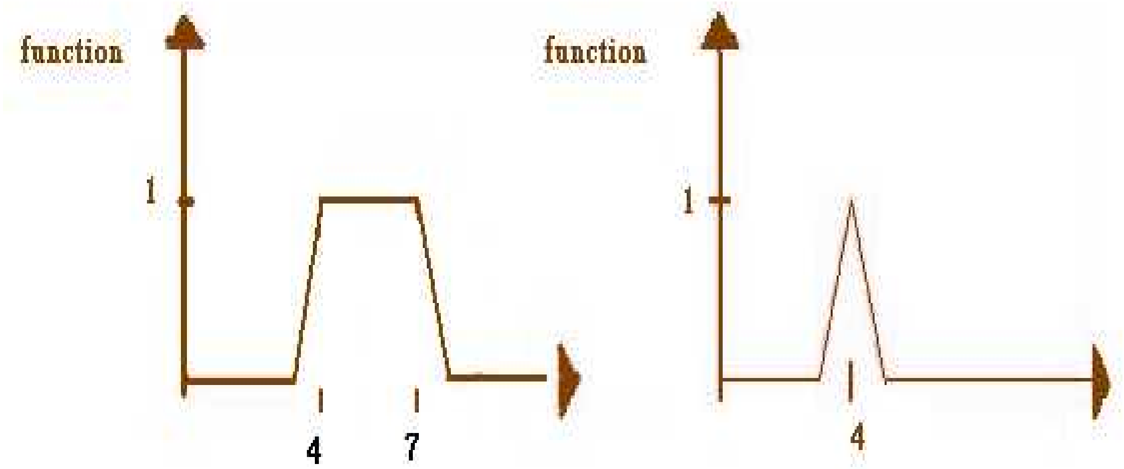


Fig. 1.4 Example fuzzy sets

In this case, the fuzzy set between 4 and 7 *AND* about 4 is

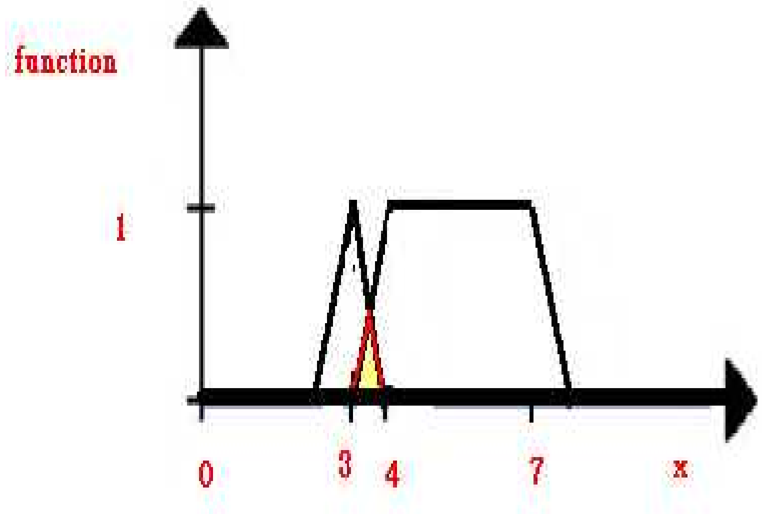


Fig 1.5 Fuzzy AND [11]

Set between 5 and 8 *OR* about 4 is shown in the fig. 1.6

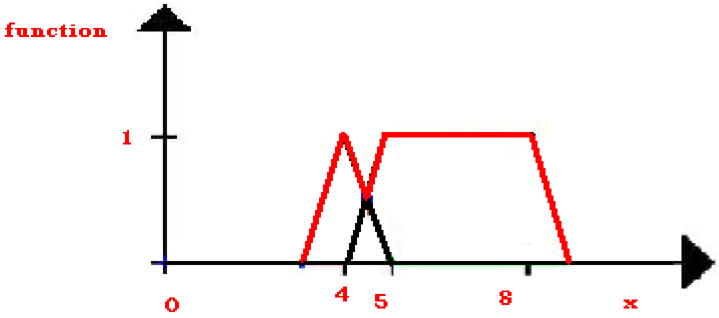


Fig.1.6 Fuzzy OR[11]

The negation of the fuzzy set A is shown below

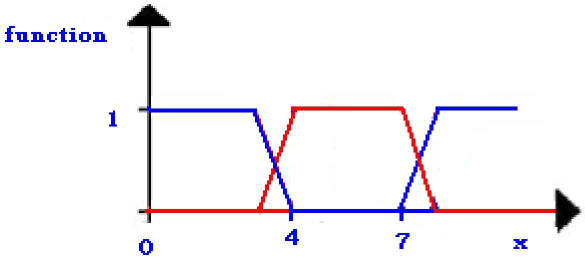


Fig.1.7 Fuzzy NEGATION [11]

Fuzzy classifiers are one application of fuzzy theory. Expert knowledge is used and can be expressed in a very natural way using linguistic variables , which are described by fuzzy sets Now the expert knowledge for this variables can be formulated as a rules like IF feature A low AND feature B medium AND feature C medium AND feature D medium THEN Class = class 4 The rules can be combined in table 1.1 calls rule base

R#	*feature A*	*feature B*	*feature C*	*feature D*	class
1:	low	medium	medium	medium	class1
2:	medium	high	medium	low	class2
3:	low	high	medium	high	class3
4:	low	high	medium	high	class 1
5:	medium	medium	medium	medium	class 4
...:	...	...	...	...	...
N:	low	high	medium	low	unknown

Table 1: Example for a fuzzy rule base: Rules read as (e.g. RULE No.1: IF A is low AND H is med. AND is med AND A is med. THEN pixel is class 1 Linguistic rules describing the control system consist of two parts; an antecedent block (between the IF and THEN) and a consequent block (following THEN). Depending on the system, it may not be necessary to evaluate every possible input combination, since some may rarely or never occur.

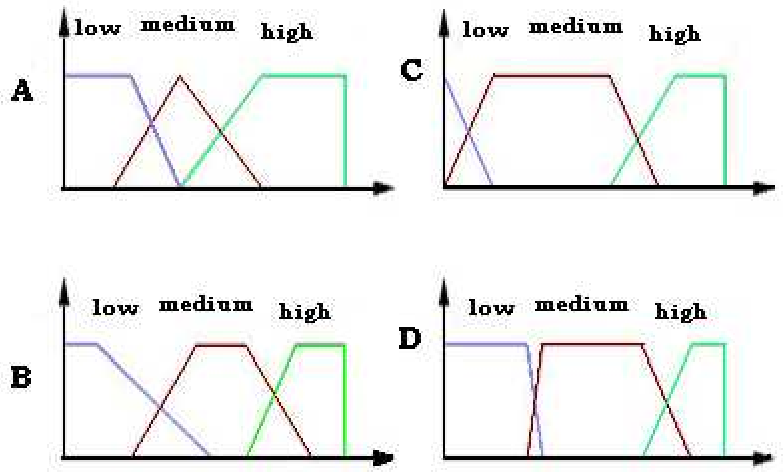


Fig.1.8 Example Linguistic variables performance.

The inputs are the active conclusions are then combined into a logical sum for each membership function. A firing strength for each output membership function is computed. All that remains is to combine these logical sums in a deffuzification process to produce the crisp output. e.g for a for the rule consequents for each class a so-called singleton or a min–max interference can be derived which is the characteristic function of the respective set . E.g. For the input pair of H = 0:35 and _ = 30 the scheme below would apply.

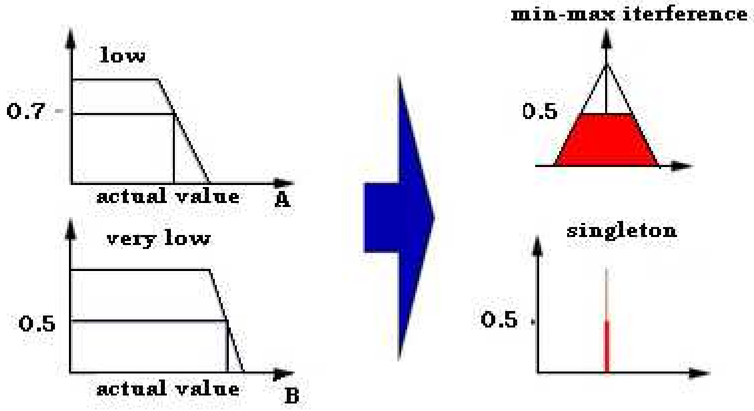


Fig.1.9 Interference of rule if H is very low and a low Then CLASS = class 1

The fuzzy outputs for all rules are finally aggregated to one fuzzy set [13]. To obtain a crisp decision from this fuzzy output, we have to defuzzify the fuzzy set, or the set of singletons. Therefore, we have to choose one representative value as the final output. There are several heuristic methods(defuzzification methods), one of them is e.g. to take the center of gravity of the fuzzy set as shown in figure which is widely used for fuzzy sets. For the discrete case with singletons usually the the maximum-method is used where the point with the maximum singleton is chosen.

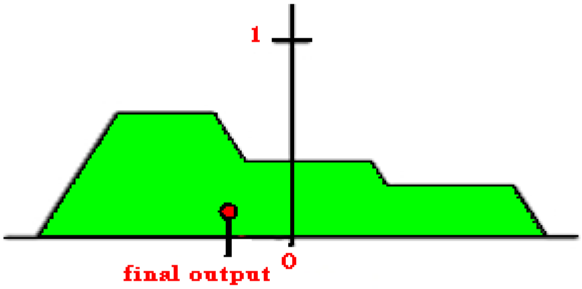


Fig.1.10 Defuzzification using the center of gravity approaches

1.8 Fuzzy Control

Automatic control belongs to the application areas of fuzzy set theory that have attracted most attention. In 1974, the first successful application of fuzzy logic to the control of a laboratory-scale process was reported (Mamdani and Assilian 1975). Control of cement kilns was an early industrial application (Holmblad and Ostergaard 1982). Since the first consumer product using fuzzy logic was marketed in 1987, the use of fuzzy control has increased substantially. A number of CAD environments for fuzzy control design have emerged together with VLSI hardware for fast execution. Fuzzy control is being applied to various systems in the process industry (Santhanam and Langari 1994, Tani et al. 1994), consumer electronics (Hirota 1993, Bonissone 1994), automatic train operation (Yasunobu and Miyamoto 1985), traffic systems in general (Hellendoorn 1993), and in many other fields (Hirota 1993, Terano et al. 1994). A fuzzy logic controller describes a control protocol by means of if-then rules, such as "if temperature is low open heating valve slightly". The ambiguity (uncertainty) in the definition of the linguistic terms (e.g., low temperature) is represented by using fuzzy sets, which are sets with overlapping boundaries, see Figure. In the fuzzy set framework, a particular domain element can simultaneously belong to several sets (with different degrees of membership). Forinstance, C belongs to the set of High temperatures with membership 0.3 and to the set of Medium temperatures with membership 0.1. This gradual transition from membership to non-membership facilitates a smooth outcome of the reasoning (deduction) with fuzzy if-then rules.

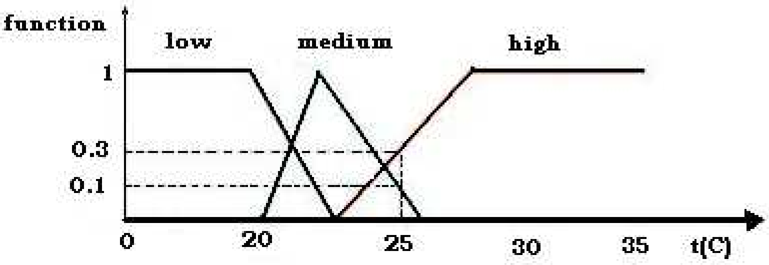


Fig.1.11 Partitioning of the temperature domain into three fuzzy sets

1.9 The Computational Environment

A small hybrid computer was available built around a PDP-8S mini-computer with 8K words (12 bit words) of magnetic core RAM [10]. Paper tape was the main form of back up store. The mouse had not been invented yet and the communication was via a teletype. The steam engine inputs and outputs were set and read via the hybrid computer. An individual run of the experiment would involve the teletype printing out the present speed and pressure and waiting for the operator to respond by typing in the values for heat and throttle settings. On pressing the return key, the computer carried out these settings and responded by typing the new readings of the speed and pressure. This went on until the boiler ran out of water whereupon a new run could be started. The study was agnostic with respect to any particular mathematical formalism to be used. Given the limited resources of our computational environment, we felt it was easiest to begin by using a Bayesian learning approach. The idea was to update the probabilities of an action given the state of the steam engine. This was obviously a naïve approach and not surprisingly the learnt probabilities failed to converge. It was naïve because it failed to take into account the fact that we were controlling a dynamic system, and the human operator did not merely take the current state into account in determine his action but was aware of the system's previous state trajectory. The algorithm had to be revised in such a way that it needed to take the previous states into account but it was feared that this would make it more complex and could stress the available RAM.

1.10 Fuzzy Systems

Fuzzy systems [28] are made of a knowledge base and reasoning mechanism called fuzzy inference engine. A fuzzy inference engine combines fuzzy if-then rules into a mapping from the inputs of the system into its outputs, using fuzzy reasoning methods. Fuzzy systems represents nonlinear mapping accompanied by fuzzy if-then rules from the rule base. Each of these rules describes the local mappings. The rule base can be constructed either from human expert or automatic generation that is extraction of rules using numerical input-output data .

1.10.1 Fuzzy Inference Engine

The early work in fuzzy control was motivated by a desire to mimic the control actions of an experienced human operator (knowledge-based part) obtain smooth interpolation between discrete outputs that would normally be obtained (fuzzy logic part). Since then the application range of fuzzy control has widened substantially. However, the two main motivations still persevere. The linguistic nature of fuzzy control makes it possible to express process knowledge concerning how the process should be controlled or how the process behaves. The interpolation aspect of fuzzy control has led to the viewpoint where fuzzy systems are seen as smooth function approximation schemes. In most cases a fuzzy controller is used for direct feedback control. However, it can also be used on the supervisory level as, e.g., a self-tuning device in a conventional PID (Proportional-Integral-Differential) controller. Also, fuzzy control is no longer only used to directly express a priori process knowledge. A fuzzy controller can be derived from a fuzzy model obtained through system identification. Most often used are:

Mamdani fuzzy inference system was first used to control a steam engine and boiler combination by a set of linguistic rules obtained from human operators. Figure illustrates how a two rule Mamdani fuzzy inference system derives the overall output z when subjected to two numeric inputs x and y. Takagi-Sugeno fuzzy inference system was first introduced by

Takagi and Sugeno. The difference of Takagi-Sugeno model is that each rule has a crisp output, and the overall output is determined as weighted average of single rules output.

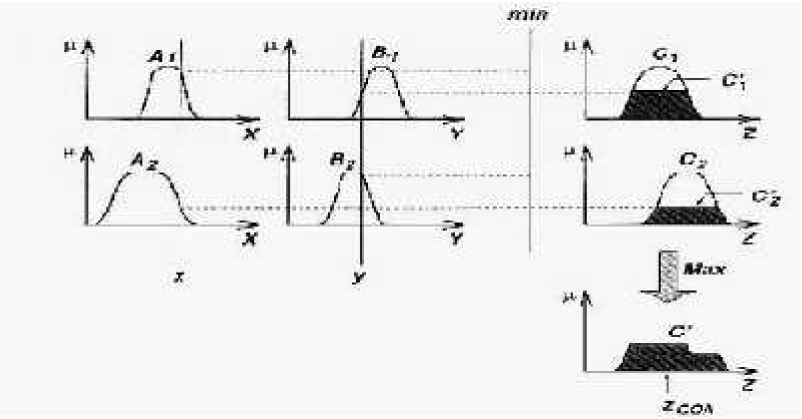


Fig. 1.12 Mamdani Fuzzy Inference System

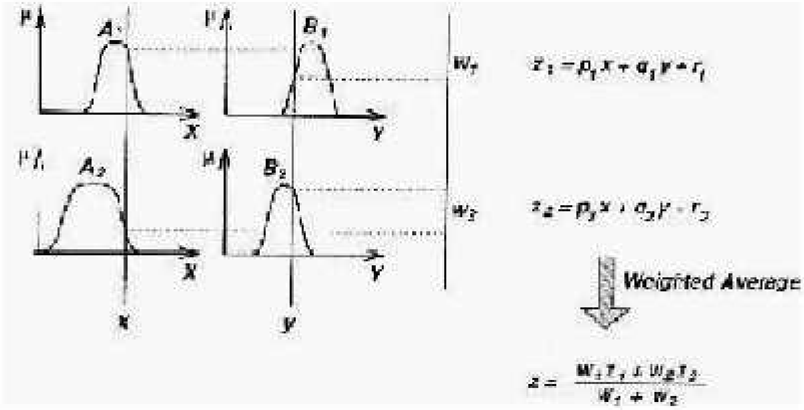


Fig.1.13 Takagi-Sugeno Fuzzy Inference System

1.10.2 Fuzzy Controller

When a person controls a process, perhaps as commonplace as a car engine or as specialized as a chemical plant, they describe their expertise in an apparently imprecise way [17]. In starting a car a person may accelerate the engine "a little" before engaging the clutch and driving away. The chemical plant operator may reduce process heating "slightly" if the product temperature is rising "slowly". While imprecise, such rules appear to work well in practice. If such a process needs to be automated one approach is to attempt to emulate the human operator. The usual starting point in the development of such a control system is the process of knowledge elicitation in which a record is constructed of the human operator's expertise. In some cases such expertise is expressed in the form of rules which make use of linguistic variables. It is at this point that the system developers can make an implementation decision. The human's expertise could be applied directly. Alternatively, control technologists could be called in to do in-depth numerical analysis of the process and to recommend an algorithm for control. In some cases the latter approach would turn out to be too expensive and time consuming, in others it might yield no results at all. In control technology terms there would be no process model obtainable within reasonable time and budget limitations. To control the process what is therefore needed is a computer based system which can use the control rules directly to implement an automatic control system. A fuzzy controller, which will make direct use of the control rules, is a strong candidate as a technical solution. Developing a fuzzy control knowledge base is divided into two main tasks. The first is to choose a suitable set of linguistic variables to describe the values of the main control parameters. This selection plays an important role in the smoothness of control. When all the different labels have been selected, their membership function must be defined. This process is highly subjective. One might also use the neural networks to learn the membership function from examples. The second task is to state the rules in the control knowledge base with the aid of the chosen linguistic description. This can be done in several ways, for instance by interviewing an experienced human operator. The basic architecture of a fuzzy controller is depicted in the fig1.14. The decision making logic consists of the inference and composition sub process.

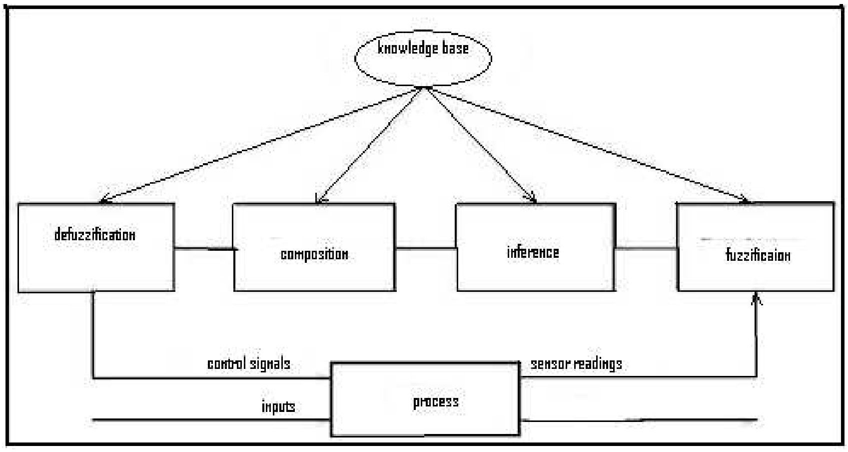


Fig.1.14 Topology of a typical fuzzy controller

1.10.3 Fuzzification

In the Fuzzification process, the membership functions defined on the input variables are applied to their actual values if the input variables are crisp. If the sensor is fuzzy (noisy), fuzzification refers to finding the intersection of the label's membership function and the distribution for the sensed data. Usually the sensor reading is crisp. The fuzzification sub process What fuzzification does is to turn the measurement into a degree of membership. Suppose a temperature measurement is made and corresponds to $80^0 C$. This measurement is required for an application in which the linguistic variable temperature might take on the values "high", "OK and "low". Fuzzification takes the measurement and decides to what degree it is "high", "OK" and "low". This matter of degree is decided on the basis of the framework suggested by the "expert" and is usually expressed as a membership function. The expert might consider that 80°C should be considered more "hot" than "OK". Where "1" represents full membership, the measurement might be "high" to a degree 0.8 because it is well above normal operating temperature, but it could get higher "OK to a degree 0.35 because operating temperature could be this high but would normally be lower "low" to a degree 0 because this temperature could never be considered low In a simple implementation the control rules will be applied to the degree implied

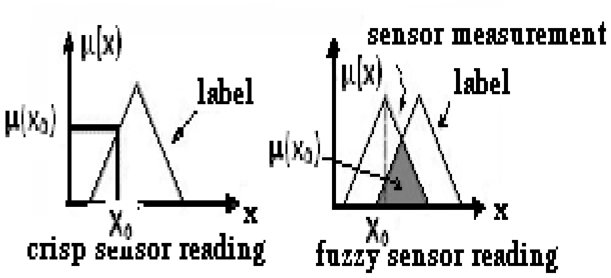


Fig.1.15 Fuzzy and Crisp readings

In this case the controller would apply rules applicable under high temperature conditions to a degree 0.8, and rule applying to "OK" temperature condition to a degree 0.35. The control action is in this way biased according to conditions. The fuzzification process thus determines how applicable each component of a rule's premise is. The applicability of the premise (if the rule applies at all) is its truth value so to speak. If a rule's premise has a non-zero degree of truth then the rule is said to fire.

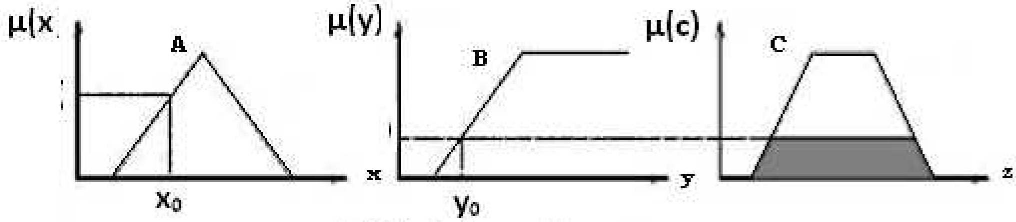
1.10.4 Inference

In the inference sub process, the truth value for the premise of each rule is computed and applied to the conclusion part of the rule. The result of this is assigning a fuzzy subset to each output variable of each rule. The truth value of the precondition of a rule is referred to as its strength and is denoted by A. The rule's strength is computed by the means of equations of complement, union and intersection.

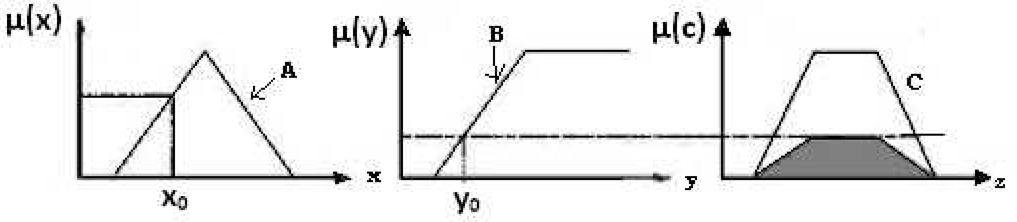
Example: Assume we have the following rule:

R: IF *x* is *A* AND *y* is *B* THEN *z* is *C*

In MIN-inferencing [21], the output membership function is clipped off at a height corresponding to rule's degree of truth. In Product-inferencing, the output membership function is scaled by the rule's computed degree of truth. A graphical illustration of the two inferencing methods is shown in figure



MIN-inferencing



Product-inferencing

Fig.1.16 Two inference methods

1.10.5 Composition

Many rules in the rule base may fire at the same time. If they involve the same output variable, some sort of conflict resolution has to be made. Consider the following rule base:

R1: IF *x* is *A1* AND *y* is *B1* THEN *z* is *C1*

R2: IF *x* is *A2* AND *y* is *B2* THEN *z* is *C2*

In the composition subprocess, all the fuzzy sets assigned to each output variable are combined together to form a single fuzzy set for each output variable. The most common rule of composition is Max-composition. In Max-composition, the combined output fuzzy subset is constructed by taking the point-wise maximum over all the fuzzy subsets assigned to the output by the inference rule. Another composition rule is Sum-composition. In Sum-composition the combined output fuzzy subset is constructed by taking the point-wise sum over all the fuzzy subsets assigned to the output variable by the inference rule. As this can result in truth values greater than one, Sum-composition is only used when followed by some kind of defuzzification method that does not have with this odd case.

1.10.6 Defuzzification

The control action must be in the form of a crisp value. Defuzzification is the process of transforming the fuzzy set assigned to a control output variable into such a crisp value. There are various methods for defuzzification. The following two are the most prominent in fuzzy control.

Centre of Area Method

Mean of Maxima Method

1.11 Fuzzy Set Operations.

1.11.1 Union

The membership function of the Union of two fuzzy sets A and B with membership functions μ_A and μ_B respectively is defined as the maximum of the two individual membership functions. This is called the maximum criterion.

$$\mu_{A \cup B} = \max(\mu_A, \mu_B)$$

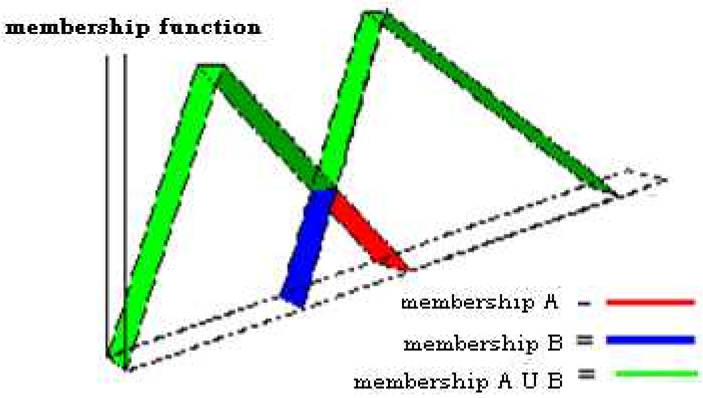


Fig.1.17 the Union operation.

1.11.2 Intersection

The membership function of the Intersection of two fuzzy sets A and B with membership functions μ_A and μ_B respectively is defined as the minimum of the two individual membership functions. This is called the minimum criterion.

$$\mu_{A \cap B} = \min(\mu_A, \mu_B)$$

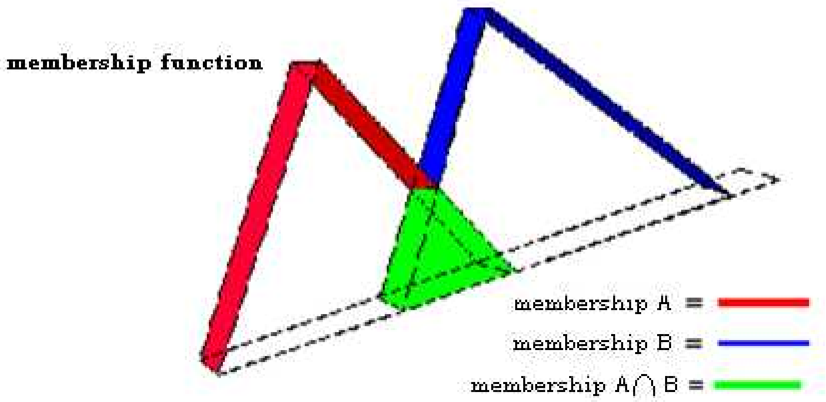


Fig1.18 The Intersection operation

1.11.3 Complement

The membership function of the Complement of a Fuzzy set A with membership function μ_A is defined as the negation of the specified membership function. This is caleed the negation criterion.

$$\mu_{\overline{A}} = 1 \quad \mu_A$$

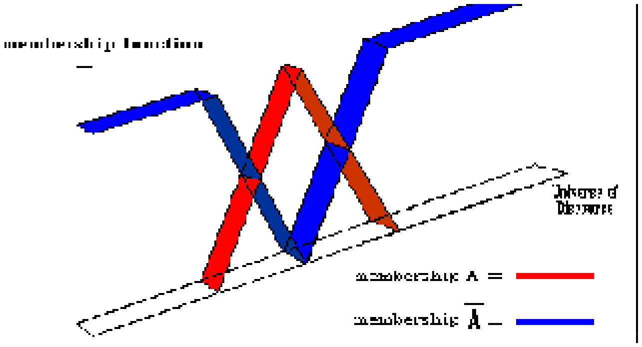


Fig.1.19 The Complement operation

The following rules which are common in classical set theory also apply to Fuzzy set theory.

De Morgan's law

$$(A\cap B)=A\cap B\ ,\ \overline{(A\cup B)}=\overline{A}\cap\overline{B}$$

Associatively

$$(A\cap B)\cap C=A\cap(B\cap C)$$

$$(A\cup B)\cup C=A\cup(B\cup C)$$

Commutatively

$$A\cap B=B\cap A\ ,\ A\cup B=B\cup A$$

Distributive

$$A\cap(B\cup C)=(A\cap B)\cup(A\cap C)$$

$$A\cup(B\cap C)=(A\cup B)\cap(A\cup C)$$

Fuzzy rules often take the place of a math model. Therefore, fuzzy logic is useful if a mathematical model of a process does not exist, is too difficult to encode, is too complex to be evaluated in real-time, or requires too much memory.

1.12 Neural Networks

A neural network is a computational and mathematical model of the brain. In neural network models computation is distributed over several simple units called neurons. Neurons are interconnected and operate parallel that's why neural networks are also called parallel-distributed-processing systems or connectionist systems [23]. An artificial neural network i.e. ANN often called as neural network i.e. NN, is an interconnected group of artificial neurons that uses a mathematical model or computational model for information processing. All the computations are based on a connectionist approach. An ANN is an adaptive system that changes its structure based on external or internal information that flows through the network. Practically the neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs. Neural networks NNs do not perform miracles. But if used sensibly they produce amazing results. Artificial Neural Networks i.e. ANNs are non linear information processing structures in which the elements called as neurons process the information. Signals are transmitted by means of connecting links. The links possess an associated weight, which is multiplied along with the incoming signal i.e. net input for any typical neural network. The output signal is obtained by applying activations to the net input. An Artificial Neural Network is characterized by

1. Its architecture i.e. connection between neurons
2. Its training or learning i.e. to determining weights on connections
3. Its activation functions

The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pits. But the technology available at that time did not make them comfortably to do too much in their inventions. But in late 1980's interest in NN increased with algorithms like Back Propagation and Kohonen. Progress continued during the 1990's also. And it lead to many fold applications in all areas. The neural network field enjoys a resurgence of interest from all. NNs are used in many commercial applications like character recognition, image recognition, fraud detection, credit evaluation, , insurance, load forecasting and stock forecasting etc [25].

In modern software implementations of artificial neural networks the approach inspired by biology has for the most part been abandoned for a more practical approach based on statistics and signal processing. In some of these systems, neural networks or parts of neural networks (such as artificial neurons) are used as components in larger systems that combine both adaptive and non-adaptive elements. While the more general approach of such adaptive systems is more suitable for real-world problem solving. It has less to do with the traditional artificial intelligence connectionist models. But they have in common the principle of non-linear, distributed, parallel and local processing and adaptation. The most popular neural network is the multi-layer perceptron, which is a feed forward network. For artificial neural network to give any results it must be trained with series of examples and conditions [25]. During the training neural network learns the governing relationships in given data sets. Input vectors to produce right solutions i.e. output vectors. For this purpose, back-propagation training algorithm is used. It is an iterative algorithm for minimizing the mean square error between predicted and desired output values. All signals flow in a single direction from the input to the output of the network. Feed forward networks can perform static mapping between an input space and an output space: the output at a given instant is a function only of the input at that instant.

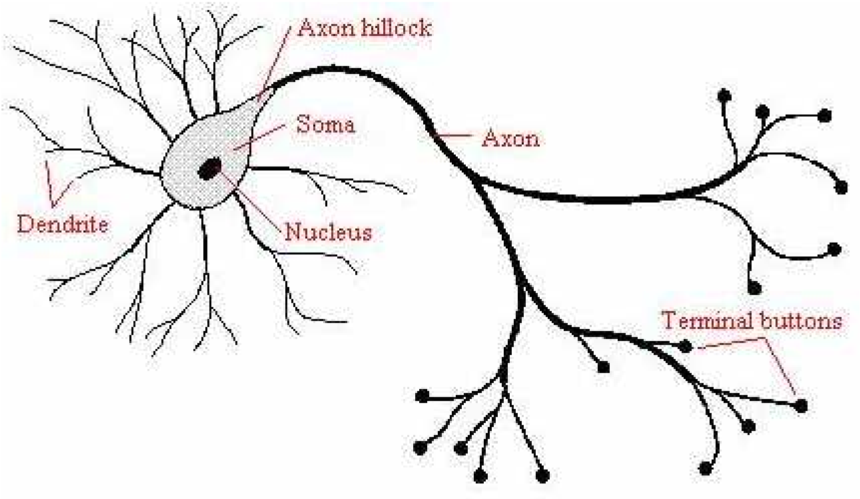


Fig.1.20 Simple Biological Neuron

1.13 Basic Neuron Structure

The structure of biological neuron and artificial neuron is briefly explained here.

1.13.1 Biological Neuron

A biological neuron or a nerve cell consists of synapses, dendrites, the cell body and the axon. The various building blocks are

- The synapses are elementary signal processing devices
- A synapse is a biochemical device, which converts a pre-synaptic electrical signal into a chemical signal and then back into a post-synaptic electrical signal.
- The input pulse train has its amplitude modified by parameters stored in the synapse. The nature of this modification depends on the type of the synapse, which can be either inhibitory or excitatory

- The postsynaptic signals are aggregated and transferred along the dendrites to the nerve cell body
- The cell body generates the output neuronal signal, a spike, which is transferred along the axon to the synaptic terminals of other neurons.

1.13.2 Mathematical Model

In accordance with the biological model, different mathematical models were suggested. The mathematical model of the neuron, which is usually utilized under the simulation of NN.The neuron receives a set of input signals x1, x2,..., xn (vector X) which are usually the output signals of other neurons. Each input signal is multiplied to a corresponding connection weight, w, and analogue of the synapse's efficiency. Weighted input signals come to the summation module corresponding to cell body, where their algebraic summation is executed and the excitement level of neuron is determined:

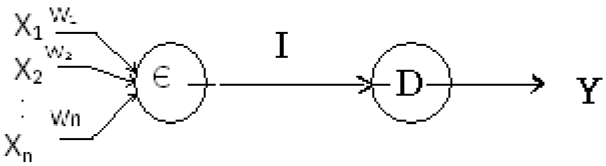


Fig.1.21 Mathematical Neuron

The output signal of a neuron is determined by conducting the excitement level through the function f, called activation function as in Equation 1.1

y=f(I)..........1.1

1.13.3 Transfer Function

The behavior of an ANN (Artificial Neural Network) [23] depends on both the weights and the input-output function (transfer function) that is specified for the units. This function typically falls into one of three categories:

Linear (or ramp): The output activity is proportional to the total weighted output.

Threshold: The output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value.

Sigmoid: The output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units, but all three must be considered rough approximations.

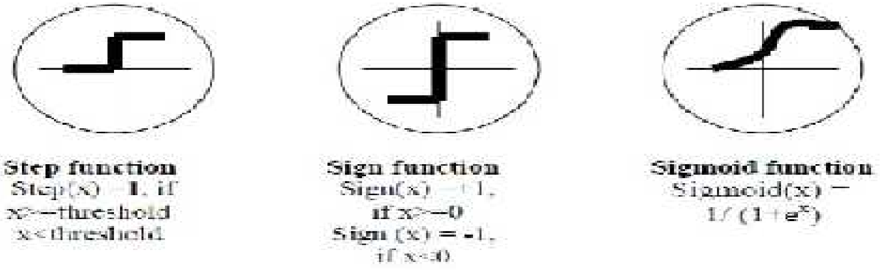


Fig.1.22 Transfer Function

1.13.4 Type of Neurons:

Neuron can be classified in two types namely, simple neuron and complicated neuron. Following is the basic information of these two given in detail:-

1.13.4.1 Simple Neuron

An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not. Figure show an example of simple neuron.

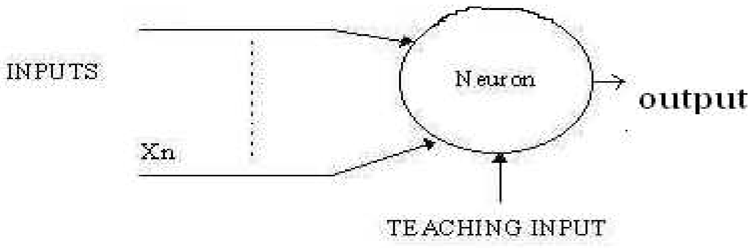


Fig.1.23 Simple Neuron

1.13.4.2 Complicated neuron

The simple neuron doesn't do anything that conventional computers don't do already. The difference from the previous model is that the inputs are ‘weighted’; the effect that each input has at decision making is dependent on the weight of the particular input. The weight of an input is a number which when multiplied with the input gives the weighted input. These weighted inputs are then added together and if they exceed a pre-set threshold value, the neuron fires. In any other case the neuron does not fire.

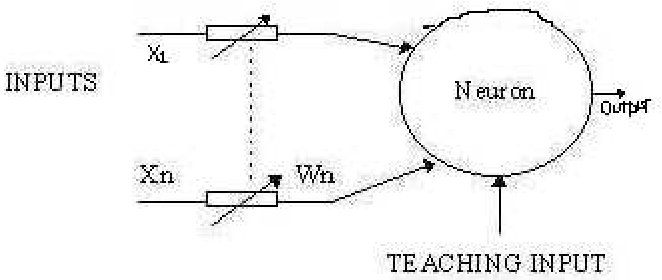


Fig.1.24 A Complicated Neuron

In mathematical terms, the neuron fires if and only if;

$$XW + XW + XW + \ldots > T$$

The addition of input weights and of the threshold makes this neuron a very flexible and powerful one. The complicated neuron has the ability to adapt to a particular situation by changing its weights and/or threshold. Various algorithms exist that cause the neuron to 'adapt'; the most used ones are the Delta rule and the back error propagation.

1.14 Architectures of Neural Networks

The totality of the neurons, connected with each other and with the environment, forms the NN. Figure shows the basic structure of the neural network. The input vector comes to the network by activating the input neurons [20]. A set of input signals of a network’s neurons is called the vector of input activeness. Connection weights of neurons are represented in form of matrix W, element w_{ij} of which is the connection weight between i-th and j-th neurons. During the network functioning process, the input vector is transformed into output one, i.e. some information processing is performed. The computational power of the network, thus, solves problems with its connections. Connections link inputs of one neuron with output of others. The connection strengths are given by weight coefficients. NN can also consist a bias term, which acts on a neuron like an offset. The function of the bias is to provide a threshold for the activation of neurons. The bias can be connected all neurons in network.

NN's can be divided into two types of architectures

- Feed-forward Networks
- Recurrent NN's.

1.14.1 Feed-forward Networks

Feed-forward networks have no feedback connections. In this type of network, neurons of the j-the layer receive signals from environment (when j=1) or the neurons of previous the (j-1)-th layer when (j>1) and pass their outputs to neurons of the next (j+1)-th layer to the environment (when j is the last layer).

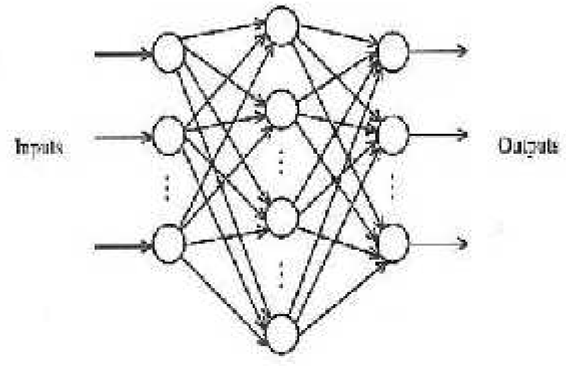


Fig.1.25 Basic structure of Neural Network [10]

Feed-forward networks can be single-layer or multi-layer. Multilayer NN's consist of input, output and hidden layer. The use of hidden layers allows an increase in the computational power of the network. Choosing the optimal structure of a network provides an increase in reliability and computational power, and a decreased processing.

1.14.2 Recurrent Neural Networks

Recurrent neural networks have structures similar to standard feed-forward NN with layers of nodes connected via weighted feed-forward connections, but also include time delayed feed-backor recurrent connections in the architecture. The important advantage of the RNN is the ability to approximate a continuous or discrete nonlinear dynamic system by neural dynamics defined by a system of nonlinear differential equations. This offers the opportunities for applications to adaptive control problems.

1.15 Training of Artificial Neural Networks:

A neural network has to be configured such that the application of a set of inputs produces (either 'direct' or via a relaxation process) the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to 'train' the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule. We can categories the learning situations in three distinct sorts. These are:

1.15.1 Supervised Learning

The vast majority of artificial neural network solutions have been trained with supervision. In this mode, the actual output of a neural network is compared to the desired output. Weights, which are usually randomly set to begin with, are then adjusted by the network so that the next iteration, or cycle, will produce a closer match between the desired and the actual output. The learning method tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until acceptable network accuracy is reached. With supervised learning, the artificial neural network must be trained before it becomes useful. Training consists of presenting input and output data to the network. This data is often referred to as the training set. This training phase can consume a lot of time. In prototype systems, with inadequate processing power, learning can take weeks. This training is considered complete when the neural network reaches an user defined performance level. This level signifies that the network has achieved the desired statistical accuracy as it produces the required outputs for a given sequence of inputs. When no further learning is necessary, the weights are typically frozen for the application.

After a supervised network performs well on the training data, then it is important to see what it can do with data it has not seen before. If a system does not give reasonable outputs for this test set, the training period is not over. Indeed, this testing is critical to insure that the network has not simply memorized a given set of data but has learned the general patterns involved within an application.

1.15.2 **Unsupervised learning** or **Self organization:**

In unsupervised learning we are given some data x and the cost function to be minimized, that can be any function of the data x and the network's output, f. The cost function is dependent on the task and our *a priori* assumptions (the implicit properties of our model, its parameters and the observed variables). As a trivial example, consider the model $f(x) = a$, where a is a constant and the cost $C = E[(x - f(x))^2]$. Minimizing this cost will give us a value of a that is equal to the mean of the data. The cost function can be much more complicated. Its form depends on the application: For example in compression it could be related to the mutual information between x and y. In statistical modelling, it could be related to the posterior probability of the model given the data.Tasks that fall within the paradigm of unsupervised learning are in general estimation problems; the applications include clustering, the estimation of statistical distributions, compression and filtering.

1.15.3 Reinforcement learning:

In reinforcement learning, data x is usually not given, but generated by an agent's interactions with the environment. At each point in time t, the agent performs an action y_t and the environment generates an observation x_t and an instantaneous cost c_t, according to some dynamics. The aim is to discover a *policy* for selecting actions that minimizes some measure of a long-term cost, i.e. the expected cumulative cost. The environment's dynamics and the long-term cost for each policy are usually unknown, but can be estimated. More formally, the environment is modeled as a Markov decision process (MDP) with states $s_1, \ldots, s_n \in S$ and actions $a_1, \ldots, a_m \in A$ with the following probability distributions: the instantaneous cost distribution $P(c_t \mid s_t)$, the observation distribution $P(x_t \mid s_t)$ and the transition $P(s_{t+1} \mid s_t, a_t)$, while a policy is defined as conditional distribution over actions given the observations. Taken together, the two define a Markov chain (MC). The aim is to discover the policy that minimizes the cost, i.e., the MC for which the cost is minimal. ANNs are frequently used in reinforcement learning as part of the overall algorithm. Tasks that fall within the paradigm of reinforcement learning are control problems, games and other sequential decision making tasks.

1.16 Learning Rates

The rate at which ANNs learn depends upon several controllable factors. In selecting the approach there are many trade-offs to consider [24]. Obviously, a slower rate means a lot more time is spent in accomplishing the off-line learning to produce an adequately trained system. With the faster learning rates, however, the network may not be able to make the fine discriminations possible with a system that learns more slowly. Network complexity, size, paradigm selection, architecture, type of learning rule or rules employed and desired accuracy must all be considered before training. These factors play a significant role in determining how long it will take to train a network. Changing any one of these factors may either extend the training time to an unreasonable length or even result in an unacceptable accuracy. Most learning functions have some provision for a learning rate, or learning constant. Usually this term is positive and between zero and one. If the learning rate is greater than one, it is easy for the learning algorithm to overshoot in correcting the weights, and the network will oscillate. Small values of the learning rate will not correct the current error as quickly, but if small steps are taken in correcting errors, there is a good chance of arriving at the best minimum convergence.

CHAPTER 2

DC MOTOR

2.1 History and Development

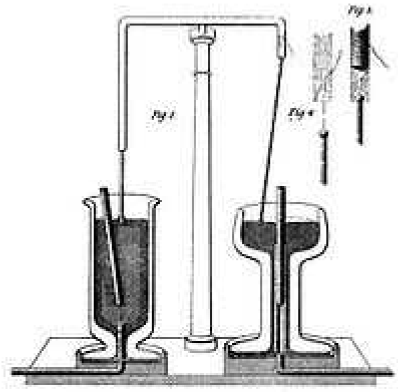


Fig.2.1 Faraday's Electromagnetic experiment, 1821.

2.1.1 The principle

The conversion of electrical energy into mechanical energy by electromagnetic means was demonstrated by the British scientist Michael Faraday in 1821. A free-hanging wire was dipped into a pool of mercury, on which a permanent magnet was placed. When a current was passed through the wire, the wire rotated around the magnet, showing that the current gave rise to a circular magnetic field around the wire. This is the simplest form of a class of devices called homopolar motors.

Fig.2.2 Jedlik's "lightning-magnetic self-rotor", 1827

In 1827, Hungarian Ányos Jedlik started experimenting with electromagnetic rotating devices he called "lightning-magnetic self-rotors". He used them for instructive purposes in universities, and in 1828 demonstrated the first device which contained the three main components of practical direct current motors: the stator, rotor and commutator. Both the stationary and the revolving parts were electromagnetic, employing no permanent magnets.

2.1.2 The First Electric Motors

The first commutator-type direct current electric motor capable of turning machinery was invented by the British scientist William Sturgeon in 1832. Following Sturgeon's work, a commutator-type direct-current electric motor made with the intention of commercial use was built by Americans Emily and Thomas Davenport and patented in 1837 [29]. Their motors ran at up to 600 revolutions per minute and powered machine tools and a printing press. Due to the high cost of the zinc electrodes required by primary battery power, the motors were commercially unsuccessful. Several inventors followed Sturgeon in the development of DC motors but all encountered the same cost issues with primary battery power. No electricity distribution had been developed at the time.

In 1855 Jedlik built a device using similar principles to those used in his electromagnetic self-rotors that was capable of useful work. He built a model electric motor-propelled vehicle that same year.

The modern DC motor was invented by accident in 1873, when Zénobe Gramme connected the dynamo he had invented to a second similar unit, driving it as a motor.

In 1886 Frank Julian Sprague invented the first practical DC motor, a non-sparking motor capable of constant speed under variable loads.

In 1888 Nikola Tesla invented the first practicable AC motor and with it the polyphase power transmission system.

The development of electric motors of acceptable efficiency was delayed for several decades by failure to recognize the extreme importance of a relatively-small air gap between rotor and stator. Early motors, for some rotor positions, had comparatively huge air gaps which constituted a very high reluctance magnetic circuit. They produced far-lower torque than an equivalent amount of power would produce with efficient designs.

Application of electric motors revolutionized industry. Industrial processes were no longer limited by power transmission using shaft, belts, compressed air or hydraulic pressure. Instead every machine could be equipped with its own electric motor, providing easy control at the point of use, and improving power transmission efficiency. Electric motors applied in agriculture eliminated human and animal muscle power from such tasks as handling grain or pumping water. Household uses of electric motors reduced heavy labor in the home and made higher standards of convenience, comfort and safety possible. Today, electric motors consume more than half of all electric energy produced.

2.2 Categorization of Electric Motors

The classic division of electric motors has been that of Alternating Current (AC) types vs Direct Current (DC) types. This is more a *de facto* convention, rather than a rigid distinction. For example, many classic DC motors run on AC power, these motors being referred to as universal motors.

Rated output power is also used to categories motors, those of less than 746 Watts [27], for example, are often referred to as fractional horsepower motors (FHP) in reference to the old imperial measurement.

The ongoing trend toward electronic control further muddles the distinction, as modern drivers have moved commutate out of the motor shell. For this new breed of motor, driver circuits are relied upon to generate sinusoidal AC drive currents, or some approximation thereof. The two best examples are: the brushless DC motor and the stepping motor, both being poly-phase AC motors requiring external electronic control, although historically, stepping motors (such as for maritime and naval gyrocompass repeaters) were driven from DC switched by contacts.

Considering all rotating (or linear) electric motors require synchronism between a moving magnetic field and a moving current sheet for average torque production, there is a clearer distinction between an asynchronous motor and synchronous types. An asynchronous motor requires slip between the moving magnetic field and a winding set to induce current in the winding set by mutual inductance; the most ubiquitous example being the common AC induction motor which must slip to generate torque. In the synchronous types, induction (or slip) is not a requisite for magnetic field or current production (e.g. permanent magnet motors, synchronous brush-less wound-rotor doubly-fed electric machine).

2.3 DC Motors

Industrial applications use dc motors because the speed-torque relationship can be varied to almost any useful form -- for both dc motor and regeneration applications in either direction of rotation. Continuous operation of dc motors is commonly available over a speed range of 8:1. Infinite range (smooth control down to zero speed) for short durations or reduced load is also common. DC motors [30] are often applied where they momentarily deliver three or more times their rated torque. In emergency situations, dc motors can supply over five times rated torque without stalling (power supply permitting).

Dynamic braking (dc motor-generated energy is fed to a resistor grid) or regenerative braking (dc motor-generated energy is fed back into the dc motor supply) can be obtained with dc motors on applications requiring quick stops, thus eliminating the need for, or reducing the size of, a mechanical brake. DC motors feature a speed, which can be controlled smoothly down to zero, immediately followed by acceleration in the opposite direction -- without power circuit switching. And dc motors respond quickly to changes in control signals due to the dc motor's high ratio of torque to inertia. A DC motor is designed to run on DC electric power. Two examples of pure DC designs are Michael Faraday's homopolar motor (which is uncommon), and the ball bearing motor, which is (so far) a novelty. By far the most common DC motor types are the brushed and brushless types, which use internal and external commutation respectively to create an oscillating AC current from the DC source—so they are not purely DC machines in a strict sense.

2.3.1 Principal of operation

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel, the internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion. Basic parts of DC motor -- axle, rotor, stator, commutator, field

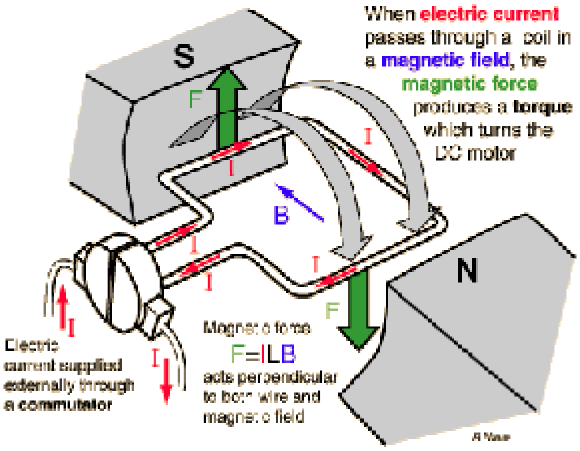


Fig.2.3 DC Motor Operation

In most common DC motors, the external magnetic field is produced by high-strength permanent magnets[1]. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotates with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnets.

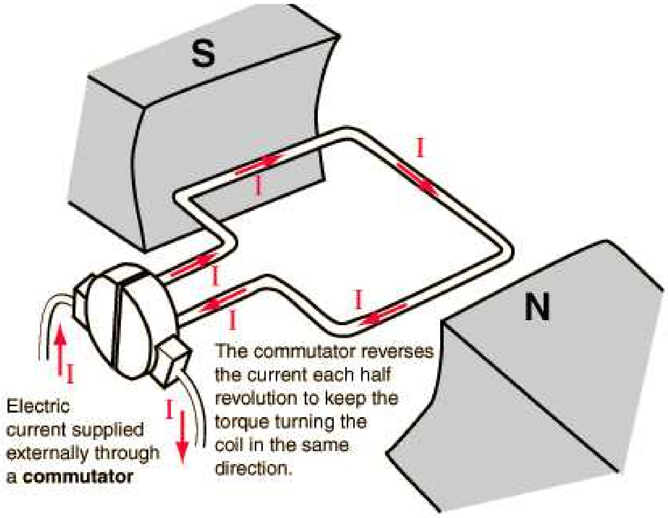


Fig.2.4 Current in DC Motor

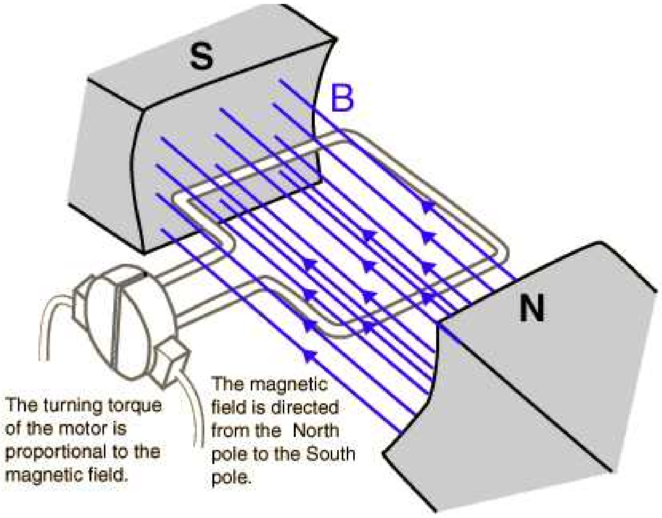


Fig.2.5 Magnetic Field in DC Motor

The geometry of the brushes, commutate contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commutates contacts, and energize the next winding.

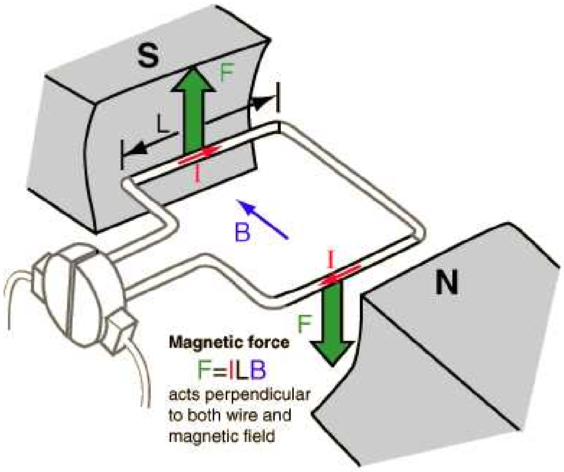


Fig.2.6 Force in DC Motor

In real life, though, dc motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in commutate. The two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutate shorts out the power supply (i.e., both brushes touch both commutate contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque "ripple" (the amount of torque it could produce is cyclic with the position of the rotor).

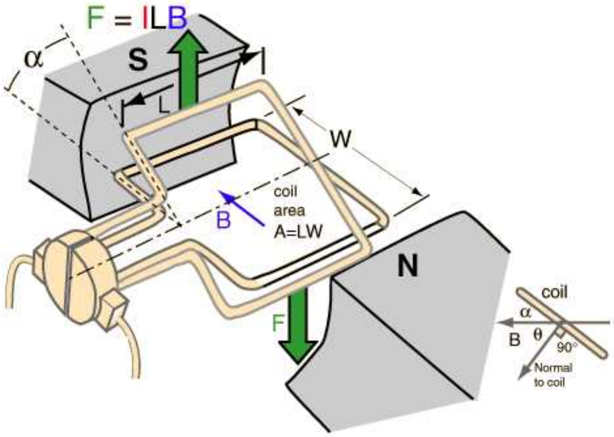


Fig.2.7 Torque in DC Motor

$$\begin{aligned}\text{Torque} &= \text{force} \times \text{lever arm}\\ &= (\text{ILB})\left[\frac{W}{2}\right] \sin\theta \times 2 \text{ sides}\\ &= \text{ILBW} \sin\theta = \text{IBA} \sin\theta\end{aligned}$$

The use of an iron core armature is quite common, and has a number of advantages. First off, the iron core provides a strong, rigid support for the windings -- a particularly important consideration for high-torque motors. The core also conducts heat away from the rotor windings, allowing the motor to be driven harder than might otherwise be the case. Iron core construction is also relatively inexpensive compared with other construction types.

But iron core construction also has several disadvantages. The iron armature has a relatively high inertia which limits motor acceleration. This construction also results in high winding inductance s which limits brush and commutates life.

In small motors, an alternative design is often used which features a 'coreless' armature winding. This design depends upon the coil wire itself for structural integrity. As a result, the armature is hollow, and the permanent magnet can be mounted inside the rotor coil. Coreless DC motors have much lower armature inductance than iron-core motors of comparable size, extending brush and commutate life.

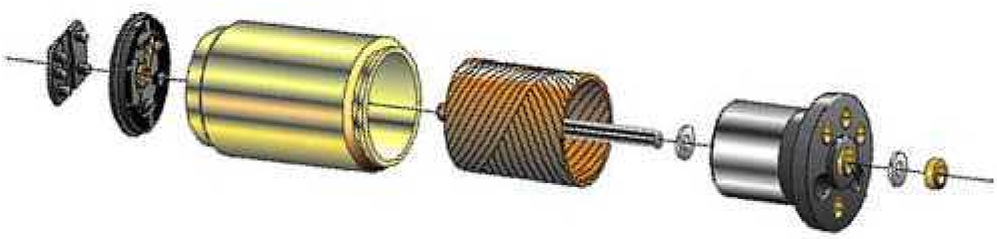


Fig.2.8 Diagram courtesy of MicroMo

The coreless design also allows manufacturers to build smaller motors; meanwhile, due to the lack of iron in their rotors, coreless motors are somewhat prone to overheating. As a result, this design is generally used just in small, low-power motors. BEAMers will most often see coreless DC motors in the form of pager motors

2.4 DC Motor Types

Wound-field dc motors are usually classified by shunt-wound, series-wound, and compound-wound. In addition to these, permanent-magnet and brushless dc motors are also available, normally as fractional-horsepower dc motors. Dc motors may be further classified for intermittent or continuous duty. Continuous-duty dc motors can run without an off period.

2.4.1 Brushed DC Motors

The brushed dc motor generates torque directly from DC power supplied to the motor by using internal commutation, stationary permanent magnets, and rotating electrical magnets.It works on the principle of Lorentz force, which states that any current carrying conductor placed within an external magnetic field experiences a torque or force known as Lorentz force. Advantages of a brushed DC motor include low initial cost, high reliability, and simple control of motor speed. Disadvantages are high maintenance and low life-span for high intensity uses.

2.4.2 Brushless DC Motors

Brushless DC motors [32] use a rotating permanent magnet in the rotor, and stationary electrical magnets on the motor housing. A motor controller converts dc to ac. This design is simpler than that of brushed motors because it eliminates the complication of transferring power from outside the motor to the spinning rotor. Advantages of brushless motors include long life span, little or no maintenance, and high efficiency. Disadvantages include high initial cost, and more complicated motor speed controllers.

2.4.3 Synchronous DC Motors

Synchronous dc motors [36], such as the brushless dc motor and the stepper motor, require external commutation to generate torque. They lock up if driven directly by dc power.

2.4.4 Uncommutated DC Motors

Other types of DC motors require no commutation.

- Homopolar motor- A homopolar motor has a magnetic field along the axis of rotation and an electric current that at some point is not parallel to the magnetic field. The name homopolar refers to the absence of polarity change.
- Ball-bearing motor- A ball bearing motor is an unusual electric motor that consists of two ball-bearing-type bearings, with the inner races mounted on a common conductive shaft, and the outer races connected to a high current, low voltage power supply. An alternative construction fits the outer races inside a metal tube, while the inner races are mounted on a shaft with a non-conductive section (e.g. two sleeves on an insulating rod).

An electric motor uses electrical energy to produce mechanical energy, very typically through the interaction of magnetic fields and current-carrying conductors. The reverse process, producing electrical energy from mechanical energy, is accomplished by a generator or dynamo. Many types of electric motors can be run as generators, and vice versa. For example a starter/generator for a gas turbine or Traction motors used on vehicles often perform both tasks. Electric motors are found in applications as diverse as industrial fans, blowers and pumps, machine tools, household appliances, power tools, and disk drives. The smallest motors may be found in electric wristwatches. Medium-size motors of highly standardized dimensions and characteristics provide convenient mechanical power for industrial uses. The very largest electric motors are used for propulsion of large ships, and for such purposes as pipeline compressors, with ratings in the millions of watts. Electric motors may be classified by the source of electric power, by their internal construction, by their application, or by the type of motion they give.

2.5 Implications

Because a DC motor operates most efficiently at less than 1/2 its stall torque, an "oversized" motor runs with the highest efficiency. IE: using a bigger motor than is necessary enables the motor to operate closest to no load, or peak operating conditions.

2.6 Torque Capability of Motor Types

When optimally designed for a given active current (i.e., torque current), voltage, pole-pair number, excitation frequency (i.e., synchronous speed), and core flux density, all categories of electric motors or generators will exhibit virtually the same maximum continuous shaft torque (i.e., operating torque) within a given physical size of electromagnetic core. Some applications require bursts of torque beyond the maximum operating torque, such as short bursts of torque to accelerate an electric vehicle from standstill. Always limited by magnetic core saturation or safe operating temperature rise and voltage, the capacity for torque bursts beyond the maximum operating torque differs significantly between categories of electric motors or generators.

2.7 DC Motors - Speed Control

There are two ways to adjust the speed of a wound-field dc motor. Combinations of the two are sometimes used to adjust the speed of a dc motor.

2.7.1 DC Motor - Shunt-field Control

Reel drives require this kind of control. The dc motor's material is wound on a reel at constant linear speed and constant strip tension, regardless of diameter. Control is obtained by weakening the shunt-field current of the dc motor to increase speed and to reduce output torque for a given armature current. Since the rating of a dc motor is determined by heating, the maximum permissible armature current is approximately constant over the speed range. This means that at rated current, the dc motor's output torque varies inversely with speed, and the dc motor has constant-horsepower capability over its speed range.

2.7.2 Armature-voltage DC Motor Control

In this method, shunt-field current is maintained constant from a separate source while the voltage applied to the armature is varied. DC motors feature a speed, which is proportional to the counter emf. This is equal to the applied voltage minus the armature circuit IR drop. At rated current, the torque remains constant regardless of the dc motor speed (since the magnetic flux is constant) and, therefore, the dc motor has constant torque capability over its speed range.

2.8 DC Motors – Selection

By selecting DC motor and associated equipment for a given application requires consideration of several factors.

2.8.1 DC Motors - Speed Range

If field control is to be used, and a large speed range is required, the base speed must be proportionately lower and the motor size must be larger. If speed range is much over 3:1, armature voltage control should be considered for at least part of the range. Very wide dynamic speed range can be obtained with armature voltage control. However, below about 60% of base speed, the motor should be de-rated or used for only short periods.

2.8.2 DC Motors - Speed Variation with Torque

Applications requiring constant speed at all torque demands should use a shunt-wound dc motor [30]. If speed change with load must be minimized, a dc motor regulator, such as one employing feedback from a tachometer, must be used.

2.8.3 DC Motors – Reversing

This operation affects power supply and control, and may affect the dc motor's brush adjustment, if the dc motor cannot be stopped for switching before reverse operation. In this case, compound and stabilizing dc motor windings should not be used, and a suitable armature-voltage control system should supply power to the dc motor.

2.8.4 DC Motors - Duty Cycle

Direct current motors are seldom used on drives that run continuously at one speed and load. Motor size needed may be determined by either the peak torque requirement or heating.

2.8.5 DC Motors - Peak Torque

The peak torque that a dc motor delivers is limited by that load at which damaging commutation begins. Dc motor brush and commutator damage depends on sparking severity and duration. Therefore, the dc motor's peak torque depends on the duration and frequency of occurrence of the overload. Dc motor peak torque is often limited by the maximum current that the power supply can deliver.

2.8.6 DC Motors–Heating

Dc motor temperature is a function of ventilation and electrical/mechanical losses in the machine. Some dc motors feature losses, such as core, shunt-field, and brush-friction losses, which are independent of load, but vary with speed and excitation.

The best method to predict a given dc motor's operating temperature is to use thermal capability curves available from the dc motor manufacturer. If curves are not available, dc motor temperature can be estimated by the power-loss method. This method requires total losses versus load curve or an efficiency curve.

2.9 Motor Characteristics

2.9.1 Torque/Speed Curves

In order to effectively design with D.C. motors, it is necessary to understand their characteristic curves. For every motor, there is a specific Torque/Speed curve and Power curve.

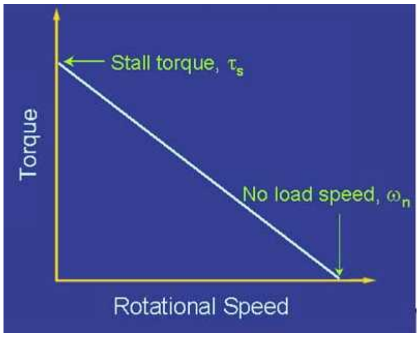


Fig.2.9 DC Motor Torque/Speed Curve

The graph above shows a torque/speed curve of a typical D.C. motor. Note that torque is inversely proportional to the speed of the output shaft. In other words, there is a tradeoff between how much torque a motor delivers, and how fast the output shaft spins. Motor characteristics are frequently given as two points on this graph:

The stall torque, τ_s, represents the point on the graph at which the torque is a maximum, but the shaft is not rotating.

The no load speed, ω_n, is the maximum output speed of the motor (when no torque is applied to the output shaft).

The curve is then approximated by connecting these two points with a line, whose equation can be written in terms of torque or angular velocity as equations eqn2.1 and eqn2.2:

$$\tau_{motor} = \tau_s - \omega\tau_s/\omega_n \text{ - - - - - - eqn2.1}$$

$$\omega_{motor} = (\tau_s - \tau)\omega_n/\tau_s \text{ - - - eqn2.2}$$

Due to the linear inverse relationship between torque and speed, the maximum power occurs at the point where $\omega = ½\ \omega_n$, and $\tau = ½\ \tau_s$.

2.9.2 Power/Torque and Power/Speed Curves

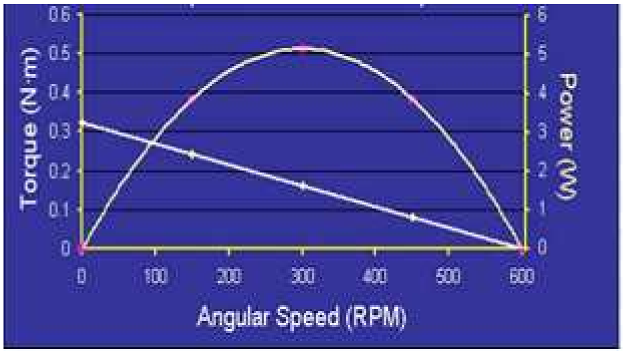


Fig.2.10 Torque and Power vs. Speed

Power curves for a D.C. motor with respect to both speed and torque are

$$P_{motor}(\omega) = -(\tau_s/\omega_n)\ \omega^2 + \tau_s\omega \text{ - - - eqn2.3}$$

$$P_{motor}(\tau) = -(\omega_n/\tau_s)\ \tau^2 + \omega_n\tau \text{ - - - eqn2.4}$$

From these equations, we again find that maximum output power occurs at $\tau = \frac{1}{2}\tau_s$, and $\omega = \frac{1}{2}\omega_n$ repectively.

2.10 Uses of Fuzzy Logic

Electric motors are used in many, if not most, modern machines. Obvious uses would be in rotating machines such as fans, turbines, drills, the wheels on electric cars, locomotives and conveyor belts. Also, in many vibrating or oscillating machines, an electric motor spins an irregular figure with more area on one side of the axle than the other, causing it to appear to be moving up and down.

Electric motors are also popular in robotics. They are used to turn the wheels of vehicular robots, and servo motors are used to turn arms and legs in humanoid robots. In flying robots, along with helicopters, a motor causes a propeller or wide, flat blades to spin and create lift force, allowing vertical motion.

Electric motors are replacing hydraulic cylinders in airplanes and military equipment.

In industrial and manufacturing businesses, electric motors are used to turn saws and blades in cutting and slicing processes, and to spin gears and mixers. Linear motors are often used to push products into containers horizontally.

Many kitchen appliances also use electric motors to accomplish various jobs. Food processors and grinders spin blades to chop and break up foods. Blenders use electric motors to mix liquids, and microwave ovens use motors to turn the tray food sits on. Toaster ovens also use electric motors to turn a conveyor to move food over heating elements.

Chapter 3

Simulation and Results

Analysis, modeling and control of dynamic systems of the digital control methods through computer required for the meaningful model the systems. These mathematical models were too time consuming and error prone. Simplifications were made possible with the simplest models of control systems. The availability of computers and the upcoming technologies made the detailed models and more complex algorithms simpler for analysis and design. A control system is combination of elements intended to act together to achieve the desired control. The conventional methods made the control problems through different methods of closed loop control, feed –forward and feed-back control strategies, but the modern methods of artificial intelligence acts like human thinking without requiring sure and certain information about data or the model which requires control.

Conventional control model of dc motor and fuzzy based control of dc motor are undertaken. Both the conditions are studied and analyzed the conclusion will be drawn. The analysis is being carried by using MATLAB and Simulink modules.

3.1 Speed of DC Motors

The speed of dc motor is ven by relationship…

$N=(V-IaRa)/k\Phi$…………eqn3.1

This equation shows that the speed is dependant upon the supply voltage V, the armature circuit resistance ra, and the field flux Φ, which is produced by the field current. In practice, the variation of these factors is used for speed control. Thus there are three general methods of speed control of dc motors:

1. Variation of resistance in the armature circuit (armature resistance control)

2. Variation of field flux Φ (field flux control)

3. Variation of applied voltage (armature voltage control)

3.2 Simulink Diagram of Conventional DC Motor

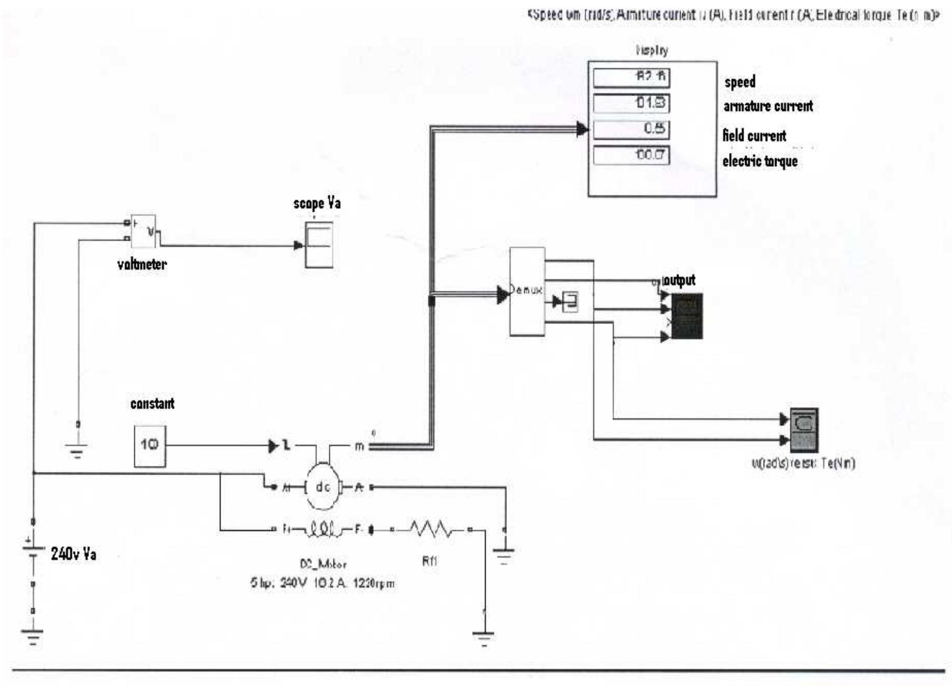
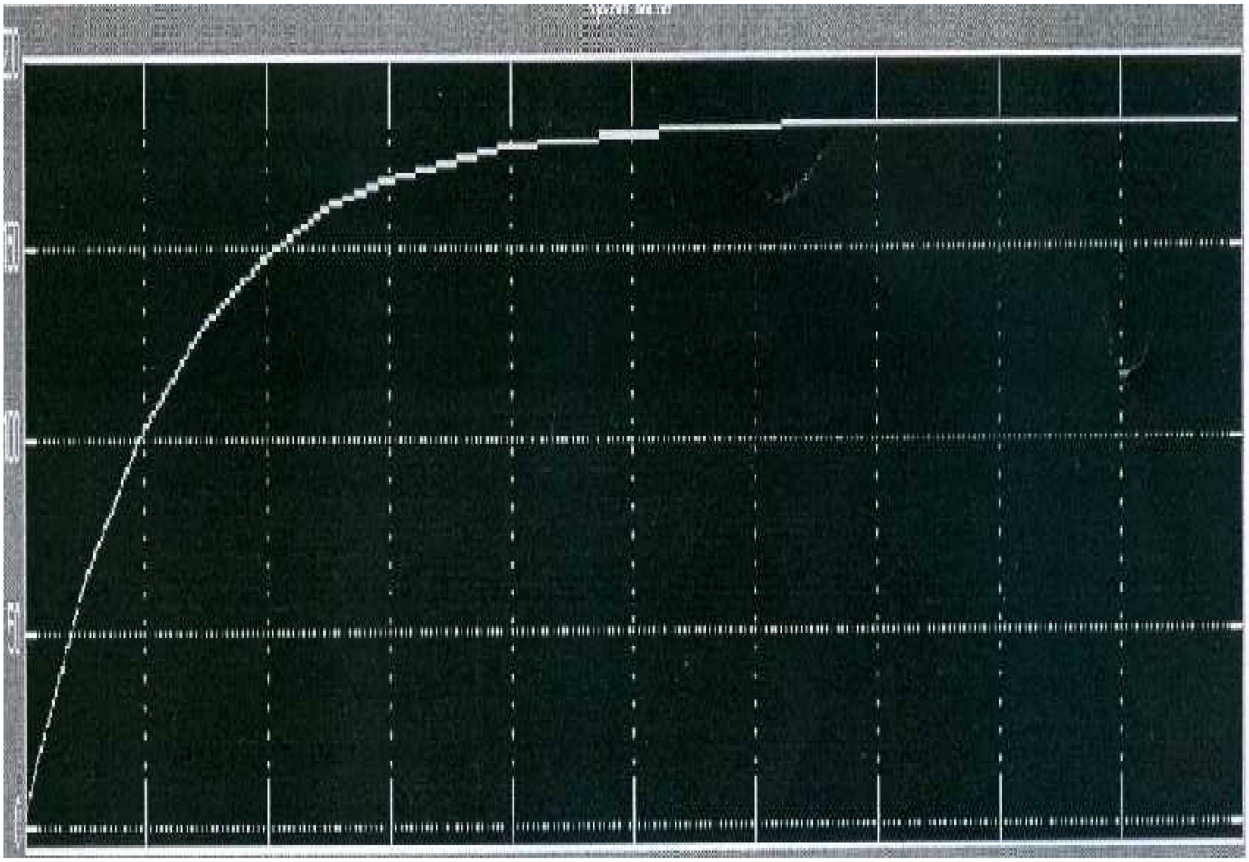


Fig.3.1 shows the study of characteristics of DC Motor.

In the field resistance control method, a series resistance (240 ohm) is inserted in the shunt-field circuit of the motor in order to change the flux by controlling the field current. It is theoretically expected that an increase in the field resistance will result in an increase in the no-load speed of the motor and in the slope of the torque-speed curve. Fig.3.1 shows the simulink implementation of the armature resistance control method. A dc motor block of Sim Power Systems toolbox is used. The dc motor block implements a separately excited dc motor. An access is provided to the field connections (F+, F-) so that the motor model can be used as a shunt-connected. The field circuit is represented by RL circuit (Rf and Lf in series) and is connected between the ports (F+, F-). The armature circuit consists of an inductor La and resistor Ra in series with an electromotive force EA and is connected between the ports (A+, A-). The load torque is specified the input port TL. The electrical and mechanical parameters of the motor could be specified by using its dialog box. Observe that 240V dc source is applied to the armature and field circuits. An external resistance Rfl is inserted in series with the field circuit to realize the field resistance speed control. The output port (port m) allows for the measurement of several variables, such as rotor speed, armature and field currents and the electromechanical torque developed by the motor.

In this fig.3.1, different scopes have been used to monitor the graphs of different parameter of DC Motor.

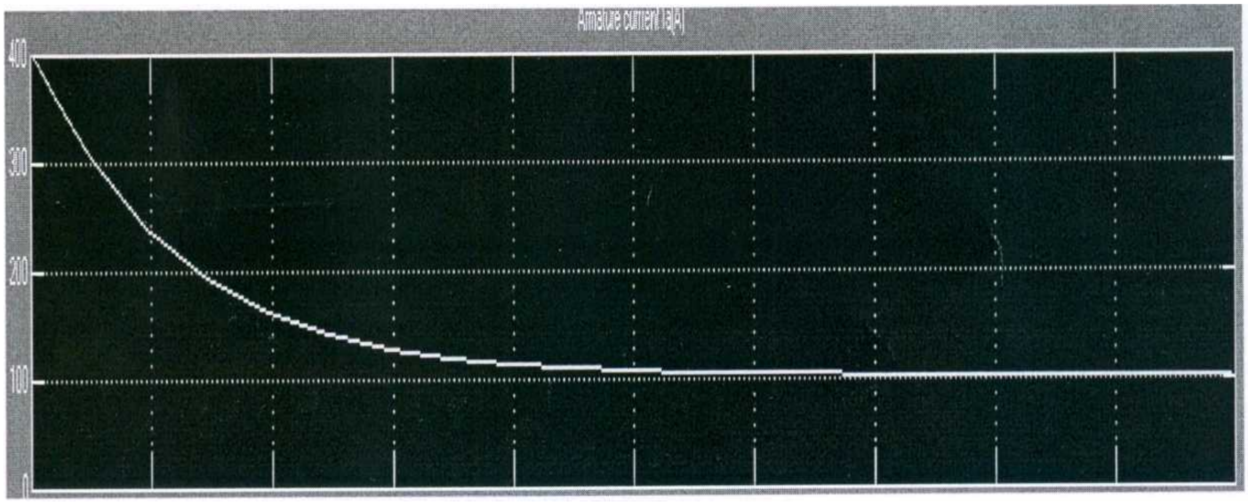
The different scopes of speed, armature current and torque developed when un controlled are given by 1.



Graph 1: Speed v/s Time

Here scope shows that when there is no control of the dc motor, the speed goes on increasing showing instability.

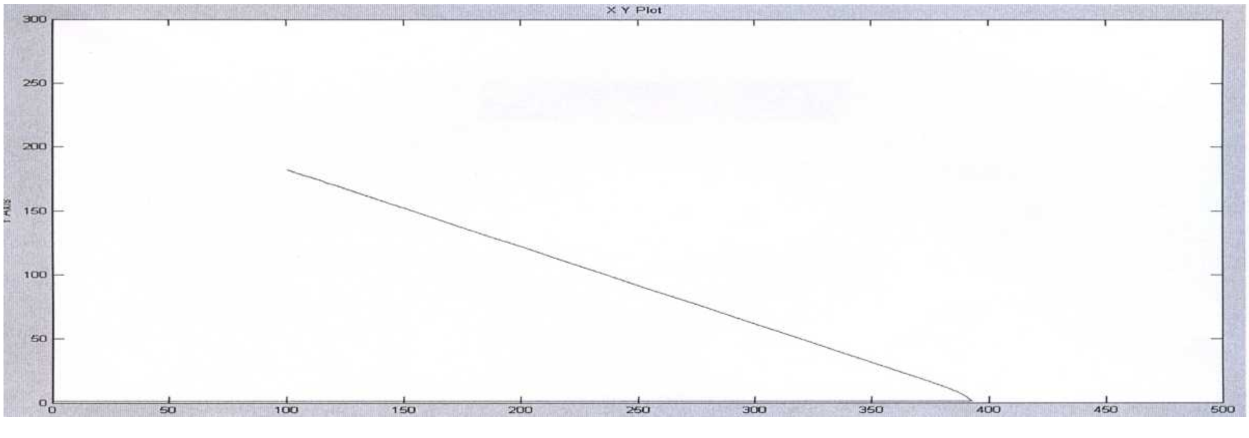
2.



Graph 2: Armature current v/s Time

Here scope shows that initially armature current attains a very high value which is undesirable, then as the time is increasing, armature current start decreasing and attains stability at that point. To limit ths undesirable thing, different types of starters are used.

3.



Graph 3: Speed v/s Torque

This graph 3 shows that the starting torque is very high starting speed is very low. But gradually as torque decreases, speed increases

This scope also shows the instability which needs to be controlled.

3.3 Block Diagram of DC Motor Speed (Using PID)

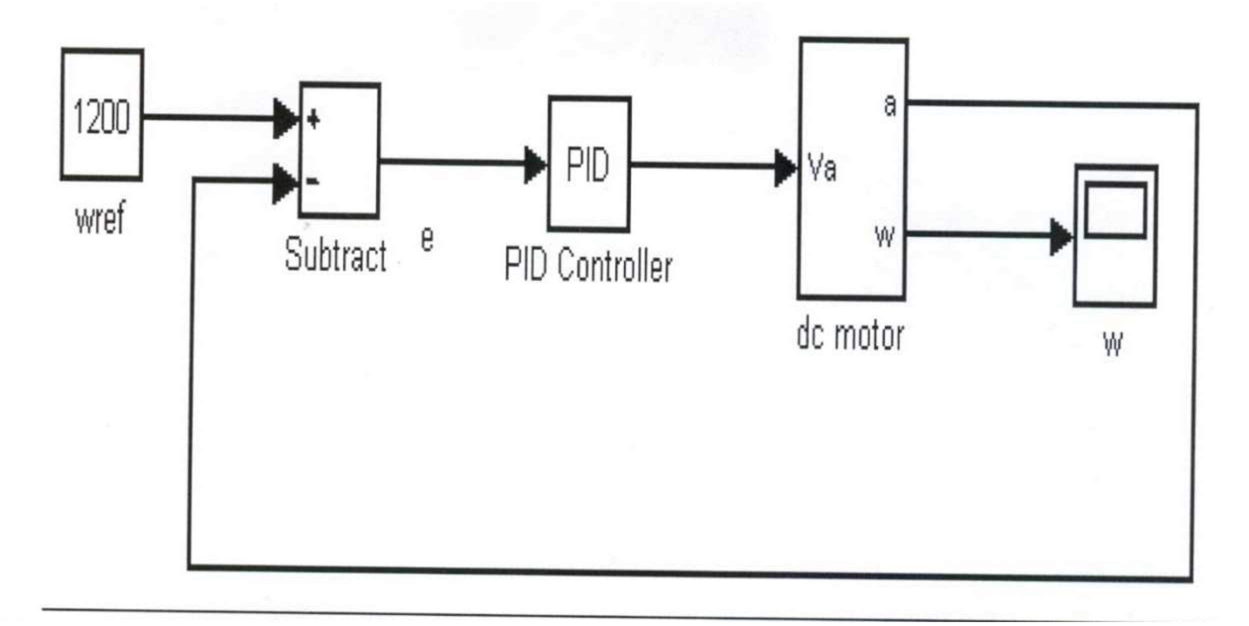
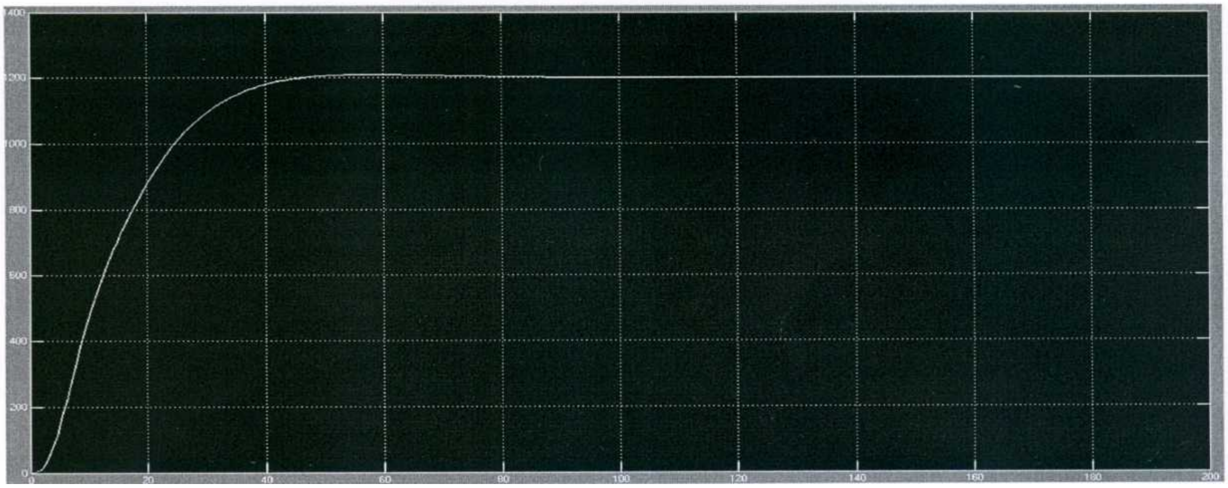


Fig.3.2 Speed Control of dc motor using PID Controller

Fig. 3.2 shows that PID controller is used to stabilize the speed of the dc motor. In this PID controller acts as a transfer function, which controls the speed of the dc motor.

Values given to this controller are:- **P=[1], I=[.25 0]**

3.4 Output Graph with PID controller



Graph 4: Speed v/s Time

Here scope shows that PID controller is taking 50-60 seconds to stabilize the speed dc motor.

3.5 Block Diagram of DC Motor Speed (Using Fuzzy Controller)

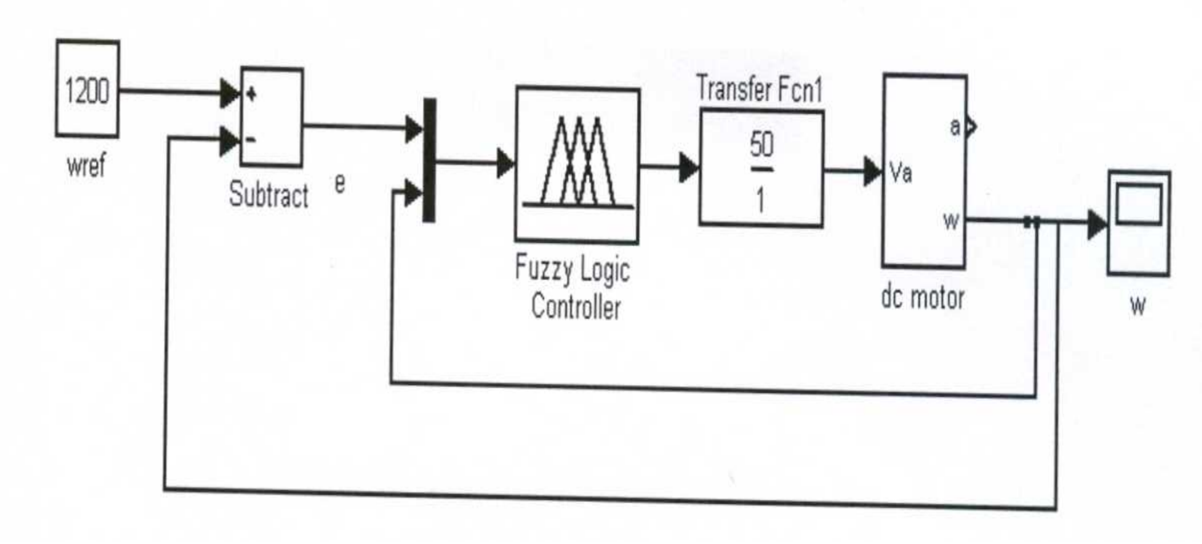
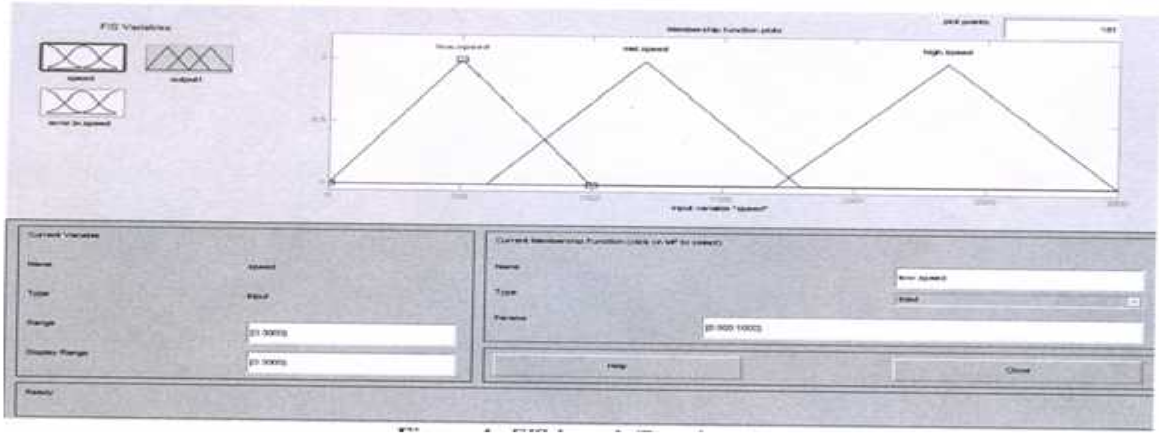


Fig.3.3 DC Motor with Fuzzy Controller

Fig.3.3 shows the simulink diagram of speed control of dc motor using Fuzzy controller. An extra transfer function has been added to properly display the rating of the dc motor speed on the scope. One input to the mux is error i.e. difference between reference speed and output speed and second input is the output speed.

3.6 FIS 1

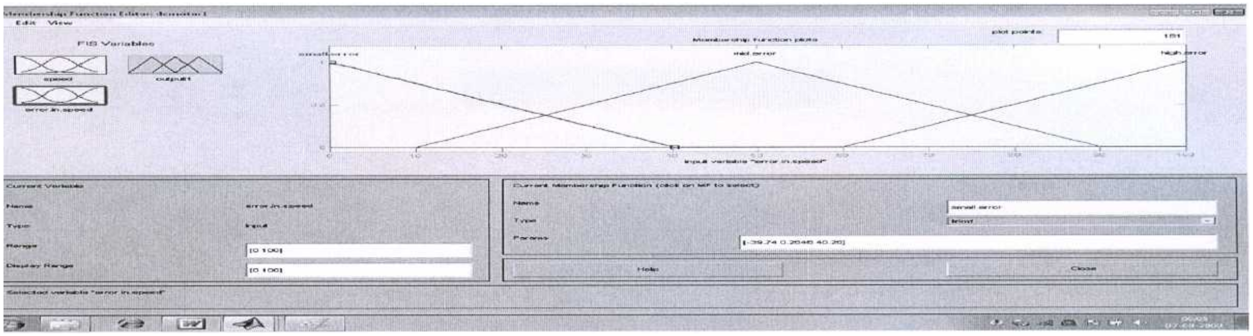


Name speed
Type input
Range [0 3000]
Display Range [0 3000]

Fig.3.4 FIS Input 1 (Speed rpm)

Fig.3.4 shows FIS diagram of the one of the inputs i.e. output speed of the fuzzy controller. The range selected for this FIS is 0-3000.

3.7 FIS 2

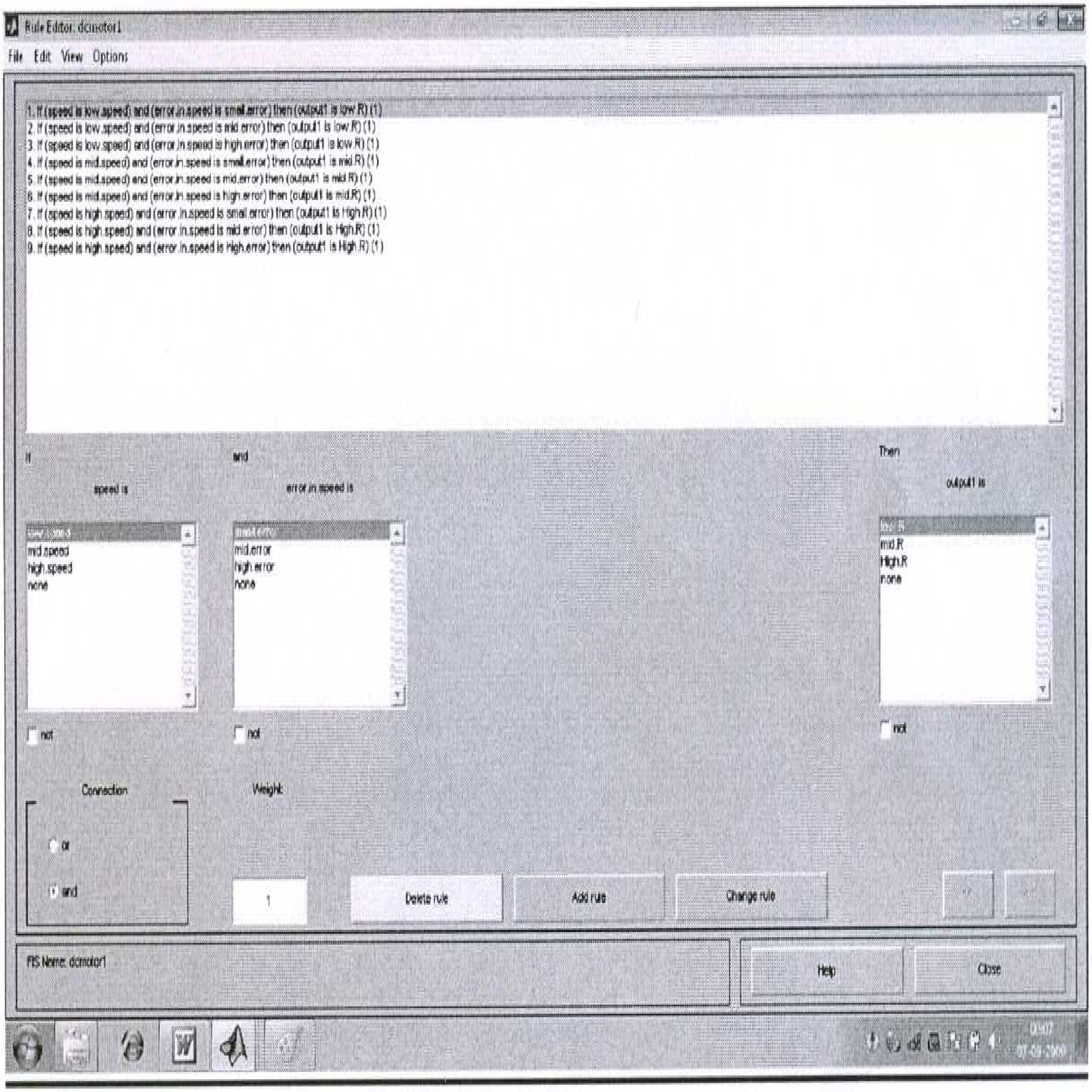


Name error in speed
Type input
Range [0 100]
Display Range [0 100]
Name small error
Type trimf
Params [-39.74 0.2646 40.26]
Selected variable "error in speed"

Fig.3.5 FIS Input 2 (Speed rpm)

Fig.3.5 shows the FIS diagram of the second input i.e. difference between reference speed and output speed of the Fuzzy controller. The range selected for this FIS is 0-100.

3.8 Fuzzy Controller Rules

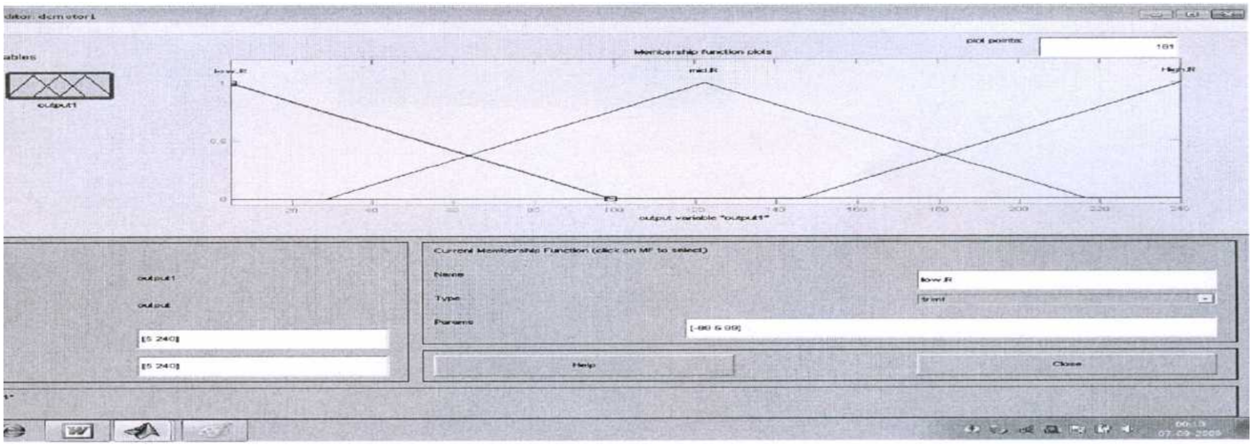


1. If (speed is low.speed) and (error.in.speed is small.error) then (output1 is low.R) (1)
2. If (speed is low.speed) and (error.in.speed is mid.error) then (output1 is low.R) (1)
3. If (speed is low.speed) and (error.in.speed is high.error) then (output1 is low.R) (1)
4. If (speed is mid.speed) and (error.in.speed is small.error) then (output1 is mid.R) (1)
5. If (speed is mid.speed) and (error.in.speed is mid.error) then (output1 is mid.R) (1)
6. If (speed is mid.speed) and (error.in.speed is high.error) then (output1 is mid.R) (1)
7. If (speed is high.speed) and (error.in.speed is small.error) then (output1 is High.R) (1)
8. If (speed is high.speed) and (error.in.speed is mid.error) then (output1 is High.R) (1)
9. If (speed is high.speed) and (error.in.speed is High.error) then (output1 is High.R) (1)

Fig.3.6 Fuzzy Controller Rule-Base

Fig.3.6 shows the rules made in the Fuzzy controller to control the speed of the dc motor.

3.9 FIS Output



Current Membership Function (click on MF to select)

Name low R

Type trimf

Params [-80 5 89]

output1

output

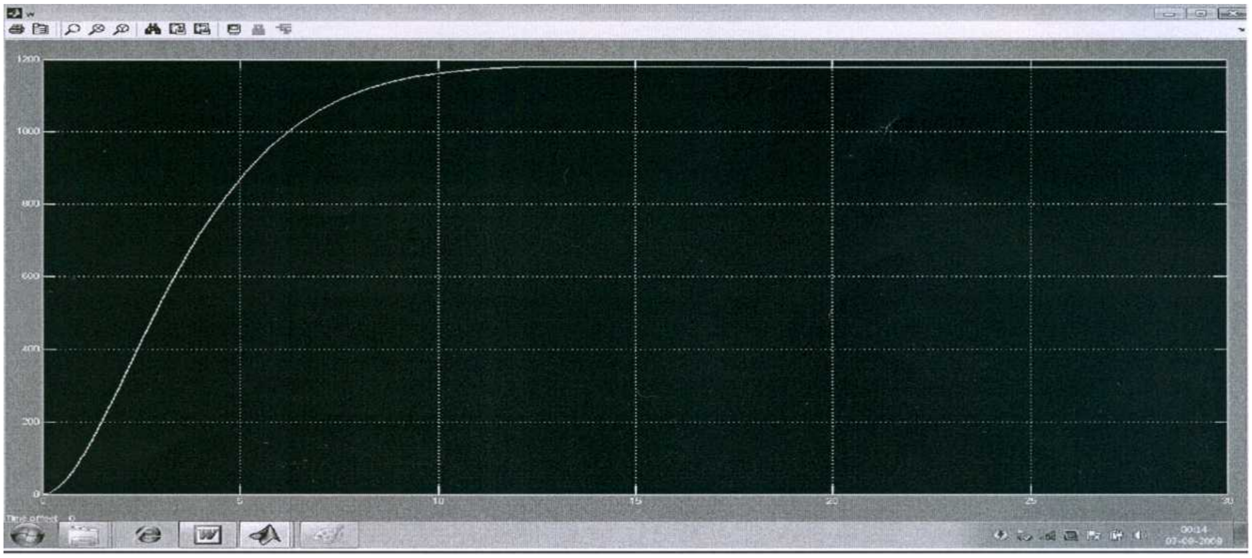
[5 240]

[5 240]

Fig.3.7 Fuzzy Controller Output

Fig.3.7 shows the FIS output of the Fuzzy controller. Range selected for this FIS is 0-240. This is the resistance of the dc motor that is to be controlled (say vary) to control the speed of the dc motor.

3.10 Output Graph of Fuzzy Controlled DC Motor



Graph 5: Speed v/s Time

Graph 5 shows that with the help of Fuzzy controller, speed of the DC motor is becoming stable in 10 seconds at 1200 rpm. And there is no spike initially.

Chapter 4

Conclusion and Future Scope

Stability of any operation for any application of DC motor performance is very important factor behind its selection for many applications. Stable system performs better and gives desirable results. So the conclusion which can be drawn out of this thesis work is that the speed of motor using fuzzy logic based controller achieves stability at its operating set point very fast then its conventional PID controller. In this work, we got the system speed stable in 10 seconds at 1200 rpm. Also the initial spike during the rise time is not appearing while used with the fuzzy logic controller. While the starting torque is increased, the starting speed is very low. But generally, torque decreases and speed increases and remains at its desired set speed without fluctuations as the uncertain reasons/causes of fluctuations are considered automatically in fuzzy.

By using the neural networks techniques the direct neural speed regulator can be easily implemented. This can be applied to regulate the speed of a DC servo motor, with better performance than a conventional PI controller. The advantages of this controller are that there is no need of reference model and dynamic model of plant. The on line learning capability of the neural regulator can be used to enhance the adaptability and stability for DC servo speed control system.

Neural networks also used for control of DC motors with unknown loads. This method may need lesser computations as well as lesser time to learn about the unknown operations and loads. A real-time implementation of control scheme for various dynamic systems using neural network makes the practical implementation of realistic speed control of DC motor drive easier. The control system with neural network based controller has higher tracking accuracy and powerful robustness than the system with tradition PID controller. It can be done using hybrid neuro-fuzzy techniques together as well. Hybrid fuzzy and neural network controller may be applied to achieve this function.

5. References

[1] W. Pedrycz, A.F. Rocha" Fuzzy-set based models of neurons and knowledge-based networks" IEEE transaction on fuzzy system vol. 1, no. 4, pp. 254-66, Nov. 1993

[2] Kang Hoon "Stability and control of fuzzy dynamic systems via cell-state transitions in fuzzy hypercubes" IEEE transaction on fuzzy system vol. 1, no. 4, pp. 267-79, Nov. 1993

[3] J. Nie and D.A. Linkens "Learning control using fuzzified self-organizing radial basis function network" IEEE transaction on fuzzy system vol. 1, no. 4, pp. 280-7, Nov. 1993

[4] Lee Jihong, Korea" On methods for improving performance of PI-type fuzzy logic controllers" IEEE transaction on fuzzy system vol. 1, no. 4, pp. 298-301, Nov. 1993

[5] H.A. Malki, Li Huaidong, Chen Guanrong "New design and stability analysis of fuzzy proportional-derivative control systems" IEEE transaction on fuzzy system vol. 2, no. 4, pp. 245-54, Nov. 1994

[6] H. Watanabe, W. J. Yakowenko "Application of a fuzzy discrimination analysis for diagnosis of valvular heart disease" IEEE transaction on fuzzy system vol. 2, no. 4, pp. 267-76, Nov. 1994

[7] G.F. Mauervol "A fuzzy logic controller for an ABS braking system" IEEE transaction on fuzzy system vol.3, no. 4, pp. 381-8, Nov. 1995

[8] K. Tanaka "Stability and stabilizability of fuzzy-neural-linear control systems" IEEE transaction on fuzzy system vol. 3, no. 4, pp. 438-47, Nov. 1995

[9] F. R. Rubio, M. Berenguel, E. F. Camacho "Fuzzy logic control of a solar power plant" IEEE transaction on fuzzy system vol. 3, no. 4, pp. 459-68, Nov. 1995

[10] D. Sinha, P. Sinha, E. R. Dougherty, S. Batman "Design and analysis of fuzzy morphological algorithms for image processing" IEEE transaction on fuzzy system vol. 5, no. 4, pp. 570-84, Nov. 1997

[11] Lin Yinghua, G. A. Cunningham III, S. V. Coggeshall "Using fuzzy partitions to create fuzzy systems from input-output data and set the initial weights in a fuzzy neural network" IEEE transaction on fuzzy system vol. 5, no. 4, pp. 614-21, Nov. 1997

[12] Xin Wang Li "Stable and optimal fuzzy control of linear systems" IEEE transaction on fuzzy system vol. 6, no. 1, pp. 137-43, Feb. 1998

[13] M. Figueiredo, F. Gomide, A. Rocha, R. Yager "Comparison of Yager's level set method for fuzzy logic control with Mamdani's and Larsen's methods" IEEE transaction on fuzzy system vol. 1, no. 2, pp. 156-9, May 1993

[14] Shieh Kung Ying, Ming Liaw Chang "A fuzzy controller improving a linear model following controller for motor drives" IEEE transaction on fuzzy system vol. 2, no. 3, pp. 194-202, Aug. 1994

[15] Liang Chen Cheng, Chih Chen Wen "Fuzzy controller design by using neural network techniques" IEEE transaction on fuzzy system vol. 2, no. 3, pp. 235-44, Aug. 1994

[16] H. Ishibuchi, K. Nozaki, N. Yamamoto, H. Tanaka "Selecting fuzzy if-then rules for classification problems using genetic algorithms" IEEE transaction on fuzzy system vol. 3, no. 3, pp. 260-70, Aug. 1995

[17] I. Hiraga, T. Furuhashi, Y. Uchikawa, S. Nakayama "An acquisition of operator's rules for collision avoidance using fuzzy neural networks" IEEE transaction on fuzzy system vol. 3, no. 3, pp. 280-7, Aug. 1995

[18] F. Klawonn, J. Gebhardt, R. Kruse "Fuzzy control on the basis of equality relations with an example from idle speed control" IEEE transaction on fuzzy system vol. 3, no. 3, pp. 336-50, Aug. 1995

[19] Jun Zeng Xiao, M. G. Singh "Decomposition property of fuzzy systems and its applications" IEEE transaction on fuzzy system vol. 4, no. 2, pp. 149-65, May 1996

[20] Orlando De Jesús and Martin T. Hagan "Back propagation Algorithms for a Broad Class of Dynamic Networks" IEEE Transactions on Neural Networks, Vol. 18, No. 1, January 2007

[21] Abhijeet V. Nandedkar and Prabir K. Biswas "A Fuzzy Min-Max Neural Network Classifier with Compensatory Neuron Architecture" IEEE Transactions on Neural Networks, VOL. 18, no. 1, January 2007

[22] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, pp. 338–353, 1965.

[23] Jianming Lu, Xue Yuan, and Takashi Yahagi "A Method of Face Recognition Based on Fuzzy c-Means Clustering and Associated Sub-NNs" IEEE Transactions on Neural Networks, Vol. 18, No. 1, January 2007

[24] Yong He, Guoping Liu, and D. Rees "New Delay-Dependent Stability Criteria for Neural Networks with Time-Varying Delay" IEEE Transactions on Neural Networks, Vol. 18, No. 1, January 2007.

[25] Tomohisa Hayakawa, Wassim M. Haddad, and Naira Hovakimyan "Neural Network Adaptive Control for a Class of Nonlinear Uncertain Dynamical Systems with Asymptotic Stability Guarantees" IEEE Transactions on Neural Networks, Vol. 19, No. 1, January 2008

[26] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," IEEE Trans. Neural Netw., vol. 1, no. 1, pp. 4–27, Mar. 1990.

[27] F. Remy, M. Week "Design of an Adaptive Control System for a DC Servo Motor" IEEE Transactions on Neural Networks, VOL. 19, NO. 1, 1995

[28] Kim Seungwoo, Cho Youngwan, Park Mignon "A multirule-base controller using the robust property of a fuzzy controller and its design method" IEEE transaction on fuzzy system vol. 4, no. 3, pp. 315-27, Aug. 1996

[29] A Costa, A. De Gloria, M. Olivieri "Hardware design of asynchronous fuzzy controllers" IEEE transaction on fuzzy system vol.4, no. 3, pp. 328-38, Aug. 1996

[30] A. Kapoor, N. Simaan, and P. Kazanzides, "A System for Speed and Torque Control of DC Motors with Application to Small Snake Robots", Proceedings of the IEEE Mechatronics & Robotics, Aachen, Germany, 2004.

[31] A. Y. Alanis, E. N. Sanchez, and A. G. Loukianov, "Discrete-Time Recurrent Neural Induction Motor Control using Kalman Learning", Proceedings of the 2006 International Joint Conference on Neural Networks, Vancouver, Canada, 2006, pp. 3951-3958.

[32] Yang Xiangyu, Yang Jin, etc, Model and simulation for brushless DC motor system [J], Journal of South China University of technology, 33, 28-32, 2005.

[33] Yildiray Y., Asif S., Khalid A. Sliding-mode neuro-controller for uncertain systems, IEEE Tans on industry and electronics , 54, 1676-1685, 2007.

[34] Zhiqiang Cheng, Chongsheng Hou "Global Sliding Mode Control for Brushless DC Motors by Neural Networks" 2009 International Conference on Artificial Intelligence and Computational Intelligence

[35] Carlos E. Casta˜neda, Edgar N. Sanchez "Real-Time Torque Control for a DC Motor Using Recurrent High Order Neural Networks" 2009 IEEE International Symposium on Intelligent Control Part of 2009 IEEE Multi-conference on Systems and Control Saint Petersburg, Russia, July 8-10, 2009

[36] F. J. Lin, R. J. Wai, and H. P. Chen, "A PM synchronous servo motor drive with an on-line trained fuzzy neural-network controller," IEEE Translate Energy Conversion, Vol. 13, pp. 319-325, 1998.

[37] F. J. Lin, R. J. Wai, and C. C. Lee, “Fuzzy Neural Network Position Controller for Ultrasonic Motor Drive Using Push-pull DC-DC Converter,” IEE Proceeding of Control Theory Application., Vol. 146, No. 1, pp. 99-107, 1999.

[38] G. Cybenko, “Approximation by Superpositions of A Sigmoidal Function, Mathematics of Controls,” Signals and Systems, Vol. 2, No.4, pp. 303-314, 1989.

[39] Yuan Kang, Ming-Hui Chu, Chuan-Wei Chang ”The Self-Tuning Neural Speed Regulator Applied to DC Servo Motor” Third International Conference on Natural Computation (ICNC 2007) 0-7695-2875-9/07 © 2007 IEEE