CHAPTER 4

FUZZY TUNED PI AND PID CONTROLLER FOR
PMSM DRIVE

4.1 INTRODUCTION

Fuzzy logic is a rule-based decision making method, used to control a process that a human can control with expertise gained from experience. Fuzzy Logic Controller (FLC) is a type of modern controller system, which is a knowledge based approach. FLC has progressed as a substitute for traditional control strategies in automatic control systems. Fuzzy control theory offers non-linear controllers that are capable of performing different complex non-linear control action. Their design does not require precise knowledge of the system model such as the poles and zeros of the system transfer function. LotfiZadeh (1960) introduced fuzzy theory as an extension of traditional control theory. The fuzzy sets were formulated from elements which have degrees of membership and are framed based on the error and change in error of a system.

Conventional controller is difficult to design if an accurate system model is not available and it requires meticulous fine tuning (Kamel et al 2004). Nour et al (2006) approached towards high performance motor drive requires fast and accurate speed response, quick recovery of speed from load disturbances. Conventional fixed gain PI controllers are sensitive to parameter variations, load and speed disturbances. PI controller is slow in adapting to speed changes, load disturbances and parameters variations without
continuous tuning of its gains. On-line self tuning scheme using fuzzy logic controller (FLC) was proposed by Nour et al (2008) to improve the drive performance compared to conventional PI controller.

In this chapter, the background study of fuzzy systems, the basic components, the membership functions, IF-THEN rules and the proposed fuzzy tuned PI controller systems are presented. The proposed fuzzy tuned PI, fuzzy tuned PID, Accelerated fuzzy PI controller schemes are simulated using MATLAB / SIMULINK. The simulation results are compared with the traditional PI controller scheme based PMSM drive systems.

4.2 BACKGROUND STUDY OF FUZZY SYSTEMS

A fuzzy system is based on fuzzy logic, which is a mathematical system that analyzes logical variables which operates on discrete values of either 1 or 0 i.e. true or false. Fuzzy logic employs the logic of approximate reasoning and continues to grow, as it provides an inexpensive solution for controlling ill-known systems. Fuzzy control algorithm is based on a linguistic control strategy, which is derived from expert knowledge. It does not need any difficult mathematical calculations, but uses them to simulate expert knowledge and provides a good performance on a control system.

Fuzzy set theory (LoftiZadeh 1960), differs from traditional Boolean set theory. It allows a partial membership in the set, whereas the traditional Boolean set is two-valued represented as 1 or 0. The partial membership or a degree of membership might be any value along the continuum of 0 to 1. Fuzzy set theory uses a linguistic term, which is defined quantitatively as a membership function. This membership function specifically defines the degrees of membership based on the property of the control variable such as speed or torque. With the membership functions
defined for controller system inputs or outputs, the IF-THEN type conditional rules are formed. The rule base and corresponding membership functions are employed to analyze the controller inputs and determines the outputs by using fuzzy logic inference.

Uddin and Rahman (1999 & 2007) proposed fuzzy logic algorithms for speed control of interior PMSM drives. The FLC was developed with less computational burden which was suitable for real time implementation at high speed operating conditions. Aissaoui et al (2007) designed and proposed a fuzzy logic controller for synchronous motor. The drive system was designed and analysed with performances to validate the theoretical concept.

Cau & Bi (2009) proposed self tuning PI and PD controller PMSM servo drive and performance of the simulation results showed to improve the robustness of the drive. Isa et al (2009), Lazi et al (2011) and Shikkewal & Nandanwar (2012) investigated the fuzzy logic based vector controlled PMSM drive with minimum number of fuzzy rules. The performances of the drive were compared with standard rules under starting and load disturbance conditions.

Pundaleek et al (2009) compared the fuzzy logic controller and PI controller performances for speed control of induction motor. Speed of the PMSM motor was regulated by using fuzzy logic controller with speed error and its derivative as input to the fuzzy controller.

Xiaping & Jidong (2010) presented self tuning PID algorithm based fuzzy logic to improve the performances of the drive system. Speed of the motor was regulated using a PI controller by adjusting the gain values using fuzzy logic approach.
The vector control PMSM drive proposed in this chapter utilizes the speed error and the change in speed error as parameters to frame the fuzzy rules. The fuzzy inference system uses Mamdani method for framing the fuzzy rules and the crisp values are fuzzified using min-max method. The center of gravity method of defuzzification is adopted to obtain the controlled outputs of the fuzzy controller. A fuzzy logic controller using these crisp values is used to control the PMSM drive. The crisp values obtained as output of the fuzzy controller forms the gain values of the PI controller. In this manner, the proposed system acts as fuzzy tuned PI controller. The proposed drive is simulated using MATLAB/SIMULINK and the results are compared with the traditional PI controller based PMSM drive system.

4.3 FUZZY LOGIC SYSTEM AND ITS COMPONENTS

The inherent approximation capability, high degree of tolerance, smooth operation, reduced effect of non-linearity and faster learning ability is the advantages of fuzzy logic system. These advantages encourage the use of fuzzy logic in wide control applications. The block diagram of fuzzy logic based motor control system is shown in Figure 4.1. The system includes a Fuzzifier, Inference Engine, Defuzzifier and Rule base.

![Figure 4.1 Block diagram of fuzzy logic based motor control system](image-url)
4.3.1  Fuzzy Set

Fuzzy sets were introduced by Lotfi Zadeh (1965) and are sets whose elements have degrees of membership. The fuzzy sets are extension of classical sets. The membership of classical set theory is assessed in binary terms as 1 or 0. But the fuzzy set membership functions are valued in the interval [0, 1]. In fuzzy set theory, the classical sets are termed as crisp sets. The fuzzy sets are inexact and defined classes, as they do not have sharply defined boundaries.

4.3.2  Membership Functions (MF)

Membership function is a function that changes the crisp values into fuzzy sets. All information contained in a fuzzy set is described by its membership function. Membership functions describe the degree of confidence of all elements in the universe of discourse to each fuzzy set. The shape of membership functions depends on the system that is to be controlled. The function that ties a number to each element x of the universe is called the membership function μ(x)

![Figure 4.2 (a) Membership function of fuzzy set](image)
The fuzzy membership functions are classified into four types based on the shapes. They are:

i. Trapezoidal Membership Functions
ii. Triangular Membership Functions
iii. Gaussian Membership Functions
iv. Generalized Bell Membership Functions

The most popular shapes are triangular and trapezoidal membership functions as these shapes are easy to represent designer’s idea and also require low computation time. Other shapes may be able to reflect natural phenomena, but may require complex equation to model, which can increase the size and complexity of the fuzzy system. The membership functions are shown in Figure 4.2.
Figure 4.2 (c) Trapezoidal membership function

Figure 4.2 (d) Gaussian membership function
4.3.3 Fuzzification

Fuzzification is the process of converting or transforming the measured inputs of the system called crisp value, into the fuzzy linguistic values, called the membership functions. It transforms the physical values of the error signal, rate of change of error which is input to the fuzzy logic controller, into a fuzzy set consisting of an interval for the range of input values and an associate membership function. The conversion process is performed by a membership function. This fuzzification takes place in the fuzzifier block.

4.3.4 Rule Base

The rule base is a collection of expert control rules which are needed to obtain the controlled output. The rule base is of IF-THEN type. The IF-THEN statement is the one in which the words are characterized by
continuous membership functions. After defining the fuzzy sets and assigning their membership functions, rules must be written to describe the action to be taken for each combination of control variables. These rules will relate the input variables to the output variable using IF-THEN statements which allow decisions to be made. Each rule is represented as follows:

IF (antecedent) THEN (consequence)

4.3.5 Inference Engine

Inference engine is a software code that processes the rules based on the facts of a given situation. It is an information processing system, which employs inference steps similar to that of a human brain. The inference mechanism provides the mechanism for referring to the rule base such that the appropriate rules are fired. The two most commonly used inference procedures in FLC are Mamdani’s Max-Min and Max-Algebraic product (or Max-Dot) composition.

The inference or firing with this fuzzy relation is performed via the operation between the fuzzified crisp input and the fuzzy relation representing the meaning of the overall set of rules. As a result of the composition, one obtains the fuzzy set describing the fuzzy values of the overall control output. In this thesis, the Mamdani’s Max-min composition inference method is used.

4.3.6 Defuzzification

Defuzzification unit in FLC is the inverse of the fuzzification process. It converts the fuzzy controller output fuzzy variables in to a crisp real signal. The output membership functions obtained from the inference engine are converted into the crisp values in the defuzzifier. This process is
termed as defuzzification. There are different methods of defuzzification and the common methods are:

i  Centre of Gravity (COG)

ii  Bistor of Area (BOA)

iii  Mean of Maximum (MOM)

Unfortunately, there is no systematic procedure for choosing a defuzzification strategy.

4.3.6.1 Centre of Gravity (COG)

COG is called Centre of Gravity for Singletons (COGS), where the crisp control value is the abscissa of the centre of gravity of the fuzzy set and is calculated using Equation (4.1).

\[ U_{COG} = \frac{\sum \mu_A(x_i) x_i}{\sum \mu_A(x_i)} \]  \hspace{1cm} (4.1)

where, \(x_i\) is a point in the universe of the conclusion \((i=1, 2, 3 \ldots)\) and \(\mu_A(x_i)\) is the membership value of the resulting conclusion set.

4.3.6.2 Bisector of Area (BOA)

The Bisector of Area (BOA) defuzzification method is a computationally complex method, in which the abscissa of the vertical line is calculated that divides the area of the resulting membership function into two equal areas. For discrete sets, \(u_{BOA}\) is the abscissa \(x_j\) that minimizes

\[ U_{BOA} = \left| \sum_{i=1}^{j} \mu_A(x_i) - \sum_{i=j+1}^{i_{max}} \mu_A(x_i) \right|, \quad i < j < i_{max} \]  \hspace{1cm} (4.2)

Here \(i_{max}\) is the index of the largest abscissa \(x_i\)
4.3.6.3 Mean of Maximum (MOM)

In Mean of Maximum (MOM) method the crisp value is to choose the point with the highest membership. When there are several points which have maximum membership value, then, the mean of all the maximum membership values is calculated using Equation (4.3).

\[ U_{MOM} = \frac{\sum_{i \in I} x_i}{|I|}, I = \{i | \mu_A(x_i) = \mu_{\text{max}} \} \]  

(4.3)

Here \( I \) is the (crisp) set of indices \( i \) where \( \mu_A(x_i) \) reaches its maximum \( \mu_{\text{max}} \), and \( |I| \) is its cardinality (the number of members).

Based on the defuzzification process the fuzzy logic controllers are classified into two types namely, Mamdani Fuzzy controller and Takagi – Sugeno Fuzzy Controller.

4.3.7 Mamdani Fuzzy Controller

The Mamdani Fuzzy Controller is a crisp based controller which produces crisp outputs from crisp inputs. The inference engine utilizes Mamdani fuzzy inference method proposed by Ebrahim Mamdani (1975). In this model the output membership functions are of fuzzy sets. These fuzzy sets are defuzzified to crisp values.

In Mamdani method, commonly the COG method of defuzzification is used to obtain the crisp outputs. The structure of Mamdani fuzzy controller shown in Figure 4.3 consists of four main parts: Fuzzification of the inputs, Rule Evaluation, Aggregation of the rules and Defuzzification.
Figure 4.3 Mamdani fuzzy controller

Fuzzification converts the input data to degree of membership functions. The data is matched with the rule condition and for any particular instance it is determined how well the data is matched with the rule. In this way, the degree of membership is determined. Based on the system requirement, the If – Then rules are written. Generally the fuzzy controller works on both multiple input and multiple output and single input and single output logics.

The inference engine aggregates the degree of fulfillment of the fuzzy sets according to the conditions specified in the rule base. In activation minimum of two aggregated value are selected and only thickened part of singleton is activated. Its multiplication results are slighter smooth control. Then all activated conclusions are accumulated using max operations. The resulting fuzzy set is converted into its crisp value by using centre of gravity method of defuzzification.
4.3.8  Takagi – Sugeno Fuzzy Controller

The Takagi-Sugeno fuzzy controller is another type of fuzzy controller, in which, the membership functions are not used by the defuzzification process to obtain the crisp outputs. This model was proposed by Takagi, Sugeno and Kang (1985). They made an effort to develop a systematic approach to generate fuzzy rules from a given input-output dataset. The typical fuzzy rule of Takagi-Sugeno model is as follows:

If $x$ is $A$ and $y$ is $B$, then $z = f(x, y)$

where, $A$ and $B$ are fuzzy sets in the antecedent, while $z=f(x, y)$ is a crisp function in the consequent. The block diagram of Takagi-Sugeno fuzzy controller is shown in Figure 4.4. The block diagram shows that the crisp outputs are obtained directly from the inference engine without using defuzzification block.

Figure 4.4 Takagi – Sugeno Fuzzy Controller
4.4 DEVELOPMENT OF FUZZY TUNED PI CONTROLLER

The fuzzy controller proposed in this thesis is a Mamdani fuzzy controller which works on Multiple Input and Single Output (MISO) logic. The structure of fuzzy tuned PI controller scheme proposed is shown in Figure 4.5.

![Fuzzy Tuned PI Controller Scheme](image)

**Figure 4.5 Structure of Fuzzy Tuned PI Controller Scheme**

The fuzzy tuned PI controller uses the speed error and error change rate as fuzzy inputs, and the proportional factor ($K_p$) and integral factor ($K_i$) as fuzzy outputs.

The error and rate of change of error are defined as:

$$e(k) = r(k) - y(k)$$  \hspace{1cm} (4.4)\\
$$ec(k) = \frac{e(k) - e(k-1)}{T_s}$$  \hspace{1cm} (4.5)

where $r(k)$ is the reference input speed signal, $y(k)$ is the output speed response, $e(k)$ is the error signal, and $ec(k)$ is the rate of change of error. The fuzzy PI controller is designed to replace the conventional PI controller. The fuzzy PI controller is used to improve the performance of the system (Limei et al 2007). The fuzzy inference engine of the fuzzy controller adjusts the gain $K_p$ and $K_i$ based on the fuzzy rule base. The fuzzy controllers adjust these gain values based on the speed error ($e$) and rate of change of error ($ec$).
functions. The calculated speed of controller is very quick, which can satisfy the rapid need of controlled object.

The control algorithm of traditional PI controller can be described as,

\[ u(k) = K_p e(k) + K_i \int e(k) \]  \hspace{1cm} (4.6)

where, \( K_p \) is the proportional gain, \( K_i \) is the integral gain and \( e(k) \) is the speed error.

The design algorithm of fuzzy PI controller is to adjust the \( K_p \) and \( K_i \) parameters online through fuzzy inference based on the speed error \( e \) and rate of change of speed error \( ec \) to make the control object attain good dynamic and static performances.

### 4.4.1 Design of Membership Functions

The fuzzy variable of input variable \( e \) is \( E \) and \( ec \) is \( EC \). The fuzzy variables are \( K_p \) and \( K_i \). The fuzzy sets of \( E \) and \( EC \) are all defined as \{NB, NS, ZO, PS, PB\}, where NB, NS, ZO, PS and PB represent Negative Big, Negative Small, Zero, Positive Small and Positive Big respectively.

The fuzzy sets of \( K_p \) and \( K_i \) are defined as \{Z, S, M, B\}, where Z, S, M and B represent Zero, Small, Medium and Big respectively. The membership functions of \( E \), \( EC \), \( K_p \) and \( K_i \) are triangular distribution functions. The membership functions for each variable are shown in Figure 4.6, Figure 4.7 and Figure 4.8 respectively.
4.4.2 Fuzzy Rule Base

The principle of designing fuzzy rules is that the output of controller can make the system output response dynamic and static performances optimal. The fuzzy rules are generalized as given in Table 4.1 and Table 4.2 according to the expert experiment in PMSM servo system and simulation analysis of the system (Limei Wang et al 2007). The Mamdani inference method is used as the fuzzy inference mode. The inference can be written as "IF A AND B THEN C". For example "IF E is NS AND EC is PS THEN KP is S, KI is M". $K_p$ and $K_i$ are written the same as 25 fuzzy
condition statements. The output variable can be obtained by the MIN - MAX inference. The weighted average method is adopted for defuzzification.

Table 4.1

Control rules for $K_p$

<table>
<thead>
<tr>
<th>$K_p$</th>
<th>EC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB</td>
</tr>
<tr>
<td>NB</td>
<td>B</td>
</tr>
<tr>
<td>NS</td>
<td>M</td>
</tr>
<tr>
<td>ZO</td>
<td>M</td>
</tr>
<tr>
<td>PS</td>
<td>S</td>
</tr>
<tr>
<td>PB</td>
<td>M</td>
</tr>
</tbody>
</table>

Table 4.2

Control rules for $K_I$

<table>
<thead>
<tr>
<th>$K_I$</th>
<th>EC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB</td>
</tr>
<tr>
<td>NB</td>
<td>Z</td>
</tr>
<tr>
<td>NS</td>
<td>M</td>
</tr>
<tr>
<td>ZO</td>
<td>B</td>
</tr>
<tr>
<td>PS</td>
<td>S</td>
</tr>
<tr>
<td>PB</td>
<td>Z</td>
</tr>
</tbody>
</table>

4.4.3 Similation Model of Fuzzy Tuned PI Controller

The MATLAB simulation model of the fuzzy tuned PI controller is shown in Figure 4.9. The fuzzy editor is used to create the fuzzy membership functions. The FIS editor shows the methods used for fuzzification, defuzzification and also the numerical range of the linguistic variables used. The FIS Editor used for the simulation and the input membership function editor generates the error fuzzy sets from the linguistic variables and the corresponding membership function plots are obtained.

The sets NB, NS, ZO, PS and PB are framed using triangular membership function. Figure 4.10(a), 4.10(b) and 4.10(c) portrays the FIS Editor, Rule Editor, and Rule Viewer of the MATLAB fuzzy logic tool box respectively. Using these dialog boxes, the fuzzified and crisp values can be obtained.
Figure 4.9 Simulink model of fuzzy tuned PI controller

Figure 4.10(a) FIS Editor
Figure 4.10(b) Rule editor

Figure 4.10(c) Rule viewer
4.4.4 Simulation Results Fuzzy PI Controller

Numerous simulation tests are carried out to verify the performance of the proposed fuzzy logic controller (FLC) based PMSM drive. In order to prove the superiority of the fuzzy tuned PI controller based PMSM drive, it is simulated under various operating conditions such as at no load, full load, sudden change in load and step change in command speed.

The speed response of proposed FLC based PMSM drive is shown in the Figure 4.11. The speed response using fuzzy tuned PI controller is compared with PI controller response and is depicted in the Figure 4.12. The Figure 4.13 illustrates the overshoot in speed response compared with PI controller response.

![Figure 4.11 No load speed response with FPI](image-url)
Figure 4.12 Speed response comparisons with PI and FPI

Figure 4.13 Overshoot in speed response with PI and FPI

The performance of the drive was also investigated and compared with conventional controller for step change in command speed and Figure 4.14 shows the speed response comparison for two speed level. The motor was running at no load condition and after some time, sudden load of 5 N-m was applied to the motor which is shown in Figure 4.15. The
restoration time required to reach the reference speed was observed and is shown in the Figure 4.16.

![Graph showing speed response of change in speed command with PI and FPI](image)

**Figure 4.14 Speed response of change in speed command with PI and FPI**

![Graph showing speed response under change in load with FPI](image)

**Figure 4.15 Speed response under change in load with FPI**

The control parameters overshoot, steady state error and settling time with respect to various set speed are recorded and tabulated in the Table 4.3. The performance characteristic curves are plotted from the simulation results and are shown in Figure 4.17.
Figure 4.16 Restoration response with FPI

Table 4.3 Fuzzy PI controller performance

<table>
<thead>
<tr>
<th>S.No</th>
<th>Ref Speed (rpm)</th>
<th>Peak Overshoot (%)</th>
<th>Steady State Error (%)</th>
<th>Settling Time $T_s$ (s)</th>
<th>Speed Change under load (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>300</td>
<td>2.66</td>
<td>1</td>
<td>0.01</td>
<td>8.33</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>1.4</td>
<td>0.9</td>
<td>0.012</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>800</td>
<td>1.25</td>
<td>0.75</td>
<td>0.014</td>
<td>3.13</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>1.11</td>
<td>0.7</td>
<td>0.014</td>
<td>2.5</td>
</tr>
<tr>
<td>5</td>
<td>1200</td>
<td>1.08</td>
<td>0.583</td>
<td>0.014</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1500</td>
<td>1.13</td>
<td>0.466</td>
<td>0.016</td>
<td>1.6</td>
</tr>
<tr>
<td>7</td>
<td>1800</td>
<td>1.22</td>
<td>0.333</td>
<td>0.017</td>
<td>1.4</td>
</tr>
<tr>
<td>8</td>
<td>2000</td>
<td>1.11</td>
<td>0.25</td>
<td>0.018</td>
<td>1.25</td>
</tr>
</tbody>
</table>
Figure 4.17 Performance characteristics of PI and fuzzy PI controller
(a) Overshoot (b) Steady state error (c) Settling time
4.4.5 Inference

Table 4.3 elucidates the performance characteristics of PMSM drive using fuzzy tuned PI controller for various speed values. The peak overshoot and steady state error decrease with increase in reference speed. The peak overshoot is high at low reference speeds. The overshoot and steady state error is 2.6% and 1.0% respectively for reference speed of 300 rpm. However PI controller based PMSM drive system has an overshoot of 11.67 % and steady state error of 0.5% for reference speed of 300 rpm. More over fuzzy tuned PI controller based drive system is faster than that of PI controlled based PMSM drive.

The fuzzy tuned PI controller provides superior performance at all speeds with reduced overshoot. When the reference speed is 1000 rpm, the peak overshoot is 1.1% and the steady state error is 0.7% only. The motor speed drops by 2.5% at 1000 rpm, when the step load of 5 Nm is applied. The drive will reach the set speed with steady state error at a restoration time of 1.0 s. The performance of the PMSM drive under sudden step change in load is similar for both conventional PI controller and fuzzy PI controller.

4.5 FUZZY PID CONTROLLER

The fuzzy PID controller is designed to replace the conventional PI controller. The fuzzy PID controller is used to improve the performance of the system. The fuzzy inference engine of the fuzzy controller adjusts the gain $K_p$, $K_i$ and $K_d$ based on the fuzzy rule base. The fuzzy controllers adjust these gain values based on the speed error ($e$) and rate of change of error ($ec$) functions. The calculated speed of controller is very quick, which can satisfy the rapid need of controlled object. The block diagram of control system is shown in Figure 4.18.
The control algorithm of traditional PID controller can be described as

\[ u(k) = K_p e(k) + K_i \int c(k) + K_d \frac{d e(k)}{dt} \]  (4.7)

where \( K_p \) is the proportional factor, \( K_i \) is the integral factor, \( K_d \) is the derivative factor, \( e(t) \) is the speed error and \( u(k) \) is the actuating signal.

The design algorithm of PID controller is to adjust the \( K_p \), \( K_i \) and \( K_d \) online through fuzzy inference based on error and rate of change of error. In this design the error in speed ‘e’ and rate of change of error ‘ec’ are used as fuzzy inputs and \( K_p \), \( K_i \) and \( K_d \) are used as fuzzy outputs. The fuzzy sets of e and ec are all defined as \{ NB, NS, ZO, PS, PB \}, where NB, NS, ZO, PS, PB represents negative big, negative small, zero, positive small, positive big respectively.

The fuzzy sets of \( K_p \), \( K_i \), \( K_d \) are all defined as \{Z, S, M, B\} where Z, S, M, B represents zero, small, medium, big respectively. The membership function of e, ec, \( K_p \), \( K_i \) and \( K_d \) are triangular distribution function. The fuzzy rules according to the simulation analysis of the system are given in Table 4.4. The membership function editor for \( K_d \) is shown in Figure 4.19.
Figure 4.19 Fuzzy membership functions of $K_D$

Table 4.4 Control rules for $K_d$

<table>
<thead>
<tr>
<th>KD</th>
<th>EC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NB</td>
</tr>
<tr>
<td>E</td>
<td>NB</td>
</tr>
<tr>
<td></td>
<td>NS</td>
</tr>
<tr>
<td></td>
<td>ZO</td>
</tr>
<tr>
<td></td>
<td>PS</td>
</tr>
<tr>
<td></td>
<td>PB</td>
</tr>
</tbody>
</table>

4.5.1 Simulation Model of Fuzzy Tuned PID Controller

The simulation model of fuzzy PID controller is shown in Figure 4.20. The output fuzzy logic controllers are $K_p$, $K_i$ and $K_d$. The simulation model of fuzzy PID controller constructed based on the Equation (4.7).
4.5.2 Simulation Results of Fuzzy PID Controller

To predict the performance of the PMSM drive using fuzzy tuned PID controller, the drive system is simulated under various operating conditions such as at no load, full load, sudden change in load and step change in command speed.

Figure 4.21 illustrates the speed response of drive using fuzzy PID controller and overshoot in speed response is compared with fuzzy PI controller. Figure 4.22 demonstrates that fuzzy tuned PID controller based drive system is capable of following the command speed without any overshoot and with less steady state error.

The speed drop of the drive under sudden increase in load and restoration time to reach the reference speed is shown in Figure 4.23 and Figure 4.24 respectively. Figure 4.25 shows the speed response for step change in command speed.
Figure 4.21 No load speed response with FPI

Figure 4.22 Overshoot in speed response with FPI and FPID
Figure 4.23 Speed response under change in load with FPI and FPID

Figure 4.24 Restoration response with FPI and FPID
Figure 4.25 Change in speed command with FPI and FPID

The control parameters overshoot, steady state error and settling time with respect to various set speed are noted and tabulated in Table 4.5. From the simulated results, the characteristic curves are obtained and shown in Figure 4.26.

Table 4.5 Fuzzy PID controller performance

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Ref Speed (rpm)</th>
<th>Peak Overshoot (%)</th>
<th>Steady State Error (%)</th>
<th>Settling Time ($T_s$) (s)</th>
<th>Speed Change under load (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>300</td>
<td>0</td>
<td>1.00</td>
<td>0.012</td>
<td>8.33</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>0</td>
<td>0.90</td>
<td>0.014</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>800</td>
<td>0</td>
<td>0.87</td>
<td>0.016</td>
<td>3.13</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>0</td>
<td>0.80</td>
<td>0.018</td>
<td>2.5</td>
</tr>
<tr>
<td>5</td>
<td>1200</td>
<td>0</td>
<td>0.75</td>
<td>0.018</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1500</td>
<td>0</td>
<td>0.64</td>
<td>0.018</td>
<td>1.6</td>
</tr>
<tr>
<td>7</td>
<td>1800</td>
<td>0</td>
<td>0.39</td>
<td>0.020</td>
<td>1.4</td>
</tr>
<tr>
<td>8</td>
<td>2000</td>
<td>0</td>
<td>0.30</td>
<td>0.020</td>
<td>1.25</td>
</tr>
</tbody>
</table>
Figure 4.26 Performance characteristics of fuzzy PI and fuzzy PID
(a) Overshoot (b) Steady state error (c) Settling time
The performances of the fuzzy tuned PID controller and fuzzy tuned PI controller are almost similar. However the peak overshoot is zero for all range of speeds in case of fuzzy PID controller based PMSM drive.

4.6 PROBLEMS WITH FUZZY PI AND FUZZY PID CONTROLLERS

In practical industrial application, the sudden change of load is a common phenomenon. Therefore it is necessary to improve the PMSM drive performance under a sudden increase or decrease in load and to maintain the drive speed at constant level with insensitive to load fluctuations.

The motor speed reduces by 5% at 500 rpm and 2.5% at 1000 rpm when the step load of 5 Nm is applied at steady state. The drive will reach the speed at a restoration time of 0.7 s and 1.0 s for 3 Nm and 5 Nm load respectively when the motor reference speed is 1000 rpm.

In order to eliminate this problem in the PMSM drive, fuzzy logic concept with accelerated rate of change error as input called accelerated fuzzy concept, accelerated fuzzy tuned PI controller (AFPI) is formulated to improve the dynamic behavior of PMSM drive under change in load.

4.7 ACCELERATED FUZZY PI SPEED CONTROLLER

This is a modified form of Fuzzy PI controller. The accelerated fuzzy PI controller is proposed in this paper to improve the transient response of the PMSM drive system. The fuzzy PI controller uses only two inputs- speed error (e) and rate of change of speed error (ec). But in this model an additional input named ‘accelerated rate of change of error’ (acc) is used to improve the transient response of the system.
With these three inputs, the structure of the FLC is composed of two independent parallel fuzzy control blocks, each of which contains the corresponding fuzzy control rules and a defuzzifier. The incremental output of the FLC is formed by algebraically adding the outputs of the two fuzzy control blocks. The block diagram of the configuration is shown in Figure 4.27.

Figure 4.27 Block diagram of accelerated fuzzy PI controller

4.7.1 Simulation Model of Accelerated Fuzzy Tuned PI Controller

Figure 4.28 illustrates the simulation model of the accelerated fuzzy tuned PI controller. Two fuzzy control blocks are used in this proposed system. The fuzzy rules for the both fuzzy controller are designed based on membership function.
Figure 4.28 Simulation model of accelerated fuzzy tuned PI controller

4.7.2 Simulation Results of Accelerated Fuzzy PI Controller

In order to prove the superiority of the accelerated fuzzy tuned PI controller (AFPI) based PMSM drive, the drive system is simulated under various operating condition such as no load, full load, sudden change in load etc.

The no load speed response is shown in Figure 4.29. The peak overshoot speed response of fuzzy PI, fuzzy PID and accelerated fuzzy PI controller are compared and shown in Figure 4.30.
Figure 4.29 No load speed response with AFPI

Figure 4.30 Overshoot in speed response with FPI, FPID and AFPI
Figure 4.31 illustrates the speed drop of the drive for 5 N-m at 0.1 s. Also speed drop under sudden change in load is compared by using all the three fuzzy controllers and it is depicted in the Figure 4.32. Figure 4.33 demonstrates the restoration time required to reach the reference speed with a load of 5 Nm.

Figure 4.31 Speed response under change in load with AFPI

Figure 4.32 Speed response comparisons under load with FPI, FPID and AFPI
Figure 4.33 Restoration response with AFPI

The control parameters overshoot, steady state error and settling time with respect to various set speed are recorded and tabulated in Table 4.6. The performance characteristic curves are plotted from the simulation results and are shown in Figure 4.34.

Table 4.6 elucidates the performance characteristics of the PMSM drive using accelerated fuzzy PI controller for various reference speeds. The overshoot and steady state error are 15.67% and 0.33% respectively for the reference speed of 300 rpm. Also overshoot of 5.5% and no steady state error for the reference speed of 1000 rpm is observed. The drive performance is poor under starting condition with more peak overshoot. But motor speed reduces by 2.6% at 500 rpm 1.2% at 1000 rpm when the step load of 5 Nm is applied at steady state.
The drive will reach the constant speed at restoration time of 0.7 s for 5 Nm load by using accelerated fuzzy PI controller whereas restoration time of 1.0 s is required to reach the constant speed by using other general fuzzy logic controllers. Therefore the proposed accelerated fuzzy based PMSM drive is found to be robust at various speeds and sudden change in load.

Table 4.6 Accelerated Fuzzy PI controller performance

<table>
<thead>
<tr>
<th>S.No</th>
<th>Ref Speed (rpm)</th>
<th>Peak Overshoot (%)</th>
<th>Steady State Error (%)</th>
<th>Settling Time $T_s$ (s)</th>
<th>Speed change under load (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>300</td>
<td>15.66</td>
<td>0.33</td>
<td>0.012</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>12.2</td>
<td>0.24</td>
<td>0.016</td>
<td>2.4</td>
</tr>
<tr>
<td>3</td>
<td>800</td>
<td>6.25</td>
<td>0.125</td>
<td>0.018</td>
<td>1.5</td>
</tr>
<tr>
<td>4</td>
<td>1000</td>
<td>5.55</td>
<td>0</td>
<td>0.02</td>
<td>1.2</td>
</tr>
<tr>
<td>5</td>
<td>1200</td>
<td>5.25</td>
<td>0</td>
<td>0.02</td>
<td>1.0</td>
</tr>
<tr>
<td>6</td>
<td>1500</td>
<td>3.53</td>
<td>0</td>
<td>0.021</td>
<td>0.8</td>
</tr>
<tr>
<td>7</td>
<td>1800</td>
<td>4</td>
<td>0</td>
<td>0.022</td>
<td>0.67</td>
</tr>
<tr>
<td>8</td>
<td>2000</td>
<td>3.35</td>
<td>0</td>
<td>0.022</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Figure 4.34 (Continued)
(b) Figure 4.34 (Continued)
Figure 4.34 Performance characteristics of fuzzy PI, accelerated fuzzy PI and fuzzy PID

(a) Overshoot (b) Steady state error (c) Settling time (d) and (e) Speed change

4.8 PROBLEMS WITH ACCELERATED FUZZY PI CONTROLLER

The PMSM drive exhibits good speed response and insensitive to load fluctuations for constant speed applications. The accelerated fuzzy PI controller has superior performance as compared to the fuzzy controllers under steady state conditions. However peak overshoot is higher under starting condition of the drive which is not good for a robust drive. Hence the hybrid fuzzy controller is proposed by combining conventional fuzzy controller and accelerated fuzzy controller. The advantages of fuzzy PI and accelerated fuzzy PI can be combined which can be used as hybrid fuzzy speed controller.
4.9 SUMMARY

The performance of the PMSM drive using fuzzy tuned PI controller and fuzzy tuned PID controller are investigated for various operating conditions. A novel accelerated fuzzy controller has been proposed to improve the transient response of PMSM drive. The fuzzy controller uses only the inputs, speed error (e), and rate of change of error (ec). But additional input named as accelerated rate of change of error (acc) is used in accelerated fuzzy PI controller.

The basics of a fuzzy system, fuzzification, defuzzificaton, rule base and fuzzy inference system have been studied for the application to verify the compatibility of fuzzy systems with PMSM drive. The fuzzy logic is used to obtain optimal values of proportional, integral and derivatives gains of PI, PID controller of the closed loop PMSM drive. The proposed system design and performance characteristics for the various values have been presented. The performance characteristics of steady state error and setting time with FPID controller showing less difference when compared to fuzzy PI controller. The problems associated with the fuzzy controllers and superiority of these controllers are analyzed and concluded to propose hybrid controllers by combining general fuzzy PI, fuzzy PID controller and accelerated fuzzy tuned PI controller.