Chapter 1: INTRODUCTION

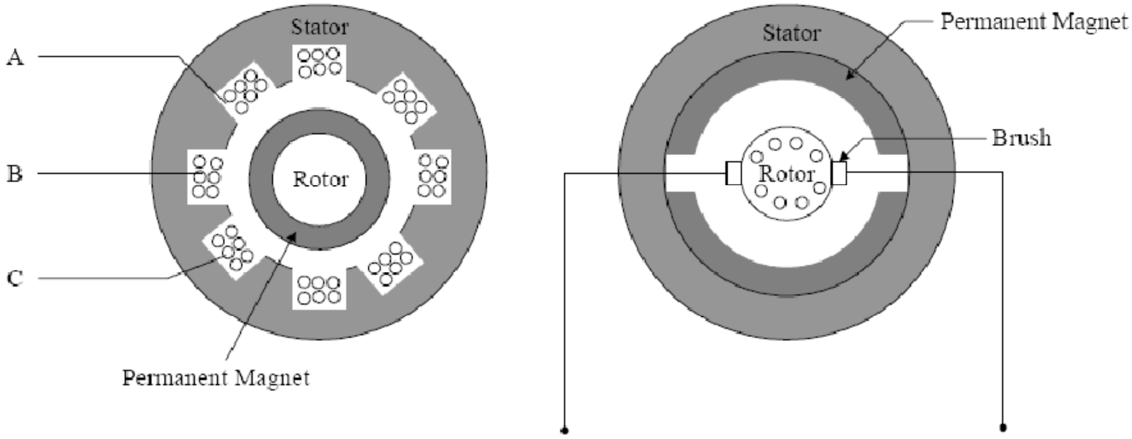
1.1 General

The availability of modern permanent magnets (PM) with considerable energy density lead to the development of DC machines with PM field excitation in the 1950s. The introduction of permanent magnets to replace electromagnets, which have windings and require an external electrical energy source, resulted in compact DC machines. The synchronous machine with its conventional field excitation in the rotor is replaced by PM excitation; dispensing the slip rings and brush assembly. With the advent of switching power transistors and SCRs, the replacement of the mechanical commutator with an electrical commutator in the form of an inverter was achieved. These two developments contributed to the PM synchronous and the Permanent magnet brushless DC machines (PMBLDCM). With this configuration, the armature of the DC machine need not be on rotor with the electrical commutator replacing its mechanical version. Therefore, the armature of the machine can be on the stator, enabling better cooling and allowing higher voltages to be achieved. The excitation field that used to be on the stator is transferred to the rotor with the PM poles. These machines are nothing but 'an inside out Dc machine' with the field and the armature interchanged from the stator to rotor, and rotor to stator respectively.

1.2 The Permanent Magnet Brushless DC Motor drive

1.2.1 The Basic construction and operation principle

The permanent magnet synchronous machines are classified on the basis of the wave shape of the induced emf, in their stator windings i.e. sinusoidal and trapezoidal. The sinusoidal type is known as the permanent magnet synchronous machine; the trapezoidal type is called the PM brushless dc machine. The PMBLDC machines have more power density than PM synchronous machine. The major reason for the popularity of this machine over its counterpart is control simplicity. Unlike a brushed DC motor, the commutation of a BLDC motor is controlled electronically. To rotate the BLDC motor, the stator windings should be energized in a sequence. It is important to know the rotor position in order to understand which winding will be energized following the energizing sequence. To initiate the onset and commutation of current in the phase of the machine, the beginning and end of the constant portion of the induced emf has to be tracked.



Brushless dc motor

Brushed dc motor

Fig. 1.1 The basic constructions of PMBLDC motor and Brushed DC motor

That amounts to only six discrete positions for a three phase machine in an electrical cycle. These signals could easily be generated with three Hall sensors displaced from each other by 120 electrical degrees. The Hall sensors are mounted facing a small magnet wheel fixed to the rotor and having the same number of poles as the rotor of the PMBLDCM, or an extended rotor beyond the stack length may provide the same information to the sensors. Such an arrangement tracks the absolute position of the rotor magnets and hence the shape and position of the induced emfs in all the machine phases. Rotor position is sensed using Hall Effect sensors embedded into the stator. Most BLDC motors have three Hall sensors embedded into the stator on the non-driving end of the motor as shown in the figure 1.2. Whenever the rotor magnetic poles pass near the Hall sensors, they give a high or low signal, indicating that the N or S pole is passing near the sensors. Based on the combination of these three Hall sensor signals, the exact sequence of commutation can be determined.

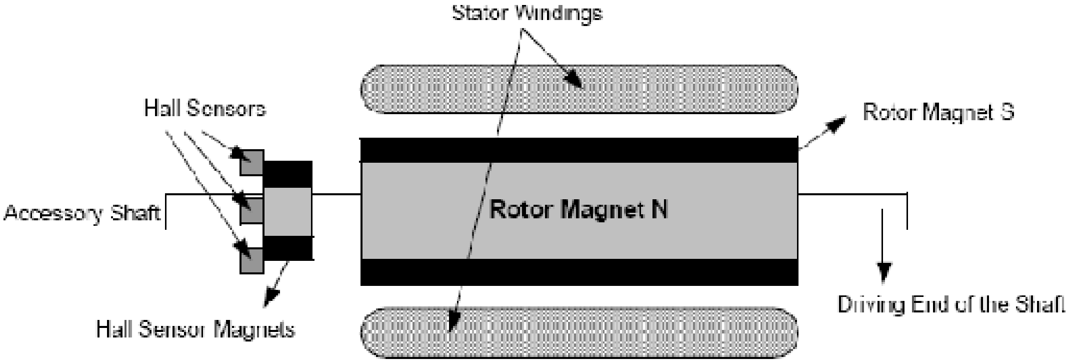


Fig. 1.2 The Hall sensors mounted on the rotor shaft

For the PMBLDCM position feedback it requires only six discrete absolute positions for a three phase machine. Further, the control involves significant vector operations in the PMSM drive, whereas such operations are not required for the operation of the PMBLDCM drive.

1.2.2 The Ideal operating behavior of PMBLDCM drive system

Even though the trapezoidal types of induced emfs have constant magnitude for 120 electrical degrees both in the positive and negative half cycles, as shown in the figure, the power output

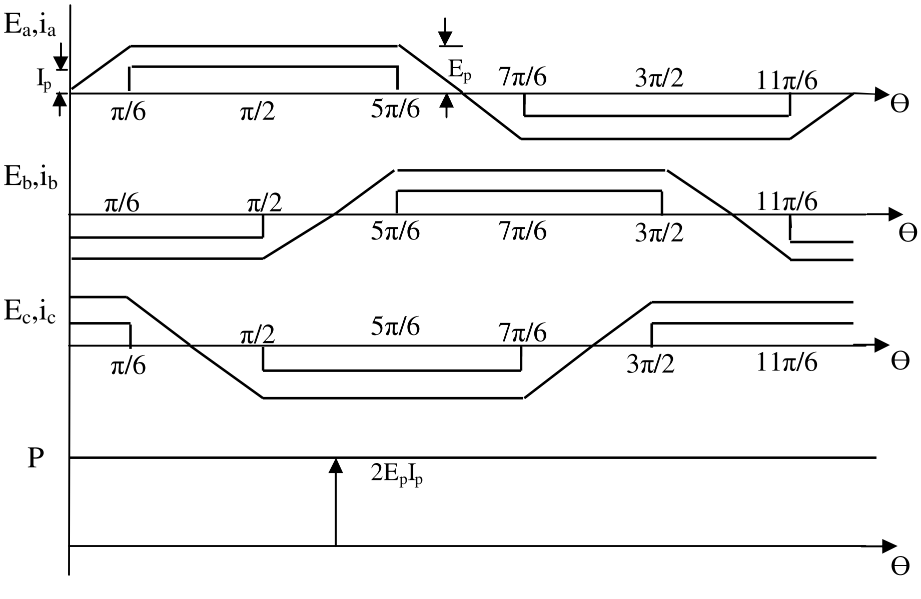


Fig. 1.3 the back emf and phase current waveforms of the three phases

can be uniform by exciting the rotor phases with 120 degrees wide currents. As shown in the figure, the currents cannot rise and fall in zero time, hence in actual operation; there are power pulsations during the turn ON and the turn OFF of the currents for each half cycle.

1.2.3 Basic PMBLDCM drive system

The terminology brushless DC motor or BLDC is used for this machine because usually the motor is combined with an optical encoder, current measurements, Hall sensors for current commutation, an amplifier and feedback controller so that it behaves like a DC motor. That is, as indicated in the Figure, the currents and motor position are fed back to the controller; the controller then uses PI current loops of the form to force the currents to track the references. The input to the controller is simply I_p, so that with the inner current control loops working properly, the equations of the motor become

$J[d\omega/dt]=KI_p-T_l$, where ‘ω’ is the angular velocity

This is the same form as the current command DC motor with torque constant ‘K’. The system of Figure is what one refers to as a "brushless DC motor" which the user obtains as a complete system from the manufacturer.

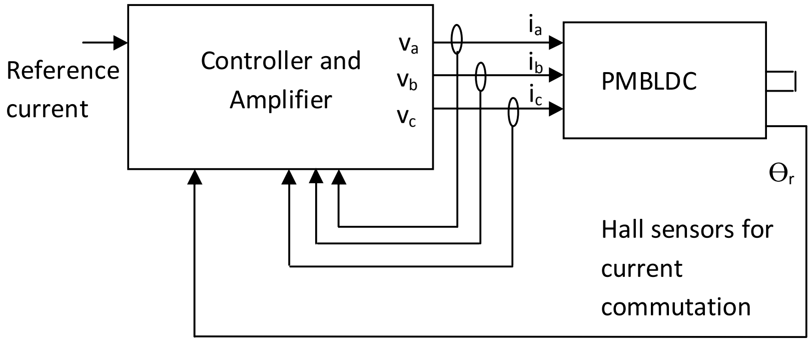


Fig. 1.4 Block diagram of the basic PMBLDCM drive

Typically, the position sensor for the current commutation, that is, for tracking the current references, is done with Hall-effect sensors. To track the current references, the phase current plots in Figure show that one only needs to determine the position of the rotor at multiples of Pi/3 or 60 degrees as the current in any particular phase changes only at some multiple of 60 degrees.

1.2.4 Advantages of the PMBLDCM over Brushed DC motor

The BLDC motor has many advantages [4] over a brushed Dc motor due to its construction, low inertia and Electronic commutation.. Table 1.1 gives the major advantages of the BLDC over the Brushed DC motor. However, the requirement of a complex controller for even a constant speed operation and the high cost of building are the disadvantages over its counterpart. The BLDC has many advantages over a Brushed Dc machine in terms of less maintenance, safety in explosive environments, achievable high speed limits with no mechanical constraints, low rotor inertia and efficiency etc.

Feature	**PMBLDC motor**	**Brushed Dc motor**
Maintenance	Less required due to absence of brushes.	Periodic maintenance is required.
Life	Longer	Shorter
Speed/Torque Characteristics	Flat–Enables operation at all speeds with rated load	Moderately flat–At higher speeds, brush friction increases, thus reducing useful torque.
Output Power/Frame Size	High- allows for better cooling facility	Moderate/Low
Rotor Inertia	Low, because it has permanent magnets on the rotor.	Higher rotor inertia, which limits the dynamic characteristics.
Speed Range	Higher	Lower – Mechanical limitations by the brushes.
Electric Noise Generation	Low	Arcs in the brushes will generate noise causing EMI in the equipment nearby.
Efficiency	High- no voltage drops against brushes	Moderate

Table 1.1 Comparison between PMBLDCM and brushed DC motor

Although the control of Brushed DC motor is simple and inexpensive, it faces the main drawback in its mechanical commutation part and the high moment of inertia.

1.3 The Concept of Controllers

In control theory, a controller is a device which monitors and affects the operational conditions of a given dynamical system. The operational conditions are typically referred to as output variables of the system which can be affected by adjusting certain input variables. Based upon the behavior of the output, the control action is taken, such that the set point value is reached. The factor based on which the control action is taken differs from one method to another. The different kinds of control methods available are discussed in this section.

1.3.1 Proportional controller

Of all the controllers, the basic controller is the Proportional, or, ‘P’ controller. The control law is simple: control is directly proportional to error. Proportional control is the easiest feedback control to implement, and simple proportional control is probably the most common kind of control loop. The proportional control is just the error signal multiplied by a constant and fed out to the drive. The most significant shortcoming of the P control is, it allows DC offset; it drops in the presence of fixed disturbances. Such disturbances are commonly come across in all the systems. DC offset cannot be tolerated in most of the systems, but where it can, the normal ‘P’ controller will solve the purpose.

1.3.2 Proportional Integral controller

With PI control, the P gain provides similar operation to that in the P controller, and the ‘I’ gain provides DC stiffness. Larger ‘I’ gain provides more stiffness but also more overshoot. The primary short coming of the P controller i.e. the DC offset can be readily eliminated by adding an integral gain to the control law. Because the integral will grow ever larger with even small Dc offset error, sufficient value of the integral gain will eliminate the DC offset droop. Integral gain is added to add long term precision to a control loop. The main drawback is that the integral controllers are more complicated to implement. The integral controller lacks a wind up function to control the integral value during saturation.

1.3.3 Proportional Derivative controller

The P controller is augmented with a ‘D’ term to allow higher proportional gain. The D gain advances the phase of the loop by the virtue of the 90 degree phase lead of a derivative. Using the D gain will usually allow the system responsiveness to increase. The differential term is the last value of the position minus the current value of the position. This gives a rough estimate of its velocity, which predicts where the position will be in a while. The ‘PD’ controller is fast, powerful but more susceptible to stability problems, sample irregularities noise and high frequency oscillations. Derivatives gave high gain at high frequencies. So some ‘D’ surely helps the gain margin but too much hurts the gain margin by adding the gain at the phase crossover, typically at high frequencies. Also the derivative gain is sensitive to noise. In case of the differential gain, the output is proportional to the position change divided by the sample time. If the position is changing at a constant pace but the sample time varies from sample to sample, we will get noise. Since the differential gain is usually high, this noise is amplifies as a great deal. Noise is usually spread evenly across the frequency spectrum hence differential control suffers from noise problems. The Control commands and the plant outputs usually have most of their content at lower frequencies. Finally we can conclude that Proportional control passes noise, integral control averages its input signal, which tends to kill noise. Differential control enhances high frequency signals, so it enhances noise. The D gain term needs to be followed by a low pass filter to reduce the noise content.

1.3.4 Proportional Integral Derivative controller

The PID controller adds differential gain to the PI controller. The proportional value determines the reaction to the current error, the integral value determines the reaction based on the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve or the power supply of a heating element. A PID controller is a two zone controller. The 'I' gain forms the low frequency zone. The benefit of the D gain is that it allows the P gain to be set higher than it could be set otherwise. The P and D gains together form the high frequency zone. The three terms describe the basic elements of a PID controller. Each of these controllers performs a different task and has a different effect on the functioning of a system. On a typical PID controller these elements are driven by combination of the system command and the feedback signal from the object that is being controlled. Their outputs are added together to form the system output. A PID controller provides faster response than a PI controller but is usually harder to control and more sensitive to changes in the plant model.

1.3.5 Fuzzy Logic controller

When confronted with a control problem for a complicated physical process, the control engineer usually follows a predetermined design procedure which begins with the need for understanding the process and the primary control objectives. The difficult task of modelling and simulating complex real world systems for control systems development, especially when implementation issues are considered, is well documented. Even if a relatively accurate model of a dynamic system can be developed it is often too complex to use in controller development, especially for many conventional control design procedures that require restrictive assumptions for the plant (e.g., linearity). It is for this reason that in practice conventional controllers are often developed via simple crude models of the plant behavior that satisfy the necessary assumptions, and via the ad hoc tuning of relatively simple linear or nonlinear controllers. 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 design have been relatively successful (the vast majority of all controllers currently in operation are conventional PI,PID controllers). The following questions may arise how much of the success can be attributed to the use of the mathematical model and conventional control design approach, and how much should be attributed to the clever heuristic 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?

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). The inference mechanism in the fuzzy controller reasons over the information in the knowledge base, the process outputs, and the user-specified goals to decide what inputs to generate for the process so that the closed-loop fuzzy control system will behave properly (e.g., so that the user specified goals are met). For the cruise control example discussed above, it is clear that anyone who has experience in driving a car can practice regulating the speed about a desired set-point and load this information into a rule base. For instance, one rule that a human driver may use is "IF speed is lower than the set point THEN press down further on the accelerator pedal': A rule that would represent even more detailed information about how to regulate the speed would be "IF speed is lower than the set point AND speed is approaching the set-point very fast THEN release the accelerator pedal by a small amount'! This second rule characterizes our knowledge about how to make sure that we do not overshoot our desired (goal) speed. Generally speaking, if we load very detailed expertise into the rule base we enhance our chances of obtaining better performance. Overall, the focus in fuzzy control is on the use of heuristic knowledge to achieve good control, whereas in conventional control the focus is on the use of a mathematical model for control systems development and subsequent use of heuristics in implementation.

There are four principal elements to a fuzzy logic controller:

- Fuzzification module
- Knowledge base.
- Decision making block
- De-fuzzification module

Other non-fuzzy elements which are also part of the control system include the sensors, the analogue–digital converters, the digital–analogue converters and the normalisation circuits. There are usually two types of normalisation circuits: one maps the physical values of the control inputs onto a normalized universe of discourse and the other maps the normalized value of the control output variables back onto its physical domain.

A. *Fuzzifier module*

The fuzzification module converts the crisp values of the control inputs into fuzzy values, so that they are compatible with the fuzzy set representation in the rule base. The choice of fuzzification strategy is dependent on the inference engine, i.e. whether it is composition based or individual-rule-firing based.

B. *Knowledge base*

The knowledge base consists of a database of the plant. It provides all the necessary definitions for the fuzzification process such as membership functions, fuzzy set representation of the input–output variables and the mapping functions between the physical and fuzzy domain.

C. *Decision making block*

The rule base is essentially the control strategy of the system. It is usually obtained from expert knowledge or heuristics and expressed as a set of IF-THEN rules. The rules are based on the fuzzy inference concept and the antecedents and consequents are associated with linguistic variables. The response of the controller to input conditions is determined by processing the rule base module. The antecedent of the rule corresponds directly to the degree of membership calculated during the fuzzification process. The value of the least true antecedent is applied to the strength of the rule. When more than one rule is applied to the same action, the common practice is to use the highest strength rule.

D. *De-fuzzification*

The output response of the controller must be non fuzzy in nature. This module defuzzifies the response after the evaluation of the rule base module. Normally the weighted average method is used for de-fuzzification.

1.3.6 The Series Hybrid (Fuzzy precompensated PI) controller

Standard controllers used in practice, such as PI, PD and PID controllers, suffer from poor performance when applied directly to systems with unknown nonlinearities like dead zone, saturation, limit cycles etc.. For example, a steady-state error occurs when applying a conventional PD controller to a system with dead zones the size of the steady-state error increases with the dead zone width. To eliminate the steady-state error, we may attempt to use a PID controller. However, the transient performance in this case is poor. It is well known that a conventional PI controller is most widely used in industry due to its simple control structure, ease of design and low cost. However, the PI type controller only cannot give a good control performance. Moreover, it suffers from disadvantages of slower response, larger overshoots, and oscillation. As the PMBLDC machine has nonlinear model, the linear PI control is not a good option. This boosted the use of nonlinear control schemes for PMBLDC motor. However, while using a pure fuzzy controller in the front end, we may observe a steady state error in the system response. A major limitation of the fuzzy controller is the lack of a systematic methodology for developing fuzzy rules. A set of fuzzy rules need to be manually adjusted on a trial and error basis before it reaches the desired level of performance. This tuning process may be non trivial and could be time consuming for a first time fuzzy logic controller developer. Even though the FLC has been designed by an expert, the limitations it may have to face may be given as below:

- Will the behaviours observed by a human expert include all situations that can occur due to disturbances, noise, or plant parameter variations?
- Can the human expert realistically and reliably foresee problems that could arise from closed-loop system instabilities or limit cycles?

- Will the expert be able to effectively incorporate stability criteria and performance objectives (e.g., rise time, overshoot, and tracking specifications) into a rule base to ensure that reliable operation can be obtained?
- Can an effective and widely used synthesis procedure be devoid of mathematical modelling and subsequent use of proven mathematical analysis tools?

Hence there is need for controllers that depend completely neither on the fuzzy controller nor on a conventional controller, but could use the combination of these two controllers to exploit the advantages of both the controllers. For this purpose, a controller can be proposed which is a hybrid of the FLC and the conventional controller. In addition to being able to adapt automatically to a new environment, this controller can further simplify the task of developing rules, for the designer only needs to come up with an initial set of rules which are roughly correct. The burden of manually fine tuning the rules is thus removed from the designers. If the output of a speed controller is a combination of outputs of two speed controllers (FL and PI), combined together as a weighted sum to eliminate certain disadvantages then the resulting controller is referred to as a Hybrid controller.

The configuration of the series hybrid controller or the “Pre-compensated controller” can be illustrated as in the figure. The scheme consists of a conventional PI control structure together with our proposed fuzzy pre-compensator. The purpose of the fuzzy pre-compensator is to modify the command signal to compensate for the overshoots and undershoots and steady state errors present in the output response when the plant has unknown non-linearities. This is achieved by the advance alteration of the reference control signal in accordance with system response. The processing occurs as follows:

(i) The speed error and rate of change of speed error are calculated and are fed to the fuzzy pre-compensator.

(ii) The output of this FLC is added with the actual speed reference signal to generate the modified speed reference speed signal.

This modified reference signal is used by the remaining PI control strategy. The hybrid controller is advantageous in many aspects. Its performance exceeds any fixed classical linear or nonlinear smooth controller. Multiple objectives such as robustness, disturbance attenuation, response speed, accuracy are achieved.

1.3.7 Self-tuning PI controller

In electrical drive control, the PI or PID controllers are mostly used because, these controllers are simple structures, and easy to design. However, these controllers are tuned to give the best performance at particular operating condition and it needs to retune if the operating condition is changed to retain that performance. And also in industrial applications, there are many uncertainties, such as system parameter uncertainty, external load disturbance, friction force, un-modeled uncertainty, always diminish the performance quality of the pre-design of the motor driving system. To cope with this problem, in recent years, many intelligent control techniques. Implementation of artificial intelligence technique for tuning the conventional controller is one of the ways to ensure the better performance of the drive for a wide range of operating conditions. Fuzzy logic controller (FLC) is one of the well accepted intelligent control technique and its applications has broadened to many successful industrial control applications. FLC as discussed above provides a formal methodology for implementing the humans' heuristic knowledge in form of control rules.

These dynamically modified gains are used by the PI speed controller for the further control action. The basis for the Self tuning PI controller corresponds to condition when the error is high i.e. the actual speed is much less than the set point speed, the proportional gain plays the key role for faster system response, when the speed is near the set point, the integral gain comes into action to completely eliminate the steady state error. It can be presented in this way that, the gains of the PI speed controller are constantly modified by the Self tuning controller in parallel depending on the speed error and the change in the speed error such that the drive system achieves adaptive nature to the variation in the operating conditions like load variations.

The advantages the proposed controller can achieve are:

- Minimum rise time during the starting response with no or very less overshoot.
- Adaptive performance during the load variations with minimum deviation from the set point speed.
- Smooth response even in the presence of unknown non-linearities in the drive system such as friction, dead zone etc.

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, modelled and simulated to identify the suitable controller for appropriate conditions. The Scope of work in the present thesis chiefly is to construct the PMBLDCM drive system in the MATLAB/SIMULINK environment. Then to carry out simulation studies for the speed control of the drive using the PI controller, Fuzzy logic controller, Series hybrid controller and the proposed Self-tuning (fuzzy tuned 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 describes elaborately 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 and the proposed Self-tuning controller. The different hybrid controller configurations proposed and implemented, and their methods are discussed in brief here.

Chapter 3

This chapter presents the modelling and simulation of the drive system, with the PI, FLC, series hybrid and the proposed Self-tuning 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 response, 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: 2 LITERATURE SURVEY

2.1 Introduction

From the available literature, it is revealed that the use of specific controllers for speed control of a PMBLDC has been used for enhancing the performance of the drive for a specific application. Motor control by using Fuzzy logic is a promising technique for extracting good performance from the available range of motors. Fuzzy logic offers a convenient way of designing controller from experiences and knowledge about the process being controlled. This heuristic performance can enhance the performance, reliability and robustness of the closed loop system more than the conventional controllers. Research has proved that a properly designed fuzzy controller can outer perform a conventional PID controller such that the overall performance can be substantially improved. The major limitation of fuzzy logic control is the lack of a systematic methodology for developing fuzzy rules. During the past few years several approaches for developing self organizing fuzzy controllers have been proposed. Dedicated simulation software like MATLAB with Simulink and fuzzy logic toolbox has made the modeling and simulation of the system efficient and simple. The advancement in the speed control techniques from a basic proportional control to fuzzy logic and other advanced techniques like Gain scheduling control, Sliding mode control, Self tuning control, etc. have resulted in a remarkable improvement in the response of the drive. Elimination of steady state error, overshoot and oscillations has lead to the practical implementation of such control techniques in the real time.

2.2 Literature Review

The improvement of the speed response of the drive has been the topic of research in the present times. The quality of the performance of the drive is generally defined through performance indices such as starting time, rise time, settling time, steady state error and the adaptability of the drive. The response of the drive is highly affected by the type of speed controller used in the control structure. The different configurations of the proposed controllers are studied in this section. The proportional and the proportional integral speed controller are considered as the basic controllers among the various other types of available speed controllers. Various controller configurations including some Hybrid controllers proposed till date are discussed in the present section.

The most widely used controller in the industrial applications is the PID-type controllers because of their simple structures and good performances in a wide range of operating conditions. In the literature, the PID controllers can be divided into two main parts: In the first part, the controller parameters are fixed during control operation. These parameters are selected in an optimal way by known methods such as the Zeigler and Nichols, poles assignment etc. Hand tuning method [1] is also one of the popular methods used today for PID tuning. The fixed gain PID controllers are simple but cannot always effectively control systems with changing parameters or having a strong nonlinearity; and may need frequent on-line retuning. In the second part, the controllers have an identical structure to PID controllers but their parameters are tuned on-line based on parameters estimation of the process. Such controllers are known as adaptive PID controllers [1].

Even though Fuzzy controllers are known for their fast response and good performance in the presence of non linearities, this standard FL controller cannot react to change in operating conditions. The FL controller needs more information to compensate nonlinearities when the operation conditions change. Moreover when the number of the fuzzy logic inputs increase, the dimension of the rule base increases too. Thus, the maintenance of the rule base is more time consuming. Another disadvantage of the FL controllers is the lack of systematic, effective and useful design methods, which can use a priori knowledge of the plant dynamics. To overcome these disadvantages of the conventional Fuzzy logic controller, different controller configurations of different structures and for self tuning of the fuzzy controller parameters have been proposed. Research is going on still to further improve the performance of the fuzzy logic controllers.

In literature, various structures for fuzzy PID (including PI and PD) controllers and fuzzy non-PID controllers have been proposed. Fuzzy PI control is known to be more practical than fuzzy PD because it is difficult for the fuzzy PD to remove steady state error. The fuzzy PI control, however, is known to give poor performance in transient response for higher order processes due to the internal integration operation. Thus, in practice the fuzzy PI controllers are more useful. To obtain proportional, integral and derivative control action altogether, the authors have combined, PI and PD actions together to form a Fuzzy PID (FPID) controller [2]. The construction of an FPID controller is achieved by summing the fuzzy PD controller output and its integrated part. This configuration resulted in a normal FPID controller. The performance of the current configuration has been further improved by adjusting the scaling factors that correspond to the derivative and integral coefficients of the fuzzy PID controller using a fuzzy inference mechanism in an on-line manner [2]. This configuration can be called as Relative rate observer based Self-tuning FPID controller. A similar self tuning controller scheme is presented [3] in which configuration, the output gain factor of the FLC is undergone tuning depending on the operating point of the system, hence generating the appropriate control signal.

Normally when designing an FLC, different values of gains and scaling factors are set by the operator. The FLC is also expected to give a better performance by allotting the values of gains by some optimizing methods. Genetic algorithm has been used successfully to solve the latter's purpose [12]. Another way of approach to improve the performance of a fuzzy PID controller by using some complex and more efficient fuzzy reasoning methods can be considered. Most FLCs are based on the simplified fuzzy reasoning, which loses much of the original fuzzy characteristics and therefore usually affect the robustness. A proper integration of the fuzzy reasoning method and its outer control structure is obviously crucial for achieving optimal control performance. A robust performance wise improved FLC can be achieved by incorporating the optimal fuzzy reasoning into the well-developed FPID type of control framework [4]. The performance comparison of the FPID controller was done by using four types of fuzzy reasoning methods. The FRM- optimal fuzzy reasoning mechanism proposed exhibited good response in terms of robustness.

It has been observed in the literature that the PI controllers attain the setpoint speed at the steady state, eliminating the offset occurring in a normal proportional controller. But the disadvantages of the PI controller are the sluggish response and the occurrence of overshoot, which may not be desirable in some applications. As discussed above, the fuzzy logic controller apart from its advantages of fast dynamic response and the fair operation in the presence of non linearities, has disadvantages like exhibiting offset in the response, inability to react to change in operating conditions. Moreover there is no systematic procedure for the development of a FLC. Hence there is need for controllers that depend completely neither on the fuzzy controller nor on a conventional PI type of controller, but could use the combination of these two controllers to exploit the advantages of both the controllers. For this purpose, a controller can be proposed which is a hybrid of the FLC and the conventional PI controller. In addition to being able to adapt automatically to a new environment, this controller can further simplify the task of developing rules, for the designer only needs to come up with an initial set of rules which are roughly correct. The burden of manually fine tuning the rules is thus removed from the designers. If the output of a speed controller is a combination of outputs of two speed controllers (FL and PI), combined together as a weighted sum to eliminate certain disadvantages then the resulting controller is referred to as a hybrid controller.

A series hybrid combination of the Fuzzy logic controller and a conventional PI controller has been proposed in [1]. Here the fuzzy controller processes the original speed error and provides a modified reference signal to the PI controller and the main control action is taken by the PI. This process of modifying the reference signal continuously is called the precompensation. The principle advantage in implementing this scheme of control is that, an existing PID controller can be easily modified in to this configuration just by adding a fuzzy precompensator in series with the PID controller. The described controller configuration is successfully implementable with the electric drives, and efficiently compensates for the overshoots and undershoots [5][6].

As mentioned in the literature, even though a fuzzy logic controller delivers fast response and functions well even in the presence of a nonlinearity, a PI controller is always preferred to be functioning in the front end, supported by the FLC at the back end. Moreover the designing of FLC requires time, experience and skills of the designer for the tedious fuzzy tuning exercise. It is well known that a conventional PI controller is most widely used in industry due to its simple control structure, ease of design and low cost. However, the PI type controller only cannot give a good control performance. Moreover, it suffers from disadvantages of slower response, larger overshoots, and oscillation. As the PMBLDC machine has nonlinear model, the linear PI control is not a good option. The main disadvantage of the constant gain PI controller when operating with systems having variation of operating conditions is that a frequent tuning of the gains is required as per the conditions. This task is very tedious and complex. This task of tuning the PI gains can be accomplished by a fuzzy logic controller in parallel. Based on the error value, and the change in error, the FLC outputs a value used in computing the Proportional, Integral and derivative gains at that particular operating condition [7].

Emerging intelligent techniques have been developed and extensively used to improve or to replace conventional control techniques because these techniques do not require a precise model. Also the results from the comparison of conventional and fuzzy logic control techniques in the form of an FL controller and fuzzy precompensator have shown that the fuzzy logic can reduce the effects of nonlinearity in a PMBLDC motor and improve the performance of a controller. As mentioned in the previous part of this section, a fuzzy logic controller has been implemented on many platforms such as digital signal processer, or an off-the shelf microcontroller. These platforms have different advantages and disadvantages. The FLC developed on DSP or PC can quickly process fuzzy computation to generate designed control action, but the physical size of the system may become too big and quite expensive for a small motor application. On the other hand, using an off-the shelf microcontroller to implement a FLC is inexpensive and the physical size of the system is small.

It can be observed that, when the FLC is employed on the control of a drive, the cost and complexity of control are more. Instead of using an FLC, the scope for efficiently varying the PI gains without an FLC has been explored, thus decreasing the complexity of control and making the drive more economical. The gain scheduling control scheme for a proportional integral controller (PI) for speed control of permanent magnet brushless dc (PMBLDC) motor drive has been proposed [8]. In this proposed scheme, the PI gains are allowed to vary within a pre- determined range, by varying the gain values as the functions of speed error. Low cost practical implementation of the procedure is possible without employing expensive dedicated computing systems. This scheme is very easy to implement in practice since an existing PI controller is tuned automatically. But a similar control technique in which the PI controller gains can be varied, based upon the decisions from a fuzzy logic controller is worth exploring, for its fast response and functionality under conditions of nonlinearities. With the availability of compact, high power computing equipments, this proposed hybrid controller scheme would not be difficult to implement.

Hybrid feedback controller for linear and nonlinear control systems provides maximal flexibility for achieving multiple performance objectives; is consistent with computer based implementations. One of the key decisions in construction of any hybrid controller is the decision regarding the family of allowable feedback functions on which the hybrid controller is based. It can be mentioned that hybrid control or logic based switching control has been extensively utilized in practical engineering control systems. At present, as important advances are being made in the theoretical aspects of Hybrid control design, it is hoped that this advances can begin to influence the practice of hybrid control engineering and can also provide new concepts for treating previously intractable control problems. Fuzzy control provides a formal methodology for representing, manipulating and implementing human's heuristic knowledge to control a system. Fuzzy control system has good robustness which can restrain influence of disturbance and fluctuation of parameter effectively. For systems with severe nonlinearities, a fuzzy logic controller can outperform a conventional PI, PID controller.

2.3 Conclusion

The exhaustive literature review has revealed that research work carried out on different controller configurations and on speed control of Brushless DC motors and other motor drives is influenced most by the advancements and developments in power electronics, microelectronics, different Simulation software and other sensor technologies. Almost all the developments covered in the section aim in the direction of increasing the robustness of the system for different operating conditions. The control hardware reduction was also the main criteria. Therefore these developments improve the motor speed control to a stage where the motor can be used extensively on various applications. Some methods are found which can be implemented with minimum control hardware and with the lesser need for processing. The main motivation is to improve the performance of the control even in the presence of the unknown nonlinearities. Fuzzy logic methods can be used effectively to complement conventional control methods for improving performance and robustness, especially in the presence of severe and unknown nonlinearities. The hybrid controller configurations have helped to improve the performances of the controller, in terms of transient and steady state response. When the conventional controllers and the intelligent controller are used in a suitable configuration it is concluded that the disadvantages of both the controllers can be eliminated and a much better performance is achievable. Although the developed controllers can be implemented on PC, Digital Signal Processor etc., the implementation through PIC micro controller is more economical.

Chapter: 3 MODELING OF THE PMBLDCM DRIVE SYSTEM

3.1 General

The concept and types of various controllers under study, applications of the brushless dc motors and an exhaustive literature review have been covered in the preceding chapters. The present chapter deals with modeling and simulation of the drive system with the different controllers the PI, Fuzzy logic controller, series hybrid controller and the Self-tuning controller in the MATLAB/SIMULINK environment.

3.2 Modeling of the system

Each component of the drive system is modeled by a set of mathematical equations; such sets of equations when combined together represent the mathematical model of the complete system. The modeling of the different components of the drive system described as below.

3.2.1 The PMBLDC drive system

The Fig.3.1 describes the basic building blocks of the PMBLDCM drive. 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).

The speed of the motor is compared with its reference value and the speed error is processed in proportional — integral (PI) speed controller. The output of this controller is considered as the reference torque. A limit is put on the speed i_c^* controller output i_a depending on permissible maximum winding currents. The reference current generator block generates the three phase reference currents (i_a^*, i_b^*, i_c^*) using the limited peak current magnitude decided by the controller and the position sensor. The reference currents have the shape of quasi-square wave in phase with respective back emfs to develop constant unidirectional torque. The PWM 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 currents are compared with the reference currents and the switching commands are generated to drive the inverter devices.

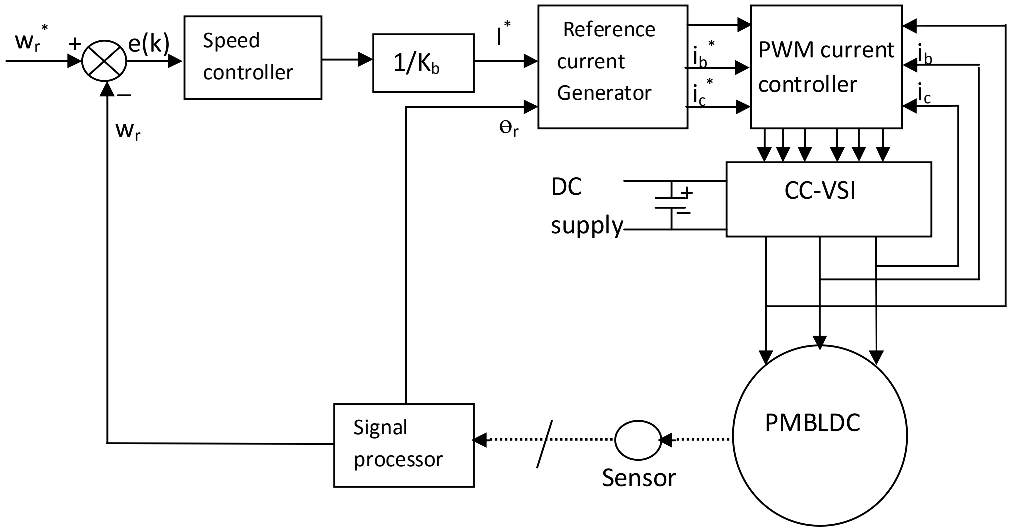


Fig. 3.1 The block diagram of the BMBLDCM drive

A. Reference Current Generator

The magnitude of the three phase current (I^*) is determined by using reference torque (T^*) and the back emf constant (K_b) as $I^* = T^* / K_b$. Depending on the rotor position signal obtained from the Hall sensors, the reference current generator block generates three-phase reference currents (i_a^*,i_b^*,i_c^*) by taking the value of reference current magnitude as I^*, -I^* and zero. The reference current generation is as shown below.

Rotor Position Signal	Reference Currents		
$\underline{\theta_r}$	$\underline{i_a}^*$	$\underline{i_b}^*$	$\underline{i_c}^*$
ɵ° - 60°	I^*	- I^*	0
60° - 120°	I*	0	- I^*
120° - 180°	0	I^*	- I^*
180° - 240°	- I^*	I^*	0
240° - 300°	- I^*	0	I^*
300° - 360°	0	-I^*	I^*

These reference currents are fed to the PWM current controller.

B. PWM Current Controller

The PWM current controller contributes to the generation of the switching signals for the Inverter devices. 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

Where, h_b is the hysteresis band around the three phase reference currents. The value of 'h_b' chosen here in simulation is 0.1A.

C. Modeling of PMBLDC Motor

The BLDCM produces a trapezoidal back electromotive force (EMF), and the applied current waveform is rectangular-shaped. In order to simplify analysis, we take one 3-phase 6-state BLDCM with Y-connected windings and two-phase excitation as the example. To the allowable extent, we make the following supposes: the three phase windings are symmetrical, magnetic saturation is neglected, hysteresis and eddy current losses is not considered, and the inherent resistance of each of the motor windings is R , the self-inductance is L, and the mutual inductance is M. Hence the three-phase stator voltage balance equation can be expressed by the following state equation.

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} + \begin{bmatrix} L-M & 0 & 0 \\ 0 & L-M & 0 \\ 0 & 0 & L-M \end{bmatrix} p \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} \tag{3.1}$$

Where, v_a, v_b and v_c are the phase voltage of three-phase windings, i_a, i_b and i_c are the phase current, e_a, e_b and e_c are the phase back EMF, and p is differential operator.

Based on the Eqn. (3.1), the equivalent circuit of motor can be obtained, as shown in Fig. 3.2. The electromagnetic torque of BLDCM is generated by the interaction of the current in stator windings and the magnetic field in rotor magnet. The electromagnetic torque equation is

$$T_e=1/\omega_r*[e_ai_a+e_bi_b+e_ci_c]= [4P_mN/\pi n]\ \phi_mI_d \qquad (3.2)$$

Where, P_m is pole numbers, N is total conductor numbers, ϕ_m is main magnetic flux, n is the motor speed, ω_r mechanical angular velocity of motor.

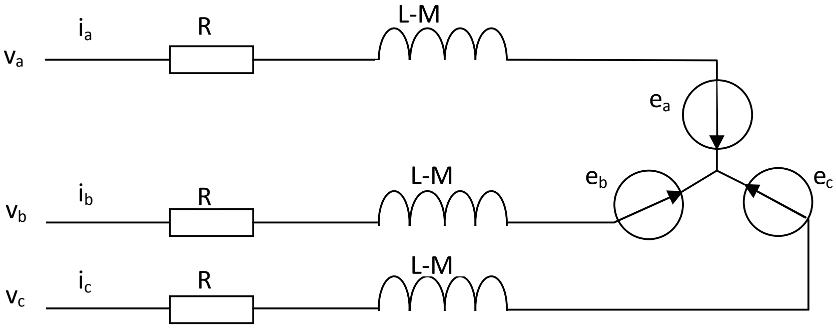


Fig. 3.2 The equivalent circuit of the PMBLDC motor

Eqn. (3.2) indicates that the developed torque of BLDCM is proportional to the magnetic flux and inverter input current, which is similar to that of a separately excited DC motor, where the developed torque is proportional to the armature current. Therefore, the torque of BLDCM will be controlled so long as the rectangle wave current amplitude is done. When inputting the three-phase rectangle wave current of 120° electrical angle and making it in phase with the EMF of each phase, the ripple of torque for BLDCM will be equal to zero.

The equation of motion can be expressed:

$$Te= Jd\omega/dt + B\omega + T_l \qquad (3.3)$$

Where, T_l is the load torque, J is the rotational inertia of rotor and load, B is the viscous damping coefficient. As it is difficult to model this motor and the inverter system, in their place, inbuilt blocks are directly taken from the SimPowerSystems library.

3.3.2 Speed Controllers

Four different types of speed controllers have been considered for the speed control of a PMBLDCM. As shown in the drive system explained above, the speed error e(n) is computed and used as an input to the speed controller. 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 machine.

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

$$e(n)= \omega_r^*(n)- \omega_r(n) \tag{3.4}$$

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

3.3.2.1 Proportional Integral (PI) controller

The figure shows the general schematic block diagram of the PI controller the output of the controller in discrete domain at the n^{th} instant is given as:

$$T^*(n) = T^*(n - 1) + K_p\{ e(n) – e(n - 1)\}+ K_i\, e(n) \tag{3.5}$$

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

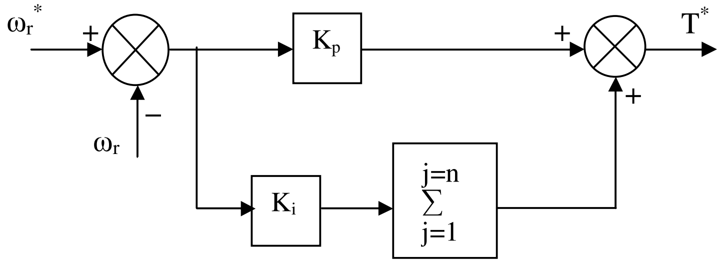


Fig. 3.3 The structure of PI controller

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

3.3.2.2 Fuzzy logic controller

The internal structure of the Fuzzy logic speed controller is as shown in the figure. The fuzzy controller is composed of the following four elements:

a. A Fuzzification interface, which converts controller inputs into information that the inference mechanism can easily use to activate and apply rules.

b. The rule-base (a set of If-Then rules), which contains a fuzzy logic quantification of the expert's linguistic description of how to achieve good control.

c. The Inference mechanism (also called an "inference engine" or "fuzzy inference" module),which emulates the expert's decision making in interpreting and applying knowledge about how best to control the plant.

d. A De-fuzzification interface, which converts the conclusions of the inference mechanism into actual inputs for the process.

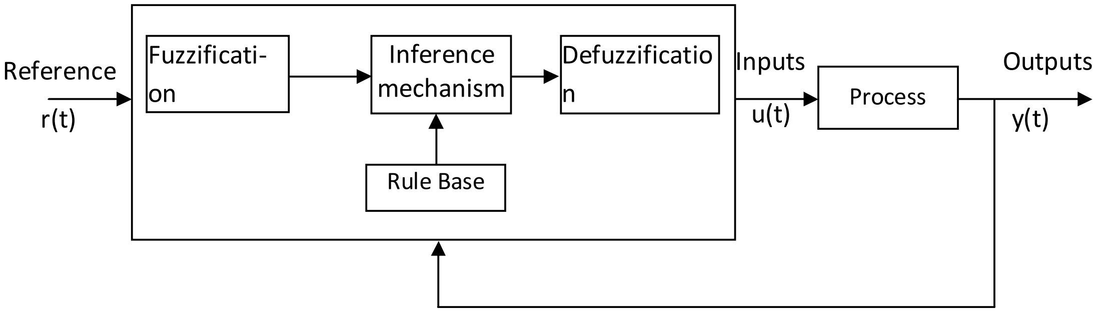


Fig. 3.4 Internal block diagram of a Fuzzy logic controller

Generally the procedure for constructing a Fuzzy controller consists of the following mechanism.

(i) Choosing the fuzzy controller inputs and outputs: here the speed error 'E' and the change in speed error 'CE' are selected as the input variables. The expected output from the controller is the reference torque (T^*).

(ii) Putting control knowledge into rule base: There will be "linguistic variables" that describe each of the time varying fuzzy controller inputs and outputs. Here, each input and the output variables are described using the variables {NH, NM, NL, ZE, PL, PM, PH}. Proper control rules are written using the variables in the "If-Then-Else" format. Hence the Rule-base is created.

(iii) Fuzzy Quantification of Knowledge: we use fuzzy logic to fully quantify the meaning of linguistic descriptions so that we may automate the control rules specified by the expert and by trial and error. Depending on the application and the designer, many different choices of membership functions are possible. Here we choose the triangular type of the membership functions and these are non symmetrical. The shapes are as shown in the Fig. 3.5,3.6..

(iv) Matching: Determining Which Rules to Use. The premises of all the rules are compared to the controller inputs to determine which rules apply to the current situation. Next the conclusions (what control actions to take) are determined using the rules that have been determined to apply at the current time.

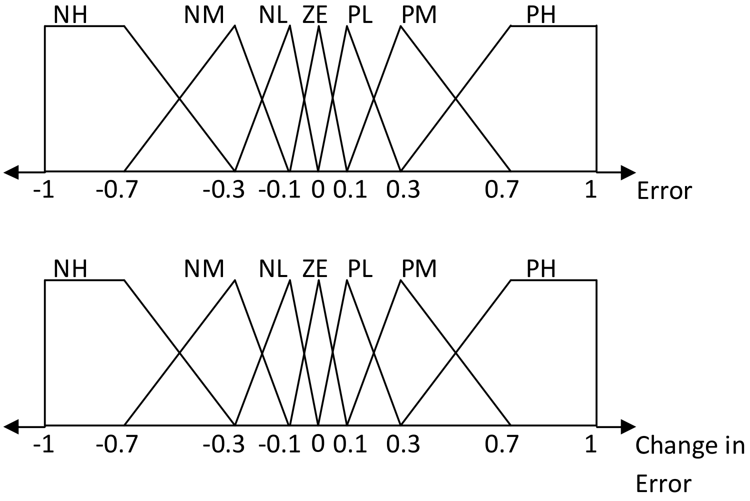


Fig. 3.5 Fuzzy membership functions for both the two input variables

The conclusions are characterized with a fuzzy set (or sets) that represent the certainty that the input to the plant should take on various values.

(v) Inference Step: Determining Conclusions. We considered how to determine which conclusions should be reached when the rules that are ON are applied to deciding what the value of the reference torque should be. To do this, we will first consider the recommendations of each rule independently. Then later we will combine all the recommendations from all the rules to determine the corresponding final value of the reference torque value.

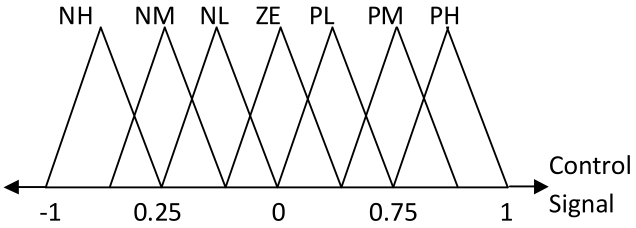


Fig. 3.6 Membership functions for the Output variable

(vi) Converting Decisions into Actions. Next, we consider the de-fuzzification operation, which is the final component of the fuzzy controller. De-fuzzification operates on the implied fuzzy sets produced by the inference mechanism and combines their effects to provide the "most certain" controller output (plant input).

Thus the necessary inputs are applied to these blocks by the rule based and the data based 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 base supplied by the data base. The Fuzzy rules are given in Table 3.1. The output of the Decision-maker passes through the De-fuzzifier where in the linguistic format signal is converted back into the numeric form or crisp form. The decision making block uses the rules in the format of "If-Then-Else".

E CE	NH	NM	NL	ZE	PL	PM	PH
NH	NH	NH	NH	NH	NH	NM	PM
NM	NH	NH	NH	NH	NM	PL	PH
NL	NH	NH	NH	NM	ZE	PM	PH
ZE	NH	NH	NH	ZE	PL	PH	PH
PL	NH	NH	ZE	PL	PM	PH	PH
PM	NH	NM	PM	PM	PH	PH	PH
PH	NM	PM	PH	PH	PH	PH	PH

Table 3.1 Rule table for Fuzzy logic controller

The entire fuzzy computation procedure can be briefed as follows.

(i) Calculation of the n^{th} instant values of the two input signals namely, speed error and rate of change in speed error.

(ii) Scaling of the two input signals namely, speed error and the change in speed error.

(iii) The scaled input signals are fed to the fuzzy logic controller.

(iv) The scaled crisp data is converted into linguistic format in accordance with the defined fuzzy sets.

(v) On accordance with the linguistic rules, value of the output signal is determined. The required rules and data are supplied by the rule base and the data base.

(vi) The linguistic output data is converted back into crisp output data by the application of the method of De-fuzzification as follows:

Given the combination of two inputs, the membership of the corresponding output is taken as minimum membership value of the two respective inputs.

Mathematically, $\alpha=\mathrm{Min}[\mu(\mathrm{input1}), \mu(\mathrm{input2})]$

Crisp value= $\{\Sigma\ (pm)\alpha\}/\Sigma\ \alpha$

Where μ refers to the membership value, the output membership is stored in α and 'pm' refers to the peak of membership function.

The crisp value obtained is rescaled back to get the controller output. The input membership functions are defined by taking into account the speed and the acceleration of the motor. The motor speed range is well converted into the seven membership functions-NH (Negative High), NM (Negative Medium), NL (Negative Low), ZE (Zero), PL (Positive Low), PM (Positive Medium), PH (Positive). The appropriate rules are given in the table. The rules are to be read as (NL-PL-ZE), "if error is NL and change in error is PL then the reference torque is ZE". They are written such that the rise time is low and to cater torque for the applied load torque.

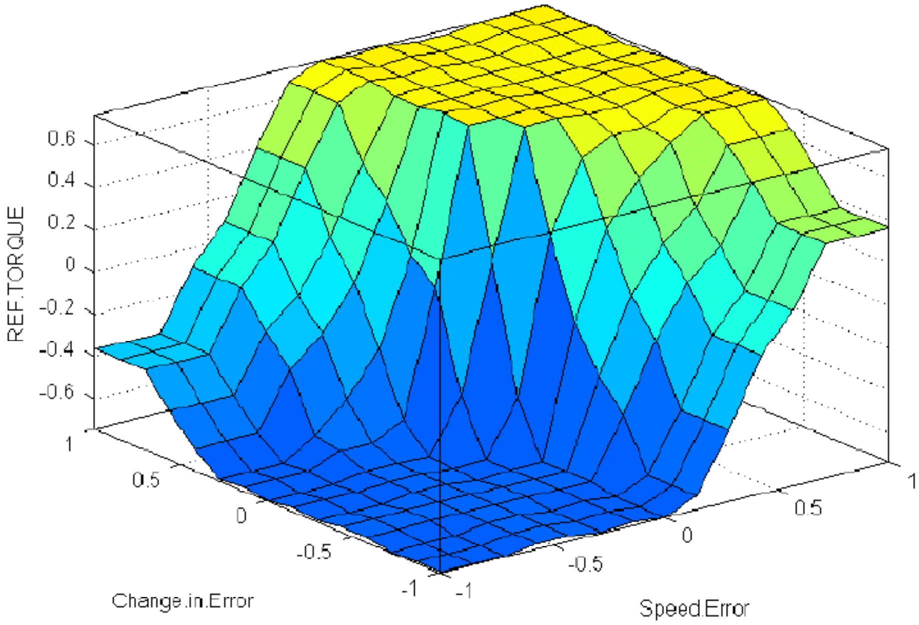


Fig. 3.7 The Fuzzy surface for the fuzzy logic controller

They have been defined by the understanding of the behavior of the system. Rules of different functionalities can be found here, rules for maintaining speed error zero (Steady state rules), rules that avoid motor speed overshoot and the rules that provide rapid response to large error resulting from command change. 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.

3.2.2.3 The Series Hybrid PI Controller (Fuzzy Precompensated PI controller)

The basic structure of the control is shown in Fig. 3.8. The purpose of the control scheme is based on trying to compensate for overshoots and undershoots in the transient response. Fuzzy logic control is generally opted when intelligence and fast dynamic response are among the prime requirements. The major disadvantage in using solely this type of control logic is the presence of steady state error on load. To eliminate this disadvantage, it is necessary to combine fuzzy logic with another suitable control technique, which is capable of removing the disadvantage existing in fuzzy logic control. Therefore a PI controller is used in combination with fuzzy logic such that at operating point, PI controller takes over eliminating the disadvantage of the FLC. Similarly when away from the operating point

FLC dominates and eliminates the error due to PI controller such as occurrence of overshoots and undershoots in drive response.

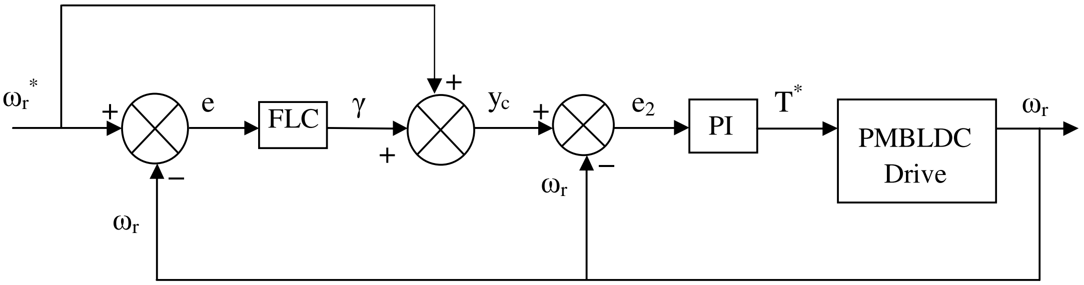


Fig. 3.8 Block diagram of the series hybrid controller

Such a controller where weighted combination of two controller outputs contributes to the net output is called a hybrid controller. The structure and the functionality of the controller is described in the section below.

A. Control Structure

The Fig. 3.8 illustrates the basic control structure of the Series Hybrid controller. The scheme consists of a conventional PI control structure together with our proposed fuzzy precompensator. The fuzzy precompensator uses the command input y_m and the plant output y_p to generate a precompensated command signal y_c, described by the following equations

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

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

$\gamma(n)= F[e(n),\Delta e(n)]$

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

In the above, e(n) is the tracking error between the command input $y_m(n)$ and the plant output $y_p(n)$, and $\Delta e(n)$ is the change in the tracking error. The term $F[e(n),\Delta e(n)]$ is a nonlinear mapping of e(n) and $\Delta e(n)$ based on fuzzy logic (described below). The term $\gamma(n)=F[e(n),\Delta e(n)]$ 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. The equations governing the PI controller are as follows.

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

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

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

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.

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 if a conventional PI control scheme is used. The precompensator uses fuzzy logic rules that are based on the above motivation.

B. Fuzzy Precompensator

We now describe the implementation of the fuzzy logic based term $\gamma(n) = F[e(n),\Delta e(n)]$. We think of e(n) and Δe(n) as inputs to the map F, and y(n) as the output. Associated with the map F is a collection of linguistic values, whose description is given in the precious section as Negative High, Negative medium, Negative Low, Zero, Positive Low, Positive Medium, Positive High.

L={NH, NM, NL, ZE, PL, PM, PH}

They represents the term set for the input and output variables of F. In our scheme, we use seven linguistic values. Associated with the term set L is a collection of membership functions

$\mu = \{ \mu_{NH}, \mu_{NM}, \mu_{NL}, \mu_{ZE}, \mu_{PL}, \mu_{PM}, \mu_{PH} \}$

Each membership function is a map from the real line to the interval [0, 1]. Fig. 3.6 shows a plot of the membership functions. As depicted in Figure, the membership functions we use are of the triangular type. The height of the membership functions in this case is one, which occurs at the points -0.7, -0.3, -0.1, 0, 0.1, 0.3, 0.7, respectively. The realization of the function F[e(n),Δe(n)], based on the standard fuzzy method, consists of three stages: Fuzzification, Decision-making logic, and De-fuzzification. We describe each of these stages in turn. The structures of each membership functions are shown in the figures for all the two input and the output variables.

E **CE**	**NH**	**NM**	**NL**	**ZE**	**PL**	**PM**	**PH**
NH	NH	NH	NH	NH	NM	NL	PM
NM	NH	NH	NH	NM	NL	PL	PH
NL	NH	NH	NM	NL	NL	PM	PH
ZE	NH	NM	NM	ZE	PL	PH	PH
PL	NM	NM	NL	PL	PM	PH	PH
PM	NM	NL	PL	PM	PH	PH	PH
PH	NM	PL	PH	PH	PH	PH	PH

Table 3.2 Rule table for the Fuzzy precompensator

***C. Fuzzifcation*:**

The process of fuzzification transforms the inputs e(n) and Δe(n) into the setting of linguistic values. This consists of scaling the inputs e(n) and Δe(n) appropriately and then converting them into fuzzy sets.

We use the symbol GE, for the scaling constant for the input e(n), and the symbol GCE, for the scaling constant for the input Δe(n).

D. Decision-Making process:

Associated with the decision making process is a set of fuzzy rules R = { R_1, R_2, . . . , &R_r}, where 'r' is the total number of rules. An example of a rule is the triplet (NH, PM, NH). Rules are often written in this form: "if e(n) is NS, and Δe(n) is PS, then γ is ZE," (here we think of γ as the output of the fuzzy logic rule). For example, in the rule represented by the triplet (PH, NH, PM), the idea of the rule is that "if e(n) is PH and Δe(n) is NH, then output PM". The set of rules used in our fuzzy precompensator is given in the Table 3.2, the fuzzy control surface is shown in Fig. 3.9. The rules are derived by using a combination of experience, trial and error, and our knowledge of the response of the system. To explain how these rules were obtained, consider for example the rule (ZE, NM, NH) in Table. Suppose that the command signal is a constant y_m, the error e(n) is zero, and the change of error Δe(n) is a negative number.

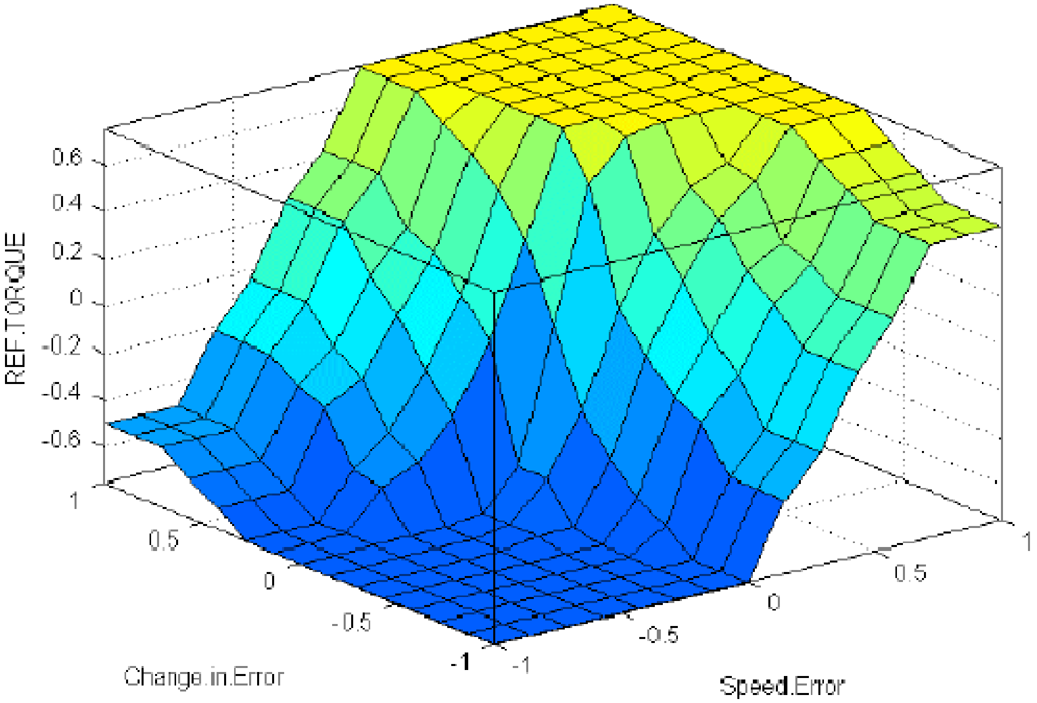


Fig. 3.9 Fuzzy surface for the precompensated controller

This means that the output $y_p(n)=y_m(n)-e(n)$ is increasing, i.e., heading in the direction of an overshoot. To compensate for this, we decrease the command signal. This corresponds to applying a correction term y(k) that is negative. Hence, we get the rule "if error is Zero and change-of-error is NM, then output a NH correction term." Similarly, consider the rule (PM, PM, PH) in Table. Correspondingly, consider the case where e(n) is positive, and so is Δe(n). This means that the plant output $y_p(n)$ is below the command signal, and is still decreasing (i.e., we are in the middle of an undershoot). This explains the control structure and functioning of the series hybrid controller. To compensate for this, we need to increase the command signal by a positive amount. This corresponds to applying a positive value of y(k). Hence the rule "if error is Positive Medium and change-of-error is Positive Medium, then output a Positive High correction." The other rules are obtained in a similar manner.

E. De-fuzzification:

The de-fuzzification process maps the result of the fuzzy logic rule stage to a real number output $F[e(n),\Delta e(n)]$. We use the Centroid de-fuzzification method. The output of the fuzzy controller is multiplied by a scaling factor GP to get the final control signal $y_p(n)$. The resultant Fuzzy Rule surface is as shown in the figure for the present set of rules. It can be observed that the function F is smooth.

3.2.2.4 The Self-tuning Controller

Fuzzy logic-based self tuning scheme for the conventional PI controllers uses fuzzy computing along with conventional control methods for enhancement of the drives performance. The Fuzzy tuned PI controller is expected to reduce the rise time and the settling time and also reduce the overshoot which generally occurs in a conventional PI controller. The structure is easy to understand and is capable of accommodating without much change in the actual system. It works on the same basic principle of a conventional PI controller, but unlike the fixed gain PI controller, in this controller, the values of the proportional and the integral gains are modified continuously based upon the operating condition. We know that as per the control structure of a normal PI controller in continuous time domain, the control action, $u(t)=K_pe(t) + K_i\int e(t)dt$, in the proportional term, control action is proportional to the “product of proportional gain K_p and error value” and in the integral term, it is proportional to “the product of the integral gain K_i and integral of the error. That means 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 i.e., when the system has offset.

Hence the control method follows that when the speed error is large, the proportional gain must be kept large and when the operating point is near the set point; the integral gain comes to action and reaches the maximum after reaching the steady state value. Fuzzy logic rules are written as per this control strategy such that the proportional gain (K_p) must be maximum when the error is large and should be started varying to the minimum when the drive system is near the set point. The integral gain is varied such that its value will be minimum, when the drive operates away from the set point and attains maximum value when it operates near to the set point.

A. Control structure

The Fig. 3.10 illustrates the basic control structure of the proposed self tuning controller.

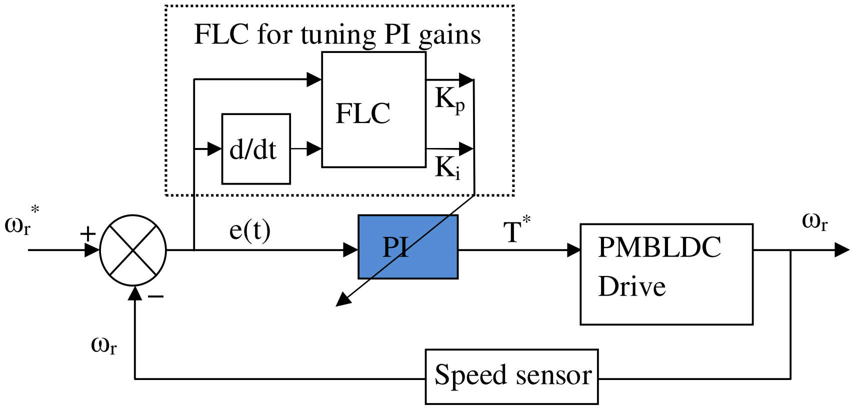


Fig. 3.10 the basic structure of the Self tuning controller

It consists of a fuzzy logic controller in parallel with a conventional PI controller. The fuzzy logic controller uses the tracking error (e(n)) between the command speed and the present rotor speed and the change in the speed error(Δe(n)) as the inputs.

$e(n)=\omega_r^*(n)-\omega_r(n)$, $\Delta e(n)=e(n)-e(n-1)$

These values are multiplied by suitable gain constants and are fed to the FLC. The FLC computes the gives corresponding values a(n) and b(n) as outputs based on the defined fuzzy rules which can be shown as

$a(n)=F_1[e(n), \Delta e(n)]$ and $b(n)=F_2[e(n), \Delta e(n)]$

These outputs of the FLC are multiplied by the corresponding scaling factors to get the appropriate values of the proportional and the integral gains denoted by K_p and K_i. These calculated gain values are supplied to the PI controller. The control torque T^* can be calculated from these values in the PI controller in discrete domain as:

$T^*=T^*(n-1)+ K_p\Delta e(n)+K_ie(n)$, where Δe(n) is the change of error.

The purpose of the fuzzy self tuning controller is to modify the values of the proportional and the integral gains depending on the error and the rate of change in error such that the rise time, the overshoot, the settling time are reduced and the effects due to unknown non linearities are eliminated, finally the controller must achieve an adaptive nature to load variations.

B. The Fuzzy controller for self tuning

The implementation of the fuzzy logic terms $a(n)=F_1[e(n), \Delta e(n)]$ and $b(n)=F_2[e(n),\Delta e(n)]$ are discussed in this section. We consider e(n) and Δe(n) as the inputs to the fuzzy logic controller and a(n) and b(n) as the outputs. Associated with map F1 and F2 is a collection of linguistic values NH- Negative High, NM- Negative Medium, NL-negative Low, ZE-Zero, PL-Positive Low, PM- Positive Medium, PH- Positive High.

X= {NH, NM, NL, ZE, PL, PM, PH}

Represents the term set for the input variables of F_1 and F_2. The set

Y= {VLOW, LOW, BMED, MED, AMED, HIG, VHIG}

Represents the term set for both the output variables a(n) and b(n). Where the terms are described as VLOW- Very Low, LOW- Low, BMED- Below Medium, MED- Medium, AMED- Above Medium, HIG-High, VHIG- Very High.

Associated with the term sets X and Y are the collection of Membership functions

$\mu_1=\{ \mu_{NH} , \mu_{NL} , \mu_{NM} , \mu_{ZE} , \mu_{PL} , \mu_{PM} , \mu_{PH} \}$

$\mu_2=\{ \mu_{VLOW} , \mu_{LOW} , \mu_{BMED} , \mu_{MED} , \mu_{AMED} , \mu_{HIG} , \mu_{VHIG} \}$

Each membership function is a map from real line to the interval [0, 1]. As depicted in the figure, the membership functions used are triangular in shape. Height of these membership functions is triangular in shape and has the maximum values at the points -0.7, -0.3, -0.1, 0, 0.1, 0.3, 0.7. The computation of the fuzzy functions $F_1[e(n), \Delta e(n)]$ and $F_2[e(n),\Delta e(n)]$, based on the standard fuzzy methods consists of three stages: Fuzzification, decision making logic and the defuzzification. These stages are described below

a. Fuzzification

The process of fuzzification transforms the inputs e(n) and Δe(n) into the setting of linguistic values. This consists of scaling of the input variables appropriately and then converting into fuzzy sets. The variables e(n) and Δe(n) are multiplied by scaling factors GE and GCE respectively.

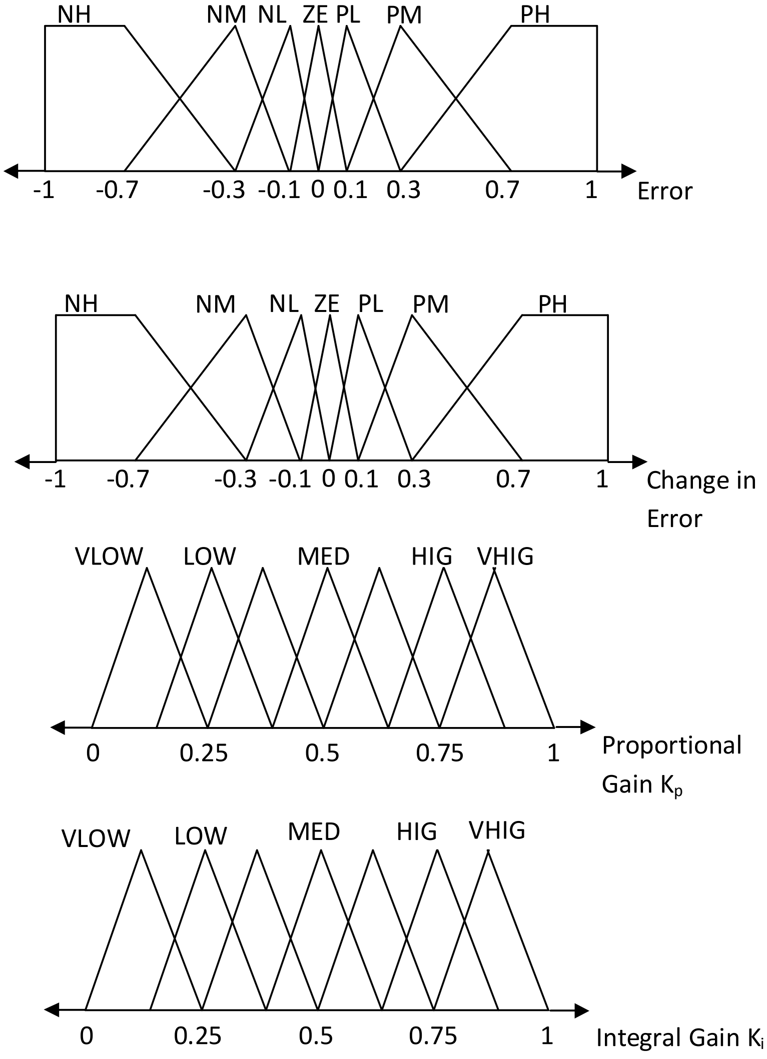


Fig. 3.11 The Membership Functions for the Input and the output variables

b. Decision making process

The sets of fuzzy rules $R_a=\{R_{1a},R_{2a},\ldots R_{49a}\}$ and $R_b=\{ R_{1b},R_{2b},\ldots R_{49b} \}$ are associated with the decision making process, for computing the Proportional and the Integral gain values respectively. The two sets of rules are given in the Table 3.3 and 3.4, the fuzzy control surfaces for the respective fuzzy variables are given in Fig. 3.12, 3.13. The rule structure for a(n) is in the form (PH,NH,VHIG,VLOW) which implies that “if the error(e(n)) is PH and the change in error(Δe(n)) is NH then proportional gain a(n) is VHIG and integral gain

E CE	NH	NM	NL	ZE	PL	PM	PH
NH	VHIG	VHIG	HIG	MED	MED	AMED	VHIG
NM	VHIG	VHIG	AMED	BMED	MED	HIG	VHIG
NL	VHIG	VHIG	AMED	LOW	MED	HIG	VHIG
ZE	VHIG	VHIG	AMED	VLOW	AMED	VHIG	VHIG
PL	VHIG	HIG	MED	LOW	AMED	VHIG	VHIG
PM	VHIG	HIG	MED	BMED	AMED	VHIG	VHIG
PH	VHIG	AMED	MED	MED	HIG	VHIG	VHIG

Table 3.3 Rule table for Proportional gain tuning

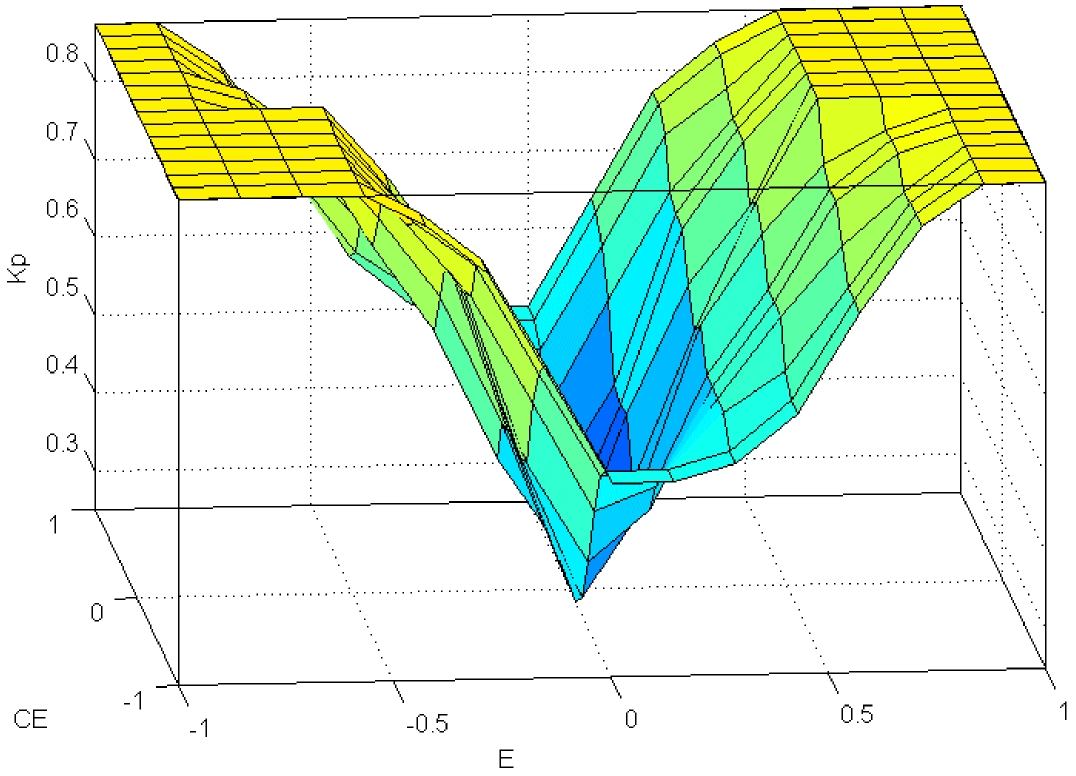


Fig. 3.12 Fuzzy surface for Proportional gain tuning (K_p)

E CE	NH	NM	NL	ZE	PL	PM	PH
NH	VLOW	VLOW	LOW	AMED	AMED	BMED	VLOW
NM	VLOW	VLOW	BMED	AMED	MED	LOW	VLOW
NL	VLOW	VLOW	BMED	HIG	MED	LOW	VLOW
ZE	VLOW	VLOW	BMED	VHIG	BMED	VLOW	VLOW
PL	VLOW	LOW	MED	HIG	BMED	VLOW	VLOW
PM	VLOW	LOW	MED	AMED	BMED	VLOW	VLOW
PH	VLOW	BMED	AMED	AMED	BMED	VLOW	VLOW

Table 3.4 rule table for the proportional gain tuning

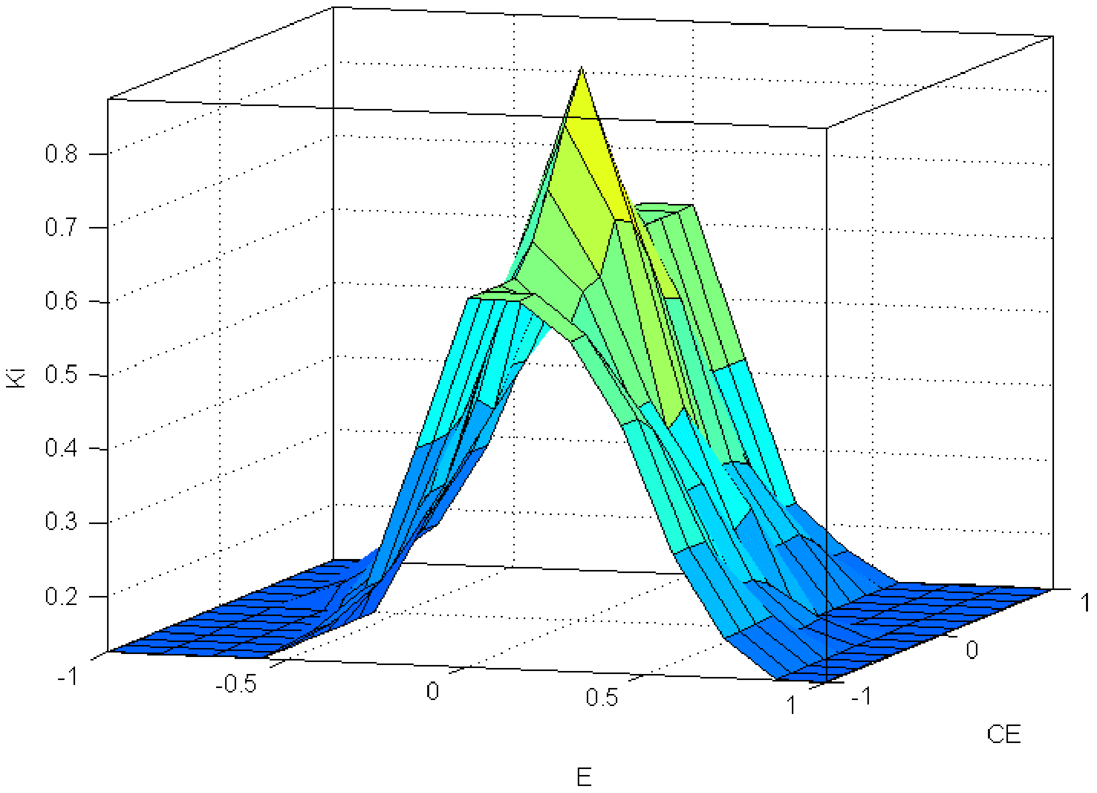


Fig. 3.13 Fuzzy surface for tuning the Integral gain (K_i)

b(n) is VLOW ". There are two sets of rules consisting of 49 rules in each. 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, VLOW, VHIG), which says "if e(n) is Zero and Δe(n) is Zero this shows that the system is operating at the set point so a(n) must be very low and b(n) must be very high" because, the integral gain is responsible to maintain zero steady state error i.e. the stiffness at the set point. Similarly in the rule (ZE, PH, MED, AMED) which says "if e(n) is Zero and Δe(n) is Positive High which means the drive is operating at set point and suddenly started Moving away from set point due to drop in speed, to compensate for this, the proportional gain a(n) is increased to Medium and integral gain b(n) is reduced to Above Medium.

c. Defuzzification

The de-fuzzification process maps the result of the fuzzy logic rule stage to a real number output F_1[e(n),Δe(n)] and F_2[e(n),Δe(n)]. We use the Centroid de-fuzzification method. The outputs of the fuzzy controller a(n) and b(n) are multiplied by a scaling factor GA and GB respectively to get the final gain values K_p and K_i. These gains are directly used by the PI controller. Hence the control strategy of the designed self tuning controller is implemented. The figures show fuzzy surfaces for the proportional and the integral gains. It can be observe that the proportional gain has its maximum values when the error is high and decrease as the error decreases and finally reaches the minimum point near the set point. The integral gain increases from minimum to maximum with the decrease of the error and as the system approaches the set point speed.

3.3 Modeling using Simulink

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

3.3.1 Simulink models of the 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(Fuzzy pre-compensated) controller and the Self tuning PI controller.

The Fig. 3.14 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.

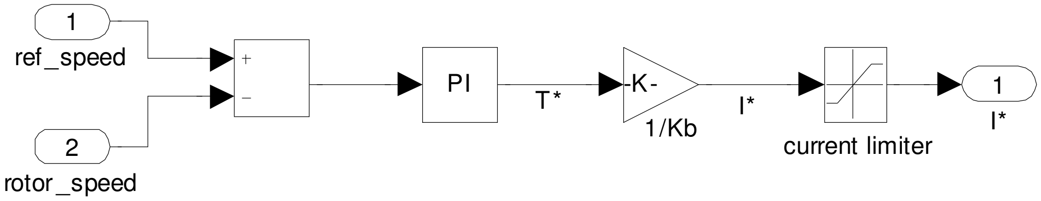


Fig. 3.14 Simulink model for a PI controller

Fig. 3.15 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.

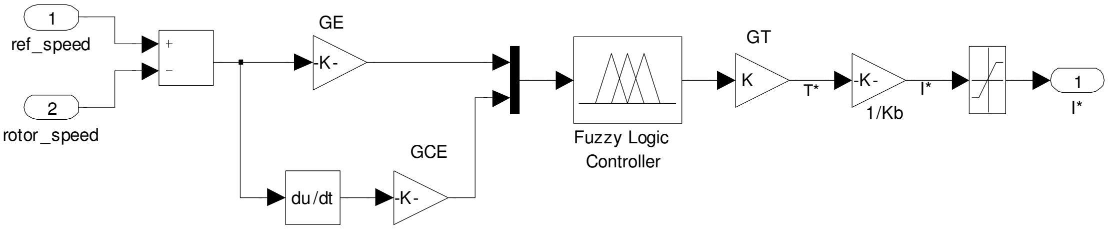
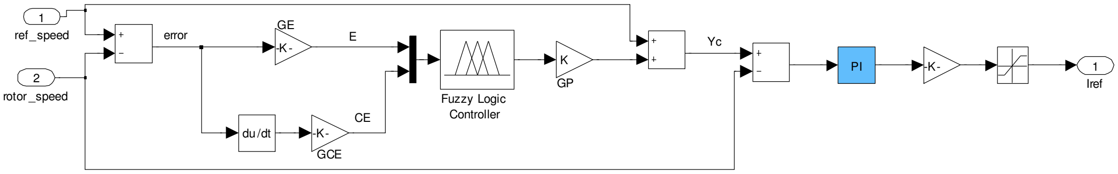


Fig. 3.15 Simulink model for the Fuzzy controller.

Fig. 3.16 below shows the MATLAB model diagram for the Series hybrid controller. Such a controller has the modified reference speed (precompensated) signal by the FLC to the PI controller.



3.16 Simulink model for the series hybrid controller

The PI controller produces the required control signal. The controller's operation has been discussed in the previous sections.

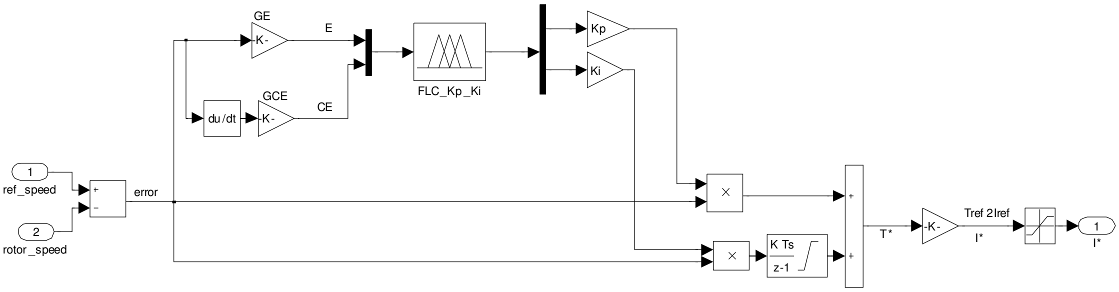


Fig. 3.17 Simulink block for the self tuning PI controller

The Fig 3.17 shows the MATLAB model diagram for the Self tuning 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 scontroller to generate the required control Torque (T^*) signal.

3.3.2 The PMBLDC Motor

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

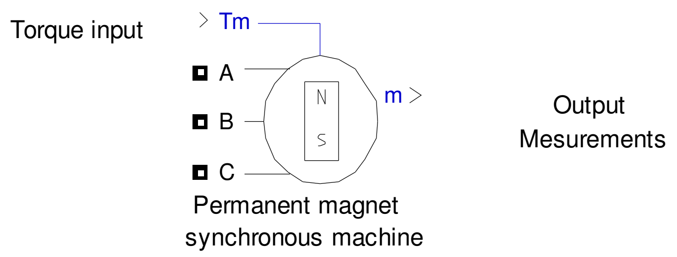


Fig. 3.18 Simulink library block of the PMSM of Trapezoidal back emf type

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 parameters **[19]** given in the appendix. The complete Simulink model of the drive is shown in Fig. 3.19.

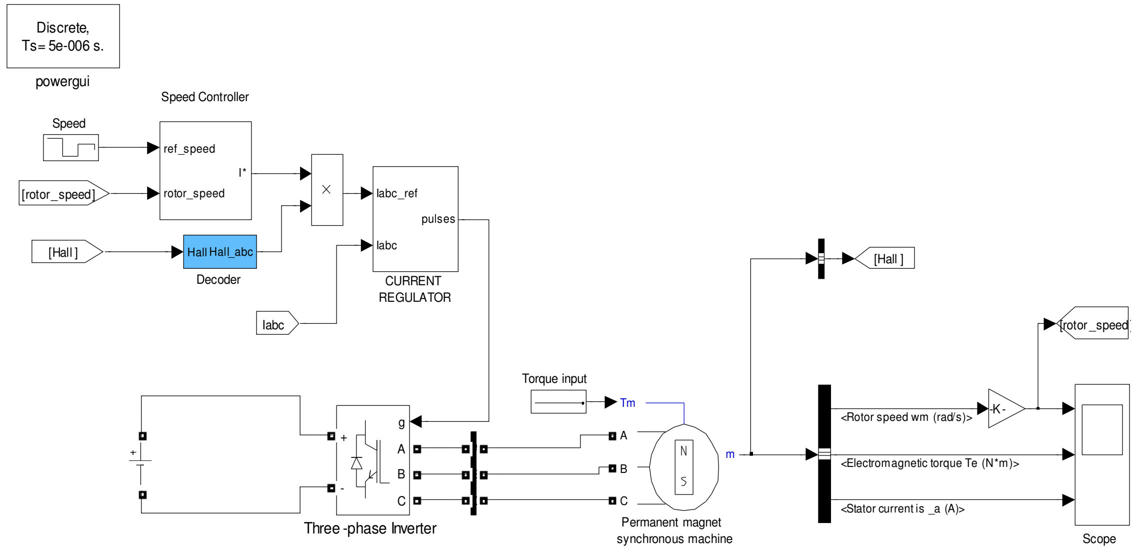


Fig. 3.19 The complete Simulink model of the drive

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. In the succeeding chapter, the drive is simulated for the different speed controller at different operating conditions.

3.4 Conclusion

The detailed modeling, analysis, design and simulation of the PI controller, the fuzzy logic controller, the series hybrid PI controller and the self tuning 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: 4 SIMULATION RESULTS AND DISCUSSIONS

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_s=5μsec. The drive performance is evaluated, separately using the four speed controllers presented in previous chapters, for different operating conditions. The results of the 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

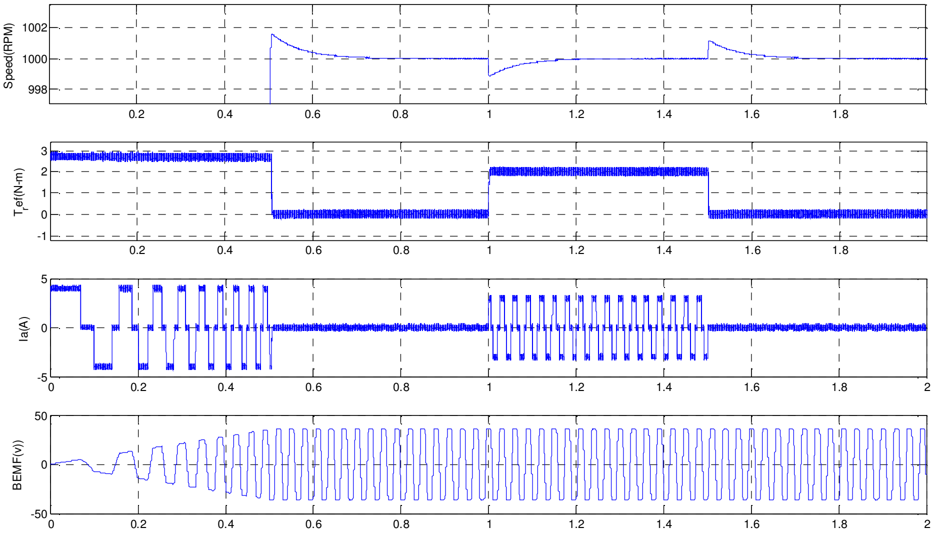


Fig. 4.1 Response of the drive with the PI controller on stating and load perturbation

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 Fig. 4.1 rotor speed is presented in revolutions per minute (RPM), the electromagnetic torque (Te) 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 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 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

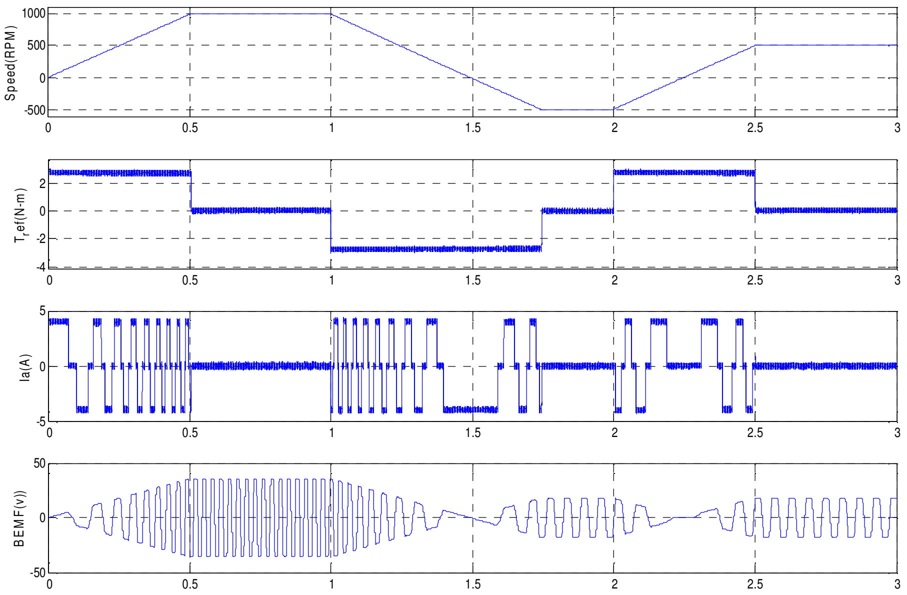


Fig. 4.2 Response of drive with PI controller on reversal of speed direction.

to start normally with set point speed of 1000 RPM; at the time instant t=1sec, the set point speed is changed to -1000 RPM.

The magnitude of the overshoot and the time taken to settle back to normal value is observed keenly. The Fig. 4.2 show the plots for Rotor speed (RPM), the torque (Te) 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 for during 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.

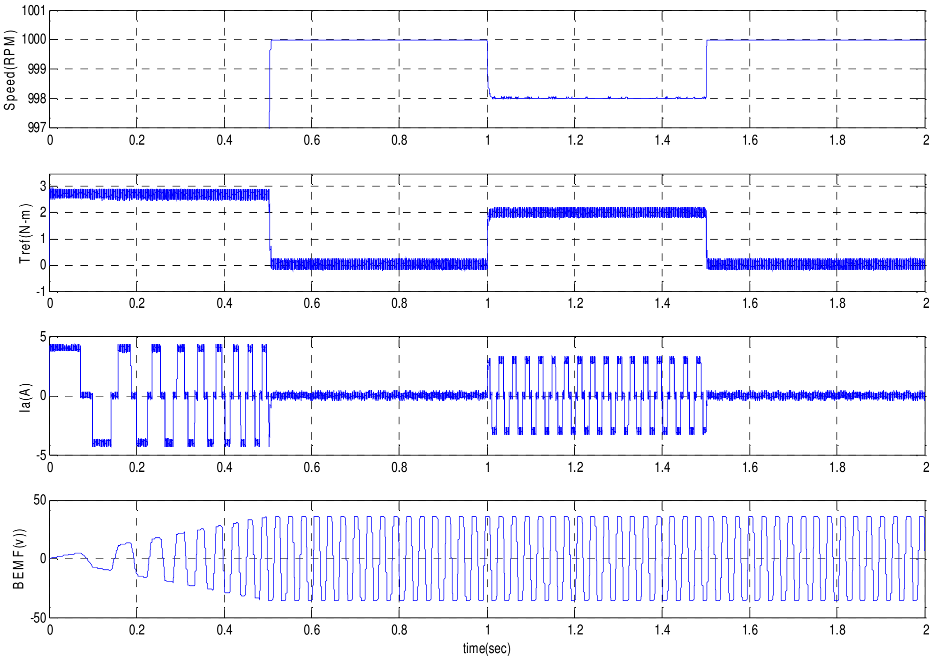


Fig. 4.3 Response of drive with Fuzzy logic controller on starting and load perturbation

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. 4.3 shows the plots for Rotor speed in revolutions per minute (RPM), the electromagnetic torque (Te) developed by the motor in (N-m), stator current (i_a) of phase a in Ampere, the back emf developed in phase a 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 setpoint speed. When load is applied at t=1sec, a dip in speed of 2RPM from the setpoint is observed, moreover the response could not reach the setpoint; 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

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.

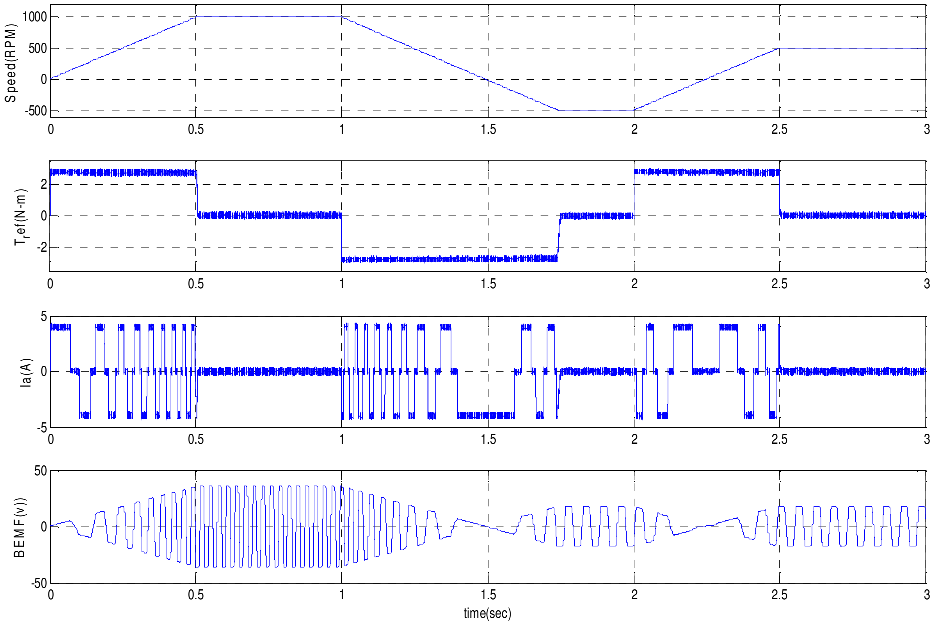


Fig 4.4 Response of drive with Fuzzy logic controller on reversal of speed direction

The magnitude of the overshoot and the time taken to settle back to normal value is observed keenly. The Fig. 4.4 shows the plots for Rotor speed (RPM), the torque (Te) 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 -499.97RPM 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 (fuzzy precompensated) speed controller

4.4.1 Response of the drive during starting and Load perturbation

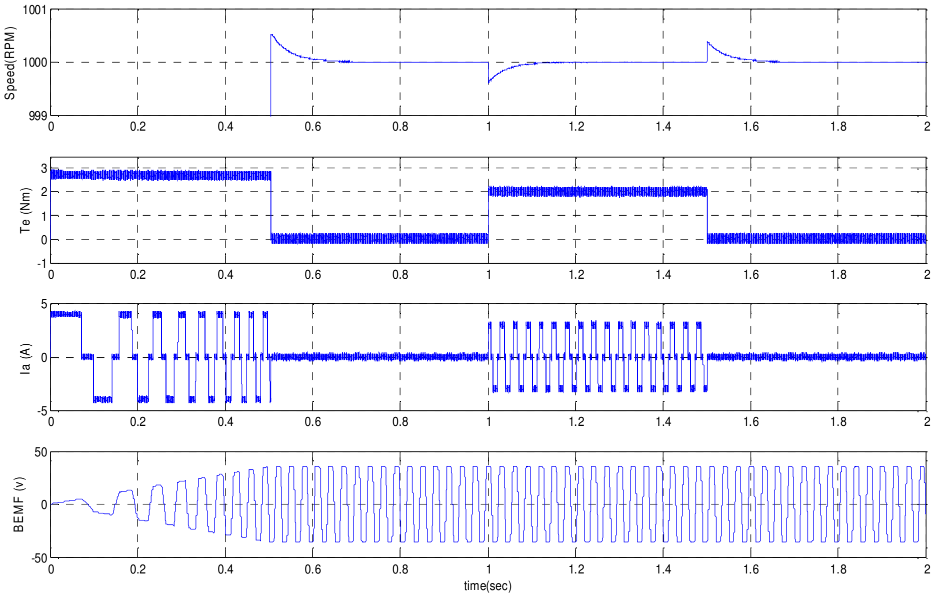


Fig. 4.5 Response of drive with series hybrid controller on starting and load perturbation

The Fig. 4.5 shows the response of the PMBLDC drive on starting at a set point speed of 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 Fig. 4.5 shows the response of the drive on starting at a setpoint 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

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.

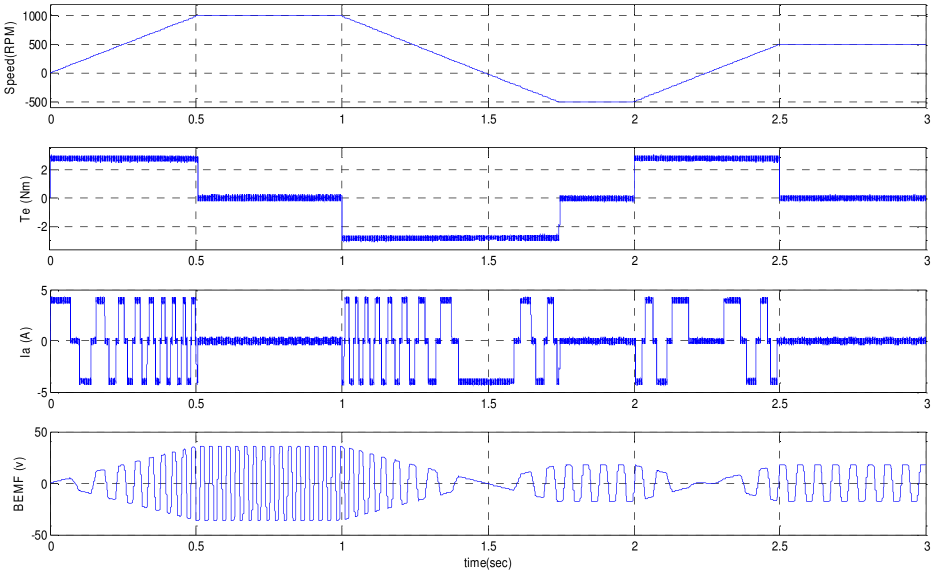


Fig. 4.6 Response of drive with the series hybrid controller on reversal of speed direction

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 -1000 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 (Te) 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 0.49RPM, and finally settles at the time instant 1.90sec. Fig. 4.6 show the plots for Rotor speed (RPM), the torque (Te) 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 0.49RPM, 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 the Self tuning PI speed controller

4.5.1 Response of the drive during starting and Load perturbation

The Fig. 4.7 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 Fig. 4.7 shows the plots for Rotor speed (RPM), torque developed (N-m), stator current (Ampere), the back emf (Volts).

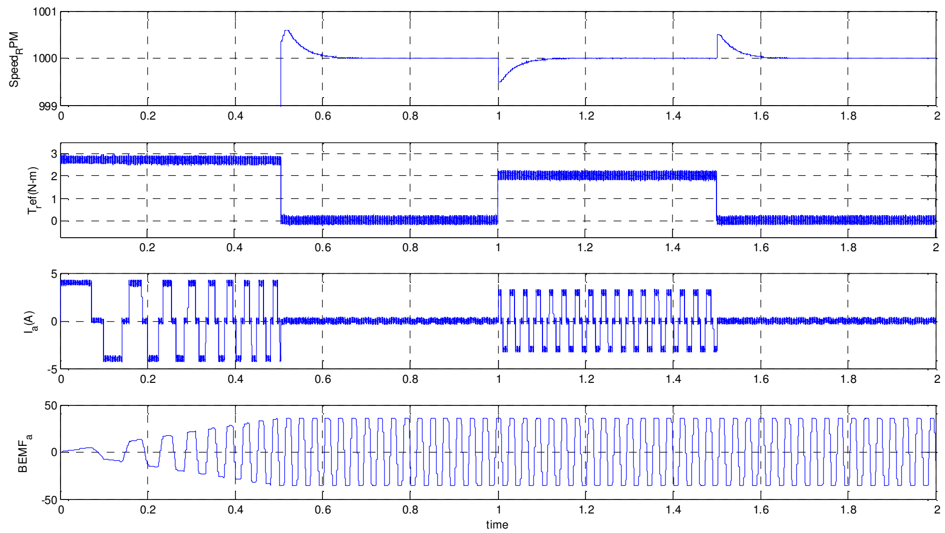


Fig. 4.7 Response of drive with the self tuning controller on starting and load perturbation

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 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

The Fig. 4.8 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 -1000 RPM i.e. it is made to rotate in the reverse direction.

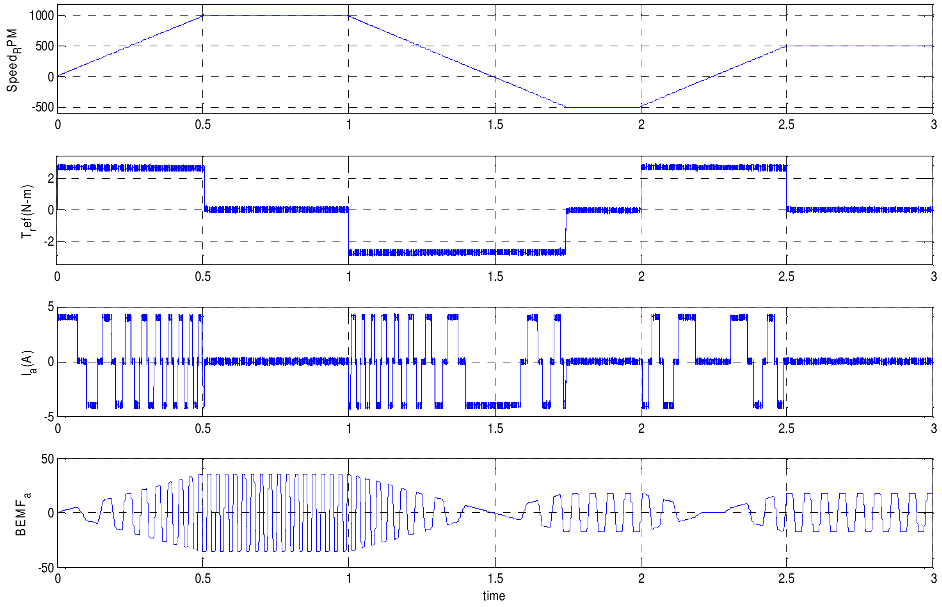
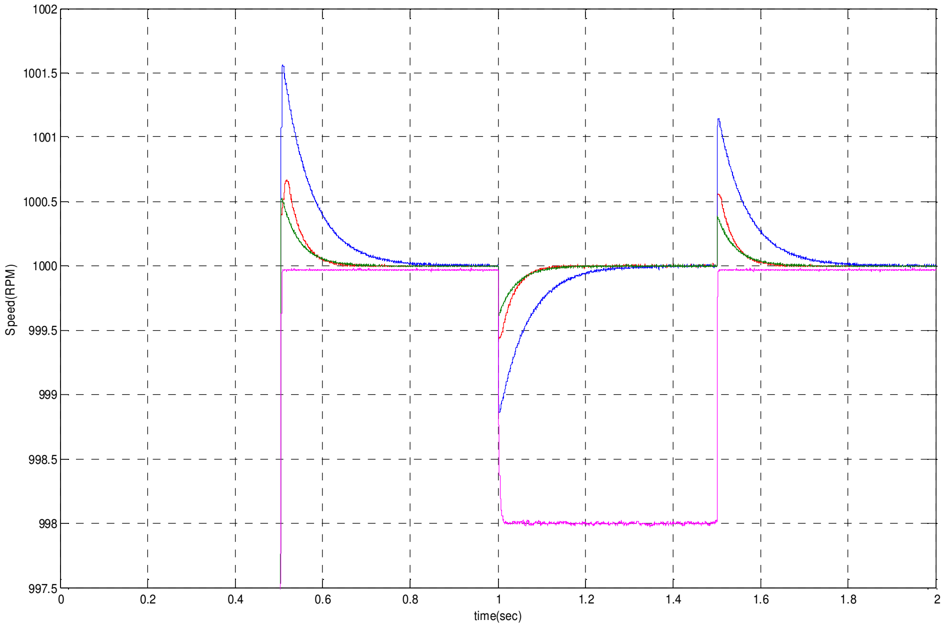


Fig. 4.8 Response of drive with the self tuning controller on reversal of speed direction.

The Fig. 4.8 shows the plots for Rotor speed (RPM), the torque (Te) 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 0.6RPM, 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 0f 0.57RPM, finally it settles at 2.625sec.

4.6 Discussion on results

The Fig. 4.9 below shows the response of the drive on starting and load perturbation for a setpoint speed of 1000RPM while using the controllers PI, Fuzzy Logic, Series Hybrid and Self tuning PI controllers. The speeds are shown at the



____Fuzzy logic controller ____Conventional PI controller

____Self tuning PI controller ____Series hybrid controller

Fig. 4.9 Response of drive on starting and load perturbation for all the controllers at 1000RPM

set point. The responses of the drive with different controller are compared in the conditions of starting and of load perturbation. The comparison for the starting response is done in terms of overshoot and the settling time (i.e. time taken to settle at the setpoint). 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
FLC controller	0.03 (offset)	0.508
Series Hybrid controller	0.38	0.65
Self tuning controller	0.65	0.6

Table 4.1 Comparison of responses at starting

The fuzzy logic controller has the least settling time but, there is an offset displayed at the setpoint which is undesirable. The response with the series hybrid controller is excellent in terms of reducing the overshoot; it also shows good response for the settling time. The designed self tuning controller shows an overshoot but an improved performance compared to the fixed gain PI controller response. The self tuning controller shows the best response in terms of settling time.

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 setpoint value after the instant of application of load) is presented in the Table 4.2

Type of the controller	**Undershoot (RPM)**	**Settling time (sec)**
PI controller	1.15	0.25
FLC controller	2 (offset)	-----------
Series Hybrid controller	0.49	0.14
Self tuning controller	0.55	0.12

Table 4.2 Comparison of responses 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 2RPM 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 quite low settling time. The designed self tuning controller also exhibited a quite lesser undershoot and also the lowest settling time. However the results displayed by all the controllers are similar in the case of speed direction reversal.

4.6.3 Comparative study of the controller performances

A. Proportional Integral speed controller

The responses of the PMBLDCM drive when using the PI speed controller for starting response; load perturbation and reversal of speed direction are shown in the Fig. 4.1 and Fig. 4.2. It can be observed from the plots that the PI controller completely eliminates the steady state error. But it brings an overshoot into the system response and also increasing the settling time for the speed. On the whole we can conclude that the PI controller makes the response of the system slower.

B. Fuzzy logic speed controller

The responses of the PMBLDCM drive with FL controller starting, load perturbation and speed reversal is shown in the figures 4.3, 4.4. It can be observed that the fuzzy logic controller gives an excellent transient state performance but introduces noise at the steady state. It has also not displayed any overshoot unlike the remaining controllers. The other advantage of the fuzzy logic controller is that it requires no exact mathematical model of the plant, a simple knowledge of the plant behaviour is sufficient to construct the FL controller. We have also seen in the literature survey that the performance of most of the electric drives can be improved by using different kinds of nonlinear speed controller techniques. In the present work we have observed that the fuzzy logic controller when used solely was not able to maintain the set point speed on the application of the load, it showed a response with offset. Hence when only fuzzy logic controller is used, it cannot improve the performance of the drive in all the terms.

C. Series Hybrid (Fuzzy precompensated PI) speed controller

The responses of the PMBLDCM drive with the series hybrid controller on starting, load perturbation and speed reversal is shown in the figures 4.5, 4.6. It has been observed that the series hybrid controller delivered the best performance than the conventional PI controller or the Non linear fuzzy controller. The disadvantages that are observed in the PI controller and the fuzzy logic controller are eliminated to a great extent when they are used in the present configuration of the controller. The construction of the series hybrid is such that the PI controller is connected in series with a FLC. The FLC will be modifying the reference command signal, which is further supplied to the PI controller. Hence the FLC will be continuously modifying the control reference and the PI will be delivering the required performance. As we have covered in the previous chapters, the function of the FLC in the present series hybrid configuration is to generate a modified reference signal based upon the actual speed error. This helped to eliminate the problem of overshoots and undershoots which normally occur in the PI controller. In the performance comparison presented graphically and in tabular form, the series hybrid has proven far more superior to the PI and FLC, during transient and steady state performance. This has increased the scope for research on different hybrid controller configurations.

D. Self tuning PI speed controller

The responses of the PMBLDCM drive with the designed self tuning PI controller on starting, load perturbation and speed reversal is shown in the figures 4.5, 4.6. In this control technique, the basic fact that in a PI controller, the proportional gain is responsible for the Transient response and the integral term is responsible for the steady state performance is used. The controller employs a Fuzzy logic controller to continuously modify the Proportional and the Integral gain values of the PI controller based on the operating point of the drive. During the transient response, the proportional gain is kept maximum; when the speed approaches near the setpoint, the Integral (I) gain comes into play. This ‘I’ gain helps to maintain the motor speed at the setpoint and to reduce the settling time, bring back to zero error on load perturbation. It is observed from the results shown in the figures that the self tuning PI controller efficiently reduces overshoots and undershoots during starting and load perturbation. The high integral gain action near the setpoint during load perturbation helps to achieve the least settling time compared to the remaining PI, FLC and the series hybrid controllers. This control strategy can work well even in the presence of severe and unknown non linearities.

4.7 Conclusion

The modelling, analysis, design and the simulation of the PMBLDC drive system has been done in the MATLAB/Simulink environment. A quite through comparative study has been carried out on the drive performance with different speed controllers. It has shown that the individual controllers have their own merits and demerits. Our choice of selection of controller for our application should be based on our 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 Self tuning 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 has made the drive to maintain speed at the set speed with quite a low undershoot and the least settling time. Besides the self tuning controller can be easily augmented with existing PI controller used in the industrial process, in parallel.

Chapter: 5 Main conclusions & suggestions for further work

5.1 General

Modeling and simulation of the performance of the permanent magnet brushless DC motor drive has been carried out using different speed controllers the PI controller, Fuzzy logic controller, series hybrid PI controller and the Self tuning PI controller. The main objective was to model, design and develop hybrid controllers and compare its performance with the conventional PI controller, Fuzzy logic controller for the speed control of a PMBLDCM drive by simulation in MATLAB/Simulink environment. This chapter is the overall summary of the investigations carried out throughout the thesis. The main conclusions are given in brief and the suggestions for further work were also presented.

5.2 Main conclusions

The mathematical model of the entire PMBLDCM drive with the PI controller, Fuzzy logic controller, series hybrid PI controller and the self tuning PI controller have been developed in the Simulink environment using the SimPowerSystems toolbox and FuzzyLogic toolbox. The drive is also simulated with the developed controllers in MATLAB/Simulink environment. The speed response of the drive with different controllers has been compared and analyzed. The comparative study has shown that the individual speed controllers have their own merits and demerits. The choice of choosing a speed controller for a particular application depends on the application. When the requirement is of simplicity and ease of application, PI speed controller would be a good choice. PI controllers are observed to have no steady state error but are slow in response.

The fuzzy logic controller offer good performance even in the presence of severe and unknown nonlinearity and an exact mathematical model of the plant is not required to develop this controller for the plant. But it has been observed that FLC offer offset error and noise at the steady state. It is further observed that if operation of motor is displaced from the setpoint, an offset remain present throughout the operation. But the FLC provides excellent transient response in terms of quickness of the response.

The series hybrid PI controller greatly helps to reduce overshoots and undershoots present in a normal PI controller response. The disadvantages present in the responses of the PI controller and the FLC when used separately, can easily be eliminated to a good extent i.e. in reducing or even completely eliminating the overshoots and undershoots. The controller also delivers a smooth transient and steady state response. The controller can work efficiently even in the presence of severe and unknown nonlinearities such as hysteresis, deadzone etc.

In the self tuning PI controller configuration, the gain values of the PI controller are modified continuously depending on the operating point of the speed response. The controller works efficiently in reducing the overshoot and making the drive more adaptive to load variations. Since all the control action is taken by the PI controller directly, the response is smooth and due to the gain tuning, it exhibits an adaptive performance during load variations.

5.3 Suggestions for further work

The proposed hybrid controller configuration displayed excellent simulation results and can be implemented on existing PI control system simply by adding the auto tuning techniques. With the availability of so advanced and powerful computing equipment like the Digital Signal Processors and PIC microcontrollers, the practical performance of the controller can be verified for control of the PMBLDCM drive.

In the present hybrid controller scheme, the PI controller gains are varied in their respective predetermined ranges to make the controller adaptive. The compatibility of other available controller techniques like the self tuning FPID controller, sliding mode controller, model reference adaptive controllers, with the PMBLDCM drive can also be verified through simulation and implementation.

A hardware setup may be implemented using the developed hybrid controller. The self tuning controller can actually be appended to existing PI controller already in use in use with industrial processes without modification in the control processes.

References

[1] Jan Jantzen, “Foundations of Fuzzy Control”, Wiley publications, 2007, ISBN 978-0-470-02963-3(HB).

[2] Onur Karaskal, Engin Yesil, Mujde Guzelkaya, Ibrahim Eksin, “Implementation of a New Self-Tuning Fuzzy PID Controller on PLC”, Turk J Elec Engin, Vol.13, No. 2, 2005

[3] Rajani K. Mudi and Nikhil R. Pal, “A Robust Self-Tuning Scheme for PI- and PD-Type Fuzzy Controllers”, *IEEE* Transactions on Fuzzy Systems, Vol. 7, No. 1, February 1999.

[4] Han-Xiong Li, Lei Zhang, Kai-Yuan Cai, and Guanrong Chen, “An Improved Robust Fuzzy-PID Controller with Optimal Fuzzy Reasoning”, IEEE Transactions on Systems, Man And Cybernetics—Part B: Cybernetics, Vol. 35, No. 6, December 2005.

[5] Jong-Hwan Kim, Kwang-Choon Kim, and Edwin K. P. Chong, “Fuzzy Precompensated PID Controllers”, *IEEE* Transactions on Control Systems Technology, Vol. 2, No. 4, December 1994.

[6] Bhim Singh, B.P. Singh, Sanjeet Dwivedi, “DSP Based Implementation of Fuzzy Precompensated PI Speed Controller for Vector Controlled PMSM Drive”, ICIEA 2006.

[7] Oyas Wahyunggoro and Nordin Saad, “Evaluations of Fuzzy-Logic-Based Self Tuning PI Controller and Fuzzy-Scheduled PID Controller for DC Servomotor”, 2008 , IEEE.

[8] Bhim Singh, A H N Reddy, S S Murthy, “Gain Scheduling Control of Permanent Magnet Brushless dc Motor”, IE(I) Journal-EL, Vol. 84, September 2003.

[9] Bhim Singh, A H N Reddy, S S Murthy, “Hybrid Fuzzy Logic Proportional Plus Conventional Integral-Derivative Controller for Permanent Magnet Brushless DC Motor”,

[10] Microchip, “Brushless DC (BLDC) Motor Fundamentals”, application notes AN855.

[11] C. K. Lee and W. H. Pang, “A Brushless DC Motor Speed Control System Using Fuzzy Rules”, Power Electronics and Variable-Speed Drives, 26 - 28 October 1994, Conference Publication No 399. IEE 1994.

[12] I. K. Bousserhane, A. Hazzab, M. RahlW, M. Kamli and B. Mazari, “Adaptive PI Controller using Fuzzy System Optimized by Genetic Algorithm for Induction Motor Control”, in Puebla, MEXICO, the CIEP, October 16-18, 2006.

[13] John Chiasson, “Modeling and High Performance Control Of Electric Motors”, *IEEE* press series on Power Engineering, John Wiley & Sons Publications.

[14] Hailong Song, Yong Yu, Ming Yang, Dianguo Xu, “A Novel SMC-Fuzzy Speed Controller for Permanent Magnet Brushless DC Motor”, 2003 IEEE.

[15] N. Kanagaraj, P. Sivashanmugam, and S. Paramasivam, “Fuzzy Coordinated PI Controller: Application to the Real-Time Pressure Control Process”, Hindawi Publishing Corporation Advances in Fuzzy Systems Volume 2008, Article ID 691808.

[16] Stanisław Skoczowski, Stefan Domek, Krzysztof Pietrusewicz, and Bogdan Broel-Plater, “A Method for Improving the Robustness of PID Control”, IEEE Transactions on,Industrial Electronics, Vol. 52, No. 6, December 2005.

[17] Kevin M. Passino and Stephen Yurkovich, “Fuzzy Control”, Addison Wesley Longman, Inc., Edition 1997, ISBN 0-201-18074-X.

[18] Guifang Cai, Kun Qian, Bangyuan Li and Xiangping Pang, “Robust PID Controller in Brushless DC Motor Application”, 2007 IEEE International Conference on Control and Automation, Guangzhou, China - May 30 to June 1, 2007.

[19] Bhim Singh, B P Singh, (Ms) K Jain, “Implementation of DSP Based Digital Speed Controller for Permanent Magnet Brushless DC Motor”, IE(I) Journal-EL, Vol. 84, June 2003.

[20] Ji Hua, Li Zhiyong, “Simulation of Sensorless Permanent Magnetic Brushless DC Motor Control System”, Proceedings of the IEEE International Conference on Automation and Logistics ,Qingdao, China. September 2008.

Appendix

I. 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.

II. Controller gain values

The gain values used for the PI controller are

K_p= 3, K_i= 45

Bio Data

Name: Vakalapudi Harish Kumar

Qualification: B.Tech (Electrical and Electronics Engineering)

JB Institute of Engineering and Technology, Hyderabad affiliated to

Jawaharlal Nehru Technological University, Hyderabad