CHAPTER I
INTRODUCTION

1.1 General

The availability of substantial energy density in permanent magnet (PM) lead to the rapid development of PM field excited DC machines. Compact structure with simplified construction is attained by replacement of conventional field excitation in the rotor with a PM excitation in a synchronous machine. The incorporation of electrical commutator by power electronic converters in place of mechanical commutator leads to evolvement of the PM synchronous and Permanent Magnet brushless DC machines (PMBLDCM). These machines are nothing but the 'inside out DC machine with field winding on rotor with permanent magnets and stator winding on stator. The armature winding on stator aids in achieving high voltage and better cooling. Permanent magnets on the rotor prompt low rotor inertia and good dynamic characteristic in motor. Electronic commutation contributes to longer life span with less maintenance, high torque to weight ratio, higher speed range, less noise, low inertia. Due to PM rotor all of these advantages lead to high efficiency and improved dynamics in PMBLDC.

The permanent magnets synchronous machines are differentiated based on the wave shape of induced emf in their stator winding that is either sinusoidal or trapezoidal. The machine with sinusoidal type induced back emf is called Permanent Magnet Synchronous Machine (PMSM) and the other one with trapezoidal waveform is called Permanent Magnet Brushless DC (PMBLDC) machine. Another major difference between these two is the power density of PMBLDC machines, which is more as compare to PM synchronous machines. The drive system of PMBLDCM is simpler than PM synchronous are the later requires vector operation for control which is not required in BLDC.

1.2 Drive system of PM Brushless DC Motor

1.2.1 Operational Principle of PMBLDC motor

The electrical energy is converted into mechanical energy by the magnetic attractive forces between the permanent magnet rotor and a rotating magnetic field induced in the wound stator poles. The motor runs as the magnetic field produced by the winding shift position and the rotor moves to catch up with the stator field. The stator windings are energized following the energy sequence. In motor, commutation of phase current is done by tracing the beginning and end of constant portion of induced emf. Hence a three phase motors has six discrete positions in an electrical cycle which is traced by a Hall Effect sensor placed at 120 electrical degrees apart from each and generates synchronizing signals pertaining to commutation from rotor position.

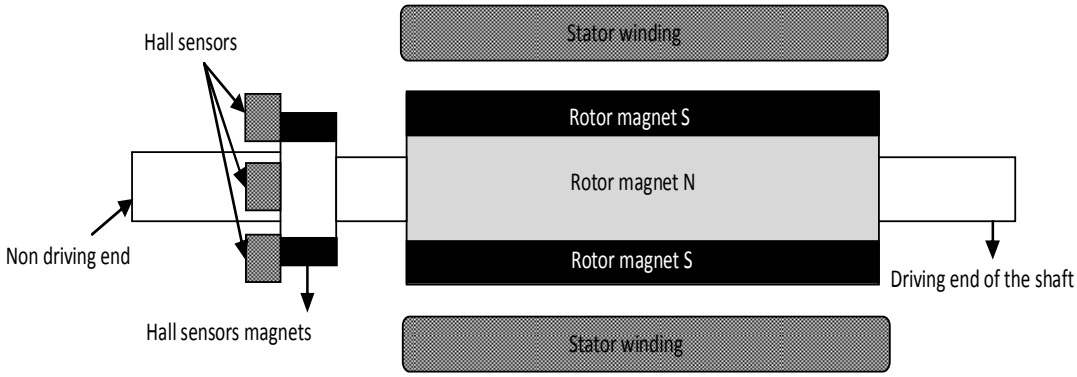

Figure 1.1 The construction of Hall sensors embedded PMBLDC motor

The Hall sensors have same number of poles as that of the rotor of motor and positioned on a wheel mounted on rotor. Hall sensor magnets on rotor are scaled down replica of rotor and produces same magnet effect as of rotor on the hall sensors embedded into the stator as shown in Figure 1.1. Rotor position is measured using Hall sensors mounted on the non driving end. These sensors indicate the rotor S or N poles passing near it, by producing a high or low signal. Each sensor output is high level for 180 electrical degrees and a low level for the other 180 electrical degrees. The rotor position is feedback through Hall sensors i.e. six discrete positions in grey codes which aids in determination of six step commutation sequence for three phase. Due to position

feedback depending on rotor position two winding are energized as one with positive, second with negative voltage and third is left non-energized.

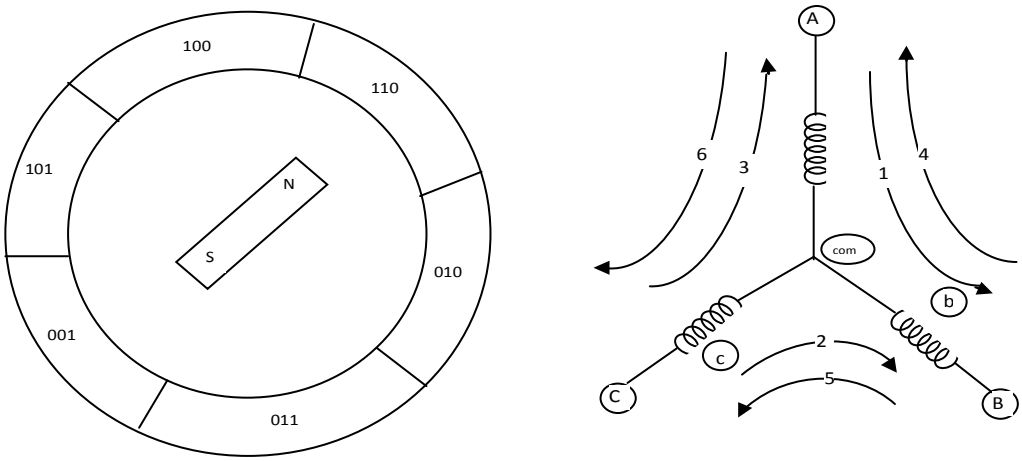

Figure 1.2 Position feedback in grey code and energizing sequence of phase winding

The current in respective phase is 120 degrees wide with respective polarity as of applied voltage. As respective phase current cannot rise and fall in zero time leads to quasi-square waveform of phase current.

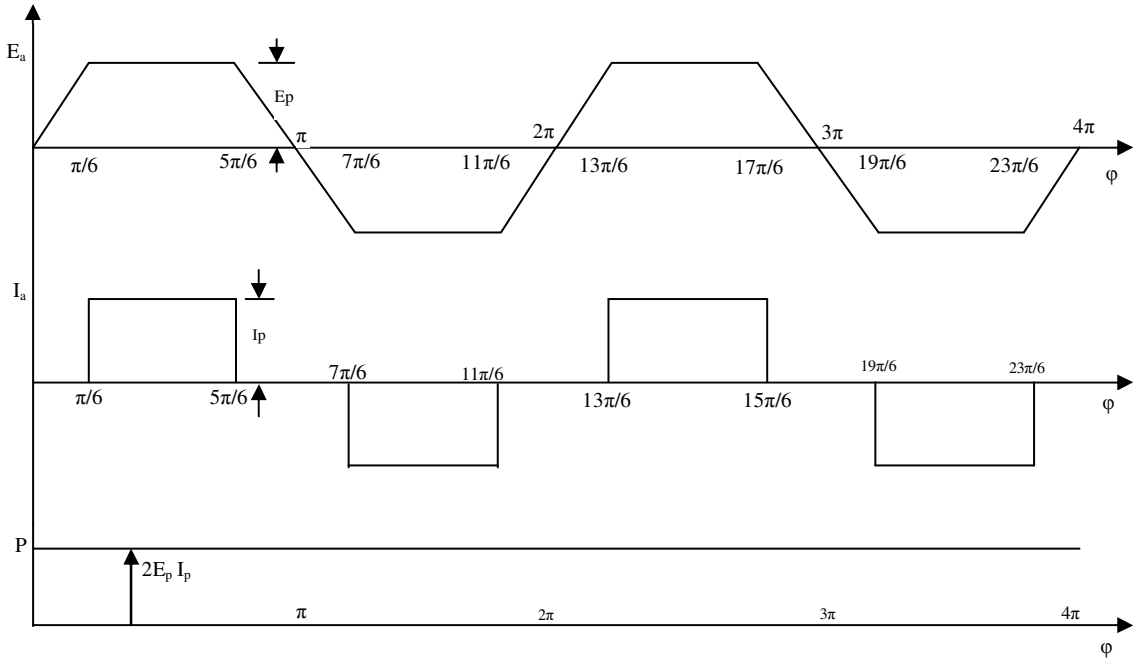

Figure 1.3 Back emf, phase current and output power waveforms

As shown in figure 1.3 wave forms of phase voltage (E_a), phase current (I_a) of one phase and the uniform output power (P) from three phases.

1.2.2 The PMBLDC Drive

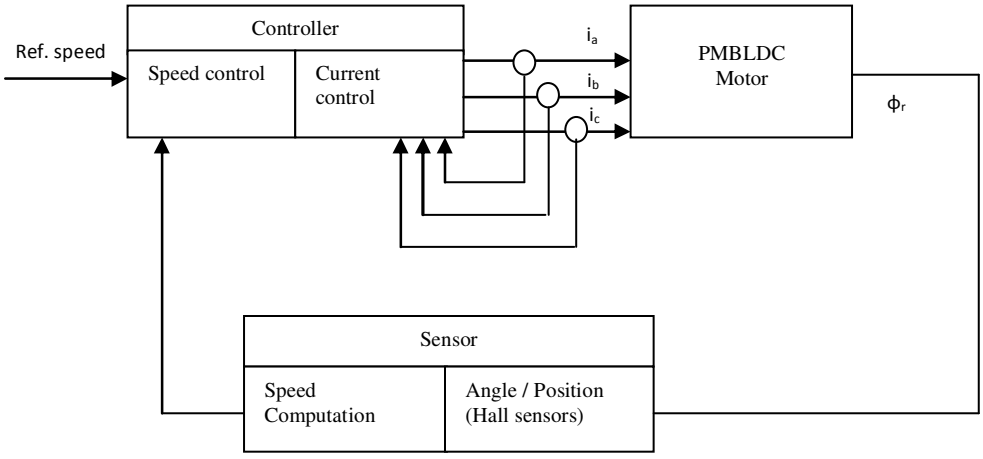

Figure 1.4 Block diagram of PMBLDC drive

An electrical drive makes is gauged under varing conditions such as starting, variable speed, speed reversal and load variation. A motor along with hall sensors, current measurements, optical encoder (position measurement), amplifiers, and feedback controller built a basic drive system. The controller consists of two blocks, each block is responsible for individual closed loop formation. The inner one is current control loop and the outer is speed control loop. These closed loops are formed as two signals speed and rotor position is getting feedbacked.

In outer loop reference speed is compared with the feedback from techogenerator i.e. actual speed. Based on the speed error a PI controller generates reference torque, form which in turn the phase reference current is generated.

Inner closed loop control the phase current $I_{p,}$ which depending on the commutation sequence provided by Hall-effect sensors. The phase currents are compared with outer loop output and processed using hysteresis band control technique. To track the current references, one only needs to determine the positions of the rotor at multiple of $\pi/3$ or 60 degrees as the current in any particular phase changes only at some multiple of 60 degrees.

1.2.3 Advantage of PMBLDC motor over induction motor

A three phase induction motor is the most extensively used motor in majority of applications. But now PMBLDC motor has emerged as an alternate to it especially in application areas such as electrical vehicles, automation, aerospace, robotics and medical instruments. A PMBLDCM has highest torque to weight ratio as compare to any other machine. For low rating application PMBLDCM maintains its ideal characteristics while induction machine suffers from flux distortion. The high efficiency, precise speed control and fast response of PMBLDCM make it better choice over induction motor.

Feature	**PMBLDC motor**	**Induction motor**
Power factor & peak point efficiency	Near to one (DC) & High peak point efficiency	Maximum upto 0.85 & High peak point efficiency but lower as compare to PMBLDCM
Torque/weight ratio	High -Since it has PM on the rotor, smaller size can be achieved for a given output power.	Moderate -Since both stator and rotor have windings, the output power to size is lower than BLDC.
Speed/Torque characteristics	Flat -Enables operation at all speeds with rated load.	Nonlinear - Lower torque as lower speeds.
Rotor Inertia	Low - Leads to better dynamic characteristics.	High –Leads to poor dynamic characteristics.
Slip	No slip is experienced between stator and rotor, both run at same frequency.	The rotor runs at a lower frequency than stator by slip frequency and slip increases with load on motor.
Stability over temperature range & controlling	Stable –Heat generation less & Easier as no complex control is needed.	Less Stable –High heat generation & Difficult as vector control is needed.
Starting current	Rated –No special starter circuit required.	Approximately up to seven time of rated –Normally Star-Delta starter is used.
Overall Efficiency	High	Medium

Table 1.1 Comparison between PMBLDC motor and induction motor

Although a controller is always required to keep PMBLDCM running but the same controller can be used for variable speed control. Whereas induction motor do not need any controller at fixed speed operation, but requires one for variable speed operation.

1.3 Paradigm and theory of controllers

In control engineering a controlled system is primarily characterized by its dynamic behavior which also determines the scope and quality required to solve a control task. The dynamic behavior of the system depends on the system parameters, input variables, output variables and operating conditions. The controller maintains the output at desired value by means of a control action. Any derivation of output from the reference input is detected by an error detector. The error thus detected is used as actuating signal for control action through a controller. The various control methods differ from each other depending on the factors based on which action is taken. Various kind of control methods that are availed and employed are discussed at length in this section.

1.3.1 Conventional types of controllers

1.3.1.1 The 'P' controller

Proportional or 'P' controller is one of the basic controller in control engineering. In 'P' control the actuating signal for the control action is proportional to the error signal. The proportional control is probably the easiest feedback control for implementation, where the error signal being the difference between the reference input signal and the actual signal (feedback) obtained. A constant denoted as 'K_p' called as proportional gain is multiplied with error signal, thus the controller actuating signal is obtained, which in turn is fed to the drive. Due to presence of fixed disturbance, a DC offset is observed in 'P' control. A trade off is to be made between the maximum overshoot and steady state error. As increase in gain value is desirable to reduce steady state error, the increased gain also increase the maximum overshoot value. So, there is need for further improvement in terms of eliminating steady state error.

1.3.1.2 The 'PI' controller

The limitation of a DC offset in 'P' control can be overcome by adding an integral term of error signal that provides desired DC stiffness to the system. The integral gain represented as 'K_I', increased value gives more stiffness at the cost of large overshoot. A sufficient value of 'K_I' gain in the controller will eliminate the DC offset, as the presence of even a small value of DC offset will make the integral term large. Although integral gain adds precision to the close loop control but it lacks in the wind up function that is needed to control the gain value during saturation. An improvement in the steady state error is introduced by 'PI' controller in the system dynamics.

1.3.1.3 The 'PD' controller

For a derivative control action the actuating signal consist of proportional error signal added with derivative of error signal. The gain of derivative term is represented as 'K_D' called as derivative gain, which induces a phase lead of 90 degrees in the loop. The derivative term can be represented as a difference term divided by sample time. The difference term is the last value of the position minus the current position value. The difference term divided by time gives a rough estimate of velocity, which helps the future position prediction. The derivative terms allow system's responsiveness to increase but make it more susceptible to noise. A noise is introduced in the system, if the change in position is constant and the sample time varies from sample to sample. The derivative action give high gain at high frequency this improves gain margin but affect the system adversely by adding gain margin at phase crossover frequency, which is typically at high frequency. Hence, the system suffers from noise that is spread evenly across the frequency spectrum and eventually worse in high frequency range. A low pass filter following the controller would eliminate the sample induced irregularities and oscillation in the system especially at high frequency. The control commands and plants outputs are usually of low frequency, so the presence of high pass filter does not affect their functioning.

1.3.1.4 The 'PID' controller

Conventional proportional integral derivative or 'PID' controllers are most commonly used controller. It can be obtained by combination of 'PI' with 'D', or 'PD' with 'I' control action. The three terms 'P', 'I' and 'D' work on the error value interpreted in terms of time, as 'P' depends on present value of error, 'I' depends on the summation of past values and 'D' depends on the rate of change of error predicting the future error. Each of the control action has different effect on the system and weighted sum of all the three actions is used by the PID controller. The PID control action can be divided in two zone based on the frequency, at lower frequency the 'K_p' gain dominated and at high frequency there is contribution from two gains 'K_p' and 'K_i'. The derivative action helps in setting 'K_p' at higher side then that can be set generally. All this actions work in tandem with in the close loop, depending on the command and the feedback signal of the system. Although the PID is superior to the P, PI, PD controller in term of system dynamics response but it come with an expense of increased sensitivity towards the changes in plant model. The rigorous task of tuning a PID controller is also a matter of great concern.

1.3.2 Non-conventional type of controller

1.3.2.1 Fuzzy logic based controller

A controller designing for a complex process with multiple input and output, ill-defined process and multiple control objectives with conflicting interest become a cumbersome task using conventional approach. In the conventional method a control system development starts with the identification, modeling and simulation of the complex process. The control system development of such complex process is difficult especially when there are real time implementation considerations. Most of the systems are defined with restrictive assumptions such as linearity and time invariable for making an accurate model of the systems. The controller that is designed for such model that are developed based on such assumptions, can have either linear or non linear nature and it will certainly need to be tuned. The heuristics enter the design process when the conventional control design process is used as long as one is concerned with the actual

implementation of the control system. It must be acknowledged, however, that conventional control engineering approaches that use appropriate heuristics to tune the deign have been relatively successful as we all know that the vast majority of all controller currently in operations are conventional PI,PID controller. The following question may arises how much of the success can be attributed to the use of mathematical model and conventional control design approach and how much should be attributed to the clever heuristics tuning that the control engineer uses upon implementation? If we exploit the use of heuristic information throughout the entire design process can we obtain higher performance control systems?

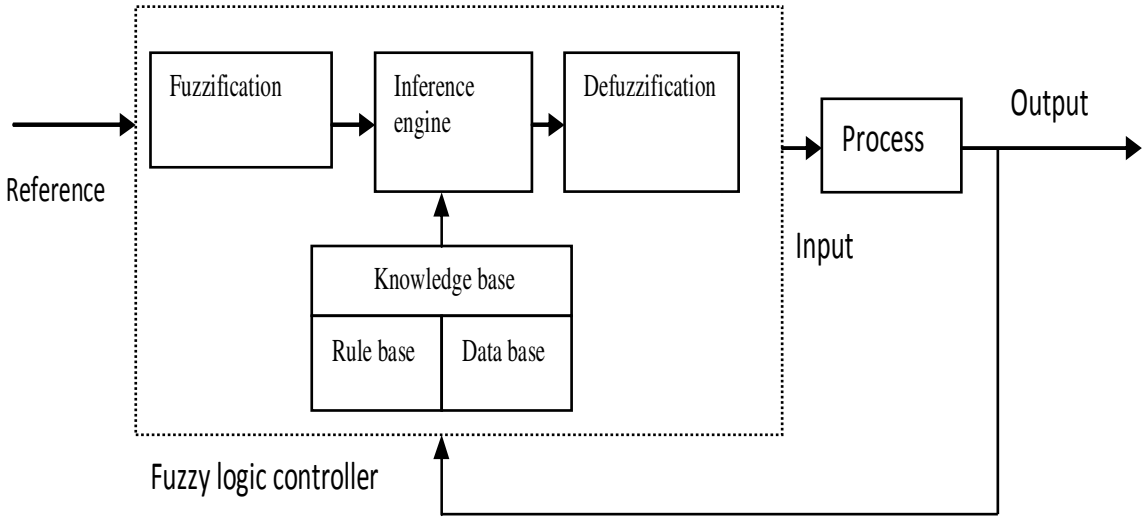

Figure 1.5 Block diagram of a fuzzy logic controller

A formal method for representing, manipulating and implementing a human's heuristic knowledge is fuzzy logic. It is also called expert control, as the human expertise pertaining how to control a system is use to implement the fuzzy control. The knowledge is used in decision making, is applied using a set of rules. The fuzzy set theory plays an important role in dealing with uncertainty while decision making in complex or ill-defined processes. Moreover, fuzzy technique utilizes a linguistic rule base that is designed taking advantages of system qualitative aspect and expert knowledge. These rules make the rule base in the simplified if-than format and the inference mechanism reasons over the information in the knowledge base, the process

outputs, and user specified goals to decided what inputs to be generate for the process so that the closed loop fuzzy control system behave properly. Overall fuzzy control is based upon the heuristic knowledge to achieve good control, whereas in conventional control the focus is on the use of mathematical model for control systems development and subsequently use of heuristic in implementation.

The fuzzy logic controller consists of six major blocks:

I. *Normalization*

Also called as input normalization perform a scalar transformation which maps the physical values of control inputs into a normalized universe of discourse (normalized domain). For a non-normalized domain, this block can be discarded.

II. *Fuzzification*

The fuzzification block converts a crisp value of the control input into fuzzy values, to make it compatible with the fuzzy set representation of the control input variable in the rule antecedent. The choice of fuzzification strategy is dependent on the inference engine, which can be either composition based or individual-rule-firing based.

III. *Knowledge base*

The knowledge base of a fuzzy control consists of data and rule base.

- *Data base*: It provide all the necessary definitions for the fuzzification process such as membership function, fuzzy set representation of the input-output variables and the mapping functions between the physical and fuzzy domain.
- *Rule base*: The rule base present the control strategy employed by expert control engineer knowledge or heuristics expressed in the form of rule sets in if-then format. The rules are based on the concept of fuzzy inference, antecedent of the rule is associated with consequent using linguistic variables.

IV. *Inference engine*

The inference engine depending on input conditions gives the response of controller that is determined by processing the rule base module. The value of the least true antecedent is applied to the strength of the rule. When more than one

rule is applied for the same action, the highest strength rule is use in general practice. They are generally two types of inference mechanism:

a. *Composition based inference*: All the rules are combined and then fires using fuzzy precomposition.
b. *Individual-rule base inference*: Each rule is fired individually depending on control input then result obtained from each rule is combined into one overall fuzzy set.

V. *Defuzzification*

The output response of the controller must be non fuzzy in nature. This module defuzzifies the response after evaluating the rule base module. There are five main types of defuzzification methods, generally the weighted average method is used for defuzzification.

VI. *Denormalization*

The value of control output obtained after defuzzification is normalized and need to be mapped back into physical domain. This reverse of normalization mapping is denomalization.

The control system consists of other elements apart from fuzzy, that are sensors, analogue to digital converters, digital to analogue converter circuits.

1.3.3 The Hybrid controller

Conventional controller such as PI, PD and PID controller are inevitably used in industry owing to its continuous nature and simple structure along with good performance in a wide range of operating conditions. The performance of such controller is adversely affected by parameter variations and unknown linearities like saturation, dead zone, delay, limit cycle etc. Frequent load variation and harsh environment leads to occurrence of oscillation in system's dynamic response. With a PD controller, a system with dead zone shows a steady state error, whose value increase directly with increase in dead zone width. An integral windup problem is common in PI controller showing saturation phenomenon in actuators. In many systems conventional PI controller is employed as a low cost solution for controlling. However, deviation of the system parameter or load conditions causes the performance of the closed loop system to

deteriorate, resulting in larger overshoot, longer rise and settling times and possibly even in unstable system.

A PMBLDC motor has nonlinear characteristics, so a non linear controller would function superior than a linear PI controller. A model free concept of fuzzy controller which is better equipping for non linear control is a viable option. The custom set of PI gains based on fuzzy reasoning attach control iteration has been used to control of high performance motor drive. The fuzzy system allows for increased sensitivity by gains across feedback error ranges. Nonetheless, the main problem with fuzzy logic control (FLC) is that there is no systematic approach for the construction of the fuzzy controller such as scaling factor, linguistic rules and shape of fuzzy sets.

A controller that can harness the benefits of both the controller i.e. fuzzy and conventional PI would suits best. For this purpose a controller can be proposed which is a hybrid of the FLC and the conventional controller. This controller simplifies the task of developing rules for designer, who only needs a roughly correct initial set of rules. It reduces designers burden of manually fine tunning, as controller is able to adapt automatically to new environment. If the output of a speed controller is a combination of outputs of two speed controller i.e. FLC & PI, combined together as a weighted sum to eliminate certain disadvantage then the resulting controller is referred to as a hybrid controller.

A hybrid controller can be implemented in two configurations depending upon the control objective. The hybrid concept is based on the augmentation of FLC to the PI controller. This can be done in two ways:

- ***Series hybrid***: The FLC is augmented in series with the inline PI controller and based on feed-forward logic. Here, FLC work as precompensator to modify reference speed signal that is observe by PI controller. The FLC as precompensator is augmented in series with the PI controller hence, called a series hybrid.
- ***Parallel hybrid***: To incorporate robustness and to make system adaptive, the FLC modify the value of proportional and integral gain of the PI controller. In this feedback loop configuration, FLC induce self tuning capability in PI.

1.3.3.1 The Series Hybrid PI (Fuzzy precompensated PI) controller

The hybrid control scheme consists of a conventional PI control structure together with our proposed fuzzy pre-compensator. The purpose of fuzzy pre-compensator is to modify the command signal to compensate for undershoot, overshoot and steady state errors present in the output response when the plant has unknown nonlinearities. This feed-forward logic is achieved by the advance alternation of the reference control signal in accordance with system response. The two inputs the speed error and rate of change of speed error are fed to the fuzzy precompensator and the output of it added to the actual speed reference signal to generate the modified speed reference signal.

The series hybrid configuration aids in robustness, disturbance attenuation, accuracy and improved response speed.

1.3.3.2 The Parallel hybrid PI (Self tuning PI) controller

In model base control strategy, the controller has to deal with many uncertainties such as system parameter, external load disturbance, and frictional force. To cope up with these uncertainties and non-linearity additions of artificial intelligence is done in the controller. Fuzzy logic based intelligent control technique for tuning the conventional controller provides a formal method for implementing the human heuristic knowledge in the form of control rules. The FLC tune the proportional and integral gain of the conventional PI controller in parallel depending upon the speed error and the change in speed error.

The adaptive mechanism based on fuzzy logic rejects the load disturbance and good tracking response is obtained with less overshoot, minimum rise time and less steady state error.

1.3.4 The Reduced rule base hybrid controllers

The computational effort required for fuzzy calculation is high, fuzzy inference process is mostly executed by personal computer (PC), hence making the practical implementation difficult.

Unlikely the PC platform, the hardware involved in embedded system usually have low computing power and limited memory space, since the aim is to carry out a particle real time task with low cost devices. Thus, efforts have to be made in order to build the hybrid controller with limited hardware.

The research work has been done to realize a hybrid combination of the FLC and a conventional PI controller for PMBLDC drive with reduced rule base. For drive applications the motor is generally operated at maximum torque in response to any disturbance/ change in reference command so as to reach the set point/ track the reference point at fastest possible way. Other controller is driven at high gain and the response is saturated to limit the torque to its rated value, which may offer to avoid the requirement of any intelligent control in that range. Thus, the influence of FLC/intelligent controller finds its usefulness near set point. The control influence of FLC is utilized in specific region of operation. By doing this, rule base is drastically reduced and thus the computation time and memory utilized for realizing the FLC on a microcontroller or DSP paving the way for implementation in real time embedded applications for drive.

1.4 Scope of the work

We have seen that the excitation to the PMBLDCM is provided by permanent magnets placed on the rotor, the torque developed in the machine is solely dependent on the stator phase currents similar to a separately excited dc motor. By choosing a suitable controller the dynamic performance of the machine can be improved to a great extent. It is therefore required that, various controllers for the speed control of the PMBLDCM should be studied, modeled and simulated to identify the suitable controller for appropriate conditions. The scope of work in the present thesis is mainly to construct the PMBLDCM drive system in the MATLAB/SIMULINK environment, then simulation studies is carried out for the speed control of the drive using the PI controller, Fuzzy logic controller, series hybrid, parallel hybrid and develop a reduced rule base series hybrid PI controller and reduced rule base parallel hybrid PI controller for varying operating conditions.

1.5 Thesis outline

The contents of the thesis have been divided into the following chapters:

Chapter 1

The basic construction, operating principle, applications and the advantages of the PMBLDC machine have been discussed in detail. The different types of controllers and the scope of the work were also discussed.

Chapter 2

This chapter elaborately describes the literature review of the different speed controllers and the significant developments in their respective areas. It also covers the various applications using the controllers PI, PID, fuzzy, series hybrid, parallel hybrid, reduced rule base series hybrid PI and reduced rule base series hybrid PI controller. The different hybrid controller configurations proposed and implemented, and their methods are discussed in brief here.

Chapter 3

This chapter presents the modeling and simulation of the drive system, with the PI, FLC, reduced rule base series hybrid PI and reduced rule base series hybrid PI controller. The various components of the drive system are discussed in detail.

Chapter 4

This chapter presents in detail the responses of the simulation models of the drive during different operating conditions such as the starting, load perturbation and speed reversal. The current, torque and the back emf wave forms were also observed during the operation. The detailed comparative study in terms of adaptive nature, settling time, rise time and steady state error on using different controllers is also presented.

Chapter 5

This chapter contains the main conclusions based on the investigations carried out on this work. It also enlists the scope for further investigations in the speed control of the PMBLDC machine.

CHAPTER II

LITERATURE SURVEY

2.1 Introduction

The reported literature reveal a rising interest towards application specific microcontroller and embedded system, which has also included the area of motor control. Many control strategies are reported employed for performance enhancement of PMBLDC motor drive. Among them are non conventional controller based on fuzzy logic, neural network, genetic algorithm and neuro-fuzzy which seems quite promising. Artificial intelligence is incorporated in the control system using such techniques. But the expense is paid in term of high computation needed for their operation/use. The concept of hybrid using fuzzy logic with conventional control has been introduced for electric drives understanding its continuous nature. Fuzzy logic offers a convenient way of designing controller from experience and knowledge of the controller designer. Using hybrid controllers a significant improvement in response of the PMBLDC drive has been reported with the elimination of steady state error, overshoot and oscillation. On the other hand advancement in the processor technology for microcontroller and DSP, practical implementation of such control in the real time is now achievable with hybrid controllers. All this has paved the way for embedded hybrid control system for motor drive applications. The requirement of high computation speed, complex calculation and large memory, force the optimization of control algorithm is a requisite.

2.2 Literature review

The embedded fuzzy system for specific application, such as improvement of speed response of the drive using various techniques has gained moment in recent time among the researchers. Fuzzy logic is utilized vividly in control structure of many speed controllers. Reported improvement in performance of speed controller is evaluated based on the dynamics response parameter i.e. oscillation, overshoot, undershoot, starting time, rise time, steady state error, disturbance rejection etc. is studied for different configuration of controllers. Feasible practical implementation of hybrid controllers

(conventional + non-conventional) presented use platforms such as DSP, FPGA and microcontroller has been looked upon. Limitations faced in such platforms and researcher methods to overcome these issues are discussed at length in present section.

The permanent magnet brushless DC (PMBLDC) motor has emerged as a suitable option for variety of applications owning to simple structure along with small frame size, large torque to weight ratio, large operational range, good dynamic performance, high reliability, high efficiency and noiseless operation [1]. The close loop control is implemented using conventional Proportional Integral Derivative (PID) controllers in the industrial applications owing to its simplicity and continuous nature. Good performance in wide range of operation makes PID controller a unanimous choice. However, in many applications the derivative term is being negated by setting the derivative gain to zero [2], leaving the versatile PI control for majority use in drive applications. In an industrial set up large load variation, nonlinearities and parameter variations are common phenomenon. Under such circumstances the performance of the controller deteriorates, resulting in large overshoot, longer rise and settling time. Thus, a fixed gain PI controller cannot effectively control the systems with changing parameter or having strong nonlinearities; and may need frequent on-line tuning. It is therefore, the control needs to be amended and must have an adaptive nature, so as to adjust itself according to the harsh environment, nonlinearities and parameter variations. Moreover, designing a PI controller becomes cumbersome if multiple objectives with conflicting interest have to be achieved [3].

A non linear controller like fuzzy logic controller (FLC) seems to be a logical replacement for linear model based conventional controllers. The linguistic rule base with the advantage of system qualitative aspects and expert knowledge in FLC is easier to formulate than tuning the PI controller using empirical methods [4]. The rigorous process of tuning a controller, with method such as Ziegler and Nichols, poles assignments and hand tuning, is a sever time consuming task. Although a FLC response is fast and performs well in the presence of non linearities, it needs more information to compensate them under varying operation conditions.

Fuzzy logic controller (FLC) exhibit offset in the response, whereas PI controller has superior performance near steady state conditions but suffers from sluggish response and

occurrence of overshoot. Even though a FLC delivers fast response and functions well even in the presence of a nonlinearities, a PI controller is always preferred as the front end controller, supported by the FLC at the back end.

Many configurations have so far been reported by researchers using fuzzy logic to implement PI controller, as fuzzy precompensated [5], fuzzy logic based self tuning [6] and fuzzy optimization, based on genetic algorithm [7]. Further improvement is reported by designing parallel fuzzy PID controller [8] and hybrid fuzzy-PI with novel switching [9]. In this architecture the integration of classical PID with fuzzy PID is done by switching between the two controllers. The Fuzzy PID consists three subcontrollers namely, fuzzy based proportional, integral and derivative controllers. Each fuzzy subcontroller with their individual rule base put tremendous burden of computation as three parallel fuzzy inference mechanisms has to be evaluated. Disturbances in the output are inevitable in these strategies and tracking time is large as frequent switching is done between the controllers.

The Hybrid of PI and fuzzy reap the benefit of both, where each complement the other's short coming [9, 10]. The concept of hybrid controller can harness the benefit of both fuzzy controller and conventional PI is being proposed, that does not depends completely on one type. Typically intermittent duty loads in process industry need fast acting and precise controller to ensure quality production with minimum time. The quest for such fast controllers with precise operation is therefore requisite with PMBLDC actuator motors envisaging enhanced production role.

A hybrid fuzzy- PI controller can be implemented as a speed controller for PMBLDC where the PI controller is active near the steady state conditions and the fuzzy controller active during transient conditions [9]. The hybrid configurations are proposed in two architectures without switching function depending on the control objectives. Series hybrid combination in which FLC act are precompensator for the reference signal advantage being, easy modification in present configuration by just adding a fuzzy precompensator in series with PI controller. The PI controller when operating with system having variation of operating conditions requires a frequent tuning of gain as per the conditions, the task is time consuming and complex. The task of tuning of PI gains can be accomplished by a fuzzy controller in parallel. Also a fuzzy control system has

good robustness which can restrain influence of disturbance and parameter fluctuation effectively. A self tuning controller is obtained which posses the adoptive nature with robustness [2, 6].

The fuzzy inference mechanism developed on DSP or PC can quickly process fuzzy computation to generate designed control action. With the availability of compact, high power computing processor, fuzzy logic control schemes are not difficult. A dedicated chip was developed to achieve the fuzzy inference process by real time online context switching [11]. Such hybrid controller with integrated fuzzy, has been implemented on many platforms such as PC, DSP, microcontroller, FPGA, integrated circuit [12, 13]. But the physical size of the system may become too big and quite expensive for a small motor application using the hybrid control based on fuzzy logic. The embedded system implementation in microcontroller seems a viable option. But hardware involved in embedded system usually present low computing power and limited memory space, since the aim is to carry out a particular real task with low cost device.

The reduction in computation burden has emerged as an important issue which requires massive attention. The concept of DSP-based switching motor controller is reported in literature [14] and further improvement is also reported using genetic optimization [15] and fuzzy-neural-network controller [16] reported improvement on PI controller, as aforesaid associated with highly computer intensive algorithms, which, with the presently available digital signal processor or microcontroller unit (MCU) cannot be implement in real time. This required highly parallel super computers to actually implement in conformity with simulation and is definitely very costly. Some efforts are reported in literature where, membership functions are reduced over entire universe of discourse, sacrificing the rules and thus may results in high degree of discontinuities and loss of effective control. Few researchers have written for reducing the size of total memory utilization for implementation of FLC by reuse of memory units [17]. Other researchers have either reported for processing of FLC at low sample rate [18], or by switching the converters at different instances [9, 19] causing slow response and disturbances in output.

At present as important advances are being made in DSP platform based practical implementations of hybrid control design, it is hoped that these advances may begin

influencing the implementation of hybrid control for task specific embedded system design and can also provide computation reduction to attain low cost solution.

2.3 Conclusion

The detailed analysis of literature has revealed that extensive work is being carried out in the field of high performance drives for PMBLDC motor and other motors. The advancement in microelectronics, embedded systems, microcontrollers, power electronics and simulation software has facilitated the development of various control strategies for its implementation in hardware. Performance enhancement and increase in robustness encourages the use of fuzzy logic control as it improves the both, in the presence of nonlinearities and varying operation conditions. Different controller configurations for speed control of PMBLDC motor are tried for better dynamic response. The hybrid controller without switching requirement is proposed in two configurations, aids disturbance less transient response. A DSP or PC platform is used to implement hybrid controller as complex fuzzy evaluation is done easily by them, but it requires huge hardware leading to high cost. A more economical way is needed to be developed for a full embedded system. Hardware involving embedded system via a low cost device would be more apt for low computing power and limited memory space. Reduction in computational effort and memory requirement can be achieved with optimization of fuzzy control algorithms. Some methods have been found which can be implemented with minimum control hardware and with lesser need for processing.

CHAPTER III

MODELING OF CONTROLLERS AND DRIVE FOR PMBLDC MOTOR

3.1 General

The paradigm and concept of controller for speed controller of brushless DC motor and detailed exhaustive literature review has been covered in the previous chapters. The present chapter deals with modeling and simulation of the drive with the different controllers the proportional Integral, fuzzy logic controller, series hybrid, parallel hybrid, reduced rule base series hybrid PI controller and reduced rule base parallel hybrid PI controller in the MATLAB/SIMULINK environment

3.2 System configuration

Drive system of motor is modeled with the help of mathematical equations, describing the behavior of the machine. The mathematical model is designed based upon modeling each component by set of equations and combining them together. Figure 3.1 shows the basic block diagram of PMBLDC motor drive configuration. The drive consists of speed controller, reference current generator, PWM current controller, position sensor, the motor and IGBT based current controlled voltage source inverter (CC-VSI).

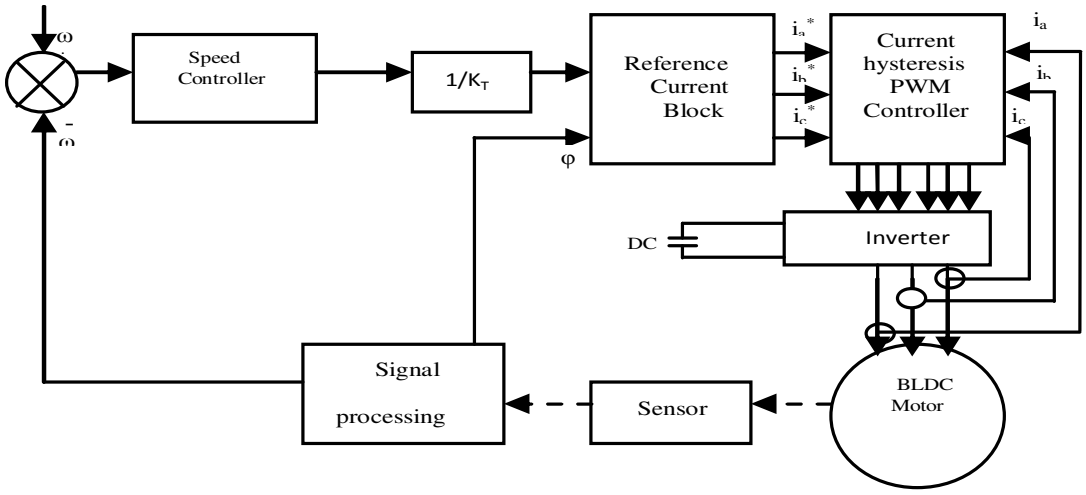

Figure 3.1 Detailed block diagram of PMBLDCM drive

The speed of the motor is compared with its reference value and the speed error is processed in a speed controller. The reference torque output of the controller is limited to restrict the operation of the drive within permissible range of current. The reference current block generates the three phase reference currents (i_a*,i_b*,i_c*) is generated using feedback from the position sensor/ Hall effect sensors. The reference currents have the shape of a quasi-square wave in phase with respective back emfs to develop constant unidirectional torque. The hysteresis current controller regulates the winding currents (i_a, i_b, i_c) within the small band around the reference currents (i_a*,i_b*,i_c*). The motor switching commands so generated drives the inverter connected to the PMBLDC drive.

3.2.1 Mathematical model of PMBLDCM

The PMBLDCM produces a trapezoidal back electromotive force (EMF), and the applied current waveform is quasi-square shaped. A 3-phase, 6-state, Y-connected PMBLDCM with two-phase excitation is considered. We make the following assumptions within the allowable extent:

- The three phase winding are symmetrical.
- Magnetic saturation is neglected.
- Hysteresis and eddy current losses are not considered.
- The inherent resistance of each of the motor winding is R, the self inductance is L and the mutual inductance is M.

The three-phase stator voltage equation can be expressed as following:

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L-M & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & L-M \end{bmatrix} p \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \quad (3.1)$$

where, v_a, v_b, v_c are the phase voltage of three-phase windings, i_a, i_b, i_c, are the phase current, e_a, e_b and e_c are the phase back EMF and p is differential operator, R resistance of each windings , L self-inductance, and M mutual inductance.

Based on the equation (3.1), the equivalent circuit of the motor is shown in the fig. 3.2. The torque produced by PMBLDCM is described by following torque -motion equation.

$$T = K_T I = J \acute{\omega} + B \omega + T_l \quad (3.2)$$

Where, I is motor current, K_T motor torque constant, T_l load torque, J rotational inertia of rotor and load, B viscous damping coefficient, ω angular velocity of motor which depend on the applied voltage V as given below.

$$\omega = K_v V \tag{3.3}$$

Here K_v is the motor voltage constant. Both K_T and K_v are important parameters for a PMBLDC are specified in the datasheet of the motor and there product is always same for all the motors.

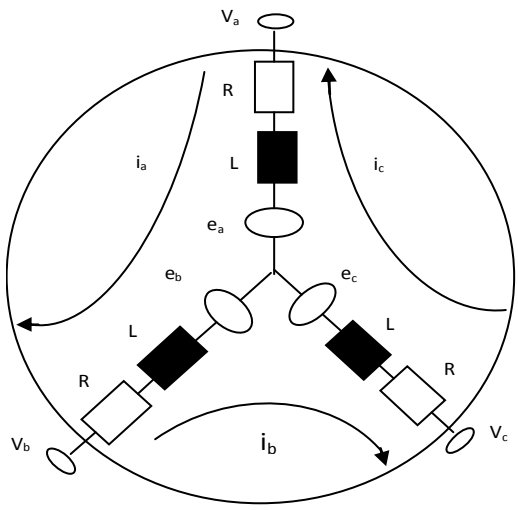

Figure 3.2 The equivalent circuit of the PMBLDC motor

3.2.2 Hysteresis current controller

The hysteresis current controller contributes to the generation of the switching signals for the inverter devices, which is based on current hysteresis control. The input to this controller are reference phase current (i_a*,i_b*,i_c*) and sensed current (i_a, i_b, i_c) and output signal will act as the control signal to the inverter.

When sensed current is greater than the reference current and the error is greater than the ring width of hysteresis comparator, the corresponding phase will conduct forward and turn off reversely. Conversely, it will be conducted reversely and turn off forward with proper selection of hysteresis ring width, the actual current will track the reference closely. The hysteresis band h_b chosen here in simulation is 0.1A.

The switching logic is formulated as given below:

If $i_a < (i_a^* - h_b)$ switch 1 ON and switch 4 OFF

If $i_a > (i_a^* + h_b)$ switch 1 OFF and switch 4 ON

If $i_b < (i_b^* - h_b)$ switch 3 ON and switch 6 OFF

If $i_b < (i_b^* + h_b)$ switch 3 OFF and switch 6 ON

If $i_c < (i_c^* - h_b)$ switch 5 ON and switch 2 OFF

If $i_c < (i_c^* + h_b)$ switch 5 OFF and switch 2 ON

3.2.3 Reference current block

The function of this block is to produce three phase reference currents. The magnitude of the three phase current (I_o) is determined by reference torque computed by hybrid controller and rotor position signal (φ).

$$I_o = T^* / K_b \quad (3.4)$$

where, T^* is the reference torque and K_b is the back emf constant. The output of this block are three phase reference currents (i_a^*, i_b^*, i_c^*). The output generated can have any of value from the set (I_o,-I_o, zero). Table 1 shows corresponding relation between input rotor position (φ) and reference currents output generated.

Input / Rotor Position (φ)	**Output / Generated Reference Currents**		
	$\mathbf{i_a^*}$	$\mathbf{i_b^*}$	$\mathbf{i_c^*}$
0 to π/3	I_o	$-I_o$	0
π/3 to 2π/3	I_o	0	$-I_o$
2π/3 to π	0	I_o	$-I_o$
π to 4 π/3	$-I_o$	I_o	0
4 π/3 to 5 π/3	$-I_o$	0	I_o
5 π/3 to 2 π	0	$-I_o$	I_o

Table 3.1 Reference current input and output logic

The reference currents are fed to the current hysteresis PWM current controller.

3.3 Speed controllers

Four types of controllers are dealt for the speed control of a PMBLD motor. The input to controller are speed error e(n) and change in speed error i.e. difference between present value of speed e(n)and past value e(n-1). The output is the reference current signal fed to the hysteresis current controller block which generates the gating pulses corresponding to the required current. The inverter supplies the required currents to the three phases of the machines.

The speed error e(n) at the n^{th} instant of time is given as:

$e(n)= \omega_r^*(n) - \omega_r(n)$

$\Delta e(n)= e(n) – e(n-1)$ (3.5)

where $\omega_r^*(n)$ is the reference speed at n^{th} instant, $\omega_r(n)$ is the rotor speed at the n^{th} instant and change in error at n^{th} instant.

3.3.1 Classical PI controller

The figure shows the general schematic block diagram of the PI controller the output of the controller in discrete domain at the nth instant is given as:

$T^*(n) = T^*(n - 1) + K_p\{ e(n) – e(n - 1)\}+ K_i e(n)$ (3.6)

Where K_p and K_i are the proportional and integral gain parameters of the PI speed controller.

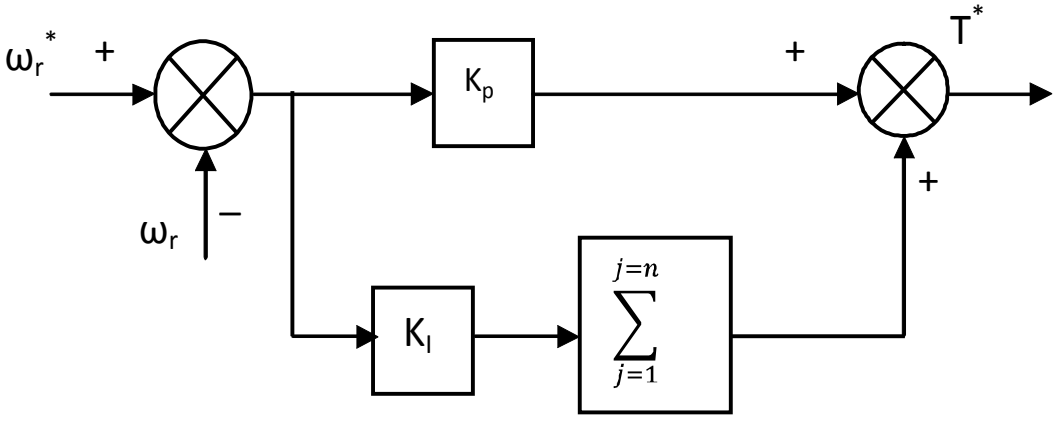

Figure 3.3 The Block Diagram of PI controller

The gain parameters are judicially selected by observing their effect on the response of the drive. The numerical values of the controller gains used in the simulation are given in the appendix.

3.3.2 Fuzzy logic controller

Fuzzy control provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system. Fuzzy controller design involves incorporating human expertise on how to control a system into a set of rules (a rule base). Generally the procedure for constructing a FLC consists of the following mechanism.

A. **Choosing the fuzzy controller inputs and outputs**: The speed error 'e' and the change in speed error 'ce' are selected as the input variables. Asymmetrical triangular type of the membership functions has been chosen to fast converse the error towards the desired steady state condition. The expected output from the controller is reference torque. The symbol GE is used for the scaling constant for the input e(n), and the symbol GCE, for the scaling constant for the input Δe(n).

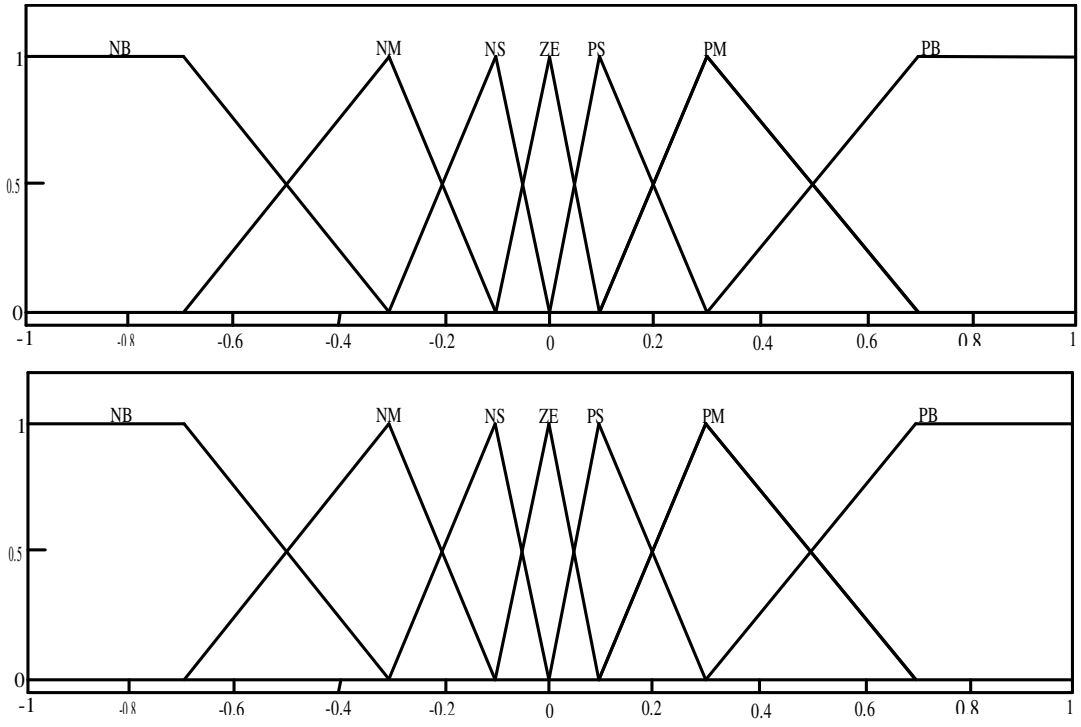

Figure 3.4 Membership functions for both the inputs for fuzzy logic controller

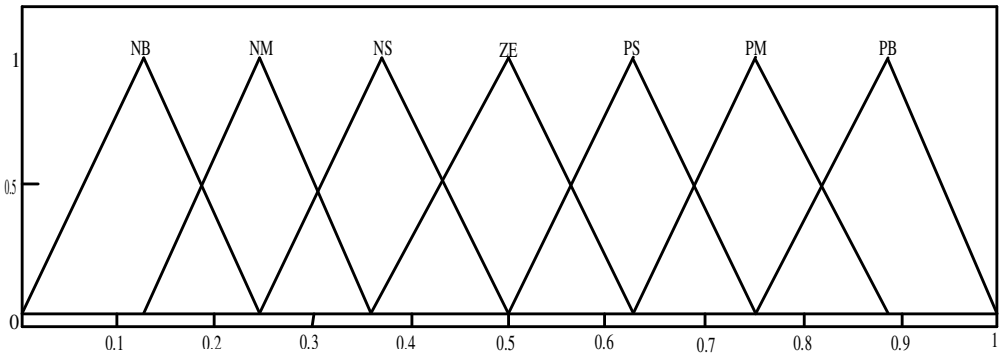

Figure 3.5 Membership function for output of fuzzy logic controller

B. **Putting control knowledge into rule base**: "Linguistic variables" that describe each of the time varying fuzzy controller inputs and outputs is defined. Each input and the output variables are described using the variables {NB, NM, NS, ZE, PS, PM, PB}. Proper control rules are written using the variables in the "if-then" format and rule base is prepared. The fuzzy controller rules are given in table 3.2.

E / CE	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NB	NB	NM	PM
NM	NB	NB	NB	NM	NM	PB	PB
NS	NB	NB	NM	NS	ZE	PM	PB
ZE	NB	NB	NB	ZE	PS	PB	PB
PS	NB	NM	ZE	PS	PM	PB	PB
PM	NB	NS	PM	PM	PB	PB	PB
PB	NM	PM	PB	PB	PB	PB	PB

Table 3.2 Rule base of fuzzy logic controller

C. **Inference Mechanism**: It leads to determination of conclusions. This emulates the expert's decision making in interpreting and applying knowledge. The rules are read as "if error is 'NB' and change in error is 'PB' then the reference torque is 'ZE'". Where the part after if called antecedent i.e. 'NB' and change in error is 'PB' and part after then is called consequent i.e. reference torque is 'ZE'. The conjunction of the rule antecedent is evaluated by the fuzzy operation intersection, which is implemented by the *min* operator. The rule strength presents the degree of membership of the output variables for a particular rule. Defining the rule strength of a particular rule as

$$\xi_{i,j} = min\ (\mu_{Fi}\ ,\ \mu_{Fj}) \tag{3.7}$$

where $i \in$ [NB, NM, NS, ZE, PS, PM, PB] is associated with the fuzzy variable e(n) and $j \in$ [NB, NM, NS, ZE, PS, PM, PB] is associated with the fuzzy variable Δe(n). The fuzzy inference engine uses the appropriate designed knowledge base to evaluate the fuzzy rules and produce the output of each rule. Figure 3.6 show the control surface generated after the inference mechanism processing on the rule base.

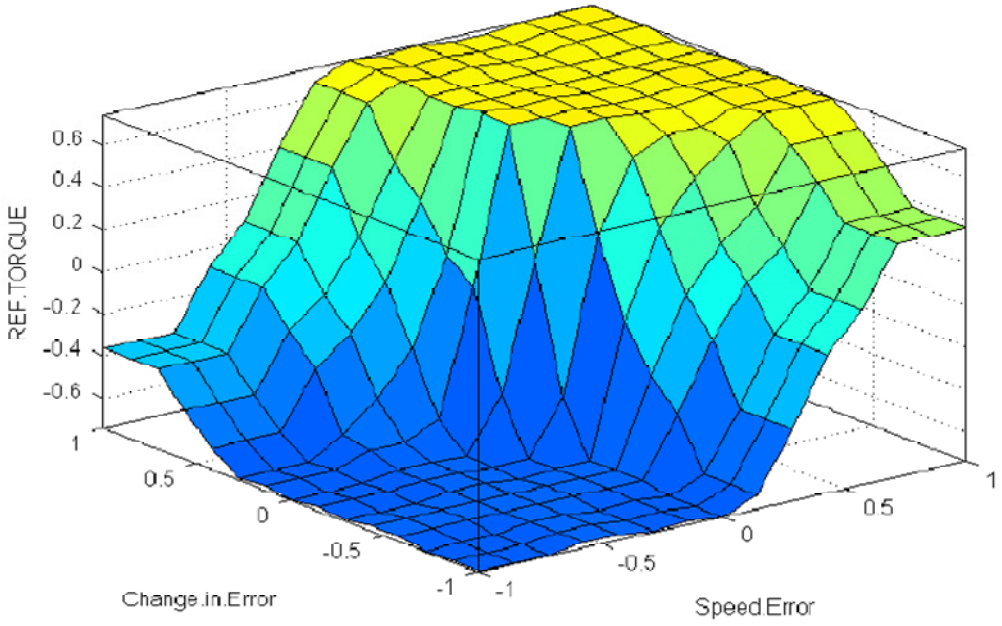

Figure 3.6 Control surface of fuzzy logic controller

D. **Converting decisions into actions**: De-fuzzification is the final component of the fuzzy controller, operates on the implied fuzzy sets produced by the inference mechanism and combines their effects to provide the "most certain" controller output. The centriod method is used for de-fuzzification. Centroid method that is used for defuzzifcation, returns the centre of area. The output is obtained using following eq. in general.

$$\Delta S = \frac{\sum_{k=1}^{N} \mu\,(dS_K).dS_K}{\sum_{k=1}^{N} \mu\,(dS_K)} \tag{3.8}$$

where 'ΔS' is the output of fuzzy logic, 'dS_K' is discrete value of 'dS' , 'dS' is the input to fuuzy and 'μ (dS_K)' is the degree of membership function associated with each 'dS_K' belonging to the active region.

To summarize the working of fuzzy functioning, the necessary inputs are applied to relative blocks by knowledge base, possessing the rule based and the data base as its sub-blocks. The fuzzifier converts crisp data into linguistic format. The decision maker decides in linguistic format with the help of logical linguistic rules supplied by the rule base and the relevant data supplied by the data base. The decision making block uses the rules in the format of "If-Then". The rule is to be read as if error is 'NB' and change in error is 'PB' then the reference torque is 'ZE'. It is defined by understanding the behavior of the system, such that rise time is low and the required torque is catered. The output of the decision-maker passes through the defuzzifier where in the linguistic format signal is converted back into the numeric form or crisp form. The denormalization is done (if required) to put the values back in physical domain.

Fuzzy logic controllers have three significant advantages over conventional techniques- they are cheaper to develop, they cover a wide range of operating conditions (i.e. are more robust), and they are more readily customizable in natural language. Any change is easier to do, as the respective rule have to be change in rule base, where in the conventional type frequent tuning is needed. The limitation of FLC large computation time the control loop time exceeds beyond sampling rate making it unsuitable for desirable PWM frequency.

3.3.3 The Hybrid controllers

The performance of the conventional controller suffers in the presence of unknown nonlinearities and parameter variations. But these controllers are still very popular in industries as they are simple in design and low cost. The nonconventional controller such as fuzzy logic controller shows near to ideal dynamic performance without oscillations, but lack in fast response and there no systematic methodology for their design. A control action which do not depend on individual controller and configured in such a way, that it can exploit benefit of both the types of controller conventional and nonconventional. Based on present idea the hybrid controllers are introduced.

3.3.3.1 The series hybrid controller

The fig. 3.7 illustrates the basic control structure of the series hybrid PI controller or fuzzy precompensated PI controller. The scheme consists of a conventional PI control structure together with the proposed fuzzy precompensator.

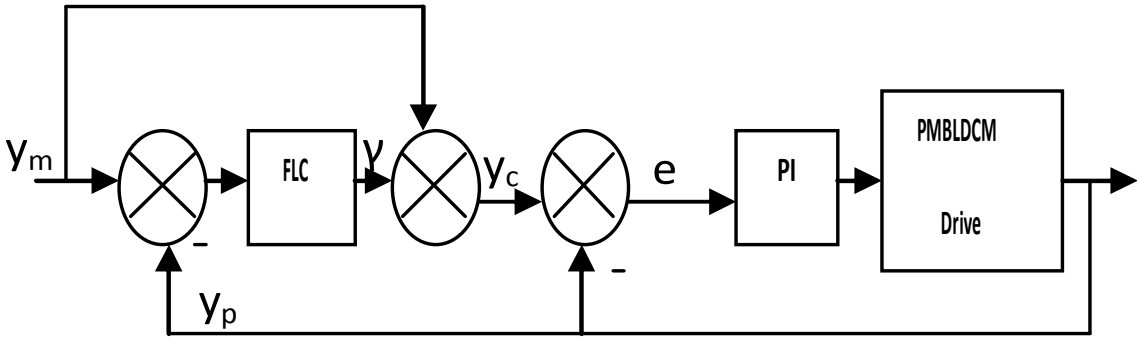

Figure 3.7 Block diagram series configuration of hybrid controller

A. Control operation of fuzzy precompensator

The purpose of the fuzzy precompensator is to modify the command signal to compensate for the overshoots and undershoots present in the output response when the plant has unknown nonlinearities, which can result in significant overshoots and undershoots in the response if a conventional PI control scheme is used. The precompensator uses fuzzy logic rules that are based on the above motivation.

The fuzzy precompensator uses the command speed input y_m and the plant speed output y_p to generate a precompensated command signal y_c, described by the following equations

$$e(n) = y_m(n) + y_p(n)$$

$$\Delta e(n) = e(n) - e(n - 1)$$

$$\gamma(n) = f\,[e(n),\Delta e(n -1)]$$

$$y_c(n) = y_m(n) + \gamma(n) \qquad (3.9)$$

In the above e(n) is the tracking error between the command speed input $y_m(n)$ and the plant speed output $y_p(n)$ and $\Delta e(n)$ is the change in the tracking error. The term $f[e(n),\Delta e(n -1)]$ is a nonlinear mapping of e(n) and $\Delta e(n)$ based on fuzzy logic. The term $\gamma(n) = f[e(n),\Delta e(n -1)]$ represents a compensation or correction term, so that the compensated command signal $y_c(n)$ is simply the sum of the external command signal $y_m(n)$ and $\gamma(n)$.The correction term is based on the error e(n) and the change of error $\Delta e(n)$.

The compensated command $y_c(n)$ is applied to a conventional PI scheme, as shown in Fig 5. The equations governing the PI controller are as follows

$$e_2(n) = y_c(n) + y_p(n)$$

$$\Delta e_2(n) = e_2(n) - e_2(n-1)$$

$$u(n) = u\,(n-1) + K_p\,\Delta e_2(n) + K_I\,e_2(n) \qquad (3.10)$$

The quantity $e_2(n)$ is the precompensated tracking error between the precompensated command input $y_c(n)$ and the plant output $y_p(n)$ and $\Delta e_2(n)$ is the change in the precompensated tracking error. The control u(n) is applied to the input of the plant.

B. Fuzzy control action as precompensator

The action of the fuzzy is to pre-compensate the reference input viewed by PI controller. If the system is error signal e(n) is negative and change is error Δ e(n) is positive then system is in oscillation towards an undershoot. The precompensator would increase the reference input value observed by the PI so that the system will

not move towards undershoot. Following are the steps of design a fuzzy precompensator:

a. **Choosing the fuzzy controller inputs and outputs:** the fuzzy logic based term $\gamma(n)=F[e(n),\Delta e(n)]$. The e(n) and $\Delta e(n)$ are the two inputs and $\gamma(n)$ is the output from the FLC. Each input and output is defined with seven linguistic variables such as NB, NM, NS, ZE, PS, PM, PB by non-symmetrical triangular membership function. The placement of these linguistic variables on the universe of discourse is same as shown in figure 3.4 and 3.5.

b. **Putting control knowledge into rule base:** each input is defined with seven variables each so the rule base consists of 49 rules. The table 3.3 shows the rule base of fuzzy precompensator in series hybrid configuration.

E / CE	**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
NB	NB	NB	NB	NB	NM	NS	PM
NM	NB	NB	NB	NM	NM	PS	PB
NS	NB	NB	NM	NS	NS	PM	PB
ZE	NB	NM	NM	ZE	PS	PB	PB
PS	NM	NM	NL	PS	PM	PB	PB
PM	NM	NL	PS	PM	PB	PB	PB
PB	NM	PS	PB	PB	PB	PB	PB

Table 3.3 Rule Base of fuzzy precompensator

c. **Inference Mechanism:** rules are often written in the form (ZE, NS, NM) the idea of the rule is "if e(n) is 'ZE' and $\Delta e(n)$ is 'NS' then output is 'NM' ". When the

error is zero and change in error is negative then the system is moving in the direction of overshoot so the reference speed is modified to decrease. Figure 3.8 shows the control surface for the fuzzy precompensator.

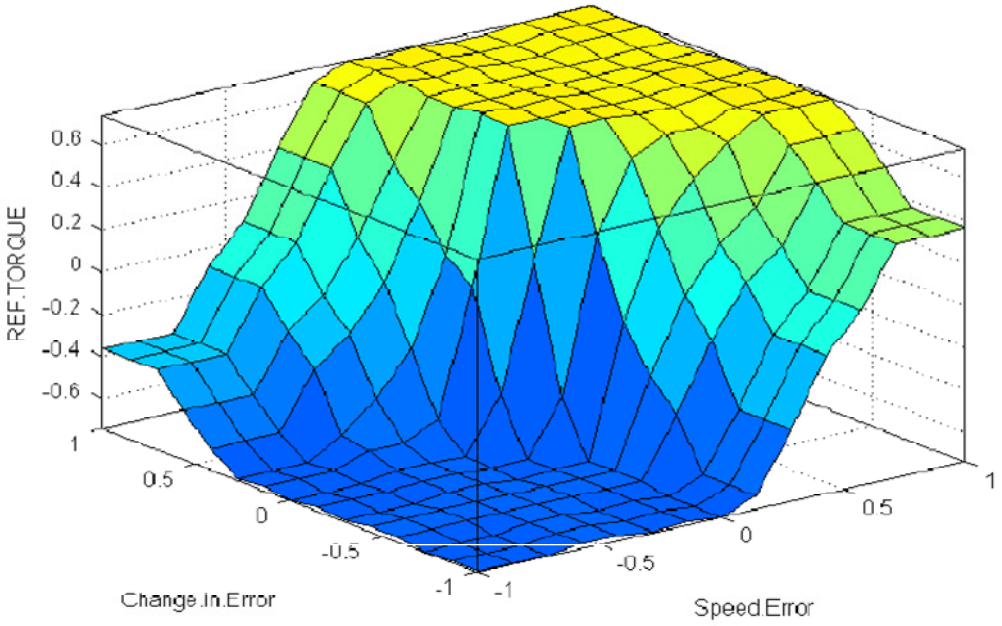

Figure 3.8 Control surface of fuzzy precompensator

d. Converting decisions into actions: The centre of area or centriod method is used for the defuzzification. Further a scaling factor to is used to get the final control signal.

3.3.3.2 The parallel hybrid controller

The structure is easy to understand and is capable of getting accommodated without much change in the hardware system. In stream controller continues to work as a PI controller, but unlike the conventional PI controller the values of the proportional and the integral gains are modified continuously based upon the operating condition. Intelligence inherited from FLC may be translated into K_p and K_i gains, which may be altered as per the situation before the drive.

In plant where the frequent system parameter variation and load perturbations are common, adaptive nature of control is highly desirable. Controllers capable of self tuning its parameters based on the current situation are best suited.

A. Control operation of fuzzy

In parallel hybrid configuration the FLC in parallel, configuring the value of proportional gain 'K_p' and Integral gain 'K_I' of the inline PI, the block diagram of the same is depicted in fig.3.9.

$$u(t) = K_p\, e(t) + K_I \int e(t)\, dt \qquad (3.11)$$

The control equation of a conventional PI controller in continuous time domain is represented as per eq. 3.11. The proportional gain provides the control action effectively when the error is more (transient response) and the integral gain delivers efficiently when the system is operating near the set point value. FLC intelligence assigns the proportional gain (K_p) which must be maximum when the error is large and should start to vary to a value when the drive system is near to the set point. Similarly FLC assigns the integral gain continuously, when the drive operates away from the set point the value is kept low and it attains a high value when it operates near to the set point. This means the proportional gain provide the control action effectively when the error is more and the integral gain delivers effectively when the system is operating near the set point value.

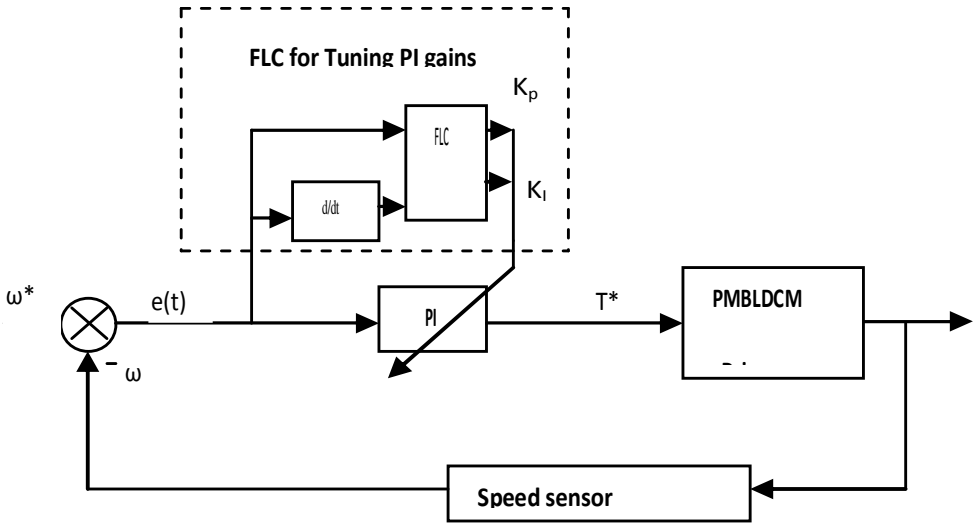

Figure 3.9 Block diagram of parallel hybrid controller

B. Fuzzy control action in parallel hybrid

The action of the fuzzy is to tune the two gain values i.e. proportional and integral gain. Following are the steps for designing the fuzzy control to tune the gains values:

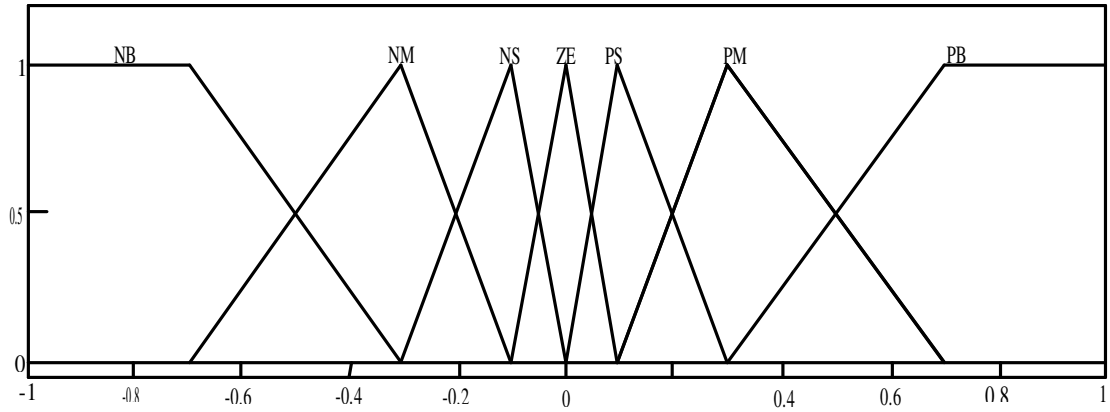

Figure 3.10 Membership function for error e(n) input for parallel hybrid

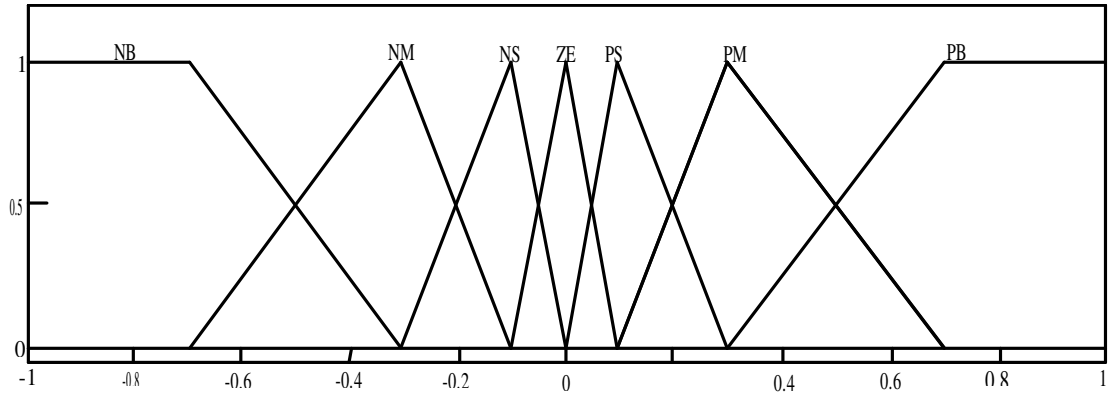

Figure 3.11 Membership function for change in error Δe(n) input for parallel hybrid

a. Choosing the fuzzy controller inputs and outputs: the two inputs error and change in error are defined using seven linguistic variables i.e. NB, NM, NS, ZE, PS, PM, PB. The two outputs are proportional gain and integral gain, are defined using seven linguistic variables i.e. VL, LO, BM, ME, AM, HI, VH. Figure 3.10 and 3.11 shows the membership function of the inputs. While figure 3.12 and 3.13 shows the membership functions of the two outputs of parallel hybrid controllers in the universe of discourse.

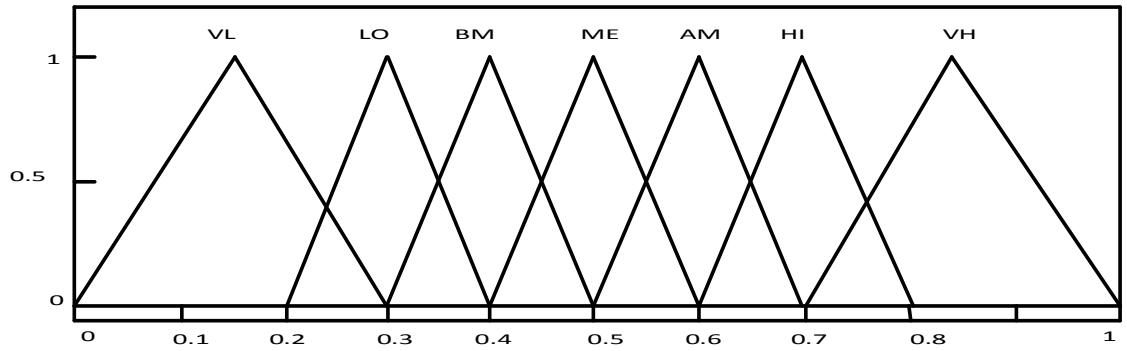

Figure 3.12 Membership function of Proportional gain (K_p)

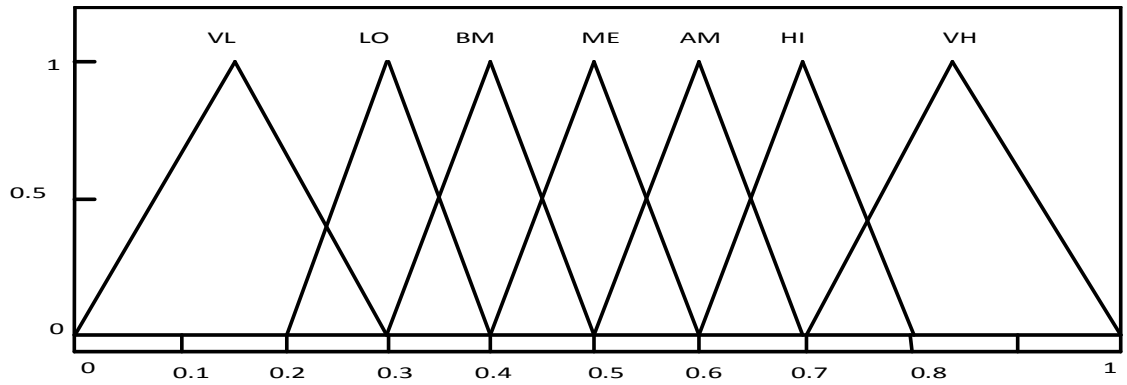

Figure 3.13 Membership function of Integral gain (K_I)

b. Putting control knowledge into rule base: the behavior of the system in the presence of disturbances and non linearities shows oscillations, so the value are gains parameter are adjust to compensate it. The rule base is consists of 49 rules sets of rules for each output. and read is "if the error e(n) is 'PB' and change in error is Δe(n) is NB then Proportional gain (K_p) is VH and integral gain (K_I) is VL". The table 3.4 and 3.5 shows the rule base of parallel hybrid controller for proportional gain (K_p) and integral gain (K_I).

c. Inference Mechanism: The rules are designed by the combination of experience, trial and error and the knowledge if the system behavior. Considering the rule (ZE, ZE, VL, VH) which says if the error e(n) is 'ZE' and change in error is Δe(n) is ZE then proportional gain (K_p) is VL and integral gain (K_I) is VH". The integral gain is kept high because it aids in reducing the steady state error, while the proportional gain (K_p) is kept low as the set point has been achieved. The figure 3.14 shows the

control surface of the proportional gain (K_p) for the parallel hybrid controller and figure 3.15 shows the control surface of the integral gain (K_I)

E / CE	NB	NM	NS	ZE	PS	PM	PB
NB	VH	VH	HI	ME	ME	AM	VH
NM	VH	VH	AM	BM	ME	HI	VH
NS	VH	VH	AM	LO	ME	HI	VH
ZE	VH	VH	AM	VL	AM	VH	VH
PS	VH	HI	ME	LO	AM	VH	VH
PM	VH	HI	ME	BM	AM	VH	VH
PB	VH	AM	ME	ME	HI	VH	VH

Table 3.4 Rule base of proportional gain (K_p) for parallel hybrid controller

E / CE	NB	NM	NS	ZE	PS	PM	PB
NB	VL	VL	LO	AM	AM	BM	VL
NM	VL	VL	BM	AM	ME	LO	VL
NS	VL	VL	BM	HI	ME	LO	VL
ZE	VL	VL	BM	VH	BM	VL	VL
PS	VL	LO	ME	HI	BM	VL	VL
PM	VL	LO	ME	AM	BM	VL	VL
PB	VL	BM	AM	AM	BM	VL	VL

Table 3.5 Rule base of integral gain (K_I) for parallel hybrid controller

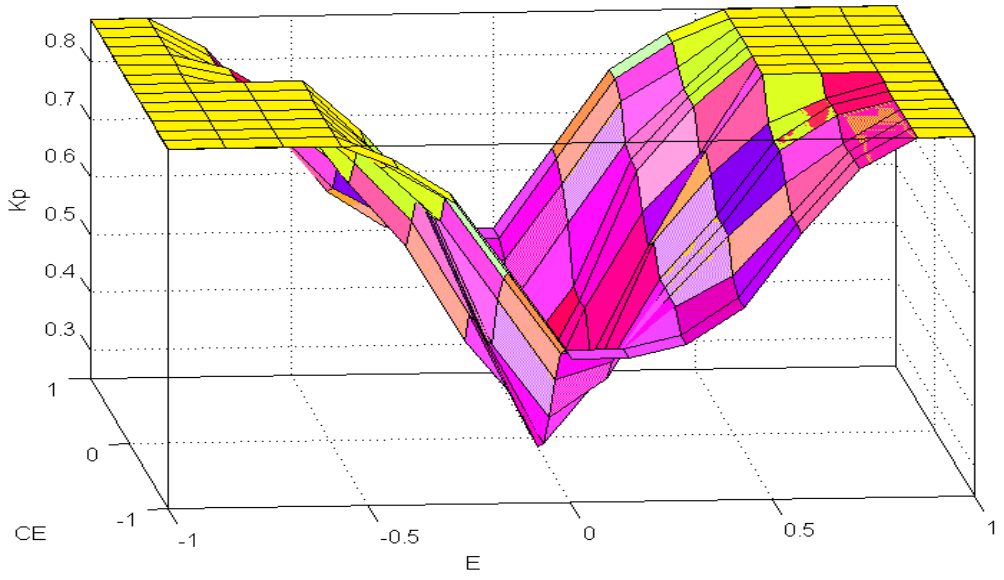

Figure 3.14 Control surface for proportional gain (K_P) for parallel hybrid

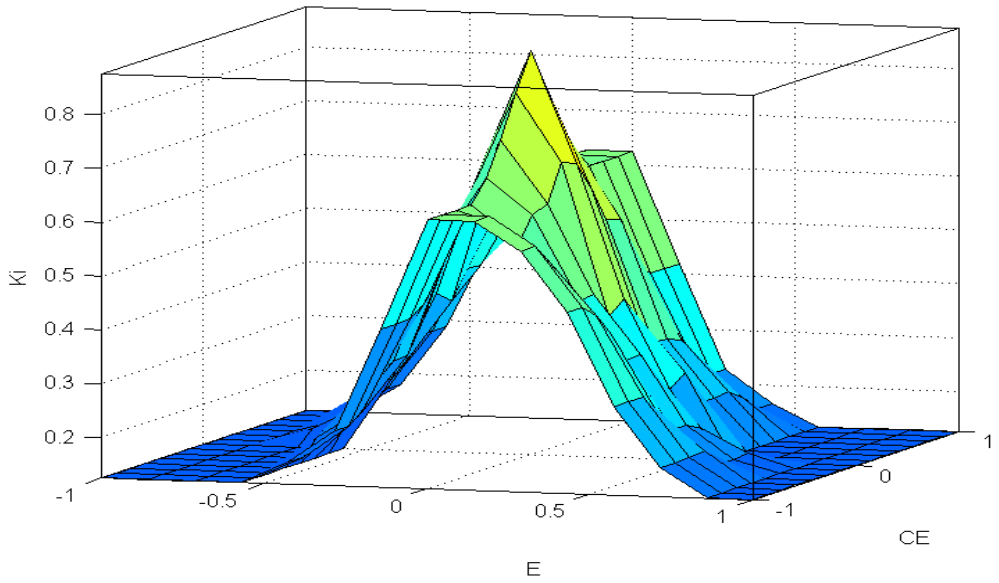

Figure 3.15 Control surface for Integral gain (K_I) for parallel hybrid

d. Converting decisions into actions: The defuzzification process maps the results of the fuzzy rule stage to a real number output. Here, we are using the centriod or centre of area defuzzification method.

3.3.4 Reduced rule base Hybrid controllers

To realize the hybrid controllers the complexity of the FLC and rule base need to be reduced. Application to drives where frequency of operation is high to reduce the torque supplied, processing time taken by fuzzy inference cannot meet the loop time requirements. Hardware implementation of FLC on digital signal processor, microcontroller or FPGA requires large memory space and high computational effort. Often for real time application low cost devices, with maximum utilization of resources is desirable. Hence the proposed approach, focus on reduction the computation burden and memory requirement have been done by reducing the rule base. It is further ascertained that switching of any kind for controller to be avoided, which leads to disturbances in the system performance.

Both hybrid PI controller configurations are proposed in the following subsection, where, the selection of the configuration is based on the application requirements.

3.3.4.1 Reduced rule base Series hybrid PI controller

The control techniques remains the same, but the fuzzy precompensator design differ from the series hybrid in respect of the rule base. This is achieved by using fuzzy control for limited range of operation. Fuzzy controller output is zero for the large range of operation; it gives non-zero output for limited range i.e. near to the set point. Implementation of this logic has helped in avoiding the switching that is generally present in hybrid configuration. The fuzzy controller is designed is such a way that mathematical calculations are minimized. Rule base of fuzzy has been reduced with tremendous effort to get desired low level of complexity in the term of computation effort and memory requirements. Following steps are taken in designing a fuzzy logic controller to have a reduced rule base:

A. **Choosing the fuzzy controller inputs and outputs**: For the proposed reduction of rule base the common universe of discourse [-1, 1] is taken for all the variables. Triangular membership functions are utilized for defining both input and output. Fig. 3.16 shows the input and fig. 3.17 output membership function. The limited range of operation of FLC helps in minimizing the fuzzy

variables, as we are operation near set point, so fuzzy variables are defined only near the set point not for the complete universe of discourse. The placement of fuzzy set on the universe of discourse is done based on experience. The linguistic variables used are “NS” is negative small; “NM” is

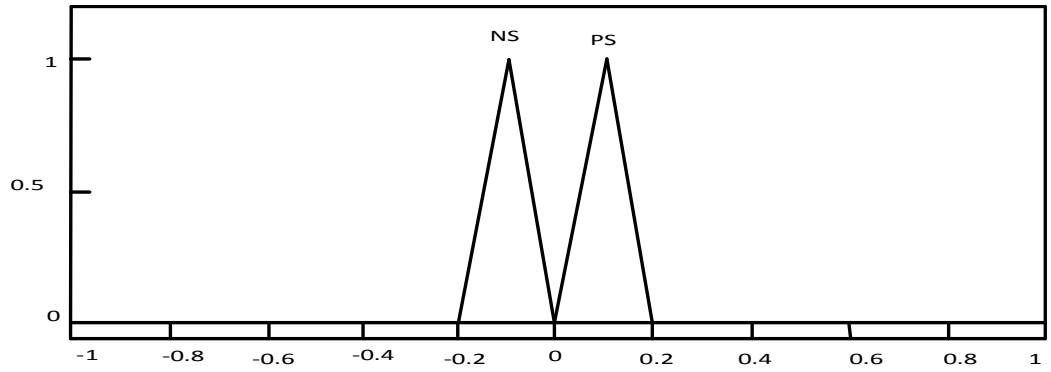

Figure 3.16 Membership function for inputs of FLC precompensator with reduced rule base

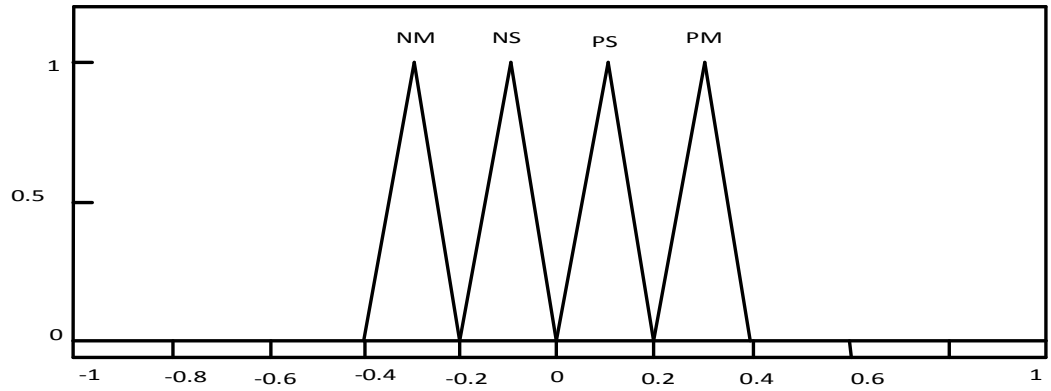

Figure 3.17 Membership function for output of FLC precompensator with reduced rule base

negative medium; “PS” is positive small; “P” is positive medium. Both inputs are defined using only two variables i.e. NS and PS. As both the inputs have two variables each, the rule base consists of only four rules.

B. **Putting control knowledge into rule base**: In series hybrid configuration the reference speed observed by the inline PI is modified by Fuzzy control. The FLC outputs is zero for the large range of operation and does not amend the speed response whereas, it furbish non zero output, when the error and change in error both are fast

approaching to zero. As shown in table 3.6 rule base for series configuration control, which clearly indicates how reference speed experienced by PI, which is modified by fuzzy precompensator to avoid the overshoot and undershoot during load perturbation, non linearities etc.

Since only two fuzzy sets NS and PS for each input are used, the rule base consist of only four fuzzy rules, indicating tremendous saving in term of computation effort. As the output has only four sets NM, NS, PS and PM, the fuzzy rule is prepare based on knowledge in IF-THEN form as follow:

E / **CE**	**NS**	**PS**
NS	NM	NS
PS	PS	PM

Table 3.6 Rule table of fuzzy precompensator for reduced rule base series hybrid controller

C. **Inference Mechanism**: It is order to find out the conclusion, decision making process has to done. For example of a rule "if e(n) is 'NS' and $\Delta e(n)$ is 'PS', then γ is 'NS' also represented as triplet (NS, PS, NS) . Here we think γ as output of the fuzzy logic rules. The rules are derived using a combination of experience, trial and error, and our knowledge of the response of the system. Suppose the command signal is a constant y_m the error e(n) is positive small and the change in error is $\Delta e(n)$ is positive small . This means that the output $y_p(n)=y_m(n)-e(n)$ is decreasing , hence it is in the middle of an undershoot. To compensate it we need to increase the command signal by a positive amount. This correspond to applying positive value to y(k). Hence the rule is obtained "if e(n) is 'PS' and $\Delta e(n)$ is 'PS', then γ is 'PM' ". The fuzzy control surface obtained, after inference from rules shown in figure 3.18.

D. **Converting decisions into actions**: The defuzzification process maps the results of the fuzzy logic rules stage to a real number output $f[e(n),\Delta e(n)]$. We use the centroid

defuzzification method. The output of the fuzzy logic controller is multiplied by a scaling factor to get the final control signal $y_p(n)$.

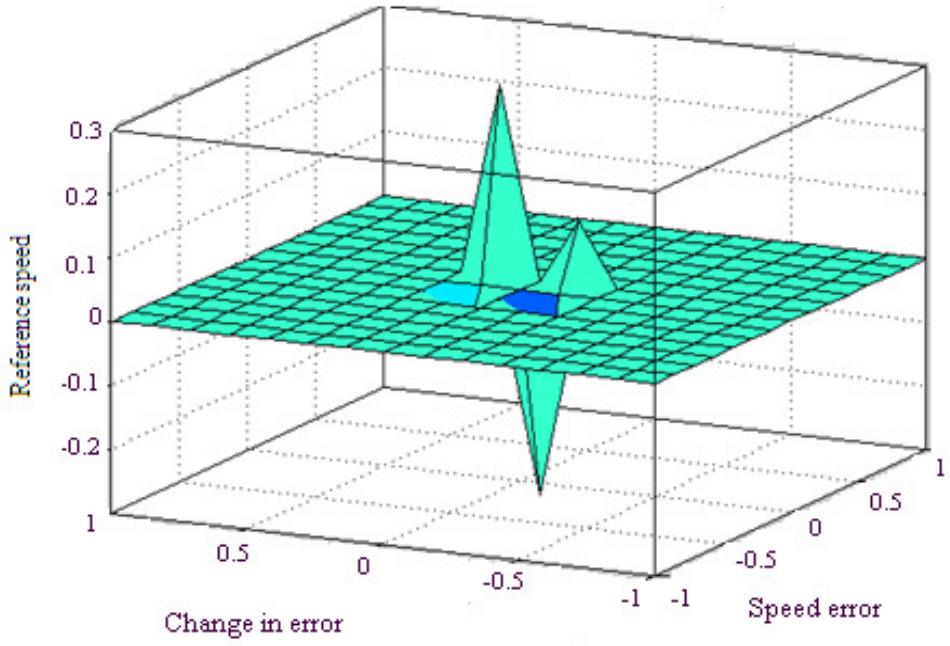

Figure 3.18 Control surface of fuzzy precompensator with reduced rule base

The precompensated reference speed output of FLC is input to the inline PI controller and amends its performance dynamically. The rules are written in a feedforward loop. If speed error is negative small and change in speed error is negative small then the output is heading in the direction of overshoot, the FLC show a reduced reference speed to PI controller. FLC precompensated reference speed input to inline PI, help in reducing the overshoot, undershoot and settling time dynamically faster, improving the overall performance of system. This configuration of conventional and fuzzy controller is called hybrid series PI controller. In case of implementation of series hybrid in existing hardware only few modifications are needed rather than full replacement of hardware.

3.3.4.2 Reduced rule base Parallel hybrid PI controller

The control structure of the controller is same as parallel hybrid controller, in which tuning of the two gains of PI controller is done. The fuzzy controller has two outputs for this configuration, the inference mechanism need to do simultaneous computation for both the outputs. Thus, increasing the memory and computational need, hence serious efforts has to be putted in order to decrease them. The idea of operating fuzzy in limited range of operation has helped in minimization of control codes. Fuzzy controller has non-zero output for limited operational range and for rest it has zero output. The controller designed on this logic has small rule base, as number of variables defined are small because of limited range in operation. Following are the steps taken for implementing the idea for fuzzy control design:

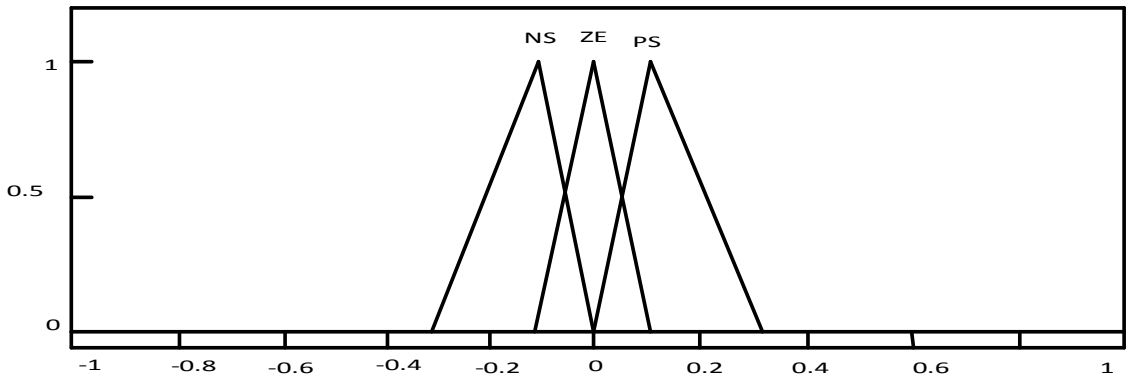

Figure 3.19 Membership function for error e(n) input

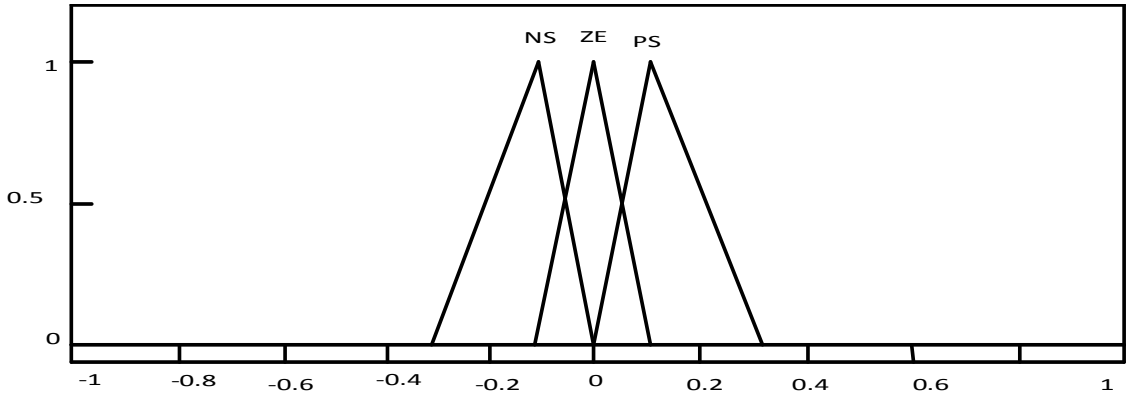

Figure 3.20 Membership function for change in error Δe(n) input

A. **Choosing the fuzzy controller inputs and outputs**: The two input of the FLC are speed error and derivative of speed error, and the two gains K_p, K_I are of PI controller, which represents its outputs. The universe of discourse for input is [-1 1] after fuzzification of inputs. As the negative values are not required for outputs the universe of discourse for output is reduced to [0 1]. For inputs only three fuzzy sets are used. The linguistic variables used for inputs are "NS" is negative small; "ZE" is zero; "PS" is positive small. On the other hand for outputs we are using seven fuzzy sets are used. The linguistic variables used for outputs are "VL" is very low; "LO" is low; "ZE" is zero; "SM" is small medium; "ME" is medium; "HM" is high medium; "HI" is high; "VH" is very high. For all type of variables triangular membership function are used. Fig 3.20, 3.21 shows the input membership function and fig. 3.12, 3.13 shows the output membership function.

E / **CE**	**NS**	**ZE**	**PS**
NS	AM	SM	SM
ZE	VL	VL	VL
PS	SM	LO	ME

Table 3.7 Rule table of Proportional gain (K_p)

E / **CE**	**NS**	**ZE**	**PS**
NS	HI	VH	HI
ZE	VH	VH	VH
PS	HI	VH	HI

Table 3.8 Rule table of integral gain (K_I)

B. Putting control knowledge into rule base: As the number of fuzzy sets for each input is three, the rule base is reduced to nine fuzzy rules. Table 3.7 shows the rule base for integral gain and table 3.8 shows rule base for proportional gain in parallel configuration. The rules are designed by the combination of the experience, trial and error and our knowledge of the system behavior. Consider the rule given (ZE, ZE, VL, VH), which says "if e(n) is ZE and Δe(n) is ZE this shows that system is operating at the set point so a proportional gain (K_P) must be very low and integral gain (K_P) must be very high" because, the integral gain is responsible to maintain zero steady state error i.e. stiffness at set point.

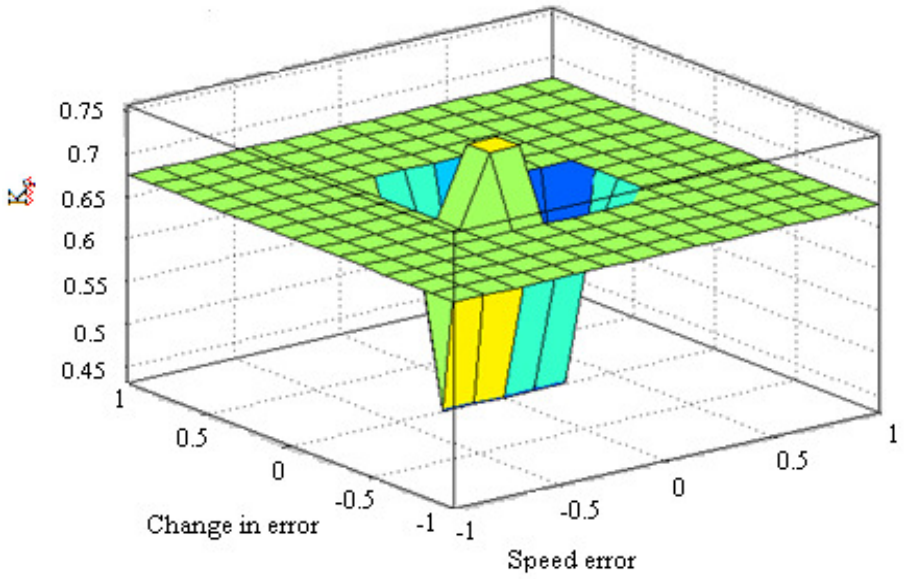

Figure 3.21 Control surface for proportional gain (K_P) for reduced rule base parallel hybrid controller

C. Inference Mechanism: The rule structure for output from fuzzy controller is in the form of (NS, ZE, VL, VH) which implies that "if the error e(n) is NS and the change is error Δe(n) is "ZE" then proportional gain (K_p) is "VL" and integral gain is (K_I) is "VH". Near the set point the integral value is kept high as it is responsible to maintain zero steady state error i.e. the stiffness at the set point. Fig. 3.21 shows the control surface for proportional gain and fig. 3.22 shows the control surface for integral gain in the proposed parallel configuration.

D. **Converting decisions into actions**: Centre of area method also called as centriod method is used for defuzzification. The defuzzification process the result of the fuzzy logic rule stage to a real number output. The output are multiplies by scaling factor to get the final gain values.

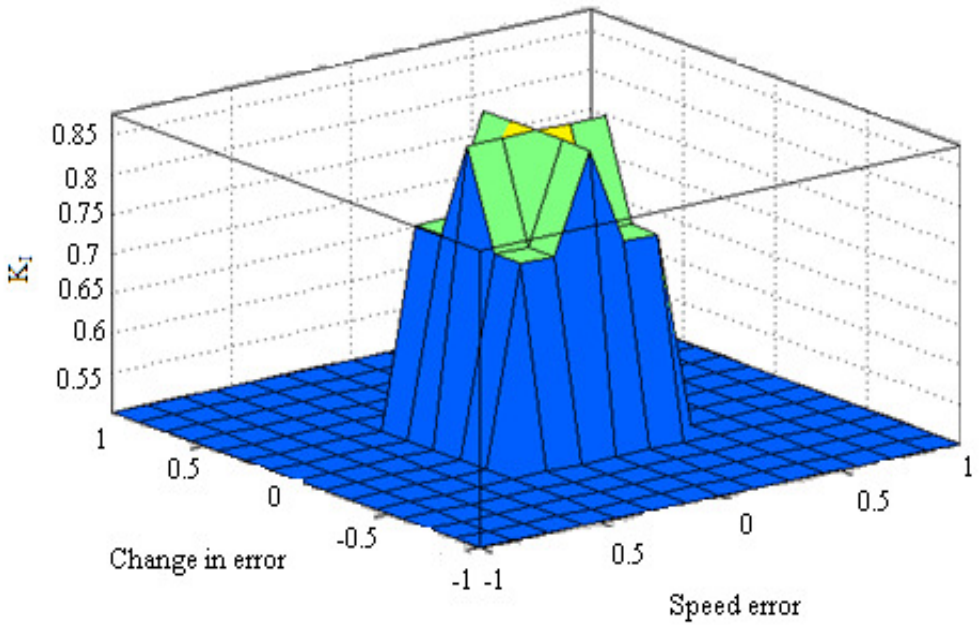

Figure 3.22 Control surface for Integral gain (K_I) for reduced rule base parallel hybrid controller

The output from FLC have non zero value for certain range of operation, for rest of the time the FLC retains the originally defined values of K_P and K_I of the inline PI controller. The necessity of switching between FLC and PI is thus avoided which generally reported in controllers. FLC furbishes the output when oscillations are present. It is observed the proportional gain has its maximum values when error is high and decrease as the error decreases and finally reaches the minimum point near the set point. The control strategy is employed in such a way that, in the starting when the error is large the proportional gain supersedes. Large proportional gain helps fast approaching to set speed. When error is near to zero the integral gain supersedes the control to suppress the oscillations.

3.4 Modeling using Simulink

In order to perform real simulation of the drive system, the control structure is developed in MTALAB environment using SIMULINK. The simulations of the main parts of the blocks diagram have been discussed in this section.

3.4.1 Simulink model of the PMBLDC drive

The motor block is directly taken from the "SimPowerSystems toolbox" given in SIMULINK library. The "Permanent magnet synchronous machine" block is taken and the trapezoidal back emf mode has been selected to function as a PMBLDCM. The parameters of the required machine to be simulated have been entered into the block. The mechanical input is selected as positive torque to make the machine function as a motor, the remaining parameters such as the stator resistance, stator inductance, the flux induced by the magnets, moment of inertia, friction factor and the pairs of pole have been entered into the block as per the requirement. The machine is simulated for the specification

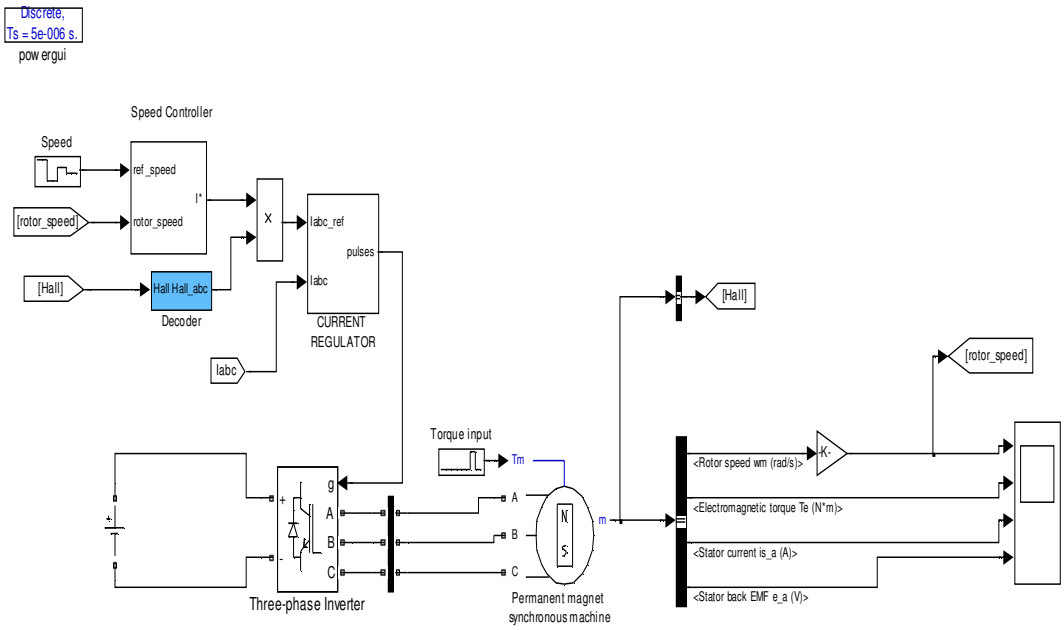

Figure 3.23 Simulink model of the PMBLDCM drive

parameters [19] given in the appendix. The complete Simulink model of the drive is shown in Fig. 3.23.

The inputs to the block are the three phase voltages and the currents from the inverter block. The speed in revolutions per minute (RPM), the torque developed (T_e) and the Hall effect signals from the sensors are taken as the outputs. Further various controllers used as speed controller for PMBLDCM drive are discussed.

3.4.2 Simulink model of Speed controllers

The model of speed controllers has been realized using the Simulink toolbox of the MATLAB software. The main function of the speed controller block is to provide a reference torque (T*) signal. The output of the speed controller block is limited to a proper value in accordance to the motor rating by using a saturation block. The speed controllers realized using the Simulink toolbox are namely, proportional integral (PI) speed controller, Fuzzy logic speed controller, Series hybrid PI (Fuzzy pre-compensated) controller ,Parallel hybrid PI (Self tuning) controller, Reduced rule base series hybrid PI controller and Reduced rule base parallel hybrid PI controller.

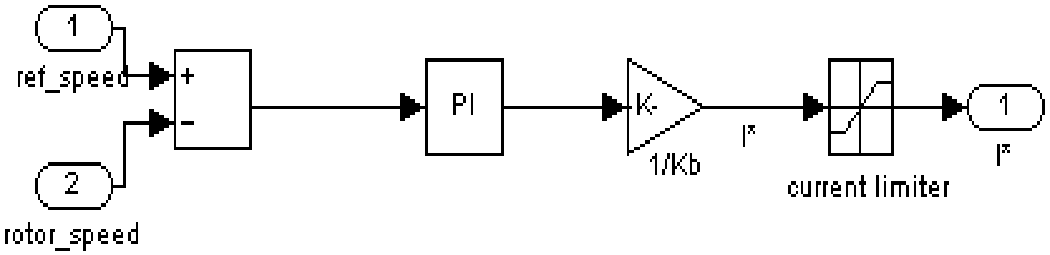

Figure 3.24 Simulink model for a PI controller

The Fig. 3.24 shows the MATLAB model block diagram for the PI controller. The basic operating equations have been stated in the previous sections. Using the proportional (K_p) and the Integral (K_I) gain parameters the reference torque signal (T^*) is generated by the PI controller, hence the desired motor speed is achieved.

Fig. 3.25 shows the MATLAB model diagram for the Fuzzy logic speed controller. The two inputs namely, speed error and change in speed error are properly

scaled and fed to the MATLAB fuzzy logic controller. The rescaled defuzzified output of the fuzzy logic block after limiting forms the output of the controller block.

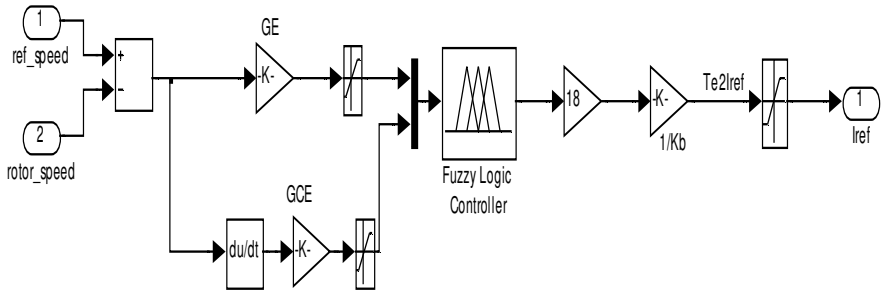

Figure 3.25 The Simulink block for the fuzzy logic controller

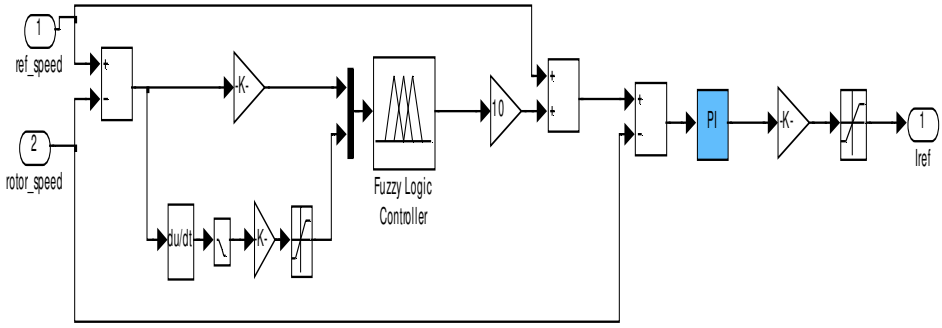

Figure 3.26 The Simulink block for the series hybrid PI controller and reduced rule base series hybrid PI controller

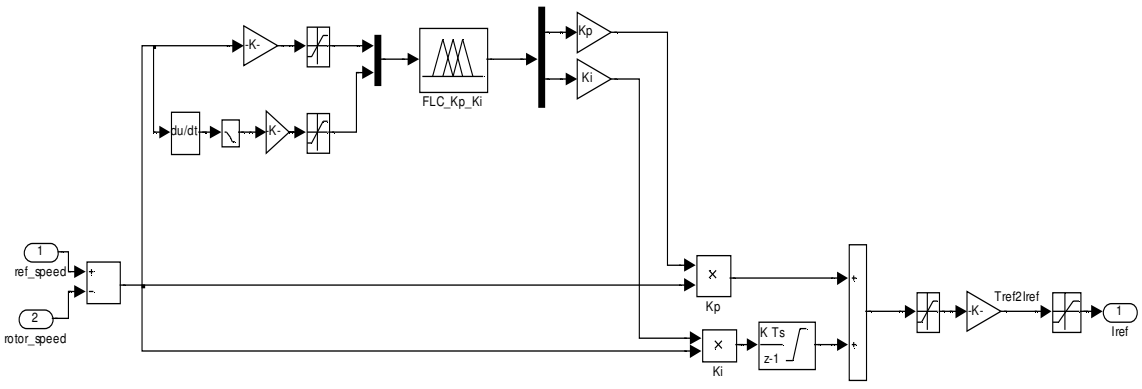

Figure 3.27 The Simulink block for the parallel hybrid PI controller and reduced rule base parallel hybrid PI controller

Fig. 3.26 below shows the MATLAB model diagram for the Series hybrid and reduced rule base series hybrid PI controller. Such a controller has the modified reference speed (precompensated) signal by the FLC to the PI controller. The PI controller produces the required control signal. The controller's operation has been discussed in the previous sections.

The Fig 3.27 shows the MATLAB model diagram for the parallel hybrid PI controller and reduced rule base parallel hybrid PI controller. In this controller, the error and the change in error are fed as the input to a Fuzzy logic controller, which generates the corresponding proportional and integral gain values depending on the fuzzy rules fed into FLC. These values are directly used by the PI speed controller to generate the required control Torque (T^*) signal.

3.5 Conclusion

The detailed modeling, analysis, design and simulation of the PI controller, the fuzzy logic controller, the series hybrid, the parallel hybrid, reduced rule base series hybrid PI controller and reduced rule base parallel hybrid PI controller have been described in this chapter. The fuzzy rules governing the performance were also given in detail. The fuzzy surfaces for the different controllers are also given in the respective sections, showing the behavior of individual controllers. The simulation results of these models are presented in the next chapter.

CHAPTER IV

RESULTS FROM SIMULATION AND DISCUSSION

4.1 General

In the following section, the simulation models developed in the previous chapter are simulated for a fixed/discrete sampling time of T =5 µsec. The drive performance is evaluated, separately using the six speed controllers presented in previous chapters, for different operating conditions. The results obtained are plotted to depict their effectiveness. Finally the results obtained from the different controllers are compared in terms of performance – overshoot, good rise time, less settling time and adaptive nature in loading conditions.

4.2 Response of the drive with a PI speed controller

The simulation model of the PMBLDC drive is simulated using the developed PI speed controller and the response is observed for different operating conditions such as the starting response, load perturbation and the speed direction reversal.

4.2.1 Response of the drive on starting and load perturbation

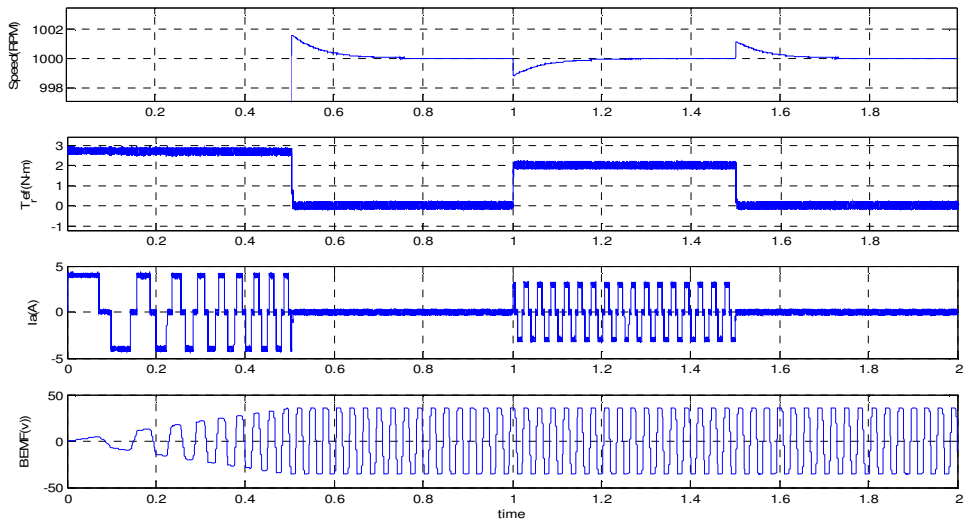

Figure 4.1 Response of the drive with the PI controller on starting and load perturbation

The Fig. 4.1 shows the response of the PMBLDC drive on starting at a set point speed of 1000 RPM with a PI speed controller. The developed model is simulated for t=2sec. At the time instant t=1sec, 2 Nm load torque is applied to the motor, and at t=1.5sec the load is removed. The ability of the drive to maintain the set point speed with the presence of load disturbance is mainly considered here. In figure rotor speed is presented in revolutions per minute (RPM), the electromagnetic torque (T_e) developed by the motor in (N-m), stator current (i_a) of phase a in Ampere, the back emf developed in phase a in (V). The a motor speed rises to the set point speed at 0.505sec; it has an overshoot of 1.55 RPM and finally settles at the set point at the time instant 0.8 sec. When load is applied at t=1sec, a dip in speed of 1.15RPM is observed, the set point speed is reached at 1.25sec. An overshoot of 1.15RPM is observed on the removal of load at t=1.5sec, the response settles at the set speed is reached at the time instant 1.75 sec.

4.2.2 Response of the drive during speed direction reversal

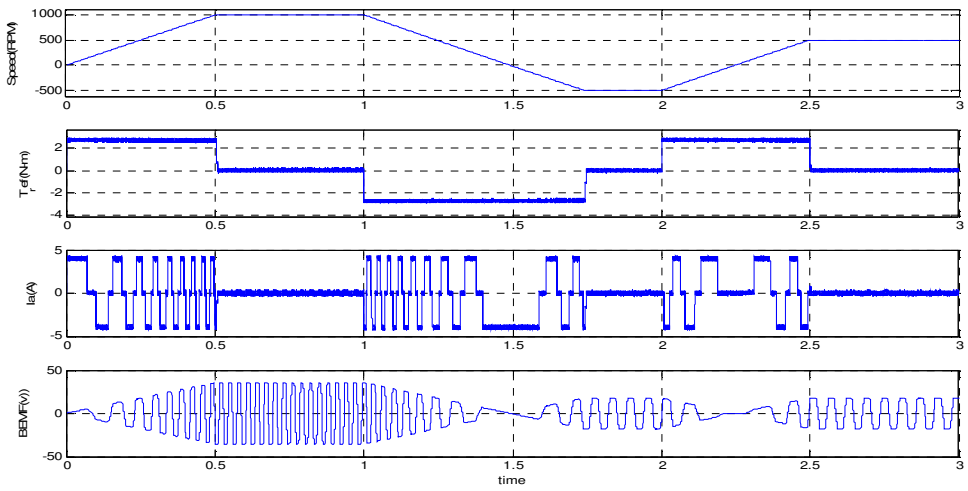

Figure 4.2 Response of the drive with the PI controller on reversal of speed direction

The Fig. 4.2 shows the response of the PMBLDC drive on speed direction reversal when using a PI speed controller. The circuit is simulated for t=3sec. The motor is allowed to start normally with set point speed of 1000 RPM; at the time instant t=1sec, the set point speed is changed to -500 RPM. The magnitude of the overshoot and the time taken to settle back to normal value is observed keenly. The figure show the plots for

rotor speed (RPM), the torque (T_e) in (N-m), stator current (i_a) in Ampere, the back emf developed in Volts. The motor speed raises from 0 RPM to the initial set point of 1000RPM in 0.505sec, when the set point is changed to -500 RPM at t=1sec, this set point speed is reached at 1.745sec. And the set point speed that is changed to 500RPM at t=2sec, is reached at the time instant 2.5sec maintaining an overshoot 0f 1.15RPM, finally it settles at 2.75sec.

4.3 Response of the drive with a Fuzzy logic speed controller

4.3.1 Response of the drive on starting and load perturbation

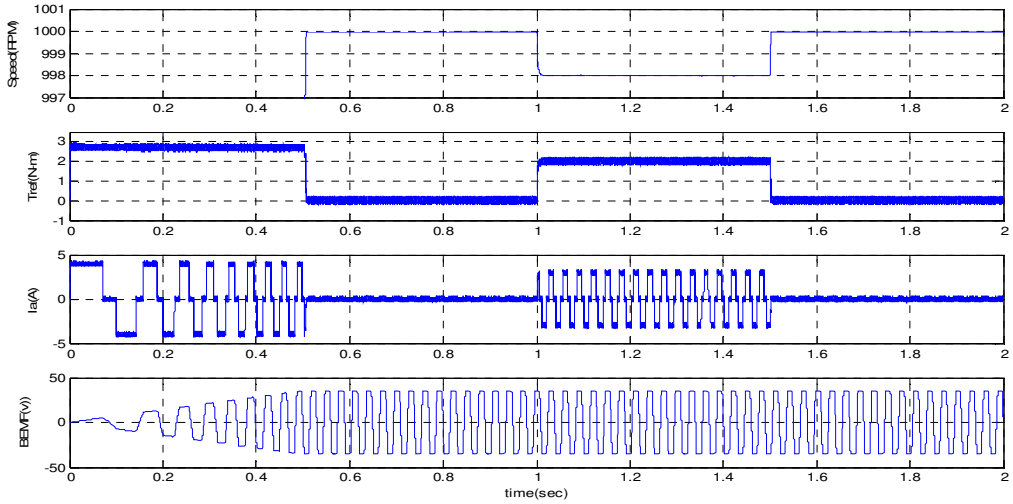

Figure 4.3 Response of the drive with the fuzzy logic controller on starting and load perturbation

The Fig. 4.3 shows the response of the PMBLDC drive on starting at a set point speed of 1000 RPM, when using a Fuzzy logic speed controller. The circuit is simulated for t=2sec. At the time instant t=1sec, 2Nm of load torque is applied to the motor, and at t=1.5sec the load is removed. The figure shows the plots for rotor speed in revolutions per minute (RPM), the electromagnetic torque (T_e) developed by the motor in (N-m), stator current (i_a) of phase a in Ampere, the back emf developed in phase in Volts. The time taken by the motor to attain the set point speed is noted, and the time in which the motor again reaches the set speed when the load is added and removed is also observed from the plot. The motor speed rises to the set point speed in 0.508 sec; the response shows an offset of 0.03 RPM, the motor could not settle at the set point speed. When load

is applied at t=1sec, a dip in speed of 2RPM from the set point is observed, moreover the response could not reach the set point; again displaying an offset of 2RPM. On removal of load at t=1.5sec an offset of 0.03RPM is observed.

4.3.2 Response of the drive during speed direction reversal

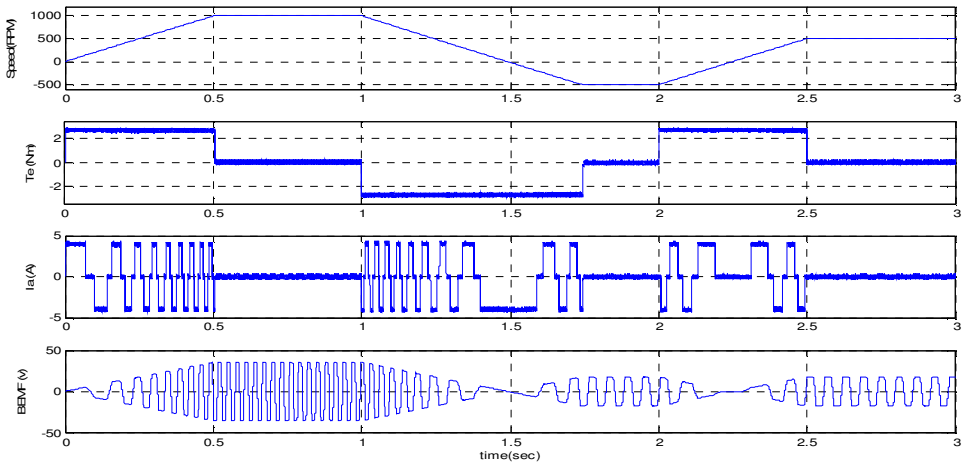

Figure 4.4 Response of the drive with the fuzzy logic controller on reversal of speed direction

The Fig. 4.4 shows the response of the PMBLDC drive on speed direction reversal when using a Fuzzy logic speed controller. The circuit is simulated for t=3sec. The motor is allowed to start normally with set point speed of 1000 RPM; at the time instant t=1sec, the set point speed is changed to -500 RPM i.e. it is made to rotate in the reverse direction. The magnitude of the overshoot and the time taken to settle back to normal value is observed keenly.

The figure shows the plots for rotor speed (RPM), the torque (T_e) in (N-m), stator current (i_a) in Ampere, the back emf developed in Volts. The motor speed raises from 0RPM to the initial set point of 1000RPM in 0.505sec, an offset of 0.03RPM is shown, when the set point is changed to -500 RPM at t=1sec, the speed rises to the 499.97 RPM at 1.745sec, the offset of 0.03RPM is maintained. When the set point speed is changed to 500RPM at t=2sec, this is achieved by the motor at the time instant 2.5sec maintaining an overshoot of 0.03RPM.

4.4 Response of the drive with a Series hybrid PI controller (Fuzzy precompensated PI)

4.4.1 Response of the drive on starting and load perturbation

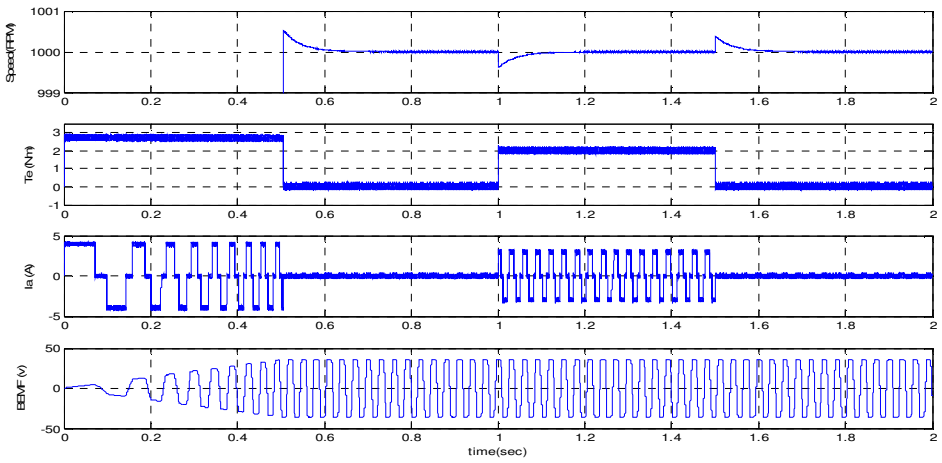

Figure 4.5 Response of the drive with the series hybrid PI controller on starting and load perturbation

The Fig. 4.5 shows the response of the PMBLDC drive on starting at a set point speed of 1000RPM, when using a series hybrid speed controller. The circuit is simulated for t=2sec. At the time instant t=1sec, 2Nm of load torque is applied to the motor, and at t=1.5sec the load is removed. The Fig. 4.5 shows the response of the drive on starting at a set point speed of 1000RPMplots for Rotor speed (RPM), torque developed (N-m), stator current (Ampere), the back emf (Volts).The time taken by the motor to attain the set point speed is noted, and the time in which the motor again reaches the set speed when the load is added and removed is also observed from the plot. The motor speed rises to the set point speed at 0.505sec, it has an overshoot of 0.38 RPM and finally settles at the set point at the time instant 0.65sec. When load is applied at t=1sec, a dip in speed of 0.49RPM is observed, the set point speed is reached at 1.16sec. An overshoot of 0.4RPM is observed on the removal of load at t=1.5sec, the response settles at the set speed is reached at the time instant 1.65 sec.

4.4.2 Response of the drive during speed direction reversal

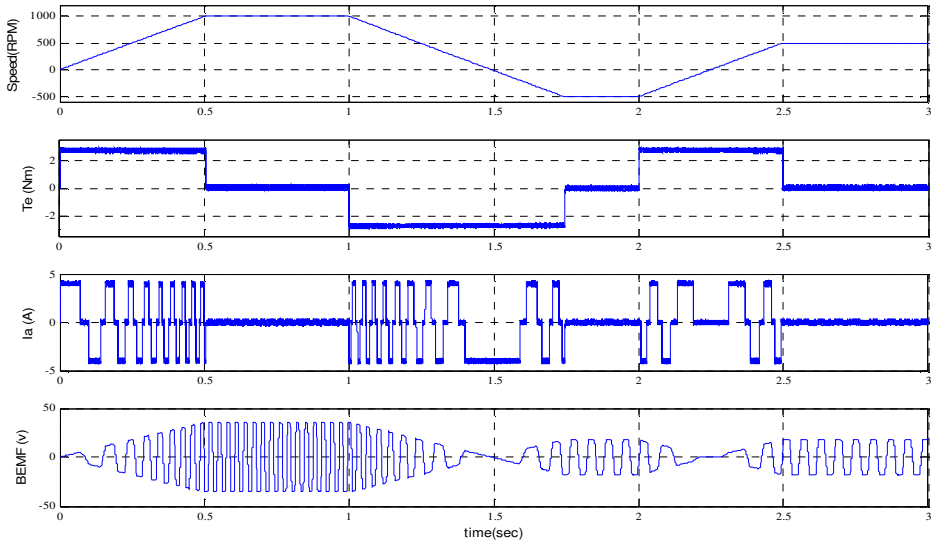

Figure 4.6 Response of the drive with the series hybrid PI controller on reversal of speed direction

The Fig. 4.6 shows the response of the PMBLDC drive on speed direction reversal when using a series Hybrid speed controller. The circuit is simulated for t=3sec. The motor is allowed to start normally with set point speed of 1000 RPM; at the time instant t=1sec, the set point speed is changed to -500 RPM i.e. it is made to rotate in the reverse direction. The magnitude of the overshoot and the time taken to settle back to normal value is observed keenly. Fig. 4.6 show the plots for Rotor speed (RPM), the torque (T_e) in (N-m), stator current (i_a) in Ampere, the back emf developed in Volts.

The motor speed rises from 0RPM to the initial set point of 1000RPM in 0.505sec, when the set point is changed to -500 RPM at t=1sec, the speed rises to the set point speed at1.74sec and has an overshoot of 0.49RPM, and finally settles at the time instant 1.90sec. The set point speed that is changed to 500RPM at t=2sec, is reached at the time instant 2.5sec and has an overshoot 0f 0.53RPM, finally it settles at 2.625sec.

4.5 Response of the drive with a Parallel hybrid PI controller (Self tuning PI)

4.5.1 Response of the drive on starting and load perturbation

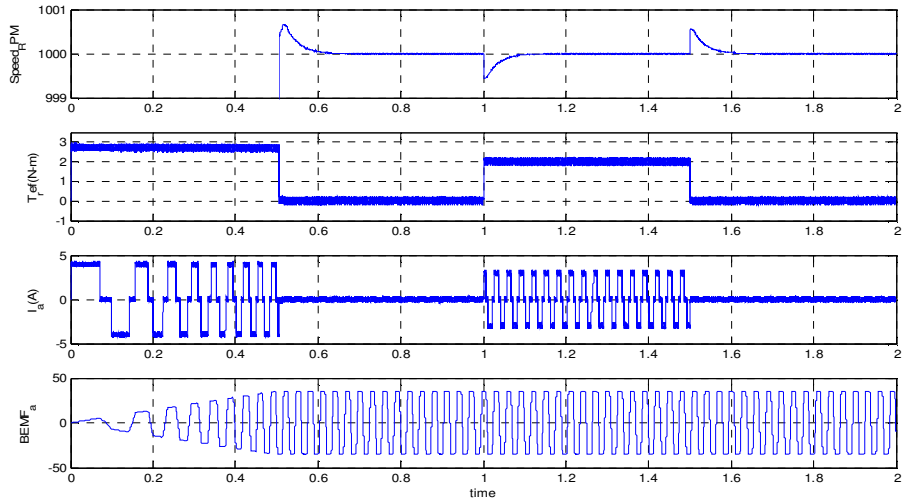

Figure 4.7 Response of the drive with the Parallel hybrid PI controller on starting and load perturbation

The Fig. 4.7 shows the response of the PMBLDC drive on starting at a set point speed of 1000RPM, when using a self tuning PI speed controller. The circuit is simulated for t=2sec. At the time instant t=1sec, 2 Nm of load torque is applied to the motor, and at t=1.5sec the load is removed. The Fig. 4.7 shows the plots for Rotor speed (RPM), torque developed (N-m), stator current (Ampere), the back emf (Volts). The circuit is simulated for t=2sec. At the time instant t=1sec, 2 Nm of load torque is applied to the motor, and at t=1.5sec the load is removed. The time taken by the motor to attain the set point speed is noted, and the time in which the motor again reaches the set speed when the load is added and removed is also observed from the plot. The motor speed rises to the set point speed at 0.505sec; it has an overshoot of 0.65 RPM and finally settles at the set point at the time instant 0.6sec. When load is applied at t=1sec, a dip in speed of 0.55RPM is observed, the set point speed is reached at 1.12sec. An overshoot of 0.56RPM is observed on the removal of load at t=1.5sec, the response settles at the set speed is reached at the time instant 1.63sec.

4.5.2 Response of the drive during speed direction reversal

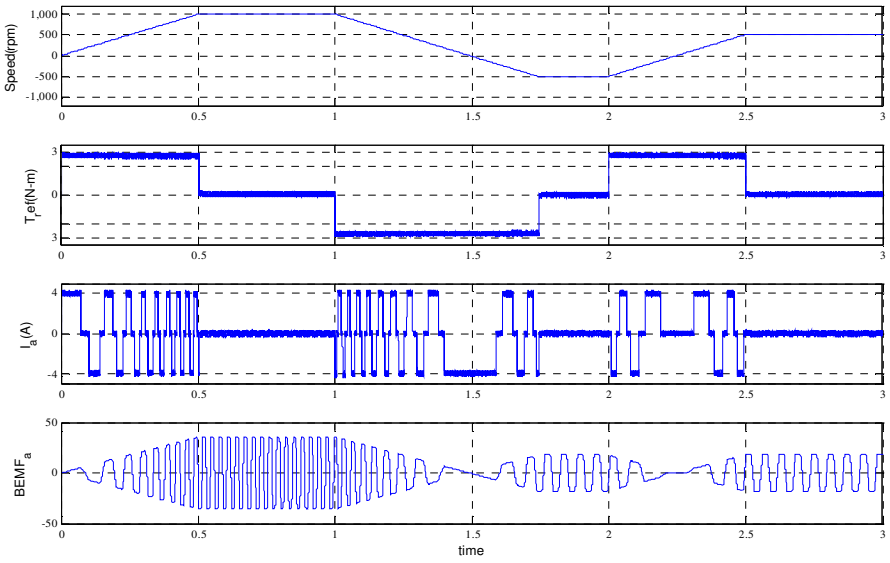

Figure 4.8 Response of the drive with the Parallel hybrid PI controller on reversal of speed direction

The Fig. 4.8 shows the response of the PMBLDC drive on speed direction reversal when using a parallel hybrid PI speed controller. The circuit is simulated for t=3sec. The motor is allowed to start normally with set point speed of 1000 RPM; at the time instant t=1sec, the set point speed is changed to -500 RPM i.e. it is made to rotate in the reverse direction. The figure shows the plots for Rotor speed (RPM), the torque (T_e) in (N-m), stator current (i_a) in Ampere, the back emf developed in Volts.

The motor speed rises from 0RPM to the initial set point of 1000RPM in 0.505sec, when the set point is changed to -500RPM at t=1sec, the speed rises to the set point speed at 1.74sec and has an overshoot of 0.6RPM, and finally settles at the time instant 1.90sec. The set point speed that is changed to 500RPM at t=2sec, is reached at the time instant 2.5sec and has an overshoot of 0.57RPM, finally it settles at 2.625sec.

4.6 Response of the drive with a reduced rule base Series hybrid PI controller

4.6.1 Response of the drive on starting and load perturbation

The Fig. 4.9 shows the response of the PMBLDC drive on starting at a set point speed of 1000 RPM, when using a series hybrid speed controller. The circuit is simulated for t=2sec. At the time instant t=1sec, 2Nm of load torque is applied to the motor, and at t=1.5sec the load is removed. The figure shows the response of the drive on starting at a set point speed of 1000RPMplots for Rotor speed (RPM), torque developed (N-m), stator current (Ampere), the back emf (Volts).The time taken by the motor to attain the set point speed is noted, and the time in which the motor again reaches the set speed when the load is added and removed is also observed from the plot. The motor speed rises to the set point speed at 0.505sec, it has an overshoot of 1.05 RPM and finally settles at the set point at the time instant 0.65sec. When load is applied at t=1sec, a dip in speed of 0.8RPM is observed, the set point speed is reached at 1.10sec. An overshoot of 0.8RPM is observed on the removal of load at t=1.5sec, the response settles at the set speed is reached at the time instant 1.55 sec. The chattering in may additionally be observed due

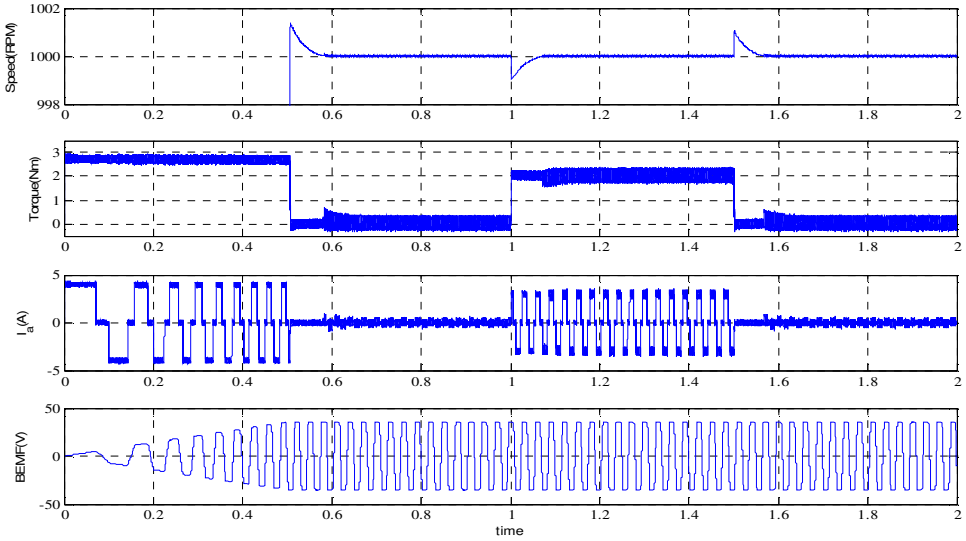

Figure 4.9 Response of the drive with the reduced rule base series hybrid PI controller on starting and load perturbation

to reduced rule base. However for practical implementation it will be filtered by the distributed resistance of the winding and inertial mass of the rotor.

4.6.2 Response of the drive during speed direction reversal

The Fig. 4.10 shows the response of the PMBLDC drive on speed direction reversal when using a series Hybrid speed controller. The circuit is simulated for t=3sec. The motor is allowed to start normally with set point speed of 1000 RPM; at the time instant t=1sec, the set point speed is changed to -500 RPM i.e. it is made to rotate in the reverse direction. The magnitude of the overshoot and the time taken to settle back to normal value is observed keenly.

The figure shows the plots for Rotor speed (RPM), the torque (T_e) in (N-m), stator current (i_a) in Ampere, the back emf developed in Volts. The motor speed rises from 0RPM to the initial set point of 1000RPM in 0.505sec, when the set point is changed to -500 RPM at t=1sec, the speed rises to the set point speed at 1.74sec and has an overshoot of 0.8RPM, and finally settles at the time instant 1.84sec. The motor speed rises from 0RPM to the initial set point of 1000RPM in 0.505sec, when the set point is changed to 500 RPM at t=2sec, the speed rises to the set point speed at instant 2.5sec and has an overshoot of 0.95RPM, and finally settles at the time instant 2.625sec.

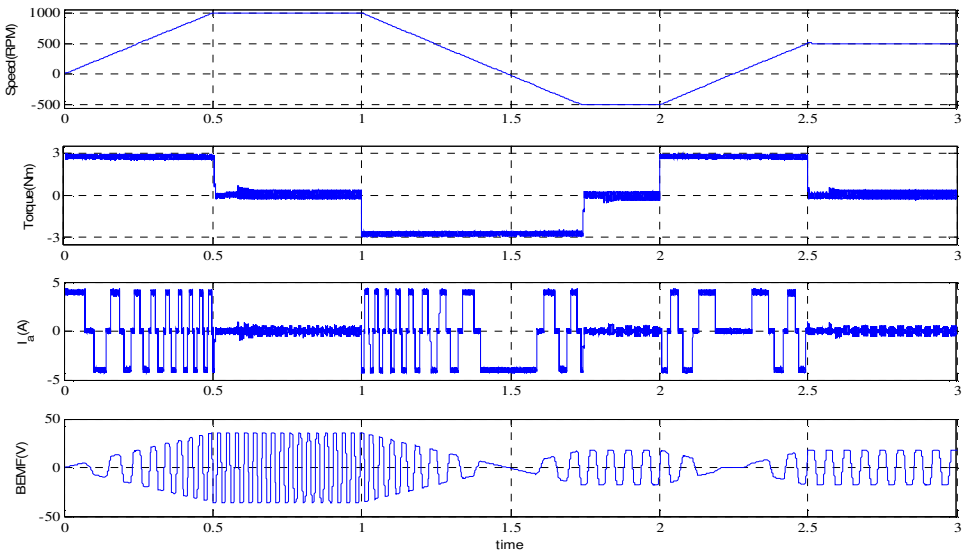

Figure 4.10 Response of the drive with the reduced rule base series hybrid PI controller on reversal of speed direction

4.7 Response of the drive with a reduced rule base Parallel hybrid PI controller

4.7.1 Response of the drive on starting and load perturbation

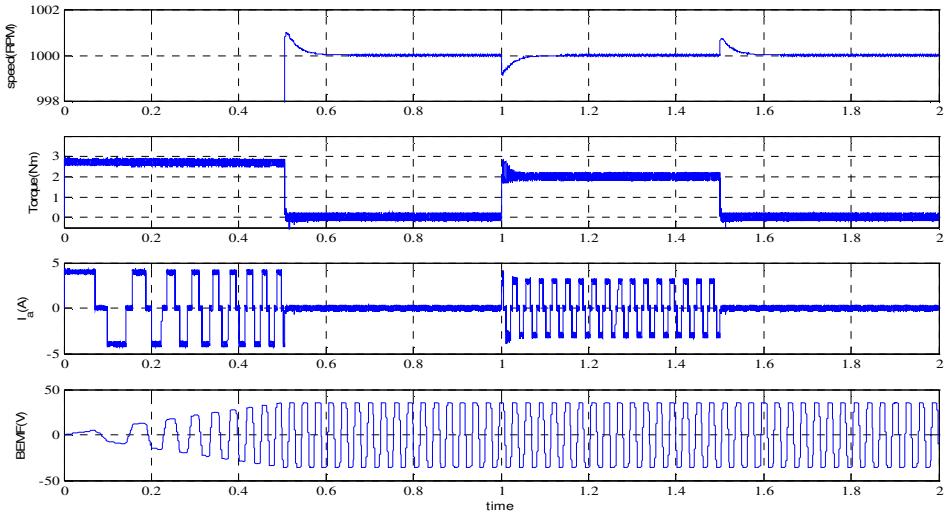

Figure 4.11 Response of the drive with the reduced rule base Parallel hybrid PI controller on starting and load perturbation

The Fig. 4.11 shows the response of the PMBLDC drive on starting at a set point speed of 1000 RPM, when using a self tuning PI speed controller. The circuit is simulated for t=2sec. At the time instant t=1sec, 2 Nm of load torque is applied to the motor, and at t=1.5sec the load is removed. The figure shows the plots for rotor speed (RPM), torque developed (N-m), stator current (Ampere), the back emf (Volts). The time taken by the motor to attain the set point speed is noted, and the time in which the motor again reaches the set speed when the load is added and removed is also observed from the plot. The motor speed rises to the set point speed at 0.505sec; it has an overshoot of 0.95 RPM and finally settles at the set point at the time instant 0.6sec. When load is applied at t=1sec, a dip in speed of 0.7RPM is observed, the set point speed is reached at 1.06sec. An overshoot of 0.84RPM is observed on the removal of load at t=1.5sec, the response settles at the set speed is reached at the time instant 1.58sec.

4.7.2 Response of the drive during speed direction reversal

The Fig. 4.12 shows the response of the PMBLDC drive on speed direction reversal when using a Self tuning PI speed controller. The circuit is simulated for t=3sec. The motor is allowed to start normally with set point speed of 1000 RPM, at the time instant t=1sec, the set point speed is changed to -500 RPM i.e. it is made to rotate in the reverse direction. The figure shows the plots for Rotor speed (RPM), the torque (T_e) in (N-m), stator current (i_a) in Ampere, the back emf developed in Volts. The motor speed rises from 0RPM to the initial set point of 1000RPM in 0.505sec, when the set point is

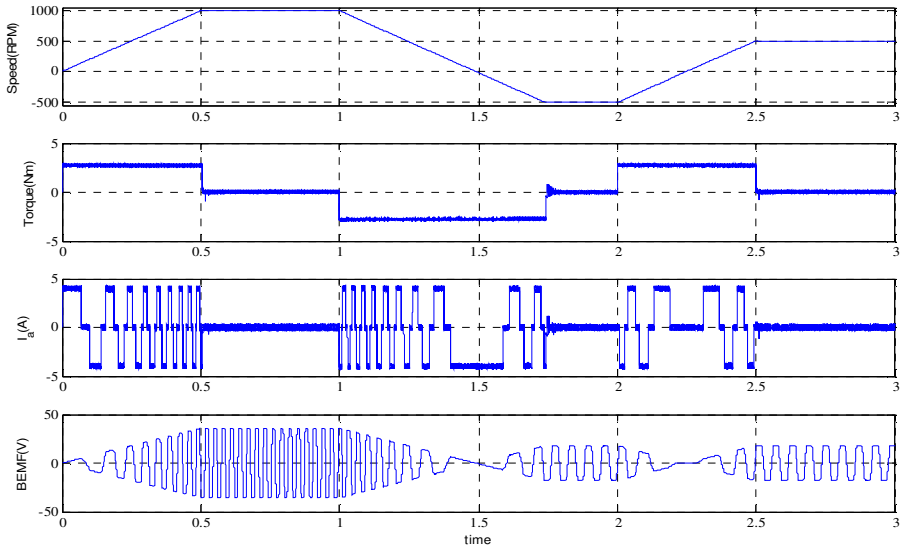

Figure 4.12 Response of the drive with the reduced rule base Parallel hybrid PI controller on reversal of speed direction

changed to -500 RPM at t=1sec, the speed rises to the set point speed at 1.74sec and has an overshoot of 0.75RPM, and finally settles at the time instant 1.82sec. The set point speed that is changed to 500RPM at t=2sec, is reached at the time instant 2.5sec and has an overshoot of 0.67RPM, finally it settles at 2.585sec.

4.8 Discussion on results

The Fig. 4.13 shows the response of the drive on starting and load perturbation for a set point speed of 1000RPM while using the controllers PI, Fuzzy Logic, Series Hybrid PI and Parallel Hybrid PI controllers. The speeds are shown at the set point. The

responses of the drive with different controller are compared in the conditions of starting and of load perturbation.

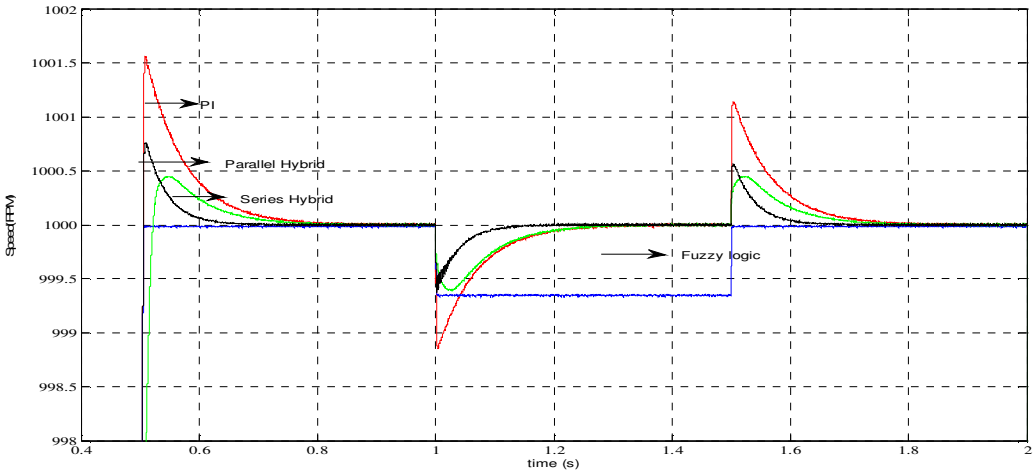

Figure 4.13 Response of drive on starting and load perturbation for all the controllers at 1000RPM

The simulation is done for t=2sec with set point speed 1000rpm throughout the simulation, a load of t=2Nm is applied to the system at t=1sec and then removed at t=1.5sec. The figure shown plot the speed with time and comparison can be done the bases of dynamic response of the drive with different speed controller.

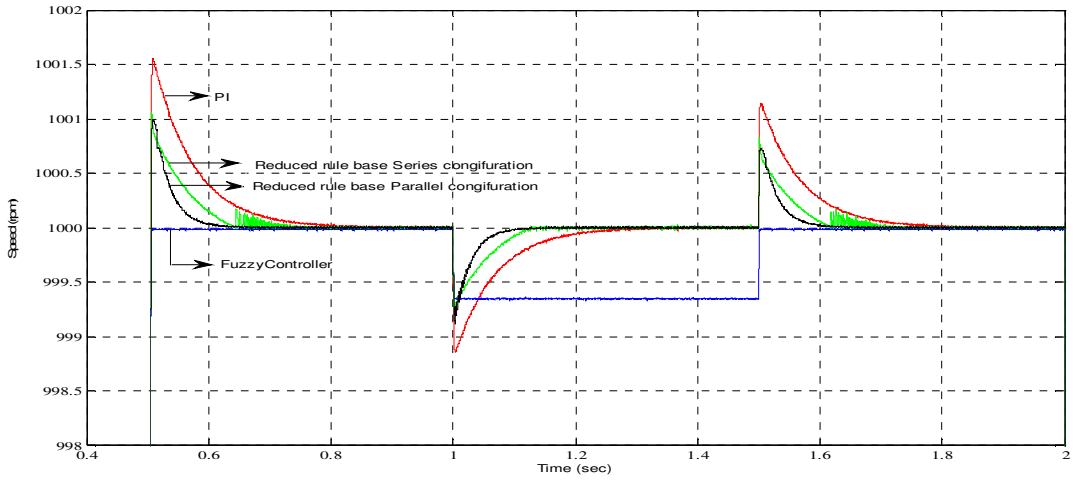

Figure 4.14 Response of drive on starting and load perturbation for the PI, FLC, reduced rule base series hybrid and parallel hybrid controller at 1000RPM

Figure 4.14 shows the response of the drive for starting and load perturbation for PI , FLC, Reduced rule base series hybrid PI, reduced rule base parallel hybrid PI controller for speed set point at 1000RPM.

The comparison for the starting response is done in terms of overshoot and the settling time (i.e. time taken to settle at the set point). It can be observed from the Table 4.1 that, the PI controller gives comparatively the higher overshoot and takes the longest time to settle making the response slower.

Type of controller	**Overshoot (RPM)**	**Settling time (sec)**
PI controller	1.55	0.8
FL controller	0.03 (offset)	0.508
Series hybrid PI controller	0.38	0.65
Parallel hybrid PI controller	0.65	0.60
Reduced rule base Series Hybrid PI	1.05	0.60
Reduced rule base Parallel Hybrid PI	0.95	0.55

Table 4.1 Comparison of response at starting

The fuzzy logic controller has the least settling time but, there is an offset displayed at the set point which is undesirable. The response with series hybrid PI controller is best in terms of reducing the overshoot; it also shows good response for the settling time. The parallel hybrid PI controller shows an overshoot, which is greater than series hybrid but lesser than rest of the controller. Both the parallel hybrid shows the best response in terms of settling time. The reduced rule base series hybrid show overshoot which is improved as compared to fixed gain PI controller response, but it greater then series hybrid. The reduced rule base parallel hybrid PI shows superior response than PI controller but inferior to the parallel hybrid controller.

The comparison between the responses is also done for load perturbation and of undershoot observed on application of load and the settling time (i.e. time taken for the

speed to reach the set point value after the instant of application of load) is presented in the Table 4.2

Type of controller	**Undershoot (RPM)**	**Settling time (sec)**
PI controller	1.15	0.25
FL controller	0.65 (offset)	-------
Series hybrid PI controller	0.49	0.14
Parallel hybrid PI controller	0.55	0.12
Reduced rule base Series Hybrid PI	0.8	0.10
Reduced rule base Parallel Hybrid PI	0.7	0.06

Table 4.2 Comparison of response on load perturbation

It can be observed from Table 4.2 that the PI controller has the highest undershoot in the present case, with a longer settling time. The FLC showed a 0.65RPM dip in speed on application of load and this is maintained for the remaining loading period. The series hybrid controller could efficiently reduce the magnitude of undershoot during load perturbation and also has a low settling time. The reduced rule base series hybrid also exhibit a reduced undershoot which is higher than series hybrid, but lower then fixed gain PI controller. Improvement in the settling time at the load perturbation is observed for reduced rule base series hybrid. A dip in undershoot is also observed for parallel hybrid with lesser settling time. Although the reduced rule base parallel hybrid controller have higher undershoot than parallel hybrid but it is lesser than PI. The settling time is best among all controllers for reduced rule base parallel hybrid. For speed reversal the results comparative for all the controllers' have similar trend.

The computation and memory requirement are limited in an embedded set up. With limited hardware the embedded system application using a low cost device such as microcontroller; the code optimization for control strategy becomes essential especially for non conventional controller.

The series hybrid controller utilized seven variables each for input and output and thus the rule base consisting of 49 rules will require huge memory space to save all the variables dynamically for one to one mapping required to realize FLC the computation requirement would also be great. This requirement further increase in the case of parallel hybrid in which there are two sets of output, the calculation has be done for two sets, have computation is doubled. The series hybrid controller configuration with minimum fuzzy variables makes a rule base consisting of only four rules. The computation time and memory required for this controller, is less as compare to parallel configuration which posses rule base of 9 rules and more number of fuzzy variables. Table 4.3 shows the comparison of computational effort and memory requirements among various controllers with respect to the rule base and number of fuzzy variables used by inputs and outputs.

Type of controller	Fuzzy variables used for each input	Number of rules in Rule base	Fuzzy variables used for each output	Memory and computational requirements
PI controller	----	-----	----	Lowest
FL controller	7	49 (7x7)	7	High
Series Hybrid	7	49 (7x7)	7	High
Parallel Hybrid	7	49 (7x7)	7	Higher
Reduced rule base Series Hybrid PI	2	4 (2x2)	4	Low
Reduced rule base Parallel Hybrid PI	3	9 (3x3)	7	Medium

Table 4.3 Comparison of controller based on computational and memory requirement

4.9 Conclusion

The modeling, analysis, design and the simulation of the PMBLDC drive system with aforesaid controllers has been done in the MATLAB/Simulink environment. A

thorough comparative study has been carried out on the drive performance with different speed controllers. It has been shown that the individual controllers have their own merits and demerits. The choice of selection of controller for a particular application should be based on typical requirement. When the requirement is of simplicity and ease of application, a PI controller is of a good choice. When the need is of intelligent and fast dynamic response then the fuzzy logic technique can be selected. When the requirement is of both intelligent response and good steady state performance with minimum overshoot, the series hybrid controller is a better choice. The parallel hybrid PI controller uses an efficient method of continuously tuning the gains of a PI controller to suitable values depending on the operating point of the system. It can deliver many advantages such as reducing the rise time of the drive to the set speed, good adaptive performance during severe load disturbance, which make the drive to maintain speed at the set speed with quite low undershoot and the least settling time. The reduced rule base hybrid PI controllers pave the way for practical implementation and achieving the control code optimization, thus decreasing the memory and computational requirement. Such a reduced rule based hybrid PI controller with a low cost microcontroller in embedded system with minimum hardware requirement depict the future for application to drives to even harsher environment and harder challenges.

CHAPTER V

MAIN CONCLUSION AND SUGGESTION FOR FURTHER WORK

5.1 General

The modeling of permanent magnet brushless DC motor drive with different type of speed controller has been successfully carried out using MATLAB/Simulink environment. Different speed controllers i.e. PI controller, fuzzy logic controller, series hybrid, parallel hybrid, reduced rule base series hybrid PI controller and reduced rule base self tuning PI controller are used for simulation study for assessment of the dynamic performance of PMBLDCM. Various techniques used for reduction in the rule base of fuzzy controller, its applied for reducing the burden on computation by code optimization. The present chapter summaries all the investigations carried out right and accordingly main conclusion are derived and suggestions for further work are also presented.

5.2 Main Concussion

The MATLAB/Simulink environment using the SimPowerSystems and fuzzy logic tool box has been extensively used for simulation of model of PMBLDCM drive using various controllers. The hybrid configuration using conventional PI and non conventional FL controller in two configurations has been investigated. The series hybrid and parallel hybrid shows good dynamic response but at the cost of large computation and memory recruitments. Need for rule base reduction in hybrid controllers along with the methods used to accomplishing the same is discussed. Further reduced rule base series hybrid PI and reduced rule base parallel hybrid PI controllers were developed. The speed response of the drive with different controllers has been compared and analyzed. The comparative study shows that individual speed controller have their own merits and demerits. The choice of selecting a speed controller for a particular application depends on the control objective.

The integration of both PI and FLC is done in such a way, so as to reap the advantage of both and avoiding their short coming. The frequent switching between the

controllers for realizing the hybrid controller reported in literature fall short on drive performance and portray large disturbances in output.

The integration of PI and fuzzy controller as two hybrid configurations and with reduced rule base is studied for PMBLDC drive operation. A successfully implementation for the PMBLDC drive is done and a comparison has been drawn between conventional PI controller, fuzzy logic controller, series hybrid, parallel hybrid and the two proposed reduced rule base hybrid controller for various transients experienced by the drives, viz, starting, speed change and load perturbations to evaluate the performance of the drive. The observed performance of the series and parallel hybrid controllers have demonstrated the ability of the proposed controllers to track the command faster than the prevalent PI controller for the similar conditions with lesser overshoot and better settling time. The simulation study has been conducted for evaluation of effectiveness of the proposed reduced rule base hybrid schemes for the PMBLDC drive. The measure to reduce the rule base has been discussed and its effect on the reduction in computation and memory requirement (in an embedded system using low cost microcontroller unit) is observed. The efforts pertaining to the optimization of control code for the hybrid schemes has been discussed. It has also been observed that the reduced rule base series hybrid controller minimize the possibilities of overshoot/ undershoot amidst transients, whereas, parallel hybrid controller operate the drive so as to attain the steady state in minimum possible time, and both the controllers drive the motor in acceleration and deceleration mode with lesser oscillations.

The comparative analysis show reduced rule base series and parallel configuration mark better response as compared to PI and FLC for suppression of oscillation and fast settling but inferior to the series and parallel hybrid. In parallel configuration the fuzzy logic is tuned online for both gains 'K_P' and 'K_I' for limited range facilitation the reduction in rule base, hence its portable implementation with micro controller unit. In series configuration, precompensation of reference speed is done by fuzzy logic with highly reduced fuzzy rule extend even greater ease in implementation with low cost device (microcontroller). However some chattering is observed near set point in series hybrid PI response. The proposed hybrid scheme has the advantage of flexibility and modularity, wherein, they may be appended to the existing PI controller without much

alteration in the hardware structure of the existing drive. Reduction in rule base would yield in code optimization for hybrid configuration, which will eventually help in hardware and cost reduction. As the necessary switch common in such hybrid controller has been discarded by limited range usage of fuzzy control, the scheme in will be applicable to other drives and can effectively work under nonlinearities and parametric changes in the motor while in operation.

5.3 Suggestion for further work

The proposed reduced rule base hybrid controller configurations displayed excellent simulation results and can be implemented on existing PI control system simply by adding the auto tuning feature. The advancement in the field of embedded systems has set up huge demand for the application specific embedded control schemes. The practical implementation of the hybrid controller (as in embedded system) can be verified for control of the PMBLDCM drive.

In the present hybrid controller scheme, the control code optimization is done using the limited range usage of fuzzy logic, thus reduced rule base and switching less hybrid schemes. The complexity reduction with code optimizations for realization of hybrid, genetic algorithm based optimization, embedded neural network with PID may be realized.

An effort to further reduce the complexity may be investigated.

References

[1] "AN885" Microchip Technology Inc. Chandler, Arizona, USA.

[2] K. Kim and R. C. Schaefer, *"Tuning a PID controller for a digital excitation control system," IEEE Trans. Ind. Appl.,* vol. 41, no. 2, pp. 485–492,Mar./Apr. 2005.

[3] K.H. Ang, G.Chong and Y.Li,*"PID control system analysis, design and technology, IEEE Trans. Control Systems Tech.,* vol.10, no.4 pp.559-576, July2005.

[4] K. Kim and R.C. Schaefer, *"Tunning a PIC controller for a digital excitation control system,", IEEE Trans. Ind. Appl.,* vol. 41, no.2, pp. 485-492, March /April 2005.

[5] Jong-Hwan Kim, Kwang-Choon Kim and Edwin K. P. Chong, *"Fuzzy Precompensated PID Controllers," IEEE Trans. Control Systems Tech.,* vol. 2, no.4, Dec. 1994.

[6] Nordin Saad and Oyas Wahyunggoro, *"Evaluations of Fuzzy-Logic-Based Self Tuning PI Controller and Fuzzy-Scheduled PID Controller for DC Servomotor," in Proc. Information Tech. ,ITSim*, August 2008.

[7] A. Hazzab, B. Mazari, K. Bousserhane, M. Kamli and M. Rahli, *"Adaptive PI Controller using Fuzzy System Optimized by Genetic Algorithm for Induction Motor Control," in Proc. CIEP*, October 16-18, 2006.

[8] Ahmed Rubaai, Abdul R. Ofoli and Marcel J. Castro-Sitiriche, *"Design and Implementation of Parallel Fuzzy PID Controller for High-Performance Brushless Motor Drives: An integrated Environment for Rapid Control Prototyping," IEEE Tranc. On Ind. App.* Vol. 44, no.4, July/Aug. 2008.

[9] Amit Vilas Sant and K. R. Rajagopal" *PM Synchronous Motor Speed Control Using Hybrid Fuzzy-PI with Novel Switching Functions," IEEE Tranc. Magnetics,* vol.45, no.10, pp.4672-4675, Oct. 2009.

[10] A. Hazzab, B. Mazari, K. Bousserhane, M. Kamli and M. Rahli, "Adaptive PI Controller using Fuzzy System Optimized by Genetic Algorithm for Induction Motor Control" in Proc. CIEP,October 16-18, 2006.

[11] Q. Cao, M. H. Lim, J. H. Li, Y. S. Ong, and W. L. Ng, *"A context switchable fuzzy inference chip," IEEE Trans. Fuzzy Syst.,* vol. 14, no. 4, pp. 552–567, Aug. 2006.

[12] I. Baturone, F. J. Moreno-Velo, V. Blanco, and J. Ferruz, *"Design of embedded DSP-based fuzzy controllers for autonomous mobile robots," IEEE Trans. Ind. Electron.,* vol. 55, no. 2, pp. 928–936, Feb. 2008.

[13] L. R. Limongi, R. Bojoi, G. Griva, and A. Tenconi, *"Performance comparison of DSP-based current controllers for three-phase active power filters," in Proc. IEEE Int. Symp. Ind. Electron.,* 2008, pp. 136–141.

[14] Ahmed Rubaai "*A DSP-based switching motor controller," in Proc. IAS,* October 2008.

[15] Ahmed Rubaai, Abdul R. Ofoli and Marcel J. Castro-Sitiriche, *"DSP-based laboratory implementation of hybrid fuzzy-PID controller using genetic optimization for high-performance motor drives," IEEE Tranc. Ind. App.* Vol. 44, no.6, Nov/Dec. 2008.

[16] Ahmed Rubaai and Paul Young, *"DEP-based Fuzzy neural network PI/PD like fuzzy controller for motion controls and drives", in Proc. IAS,* 3-7Oct.2010.

[17] A. Goedtel, Ivan N. da Silva and M. Suetake, *"Embedded DSP-based compact fuzzy system and its application for indiction–motor V/f speed control," IEEE Tranc. Ind. Elect.* Vol. 58, no.3, March 2011.

[18] Ahmed Rubaai and Paul Young, *"EKF-based PI/PD-like fuzzy-neural-network controller for brushless drives," IEEE Tranc. Ind. App.* Vol. 47, no.6, Nov/Dec. 2011

[19] B. Singh, B. P. Singh, and S. Dwivedi, *"DSP based implementation of hybrid fuzzy PI speed controller for direct torque controlled permanent magnet synchronous motor drive," Int. J. Emerging Electric Power Syst.,* vol. 8, no. 2, pp. 1–22, 2007, art. 6.

Appendix

1. **Motor specifications**

   Rating: 2.0 h.p.

   No. of Poles: 4

   Type of connection: Star

   Rated Speed: 1500 rpm

   Rated current: 4A

   Resistance/Ph: 2.8 Ohm

   Back EMF Constant: 1.23VSec/rad

   Self & Mutual Inductance: 0.00521 H/phase

   Moment of Inertia: 0.013 Kg-m^2

2. **Controller gain values**

   The gain values used for the PI controller are

   $K_P = 3$, $K_I = 45$