CHAPTER 2

CONVENTIONAL AND FUZZY BASED CONTROLLERS FOR INDUCTION MOTOR DRIVES

2.1 INTRODUCTION

The field oriented control has made ac drives equivalent to dc drives in independent flux and torque control and superior in their dynamic performance. These developments positioned the ac drives for high performance applications, hitherto reserved for separately excited dc motor drives (Bose 1982). This chapter deals with the speed controllers used for indirect field oriented control (IFOC) of induction motor drive. The IFOC of the induction motor drive is simulated with conventional PI and PID controllers. A fuzzy based self-tuning controller is developed and simulated for the induction motor drive. The performance of the drive system for these speed controllers is compared.

2.2 IMPLEMENTATION OF IFOC FOR INDUCTION MOTOR

The dynamic modeling of the induction motor drive for IFOC discussed in Bose (2006) is given in appendix 1 for ready reference. The configuration of the indirect field oriented control of induction motor drive is

Part of the thesis work reported in this chapter has been published as detailed below:

shown in Figure 2.1 (Bose 2006). In this configuration the motor is fed from a voltage source inverter. The command speed of the motor is compared with the actual speed to generate the error speed. The speed error is given as input to the speed controller.

**Figure 2.1 Configuration of IFOC scheme for Induction motor drive**

The speed controller generates the command torque of component current and thus the q-axis command current $i_{qs}^*$ for the induction motor drive. The command currents are compared with their respective d-axis and q-axis currents generated by the transformation of the stator currents. The respective errors generate the voltage command signal $v_{ds}^*$ and $v_{qs}^*$ through the PI controllers. These voltages are converted into stationary reference frame voltages $v_{ds}^*$ and $v_{qs}^*$ and are used for generating the switching signals for the pulse width modulator (PWM). The outputs of the PWM are the signals that drive the inverter. The current model generates rotor flux position and hence the slip speed (Krishnan 2001).
2.3 SPEED CONTROLLER DESIGN

The speed controller may include conventional controllers such as PI and PID controller, adaptive controllers or intelligent controllers such as fuzzy and artificial neural network based controllers. The speed controller uses the speed error between the actual speed and the command speed to generate the command torque for the induction motor drive.

2.3.1 Conventional Controllers

PI and PID controllers are the most commonly used conventional controllers because of their simple design and easy construction. The mathematical description of the PI controller is (Bose 2006)

\[ u = K_p[e(t) + \frac{1}{T_i} \int e(t) \, dt] \]  

(2.1)

where \( e = (\omega_{ref} - \omega_r) \) represents the speed error between the command speed \( \omega_{ref} \) and the actual motor speed \( \omega_r \), \( K_p \) is the proportional error constant and \( T_i \) is the integral time. Similarly the mathematical description of the PID controller can be written as

\[ u = K_p[e(t) + \frac{1}{T_i} \int e(t) \, dt + T_D \frac{de(t)}{dt}] \]  

(2.2)

where \( T_D \) is the derivative time. \( u \) represents the output of the controller which corresponds to the q-axis command current \( i_{qs}^* \) for the induction motor drive (Bose 2006).
2.3.1.1 Tuning of PI and PID Controller

The tuning of the PI and the PID controller for speed control of induction motor drive is done using Ziegler-Nichols tuning method (Nichols and Ziegler, 1963). Since this tuning method is one of the most accepted standard tuning techniques in control system design and is also simple in procedure, this method has been selected.

The step by step procedure to determine the controller parameters are as follows:

- Reduce the integrator and derivative gains to 0.
- Increase $K_p$ from 0 to some critical value $K_p = K_c$ at which sustained oscillations occur.
- Note the value $K_c$ and the corresponding period of sustained oscillation, $T_c$.
- The controller gains are now specified as follows

<table>
<thead>
<tr>
<th>Type</th>
<th>$K_p$</th>
<th>$T_I$</th>
<th>$T_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>$0.5K_c$</td>
<td>Inf</td>
<td>0</td>
</tr>
<tr>
<td>PI</td>
<td>$0.45K_c$</td>
<td>$T_c/1.2$</td>
<td>0</td>
</tr>
<tr>
<td>PID</td>
<td>$0.6K_c$</td>
<td>$T_c/1.2$</td>
<td>$T_c/8$</td>
</tr>
</tbody>
</table>

The gain constants for PI and PID Controller can be obtained from the formulas given in Table 2.1. Applying this tuning technique for the speed control system under study, the gain values are PI controller as
$K_p = 2.164$ and $K_i = 3.24$. Similarly the PID controller gain value are obtained as $K_p = 2.88$ and $K_i = 4.31$ and $K_d = 0.288$.

### 2.3.2 Fuzzy Based PI Controller

The Figure 2.2 shows the block diagram of a standard fuzzy logic controller (FLC), where the speed error $e$ and its rate of change $de$ are the input variables (Krishnan 2001). The basic FLC block is composed of fuzzification interface, fuzzy rules and inference mechanism, and defuzzification interface.

The FLC has two inputs speed error $e(k)$ and change in speed error $de(k)$ and one output $c(k)$ which represent the q-axis command current $i_{qs}^*$ for the induction motor drive. In the first stage, the crisp variables $e(k)$ and $de(k)$ are converted into fuzzy variables $E(k)$ and $dE(k)$ using the triangular membership functions shown in Figure 2.3. Each universe of discourse is divided into five fuzzy sets: NL (negative large), NS (negative small), ZE (zero), PS (positive small) and PL (positive large). Each fuzzy variable is a member of the subsets with a degree of membership varying between 0 (non-member) and 1 (full-member). In the second stage of the FLC, the fuzzy variables $E(k)$ and $dE(k)$ are processed by an inference engine that executes a set of control rules contained in a 5×5 rule bases. The associated fuzzy rule matrices of the FLC are given in Table 2.2. These rules were designed based on the dynamic behaviour of the error signal, resulting in the symmetrical matrix. This is a general rule-based design with a 2-D phase plane. Each rule is expressed in the form

$$\text{Rule : If } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z \text{ is } C.$$  

Different inference algorithms can be used to space the fuzzy set values for the output fuzzy variable $c(k)$. In this work, the max-min inference
algorithm is used, in which the membership degree is equal to the maximum of the product of $E$ and $dE$ membership degree (Krishnan 2001).

Figure 2.2 Structure of Fuzzy Logic Controller

Figure 2.3 Membership function of the Controller
Table 2.2 Fuzzy Rule Matrix

<table>
<thead>
<tr>
<th>de(k)</th>
<th>NL</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>NL</td>
<td>NL</td>
<td>NL</td>
<td>NL</td>
<td>NS</td>
<td>ZE</td>
</tr>
<tr>
<td>NS</td>
<td>NL</td>
<td>NL</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
</tr>
<tr>
<td>ZE</td>
<td>NL</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PL</td>
</tr>
<tr>
<td>PS</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
<td>PL</td>
<td>PL</td>
</tr>
<tr>
<td>PL</td>
<td>ZE</td>
<td>PS</td>
<td>PL</td>
<td>PL</td>
<td>PL</td>
</tr>
</tbody>
</table>

The output variable from the inference engine is converted into a crisp value in the defuzzification stage. Various defuzzification algorithms have been proposed in the literature. In this work, the centroid defuzzification algorithm is used, in which the crisp value is calculated as the centre of gravity of the membership function. The definition of the spread of each partition, or conversely the width and symmetry of the membership functions, is generally a compromise between dynamic and steady state accuracy. Equally spaced partitions and consequently symmetrical triangles are reasonable choices (Nasir Uddin et al 2002).

2.3.3 Fuzzy based Self-tuning PID Controller

The fuzzy based self-tuning controller is a combination of conventional fuzzy controller and PID controller. In this approach, the PID controller parameters are tuned on-line by an adaptive mechanism based on fuzzy logic speed control of induction motor drive. It employs the fuzzy inference system to tune the gain values $k_p$, $k_i$ and $k_d$ of the conventional PID controller according to the speed error ($e$) and the derivative of the speed error.
(de/dt). The structure of the fuzzy based self-tuning PID controller is shown in Figure 2.4.

![Figure 2.4 Structure of Fuzzy based Self-tuning PID Controller](image)

The fuzzy controller generates the tuning values $k_{pf}$, $k_{if}$ and $k_{df}$ for updating the gain values of the PID controller. Once the tuning values are obtained, the new parameters of the PID controller are calculated by the following equations:

\[
k_{pnew} = (k_{pmax} - k_{pmin})k_{pf} \tag{2.3}
\]

\[
k_{inew} = (k_{imax} - k_{imin})k_{if} \tag{2.4}
\]

\[
k_{dnew} = (k_{dmax} - k_{dmin})k_{df} \tag{2.5}
\]

Where $k_{pnew}$, $k_{inew}$ and $k_{dnew}$ are the updated gain values of the PID controller, $k_{pmax}$, $k_{imax}$, $k_{dmax}$ and $k_{pmin}$, $k_{imin}$, $k_{dmin}$ are the maximum and minimum values of the PID controller gains.
The membership function for the inputs $e$ and $de$ of the fuzzy controller are defined in the range of $[-1, 1]$ and the outputs are defined in the range of $[0, 1]$. The rule base for the fuzzy controller is generated using 5x5 rule matrix as given in Table 2.1. The indirect field oriented control of the induction motor drive is simulated with the fuzzy based self-tuning PID controller and the simulation results are presented in section 2.4.1.

The membership functions can be even increased to seven with 49 rules, however it increases the computational burden and there is no significant improvement in the performance of the induction motor drive. The computation is faster when tried with three membership functions and 9 rules, but when compared with five membership functions and 25 rules, the performance is poor in terms of higher overshoot and longer settling time.

2.4 SIMULATION OF THE SPEED CONTROLLERS FOR IFOC OF INDUCTION MOTOR DRIVE

The simulation of the indirect field oriented control configuration of the induction motor drive is been carried out using Matlab/Simulink software. A 1.47 kW squirrel cage induction motor is used for simulation. Conventional speed controllers like PI and PID controller and a fuzzy based self-tuning speed controller are investigated. The fuzzy tool box available in the Matlab software is used for designing the fuzzy based self-tuning PID controller. Based on the PWM technique, the switching signals are generated for the IGBT based 3-phase inverter. The induction motor parameters and the command speed $\omega_{ref}$ are used as inputs for the simulation model. The drive system is investigated under steady state and transient operating conditions. The speed and the current responses of the induction motor drive are plotted for comparison and analysis.
2.4.1 Simulation Results

The simulated speed response of the IFOC of induction motor drive using PI controller, PID controller and fuzzy based self-tuning PID controller are shown in Figure 2.5 - 2.12. The switching frequency of the PWM signal is fixed as 2 MHz. The sampling time of simulation is 2 μsec.

The starting speed response of the induction motor drive for PI, PID and fuzzy based self-tuning PID controller is shown in Figure 2.5. The motor is started at no load and command speed is fixed at 183.3 rad/sec i.e the rated speed. Simulation results show that the PI and PID speed controller suffers from overshoot and takes more settling time to reach the steady state. The fuzzy based self-tuning PID controller responded with lesser overshoot and took less time to reach the steady state value compared to PI and PID controllers.

The speed response of the induction motor drive when started with a load of 2.5 Nm at a command speed of 183.3 rad/sec for PI controller, PID controller and fuzzy based self-tuning PID controller respectively is shown in Figure 2.6. Comparing the performances, the fuzzy based self-tuning PID controller is having better time domain parameters compared to PI and PID controller.

The simulated speed response of the induction motor drive for step increase and decrease in command speed at no load is shown in Figure 2.7 for PI controller, PID controller and fuzzy based self-tuning PID controller respectively. The speed is increased from 120 rad/sec to 180 rad/sec at \( t = 1.0 \) sec and again decreased to 90 rad/sec at \( t = 1.5 \) sec. It is observed that the fuzzy based self-tuning PID controller exhibits lesser overshoot and undershoot compared to conventional PI and PID controllers.
Figure 2.5  Simulated starting response of the induction motor drive at no load for a command speed of 183.3 rad/sec for PI controller, PID controller and Fuzzy based self-tuning PID controller

Figure 2.6  Simulated starting response of the induction motor drive with a load of 2.5 Nm and a command speed of 183.3 rad/sec PI controller, PID controller and Fuzzy based self-tuning PID controller
Figure 2.7  Simulated speed response of the induction motor drive at no load for step change in command speed from 120 rad/sec to 180 rad/sec and again decreased to 90 rad/sec (a) PI controller (b) PID controller (c) Fuzzy based self-tuning PID controller
Figure 2.8  Simulated speed response of the induction motor drive when 25% of rated load is applied at $t = 1.0$ sec (a) PI controller (b) PID controller (c) Fuzzy based self-tuning PID controller
Figure 2.9  Simulated speed response of the induction motor drive started with 25 % of rated load and the load is removed at t = 1.0 sec (a) PI controller (b) PID controller (c) Fuzzy based self-tuning PID controller
Figure 2.10  Simulated starting response of the induction motor drive for doubled rotor inertia, at no load with a command speed of 180 rad/sec (a) PI controller (b) PID controller (c) Fuzzy based self-tuning PID controller
Figure 2.11  Simulated starting response of the induction motor drive for doubled stator resistance, at no load with a command speed of 180 rad/sec (a) PI controller (b) PID controller (c) Fuzzy based self-tuning PID controller
Figure 2.12 Simulated starting response of the induction motor drive for 25% reduction in mutual inductance, at 25% of load with a command speed of 180 rad/sec (a) PI controller (b) PID controller (c) Fuzzy based self-tuning PID controller
The performance of the induction motor drive is analyzed for sudden changes of load. Figure 2.8 show the speed response when the motor is started with a command speed of 180 rad/sec at no load and 25% of the rated load is \( t = 1.0 \) sec for PI controller, PID controller and fuzzy based self-tuning PID controller respectively. The PI controller and PID controller show large undershoot at the instant when the load is applied.

The speed response when the motor is started with 25 % of the rated load at a command speed of 180 rad/sec and the load is removed suddenly at \( t=1.0 \) sec for PI controller, PID controller and fuzzy based self-tuning PID controller respectively is shown in Figure 2.9. A speed overshoot is observed at the point where the load is removed. The overshoot of the speed response is large for PI and PID controller when compared to fuzzy based self-tuning controller.

The control performances of the induction motor drive system can be influenced by system uncertainties. These uncertainties include parameter variation due to change in slip speed and temperature variation due to high starting current. Hence, simulation studies have been carried for variation of motor parameters such as moment of inertia and stator resistance. The speed response at no load with a command speed of 180 rad/sec for double the rotor inertia and double the stator resistance is shown in Figure 2.10 and 2.11 respectively. The speed response at 25% of load, with a command speed of 180 rad/sec and for 25% reduction in mutual inductance is shown in Figure 2.12. The performances of the fixed gain controllers are not satisfactory under these operating conditions.
2.5 SUMMARY

The indirect field oriented control of the induction motor drive is implemented using conventional and fuzzy based controllers. Conventional PI controller, PID controller as well as fuzzy based self-tuning PID controller is simulated for the drive system and the results are presented. PI and PID controllers, though simple, have difficulties in dealing with dynamic speed tracking, parameter variation and load disturbances since the controllers’ gain values are fixed. The performance of the speed control system also depends on the slip calculation. Unfortunately, the slip calculation depends on the rotor time constant, which varies continuously according to the operating conditions. From the simulation results, it can be concluded that even after proper tuning, the conventional controllers are adversely affected by load disturbances and there is significant overshoot and disturbances in the speed responses. Therefore, they are not suitable for high performance applications. Compared to conventional PI controller and PID controller, the fuzzy based self-tuning PID controller performs better in terms of peak overshoot and settling time. The speed response of the fuzzy based self-tuning controller is found to be more robust in control system sense. However, there are still chances of improvement in the speed response for high performance applications. Since wavelet transform has the ability to control dynamic systems and there is a trend of research on the application of wavelet transform for the control of motor drives, a wavelet based speed controller for indirect field oriented control of induction motor drive is to be disused in the next chapter.